Atmospheric Modeling with High-Order Finite-Volume Methods

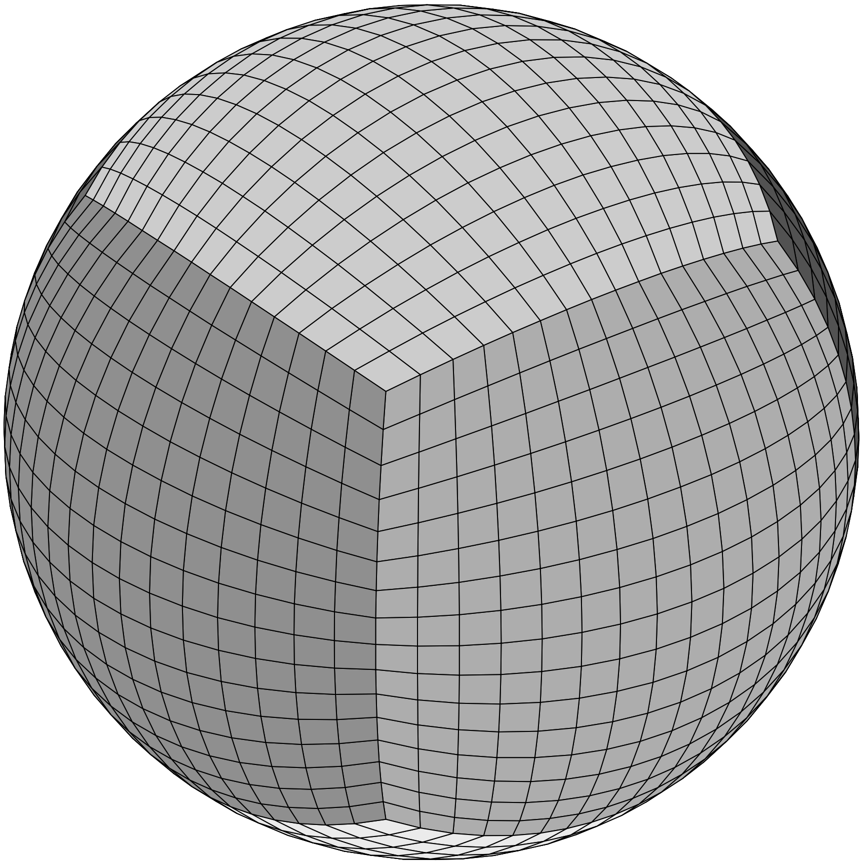
by

Paul Aaron Ullrich

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Atmospheric and Space Sciences and Scientific Computing)
in The University of Michigan
2011

Doctoral Committee:

Professor Christiane Jablonowski, Chair
Professor Smadar Karni
Professor Richard Rood
Professor Bram van Leer



To Youna

ACKNOWLEDGEMENTS

Throughout the years there have been numerous people who have helped bring this thesis to fruition. I would like to extend a special thank you to my advisor, Christiane Jablonowski for her support and insight into both the mathematics and science of atmospheric dynamics. Another special thank you goes to Bram van Leer for his seemingly unbounded knowledge on computational fluid dynamics. His insight and support made the development of this work significantly easier. In addition, many thanks go to Smadar Karni and Ricky Rood, both of whom have offered unique insights and perspectives during the development of this project.

I would also like to offer thanks to the many collaborators I have had the pleasure of working with for the past four years, in particular Peter Lauritzen at NCAR and Phil Colella at Lawrence Berkeley Lab. I look forward to working with them in the future.

A special mention goes to all the friends I have met at the University of Michigan, who have made for many enjoyable experiences both inside and outside the office. I'd like to give a shout out to Kevin, Ahmed, Jacob, Dan, Kristen, Sid, Matt and many others who have all been close friends and good company. All of you helped me keep my sanity in these few years.

Most importantly, I would like to thank my family for all of their support and encouragement. A special thank you goes to my father, who unfortunately will not be able to see my graduation – I can honestly say that by accepting nothing short of the best he led me to where I am today. Of course, thanks go to my mom for

her consistent encouragement and valuable teachings. I would also like to thank my brother Chris, whose lectures in calculus to me while I was still in elementary school drove my interest in mathematics, and my sister Carolyn who taught me not to take life too seriously. Finally, I would like to thank my wife Youna, whose companionship has made my life here a joy. Her support through stressful times with delicious cooking, hugs and games of badminton whenever needed will always be cherished.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

<u>**Table**</u>

LIST OF APPENDICES

Appendix

ABSTRACT

Atmospheric Modeling with High-Order Finite-Volume Methods

by

Paul Aaron Ullrich

Chair: Christiane Jablonowski

This thesis demonstrates the versatility of high-order finite-volume methods for atmospheric general circulation models. In many research areas, these numerical methods have been shown to be robust and accurate, and further have many properties which make them desirable for modeling atmospheric dynamics. However, there have been few attempts to implement high-order methods in atmospheric models, and none that use finite-volume methods. High-order methods are desirable for future model development due to their superior wave propagation properties and necessity when using adaptive mesh refinement.

The thesis describes in detail a hierarchy of atmospheric models that utilize high-order finite-volume methods. The hierarchy includes a 2D shallow-water model, both 2D and 3D non-hydrostatic models and a 3D non-hydrostatic dynamical core in spherical geometry. These models span atmospheric motions that range from the microscale, mesoscale to the global-scale regime while essentially leaving the underlying numerical scheme unchanged. A cubed-sphere computational grid has been chosen for the global models, due to its relative uniformity as compared with the traditional regular latitude-longitude grid. First, the thesis documents the development of a finite-volume-based remapping scheme for accurately converting data between cubed-sphere and latitude-longitude meshes. An analysis of several finite-volume-type methods in 1D for advection is then presented, with some emphasis on models with grid adaptation. Furthermore, the thesis describes the formulation of the model hierarchy that represents a gradual increase in complexity and thereby serves as a testbed. The 2D (x-y) shallow-water model on the sphere evaluates explicit time-stepping algorithms and demonstrates how to accurately handle the panel boundaries of the cubed-sphere mesh. The 2D-slice (x-z) and 3D non-hydrostatic finite-volume models in Cartesian geometry introduce an implicit-explicit time-splitting technique needed to properly handle the small grid spacings and high-speed waves in the vertical direction. Finally, a novel 3D non-hydrostatic high-order finite-volume dynamical core in cubed-sphere geometry is presented. The thesis demonstrates that high-order finite-volume methods are a viable and promising option for future atmospheric models, and an important stepping stone for next-generation atmospheric model development.

CHAPTER I

Introduction

1.1 Why do we Build Models?

In 1963, American mathematician and meteorologist Edward Lorenz published his seminal paper "Deterministic Nonperiodic Flow" (*Lorenz*, 1963). Lorenz demonstrated that even a relatively simple system of partial differential equations that arose from the governing equations of atmospheric motions can lead to chaotic nondeterministic behavior with very strong dependence on the initial conditions. Hence, he argued, without exacting knowledge of the initial conditions the atmosphere was effectively unpredictable after even a few days. Lorenz's work had triggered significant debate within the atmospheric community that still lingers to today; namely, if predicting the future behavior of the atmosphere is not only difficult, but effectively impossible, what is the point in trying to model it?

To some degree, models do not yield explanations analogous to those of rigorous theory. Yet, models cannot be classified as observational science, which cannot make future predictions based on incomplete data. In fact, since its inception, the numerical approach has been an altogether different beast. With the advent of modern numerical methods, modelers could now conduct "experiments" that were previously inaccessible. If one were interested in how the general circulation of the atmosphere would change if the Earth had no continents, for example, models could now give answers that could not be found by other methods. Over the past fifty years, atmospheric models have given us incredible insight into the regional and global influences of the changing climate. However, Lorenz's words still hold true – atmospheric models cannot determine the weather next year, but their value is instead found in answering questions about statistical properties or long term trends of global behavior. As models more closely match observations of our world, we can be reassured that we have understood the underlying equations and mechanisms that drive the climate system.

This thesis focuses on the *dynamical core* component of general circulation models (GCMs), as depicted in Figure 1.1. The dynamical core is an essential component of any large-scale model and is responsible for the solution of the fluid equations. It manages thermodynamic quantities, including density, pressure and temperature as well as the wind velocity. With the advent of modern supercomputing, massively parallel computers have become available that can now model the Earth down to scales of only a few kilometers. Most existing dynamical cores are not well-designed for computing on these scales and so there is an increasing push for the development of next-generation software for atmospheric models. This work focused on developing new technologies, as well as translating existing technologies from other fields, that would allow us to build a next-generation atmospheric model using methods which have been proven to be robust, efficient and accurate.

1.2 A Brief History of Numerical Weather Prediction (Before 1955)

Before proceeding, it is pertinent to review the history of general circulation models and the developments that have led us to the software system we use today. It is not our goal to provide a complete history of numerical methods for the atmospheric sciences. For a more complete story, we refer the reader to books such as

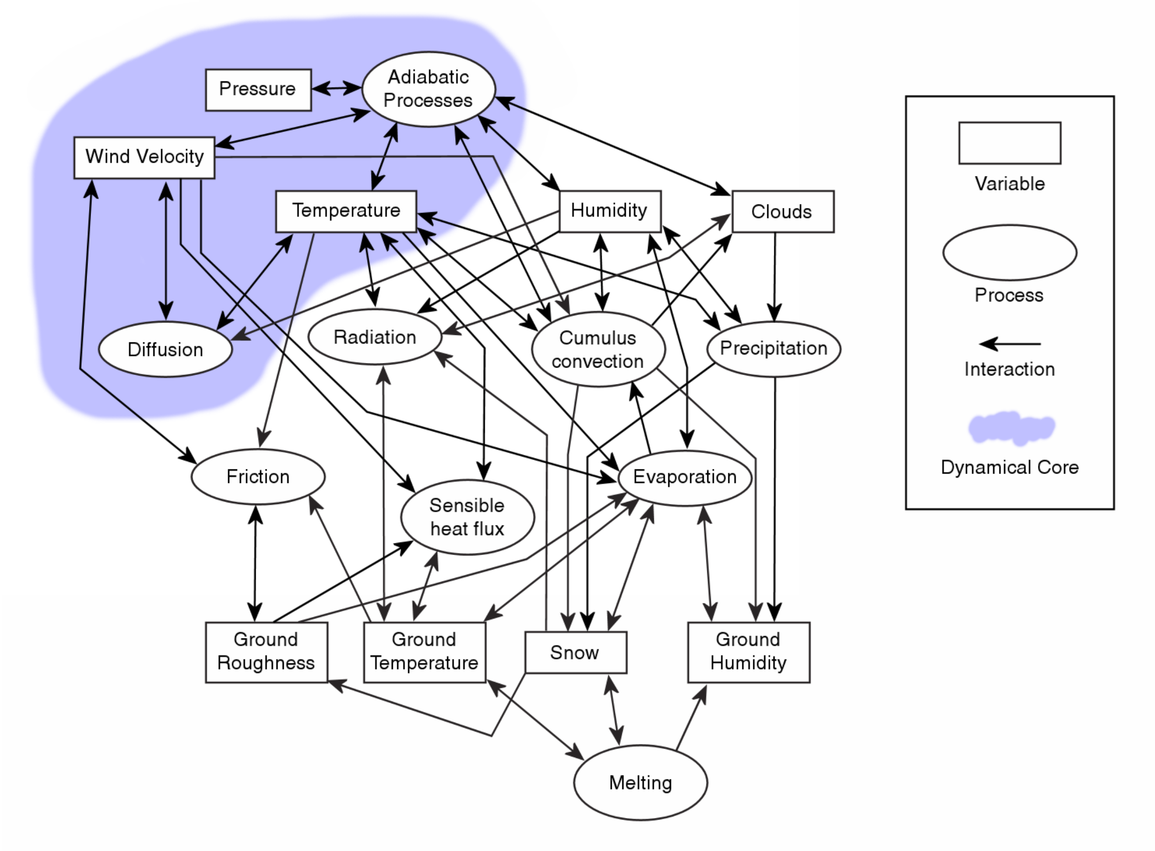


Figure 1.1: A cross-section of a general circulation model, identifying variables and interactions which compose the dynamical core (shaded region) and those which are handled by physics (unshaded region).

Edwards (2010), *Harper* (2008) or *Randall* (2000). An excellent essay by Spencer Weart on the development of general circulation models for climate is also available from `http://www.aip.org/history/climate/GCM.htm`.

The first attempts at predicting the behavior of the atmosphere occurred nearly a century and a half ago, pioneered by the work of Robert FitzRoy in the 1860s. Using only telegraph systems to relay local weather information between base stations across Europe, he produced the first synoptic charts of England and coined the term "weather forecast." Into the 1900s advancing technology led to increasingly better observation data, however, meteorologists of the time still constructed their forecasts exclusively via historical weather patterns. The idea that the atmosphere could be treated as a mathematical system was not explored until 1916 when Norwegian meteorologist Vilhelm Bjerknes introduced the first set of equations to describe motions of the atmosphere.

Although Bjerknes' first work is considered to be the cornerstone in the study of atmospheric motions, his equations were too complicated to provide insight into the fundamental dynamics of the atmosphere. In 1922, British mathematician Lewis Fry Richardson introduced a more complete numerical system for the atmosphere (*Richardson*, 1922) and with it a method for performing weather forecasting in a numerical framework. His idea, modeled after a method mathematicians referred to as finite-difference solutions, was to break up a given regional domain into a set of grid cells, each of which stored some component of the state of the atmosphere. The equations governing the atmosphere could then be discretized and evaluated to step forward in time. His greatest achievement was in putting this method into practice, performing months of tedious hand calculations with the goal of predicting the evolution of a weather system over central Europe over the period of one day. Sadly, his calculation was also a dramatic failure, as he predicted a huge rise in pressure (145 mbars) when, in fact, the pressure was more or less static. Although it would not arise until years later, his error was effectively in using an unstable choice of timestep for his calculations. Under his approach small ampltiude variations in the pressure field were amplified and led to large-scale disturbances in the solution.

Another shortfall of Richardson's approach was its complexity. As stated by Richardson himself, "the scheme is complicated, because the atmosphere itself is complicated." Even a one day forecast required thousands of tedious arithmetic calculations. With no concept of modern computing, Richardson's vision for weather prediction required tens of thousands of people to simultaneously perform "computations." Even then, results would only arrive as fast as weather occurred in reality.

In the mathematics community, work on finite-difference solutions of partial differential equations continued. In 1928, Courant, Friedrichs and Lewy published their fundamental work on stability of numerical methods (*Courant et al.*, 1928). Nonetheless, it was not until the advent of modern computing that it became feasible to perform the computations necessary to predict atmospheric motions.

In the 1940s, John von Neumann had significant success in computing the behavior of explosions using numerical methods. He could see the parallels of his explosion simulations and numerical weather prediction, and vocally advocated for the use of modern computers in numerical models of the atmosphere. Jule Gregory Charney, who had come from Carl-Gustaf Rossby's pioneering meteorology department at the University of Chicago, was recruited by Von Neumann to develop a numerical framework for weather prediction. Richardson's equations were the starting point, but Charney quickly realized that filtering of these equations was necessary to make large-scale calculations feasible. The first successful numerical weather prediction experiment finally came about in 1950, performed by Charney at Princeton University on ENIAC (*Charney et al.*, 1950). Following Richardson's approach, they divided the atmosphere over North America into grid cells with a spacing of roughly 700 km. The time step of these simulations was approximately 3 hours and the calculation was computed purely in 2D. The simulations were far from perfect, but numerically stable and had enough observed features to motivate continued research.

It did not take long for real-time numerical weather prediction to be adopted by meteorologists worldwide. The first real-time numerical weather prediction experiments were performed by the Royal Swedish Air Force Weather Service in 1954 (*Bergthorsson et al.*, 1955). In North America, the Weather Bureau established the Joint Numerical Weather Prediction Unit, which in May of 1955 began issuing real-time forecasts. Although primitive, these calculations were believed to be reasonably reliable for up to three days in advance.

Since these early simulations, the infrastructure of predicting the weather has become a resounding success. With advancing technology and computational power, it has even become an integral part of our everyday lives.

1.3 The First General Circulation Models (1955-1965)

Up until 1955, weather prediction efforts were limited to regional scales. Observational data, which were necessary for a model's initialization, were not reliable enough to initialize global models so the idea of extending weather forecasting models to global scales was perceived as unnecessary. Nonetheless, academic efforts began in 1955 to model the general circulation of the Earth. The first atmospheric general circulation model was developed by Norman Phillips at Princeton University in 1955 (*Phillips*, 1956). His computer system held a mere five kilobytes of memory, with an additional ten kilobytes of data storage. He developed an improved set of equations for the two-layer atmosphere and modeled circulation on a cylinder 17 cells high and 16 in circumference. The resulting simulation produced a plausible jet stream and a realistic-looking weather disturbance that evolved in time.

Von Neumann enthusiastically publicized Phillips' results, which quickly led to government funding for a long-term project to model the global circulation. In 1955, Joseph Smagorinsky, who at the time worked at the U.S. Weather Bureau, directed the development of a general circulation model of the entire three-dimensional atmosphere built from the primitive equations (*Smagorinsky*, 1983). In 1958, Smagorinsky invited Syukuro Manabe, who had studied at Tokyo University, to join the team. The contributions of Manabe led to the development of physical parameterizations for radiative transfer, as well as ocean, land and ice exchange processes. It took until 1965 before Manabe's group had completed, to some degree, a three-dimensional global model that incorporated nine vertical levels (*Manabe and Smagorinsky*, 1965). This model would be the forerunner for the GCMs developed under the banner of the Geophysical Fluid Dynamics Laboratory (GFDL, Princeton).

Simultaneous with the development of the U.S. Weather Bureau model, another model was under development at the University of California at Los Angeles (UCLA). Motivated by Phillips' 1956 paper, Yale Mintz undertook an ambitious program to advance the development of GCMs. He recruited Akio Arakawa, also from Tokyo University, to develop the mathematical foundation for another general circulation model. By 1964 they had developed a two-layer model that, unlike the Manabe model, incorporated realistic orography over the entire globe (*Mintz*, 1965; *Arakawa*, 1970). Their work ended up being a forerunner for many future modeling groups, including the Goddard Institute for Space Sciences (GISS) model. Work from both Mintz and Manabe was later incorporated into the European Centre for Medium-Range Weather Forecasts (ECMWF) model.

By the end of the 1960s, half a dozen GCMs were already in development, including teams at the UK Met Office and the National Maritime Center (NMC) and a team at the National Center for Atmospheric Research (NCAR), led by Warren Washington and yet another Tokyo University graduate, Akira Kasahara (*Kasahara and Washington*, 1967). For a "family tree" of GCM development, we refer to `http://www.aip.org/history/climate/xAGCMtree.htm`.

1.4 Algorithmic Development (1965-2000)

In 1965 a panel of the U.S. National Academy of Sciences reported on recent developments in GCMs, observing that global models were largely successful at reproducing gross features of the atmosphere. Nonetheless, there were significant shortfalls in these models that could only be addressed by substantially increased computational power (*National Academy of Sciences*, 1966). Equally important, however, was the development of algorithms that produced improved results with fewer computations.

Two problems that were a proverbial thorn in the side of GCM developers were the poles of the simulation grid. Up until 1965 GCMs had largely used a latitude-longitude plane for their simulations on the sphere. Although this grid is perhaps the most natural choice, it is inefficient computationally since it leads to small physical grid spacing near the poles. A reduced-resolution grid was proposed by *Kurihara* (1965), which kept the latitude-longitude mesh but removed grid cells in the longitudinal direction at high latitudes to maintain grid uniformity. Grids based on an icosahedral projection were proposed by *Sadourny et al.* (1968) and *Williamson* (1968), and referred to as geodesic grids. A grid based on a cubic projection and known as a "cubed-sphere grid" was later developed by *Sadourny* (1972). These grids were an elegant solution to the pole problem, but in many cases were not competitive with existing models. As a consequence, uniform grids were largely not used in operational atmospheric models until the mid-1990s. One notable exception was the GFDL SKYHI model (*Fels et al.*, 1980), which used a reduced resolution grid throughout the 1980s.

In the 1970s other innovations aimed at the finite-difference methods that had been used in most GCMs. Instead of dividing the planet's surface into a grid of elements, the equations of motion were rewritten in terms of spherical harmonics. This "spectral transform" technique simplified many computations and allowed the pole problem to be sidestepped entirely, but was only feasible with faster computers. Nonetheless, the spectral transform method was very popular in the GCM community and is even in use in models today. For example, the ECMWF model still uses the spectral transform method for forecasts, having adopted it into their forecasting systems in 1983. However, the spectral transform method is not without its disadvantages. Firstly, monotonicity and positivity are not guaranteed – that is, numerical errors can result in significant spurious overshoots and undershoots in flow variables, which may result in negative tracer concentrations. Secondly, dispersive errors in the spectral model can lead to "Gibbs ringing" in regions where solutions are not perfectly smooth, and so high-frequency waves must be explicitly damped. Finally, spectral transform methods require the use of global Fourier transforms, which reduces the efficiency of these models on parallel computational architectures. Nonetheless, the spectral transform method proved to be a very effective technique for global atmospheric models.

By the mid-1980s, substantial progress had been made in other research areas that were also tied to hydrodynamics. Aerospace and astrophysics, in particular, had developed their own algorithmic treatments of the fluid equations, although their work often dealt with very-high-speed flows. Interestingly, much of this work was again due to the involvement of John von Neumann, who had driven interest in numerical methods for atmospheric phenomena. Finite-volume methods had become very popular in these fields, but were largely unheard of in the atmospheric sciences. Finite-volume methods had their own fundamental history, tracing their roots back to the work of *Godunov* (1959), who had developed conservative finite-volume methods for modeling shockwaves, and subsequently was involved in developing a class of methods which are today referred to as Godunov-type methods. This work was later extended by Bram van Leer in a series of papers (*van Leer*, 1974, 1977, 1979) which described an approach for extending Godunov's method to second-order accuracy. This work was simultaneous with that of *Boris and Book* (1973), who developed flux-corrected-transport (FCT) methods (extended by *Zalesak*, 1979). Later, *Colella and Woodward* (1984) developed the piecewise-parabolic method (PPM) for gas hydrodynamics problems, which was a Godunov-type finite-volume method of third-order accuracy. Other prominent figures in computational fluid dynamics research at this time included Peter Lax (*Lax and Wendroff*, 1960), Robert MacCormack (*MacCormack*, 2003), Philip Roe (*Roe*, 1981), Amiram Harten (*Harten et al.*, 1983) and Stanley Osher (*Osher and Sethian*, 1988).

In 1987, Richard Rood wrote a fundamental paper comparing many numerical methods for advection (*Rood*, 1987). Advection had been a significant problem in atmospheric models, since it required monotonicity preserving filters and was generally poorly handled by spectral transform methods. His work drew heavily on research from other fields, which generally had only minimal contact with the atmospheric sciences. In particular, he advocated for finite-volume methods, which preserved positive-definite results, guaranteed conservation of mass and maintained high accuracy. This work contributed prominently to the development of a finite-volume dynamical core with Shian-Jiann "S.J." Lin (*Lin and Rood*, 1996; *Lin and Rood*, 1997; *Lin*, 2004) at NASA, which used a staggered grid and a piecewise-parabolic-type reconstruction procedure. This dynamical core is perhaps one of the most well-known dynamics models today, and remains well-used within GFDL, NASA's Goddard Earth Observing System Model, Version 5 (GEOS-5) and NCAR's Community Atmosphere Model (CAM).

1.5 The Modern State of Numerical Methods for Atmospheric Models (2000-Today)

By the end of the 20th century, computational power was continuing to increase exponentially. The advent of supercomputing had led to massively parallel systems, which consisted of thousands or more interconnected processors. Models running on these systems can now reach resolutions on the order of 10 kilometers or less – far beyond anything Richardson could have dreamt of! However, it was also clear that many of the operational GCMs were not well-designed to truly harness these systems. Communication between processors was beginning to be a bottleneck, since many prominent numerical methods (spectral transform methods, as well as finite-difference or finite-volume methods which are built on a latitude-longitude grid) require a substantial amount of communication between processors at each timestep.

Dozens of atmospheric models are in use today, with applications ranging from experimental science to operational forecasting. Recent developments in modeling atmospheric dynamics have tended away from the latitude-longitude grid, instead returning to uniform grids, including the icosahedral or cubed-sphere grids. This choice has allowed for the design of models which can be run at very fine resolutions on vast parallel computing systems. The drive towards finer and finer resolutions has also forced many models to re-evaluate the basic equations that governs their dynamics. At scales less than ten kilometers the hydrostatic approximation is no longer valid, and so models typically resort to using the full non-hydrostatic primitive equations. A list of many models that are either under development or operational is given below.

1.5.1 Hydrostatic Models

Hydrostatic models approximate the vertical structure of the atmosphere to be in a state of hydrostatic balance. Under this approximation, the vertical velocity is no longer a prognostic quantity but is instead determined using the computed pressure and divergence of the horizontal velocity field. This approximation works well when the horizontal grid spacing is much larger than the vertical grid spacing, and so has been a mainstay of atmospheric models for the past several decades.

- **CAM Eulerian model:** (NCAR, Boulder, Colorado) (*Collins et al.*, 2004) A hydrostatic model that uses the spectral transform method with triangular truncation on a Gaussian grid with hybrid η vertical coordinate (*Simmons and Burridge*, 1981).

- **CAM/GEOS finite-volume model:** (NCAR, Boulder, Colorado and NASA Goddard Space Flight Center, Greenbelt, Maryland) (*Lin and Rood*, 1996; *Lin*, 2004) A hydrostatic model that uses a monotonic and potentially third-order piecewise-parabolic finite-volume reconstruction (overall second-order in space due to flux evaluation) on a latitude-longitude grid, mixed Arakawa D/C-grid staggering and floating Lagrangian vertical coordinate. A polar filter is employed to remove grid-scale noise in the polar regions.

- **CSU model:** (Colorado State University, Fort Collins, Colorado) (*Ringler et al.*, 2000) A third-order finite-differences based model with icosahedral hexagonal grid.

- **NASA/GFDL finite-volume cubed-sphere model:** (NASA Goddard Space Flight Center, Greenbelt, Maryland / Geophysical Fluid Dynamics Laboratory, Princeton, New Jersey) (*Putman and Lin*, 2007; *Putman and Lin*, 2009) As CAM/GEOS finite-volume model except on a cubed-sphere grid, removing the need for polar filtering.

- **NOAA Flow-following finite-volume Icosahedral Model (FIM):** (*Lee et al.*, 2006; *Henderson et al.*, 2010) A second-order finite-volume model on the icosahedral grid using flux-corrected transport with semi-Lagrangian vertical levels.

- **German Weather Service GME model:** (*Majewski*, 1998; *Majewski et al.*, 2002) A hydrostatic model that uses second-order finite-differences on an icosahedral grid with unstaggered variables and hybrid η vertical coordinate.

- **GISS ModelE:** (NASA Goddard Institute for Space Studies, New York, NY) (*Schmidt et al.*, 2006) A hydrostatic model that uses second-order centered finite-differences plus the quadratic upstream method of *Prather* (1986) (semi-Lagrangian discontinuous Galerkin) for advection. The model is built on a latitude-longitude grid with B-grid staggering and hybrid η vertical coordinate.

- **High-Order Method Modeling Environment (HOMME) models:** (NCAR, Boulder, Colorado) (the Spectral Element Atmosphere Model, SEAM) (*Fournier et al.*, 2004; *Taylor et al.*, 2008) A set of hydrostatic models built on finite-element-type compact methods, including spectral element and discontinuous Galerkin in the horizontal and second-order finite-differences with hybrid η coordinate in the vertical.

1.5.2 Non-hydrostatic Models

Non-hydrostatic models make no approximation to the vertical structure of the atmosphere and so allow for features such as horizontal transport of vertical momentum.

- **Icosahedral Non-hydrostatic (ICON) GCM:** (Max-Planck Institute for Meteorology, Hamburg, Germany and DWD) (*Wan*, 2009; *Gaßmann*, 2010) Initially, ICON was developed as a hydrostatic prototype model, but has recently been updated to use the full non-hydrostatic equations. This model uses a finite-difference method on an icosahedral grid with Arakawa C-grid staggering and hybrid η vertical coordinate.

- **ECMWF Integrated Forecast System (IFS) model:** (*Wedi et al.*, 2010) A non-hydrostatic model using the spectral transform method with semi-Lagrangian transport and built on the reduced latitude-longitude grid. Note that IFS was developed as a hydrostatic model and has been in operational use for decades. Recently, it was extended to a non-hydrostatic version that is currently undergoing testing.

- **Global Environmental Multiscale (GEM) model:** (*Yeh et al.*, 2002) An implicit second-order semi-Lagrangian model using an Arakawa C-grid and hydrostatic-pressure based vertical coordinate, and built on a latitude-longitude grid.

- **MIT GCM:** (Massachusetts Institute of Technology, Boston, Massachusetts) (*Adcroft et al.*, 2004) A finite-volume cubed-sphere-grid model with height-based vertical coordinate and shaved-cell topography.

- **Model for Prediction Across Scales (MPAS):** (*Skamarock et al.*, 2010) (NCAR, LANL/DOE) A new non-hydrostatic global model designed to supercede the Weather Research and Forecasting (WRF) model. Uses conservative 2nd-order finite-differences on a icosahedral hexagonal mesh with Arakawa C-grid staggering, height-based vertical coordinate and 3rd-order split-explicit Runge-Kutta time integration.

- **Non-hydrostatic ICosahedral Atmospheric Model (NICAM):** (*Tomita and Satoh*, 2004) Developed in cooperation with the Center for Climate System Research (CCSR, Japan). This non-hydrostatic atmospheric model uses 2nd-order finite-differences on an icosahedral grid with horizontally unstaggered variables and vertically staggered vertical velocity. The vertical coordinate is height-based.

- **Non-hydrostatic Icosahedral Model (NIM):** (*Govett et al.*, 2010) (Earth System Research Laboratory, NOAA) A Riemann-solver-based finite-volume model on an icosahedral hexagonal grid with monotonic Adams-Bashforth third-order multistep time integrator and height-based vertical coordinate.

- **Ocean-Land-Atmosphere Model (OLAM):** (*Walko and Avissar*, 2008) (Duke University) A non-hydrostatic, finite-volume based model using an icosahedral grid with Arakawa C-grid staggering, height-based vertical coordinate and shaved cells for representing topography.

- **UK Met Office Unified Model:** (*Davies et al.*, 2005; *Staniforth and Wood*, 2008) A non-hydrostatic model using a conservative finite-difference approach on a latitude-longitude grid with height-based vertical coordinate. Both the shallow and deep-atmosphere equations as well as the hydrostatic and non-hydrostatic equations are supported in this model.

1.6 Future Trends in Model Development

Looking forward, it is clear that substantial work remains to be done in developing next-generation general circulation models. Scalability on massively parallel systems is an essential requirement of any future model, and so should be a cornerstone of any future designs. Further, new technologies will play an increasingly important role, such as graphical processing units (GPUs), which have the potential to dramatically speed up existing simulations.

1.6.1 The Cubed-Sphere Grid

As mentioned previously, the icosahedral (geodesic) grid has many desirable properties, but since it relies on either triangles or hexagons (and pentagons) to form each grid cell, it is more difficult to optimize organization of computational grid cells than on a more structured grid. In 1996, the cubed-sphere grid was revived by *Ronchi et al.* (1996) and later used as the basis for a shallow-water model by *Rančić et al.* (1996). Since then, shallow-water models have been developed using the cubed-sphere that utilize finite-volume methods (*Rossmanith*, 2006; *Ullrich et al.*, 2010), multi-moment finite-volume (*Chen and Xiao*, 2008), the discontinuous Galerkin method (*Nair et al.*, 2005) and the spectral element method (*Taylor et al.*, 1997). The spectral element method was successfully extended to a full hydrostatic atmospheric model (the Spectral Element Atmosphere Model, SEAM) (*Fournier et al.*, 2004), which is part of the High-Order Method Modeling Environment (HOMME). HOMME incorporates both the spectral element and discontinuous Galerkin methods, and has proven to scale efficiently to hundreds of thousands of processors. More recently, the GFDL finite-volume dynamical core has been modified to utilize a cubed-sphere grid (*Putman and Lin*, 2009; *Putman and Suarez*, 2009), and has been demonstrated to also be very effective at high resolutions.

1.6.2 Adaptive Mesh Refinement

Although computational power has increased substantially in recent years, large parallel systems are still required to properly resolve many important atmospheric features. In order to reduce the computational burden of these fine-scale simulations, the next generation of atmospheric models will likely need to rely on adaptive mesh refinement (AMR). Mesh refinement refers to the addition of grid elements to regions with small-scale features so as to reduce errors that arise due to insufficient resolution. Static mesh refinement implies that the grid does not change in time, but regions of significant dynamical behavior, such as the equator, are initially enhanced. Dynamic mesh refinement is similar, but allows the grid to change over the process of the simulation.

Mesh refinement can be categorized into either using conformal or non-conformal grids. Conformal grids use a smooth mapping between some initial regular mesh (such as a Cartesian or hexagonal grid) and the desired mesh. These grids include stretched grids, wherein grid elements are slowly distorted so as to enhance resolution in certain locations (*Fox-Rabinovitz et al.*, 1997, 2006). Non-conformal grids, on the other hand, are not the product of such a conformal mapping. Block adaptive grids, for instance, are an example of a non-conformal grid (*Berger and Oliger*, 1984; *Berger and Colella*, 1989; *Skamarock et al.*, 1989; *St-Cyr et al.*, 2008). Non-conformal meshes are largely preferred for dynamical mesh refinement since they only require local modification of the mesh when additional resolution is added or removed.

Grid reflection is a prominent issue with non-conformal meshes that will be discussed in this thesis. When a wave packet propagates through a resolution discontinuity, the discontinuous modification of the dispersion relation leads to behavior analogous to what one would expect at a discontinuity in the physical properties of the fluid. As a consequence, part of the wave is transmitted and the remainder reflected. The artificial reflection of the wave packet can lead to the phenomenon of trapped waves in regions of fine grid resolution, which can in turn affect the accuracy of the solution.

1.7 Outline of Thesis

In this thesis we present our ongoing work on high-order finite-volume methods in the context of atmospheric GCMs. In many research areas, these methods have been demonstrated to be robust and accurate, and further have many properties which make them desirable for modeling atmospheric dynamics.

This thesis is organized as follows. In Chapter 2 we introduce the cubed-sphere geometry in the context of the Geometrically Exact Conservative Remapping (GECoRe) scheme, which was developed for conservative remapping between the cubed-sphere and latitude-longitude grid. Chapter 3 pursues a theoretical analysis of the properties of various high-order finite-volume methods, particularly in the context of refined grids. In Chapter 4 we develop a high-order finite-volume shallow water model on the cubed-sphere and compare several methods for computing element fluxes using

Riemann solvers. The problem of horizontal-vertical aspect ratio in non-hydrostatic atmospheric models in Cartesian geometry is tackled in Chapter 5, wherein we propose an implicit-explicit Runge-Kutta-Rosenbrock (RKR) approach for coupling horizontal and vertical motions while maintaining high-order-accuracy and a timestep limit that is only proportional to the horizontal grid spacing. In Chapter 6 we discuss an extension of the high-order finite-volume shallow-water model on the cubed-sphere grid to a fully non-hydrostatic shallow-atmosphere model utilizing the implicit-explicit splitting approach. Finally, conclusions and future work are presented in Chapter 7. A substantial portion of the mathematics behind the cubed-sphere grid and high-order finite-volume methods can be found in the Appendices.

CHAPTER II

Geometrically Exact Conservative Remapping

2.1 Introduction

Land, ocean and atmosphere components of coupled climate system models are often implemented on different spherical grids, individually designed to enhance the accuracy or capture features unique to their respective settings. Historically, the regular latitude-longitude (RLL; see Table 2.1 for a complete list of acronyms used in this chapter) grid has been the predominant choice for global atmospheric models, but problems associated with the polar singularity persist, and hence this grid is not well-suited for highly scalable atmospheric models. Much interest in recent years has been instead directed towards the development of atmospheric solvers defined on more isotropic spherical grids. For example, the cubed-sphere grid, which divides the polar singularities among eight weaker singularities located at the corners of a cube, and is otherwise highly scalable on parallel architectures. The cubed-sphere grid was originally introduced by *Sadourny* (1972), and more recently reintroduced by *Ronchi et al.* (1996) and *Rančić et al.* (1996) with equiangular grid spacing and orthogonality. For the land component, however, the RLL grid does not pose polar singularity problems as is the case for the atmosphere (with the current complexity of land models). Neither does the land model seem to be susceptible to scalability problems since most of the computation is in vertical columns rather than in the

Table 2.1: List of acronyms used in this chapter.

Acronym	
GECoRe	Geometrically Exact Conservative Remapping
()-M	suffix stands for Monotone filter applied
PCoM	Piecewise Constant method
PLM	Piecewise Linear method
PPM	Piecewise Parabolic Method
RLL	Regular Latitude-Longitude
ABP	Alpha-Beta-Panel (Equiangular cubed-sphere Coordinates)
SCRIP	Spherical Coordinate Remapping and Interpolation Package
CaRS	Cascade Remapping between Spherical grids

horizontal. Hence for the foreseeable future the RLL grid seems to be a viable and convenient grid for land model components.

An intricate problem introduced by defining the model components on different spherical grids is that the exchange of information between the grids is non-trivial and requires a regridding algorithm. In a coupled climate system model it is paramount that the regridding process is not a spurious source or sink for first-order moment variables such as mass. To prevent the generation of unphysical negative and/or large values, the regridding must also be shape-preserving/monotone for mixing-ratio related variables. Regridding with these constraints, conservation and monotonicity, is a non-trivial problem if higher than first-order accuracy is desired.

The regridding problem is not only limited to a static grid-to-grid information transfer setting. The problem is essentially the same for finite-volume advection schemes where the mass-transport into a given cell is given in terms of integrals over overlapping areas. In fact, methods developed for advection schemes can be readily applied in grid-to-grid regridding problems such as articulated by *Margolin and Shashkov* (2003). A major difference between the advection regridding problem and static grid-to-grid regridding is that the source or target grid is not static for advection problems. Hence the regridding algorithm must be able to deal with a large class of source or target grids changing dynamically at each time step. For grid-to-grid regridding the problem is static, facilitating certain parts of the algorithm. For example, the regridding problem can be optimized for specific grid pairs. On the other hand, the advection problem is usually constrained by Courant numbers and the number of source and target grid cells are identical which constrains the overlap regions. On the contrary, grid-to-grid regridding does not have that constraint so many source grid cells can overlap a particular target grid cell and vice versa.

A strategy for doing conservative regridding without *ad hoc* conservation fixers is to reconstruct a sub-grid-cell distribution in each source grid cell with conservation as a constraint and then integrate the sub-grid-cell distributions for the respective source grid cells over the overlap areas. This process of conservative transfer of variables between grids is referred to as remapping or rezoning. Depending on the source and target grid cell geometries the overlap regions over which one must integrate can be very complex. Hence direct integration on the sphere of the overlap areas seems like an almost impossible task in terms of algorithmic complexity. Note, however, that it has been done in Cartesian geometry in the context of advection (see *Rančić* 1992). The problem can be greatly simplified by making use of the powerful mathematical theorem, Gauss's divergence theorem, that converts area integrals into line integrals (see *Dukowicz and Kodis* 1987). This is the approach taken in the most widely used regridding software in the climate community called the Spherical Coordinate Remapping and Interpolation Package (SCRIP, *Jones* 1999). Also in the algorithm presented in this chapter we make use of Gauss's divergence theorem.

In order to perform the line integrals on the sphere one usually makes simplifying assumptions about the cell sides. For example, the sides of the grid cells are approximated by straight lines in (λ, θ)-space in SCRIP. This obviously leads to exact cell wall representations for the RLL grid but other spherical grids such as the cubed-sphere grids do not share that property (see Figure 2.1). The remapping algorithm's

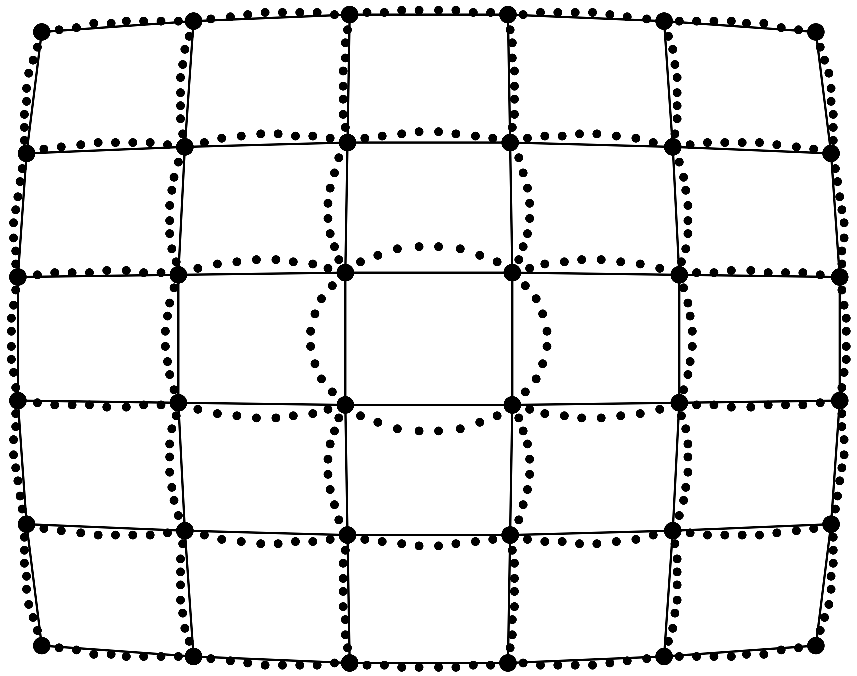


Figure 2.1: The cell boundaries of the cubed-sphere south polar panel plotted in Cartesian coordinates. SCRIP approximates cell edges by connecting the cell vertices (filled circles) with straight lines in RLL coordinates (dotted lines). The solid lines are the exact ABP cell walls that are great circle arcs.

inability to represent the cell sides exactly is here referred to as the *geometric error* (*Lauritzen and Nair* 2007; hereafter referred to as LN2007). In other words, the geometric error is the deviation from exact closed-form integration along the cell walls. Unfortunately, exact closed-form integration is not always achievable, except for a small set of source and target grids. As a consequence, generic conservative remapping schemes, such as SCRIP and CaRS, must apply potentially crude geometrical approximations in order to yield a conservative and computationally efficient scheme (Figure 2.1). Perhaps serendipitously, we find that complete removal of geometric error via exact closed-form integration is possible for the RLL grid and gnomonic cubed-sphere grids, and hence these grids immediately lend to the development of a geometrically exact conservative scheme.

Assuming small geometric error, the order of the remapping algorithm is determined by the accuracy of the sub-grid cell reconstruction. The errors introduced by the reconstruction are referred to as the *derivative error*. The simplest reconstruction is a piecewise constant (first-order) representation in each source grid cell. This method is inherently monotone but is excessively damping at least in idealized remapping problems. First-order reconstructions are, for example, used in NCAR's Coupled Climate System Model (CCSM3, *Collins et al.* 2006) through SCRIP for remapping variables requiring conservation. The effect on climate by using higher-order remapping in a coupled climate system model is unknown (as far as the authors are aware) but it seems instinctive to speculate that it could have a significant effect; especially in areas where remapping is done from a coarse to a fine grid.

Unfortunately, high-order reconstructions are not inherently monotone, making it harder to achieve conservation, monotonicity and high-order accuracy simultaneously. LN2007 choose to apply the cascade remapping approach to achieve monotonocity with high-order reconstructions. This algorithm is referred to as CaRS (monotone and conservative cascade remapping between spherical grids). In CaRS the remapping problem is split into two one-dimensional problems and hence only one-dimensional limiters are needed to guarantee monotonicity, making it relatively easy to impose shape-preservation simultaneously with higher-order reconstructions. Unfortunately, the CaRS method is inherently prone to geometric errors, although it seems to compensate for these via the higher-order reconstructions that are easily and efficiently applied in one dimension. The geometric errors can be reduced by artificially increasing the resolution in areas of higher geometric error (see LN2007 for details). A necessary, but insufficent, condition for completely eliminating the geometric error is to approximate the grid cells in a two-dimensional manner rather than the dimensional split approach such as done in CaRS. However, a fully two-dimensional approach requires fully two-dimensional integrals, reconstructions and limiters. This greatly increases the complexity of the problem, especially when aiming for higher-order remapping. SCRIP has the option of performing second-order remapping if the user supplies the gradient in latitude-longitude coordinates. In theory SCRIP could be extended to higher-order by including the curvature and high-order derivatives in the line-integral computations but that route was not explored by *Jones* (1999) and is not a trivial extension.

In this chapter we present a new conservative remapping method between gnomonic cubed-sphere grids and the RLL grids hereafter referred to as GECoRe (geometrically exact conservative remapping). GECoRe uses Gauss's divergence theorem to convert area integrals into line integrals. The line integrals are exact (to machine precision) for given polynomial sub-grid-cell reconstructions since the lines along which the line integrals are computed exactly coincide with grid lines on the cubed-sphere and RLL grid. Consequently the geometric error is completely eliminated in GECoRe. We also use up to third-order-accurate reconstructions and apply limiters to obtain monotonicity thereby obtaining high-order accuracy and shape-preservation simultaneously in two dimensions.

The paper is organized as follows. In Section 2 we introduce the mathematical basis for GECoRe, that is, how to compute the potentials needed to convert area integrals into line integrals, how line integrals are computed exactly and how the fully two-dimensional reconstructions are approximated. Section 3 covers some practical considerations such as finding intersections between line segments of the two grids. The accuracy of GECoRe is assessed in Section 4 by comparing standard error measures for remapping analytical functions with GECoRe, SCRIP and CaRS between equi-angular cubed-sphere and RLL grids with different resolutions. A summary is given in Section 5.

2.2 Geometrically exact conservative remapping

In the context of finite volume methods, we are given the cell-averaged value of a scalar field for each cell in the source grid, denoted by $\overline{f}_n$. The remapping problem then reduces to finding corresponding cell-averaged values in the target grid, denoted $\overline{f}_k$, that accurately represent the underlying scalar field. Throughout this chapter we will use the subscript n to denote a quantity corresponding to a cell of the source grid and the subscript k to denote a quantity corresponding to a cell of the target grid.

2.2.1 Source and target grids

Of the numerous choices for the cubed-sphere, we will focus on the equiangular cubed-sphere, where a sphere is decomposed into six identical regions (panels or faces) and the grid lines on each panel are defined by equispaced central angles (see Fig. 2.2). Hereafter, we will refer to the equiangular cubed-sphere as the ABP (alpha, beta, panel) coordinate system, in reference to coordinates on the sphere being identified by the three-element vector (α, β, n_p), where $(\alpha, \beta) \in [-\pi/4, \pi/4]$ are the coordinates on each panel and $n_p \in \{1, 2, 3, 4, 5, 6\}$ is the panel index. A mathematical description of the properties of the equiangular cubed-sphere is given in appendix D.

Although we focus on the equiangular projection, we should emphasize that the GECoRe method is trivially extended to any cubed-sphere grid which is composed of grid lines parallel to the panel edges, *i.e.* gnomonic cubed-sphere grids. Notably, conformal or spring-dynamics grids, where grid lines are "warped" near the corners, do not fall into this category (see, for example, *Putman and Lin* (2007) for a review of several types of cubed-sphere grids). Grids with grid lines that are not parallel to panel sides can also be captured via this scheme by replacing the exact line-integral formulas with Gaussian quadrature approximations to the integrals in gnomonic coordinates. Such an approach is taken in the tracer transport scheme described in *Lauritzen et al.* (2008). Of course, the geometric error will not be completely eliminated if the cell sides are not great spherical arcs. Note that in such a situation the geometric error could be reduced by approximating the cell sides by several great-spherical arc segments.

We will also use Gnomonic coordinates interchangeably with equiangular coordinates. Gnomonic coordinates (x, y) are defined in terms of equiangular coordinates via

$$x = a \tan\alpha, \qquad y = a \tan\beta, \qquad (\alpha,\beta) \in \left[-\tfrac{\pi}{4}, \tfrac{\pi}{4}\right]^2, \tag{2.1}$$

where a is a constant denoting the edge length of a concentric cube, which, without loss of generality we will take to be equal to unity. The Gnomonic projection is important for our analysis since any straight line in the Gnomonic projection corresponds to a *spherical arc* on the surface of the sphere. A mathematical description of the properties of the Gnomonic cubed-sphere is given in appendix C.

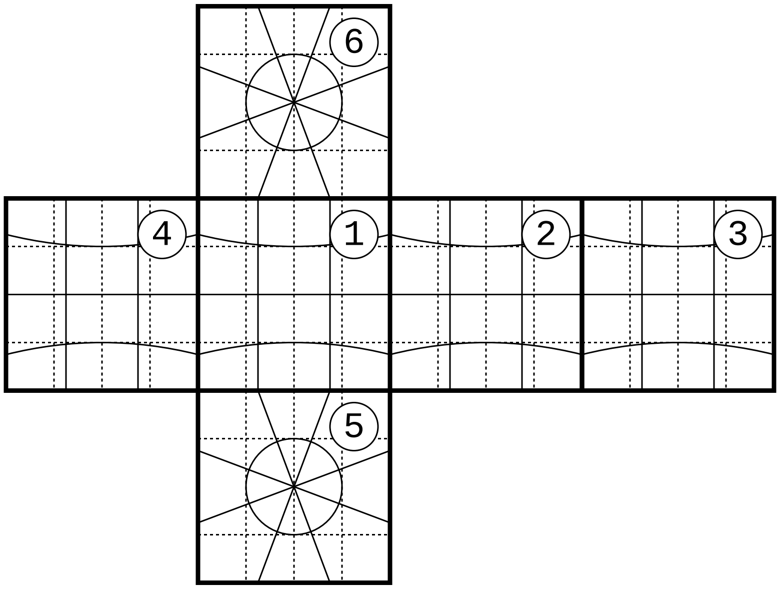


Figure 2.2: An illustration of the regular latitude-longitude (RLL) grid (thin solid lines) and cubed-sphere grid (dotted lines). Thick lines mark the boundaries of each panel, distinguished by the panel index given in the upper-right corner. By convention we choose for the southern and northern polar panels to have indices 5 and 6, respectively.

2.2.2 Overview of the method

Typically, a conservative remapping scheme is one that satisfies the *global conservation condition*,

$$\int\limits_{A} f_{\text{target}} dA = \int\limits_{A} f_{\text{source}} dA, \tag{2.2}$$

where f_{target} and f_{source} are the global scalar field on the target grid and source grid, respectively. Here the integral is taken over the entire grid surface A (in our case, the surface of a sphere). The stricter *local conservation condition* states that for every cell k on the target grid, the scalar field must satisfy

$$\overline{f}_k = \frac{1}{A_k} \int\limits_{A_k} f dA, \tag{2.3}$$

where $\overline{f}_k$ denotes the area-averaged scalar field, f the global piecewise reconstruction on the source grid and A_k the area of cell k. Note that the local conservation condition can be trivially demonstrated to be a sufficient condition for the global conservation condition. Now, if cell k in the target grid overlaps N cells in the source grid, one can write (2.3) as

$$\overline{f}_k = \frac{1}{A_k} \sum_{n=1}^{N} \int\limits_{A_{nk}} f_n dA, \tag{2.4}$$

(see *Jones* 1999 equation (3)) where A_{nk} is the area of the source grid cell n that is overlapped by the destination cell k, and f_n is the local value of the scalar field in grid cell n (see Fig. 2.3). That is, the averaged value in the destination cell is equal to the area-normalized contribution from all overlapping cells on the source grid.

A GECoRe type remapping scheme can generally be obtained at any order, with the order of the method generally depending on the order of the sub-grid scale reconstruction within each source volume. In general, for a remapping scheme of order h,

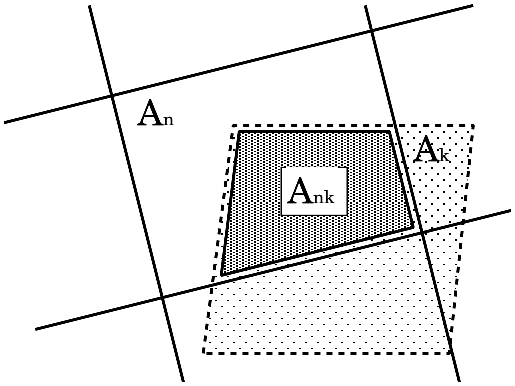


Figure 2.3: An example of a quadrilateral target grid cell A_k that overlaps several source grid cells. The region overlapped by both A_k and A_n is denoted by A_{nk}.

the sub-grid scale reconstruction takes the form

$$f_n(x,y) = \sum_{p+q<h} a_n^{(p,q)}(x-x_0)^p(y-y_0)^q, \tag{2.5}$$

where the reconstruction coefficients $a_n^{(p,q)}$ are constants and x_0 and y_0 denote the x and y components of the source-grid cell centroids, defined by

$$x_0 = \frac{1}{A_n}\int\limits_{A_n} xdA, \qquad y_0 = \frac{1}{A_n}\int\limits_{A_n} ydA. \tag{2.6}$$

In practice, these quantities are computed by transforming the area integrals to line integrals via Gauss's divergence theorem (as discussed later). In order to obtain the desired order of accuracy of the method, the reconstruction coefficients must also be obtained via a suitably accurate method. The reconstruction (2.5) must also yield the cell-averaged value of the source volume when integrated over the entire source volume, *i.e.*

$$\overline{f}_n = \sum_{p+q<h} a_n^{(p,q)} \frac{1}{A_n}\int\limits_{A_n} (x-x_0)^p(y-y_0)^q dA. \tag{2.7}$$

In appendix A we present reasonable choices of the reconstruction coefficients that lead to first-, second- and third-order methods.

The conservative remapping scheme that follows from (2.4) and (2.5) can then be written as

$$\overline{f}_k = \sum_{n=1}^{N} \sum_{p+q<h} a_n^{(p,q)} w_{nk}^{(p,q)}, \tag{2.8}$$

where, following *Jones* (1999), we have defined the mesh-dependent weights via

$$w_{nk}^{(p,q)} = \frac{1}{A_k} \int\limits_{A_{nk}} (x - x_0)^p (y - y_0)^q dA. \tag{2.9}$$

The form (2.8) is particularly meaningful, as it emphasizes the separation of the purely reconstruction-dependent coefficients $a_n^{(p,q)}$ and the purely mesh-dependent sub-cell weights $w_n^{(p,q)}$.

Following *Dukowicz and Kodis* (1987), we compute the weights (2.9) by converting them into line integrals using the divergence theorem. Consider an arbitrary vector field $\mathbf{\Psi}$ defined in terms of ABP unit basis vectors $(\mathbf{e}_\alpha, \mathbf{e}_\beta)$ as $\mathbf{\Psi} = \Psi_\alpha \mathbf{e}_\alpha + \Psi_\beta \mathbf{e}_\beta$. In general 2D curvilinear coordinates the divergence is given by

$$\nabla \cdot \mathbf{\Psi} = \frac{1}{\sqrt{g}} \left[\frac{\partial}{\partial x^1} (\sqrt{g}\ \Psi^1) + \frac{\partial}{\partial x^2} (\sqrt{g}\ \Psi^2) \right], \tag{2.10}$$

where Ψ^1 and Ψ^2 are the contravariant components of the vector $\mathbf{\Psi}$ and g is the determinant of the metric. Hence, specifically for cubed-sphere (see (D.8) and (D.9)) we have

$$\nabla \cdot \mathbf{\Psi} = (\rho^3 \cos^2 \alpha \cos^2 \beta) \left[\frac{\partial}{\partial \alpha} \left(\frac{\Psi_\alpha}{\rho \cos \beta} \right) + \frac{\partial}{\partial \beta} \left(\frac{\Psi_\beta}{\rho \cos \alpha} \right) \right], \tag{2.11}$$

where

$$\rho = \sqrt{1 + \tan^2 \alpha + \tan^2 \beta}. \tag{2.12}$$

After integrating (2.11) over the sub-cell A_{nk} and assuming sufficiently smooth boundaries, we can write the resulting expression in the form

$$\int\limits_{A_{nk}} \nabla \cdot \mathbf{\Psi}\ dV = \int\limits_{\beta_1}^{\beta_2} \int\limits_{\alpha_1(\beta)}^{\alpha_2(\beta)} \frac{\partial}{\partial\alpha}\left(\frac{\Psi_\alpha}{\rho\cos\beta}\right) d\alpha\ d\beta + \int\limits_{\alpha_1}^{\alpha_2} \int\limits_{\beta_1(\alpha)}^{\beta_2(\alpha)} \frac{\partial}{\partial\beta}\left(\frac{\Psi_\beta}{\rho\cos\alpha}\right) d\beta\ d\alpha, \quad (2.13)$$

where α_1, α_2, $\beta_1(\alpha)$ and $\beta_2(\alpha)$ represent the boundaries of the domain of integration. Then, on applying the fundamental theorem of calculus, we obtain the divergence theorem for cubed-sphere coordinates,

$$\int\limits_{A_{nk}} \nabla \cdot \mathbf{\Psi} dV = -\oint\limits_{\partial A_{nk}} \left[\frac{\Psi_\alpha}{\rho\cos\beta} d\beta + \frac{\Psi_\beta}{\rho\cos\alpha} d\alpha\right], \quad (2.14)$$

where the contour integral is taken in the counter-clockwise direction around the boundary of a given overlapping volume A_{nk}, here denoted by ∂A_{nk}. Note that the Jacobian term obtained by expanding the area integral (2.13) cancels with the $\sqrt{g}$ from the divergence (2.10) and so does not appear in the final form of the contour integral. The spatial curvature instead comes into play when solving for $\mathbf{\Psi}$ via (2.10).

In order to apply the divergence theorem to compute the weights associated with each line segment over an integrable scalar field $\phi(\alpha, \beta, n_p)$, we must first obtain a potential $\mathbf{\Psi}$ associated with that field – namely, one that satisfies

$$\nabla \cdot \mathbf{\Psi} = \phi. \quad (2.15)$$

Note that for a given ϕ, there does not exist a unique potential $\mathbf{\Psi}$, but a family of potentials $\mathbf{\Psi}$ that satisfy (2.15). However, one can obtain a unique potential on imposing $\Psi_\alpha = 0$ and choosing any constant of integration for the potential to equal zero. This limitation is largely by convention, as we could also choose $\Psi_\beta = 0$, for example. However, we find that integration on the RLL grid is generally easier upon imposing the former constraint, and so we will henceforth apply $\Psi_\alpha = 0$ when deriving each potential. Further, we restrict our attention to solving integrals of the form (2.9) – noting that the integrand of (2.9) can be expanded as terms of the form $x^p y^q$ (using (2.1)) – and so use the notation $\mathbf{\Psi}^{(p,q)}$ to denote the potential associated with the scalar field $\tan^p \alpha \tan^q \beta$. That is, we use (2.11) to define $\mathbf{\Psi}^{(p,q)}$ via the differential equation

$$\frac{\partial}{\partial \beta} \left(\frac{\Psi_\beta^{(p,q)}}{\rho \cos \alpha} \right) = \frac{\tan^p \alpha \tan^q \beta}{\rho^3 \cos^2 \alpha \cos^2 \beta}, \quad \text{with} \quad \Psi_\alpha^{(p,q)} = 0. \tag{2.16}$$

Note that one may solve (2.16) in terms of either equiangular or Gnomonic coordinates, which are connected via the relation (2.1). In either case we will obtain an identical expression for the potential.

Since our search algorithm will provide a list of line segments, rather than a list of contours, a computational implementation of (2.14) will take the form

$$\int_{A_{nk}} \tan^p \alpha \tan^q \beta dA = -\sum_s I_s^{(p,q)}\big|_{(\partial A_{nk})_s}, \tag{2.17}$$

where the summation is taken over all line segments s along the boundary of sub-cell A_{nk}. Here, $I_s^{(p,q)}$ is shorthand notation for the antiderivative over the potential field obtained from (2.16),

$$I_s^{(p,q)} = \int \frac{\Psi_\beta^{(p,q)}}{\rho \cos \alpha} d\alpha, \tag{2.18}$$

and hence it is evaluated at the endpoints of each line segment. A detailed presentation of the calculations and implementation details required for the first-, second- and third-order accurate schemes are given in appendix A.

2.2.3 Summary of the GECoRe Algorithm

A GECoRe remapping scheme is initialized as follows:

1. Perform a search on the source and target grids, classifying line segments by type (*i.e.* constant α, constant β, constant latitude or constant longitude) and

keeping track of their endpoints and orientation. Each line segment should then be associated with at most one ABP cell and one RLL cell.

2. Using (2.9) calculate the weights $w_{nk}^{(p,q)}$ associated with each line segment over the fields $\tan^p \alpha \tan^q \beta$, for $p + q < h$, where h is the order of the method (this leads to $\frac{1}{2}h(h+1)$ weights per line segment). The weights of each line segment are computed by simply evaluating the associated antiderivative $I_s^{(p,q)}$ at each endpoint (see Appendix A). Note that one can save memory and online computation time by instead storing the sum of all weights for a given overlap cell rather than the weights for individual line segments.

Once the GECoRe scheme is initialized for a particular mesh pair, the resulting initialization data can be saved to a file for later use. The actual remapping is then performed as follows:

1. Calculate the reconstruction coefficients $a_s^{(p,q)}$ associated with the scalar field, potentially using neighbouring cell values.

2. Use the weights $w_{nk}^{(p,q)}$ computed in the initialization step, along with (2.8) to compute the remapped field in each of the target grid cells.

2.3 Practical considerations

In this section we present issues relating to the implementation of the GECoRe scheme.

2.3.1 Search algorithm

Since we are restricted to RLL and cubed-sphere grids, the search algorithm for finding line segments is dramatically simplified when compared to the SCRIP algorithm. In fact, this knowledge of the coordinate systems allows us to exactly calculate the line segment endpoints up to machine precision. The proposed technique involves first searching along longitude and latitude lines to compute intersection points, binning each line accordingly depending on its corresponding ABP cell. Then a search within each cell can be performed to obtain all lines of constant α and β within a RLL cell. This binning procedure results in memory locality of line segments associated with a given RLL cell, and hence is optimal for remappings from the RLL grid to ABP grid. A similar algorithm can be performed to obtain line segments binned by ABP cell, or the results from the forward algorithm can simply be resorted to obtain the desired result. Note that special attention must be paid to special cases, such as coincident lines, so as to avoid double-counting of line segments.

2.3.2 Spherical coordinates

Similar to the SCRIP algorithm, certain aspects of the spherical coordinate system introduce additional problems when applied in practice. Unlike SCRIP, the pole points do not pose any particular problem since all calculations are performed along the surface of the cubed-sphere, which has no special treatment of the pole points. However, when calculating line segment weights the multiple-valued longitude coordinate in spherical coordinates must be taken into account. Antiderivatives which require the evaluation of the longitude coordinate at each endpoint must ensure that only the “shortest” distance between longitudes is used. Similarly, additional checks must be performed to ensure antiderivatives which require the evaluation of an arctangent have endpoints evaluated along the same branch of the arctangent function. Failure to take into account either of these factors will result in spurious factors of π being introduced into the calculation over line segments where the longitude coordinate becomes discontinuous.

2.3.3 Extensions to higher orders

Although it is not proven here, symbolic computations have shown that the line segment weights $w_{nk}^{(p,q)}$ can be computed in exact closed-form up to any choice of p and q. However, as mentioned earlier, the number of weights (and reconstruction coefficients) that need to be computed for a method of order h is quadratic in h, and hence the resulting computation becomes increasingly infeasible at higher orders. Further, on increasing the order of the scheme, one finds that the resulting antiderivatives also become increasingly complicated expressions (observe the differences between antiderivatives for the first-, second- and third- order schemes given in appendices A.1, A.2 and A.3, respectively).

2.3.4 Parallelization considerations

The non-locality of this algorithm required during on-line calculations is largely constrained to calculating the reconstruction coefficients associated with each cell (which requires a stencil size that increases with the order of accuracy of the method). In this sense we conclude that the GECoRe scheme is potentially highly parallelizable, for example by using parallelization techniques currently employed in an existing finite-volume model.

2.3.5 Bisected Elements

Since the sub-grid-scale reconstruction is computed in gnomonic coordinates, special consideration is required for RLL elements which span multiple cubed-sphere panels. In this case, the RLL element is split into two sub-elements which will have their own sub-grid-scale reconstruction. This process is described in detail in appendix B.

2.4 Results

The new remapping algorithm (GECoRe) has been implemented for both RLL to cubed-sphere and cubed-sphere to RLL remapping schemes and tested on a variety of analytical fields. For comparison, we have provided results from SCRIP and CaRS for the piecewise constant first-order reconstruction (PCoM), piecewise linear second-order reconstruction (PLM) and piecewise parabolic third-order reconstruction (PPM). Particular interest should be paid to comparing the results from SCRIP and GECoRe, since much of the underlying structure of the algorithms in these two cases are directly comparable. We note, however, that the method referred to in this chapter as SCRIP PPM is an extension of the SCRIP scheme of *Jones* (1999) to high-order accuracy. Here we have implemented this scheme by using the PPM method of *Colella and Woodward* (1984) to compute left and right edge values (ϕ_L and ϕ_R, respectively) in each cell, and hence use these values for reconstructing the gradient within each cell. As a consequence, this method is not a true PPM method, in that the sub-grid scale reconstruction is not composed of parabolic terms. As of the current time, no true PPM implementation of SCRIP is believed to exist.

2.4.1 Test cases

Our analysis mirrors the approach of LN2007, in that we consider three idealized test cases and computed error measures for both equiangular cubed-sphere to RLL remapping and vice versa.

Following *Jones* (1999) and LN2007 a relatively smooth function resembling a spherical harmonic of order 2 and azimuthal wavenumber 2 (see Fig. 2.4a),

$$\psi = 2 + \cos^2\theta\cos(2\lambda), \qquad (Y_2^2) \tag{2.19}$$

and a relatively high frequency wave similar to a spherical harmonic of order 32 and azimuthal wavenumber 16 (see Fig. 2.4b),

$$\psi = 2 + \sin^{16}(2\theta)\cos(16\lambda), \qquad (Y_{32}^{16}) \tag{2.20}$$

are used. These waves are useful for testing the performance of the algorithm for a large-scale well-resolved field as well as a higher-frequency wave in the midlatitudes with relatively rapidly changing gradients. In addition, as in LN2007, we test all three schemes with the dual stationary vortex fields (*Nair and Machenhauer* 2002), since this test leads to significant variation of the field over the cubed-sphere corners (see Fig. 2.4c). The analytical form of this field is given by

$$\psi = 1 - \tanh\left[\frac{\rho'}{d}\sin(\lambda' - \omega' t)\right], \qquad (V_X) \tag{2.21}$$

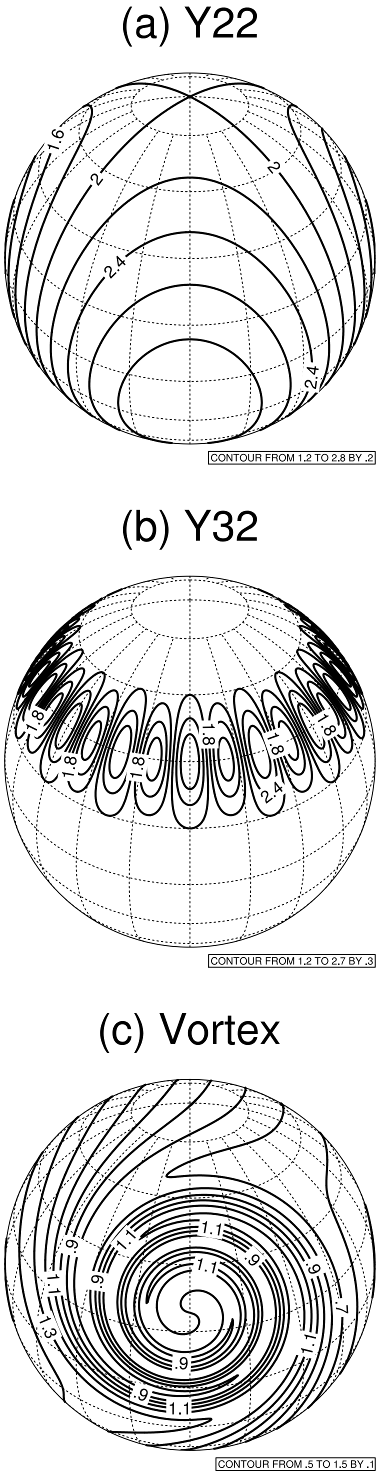
where the radius $\rho' = r_0 \cos\theta'$, with angular velocity

$$\omega'(\theta') = \begin{cases} 0 & \text{if } \rho' = 0, \\ \frac{V_t}{\rho'} & \text{if } \rho' \neq 0, \end{cases} \tag{2.22}$$

and normalized tangental velocity

$$V_t = \frac{3\sqrt{3}}{2}\text{sech}^2\rho' \tanh\rho'. \tag{2.23}$$

The (λ', θ') refer to a rotated spherical coordinate system with a pole located at (λ_0, θ_0). Following LN2007 we choose $(\lambda_0, \theta_0) = (0, 0.6)$, $r_0 = 3$, $d = 5$ and $t = 6$.


(a) Y22

(b) Y32

(c) Vortex

Figure 2.4: Contours of the analytical function (a) Y_2^2, (b) Y_{32}^{16}, and (c) the vortex fields with one of the vortices centered about $(\lambda_0, \theta_0) = (0, 0.6)$. Dotted lines show the regular latitude-longitude grid.

2.4.2 Error measures

The performance of the algorithm is quantified using standard error measures:[1]

$$l_1 \equiv \frac{I\left(\left|\overline{\psi}_{num} - \overline{\psi}_{exact}\right|\right)}{I\left(\overline{\psi}_{exact}\right)}, \tag{2.24}$$

$$l_2 \equiv \frac{\sqrt{I\left[\left(\overline{\psi}_{num} - \overline{\psi}_{exact}\right)^2\right]}}{\sqrt{I\left[\left(\left|\overline{\psi}_{exact}\right|\right)^2\right]}}, \tag{2.25}$$

$$l_\infty \equiv \frac{\max\left(\left|\overline{\psi}_{num} - \overline{\psi}_{exact}\right|\right)}{\max\left(\left|\overline{\psi}_{exact}\right|\right)}, \tag{2.26}$$

where I is the global integral

$$I(\overline{f}) = \sum_n \overline{f}_n A_n. \tag{2.27}$$

The numerically generated "exact" solution

$$\overline{f}_n = \frac{1}{A_n} \int\limits_{A_n} f dA, \tag{2.28}$$

is computed by fourth-order Gaussian quadrature. One finds that the error measure l_1 tends to identify the error in large-scale features of the field, whereas as $p \to \infty$ the errors that are described by the norm become increasingly localized.

2.4.3 Calculation of reconstruction coefficients

Calculation of the reconstruction coefficients $a^{(p,q)}$ that describe the sub-grid cell reconstruction is required for the second- and third- order schemes. Recall that the grid is an equiangular cubed-sphere grid that, when translated to Gnomonic coordinates, leads to cell centroids that are far from equidistant. As a consequence,

[1] Note that the ℓ_2 error employed in LN2008 corresponds to $(\ell_2)^2$.

if applied in Gnomonic coordinates, an equidistant discrete derivative operator will lead to large derivative errors. We examine two possible solutions to this problem: First, we can compute the derivatives in equiangular coordinates using an equidistant discrete derivative operator and then apply a stretching factor to obtain the derivatives in Gnomonic coordinates. Second, we can use a non-equidistant discrete derivative operator in Gnomonic coordinates. These methods will be compared, in terms of the resulting error measures, in section 2.4.2.4.5.

We briefly discuss the method of computing the reconstruction coefficients via the non-equidistant fitting in Gnomonic coordinate space. In general, these coefficients can be computed by fitting a parabola through the neighbouring centroids, where we have assumed that these centroids take on the cell-averaged value, and extracting the reconstruction coefficients from the quadratic coefficients. If we define

$$\Delta x_L = x_{i-1} - x_i, \quad \text{and} \quad \Delta x_R = x_{i+1} - x_i, \tag{2.29}$$

it follows that a parabola $p(x)$ fitted through the points (x_{i-1}, y_{i-1}), (x_i, y_i) and (x_{i+1}, y_{i+1}) will satisfy

$$\left(\frac{\partial p}{\partial x}\right)_i = \frac{(y_{i-1})(\Delta x_R)^2 - (y_i)\left((\Delta x_R)^2 - (\Delta x_L)^2\right) - (y_{i+1})(\Delta x_L)^2}{(\Delta x_R)(\Delta x_L)(\Delta x_R - \Delta x_L)}, \tag{2.30}$$

and

$$\frac{1}{2}\left(\frac{\partial^2 p}{\partial x^2}\right)_i = \frac{(y_{i-1})(\Delta x_R) - (y_i)\left(\Delta x_R - \Delta x_L\right) - (y_{i+1})(\Delta x_L)}{(\Delta x_R)(\Delta x_L)(\Delta x_L - \Delta x_R)}. \tag{2.31}$$

Note that the discretized derivatives (2.30) and (2.31) reduce to the usual central difference discretization in the case of equispaced grid points ($\Delta x_L = -\Delta x_R$). A discretization for the third-order cross term $(\partial^2 f / \partial x \partial y)_n$ can be easily calculated on repeatedly applying (2.30) in each coordinate direction. It is simple but mathematically intensive to extend this method to higher orders by fitting a quartic or

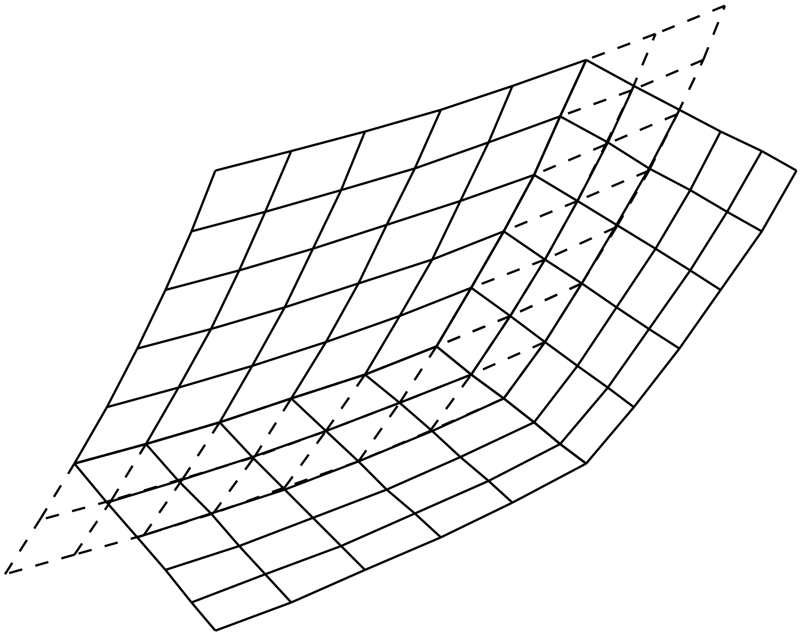


Figure 2.5: A depiction of the halo region along the boundary of the top panel (dashed lines), showing the overlap with cells of the neighbouring panels. Observe that accurate modeling of the halo region only requires 1D interpolation for this choice of grid.

higher-order curve, and so the resulting formula is not presented here.

Although both the parabolic fit (3-point) and quartic fit (5-point) can be used to derive reconstruction coefficients for the third-order scheme, we have chosen to use the 3-point stencil for the second-order scheme and the 5-point stencil for the third-order scheme. This choice is made since the parabolic fit corresponds most closely to a piecewise-linear reconstruction, whereas the quartic fit corresponds most closely to the piecewise-parabolic reconstruction of *Colella and Woodward* (1984).

Note that when applying the discretized derivative operator in ABP coordinates, halo regions must be provided for cells along the boundary of each panel. These halo cells should correspond to the cells that would be obtained by extending each panel outward, overlapping cells of the neighbouring panels (see Fig. 2.5). Hence, halo cells do not exactly correspond to boundary cells on the neighbouring panels, and their cell-averaged values must be obtained via a 1D remapping. Note that this 1D remapping does not require that the conservation criteria be fulfilled, since the area-averaged property of the interior cells is satisfied for any choice of reconstruction coefficients. For our purposes, we obtained the best accuracy from a straightforward fourth-order non-conservative cubic fit.

2.4.4 Discussion

The error measures associated with remapping from the equiangular cubed-sphere grid to RLL grid are given in Figures 2.6 and 2.7, for a high-resolution cubed-sphere grid ($N_c = 80$ grid lines on each panel) and a medium-resolution cubed-sphere grid ($N_c = 40$) mapped to a RLL grid with $N_\lambda = 128$ longitudes and $N_\theta = 64$ latitudes. The results of remapping from the same RLL grid ($N_\lambda = 128, N_\theta = 64$) to a high-resolution cubed-sphere grid ($N_c = 80$) are given in Fig. 2.8.

It should be emphasized that, in each case, the GECoRe method should perform *at least as well as* SCRIP, since both methods lead to a derivative error that should be roughly identical for PCoM and PLM. The two methods deviate only in their geometric error, in that the geometrically exact techniques used for GECoRe lead to geometric error that is roughly on the order of machine epsilon. It should be noted that slight deviations from machine epsilon occur in GECoRe due to poorly conditioned function evaluations in some of the antiderivatives (that is, calculations involving the difference of two nearly equal floating point numbers, or evaluations of arcsin or arccos near ± 1), but for our purposes we can assume these deviations are

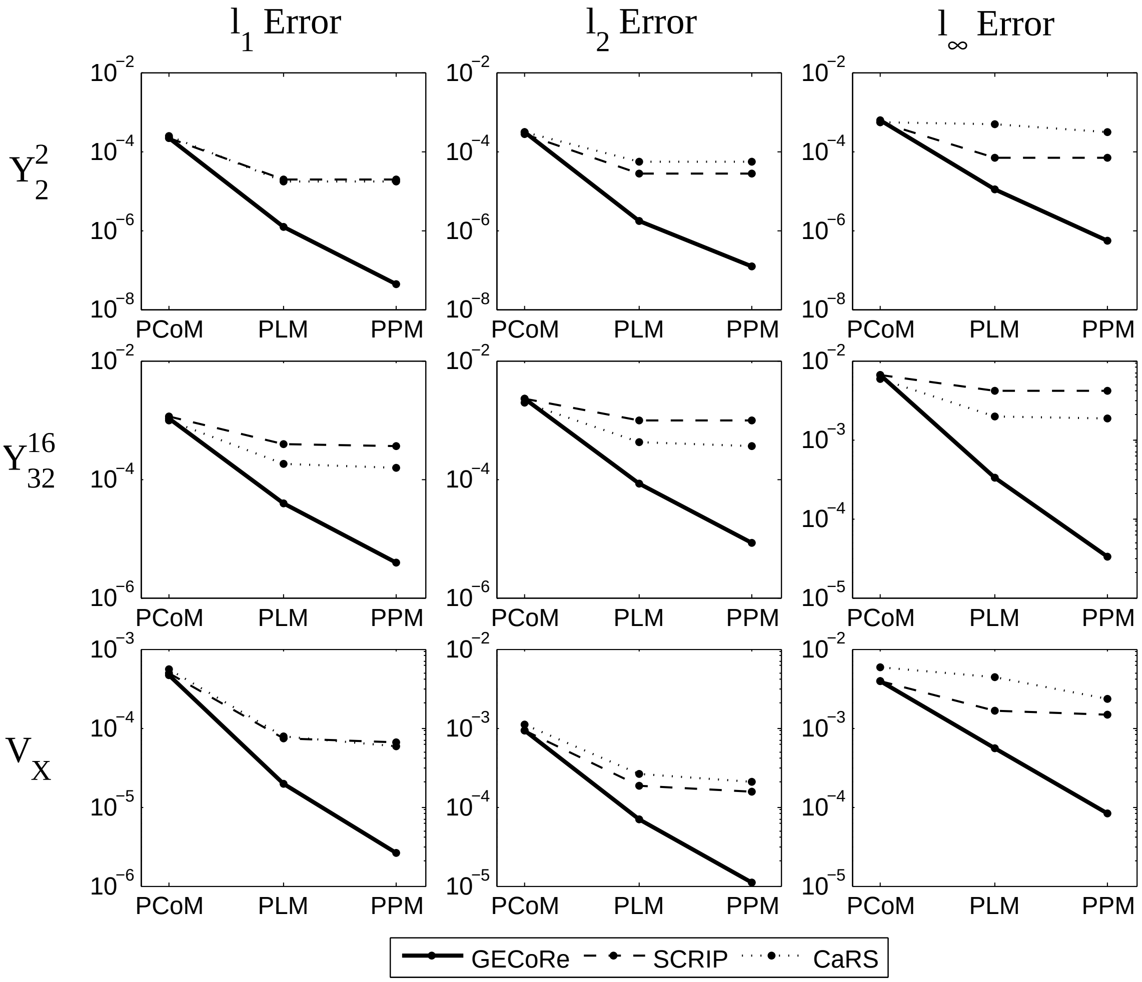


Figure 2.6: Performance measures for the remapping of Y_2^2, Y_{32}^{16} and the idealized vortices (V_X) from a medium-resolution ABP grid ($N_c = 80$) to a RLL grid ($N_\lambda = 128$, $N_\theta = 64$) using GECoRe, SCRIP and CaRS with piecewise constant (PCoM), piecewise linear (PLM) and piecewise parabolic (PPM) reconstructions.

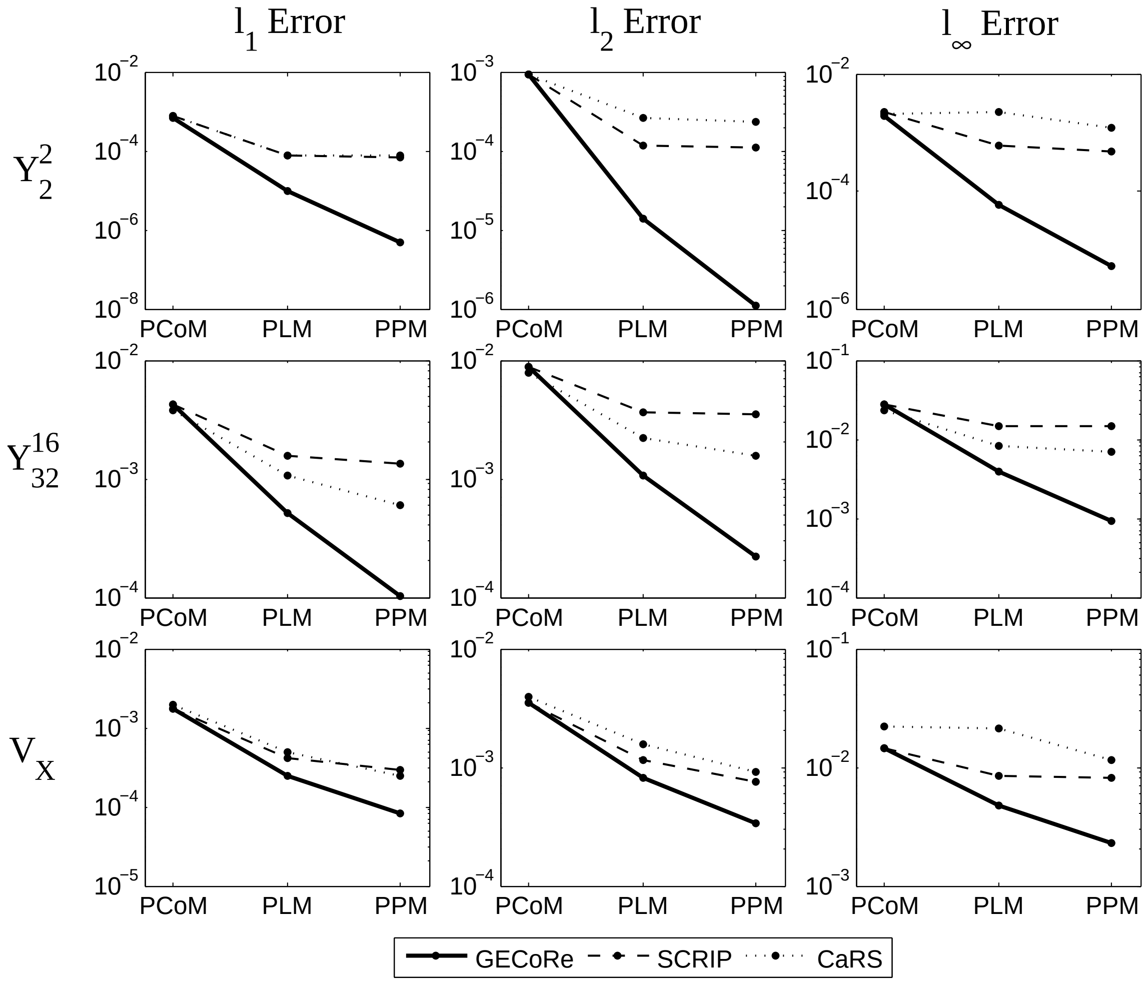


Figure 2.7: As Fig. 2.6, except with $N_c = 40$.

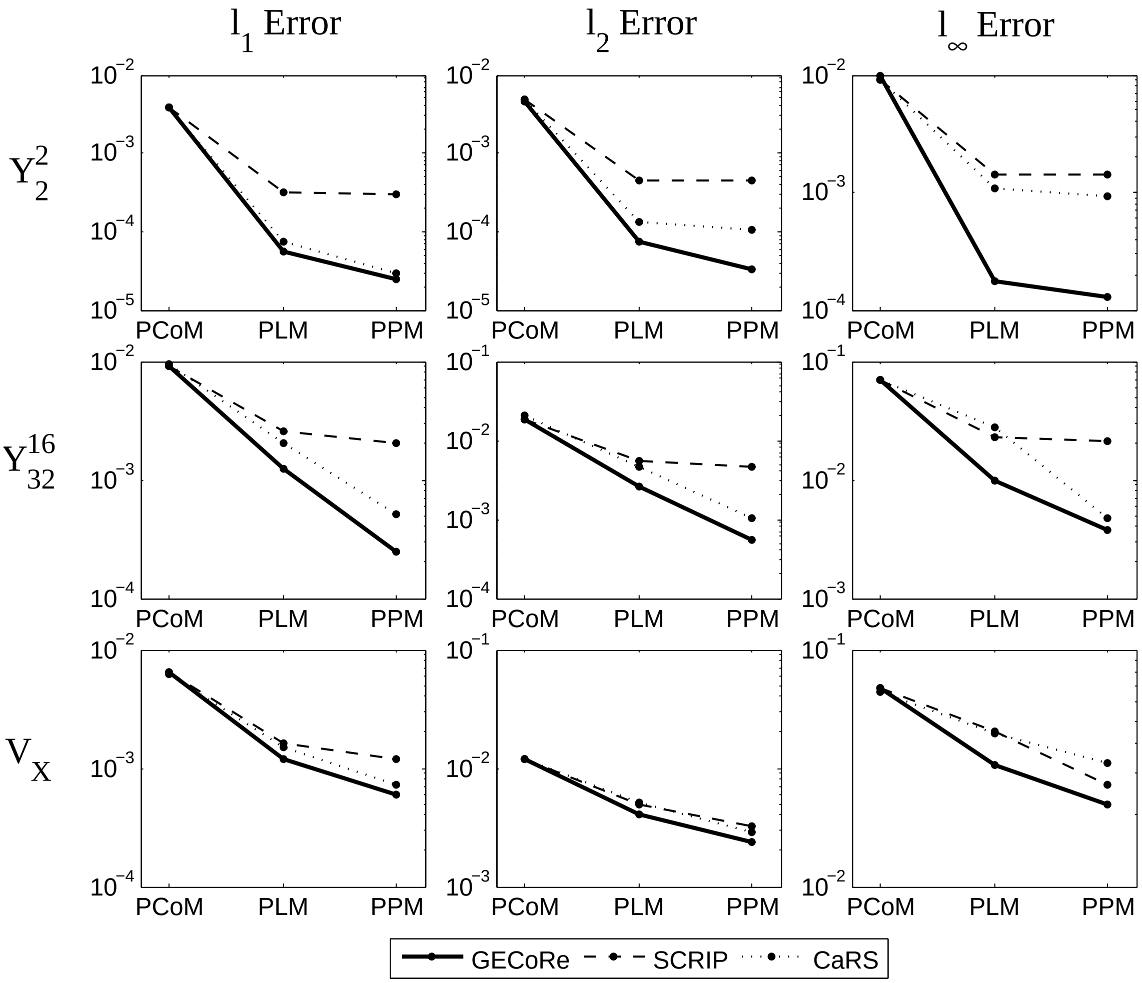


Figure 2.8: As Fig. 2.6, except remapping from a RLL grid ($N_\lambda = 128, N_\theta = 64$) to an ABP grid ($N_c = 80$).

effectively negligible.

As expected, all error norms for SCRIP, CaRS and GECoRe tend to decrease when going from PCoM to PLM to PPM. We expect that extending these methods by including the piecewise cubic scheme (PCM, as done for CaRS in LN2008) and higher-order reconstructions will lead to smaller, and perhaps worthwhile, improvements in the accuracy of the method (Figs. 2.6, 2.7, 2.8).

For the first-order piecewise constant method, we observe nearly identical behaviour for GECoRe and SCRIP since, in both cases, line segments are effectively integrated along constant fields. Hence, small perturbations in the geometrical orientation of the line segments will not lead to significant differences in the resulting line integral.

However, we find that the error measures in GECoRe and SCRIP deviate significantly for the second- and third-order methods. The effect of geometric error is clearly apparent in our calculations, as GECoRe produces results that are often one or two orders of magnitude better than the associated error from SCRIP. The results are particularly apparent for the smooth field Y_2^2, where each of the error measures shows an improvement of two to four orders of magnitude.

Comparing the spatial distribution of error for the three schemes (see Fig. 2.9), we observe that the error for SCRIP and CaRS tends to be most significant near the corners, whereas GECoRe has a much more uniform distribution of the resulting error. This result reflects the lack of a contribution from geometric error in the GECoRe scheme, which tends to dominate near the singularities of the cubed-sphere grid. Hence, we can further conclude that, compared to SCRIP and CaRS, the GECoRe scheme tends to be affected less strongly by the shape of the underlying grid.

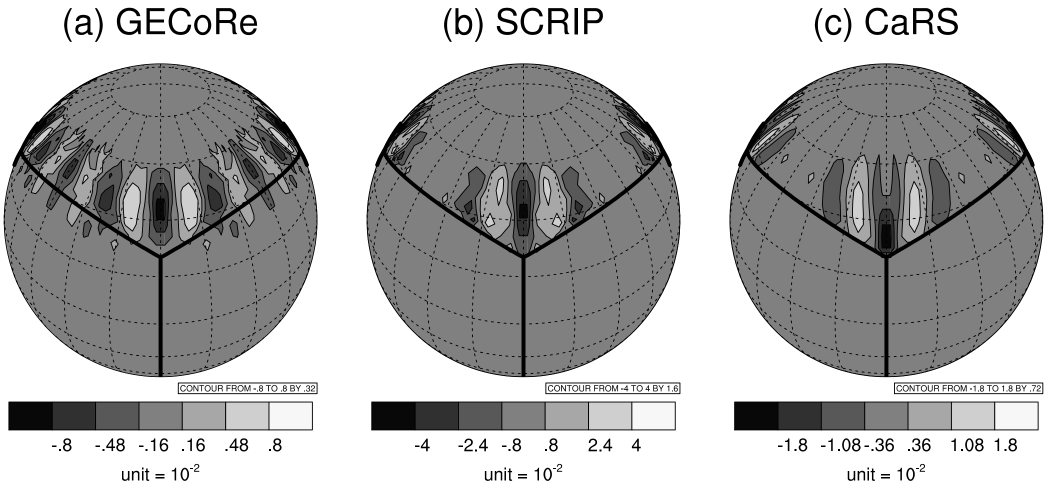


Figure 2.9: Spatial distribution of the error ($\times 10^{-2}$) for the remapping of Y_{32}^{16} from a coarse-resolution ABP grid ($N_c = 40$) to a RLL grid ($N_\lambda = 128$, $N_\theta = 64$) using (a) GECoRe, (b) SCRIP and (c) CaRS with piecewise parabolic (third-order) reconstructions. Both SCRIP and CaRS show clear correlation between the errors are the underlying cubed-sphere grid, whereas this "grid imprinting" is reduced under the GECoRe scheme.

2.4.5 Impact of the reconstruction method

We briefly turn our attention to derivative error in the GECoRe schemes by comparing four methods for computing the reconstruction coefficients in each cell. We focus on the non-equidistant parabolic fit (3-point) and quartic fit (5-point) methods, both computed in Gnomonic coordinates, to the stretched equidistant 3-point and 5-point methods computed in equiangular coordinates. Results from this comparison are given in Fig. 2.10.

Observe that there is no significant difference between both 3-point schemes and both 5-point schemes for the non-smooth Y_{32}^{16} and V_X test cases, since the natural derivative error dwarfs any error that would be present from stretching of the equiangular reconstruction. However, for the smooth test case, there are obvious deviations between the four methods in the third-order PPM scheme. In particular, we clearly observe an immediate benefit to computing the reconstruction coefficients directly in

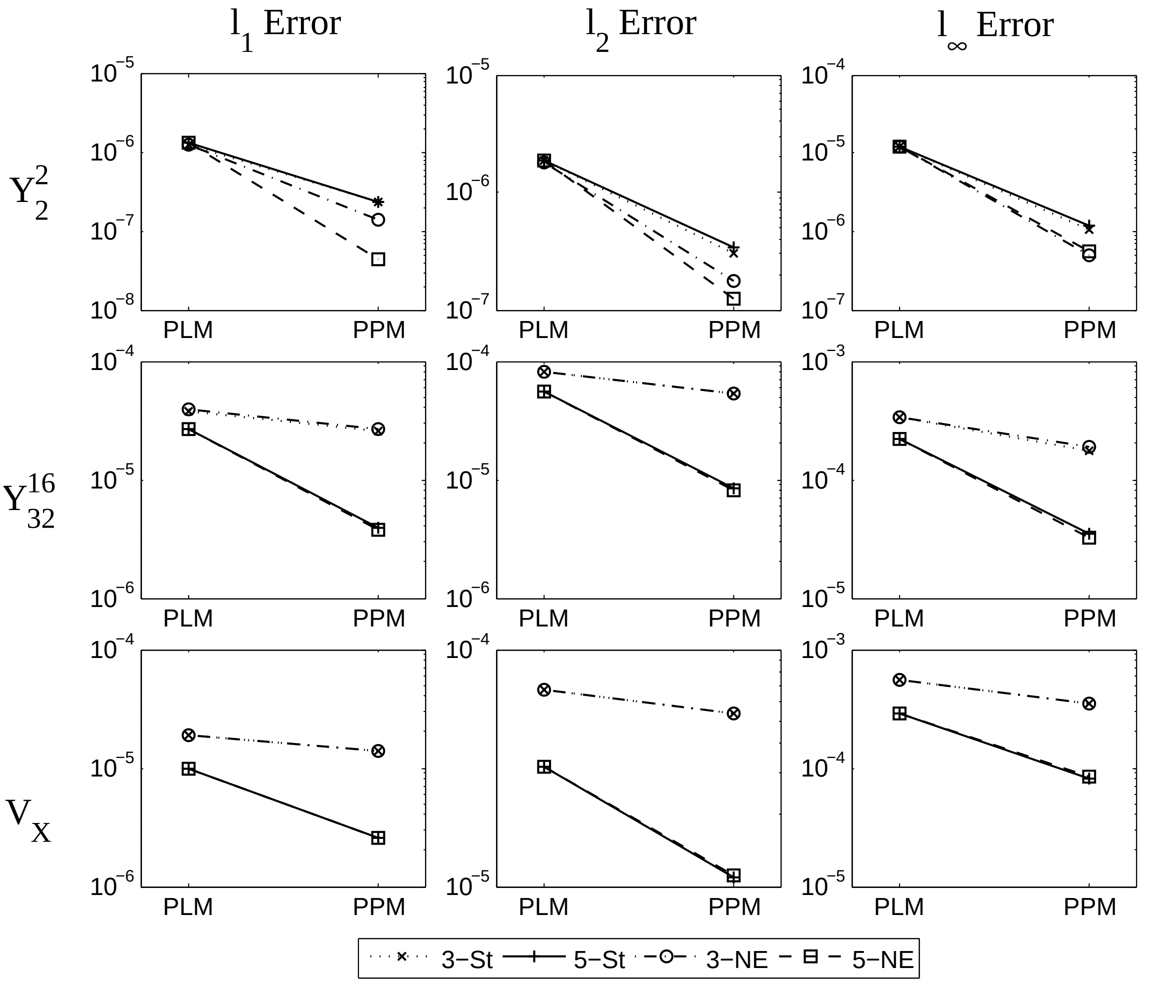


Figure 2.10: Performance measures for the remapping of Y_2^2, Y_{32}^{16} and the idealized vortices (V_X) from a high-resolution ABP grid ($N_c = 80$) to a RLL grid ($N_\lambda = 128$, $N_\theta = 64$) using GECoRe with four choices of sub-grid scale reconstruction techniques: 3-point stretched equiangular (3-St), 5-point stretched equiangular (5-St), 3-point non-equidistant Gnomonic (3-NE) and 5-point non-equidistant (5-NE).

Gnomonic coordinates. From the non-smooth test cases we also observe an obvious benefit to increasing the stencil size.

2.4.6 Impact of the monotone filter

In order to ensure monotonicity in the reconstruction, we employ the monotone filter of *Barth and Jespersen* (1989). This simple monotone filter simply scales the sub-grid scale reconstruction so that its minimum and maximum values do not exceed the cell-averages of the neighbouring cells. In the case of the second-order linear reconstruction, the extreme values within a cell will occur at the four corner points. For the third-order reconstruction, the extrema could also possibly occur along the boundary or within the cell. Hence, five additional points must be checked in the third-order scheme.

The effect of applying the monotone filter to the remapping scheme is given in Fig. 2.11. We observe that this simple monotone limiter tends to reduce the accuracy of the method by an order of magnitude, but, as a consequence, clearly maintains that global extreme points are not enhanced. Tests performed on a simple cosine hill field, which is more susceptible to overshoots and undershoots, actually result in improved accuracy of the remapped field under the monotone limiter (not shown).

As resolution is increased, we expect that the loss of accuracy due to the monotone limiter will be reduced, since higher resolution results in less relative variation in the scalar field. Further, advanced limiters, such as that of *Zerroukat et al.* (2005), if extended to two dimensions, are certain to result in an improved accuracy of these results.

2.5 Summary

A general modeling environment consists of different model components, usually implemented on different grid systems. Hence, accurate translation of scalar field

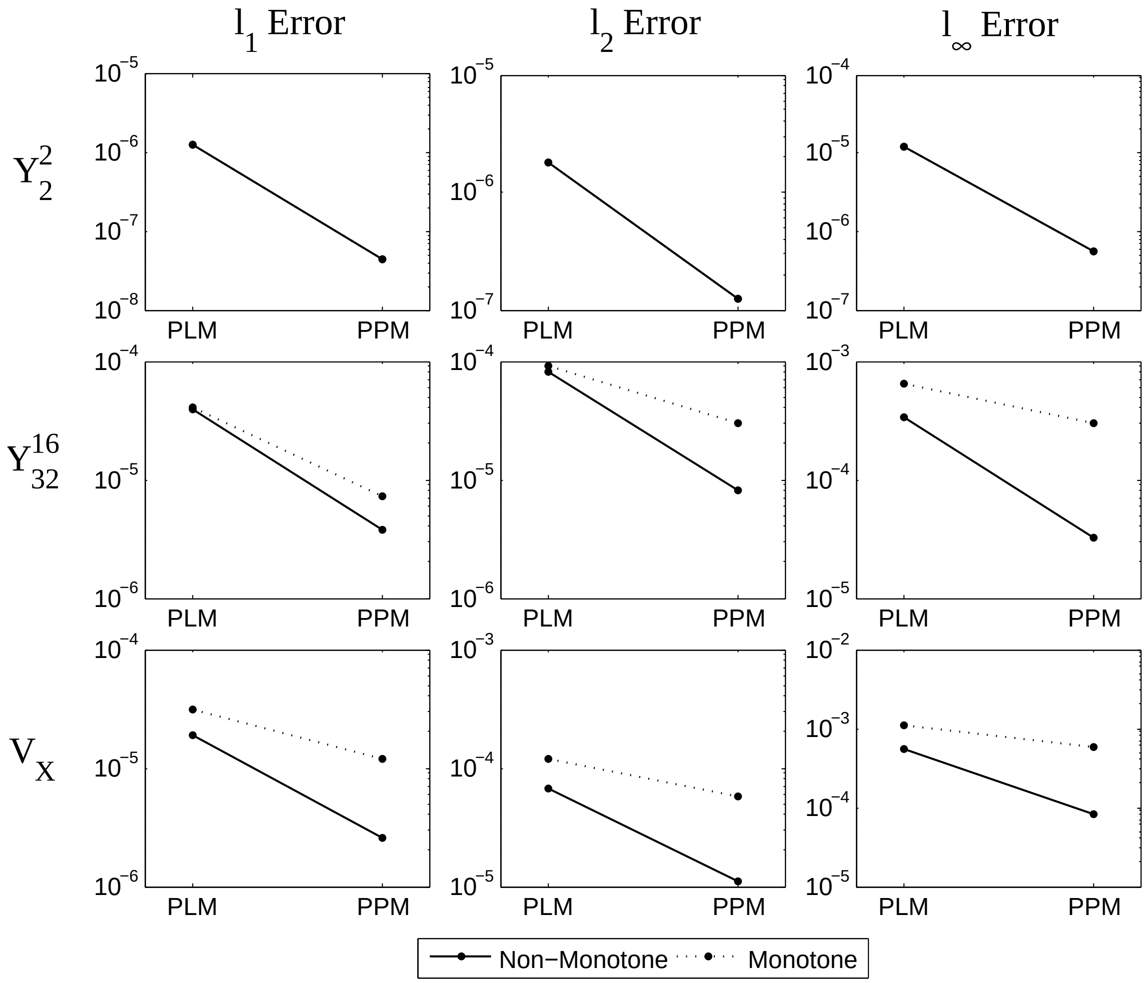


Figure 2.11: Performance measures for the remapping of Y_2^2, Y_{32}^{16} and the idealized vortices (V_X) from a high-resolution ABP grid ($N_c = 80$) to a RLL grid ($N_\lambda = 128$, $N_\theta = 64$) using GECoRe with (dotted line) and without (solid line) monotone filtering.

data between grid systems is important in order to ensure overall accuracy of the model. For variables such as fluxes conservation is a particularly important property that should also be maintained by the remapping scheme, as it ensures that a given model does not disobey the fundamental laws of nature. The accuracy of existing conservative schemes, such as SCRIP (Spherical Coordinate Remapping and Interpolation Package) and CaRS (Cascade Remapping between Spherical grids) is often limited by the capacity of these methods to accurately model the geometry of the problem. Here we have presented a high-order geometrically exact scheme for conservative and monotone remapping of scalar fields between the regular latitude-longitude and gnomonic cubed-sphere geometries. The new remapping scheme is referred to as GECoRe (Geometrically Exact Conservative Remapping).

GECoRe is based on the principle that we can integrate certain fields in exact closed-form along grid lines in the gnomonic cubed-sphere grid and regular latitude-longitude grid. The advantage of this approach is the removal of geometric error associated with low-order approximations to line segments at a similar computational cost to existing techniques. Here we provide mechanisms for constructing schemes up to third-order, possibly combined with an inexpensive monotone filter.

GECoRe is validated by remapping a standard set of both smooth and rapidly varying test functions. Standard error measures are compared with existing SCRIP and CaRS schemes at low and medium resolutions of the cubed-sphere grid. We observe that the GECoRe scheme excels in both cases when applied at second-order or higher, often yielding a one or two order of magnitude improvement over the existing schemes without additional tweaking. However, SCRIP is more general than GECoRe, since it can, in principle, handle any kind of spherical grids. Generalized versions of GECoRe can be obtained by replacing the exact integrals, where the exact formulas are too complicated or do not exist, with Gaussian quadrature of the potentials (*Lauritzen et al.* 2008).

This work can be found in *Ullrich et al.* (2009).

CHAPTER III

Wave Reflection

3.1 Introduction

The atmosphere and ocean are two facets of a vast nonlinear system that works on a broad range of interacting scales. Models are an invaluable tool for enhancing our understanding of this system, but the immensity of the problem – spanning the entirety of the Earth's surface – is at the edge of our present computational power. In order to meet growing demand for fine-scale simulations, the next generation of atmospheric models will likely need to rely on adaptive mesh refinement (AMR) in order to properly capture features of interest. A dynamically adaptive model, for instance, would be capable of hurricane tracking and modeling on a global scale, and would enhance our knowledge of mountain waves and extreme weather events. However, the benefits of adaptively refined grids do not come without a fair share of problems. Perhaps the most significant of these problems, in the context of geophysical modeling, is related to the mathematical handling of wave phenomena at coarse-fine grid boundaries. Except for a handful of numerical methods, most schemes allow wave groups to be spuriously reflected at grid resolution interfaces (see, for example, *Vichnevetsky* (1987)). This type of wave reflection is attributed to sudden changes in the numerical structure of the system, analogous to changes in the physical characteristics of the system.

As shown by *Vichnevetsky* (1987) and later by *Vichnevetsky and Turner* (1991), the significance of wave reflection increases substantially when the grid resolution is varied abruptly. As a consequence, many approaches (see, for example, *Côté et al.* (1998); *Fox-Rabinovitz et al.* (2001); *Staniforth and Mitchell* (1978)) instead rely on a smooth variation of the grid between coarse and fine regions. However, abruptly-varying grids generally perform better on parallel architectures when dynamic refinement is desired; in this case, dynamic grid refinement in a compact region can be performed without having to reconstruct the grid over a wide regional or global scale (which is necessary if smooth variation of the grid is required). Although these block-adaptive grids have been long used for astrophysical, aerospace and other computational fluid dynamics problems (see, for example, *Berger and Colella* (1989); *Berger and Oliger* (1984)), they were only first applied to geophysical limited area models by *Skamarock et al.* (1989) and *Skamarock and Klemp* (1993). More recently, these methods have been applied in spherical geometry by *Jablonowski et al.* (2006) and *Jablonowski et al.* (2009). Hence, our approach in this chapter is to consider only grids with abrupt variation in resolution.

To a close approximation, the atmosphere and ocean are in a state of geostrophic and hydrostatic balance. For geophysical flows, departures from geostrophy are approximately linear. The Mach number of these flows is generally much less than one, and shock waves are not present. It is here where linear numerical discretizations are the most relevant, since slope limiters are generally not required to ensure positivity of the thermodynamic variables. Long-term simulations of geostrophic flows also require perhaps the most stringent conservation constraints since a slow escape of air from the atmosphere on the order of the scheme's truncation error can lead to substantial atmospheric loss over time.

Although many previous papers have discussed the issue of wave reflection due to a discontinuity in grid resolution, these works have not focused on the issue of wave reflection in dissipative finite-volume methods. In this case, energy is not conserved over time, and so it is uncertain whether previous results, which have been derived in the case of zero energy dissipation, will still hold. Hence, it is our objective in this chapter to use mathematical analysis and numerical experimentation to determine which unstaggered dissipative finite-volume methods are best suited for geophysical modeling in the presence of a refined grid.

Before proceeding, we briefly discuss the wave reflection properties of other approaches. *Frank and Reich* (2004) demonstrated that the Box scheme is free of spurious reflections, but this scheme does not easily generalize to multiple dimensions and is implicit in a periodic domain. Further, purely upwind schemes, such as the first-order Godunov scheme, the discontinuous Galerkin (DG) scheme and spectral-volume methods, all of which do not use downstream information, are free from spurious wave reflection when applied to the advection equation. Nonetheless, these methods still suffer from nonlinear wave reflection when applied to the 1D shallow-water equations, for instance.

The foundation for our analysis will be the 1D advection equation, which describes the motion of a tracer field $q(x, t)$ in the presence of an underlying velocity field u. In its simplest form, this equation reads

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} = 0. \tag{3.1}$$

For simplicity, much of our analysis will be for the case that $u = \text{const} > 0$.

This paper is organized as follows. First, we present a framework for the set of finite-volume methods we will consider in section 3.2, and give some of their numerical properties. The results of a set of wave-reflection experiments are then given in section 3.3. In section 3.4 we introduce the shallow-water equations and linearized shallow-water equations, and show their connection to the advection equation. Lastly, our conclusions are presented in section 3.6. Note that throughout this chapter we will be making use of dimensionless length and time units.

3.2 Numerical discretizations

Numerous methods have been devised for the construction of finite-volume schemes, and all have their benefits and disadvantages. The method-of-lines approach is perhaps the most popular for constructing high-order finite-volume methods that are applicable to general systems of equations. Under this framework, a spatial sub-grid-scale reconstruction is combined with a numerical flux function to provide a discretization of the spatial component of the differential equation, which is then combined with a timestepping scheme that guarantees stability and accuracy when integrating forward in time. Timestepping schemes vary substantially in their properties, but must be chosen so that the eigenvalues of the spatial operator fit within the stability region of the timestepping scheme. We consider three types of finite-volume methods constructed under this framework:

- **Symmetric finite-volume.** If we assume continuity of our solution between elements we can directly reconstruct the value of the underlying field at edge-points. An interior reconstruction, which is necessary for evaluating source terms, is then obtained from the edge-values and value of the cell-averaged scalar field. Once the initial approximation is made, monotonicity constraints can be applied, which may cause the field to again become discontinuous at edges and hence require the solution of a Riemann problem. This approach does not rely more strongly on upwind-biased information, and so leads to a spatial discretization which is symmetric about the element being updated. The simplest symmetric finite-volume method is the so called central-in-space discretization, which has been thoroughly studied in the context of grid reflection (see, for example, *Trefethen* (1982); *Vichnevetsky* (1987); *Vichnevetsky and Turner* (1991)). It is obtained by assuming the value of the scalar field at each edge is simply the average of the values of neighbouring elements.

- **Upwind finite-volume.** The Monotone Upstream-centered Schemes for Conservation Laws (MUSCL) formalism of *van Leer* (1979) provides a mechanism for computing a sub-grid-scale reconstruction via a local reconstruction obtained from adjacent cell-averaged values. When evaluated at edge points, these reconstructions can then act as left and right states that are then used to solve a Riemann problem. Slope limiters can also be applied in the reconstruction step to enforce monotonicity and limit spurious oscillations. This approach leads to the upwind family of finite-volume schemes, so-named because they rely more heavily on information propagated in the direction of the fluid motion. Unstaggered high-order upwind finite volume schemes have been recently shown to be viable for shallow-water models on the sphere by *Ullrich et al.* (2010).

- **Semi-Lagrangian integrated-mass (SLIM) finite-volume methods.** A popular method that has been widely adopted for discretizations of the advection equation is the semi-Lagrangian (SL) approach. This method comes in two flavors – namely, forward SL and backward SL. In the forward SL approach, the velocity field is used to deform the grid, which is then remapped back to the original cell positions. In the backward approach, the velocity field is first evolved to time $n+1$. The evolved velocity field is then used to "devolve" the grid cells at time $n+1$ into a deformed grid that represents the locations of these cells at time n. Finally, the original grid information is remapped onto the deformed grid, giving new cell averages. These two approaches are identical for the 1D advection equation with $u = \text{const}$. Semi-Lagrangian methods are a physically motivated treatment of the advection equation, and have effectively no timestep limit (but accuracy degrades substantially for large timesteps and non-constant flow fields). Nonetheless, there is some difficulty in adapting this method to general hyperbolic systems. An analysis of SLIM schemes can be found in *Laprise and Plante* (1995). Examples of this approach include Fromm's scheme *Fromm* (1968), the advective form of the piecewise-parabolic method presented by *Colella and Woodward* (1984) and, in higher dimensions, the recently introduced CSLAM transport scheme of *Lauritzen et al.* (2010).

3.2.1 Diffusion, phase velocity and group velocity

An excellent tool for describing the properties of numerical discretizations is wave-mode analysis, which forms the backbone of our study of spurious wave reflection. In particular, this approach has been successfully applied by *Trefethen* (1982) and *Grotjahn and O'Brien* (1976) in the analysis of numerical methods for hyperbolic equations.

In general, any linear partial differential equation with constant coefficients supports wave-like solutions of the form

$$q(x,t) = \hat{q}\exp(i(kx - \omega t)), \tag{3.2}$$

where $q(x,t)$ denotes the state variable in physical space, $\hat{q}$ is the corresponding amplitude, k is the wave number and ω is the frequency. If we substitute this solution into (3.1) we obtain a *dispersion relation* of the form

$$\omega(k) = uk. \tag{3.3}$$

These modes propagate with speed

$$c_p(k) = \frac{\omega(k)}{k}, \tag{3.4}$$

which is known as the *phase speed.* The evolution of a wave packet however, is determined by the *group speed*, defined via

$$c_g(k) = \frac{\partial \omega}{\partial k}. \tag{3.5}$$

It is well-known (see, for example, *Brillouin* (1960)) that the group speed is the speed at which energy propagates in a system, as well as the speed associated with a traveling wave packet. In the case of the advection equation, these velocities are equivalent and given by $c_p = c_g = u$.

Linear discretizations of (3.1) similarly support wave modes of the form (3.2) but only allow us to roughly approximate the correct dispersion relation (3.3). When analyzing these numerical methods we will assume a uniform spatial grid, defined at discrete points via

$$x_j = j\Delta x, \quad \text{and} \quad t_n = n\Delta t, \tag{3.6}$$

where Δx and Δt are the element width and timestep, respectively, and j and n are spatial and temporal indices. Hence, wave-like solutions (3.2) take the form

$$q_j^n = \hat{q}\exp(i(kj\Delta x - \omega n\Delta t)), \tag{3.7}$$

where $k\Delta x$ is the normalized wave number, whose real component takes on values in the range $[0, \pi]$. On substituting this expression into a discrete numerical scheme, we obtain the *numerical dispersion relation*, which describes the relationship between ω and k and usually incorporates the dimensionless Courant-Friedrichs-Lewy (CFL) number

$$K = \frac{u\Delta t}{\Delta x}. \tag{3.8}$$

The numerical dispersion relation is a powerful tool for describing the properties of a numerical method:

- For two time-level schemes, such as forward Euler, backward Euler, Crank-Nicolson and all Runge-Kutta schemes, every value of k is associated with a single value ω. For three time-level schemes, such as the Leapfrog scheme, every value of k is associated with two values of ω.

- The imaginary component of $\omega(k)$ describes the growth rate of the mode k. Von Neumann stability *Charney et al.* (1950) is obtained by guaranteeing that $\mathrm{Im}(\omega) \leq 0$ for all real wave numbers. Stable numerical schemes which satisfy $\mathrm{Im}(\omega(k)) < 0$ for some k are known as *diffusive* (or *dissipative*). Note that the advection equation is naturally non-diffusive, so any non-zero diffusivity leads to *diffusive error*. The *amplification factor* after one timestep is then defined as

  $$A_{\Delta t} = \exp(\mathrm{Im}(\omega(k))\Delta t). \tag{3.9}$$

  However, $A_{\Delta t}$ is not desirable for describing the diffusivity of a scheme over a range of CFL numbers, since schemes with smaller timestep Δt must be applied multiple times to advance to the same time as schemes with larger Δt. Hence, one can alternatively describe diffusivity in terms of the *normalized amplification factor* A, defined for a fixed wavenumber k by

  $$A = \exp(\mathrm{Im}(\omega(k))T), \tag{3.10}$$

  where T is some fixed time. In the following analysis we choose $T = 1$ for simplicity.

- In general, a numerical dispersion relation will yield a frequency whose real component is nonlinearly dependent on k. In this case, the scheme will be *dispersive*, indicating that different wave numbers will travel at different phase speeds. The advection equation is non-dispersive, since ω and k are linearly

> related, however, numerical discretizations generally introduce *dispersive error* in the form of a nonlinear dispersion relation. The dispersive characteristics of a numerical method can be effectively described in terms of the phase velocity and group velocity, which are obtained from the numerical dispersion relation, when combined with (3.4) and (3.5). Since a numerical method can also be dissipative, we substitute $\mathrm{Re}(\omega)$ for ω in these relationships, which yields the numerical phase velocity and group velocity.

Note that group velocity analysis only applies directly for nondiffusive schemes, since diffusivity introduces a wavenumber-dependent attenuation of different wave modes. As a consequence, wave packets can lose their distinctive shape over time as certain wave modes are diffused from the simulation. However, the results of group velocity analysis will still hold approximately as long as the wave packet is composed of waves with similar diffusion rates.

3.2.2 Linear discretizations

In this chapter we focus our attention on high-order linear discretizations. Namely, we are interested in spatial semi-discretizations of the advection equation (3.1) that take the form

$$\frac{\partial q_j}{\partial t} = \sum_{m=-\ell}^{r} c_m q_{j+m}, \tag{3.11}$$

where the coefficients c_m are purely a function of the grid spacing Δx, timestep Δt and advection speed u (possibly via the CFL number). Here ℓ and r denote the number of leftward-elements and rightward-elements in the semi-discrete stencil. For simplicity, in this chapter we only analyze methods with $r \leq \ell = 2$, which leads to a stencil with at most five elements. Conservation is guaranteed by utilizing the finite-volume framework, which requires that all of the schemes can be written in the form

$$\frac{\partial q_j}{\partial t} = -\frac{F^*_{j+1/2}(\ldots, q_{j-1}, q_j, q_{j+1}, \ldots) - F^*_{j-1/2}(\ldots, q_{j-1}, q_j, q_{j+1}, \ldots)}{\Delta x}, \tag{3.12}$$

for a numerical flux function $F^*_{j+1/2}$. The numerical flux functions are defined at cell edges, which are denoted by half-indices.

When combined with an appropriate two-time-level explicit timestepping operator, the spatial semi-discretization (3.11) then leads to a complete discretization of the advection equation that we can write as

$$q_j^{n+1} = \sum_{m=-\ell\cdot s}^{r\cdot s} C_m q_{j+m}^n, \tag{3.13}$$

where the C_m are again purely functions of the grid spacing, timestep and advection speed. Here s denotes the number of stages used by the timestepping operator (for multistage schemes, such as the Runge-Kutta methods).

To determine the numerical dispersion relation for the scheme (3.13), we simply substitute (3.7) and solve for ω, obtaining

$$\omega = -\frac{1}{\Delta t} \arctan\left[\frac{\displaystyle\sum_{m=-\ell\cdot s}^{r\cdot s} C_m \sin(km\Delta x)}{\displaystyle\sum_{m=-\ell\cdot s}^{r\cdot s} C_m \cos(km\Delta x)}\right]. \tag{3.14}$$

Solutions to this equation are not unique, and the choice of an appropriate branch cut for the arctan function can lead to some confusion. Herein we will take the branch cut that gives $\omega(0) = 0$ and otherwise is a continuous function of k.

3.2.3 The $2\Delta x$ mode problem

Symmetric finite-volume (semi-) discretizations of the advection equation satisfy the property $c_m = -c_{-m}$, which usually arises from the application of a centered differencing operator within each element. They are characterized by low diffusivity and consistent behaviour regardless of CFL number, but suffer from the $2\Delta x$ mode problem; namely, under such a semi-discretization both the constant field $q_j =$ const. and the $2\Delta x$ mode $q_j = (-1)^j$ satisfy

$$\frac{\partial q_j}{\partial t} = 0, \tag{3.15}$$

implying that both modes are invariant in time. Thus, under a symmetric finite-volume method, the constant mode and $2\Delta x$ mode are indistinguishable, regardless of the size of the stencil and choice of temporal discretization. As a consequence, this mode will feature both zero diffusion and zero phase velocity when simulating the advection equation, and so any nonlinear effects or source terms which contribute energy to this scale will not be dissipated. The result is "checkerboarding" of the state variables under this operator (see, for example, *Randall* (1994)).

Upwind finite-volume (semi-) discretizations are always upwind biased, satisfying $r < \ell$. This criterion implies that more information is drawn from the upwind direction as the flow evolves. In general, these methods do not suffer from the $2\Delta x$ mode problem, but may possess high-frequency wave modes which are weakly damped, usually at specific CFL numbers.

3.2.4 The gas dynamics form of the piecewise-parabolic method (PPM)

The gas dynamics form of the piecewise-parabolic method (PPM) of *Colella and Woodward* (1984) (Chapter 3) is a high-order symmetric discretization that, for a locally smooth field q, estimates the field at edge points via the central reconstruction

$$q_{j+1/2} = \frac{-q_{j-1} + 7q_j + 7q_{j+1} - q_{j+2}}{12} + O(\Delta x^4). \tag{3.16}$$

Because the field is assumed continuous at edge points (with no limiter applied), the numerical flux can be computed directly without applying a Riemann solver,

$$F^*_{j+1/2} = u \cdot q_{j+1/2}. \tag{3.17}$$

This choice leads to the semi-discretization

$$\frac{\partial q_j}{\partial t} = -\frac{K}{\Delta t} \left(\frac{1}{12} q_{j-2} - \frac{8}{12} q_{j-1} + \frac{8}{12} q_{j+1} - \frac{1}{12} q_{j+2} \right), \tag{3.18}$$

where the RHS of (3.18) is exactly the fourth-order symmetric approximation to $\partial q / \partial x$ centered at element j. The eigenvalues of this spatial operator are purely imaginary, and so must be paired with at least a three-stage third-order Runge-Kutta (RK3) timestep. Doing so, we obtain a scheme that is stable up to $K \leq 1.26$.

We plot the normalized amplification factor, group velocity and phase velocity associated with this scheme in Fig. 3.1. Observe, in particular, that this scheme supports waves with negative group velocity, leading to a maximum negative group velocity of $-5/3$ at $k\Delta x = \pi$. The range of dimensionless wavenumbers $k\Delta x \in [\pi/2, \pi]$ corresponds to the waves with wavelength between $4\Delta x$ and $2\Delta x$. Most waves in this range travel with negative group speeds and so are not truthfully represented by the numerical scheme. Note that the phase velocity also drops to zero at $k\Delta x = \pi$, whereas at this wavenumber the amplification factor is exactly 1 regardless of CFL number. As mentioned previously, this implies that the $2\Delta x$ mode is an undamped "standing wave" that is retained by the numerical method.

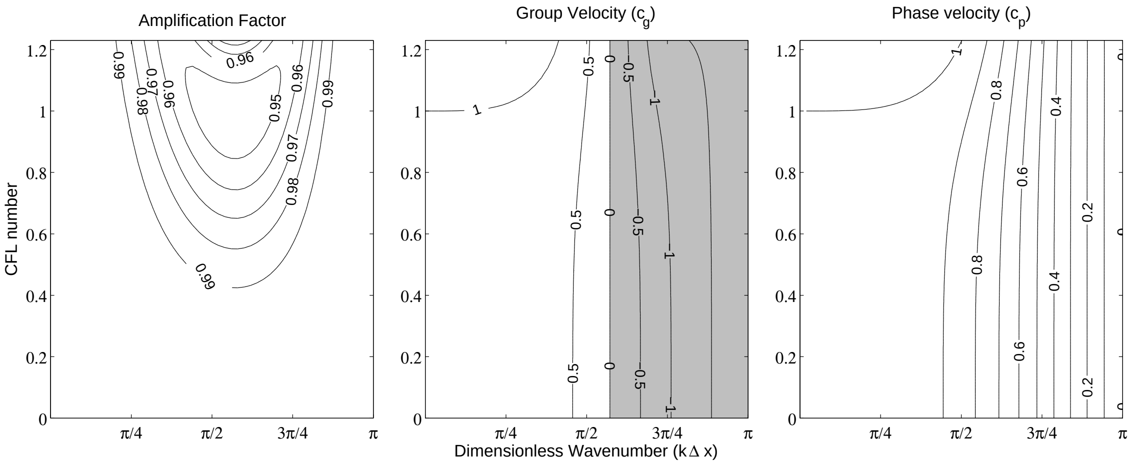


Figure 3.1: Contour plots showing diffusive and dispersive characteristics associated with PPM with RK3 timestep. Here Δt is varied so as to span to CFL range $[0, 1.26]$ with constant wave speed $u = 1$ and fixed Δx. Gray regions in the group velocity plot indicate regions of negative group velocity.

3.2.5 A second-order upwind (FV2) scheme

A linear sub-grid-scale reconstruction was adopted by *van Leer* (1974, 1979) for shock-hydrodynamics problems, where the derivative was obtained via a non-linear limiting procedure. Without a strict monotonicity constraint, we can instead forego the limiting procedure and hence obtain a linear second-order-accurate upwind scheme. A sub-grid-scale reconstruction of the form

$$\tilde{q}_j(x) = q_j + (x - x_j)Dq_j, \tag{3.19}$$

is computed in each cell, where Dq_j denotes the numerical approximation to the first derivative, obtained from the central-difference formula

$$Dq_j = \frac{q_{j+1} - q_{j-1}}{2\Delta x}. \tag{3.20}$$

Since this reconstruction is discontinuous at cell edges we must use a Riemann flux operator, taken simply to be the upwind flux

$$F^*_{j+1/2} = u\tilde{q}_j \left(x_j + \frac{\Delta x}{2} \right). \tag{3.21}$$

After simplifying, the evolution equation reads

$$\frac{\partial q_j}{\partial t} = -\frac{K}{\Delta t} \left(\frac{1}{4} q_{j-2} - \frac{5}{4} q_{j-1} + \frac{3}{4} q_j + \frac{1}{4} q_{j+1} \right), \tag{3.22}$$

This scheme is unstable under a forward Euler timestep, but stable for Runge-Kutta operators of at least second order. Making use of the two-stage second-order Runge-Kutta scheme (RK2), we obtain a discretization that is stable for $K \leq 1$.

We plot the normalized amplification factor, group velocity and phase velocity associated with this scheme in Fig. 3.2. In particular, observe that for this scheme wave modes which are propagated with negative group velocities are also modes which experience strong diffusion. This result was previously obtained by *Karni* (1994), who observed that even in the limit of vanishing CFL number wave groups were not permitted to propagate backwards in upwind schemes. A singularity can also be observed in the group velocity and phase velocity plots, corresponding to the point where the amplification factor is identically zero. The solid line that extends to the right of the singularity in the phase velocity plot is a discontinuity in the branch cut when evaluating the frequency ω from (3.14). Comparing this scheme against PPM in Fig. 3.1, we observe that the FV2 scheme possesses a much more complicated structure and introduces stronger diffusion at high wavenumbers. As with PPM, the $2\Delta x$ mode is a standing mode with non-zero group velocity; however, unlike in PPM, this mode is strongly damped at all but the largest of the stable CFL numbers. In the nonlinear case this analysis suggests bounding the CFL number by 0.9 so as to prevent artificial enhancement of this mode against the background field.

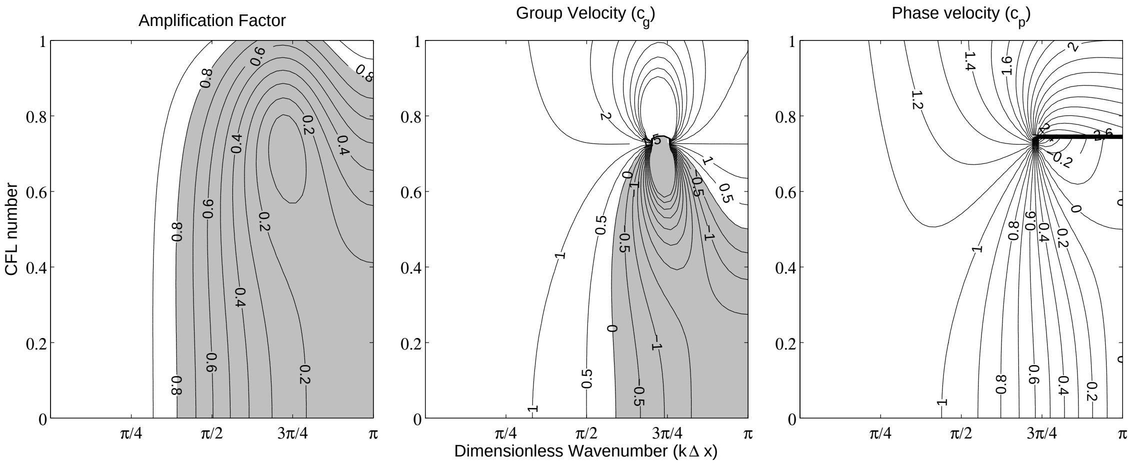


Figure 3.2: Contour plots showing diffusive and dispersive characteristics associated with the FV2 scheme with RK2 timestep. Here Δt is varied so as to span to CFL range $[0, 1]$ with constant wave speed $u = 1$. Gray regions indicate regions of significant damping on the plot of the amplification factor ($A \leq 0.8$) and negative (backwards propagating) group velocities on the plot of the group velocity.

3.2.6 A third-order upwind (FV3p3) scheme

Extending on the ideas of *van Leer* (1974, 1977), we make use of a sub-grid-scale reconstruction of the form

$$\tilde{q}_j(x) = q_j + (x - x_j)Dq_j + \left((x - x_j)^2 - \frac{\Delta x^2}{12}\right)\left(\frac{1}{2}D^2 q_j\right), \tag{3.23}$$

where x_j is the centerpoint of element j and Dq_j and D^2q_j denote numerical approximations to the first and second derivatives of the field q in element j, obtained from

$$Dq_j = \frac{q_{j+1} - q_{j-1}}{2\Delta x}, \quad \text{and} \quad D^2 q_j = \frac{q_{j+1} - 2q_j + q_{j-1}}{\Delta x^2}. \tag{3.24}$$

As with the FV2 scheme, we compute edge fluxes via the upwind flux (3.21), leading to the evolution equation

$$\frac{\partial q_j}{\partial t} = -\frac{K}{\Delta t}\left(\frac{1}{6}q_{j-2} - q_{j-1} + \frac{1}{2}q_j + \frac{1}{3}q_{j+1}\right), \tag{3.25}$$

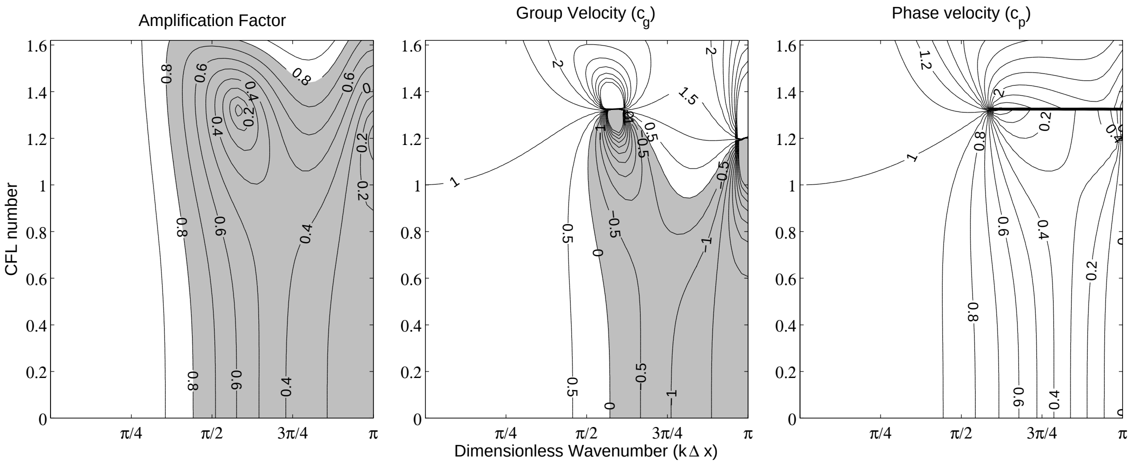


Figure 3.3: As Fig. 3.2 except with the FV3p3 scheme with RK3 timestep. Here we span the CFL number over the range $[0, 1.63]$.

This scheme is unstable under both the forward Euler method and RK2 timestep since the eigenvalues of the spatial operator closely shadow the imaginary axis near the origin. Hence, we will combine this spatial stencil with the third-order Runge-Kutta (RK3) scheme, leading to a scheme that is stable up to $K \leq 1.63$.

We plot the normalized amplification factor, group velocity and phase velocity associated with this scheme in Fig. 3.3. As with the FV2 scheme (see Fig. 3.2), the behaviour is complicated for wavenumbers in the range $[\pi/2, \pi]$, featuring two singularities due to the presence of a zero amplification factor. Again we observe a similar branch cut discontinuity in the phase velocity plot. Interestingly, the group velocity is positive for virtually all wavenumbers at $K \geq 1.32$. Also, the $2\Delta x$ mode always experiences significant diffusion under this scheme, whereas the $3\Delta x$ mode ($k\Delta x = 2\pi/3$) is relatively undamped at high CFL numbers. Diffusion in this method is comparable to the FV2 scheme, with slightly stronger diffusion at CFL numbers greater than about 0.6.

3.2.7 A third-order semi-Lagrangian integrated-mass (SLIM3p3) scheme

The third-order semi-Lagrangian scheme we will analyze is based on the discontinuous piecewise-parabolic reconstruction of *Laprise and Plante* (1995). In this case we again make use of a reconstruction of the form (3.23) and (3.24), except now we apply the SLIM methodology to compute fluxes by integrating upstream from each cell edge. For CFL numbers in the range $0 \leq K \leq 1$ the numerical flux function then takes the form

$$\begin{aligned} F^*_{j+1/2} &= \frac{1}{\Delta t} \int_{x_j+\Delta x/2-u\Delta t}^{x_j+\Delta x/2} \tilde{q}(x')dx' \qquad (3.26) \\ &= \frac{u}{6}\left[(K^2-1)q_{j-1} - (K+1)(2K-5)q_j + (K-1)(K-2)q_{j+1}\right]. \end{aligned}$$

The integrated transport scheme then takes the form

$$\begin{aligned} \frac{\partial q_j}{\partial t} &= -\frac{K}{\Delta t}\left[-\frac{1}{6}(K^2-1)q_{j-2} + \frac{1}{2}(K+1)(K-2)q_{j-1} \right. \qquad (3.27) \\ &\qquad \left. -\frac{1}{2}(K^2-2K-1)q_j + \frac{1}{6}(K-1)(K-2)q_{j+1}\right]. \end{aligned}$$

As stated earlier, SLIM methods can be extended to have an arbitrarily large CFL number if the integration is applied only to elements that overlap the Lagrangian control volume (see, for example, *Lauritzen et al.* (2010)).

Unlike (3.18), (3.22) and (3.25), the SLIM framework leads to an evolution equation that is nonlinear in K. This scheme is stable if combined with a forward Euler timestep and further is exact for $K = 1$. Unlike the previous schemes, we do not recommended combining this scheme with a Runge-Kutta timestep since the resulting scheme is highly diffusive.

We plot the normalized amplification factor, group velocity and phase velocity associated with this scheme in Fig. 3.4. The remapping step in the SLIM framework

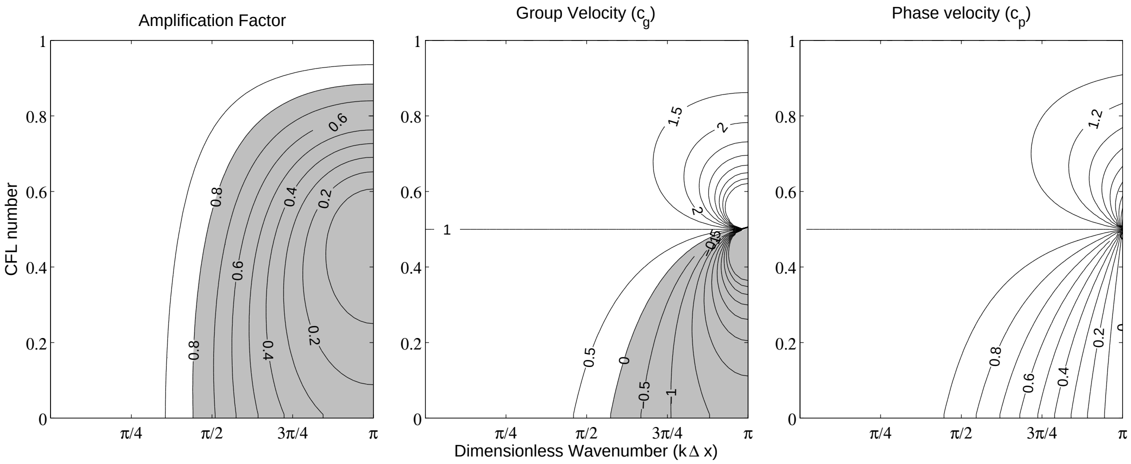


Figure 3.4: As Fig. 3.2 except with the SLIM3p3 scheme with forward Euler timestep. Here we span the CFL number over the range $[0, 1.0]$.

is responsible for the diffusivity of the scheme, and leads to an amplification factor of zero at $k = \pi$ and $K = \frac{1}{2}$ (with a corresponding singularity in the group velocity). As with the upwind finite-volume schemes, waves in the region of negative group velocity also experience strong diffusion. Notably, diffusion in this scheme is weaker than for the corresponding upwind schemes.

3.3 Wave reflection

In this section we tackle the problem of spurious wave reflection due to a grid resolution discontinuity. In section 3.3.1, we introduce our test environment for spurious wave reflection. We analyze the decay rate of spurious modes in section 3.3.2 and the initial amplitude of a reflected wave in section 3.3.3. We present several reflected wave tests using a symmetric FV scheme in section 3.3.4 with and without a slope limiter. A similar analysis is performed for upwind finite-volume schemes in section 3.3.5 and for SLIM finite-volume schemes in section 3.3.6.

Spurious wave reflection in the advection equation can be attributed to an artificial transfer of energy from forward-propagating physical modes into spurious backward-propagating parasitic modes. When a forward-propagating wave packet hits a resolution discontinuity, frequency must be conserved across the interface, but errors are accumulated at all wave numbers with the supported frequency. Certain modes are then propagated backwards relative to the flow. This type of wave reflection can occur even when a scheme does not possess wave modes with negative group velocity, and depends largely on the amount of downstream information that is used in the evolution equations; it is a linear effect, and so tends to be the dominant source of error even among nonlinear differential equations.

3.3.1 Wavemaker driven grid reflection

Wave-like solutions are not, in general, eigenfunctions of the discrete update equations on a grid with a resolution discontinuity. However, if we neglect boundary conditions, they are eigenfunctions of the update equations on each uniformly spaced grid.

In order to analyze wave reflection at grid resolution discontinuities, we follow the approach proposed by *Trefethen* (1982). Under this simplified model, the advection equation is simulated over a domain $x = [0, 1]$ with a grid resolution discontinuity introduced at $x = 1/2$. In the regions $x = [0, 1/2]$ and $x = [1/2, 1]$ we make use of discrete grid spacing Δx_f (on the fine grid) and Δx_c (on the coarse grid), respectively, with $\Delta x_f < \Delta x_c$. The resolution ratio $R \geq 1$ at the discontinuity is then defined as

$$R = \frac{\Delta x_c}{\Delta x_f}. \tag{3.28}$$

At time $t = 0$, we begin forcing the left boundary with real frequency ω and amplitude A. As a consequence, we observe a wave of the form

$$q(x, t) = A \exp(i(kx - \omega t)), \tag{3.29}$$

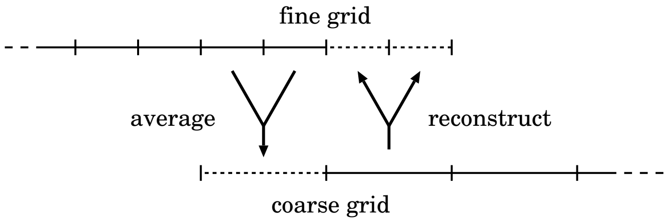


Figure 3.5: We maintain the illusion of resolution regularity by averaging from the fine grid to coarse grid ghost elements. To obtain the cell-averaged values on the fine grid, we first construct a sub-grid-scale reconstruction on the coarse grid and then integrate it to obtain the cell-averages on the fine grid. The dotted (overlapping) regions contain the ghost cells.

with complex wavenumber $k(\omega)$ satisfying $\mathrm{Im}(k) \geq 0$ (under sufficient stability conditions).

When running simulations, the resolution discontinuity is treated much like any other boundary on a uniform resolution domain (see Fig. 3.5). On both the fine and coarse grid the boundary conditions at this point must be obtained from the overlapping grid panel. As a consequence, we are able to maintain the illusion of grid resolution uniformity during the simulation, as long as the boundary conditions are correctly applied.

In order to obtain element-averages for the coarse grid ghost elements that are consistent with the fine grid, we simply average from the fine grid elements. To obtain element-averages on the fine grid from the coarse grid, we first build a sub-grid-scale reconstruction of the form (3.23) using the element-averaged values from the fine grid and known coarse grid element-averages. Piecewise-linear and piecewise-constant reconstructions can alternatively be obtained by setting one or both of Dq_j or D^2q_j to zero. Then for each fine-grid ghost cell we average over the corresponding reconstruction. This process easily generalizes to higher dimensions, and does not require any additional special treatment of elements near the resolution discontinuity.

3.3.2 Decay of parasitic modes

The decay rate of the reflected wave modes can be determined directly from the discretization. Once the initial perturbations from kick-starting the system have died down the frequency ω becomes invariant. Hence, both the "true wave" and "parasitic wave" must oscillate at frequency ω. To determine all complex wavenumbers k with natural frequency ω, we assume wavelike solutions of the form (3.29). If we define $\beta = \exp(ik\Delta x)$, unstaggered FV schemes of the form (3.13) can be reduced to a polynomial of the form

$$(\exp(-i\omega\Delta t) - 1)\beta^{\ell} = \sum_{m=-\ell\cdot s}^{r\cdot s} C_m \beta^{m+\ell}. \tag{3.30}$$

Hence, this polynomial will have $r \cdot s + \ell \cdot s$ roots that represent all wavenumbers that oscillate at frequency ω. In particular, if we assume sufficient stability conditions, roots with $\text{Im}(k) \geq 0$ will be decaying modes that are propagated forward by our scheme, whereas roots that satisfy $\text{Im}(k) < 0$ will be growing modes that are propagated backwards (these are the "parasitic modes"). In fact, the smaller in magnitude (or closer to zero) we observe for $\text{Im}(k)$, the longer the resulting "tail" is from the point of generation. Hence, we define the *dominant parasitic mode* for an FV scheme to be the mode $k(\omega)$ that satisfies (3.30) with $\text{Im}(k) < 0$ such that for any other parasitic mode k' we have $\text{Im}(k) > \text{Im}(k')$. The *spatial decay rate* of the dominant parasitic mode is then defined as $-\text{Im}(k\Delta x)$. A large positive decay rate leads to a sharp drop-off of the parasitic mode, whereas a small positive decay rate leads to an elongated tail. A decay rate of zero corresponds to a parasitic mode which does not decay away from the resolution discontinuity.

We plot the dominant parasitic mode for various choices of frequency ω in Fig. 3.6. In general, the complexity of the polynomial (3.30) prevents us from obtaining any general results in all but the simplest of cases, but we can nonetheless make some observations based on these four schemes:

- In all cases the decay rate appears to be smallest at $\omega = 0$, which is associated with a constant forcing. The decay rate then increases monotonically as ω increases.

- At $\omega = 0$ the decay rate can be derived analytically if we observe that, for the schemes we have analyzed, the dominant parasitic mode is the same if we use the coefficients c_m (see (3.11)) in place of C_m in (3.30). In fact, for the two upwind schemes and the SLIM scheme with a forward Euler timestep, the degree of the polynomial (3.30) is only three. If we further observe that for any consistent scheme $\beta = 1$ (the constant mode) must be a root of (3.30), then the remaining roots are

$$\beta = \frac{-(c_0 + c_1) \pm \sqrt{(c_0 + c_1)^2 - 4c_1(c_{-1} + c_0 + c_1)}}{2c_1}. \tag{3.31}$$

  Using this formula, we can calculate that the $\omega = 0$ decay rate of the dominant parasitic mode is $1.44 \approx \log(-2 + \sqrt{5})$ for the FV2 scheme, $0.99 \approx \log((-5 + \sqrt{33})/2)$ for the FV3p3 scheme and

$$\beta = \frac{(2K^2 - 3K - 5) - \sqrt{-15K^2 + 18K + 33}}{2(K^2 - 3K + 2)}, \tag{3.32}$$

  for the SLIM3p3 scheme. In the limit as $K \to 1$, the SLIM3p3 scheme does not have any parasitic modes, and so the decay rate tends to infinity. In the limit as $K \to 0$, the SLIM3p3 scheme converges to a decay rate of $\log((5 + \sqrt{33})/4)$. Observe that since the coefficients c_m are linear in K for the symmetric and upwind FV schemes, all dependence on K divides out of (3.31).

- For the schemes that use pointwise edge values to calculate edge fluxes (namely, PPM, FV2 and FV3p3), the decay rate decreases monotonically with increasing

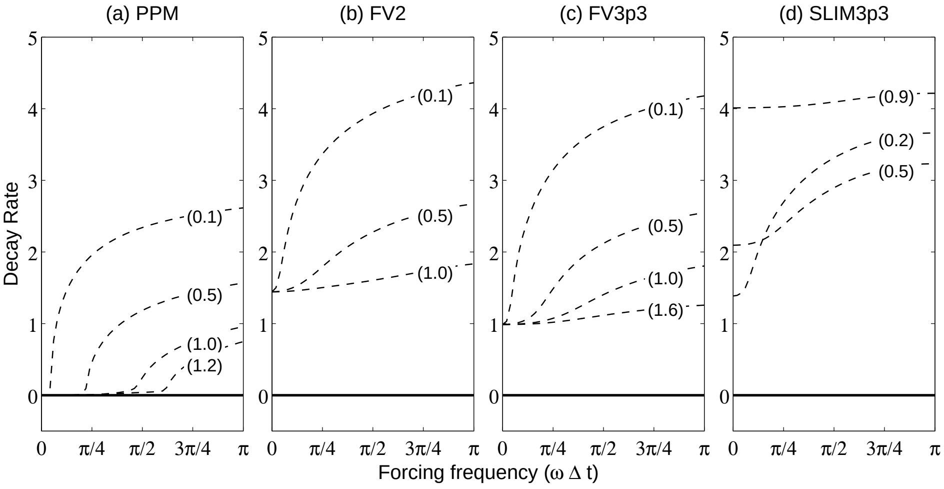


Figure 3.6: The decay rate ($-\text{Im}(k\Delta x)$) of the dominant parasitic modes for (a) PPM, (b) FV2, (c) FV3p3 and (d) SLIM3p3 under sinusoidal forcing of frequency ω and for several choices of CFL number (indicated in parenthesis on each curve).

CFL number at all frequencies $\omega > 0$. At $\omega = 0$, we observe that the decay rate is independent of CFL number.

The number of elements affected by the parasitic mode before it is damped below a fixed threshold is independent of the grid spacing. For a decay rate of 1.44, as with the FV2 scheme, a perturbation at the grid resolution interface of magnitude 1 will require approximately 8 elements to be damped to 10^{-5}. For a decay rate of 0.99, as with the FV3p3 scheme, the same perturbation will require 12 elements to decay to 10^{-5}.

3.3.3 Amplitude of the parasitic mode at the discontinuity

In addition to knowing the decay rate of a given parasitic mode, it is important to also understand its initial amplitude at a grid resolution discontinuity. To study the amplitude of the parasitic mode, we carried out a sequence of simulations using the PPM scheme at forcing frequencies that were sufficiently low to prevent the parasitic mode from decaying significantly (see Fig. 3.6). The amplitude of the parasitic mode was calculated empirically by differencing the unrefined and refined grid simulations near the grid resolution discontinuity. This result was then normalized by the amplitude of the incident wave at the discontinuity (since, especially at high wave-numbers, substantial decay of the incident mode was observed).

The empirically calculated ratio of the amplitude of the parasitic mode to the amplitude of the incident wave is depicted in Fig. 3.7 for various simulations with resolution ratios $R = 2, 4$ and 8. For waves that are well-resolved on both grids, the initial amplitude of the parasitic mode at the discontinuity is largely due to the discrepancy between the “true” solution on the fine grid and the solution in the overlapping grid elements obtained from remapping the coarse grid solution. Improving the order of accuracy in the remapping stage (using a piecewise linear or piecewise parabolic reconstruction, for instance, instead of a piecewise constant reconstruction) will reduce the initial amplitude of the perturbation. For wavenumbers which are poorly resolved or unresolved on the coarse grid, the reconstruction will not carry any information about the “true” wave and so the discrepancy between the incident wave and the representation on the coarse grid will be essentially maximal. In this case, improving the formal accuracy of the remapping procedure will not improve the outcome, since the element averages of the reconstruction on the coarse grid still do not contain any information about these waves.

As observed previously in the literature (see, for example, *Vichnevetsky* (1987)), wave reflection can be dramatically reduced by smoothly adjusting the grid spacing, rather than through abrupt changes in resolution. Intuitively, this result follows since the sub-grid-scale reconstruction on adjacent grid cells closely matches up under a smoothly varying element width. This result is also consistent with the observations of *Trefethen* (1982), who noted that forcing at the outflow boundary produced parasitic modes of the same amplitude as the prescribed forcing.

For comparison, we have devised a simple model for predicting the parasitic amplitude. Our goal is to demonstrate that the discrepancy between the solution on the uniform-resolution grid and the solution on the refined grid is the dominant influence in determining the amplitude of the parasitic mode for well-resolved incident waves. Given incident wave solution of the form (3.29), we can define a moving average operator via

$$\overline{q}(x; R, \Delta x_f) = \frac{1}{R\Delta x_f} \int_{x}^{x+R\Delta x_f} q(x)dx. \tag{3.33}$$

If a wave is well-resolved on the fine grid, it will propagate without modification, and so the element average on the fine grid $\overline{q}_f$ will be given by

$$\overline{q}_f = \overline{q}(x; 1, \Delta x_f) = \frac{\exp(i(kx - \omega t))}{k\Delta x_f}(\exp(ik\Delta x_f) - 1). \tag{3.34}$$

If the wave is well-resolved on the coarse grid as well, the corresponding element average of the first element on the coarse grid $\overline{q}_c$ will be

$$\overline{q}_c = \overline{q}(x; R, \Delta x_f) = \frac{\exp(i(kx - \omega t))}{Rk\Delta x_f}(\exp(iRk\Delta x_f) - 1). \tag{3.35}$$

For a piecewise-constant reconstruction at the discontinuity, the discrepancy between the true and approximate solution (copied directly from the coarse grid) is then given by $D = \overline{q}_f - \overline{q}_c$. This approach can be easily extended to higher-order reconstructions (such as we have done with the piecewise-parabolic reconstruction in Fig. 3.7), but for sake of brevity we have not included the corresponding formula here.

Waves that are not well-resolved on both grids will deviate from the ideal propagation model described above. As frequency is increased, we observe that these wave modes will lead to greater reflection than the ideal model at almost the same frequency for both reconstruction schemes. In fact, if we compare this point of departure with the group velocity plots given in Fig. 3.1, we observe it is approximately correlated with regions where the group velocity tends away from $u = 1$. Thus, for waves in

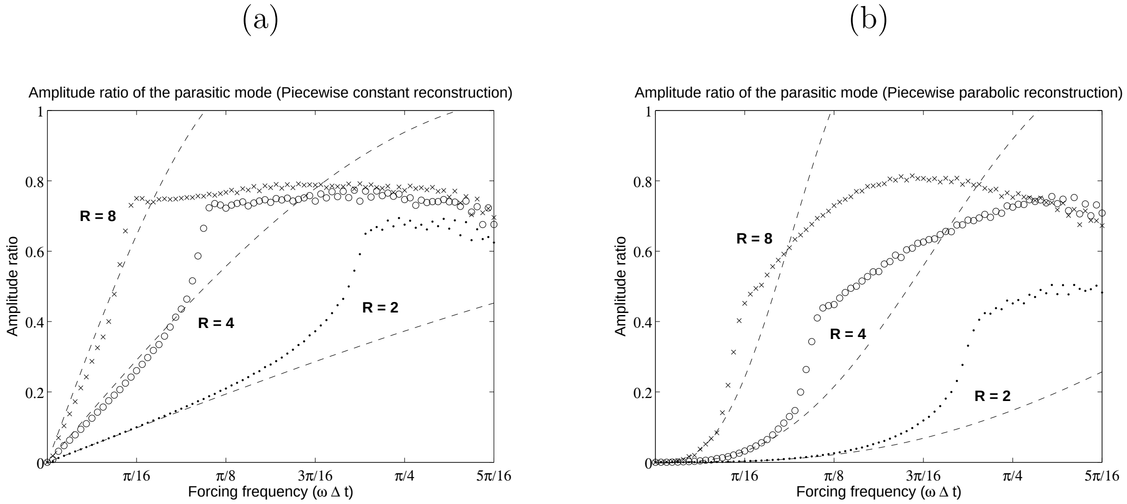


Figure 3.7: The amplitude ratio of the parasitic mode for various choices of resolution ratio R and for (a) piecewise constant and (b) piecewise parabolic reconstructions at the grid resolution discontinuity computed from repeated simulations using the PPM scheme. The dashed line indicates the predicted parasitic amplitude from the ideal wave propagation model, whereas symbols indicate the results of numerical simulation. Observe that the higher-order reconstruction greatly reduces the amplitude ratio at small forcing frequency, but does not substantially affect the reflection amplitude at larger forcing frequencies.

this moderate-frequency regime, it seems that the numerical method is not able to effectively propagate the energy of the wave away from the discontinuity, implying an increase in the amplitude of the parasitic mode. At high-frequencies the normalized amplitude of the parasitic mode flattens (especially dramatic for the piecewise constant reconstruction), suggesting some maximal efficiency of the reflection process has been achieved. Beyond a forcing frequency of $\omega\Delta t = 5\pi/16$ our empirical analysis method is no longer valid due to the rapid decay rate of the incident mode and apparent decay of the parasitic mode away from the resolution discontinuity.

3.3.4 Wave reflection by symmetric FV schemes

As observed in Fig. 3.6, PPM does not significantly damp reflected oscillations at any frequency. This result is apparent in any of the wave-driver simulations using undamped PPM (see Fig. 3.8). Here we clearly observe a very strong, undamped high-frequency wave that travels away from the grid resolution interface. The reflected wave has a normalized wavenumber $k\Delta x \approx \pi$ and so travels at a group velocity of $\approx 5/3$ (see Fig. 3.1). As a consequence, the parasitic mode has almost reached the left boundary by the time the rightgoing wave has only traversed half of the coarse domain.

In agreement with the results in section 3.3.3 we see that increasing the order of accuracy of the remapping process at the resolution discontinuity does not significantly affect the qualitative properties of the parasitic wave, and only has an effect on the amplitude of the parasitic wave at smaller wavenumbers (see Fig. 3.9).

One might wonder if combining this symmetric scheme with a slope/curvature limiter would be sufficient to remove spurious parasitic waves. If we apply the limiting procedure described in *Colella and Woodward* (1984) we can no longer guarantee continuity at cell edges, and so must utilize a Riemann flux where discontinuities occur. In this case we adopt an upwind flux operator analogous to (3.21). Simulations were again carried out using a unlimited piecewise parabolic reconstruction at the resolution discontinuity and are plotted in Fig. 3.10. The oscillations have been suppressed substantially in this case, but have not been removed completely. In fact, in the small wavenumber case ($\omega = 20.0$) we clearly see that the parasitic mode is persistent at about 10^{-3} of the magnitude of the initial wave – we observe the spurious mode is able to “hide” in the low-frequency incident wave, creating a subtle staircasing effect. Here the incident wave also plays the role of a carrier wave for the parasitic mode, since the parasitic mode would be removed almost immediately by the limiting procedure if no incident wave was present. On the other hand, the

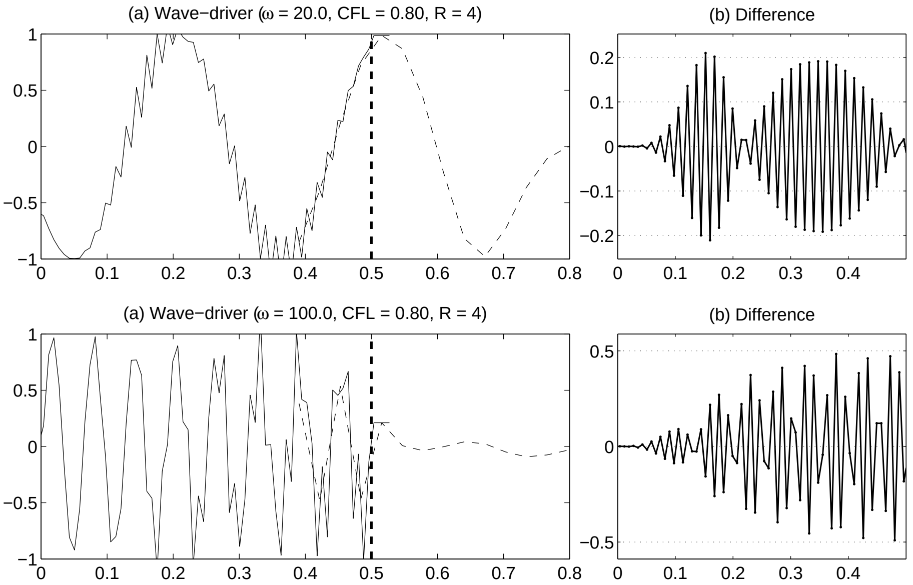


Figure 3.8: A wavemaker-driven simulation with PPM, $\Delta x_f = 1/128$, resolution ratio $R = 4$ and $CFL = 0.8$ taken at time $t = 0.75$. The forcing frequency is $\omega = 20.0$ (top) and $\omega = 100.0$ (bottom). A piecewise constant reconstruction is used at the resolution discontinuity ($x = 0.5$, thick dashed line) for remapping from the coarse grid ($x > 0.5$, thin dashed line) to the fine grid ($x < 0.5$, solid line). The simulation results are plotted in (a) and the parasitic mode (obtained from differencing the homogeneous resolution and refined resolution simulations) is plotted in (b). The abscissa represents the x coordinate and the ordinate shows the amplitude (both dimensionless).

parasitic mode decays away when $\omega = 100.0$. These results suggest that this choice of limiter is responsible for some damping of the spurious reflected mode, but is unable to remove it entirely in the presence of a low-frequency carrier wave. Nonetheless limiting is effective at high frequencies where the incident wave does not make an effective carrier.

3.3.5 Wave reflection by upwind FV schemes

Unlike the symmetric FV schemes, upwind FV schemes strongly damp high-frequency modes. We plot the results of four simulations using the FV2 and FV3p3

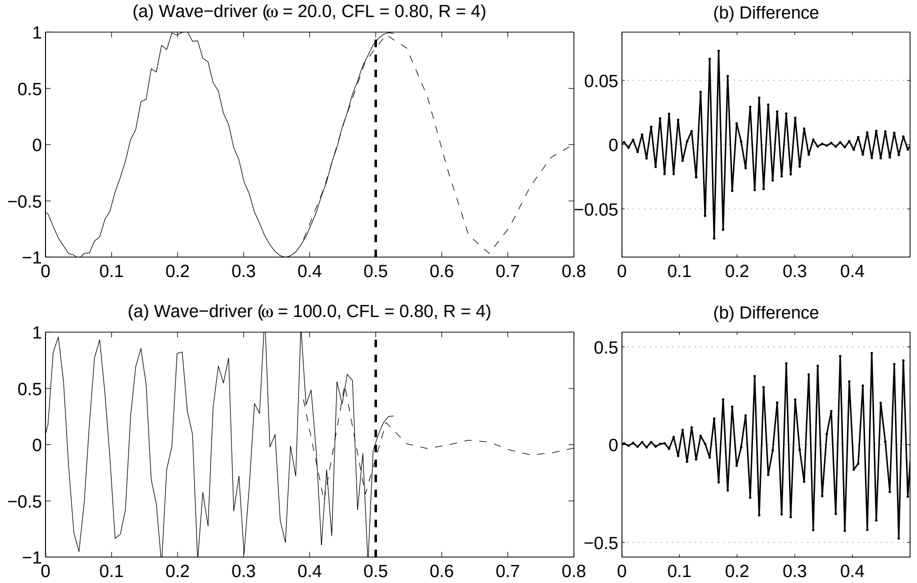


Figure 3.9: As Fig. 3.8 except with piecewise parabolic reconstruction at the resolution discontinuity.

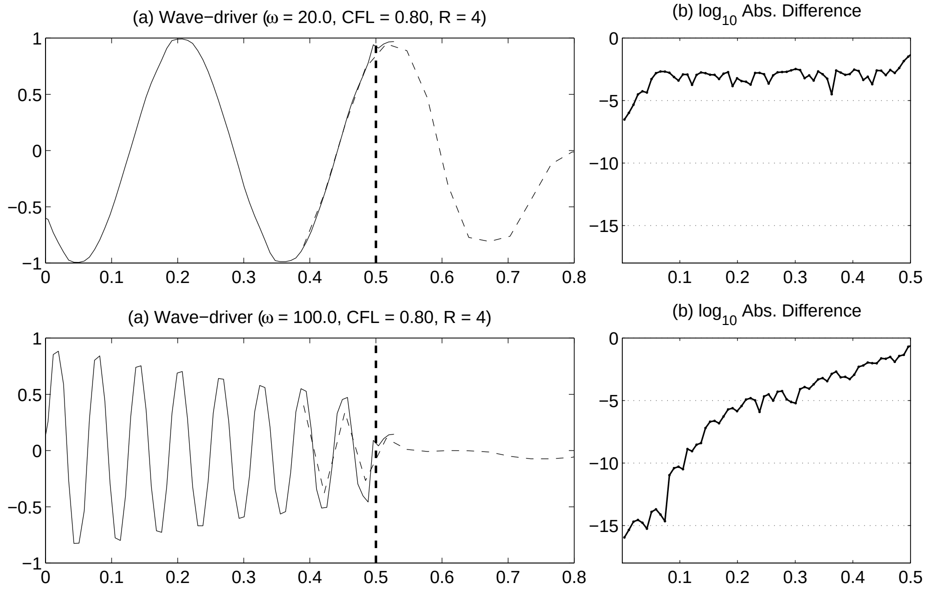


Figure 3.10: As Fig. 3.9 except with slope/curvature limiter. Note that we have plotted the difference on a logarithmic scale.

schemes with driving frequencies $\omega = 20.0$ and $\omega = 100.0$ in Fig. 3.11. We observe that the parasitic mode is present with the same initial amplitude as with PPM, but is quickly damped out. In all cases the decay rate of the parasitic mode agrees well with the theory derived in section 3.3.2.

3.3.6 Wave reflection by SLIM FV schemes

SLIM FV schemes behave similarly to upwind FV schemes, except the decay rate tends to exhibit a more interesting structure. We plot the results of two simulations using the SLIM3p3 schemes with driving frequencies $\omega = 20.0$ and $\omega = 100.0$ in Fig. 3.13. Again, our predictions for the decay rates from section 3.3.2 agree well with the simulations.

3.4 The 1D shallow-water equations and linearized 1D shallow-water equations

In this section we briefly turn our attention to the 1D shallow-water equations and demonstrate how the previous results for the advection equation can be generalized to this case. Unfortunately, our analysis of the 1D shallow-water equations does not generalize to higher dimensions, as would be relevant for geophysical flows, except for wave modes that encounter a grid resolution discontinuity at a right angle. In the case of a higher-dimensional flow, one must also take into consideration *wave refraction*, which results in the splitting of incident waves into reflected and transmitted components when the wave packet hits a grid resolution discontinuity at an oblique angle. Some analysis of wave refraction was tackled by *Cathers and Bates* (1995), but the complexity of this problem has largely prevented further study. Nonetheless, we believe that there is value in understanding how our results on pure wave reflection can be generalized to the 1D shallow-water equations.

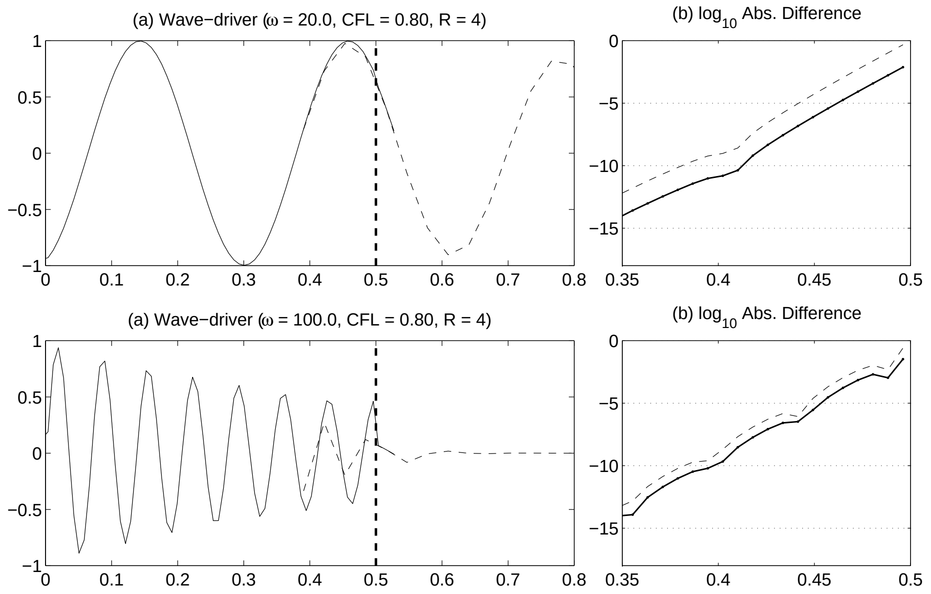


Figure 3.11: As Fig. 3.8 except for the FV2 scheme taken at time $t = 1.0$. The decay rate predicted in section 3.3.2 is shown as a dashed line in (b). Note the shorter horizontal range in (b).

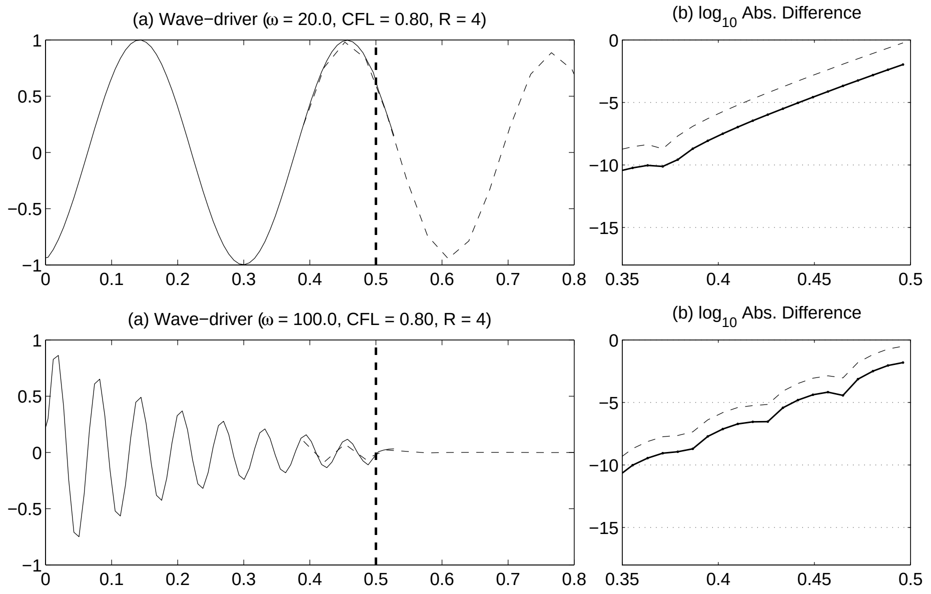


Figure 3.12: As Fig. 3.11 except for the FV3p3 scheme.

Traditionally, geophysical flows have been modeled using staggered grids (*i.e.* with mass and momentum variables stored at different points), since many unstaggered finite-difference approaches admit both spurious $2\Delta x$ modes and, for certain ranges of wavenumber k, lead to group velocities that have the wrong sign relative to the flow field. Such properties are absent in certain staggered discretizations. Previously, these problems with unstaggered schemes have been pointed out by *Fox-Rabinovitz* (1991) and *Randall* (1994). These results have led to a widespread adoption of staggered grids in geophysical flow models that make use of both finite-difference or finite-volume discretizations (see *Bonaventura and Ringler* (2005); *Lin* (2004); *Ahmad and Lindeman* (2007); *Schmidt et al.* (2006); *Thuburn et al.* (2009)) discretizations. For example, C-grid discretizations have desirable inertio-gravity wave dispersion characteristics. Unfortunately, not all of the attractive properties of staggered schemes on uniform grids carry over to refined grids with a resolution discontinuity. As observed by *Chin and Hedstrom* (1982) and more recently by *Frank and Reich* (2004), although staggered grids do not admit spurious backwards-propagating high-frequency modes, coupling of left- and rightgoing wave solutions leads to the generation of spurious physical modes at the point of grid refinement (see also 3.5). As a consequence, unstaggered schemes are potentially more desirable in this case.

To begin, we consider the 1D shallow-water equations in conservative form,

$$\begin{aligned}\frac{\partial h}{\partial t} &= -\frac{\partial m}{\partial x}, & (3.36)\\ \frac{\partial m}{\partial t} &= -\frac{\partial}{\partial x}\left(\frac{m^2}{h} + \tfrac{1}{2}gh^2\right), & (3.37)\end{aligned}$$

where h is the total height, $m = hu$ is the momentum and g is the gravitational constant. If we consider only linearized wave motions on a constant background height field H,

$$h = H + h', \quad \text{and} \quad m = m', \tag{3.38}$$

(where the prime denotes the deviations from the background fields) then the 1D shallow-water equations reduce to the linearized 1D shallow-water equations,

$$\begin{aligned}\frac{\partial h'}{\partial t} &= -\frac{\partial m'}{\partial x}, \qquad (3.39)\\ \frac{\partial m'}{\partial t} &= -gH\frac{\partial h'}{\partial x}. \qquad (3.40)\end{aligned}$$

The linearized 1D shallow-water equations support wave-like solutions of the form

$$\mathbf{q}(x,t) = \hat{\mathbf{q}}\exp(i(kx-\omega t)), \qquad (3.41)$$

where $\mathbf{q} = [h, m]$ is the state vector with amplitudes $\hat{\mathbf{q}} = [\hat{h}, \hat{m}]$. On substituting this relation into (3.39) and (3.40), we obtain

$$\underbrace{\begin{bmatrix} -i\omega & ik \\ gHik & -i\omega \end{bmatrix}}_{M} \begin{bmatrix} \hat{h} \\ \hat{m} \end{bmatrix} = 0. \qquad (3.42)$$

Hence, in order for wave solutions to exist, we must have $\det(M) = 0$, which implies a dispersion relation of the form

$$\omega = \pm k\sqrt{gH}, \qquad (3.43)$$

corresponding to a rightgoing mode ($\omega > 0$) and a leftgoing mode ($\omega < 0$), for each positive wave number. The quantity $\sqrt{gH}$ is the shallow-water gravity wave speed.

3.4.1 Riemann invariants

The full 1D shallow-water equations admit two Riemann invariants, denoted $\mathcal{L}$ and $\mathcal{R}$, of the form

$$\mathcal{L} = \frac{m}{h} - 2\sqrt{gh}, \quad \text{and} \quad \mathcal{R} = \frac{m}{h} + 2\sqrt{gh}. \tag{3.44}$$

These are propagated according to

$$\frac{\partial \mathcal{L}}{\partial t} + \left(\frac{m}{h} - \sqrt{gh}\right)\frac{\partial \mathcal{L}}{\partial x} = 0, \qquad \frac{\partial \mathcal{R}}{\partial t} + \left(\frac{m}{h} + \sqrt{gh}\right)\frac{\partial \mathcal{R}}{\partial x} = 0, \tag{3.45}$$

with only weak coupling between these modes due to the nonlinear wave speed (the parenthesized terms in (3.45)). Observe that for subcritical flow $m/h < \sqrt{gh}$, these modes are propagated leftward and rightward, respectively.

The linearized 1D shallow-water equations admit a leftgoing Riemann invariant $\mathcal{L}'$ and a rightgoing Riemann invariant $\mathcal{R}'$, defined in terms of h' and m' as

$$\mathcal{L}' = \frac{m'}{\sqrt{gH}} - h', \quad \text{and} \quad \mathcal{R}' = \frac{m'}{\sqrt{gH}} + h'. \tag{3.46}$$

The evolution of these quantities is then described by

$$\frac{\partial \mathcal{L}'}{\partial t} - \sqrt{gH}\frac{\partial \mathcal{L}'}{\partial x} = 0, \qquad \frac{\partial \mathcal{R}'}{\partial t} + \sqrt{gH}\frac{\partial \mathcal{R}'}{\partial x} = 0, \tag{3.47}$$

which is exactly the leftgoing and rightgoing advection equation with wave speed $\sqrt{gH}$.

Thus, for the subcritical shallow-water equations or linearized shallow-water equations we only obtain a well-posed system of equations if we specify the rightgoing Riemann invariant $\mathcal{R}$ at the left boundary and the leftgoing Riemann invariant $\mathcal{L}$ at the right boundary.

The main problem in generalizing the results for the 1D shallow-water equations to higher dimensions arises largely with the Riemann invariants, which are not well defined for higher-dimensional systems.

3.4.2 Numerical discretizations

The numerical discretizations introduced in section 3.2 can be easily formulated for the linearized 1D shallow-water equations. The SLIM scheme can also be generalized to the linearized 1D shallow-water equations by operating on Riemann invariants, but adapting this scheme to the full non-linear shallow-water equations is not immediately obvious. For this reason, in this section we will concentrate our efforts on the symmetric and upwind finite-volume schemes.

The gas-dynamics form of the PPM scheme (see section 3.2.4) for the 1D linearized shallow-water equations takes on the semi-discretization

$$\frac{\partial h_j}{\partial t} = -\left[\frac{m_{j-2} - 8m_{j-1} + 8m_{j+1} - m_{j+2}}{12\Delta x}\right], \tag{3.48}$$

$$\frac{\partial m_j}{\partial t} = -gH\left[\frac{h_{j-2} - 8h_{j-1} + 8h_{j+1} - h_{j+2}}{12\Delta x}\right]. \tag{3.49}$$

The familiar centered-difference operators are apparent on the right-hand-side of this formulation.

The upwind FV2 scheme (see section 3.2.5), on the other hand, takes on the semi-discretization

$$\begin{aligned}\frac{\partial h_j}{\partial t} &= -\left[\frac{m_{j-2} - 6m_{j-1} + 6m_{j+1} - m_{j+2}}{8\Delta x}\right] \\ &\quad + \frac{\Delta x^3\sqrt{gH}}{8}\left(\frac{-h_{j-2} + 4h_{j-1} - 6h_j + 4h_{j+1} - h_{j+2}}{\Delta x^4}\right), \end{aligned} \tag{3.50}$$

$$\begin{aligned}\frac{\partial m_j}{\partial t} &= -gH\left[\frac{h_{j-2} - 6h_{j-1} + 6h_{j+1} - h_{j+2}}{8\Delta x}\right] \\ &\quad + \frac{\Delta x^3\sqrt{gH}}{8}\left(\frac{-m_{j-2} + 4m_{j-1} - 6m_j + 4m_{j+1} - m_{j+2}}{\Delta x^4}\right). \end{aligned} \tag{3.51}$$

We observe that this method combines an $O(\Delta x^3)$ approximation to the advective term (first term on the RHS) with a diffusion term proportional to the fourth-derivative of the field (second term on the RHS). This combination of advective and diffusive terms is typical for upwind-type methods.

Finally, the upwind FV3p3 scheme (see section 3.2.6) has semi-discretization

$$\begin{aligned}
\frac{\partial h_j}{\partial t} &= -\left[\frac{m_{j-2} - 8m_{j-1} + 8m_{j+1} - m_{j+2}}{12\Delta x}\right] \\
&\quad + \frac{\Delta x^3 \sqrt{gH}}{12}\left(\frac{-h_{j-2} + 4h_{j-1} - 6h_j + 4h_{j+1} - h_{j+2}}{\Delta x^4}\right), \qquad (3.52) \\
\frac{\partial m_j}{\partial t} &= -gH\left[\frac{h_{j-2} - 8h_{j-1} + 8h_{j+1} - h_{j+2}}{12\Delta x}\right] \\
&\quad + \frac{\Delta x^3 \sqrt{gH}}{12}\left(\frac{-m_{j-2} + 4m_{j-1} - 6m_j + 4m_{j+1} - m_{j+2}}{\Delta x^4}\right). \qquad (3.53)
\end{aligned}$$

Here we observe that (3.52) is identical to (3.50), except with the $O(\Delta x^3)$ approximation to the first-derivative term replaced by a more accurate $O(\Delta x^4)$ approximation on the same stencil and with a slightly weaker diffusion term. Further, comparing with (3.48), we observe that this scheme has an identical advective component, and only differs in the addition of a diffusive term.

3.4.3 Leftgoing and rightgoing mode separation

As we see in equations (3.48)-(3.49), (3.50)-(3.51), (3.52)-(3.53), unstaggered linear finite-volume schemes lead to semi-discretizations of the linearized shallow-water equations that take the form

$$\begin{aligned}
\frac{\partial h_j}{\partial t} &= -\left(\sum_{a=-\ell}^{r} c_a m_{j+a}\right) + \sum_{a=-\ell}^{r} d_a h_{j+a}, \qquad (3.54) \\
\frac{\partial m_j}{\partial t} &= -gH\left(\sum_{a=-\ell}^{r} c_a h_{j+a}\right) + \sum_{a=-\ell}^{r} d_a m_{j+a}, \qquad (3.55)
\end{aligned}$$

where the coefficients c_a and d_a are constant in h and m, but are a function of the grid spacing Δx and wave speed $\sqrt{gH}$ (and should not be confused with the coefficients of the advection equation (3.11)). Here ℓ and r again denote the number of leftward-elements and rightward-elements in the stencil. Hence, under a linear timestepping operator, the discretizations (3.54) and (3.55) lead to an evolution equation for the leftgoing Riemann invariant $\mathcal{L}'_j$ given by

$$\frac{\partial \mathcal{L}'_j}{\partial t} = \sqrt{gH} \left(\sum_{a=-\ell}^{r} c_a \mathcal{L}'_{j+a} \right) + \sum_{a=-\ell}^{r} d_a \mathcal{L}'_{j+a}. \tag{3.56}$$

Similarly, the evolution equation for the rightgoing Riemann invariant $\mathcal{R}'_j$ is given by

$$\frac{\partial \mathcal{R}'_j}{\partial t} = -\sqrt{gH} \left(\sum_{a=-\ell}^{r} c_a \mathcal{R}'_{j+a} \right) + \sum_{a=-\ell}^{r} d_a \mathcal{R}'_{j+a}. \tag{3.57}$$

First, observe that as long as the discretization of the temporal derivative is linear with respect to the spatial derivative (such as from an Eulerian or Runge-Kutta timestepping scheme) these equations have decoupled from one another. Second, observe that these equations are simply the discretization of the advection equation associated with the same discrete spatial operator as in (3.54). This result implies that our analysis of the advection equation in section 3.3.1 should also apply to the 1D linearized shallow-water equations, and hence our analysis should accurately describe the dominant forcing mechanism for parasitic waves in the full 1D shallow-water equations.

3.4.4 Wave reflection due to coupling of Riemann invariants

Unlike the advection equation, the shallow-water system can generate spurious waves at a grid resolution discontinuity by artificially transferring energy between Riemann invariants. The 1D shallow-water equations, and their linear counterparts admit both leftgoing and rightgoing Riemann invariants for the case of subcritical flow. A grid resolution discontinuity can trigger an interaction between these modes which results in a spurious transfer of energy between one or more Riemann invariants. On the unstaggered grid, wave reflection of this type does not arise in discretizations of the 1D linearized shallow-water equations, but does arise in the nonlinear shallow-water equations due to the aforementioned weak coupling of Riemann invariants (see section 3.4.1).

The amplitude of these spurious physical modes is strongly dependent on the degree of nonlinearity present in the system, which can be characterized via the shallow-water Froude number,

$$Fr = \frac{u'}{\sqrt{gh}}, \tag{3.58}$$

where u' denotes the perturbation from some mean velocity. Systems with larger Froude number have a greater tendency to lead to abruptly varying flows (such as breaking waves) that are more strongly reflected. In general, repeated simulations have shown that the amplitude of a spurious physical mode generated by interaction of Riemann invariants will increase on coarser grids (larger Δx) and increasing resolution ratio R. For a flow with a Froude number of 0.1, we have observed that sharp gradients can readily lead to accumulated errors in the "backwards" propagating Riemann invariant on the order of 1%.

3.5 A brief note on staggered FV schemes for the linear shallow-water equations

Strict conservation of momentum is sometimes unnecessary and external forcing from source terms – especially in the context of geophysical flows – often prevents exact conservation. Hence, it has become common practice (see, for instance, *Lin and Rood* (1997); *Marshall et al.* (1997)) to combine a finite-volume scheme for the mass variable (in this case h) with a finite-difference scheme for the momentum component. Further, due to the resulting beneficial numerical properties (see, for example, *Randall* (1994)), velocity points are typically placed along edges of the height volumes. This combination leads us to a family of staggered grid schemes based on the finite-volume framework.

The most basic staggered FV scheme is the second-order central-in-space (CiS) scheme. Under this scheme, the height and momentum evolution equations satisfy the semi-discretization

$$\begin{aligned} \frac{\partial h_j}{\partial t} &= -\left[\frac{m_{j+1/2} - m_{j-1/2}}{\Delta x_j}\right], && (3.59) \\ \frac{\partial m_{j+1/2}}{\partial t} &= -gH\left[\frac{h_{j+1} - h_j}{\Delta x_{j+1/2}}\right]. && (3.60) \end{aligned}$$

As with the symmetric FV scheme, the eigenvalues of this semi-discretization are purely imaginary, and so must be paired with at least a third-order Runge-Kutta timestepping operator. If we do so, we obtain a scheme which is stable up to a CFL number of approximately 0.866.

The grid resolution discontinuity is more problematic for staggered schemes, since the momentum is stored pointwise at the grid discontinuity. In order to maintain high-order accuracy consistent with the CiS scheme, we require a $O(\Delta x^2)$ reconstruction for $\partial h/\partial x$ that cannot be obtained by simply using neighboring element-averages of h. For a discontinuity with resolution ratio $R \geq 1$, left-grid width Δx_f and right-grid width $\Delta x_c = R\Delta x_f$, we find

$$\frac{\partial m_{N+1/2}}{\partial t} = -gH\frac{R}{\Delta x_c(2+R)(1+R)}\left[(1-R^2)h_{N-1} - (7-R^2)h_N + 6h_{N+1}\right] + O(\Delta x_c^2), \quad (3.61)$$

where index $N + 1/2$ corresponds to the position of the resolution discontinuity.

The main problem with using a staggered scheme in combination with an abrupt grid resolution discontinuity can be observed directly from numerical experiments. We plot two such experiments in Fig. 3.14. Unlike with unstaggered schemes, the staggered finite-volume discretizations do not have decoupled Riemann invariants, and so allow energy to be transferred between leftgoing and rightgoing waves. This interaction is strongly dependent on the choice of boundary reconstruction and timestep scheme, since these two factors determine which wavenumbers are available at a given frequency. Observe that in Fig. 3.14 (top) the parasitic mode does not have the same wavelength as the incident wave; a similar numerical experiment using a reconstruction analogous to (3.60) at the discontinuity (note that such a reconstruction is first-order at this point) produces a parasitic mode with wavelength equal to the incident wave. At high-frequencies the scheme is unable to transfer energy into the forward-propagating mode on the coarse grid (since the wave cannot be resolved in this region) and so must transfer this energy into a backward-propagating mode. As a result, we observe that at high-frequencies most of the energy of the incident wave translates into a backwards-propagating mode.

Coupling of wave modes in the staggered scheme described in this section suggests that the reflected mode is almost indistinguishable from an incident physical mode. As a consequence, filters that remove high-frequency Fourier modes near the resolution discontinuity will not be able to detect these waves.

From these observations we conclude that staggered schemes that are constructed similar to the one described above are unsuitable for application on grids that have an abrupt grid resolution discontinuity. A thorough analysis of staggered schemes on a refined grid has been given by *Long* (2009) (also see *Long and Thuburn* (2011)).

3.6 Conclusions

In this chapter we have considered symmetric, upwind and semi-Lagrangian integrated mass (SLIM) numerical discretizations of the 1D advection equation in the

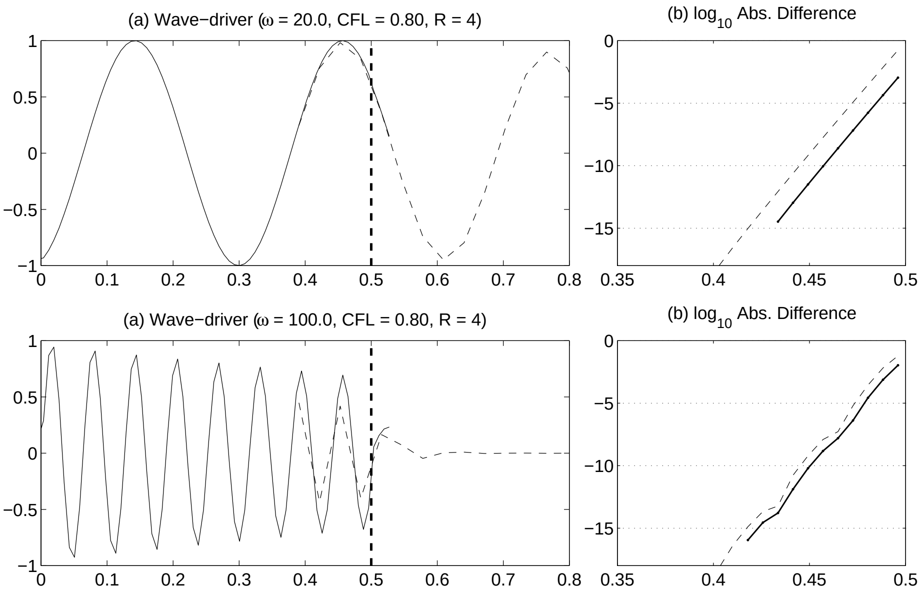


Figure 3.13: As Fig. 3.11 except for the SLIM3p3 scheme.

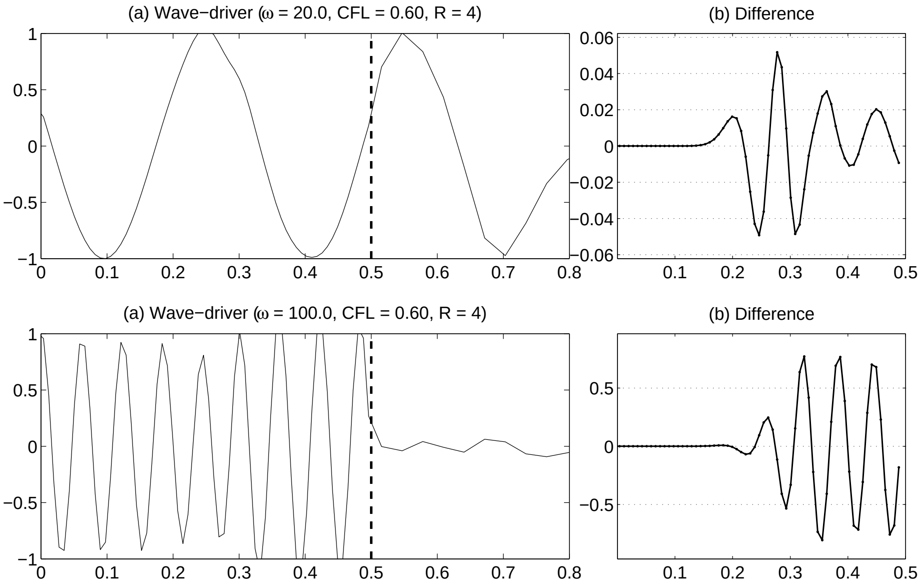


Figure 3.14: A wavemaker-driven simulation with the second-order CiS scheme with $\Delta x_f = 1/128$, resolution ratio $R = 4$ and $CFL = 0.6$. The forcing frequency is $\omega = 20.0$ (top) and $\omega = 100.0$ (bottom). The simulation results at $t = 0.8$ are plotted in (a) and the parasitic mode (obtained from differencing the homogeneous resolution and refined resolution simulations) is plotted in (b). The abscissa represents the x coordinate and the ordinate shows the amplitude of h (both dimensionless).

presence of an abrupt discontinuity in grid resolution. We have presented approaches for characterizing the initial amplitude of a parasitic mode as well as its decay rate away from a grid resolution discontinuity. An analysis of the diffusion and group velocity of the upwind and SLIM schemes has revealed that upwind schemes largely remove any spurious modes that would normally be carried "backwards" relative to the flow. Unstaggered symmetric schemes, on the other hand, do not damp the parasitic modes, which must be dealt with through some alternative mechanism. We have also examined symmetric schemes which have been combined with a typical slope/curvature limiter, but found that although this strategy is effective at removing oscillations in the high-frequency regime, the parasitic mode is retained for relative low frequencies. Tests using high-order accurate reconstructions at the grid resolution discontinuity have shown that although increasing the order of accuracy of the reconstruction is effective for low-frequency modes, at high-frequencies parasitic modes are again retained.

The 1D shallow-water equations were also considered briefly. Although staggered discretizations have typically been used in the context of geophysical flows, it has been argued that these schemes are unsuitable in the presence of an abrupt resolution discontinuity, since they trigger spurious physical modes which are difficult to remove (additional details are presented in 3.5 and *Frank and Reich* (2004)). In the 1D case, the unstaggered schemes we have considered have the benefit of decoupling leftgoing and rightgoing Riemann invariants, and so our analysis of the advection equation can be applied directly.

Clearly, spurious wave reflection due to an abrupt grid resolution discontinuity is a significant problem that can result in severe degradation of the performance of any numerical method. Hence, some mechanism must be present in order to remove contamination by these modes. For symmetric schemes this mechanism likely should come in the form of an explicit diffusion term, which is naturally present in upwind schemes. With this additional diffusion term, upwind schemes perform very well at damping out reflected oscillations for the linear equations. Nonetheless, proper care must be taken for the treatment of nonlinear effects which can also lead to the generation of spurious physical modes.

This work can be found in *Ullrich and Jablonowski* (2011b).

CHAPTER IV

High-order Finite-Volume Methods

4.1 Introduction

Atmospheric models are difficult to engineer, largely due to two factors. Firstly, the flow occurs over the surface of a sphere, rather than in much simpler planar Cartesian geometry and secondly, there are vast scale differences between the large-scale horizontal flow, with length scales that extend to thousands of kilometers, and vertical motions with length scales of about 1-10 km. In addition, the dominant motions in the atmosphere are an example of a low Mach number regime that is mostly characterized by Mach numbers around $M < 0.4$. Therefore, care must be taken when applying numerical methods from other research fields. In particular, in atmospheric flows high-speed motions are only present in fast atmospheric gravity waves or sound waves. The latter are a solution to the 3D nonhydrostatic equation set, but play a negligible role from a physical viewpoint. Nevertheless, an adequate numerical scheme for atmospheric flows must guarantee stability for fast waves and treat the slow, physically important, motions with high accuracy.

A common test bed for atmospheric model development is based on the shallow-water equation set that mimics atmospheric flow in a single layer. A shallow-water model thereby tests the horizontal and temporal discretizations and provides guidance for the numerical schemes suitable for flows with low Mach numbers. Note that the shallow-water equations do not support sound waves but do capture the fast gravity wave propagation.

There are many numerical schemes that have been tested in shallow-water models on the sphere, all of which have both pros and cons. The spectral transform method discussed in *Jakob-Chien et al.* (1995) achieves high accuracy but tends to exhibit non-physical numerical oscillations near sharp gradients – known as Gibb's ringing. Spectral transform methods also demand a high computational expense at high resolution that is associated with the computational cost of the Legendre transforms. Finite-difference approaches include those of *Heikes and Randall* (1995) and *Ronchi et al.* (1996). Hybrid finite-volume methods incorporate both a finite-volume treatment of conservative variables and a finite-difference treatment of momentum and include the models of *Lin and Rood* (1997) and *Chen and Xiao* (2008). Finite-element type models, including spectral-element (SE) and discontinuous-Galerkin (DG) models have been presented by *Taylor et al.* (1997), *Côté and Staniforth* (1990), *Thomas and Loft* (2005), *Giraldo et al.* (2002) and *Nair et al.* (2005).

The aforementioned models represent a wide variety of computational grids on the sphere such as the latitude-longitude mesh, icosahedral and hexagonal grids, and cubed-spheres meshes. The latter three have become popular over the last decade as they provide an almost regular grid point coverage on the sphere. The uniform distribution of elements avoids the convergence of the meridians that is characteristic for latitude-longitude grids, and thereby alleviates the use of polar filters and other numerical damping techniques. The cubed-sphere grid has also been proven to scale efficiently on massively parallel computing platforms as shown by *Taylor et al.* (2008) and *Putman and Lin* (2009). These two models are therefore under consideration for operational climate and weather applications at atmospheric modeling centers in the U.S..

This paper introduces a set of third- and fourth-order-accurate fully-conservative finite-volume methods on cubed-sphere grids and assesses the impact of the high-order accuracy. These finite-volume methods are built upon the reconstruction techniques adopted by the Monotone Upstream-centered Schemes for Conservation Laws (MUSCL) pioneered by *van Leer* (1979). Previously, second-order finite-volume methods of this type have been studied for geostropic flows on the sphere by *Rossmanith* (2006), which is based on the flux-difference-splitting technique of *LeVeque* (1997) on a curved manifold.

Fully-conservative finite-volume methods share local conservation properties with spectral-element and discontinuous-Galerkin discretizations, but are potentially more computationally efficient due to their relatively weak Courant-Friedrichs-Lewy (CFL) constraints. Explicit timestepping techniques, when used in combination with these methods, suffer from severe CFL timestep restrictions related to the clustering of nodal points near element edges (which worsens at high-order). On the other hand, finite-volume methods possess a large computational stencil at high-order and so are also potentially difficult to parallelize as effectively as these more compact methods. This difficulty arises primarily in the algorithmic complexity associated with determining which information needs to be communicated between processors. Although DG and SE methods only require information to be communicated between elements and their immediate neighbors, the number of prognotistic quantities associated with each element is significantly larger for these schemes. Whereas for each state variable finite-volume methods store only one value per element, DG and SE methods can, at fourth-order-accuracy, can have up to ten values per element.

The use of neighboring elements by high-order FV schemes also means that element values must be remapped across coordinate discontinuities, such as those that appear on the cubed-sphere grid. This requirement results in the need for wider ghost regions near coordinate discontinuities on parallel systems in order to accommodate remapping. Schemes with local degrees of freedom, on the other hand, including

DG and SE methods, may be more attractive in this regard since remapping is not required, and hence work can be distributed more evenly on parallel architectures.

Although we do not present a technique for constructing a monotone or non-oscillatory scheme in this chapter, significant research has been done on this topic for applications in other research areas. For instance, (Weighted) Essentially Non-Oscillatory ((W)ENO)-type reconstructions (e.g. (*Alcrudo and Garcia-Navarro*, 1993; *Noelle et al.*, 2006)), slope limiters (e.g. (*Ullrich et al.*, 2009; *Liu et al.*, 2007)) or flux-corrected transport methods (*Zalesak*, 1979) can all be applied to this class of finite-volume methods presented herein. Monotone DG methods, on the other hand, are an active research area.

The method we present involves the use of approximate Riemann solvers to calculate edge fluxes. Most widely used approximate Riemann solvers (such as the solver of *Roe* (1981)) are designed to model flow in the transsonic or supersonic regime rather than in the relatively slow flow regime that is typical for the atmosphere. However, recent advances in the design of approximate Riemann solvers have led to an extension of the Advection Upstream Splitting Method (AUSM, (*Liou and Steffen*, 1993)) to low Mach numbers (*Liou* (2006)). The use of this new numerical flux formulation, known as AUSM^{+}-up, has so-far been largely limited to the aerospace community. Hence, a test of this new approximate Riemann solver will gauge its applicability for atmospheric models. We also compare the Roe and AUSM^{+}-up schemes to the widely-used and simpler Rusanov solution (*Lax*, 1954; *Rusanov*, 1961; *Toro*, 1999).

The performance of all schemes will be analyzed via selected standard test cases from the suite of *Williamson et al.* (1992). Among them are the advection of a cosine bell, steady-state geostrophic flow, steady-state geostrophic flow with compact support, flow over an isolated mountain and the Rossby-Haurwitz wave. In addition, we assess the barotropic instability problem of *Galewsky et al.* (2004) that exhibits sharp vorticity gradients.

The paper is organized as follows. In section 4.2 we introduce the cubed-sphere grid with an equiangular projection, which is the underlying grid for all simulations. Section 4.3 discusses the shallow-water equations in cubed-sphere geometry. The high-order finite-volume framework is described in section 4.4. Special attention is paid to a careful discretization of the topography term. Section 4.5 gives two examples of finite-volume methods that can be composed under this framework. In particular, the third-order dimension-split piecewise parabolic method and a fourth-order piecewise cubic-method are introduced. Section 4.6 describes the three approximate shallow-water Riemann solvers we will be using in our analysis. The simulation results, discussion and performance assessments are presented in section 4.7. Finally, the main findings are summarized in section 4.8.

4.2 The Cubed-Sphere

We make use of the cubed-sphere grid, which is obtained from projecting a gridded cube onto the surface of the sphere. The cubed-sphere grid has been suggested by *Sadourny* (1972) and *Ronchi et al.* (1996), and has become popular in recent years as an alternative to the classical spherical latitude-longitude mesh. The latter requires special treatment of singularities at the North and South poles due to convergence of the meridians. The cubed-sphere grid instead replaces these two strong singularities with eight weaker singularities that occur at the intersections of three cube faces. These intersections are the corner points of the original cube.

From a mathematical standpoint, the cubed-sphere grid is a tiling of the sphere consisting of six panels that form the faces of a concentric cube projected onto the surface of the sphere. Multiple options exist for the choice of grid on each panel, such as the gnomonic grid, which follows from applying a Cartesian grid to each panel of the cube, or the cubic conformal grid, which maximizes the orthogonality of coordinate vectors (see, for instance, *Putman and Lin* (2007) for a review of the types

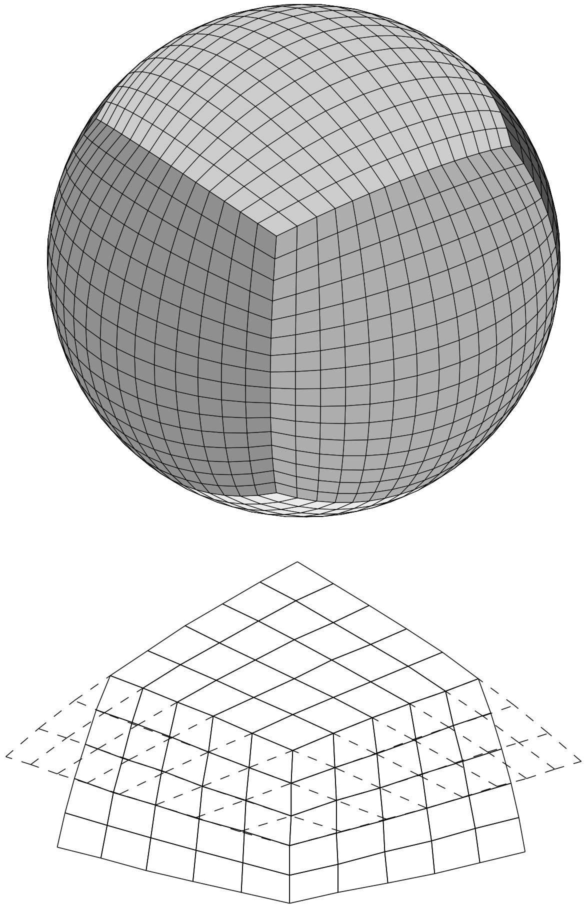


Figure 4.1: **Top:** A 3D view of the tiling of the cubed-sphere, shown here with a 16×16 tiling of elements on each panel. **Bottom:** A closeup view of the corner of the cubed-sphere, showing the overlap of grid lines from the upper panel ghost cells on the neighbouring panels.

of cubed sphere grids). In our model we will make use of the gnomonic (equiangular) cubed-sphere grid, which uses grid lines that have equal central angles relative to the center of the sphere (*i.e.*, this property is also exhibited by equispaced lines of constant longitude). This choice of grid projection leads to elements of similar size, and further leads to coincident grid lines on neighbouring panels. A depiction of the cubed-sphere grid and its singularities are given in Figure 4.1.

A point on the cubed-sphere in the equiangular projection can be given in terms of equiangular coordinates (α, β, n_p), with $\alpha, \beta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, or in terms of gnomonic coordinates (X, Y, n_p), with $X, Y \in [-1, 1]$. In both cases the panel number $n_p \in \{1, 2, 3, 4, 5, 6\}$. By convention, we choose panels $1-4$ to be along the equator, with panels 5 and 6 centered on the northern and southern pole, respectively. One can think of equiangular coordinates as being along the surface of the sphere, whereas gnomonic coordinates are along the surface of the cube. These two coordinate systems are related via

$$X = \tan\alpha, \qquad Y = \tan\beta. \tag{4.1}$$

In this chapter we will also make use of the definition

$$\delta = \sqrt{1 + X^2 + Y^2}, \tag{4.2}$$

which appears frequently in the calculation of metric quantities associated with the cubed-sphere.

The discrete resolution of the cubed sphere is usually written in the form $N_c \times N_c \times 6$ (in the case of symmetric tiling in the α and β direction), where N_c denotes the number of grid cells in each horizontal direction on a panel. A list of some properties of the cubed sphere grid is given in Table 4.1. The table lists the approximate equatorial spacing of grid elements, average area per element on the sphere, maximum area ratio and equivalent model resolutions of the regular latitude-longitude finite-volume and

Table 4.1: Properties of the cubed sphere grid for different resolutions. Here Δx is the grid spacing at the equator, A_{avg} is the average area of all cubed sphere grid elements, A_{min} is the minimum element area and A_{max} is the maximum element area. RLL_{equiv} denotes the equivalent grid spacing (in degrees) on the regular latitude-longitude grid with the same number of elements and T_{equiv} denotes the approximate triangular truncation of a spectral transform method.

Resolution	Δx	A_{avg}	A_{min}/A_{max}	RLL_{equiv}	T_{equiv}
$20 \times 20 \times 6$	500 km	2.125×10^5 km^2	0.7359	5.2°	$T21$
$40 \times 40 \times 6$	250 km	5.313×10^4 km^2	0.7213	2.6°	$T42$
$80 \times 80 \times 6$	125 km	1.328×10^4 km^2	0.7141	1.3°	$T85$
$160 \times 160 \times 6$	62.5 km	3.321×10^3 km^2	0.7106	0.65°	$T170$

spectral transform models (under triangular truncation, as argued by *Williamson* (2008)).

4.3 The Shallow-Water Equations on the Cubed-Sphere

Under equiangular coordinates, the covariant 2D metric on the cubed-sphere (see, for example, *Nair et al.* (2005)) is given by

$$g_{ij} = \frac{r^2(1+X^2)(1+Y^2)}{\delta^4} \begin{pmatrix} 1+X^2 & -XY \\ -XY & 1+Y^2 \end{pmatrix}, \tag{4.3}$$

with contravariant inverse

$$g^{ij} = \frac{\delta^2}{r^2(1+X^2)(1+Y^2)} \begin{pmatrix} 1+Y^2 & XY \\ XY & 1+X^2 \end{pmatrix}. \tag{4.4}$$

The square root of the metric determinant, denoted by J, is then

$$J = \sqrt{\det(g_{ij})} = \frac{r^2(1+X^2)(1+Y^2)}{\delta^3}. \tag{4.5}$$

The quantity J is exactly the Jacobian of the associated coordinate transform, and corresponds to the area of an infinitesmal region $d\alpha \times d\beta$. Without loss of generality we will set the radius of the sphere r to one, which fixes the characteristic length scale to be in terms of Earth radii.

The shallow-water equations in equiangular coordinates can be written as a conservation law of the form (summation over repeated indices is implied)

$$\frac{\partial}{\partial t}\mathbf{q}(\mathbf{x},t) + \frac{1}{J}\frac{\partial}{\partial x^k}\mathbf{F}^k = \mathbf{\Psi}(\mathbf{q},\mathbf{x}), \tag{4.6}$$

where $\mathbf{q}$ is the state vector describing the properties of the fluid at each point, $\mathbf{F}^k$ are flux vectors describing the physical response of the flow to gradients in the state vector and $\mathbf{\Psi}$ denotes forcing due to source terms. The state vector $\mathbf{q}$ consists of the height of the fluid, denoted by h, and its horizontal momentum $h\mathbf{u}$, which we can write as a linear combination of the basis vectors along the cubed-sphere as

$$h\mathbf{u} = hu^1\mathbf{g}_1 + hu^2\mathbf{g}_2, \tag{4.7}$$

where $\mathbf{g}_1$ and $\mathbf{g}_2$ denote the geometric basis vectors in the α and β directions, respectively. Hence, hu^1 and hu^2 can be thought of as the components of the angular momentum along geodesics that are aligned with the grid.[1] The state vector and flux vector then take the form

$$\mathbf{q}(\mathbf{x},t) = \begin{bmatrix} h \\ hu^1 \\ hu^2 \end{bmatrix}, \qquad \mathbf{F}^k = J \begin{bmatrix} U^k \\ \mathcal{T}^{k1} \\ \mathcal{T}^{k2} \end{bmatrix}. \tag{4.8}$$

[1] Note that u^k does not denote exponentiation of u, and should instead be read as "the k^{th} contravariant component of the vector $\mathbf{u}$." Unfortunately, the overloaded nature of this notation may be (understandably) confusing.

The source terms can be written as

$$\boldsymbol{\Psi} = \begin{bmatrix} 0 \\ \Psi^1_M + \Psi^1_C + \Psi^1_B \\ \Psi^2_M + \Psi^2_C + \Psi^2_B \end{bmatrix}, \tag{4.9}$$

where $\boldsymbol{\Psi}_M$, $\boldsymbol{\Psi}_C$ and $\boldsymbol{\Psi}_B$ are 2-component vectors and correspond to the forcing of the momentum terms due to the metric, the Coriolis force and the bottom topography respectively. Here we have denoted the "mass" flux vector by U^k and the "momentum" flux tensor by $\mathcal{T}^{kn}$. In terms of the state vector $\mathbf{q}$ the components of the flux can be written as

$$U^k = hu^k, \quad \text{and} \quad \mathcal{T}^{kn} = hu^k u^n + g^{kn}\tfrac{1}{2}Gh^2, \tag{4.10}$$

where G denotes the gravitational constant.

In general, the metric source term describes forcing due to the underlying curvature of the coordinate system and, in general curvilinear coordinates, takes the form

$$\boldsymbol{\Psi}^i_M = -\Gamma^i{}_{nk}\mathcal{T}^{kn}, \tag{4.11}$$

where $\Gamma^i{}_{nk}$ are the Christoffel symbols of the second kind associated with the metric. In particular, Christoffel symbols can be thought of as terms describing the k^{th} component of the variation of the n^{th} geometric basis vector in the i^{th} direction, and hence can be written as

$$\Gamma^i{}_{nk} = \frac{\partial \mathbf{g}_n}{\partial x^k} \cdot \mathbf{g}^i. \tag{4.12}$$

Note that this definition implies that the Christoffel symbols are exactly zero in a

Cartesian frame. In component form, the Coriolis source term is given by

$$\mathbf{\Psi}_C^i = -f\mathbf{k} \times h\mathbf{u} = \frac{f}{J}\begin{bmatrix} g_{12} & g_{22} \\ -g_{11} & -g_{12} \end{bmatrix}\begin{bmatrix} hu^1 \\ hu^2 \end{bmatrix}, \tag{4.13}$$

where $f = 2\Omega \sin\theta$ is the Coriolis parameter in terms of the angular velocity of the Earth Ω and the latitude θ. Finally, the source terms due to varying bottom topography (denoted by z) can be written as

$$\mathbf{\Psi}_B^i = -Gh\nabla^i z = -Ghg^{ij}\frac{\partial z}{\partial x^j}. \tag{4.14}$$

Given the metric (4.3) associated with the equiangular cubed sphere, the metric source term can be written as

$$\mathbf{\Psi}_M = \frac{2}{\delta^2}\begin{bmatrix} -XY^2hu^1u^1 + Y(1+Y^2)hu^1u^2 \\ X(1+X^2)hu^1u^2 - X^2Yhu^2u^2 \end{bmatrix}. \tag{4.15}$$

Note that in the special case of equiangular coordinates we have that $\Gamma_{kn}^i g^{kn} = 0$, which removes any dependence of the metric source term on the gravitational term $\frac{1}{2}Gh^2$. The Coriolis source term differs depending on whether the underlying panel is equatorial or polar, since

$$\sin\theta = \begin{cases} \frac{Y}{\delta} & \text{if } n_p \in \{1,2,3,4\}, \\ \frac{p}{\delta} & \text{if } n_p \in \{5,6\}, \end{cases} \tag{4.16}$$

where p is a panel indicator given by, for instance,

$$p = \text{sign}(\theta) = \begin{cases} 1 & \text{on the northern panel } (n_p = 5), \\ -1 & \text{on the southern panel } (n_p = 6). \end{cases} \tag{4.17}$$

Hence, for equatorial panels, we have

$$\mathbf{\Psi}_{C,eq.} = \frac{2\Omega}{\delta^2} \begin{bmatrix} -XY^2 & Y(1+Y^2) \\ -Y(1+X^2) & XY^2 \end{bmatrix} \begin{bmatrix} hu^1 \\ hu^2 \end{bmatrix}. \tag{4.18}$$

And for polar panels, we have

$$\mathbf{\Psi}_{C,pol.} = \frac{2p\Omega}{\delta^2} \begin{bmatrix} -XY & (1+Y^2) \\ -(1+X^2) & XY \end{bmatrix} \begin{bmatrix} hu^1 \\ hu^2 \end{bmatrix}. \tag{4.19}$$

4.4 The High-Order Finite-Volume Approach

In this section we present the high-order finite-volume approach we use as a framework for solving the shallow-water equations in cubed-sphere geometry.

4.4.1 Overview

In the full finite-volume approach we first integrate the shallow-water conservation laws in the form (4.6) over a given element $\mathcal{Z}$ (with area $|\mathcal{Z}|$) and make use of Gauss' divergence theorem to write the flux term as an integral around the boundary $\partial\mathcal{Z}$, giving

$$\frac{\partial}{\partial t}\overline{\mathbf{q}} + \frac{1}{|\mathcal{Z}|} \oint\limits_{\partial\mathcal{Z}} \mathbf{F}^k(\mathbf{U}) d\ell = \overline{\mathbf{\Psi}}, \tag{4.20}$$

where the integration is along the line segment $d\ell$ and the overline denotes an average of the form

$$\overline{\phi} = \frac{1}{|\mathcal{Z}|} \int\limits_{\mathcal{Z}} \phi dV, \tag{4.21}$$

where $dV = J d\alpha d\beta$ denotes the infinitesmal volume element. Note that the volume-averaged formulation (4.20) is exactly equivalent to the original shallow-water equations, and it is left to us to define an appropriate discretization over each of the terms in this expression.

In this chapter we will only consider discretizations where each panel consists of a regular rectangular arrangement of elements of dimension $N_c \times N_c$. The angle subtended by an element is then defined by

$$\Delta = \frac{1}{N_c}\frac{\pi}{2}. \tag{4.22}$$

Hence, for each cubed sphere panel with equiangular coordinate axes (α, β) and equiangular element arrangement, we can define

$$\alpha_i = -\frac{\pi}{4} + \left(i - \tfrac{1}{2}\right)\Delta, \quad \text{and} \quad \beta_j = -\frac{\pi}{4} + \left(j - \tfrac{1}{2}\right)\Delta, \tag{4.23}$$

where full indices $i, j = 1, \ldots, N_c$ are used to denote element center-points and half-indices $i, j = (\frac{1}{2}, \frac{3}{2}, \ldots, \frac{N_c+1}{2})$ are used to denote element edges. Hence, the region in (α, β)-space occupied by the element (i, j) is defined by

$$\mathcal{Z}_{ij} = \left[\alpha_{i-1/2}, \alpha_{i+1/2}\right] \times \left[\beta_{j-1/2}, \beta_{j+1/2}\right]. \tag{4.24}$$

4.4.2 Orthonormalization and the Orthonormal Riemann Problem

The schemes discussed in this chapter all transform the reconstructed velocity field at element edges into an orthonormal frame consisting of velocity components perpendicular and parallel to the element edge (here denoted by u and v). This approach significantly reduces the complexity of the problem since we only need to solve orthonormal Riemann problems (see Section 4.6) in order to obtain the corresponding fluxes across each interface. In particular, this approach allows us to sidestep problems due to the discontinuity of the coordinate system at panel edges, since the Riemann problem is solved in a single consistent reference frame. Orthonormalization is performed via multiplication with an *orthonormalization matrix* at each point (see, for instance, *Bale* (2002)),

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathcal{O}(X,Y) \begin{bmatrix} u^1 \\ u^2 \end{bmatrix}, \tag{4.25}$$

where $\mathcal{O}$ also depends on the type of edge. In general, for translating velocity components along lines of constant α and constant β we have

$$\mathcal{O}^1 = \begin{bmatrix} \dfrac{1}{\sqrt{g^{11}}} & 0 \\ \dfrac{g_{12}}{\sqrt{g_{22}}} & \sqrt{g_{22}} \end{bmatrix}, \quad \text{and} \quad \mathcal{O}^2 = \begin{bmatrix} 0 & \dfrac{1}{\sqrt{g^{22}}} \\ \sqrt{g_{11}} & \dfrac{g_{12}}{\sqrt{g_{11}}} \end{bmatrix}. \tag{4.26}$$

Using the metric for equiangular cubed sphere coordinates, we have for grid lines of constant α that the orthonormalization matrix is

$$\mathcal{O}^1 = \begin{bmatrix} \dfrac{\sqrt{1+X^2}}{\delta} & 0 \\ \dfrac{-XY\sqrt{1+X^2}}{\delta^2} & \dfrac{(1+Y^2)\sqrt{1+X^2}}{\delta^2} \end{bmatrix}, \tag{4.27}$$

and for grid lines of constant β we have

$$\mathcal{O}^2 = \begin{bmatrix} 0 & \dfrac{\sqrt{1+Y^2}}{\delta} \\ \dfrac{(1+X^2)\sqrt{1+Y^2}}{\delta^2} & \dfrac{-XY\sqrt{1+Y^2}}{\delta^2} \end{bmatrix}. \tag{4.28}$$

Similarly, to obtain the components of the momentum flux in the equiangular cubed-sphere basis we apply a *deorthonormalization matrix* at each point, which is simply the inverse of the corresponding orthonormalization matrix.

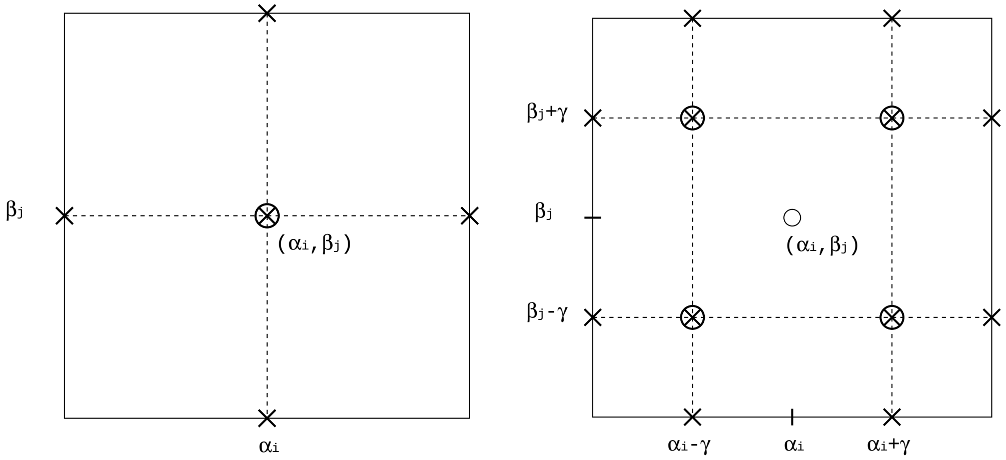


Figure 4.2: Gaussian quadrature points used for a first- or second-order finite-volume scheme (left) and for a third- and fourth-order finite-volume scheme (right). Edge points used for calculating fluxes through the boundary are depicted as uncircled ×'s. Interior quadrature points are depicted as circled ×'s. Here γ is chosen so that the Gaussian quadrature is fourth-order accurate in the size of the grid.

In orthonormal form, the source-free shallow-water equations are given by

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0, \tag{4.29}$$

$$\frac{\partial (hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \tfrac{1}{2}Gh^2\right) + \frac{\partial}{\partial y}(huv) = 0, \tag{4.30}$$

$$\frac{\partial (hv)}{\partial t} + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(hv^2 + \tfrac{1}{2}Gh^2\right) = 0, \tag{4.31}$$

where x and y denote components of the coordinate vector within the orthonormal frame and u and v are the corresponding velocities.

4.4.3 Discretization of the Metric and Coriolis Terms

To discretize the metric and Coriolis terms, we make use of Gaussian quadrature in 2D to evaluate the integral. For a second-order scheme, this requires the evaluation of these source terms at one point within each element – namely, at the element center point (see Figure 4.2). For a fourth-order scheme, we can obtain fourth order accuracy by evaluating the source terms at four points within each element – in particular, for an element defined on the region $[\alpha_1, \alpha_2] \times [\beta_1, \beta_2]$, at

$$(\alpha, \beta) = \left(\frac{\alpha_1 + \alpha_2}{2} \pm \frac{\alpha_2 - \alpha_1}{2\sqrt{3}}, \frac{\beta_1 + \beta_2}{2} \pm \frac{\beta_2 - \beta_1}{2\sqrt{3}} \right). \tag{4.32}$$

For Gaussian quadrature up to fourth order, all points contribute equally to the integral.

4.4.4 Discretization of the Topography Term

Before making a choice of discretization for the underlying topography (denoted by $z(\alpha, \beta, n_p)$), we must consider the important equilibrium case of stationary flow. In this case we have $\mathbf{u}^i = 0$ with constant total height $H = h + z$ everywhere, which physically is an equilibrium solution that should be maintained indefinitely. Not all discretizations will automatically retain this property, however those that do are referred to as *well-balanced* schemes (or, alternatively, schemes that preserve the C-property, *e.g.* see *Noelle et al.* (2006) or *Chen and Xiao* (2008)). It is a well-known fact that discretizations that are not well-balanced may introduce spurious oscillations into the flow that are especially evident for states near this equilibrium solution. In order to develop our topography discretization, we begin with the topography source in the form,

$$\mathbf{\Psi}_B = -Gh\nabla z, \tag{4.33}$$

and observe that it can be rewritten as

$$\mathbf{\Psi}_B = -Gh\nabla H + \nabla \left(\tfrac{1}{2} G h^2 \right). \tag{4.34}$$

We note that for $\mathbf{u} = 0$ the flux-form momentum equations from (4.6)-(4.14) take the form of a balance law,

$$\langle \text{Flux Terms} \rangle = \underbrace{-G \int\limits_{\mathcal{Z}} h \nabla H dV}_{(a)} + \underbrace{\int\limits_{\mathcal{Z}} \nabla \left(\tfrac{1}{2} G h^2 \right) dV}_{(b)} . \tag{4.35}$$

The benefit of writing the topography source in the form (4.34) is now revealed; namely, if we can guarantee that $\nabla H = 0$ when H is constant and calculate (4.35.b) in a manner identical to the calculation of the flux, then our discretization will satisfy the well-balanced property.

We observe that this choice of discretization is, in general, non-zero even when no bottom topography is present ($z = 0$). This observation follows from the fact that in the discrete case, the left-hand-side flux term is non-zero when calculated on the manifold (even over a constant field) since we cannot boast the symmetry properties present in purely Cartesian coordinates. Without this additional correction, the model would be unable to maintain a constant height field h with zero flow velocity $\mathbf{u} = 0$, since in discrete form the contour integral would not be exactly zero.

Our choice of discretization of the first term of (4.34) for the third- and fourth-order schemes is based on using only the evaluated state vector at the Gauss points given in Figure 4.2. This choice has the twofold benefit of only requiring the user-specified topography to be given at each Gauss point, and further enforces consistency of the discretization with the second term of (4.34), which is obtained from the Gauss points along each edge.

Our scheme follows an approach similar to that of *Noelle et al.* (2006). To begin, we write the first term of (4.34) in the form

$$-Gh\nabla H = -G \begin{bmatrix} hg^{11}\frac{\partial H}{\partial \alpha} \\ \\ hg^{21}\frac{\partial H}{\partial \alpha} \end{bmatrix} - G \begin{bmatrix} hg^{12}\frac{\partial H}{\partial \beta} \\ \\ hg^{22}\frac{\partial H}{\partial \beta} \end{bmatrix}. \qquad (4.36)$$

Integrating this expression term-by-term gives

$$\int_{\mathcal{Z}} -Gh\nabla H dV = -G \begin{bmatrix} \int_{\mathcal{Z}} p^{11}\frac{\partial H}{\partial \alpha} d\alpha d\beta \\ \int_{\mathcal{Z}} p^{21}\frac{\partial H}{\partial \alpha} d\alpha d\beta \end{bmatrix} - G \begin{bmatrix} \int_{\mathcal{Z}} p^{12}\frac{\partial H}{\partial \beta} d\alpha d\beta \\ \int_{\mathcal{Z}} p^{22}\frac{\partial H}{\partial \beta} d\alpha d\beta \end{bmatrix}, \qquad (4.37)$$

where we have defined

$$p^{ij} = hg^{ij}J = \frac{h}{\delta} \begin{bmatrix} 1+Y^2 & XY \\ \\ XY & 1+X^2 \end{bmatrix}. \qquad (4.38)$$

Without loss of generality, we consider an approach for discretizing an expansion of the form

$$\int_{\mathcal{Z}} p\frac{\partial H}{\partial \alpha} d\alpha d\beta, \qquad (4.39)$$

observing that this form closely matches that of each of the topography terms in (4.37). We will make use of the Gauss points given by $(\alpha_i \pm \Delta/2, \beta_j \pm \gamma)$ and $(\alpha_i \pm \gamma, \beta_j \pm \gamma)$ so that the usual Gaussian quadrature can be performed in the β direction. We now require a $O(\Delta^4)$ discretization of the integral in the α direction, which we now construct from two $O(\Delta^2)$ discretizations. Consider centered discretizations of (4.39) given by

$$\int p\frac{\partial H}{\partial \alpha} d\alpha \approx \left(\frac{p_{-\Delta/2} + p_{\Delta/2}}{2}\right)\left(H_{\Delta/2} - H_{-\Delta/2}\right) + O(\Delta^2), \qquad (4.40)$$

and

$$\begin{aligned}\int p\frac{\partial H}{\partial\alpha}d\alpha &\approx \left(\frac{p_{-\Delta/2}+p_{-\gamma}}{2}\right)\left(H_{-\gamma}-H_{-\Delta/2}\right)\\ &+\left(\frac{p_{\gamma}+p_{-\gamma}}{2}\right)\left(H_{\gamma}-H_{-\gamma}\right)\\ &+\left(\frac{p_{\Delta/2}+p_{\gamma}}{2}\right)\left(H_{\Delta/2}-H_{\gamma}\right)+O(\Delta^2).\end{aligned} \tag{4.41}$$

Here the subscripts $\pm\Delta/2$ and $\pm\gamma$ denote evaluation of these quantities at $(x_i \pm \Delta/2)$ and $(x_i \pm \gamma)$. Now, for any γ that is a linear function of Δ (except, of course, $\gamma = \pm\Delta/2$) there exists coefficients A_1 and A_2 so that

$$\int_X p\frac{\partial H}{\partial X}dX = A_1[eq.(4.40)] + A_2[eq.(4.41)] + O(\Delta^4), \tag{4.42}$$

and, in particular, for $\gamma = \Delta/(2\sqrt{3})$ we obtain a fourth-order accurate approximation with

$$A_1 = \frac{\sqrt{3}-3}{\sqrt{3}+3} \quad \text{and} \quad A_2 = \frac{6}{\sqrt{3}+3}. \tag{4.43}$$

4.4.5 The Sub-Grid-Scale Reconstruction

All of the numerical approaches discussed in this chapter make use of a high-order polynomial sub-grid-scale reconstruction to evaluate the underlying state variables. The order of the sub-grid-scale reconstruction then determines the underlying order of the scheme, when combined with a sufficiently high order integration scheme for the boundary and the interior.

For a general order-n finite-volume method on the cubed sphere, we define a sub-grid-scale reconstruction (an approximation to the exact field q, here denoted by $\tilde{q}$) of the form

$$\tilde{q}(\alpha,\beta) = \overline{q} + \sum_{s+t<n}\left[(\alpha-\alpha_0)^s(\beta-\beta_0)^t - a^{(s,t)}\right]\left(\frac{1}{s!t!}\frac{\delta^n q}{\delta\alpha^s\delta\beta^t}\right)_0, \tag{4.44}$$

where (α_0, β_0) is the element centroid and the $a^{(s,t)}$ are a family of constants defined so that

$$\int\limits_{\mathcal{Z}} \left[(\alpha - \alpha_0)^s (\beta - \beta_0)^t - a^{(s,t)}\right] dV = 0. \tag{4.45}$$

Here we have made use of s and t to denote terms containing s derivatives in α and t derivatives in β. Further, the discrete approximations to the derivatives of q are denoted with δ instead of ∂ so as to distinguish them from the exact operators; the subscript 0 further denotes evaluation at the element centroid (α_0, β_0). The reconstruction (4.44)-(4.45) is chosen so as to preserve the element average, *i.e.*

$$\frac{1}{|\mathcal{Z}|} \int\limits_{\mathcal{Z}} q \; dV = \overline{q}. \tag{4.46}$$

In order to achieve order-n accuracy with the reconstruction (4.44), we also require that

$$\left(\frac{1}{s!t!} \frac{\delta^n q}{\delta \alpha^s \delta \beta^t} \right)_0 = \frac{1}{s!t!} \frac{\partial^n q}{\partial \alpha^s \partial \beta^t} + O(\Delta^{n-s-t}). \tag{4.47}$$

For instance, this restriction requires that for a third-order scheme all first derivative terms must be at least $\sim O(\Delta^2)$, and all second-derivative terms be at least $\sim O(\Delta)$.

Notes

1. In general, an order-n sub-grid-scale reconstruction will yield an order-n scheme when combined with a flux integral (taken around the boundary of each element) of order n and an interior integration procedure (for source terms) of order $n-1$. Thus, one could potentially sacrifice an order of accuracy for evaluating the source terms and still obtain an order-n scheme. In fact, experiments involving the methods described in this chapter have shown that a reduction in the order of the metric and Coriolis source terms has little discernable impact on the results of each simulation since the error in these schemes is primarily dominated

by the flux terms.

2. Reconstruction-based schemes have the beneficial property of being, in general, Riemann-solver agnostic. That is, in order to solve for the edge fluxes we can make use of any approximate Riemann solver that takes as input a left state $\mathbf{q}_L$ and a right state $\mathbf{q}_R$ as input.

4.4.6 Treatment of Panel Edges

The edges of each panel of the cubed-sphere require special consideration, since they represent discontinuities in the coordinate system. There are two instances where data must be communicated across panel edges: first, we require this information to calculate the reconstructed derivatives in elements near panel edges, and second, we require this information when computing fluxes across these interfaces. For the latter case, fluxes computed across panel interfaces are handled no differently than interfaces within panels, since the orthonormalization procedure rewrites the velocities in terms of a local coordinate system that is valid regardless of the panel. Note that this technique differs from that of *Rossmanith* (2006), where fluxes are instead calculated on each panel independently, with the obvious shortfall being that there is no guarantee of mass conservation along panel edges.

In order to provide boundary information to the reconstruction calculation, our choice of boundary conditions can have significant influence on the numerical method. We consider two possible approaches for handling remapping of information across panel boundaries:

- Under the first approach, we interpolate a 1D polynomial parallel to the panel boundary (either through cell centers of elements on the source panel, or using element averages). This approach is known as *cascade interpolation*, made possible by a previously identified feature of the equiangular coordinate system;

namely that grid lines parallel to panel edges are shared by both coordinate systems (see Figure 4.1). Unfortunately, this approach is at most second-order accurate since any 1D interpolated polynomial does not take into account the variation perpendicular to the edge.

- Under the second approach, we compute one-sided approximations of the derivatives within each boundary element. Then, by applying four-point Gaussian quadrature within each destination element we approximate the element average in the neighbouring panel's ghost elements. A detailed description of this method is given in Appendix E.

In CFD lore, it is generally believed that an n-order accurate numerical method with $(n-1)$-order boundary conditions will still be n-order accurate. However, the strong coupling of the interior cubed-sphere panel boundaries to the numerical solution does not give us this grace. We have witnessed in several tests that a third-order interior scheme, when combined with a second-order boundary scheme (in this case, a cubic cascade approach), will lead to purely second-order performance in convergence tests. Based on this conclusion, we will make use of the second approach for all of our third- and fourth-order schemes.

As an aside, we note that we must also transform the vector velocities between the panel-specific coordinate systems when performing remapping. This adjustment is achieved by simply applying a transformation matrix in the interpolation step (hence, for each ghost cell the transformation matrix must be applied once for a second-order scheme and four times for a third- or fourth-order scheme).

4.4.7 Extensions to Arbitrary Order-of-Accuracy

The framework for finite-volume methods presented in this section can be easily extended to arbitrarily large order-of-accuracy, however for every increase in the order-of-accuracy, the finite-volume approach requires a corresponding increase in the size of the stencil. Besides ensuring that the reconstruction is sufficiently accurate, one must also ensure that the flux calculation and source terms are handled appropriately.

In order to ensure that flux calculations are at least n^{th} order accurate, we must solve Riemann problems at $\lceil n/2 \rceil$ points per edge. Similarly, source terms must be evaluated at $\lceil n/2 \rceil^2$ interior points.

4.5 Numerical Approaches

Using the general framework presented in section 4.4, we construct two schemes for the shallow-water equations on the sphere. The first scheme is a dimension-split piecewise-parabolic method (FV3s) that is formally second-order accurate, but leads to a method which behaves with third-order accuracy on smooth problems. The second scheme we consider is a fourth-order piecewise-cubic method (FV4).

4.5.1 A Dimension-Split Piecewise-Parabolic Scheme (FV3s)

Dimension-split techniques which do not make use of cross-derivatives are formally limited to be no more than second-order accurate. In many cases however, one finds that the error introduced due to neglecting the cross-derivatives is approximately negated when using a symmetric approach. Since dimension-split schemes can be efficiently parallelized, we consider here one such dimension-split approach that combines a formally second-order reconstruction (because cross-derivatives are suppressed) with higher-order approaches for the flux, source terms and panel boundaries.

This method differs from the well-known piecewise-parabolic method of *Colella and Woodward* (1984) since it uses an inherently discontinuous reconstruction, does not explicitly limit the reconstructed derivatives and uses two Gaussian quadrature points along each edge in order to achieve high-order accuracy in the flux estimates.

In our approach, the discrete derivatives are calculated in equiangular coordinates

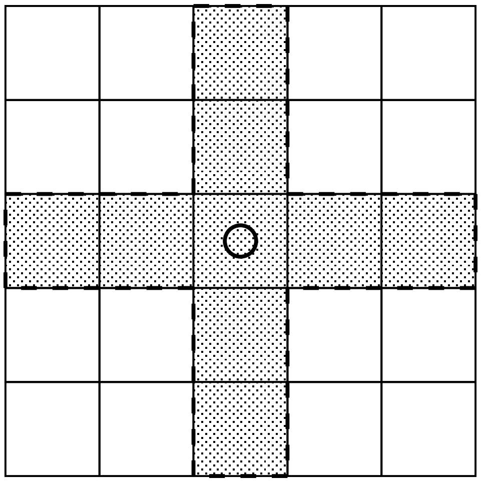


Figure 4.3: The stencil for the dimension-split FV3s scheme.

via

$$\left(\frac{\delta q}{\delta \alpha}\right)_{i,j} = \frac{-\overline{q}_{i+2,j} + 8\overline{q}_{i+1,j} - 8\overline{q}_{i-1,j} + \overline{q}_{i-2,j}}{12\Delta}, \tag{4.48}$$

$$\left(\frac{\delta q}{\delta \beta}\right)_{i,j} = \frac{-\overline{q}_{i,j+2} + 8\overline{q}_{i,j+1} - 8\overline{q}_{i,j-1} + \overline{q}_{i,j-2}}{12\Delta}, \tag{4.49}$$

$$\left(\frac{\delta^2 q}{\delta \alpha^2}\right)_{i,j} = \frac{-\overline{q}_{i+2,j} + 16\overline{q}_{i+1,j} - 30\overline{q}_{i,j} + 16\overline{q}_{i-1,j} - \overline{q}_{i-2,j}}{12\Delta^2}, \tag{4.50}$$

$$\left(\frac{\delta^2 q}{\delta \beta^2}\right)_{i,j} = \frac{-\overline{q}_{i,j+2} + 16\overline{q}_{i,j+1} - 30\overline{q}_{i,j} + 16\overline{q}_{i,j-1} - \overline{q}_{i,j-2}}{12\Delta^2}. \tag{4.51}$$

Using this form of the reconstructed derivatives all derivative terms are formally $O(\Delta^2)$, since element averages only represent a $O(\Delta^2)$ approximation to the centroid value. Nonetheless, this scheme would be formally third-order accurate if the cross-derivative $\delta^2 q/\delta\alpha\delta\beta$ was included in the reconstruction. The stencil used by this scheme is depicted in Figure 4.3.

In order to preserve stability and high-order accuracy in time, we also make use of the total-variation-diminishing (TVD) third-order Runge-Kutta (RK3) timestepping scheme (see, for example, *Gottlieb et al.* (2001)). For a given semi-discretization with right-hand-side $\mathbf{L}(\mathbf{q})$, this scheme can be written as

$$\begin{aligned}
\mathbf{q}^{(1)} &= \mathbf{q}^n + \Delta t \mathbf{L}(\mathbf{q}^n), \\
\mathbf{q}^{(2)} &= \tfrac{3}{4}\mathbf{q}^n + \tfrac{1}{4}\mathbf{q}^{(1)} + \tfrac{1}{4}\Delta t \mathbf{L}(\mathbf{q}^{(1)}), \\
\mathbf{q}^{n+1} &= \tfrac{1}{3}\mathbf{q}^n + \tfrac{2}{3}\mathbf{q}^{(2)} + \tfrac{2}{3}\Delta t \mathbf{L}(\mathbf{q}^{(2)}).
\end{aligned} \tag{4.52}$$

We make use of the CFL number in the form

$$CFL = \frac{\Delta t}{4|\mathcal{Z}|} \oint_{\partial \mathcal{Z}} |\lambda|_{max} ds, \tag{4.53}$$

where the contour integral is taken around the outside of the element $\mathcal{Z}$ and $|\lambda|_{max}$ denotes the maximum value of the absolute gravity wave speed. Under this definition, the maximum CFL number for the FV3s scheme can be empirically determined to be ~ 1.05. The maximum CFL number is strongly dependent on the choice of timestepping scheme; if we discretize FV3s with a RK4 timestep scheme, the CFL limit for this scheme increases to ~ 1.30.

4.5.2 The Piecewise-Cubic (FV4) Scheme

Our fourth-order finite-volume scheme makes use of a piecewise-cubic sub-grid-scale reconstruction. We first make use of a convolution operator to obtain point values at the centerpoint of each element. Using these point values, we can then apply a set of standard finite difference operators in order to obtain approximations to the derivatives at the center-points of each element.

Note that the scheme described in this section requires three ghost-elements, which will lead to an increase in parallel communication over the FV3s scheme, for instance. A more efficient fourth-order method (in terms of parallel computational) can be formulated using a 5×5 stencil, but such a decrease in stencil size will also lead to a reduction in computational accuracy. By comparison, the resulting method would only require two ghost cells.

Following the approach of *Barad and Colella* (2005) element-averages of a scalar field $\overline{q}$ and the corresponding point-values at element centers q_0 can be interchanged via the formula

$$\overline{q} = q_0 + \frac{\Delta^4}{12|\mathcal{Z}|}\left[\left(\frac{\delta q}{\delta\alpha}\right)_0\left(\frac{\partial J}{\partial\alpha}\right)_0 + \left(\frac{\delta q}{\delta\beta}\right)_0\left(\frac{\partial J}{\partial\beta}\right)_0\right] + \frac{\Delta^2}{24}\left[\left(\frac{\delta^2 q}{\delta\alpha^2}\right)_0 + \left(\frac{\delta^2 q}{\delta\beta^2}\right)_0\right] + O(\Delta^4). \tag{4.54}$$

In order to obtain $O(\Delta^4)$ accuracy via this formula, the first and second derivatives in this expression must be approximated to $O(\Delta^2)$, and so can be obtained from

$$\begin{aligned}\left(\frac{\delta q}{\delta\alpha}\right)_0 &= \frac{\overline{q}_{i+1,j} - \overline{q}_{i-1,j}}{2\Delta} + O(\Delta^2),\\ \left(\frac{\delta^2 q}{\delta\alpha^2}\right)_0 &= \frac{\overline{q}_{i+1,j} - 2\overline{q}_{i,j} + \overline{q}_{i-1,j}}{\Delta^2} + O(\Delta^2),\end{aligned}$$

and similarly in the β-direction. The derivatives of the metric terms that appear in (4.54) can be computed analytically at the element center-points from (4.5).

Using the point-values obtained from (4.54) we then apply a standard set of 5-point finite-difference operators in order to approximate derivatives at the center-point. With this technique we obtain sufficiently accurate approximations to the derivatives to support a fourth-order scheme. The stencil obtained from this approach is depicted in Figure 4.4.

In order to preserve stability and high-order accuracy in time, we combine our spatial discretization with a fourth-order Runge-Kutta (RK4) timestepping scheme

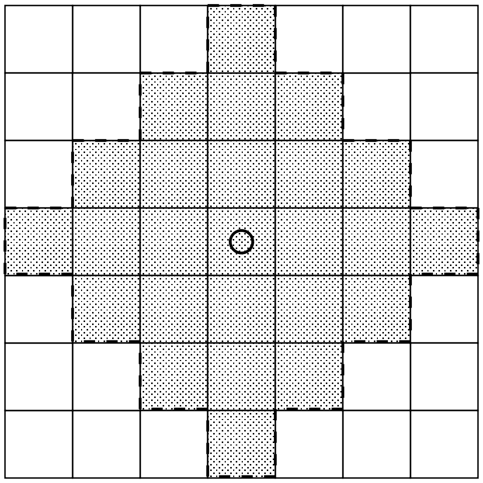


Figure 4.4: The reconstruction stencil for the FV4 scheme.

of the form

$$\begin{aligned}
\mathbf{q}^{(1)} &= \mathbf{q}^n + \tfrac{1}{2}\Delta t\mathbf{L}(\mathbf{q}^n), \\
\mathbf{q}^{(2)} &= \mathbf{q}^n + \tfrac{1}{2}\Delta t\mathbf{L}(\mathbf{q}^{(1)}), \\
\mathbf{q}^{(3)} &= \mathbf{q}^n + \Delta t\mathbf{L}(\mathbf{q}^{(2)}), \\
\mathbf{q}^{n+1} &= -\tfrac{1}{3}\mathbf{q}^n + \tfrac{1}{3}\mathbf{q}^{(1)} + \tfrac{2}{3}\mathbf{q}^{(2)} + \tfrac{1}{3}\mathbf{q}^{(3)}.
\end{aligned} \tag{4.55}$$

Using this choice of timestep scheme, the maximum CFL number for the FV4 scheme can be empirically determined to be ~ 1.30.

4.6 Approximate Riemann Solvers

In each of the methods presented here we make use of an approximate Riemann solver to obtain the local flux across a discontinuous interface. We will compare three approximate Riemann solvers, given in order of increasing complexity as Rusanov, Roe and AUSM^{+}-up.

It is a well-known (see, for example, *van Leer et al.* (1987)) result that the Roe solver is less dissipative than Rusanov's scheme. In general, dissipiation will be proportional to the wave speed of each characteristic flow variable, which is exaggerated in Rusanov's scheme by taking the wave speed of each wave to be equal to the largest wave speed. Roe's scheme, on the other hand, distinguishes all waves and hence provides a significantly tighter bound on the diffusivity. The AUSM$^+$-up scheme does not distinguish all waves, but instead uses asymptotic analysis and a separation of the advective and pressure terms in order to fight excess diffusivity at small Mach numbers. For details on specific Riemann solvers and their properties, we recommend *Toro* (1999).

4.6.1 Rusanov

The Rusanov solution to the Riemann problem (first given in *Rusanov* (1961)) is perhaps the simplest to implement, using a straight flux difference between left and right edge values plus the maximum wave speed across the interface to regulate diffusion. Given left state vector $\mathbf{q}_L = (h_L, u_L, v_L)$ and right state vector $\mathbf{q}_R = (h_R, u_R, v_R)$, the Rusanov numerical flux assumes the form

$$\mathbf{F}^* = \frac{\mathbf{F}(\mathbf{q}_L) + \mathbf{F}(\mathbf{q}_R)}{2} - \frac{1}{2} \left| \lambda \left(\frac{\mathbf{q}_L + \mathbf{q}_R}{2} \right) \right| (\mathbf{q}_R - \mathbf{q}_L), \tag{4.56}$$

where, in the orthonormal frame,

$$\mathbf{F}(\mathbf{q}) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}Gh^2 \\ huv \end{pmatrix} \tag{4.57}$$

is the flux function of the associated continuous equations (see (4.29)-(4.31)) and $|\lambda(\mathbf{q})|$ is the absolute value of the maximum wave speed. For the orthonormal shallow-water equations, one can quickly verify

$$|\lambda(\mathbf{q})| = |u| + \sqrt{Gh}. \tag{4.58}$$

As documented in the literature, the major downfall of the Rusanov scheme is its tendency for strong diffusivity compared with other approximate Riemann solvers. However, its ease of implementation and relative computational efficiency have resulted in its frequent use in numerical models.

4.6.2 Roe

The approximate Riemann solver of *Roe* (1981) is ubiquitous in aerospace applications, but its use has been fairly limited in the atmospheric science community. A description of this method as applied to the shallow-water equations in Cartesian coordinates can be found in *Alcrudo and Garcia-Navarro* (1993), for instance.

As with the Rusanov scheme, we are given left state vector $\mathbf{q}_L = (h_L, u_L, v_L)$ and right state vector $\mathbf{q}_R = (h_R, u_R, v_R)$ and must solve for the flux from the associated Riemann problem. We begin with the flux Jacobian for the orthonormal shallow-water equations, given by

$$\tilde{\mathbf{A}} = \frac{d\mathbf{F}}{d\mathbf{q}} = \begin{pmatrix} 0 & 1 & 0 \\ (Gh - u^2) & 2u & 0 \\ -uv & v & u \end{pmatrix}. \tag{4.59}$$

Following the approach of Roe, we construct a modified system of conservation laws with eigenvalues

$$\tilde{\lambda}^1 = \tilde{u} + \tilde{c}, \quad \tilde{\lambda}^2 = \tilde{u}, \quad \tilde{\lambda}^3 = \tilde{u} - \tilde{c}, \tag{4.60}$$

and corresponding eigenvectors

$$\mathbf{e}^1 = \begin{pmatrix} 1 \\ \tilde{u} + \tilde{c} \\ \tilde{v} \end{pmatrix}, \quad \mathbf{e}^2 = \begin{pmatrix} 0 \\ 0 \\ \tilde{c} \end{pmatrix}, \quad \mathbf{e}^3 = \begin{pmatrix} 1 \\ \tilde{u} - \tilde{c} \\ \tilde{v} \end{pmatrix}. \tag{4.61}$$

Here the eigenvalues and eigenvectors are written in terms of Roe-averaged velocities $\tilde{u}$ and $\tilde{v}$ and the Roe-averaged gravity wave speed $\tilde{c}$, defined by

$$\tilde{u} = \frac{u_R + u_L w}{1 + w}, \quad \tilde{v} = \frac{v_R + v_L w}{1 + w}, \quad \tilde{c} = \sqrt{\frac{G(h_R + h_L)}{2}}, \tag{4.62}$$

where $w = \sqrt{h_L}/\sqrt{h_R}$.

The Roe numerical flux function then takes the form

$$\mathbf{F}^* = \frac{\mathbf{F}_R + \mathbf{F}_L}{2} - \tfrac{1}{2} \sum_{k=1}^{3} \tilde{\alpha}^k |\tilde{\lambda}^k| \tilde{e}^k, \tag{4.63}$$

where $\tilde{\alpha}^k$ are the coefficients obtained by decomposing the difference $\mathbf{q}_R - \mathbf{q}_L$ in terms of the basis of eigenvectors via

$$\mathbf{q}_R - \mathbf{q}_L = \sum_{k=1}^{3} \tilde{\alpha}^k \mathbf{e}^k. \tag{4.64}$$

They can be written in terms of the jumps $\Delta = (\ )_R - (\ )_L$ in the height and momentum field via

$$\begin{aligned} \tilde{\alpha}^1 &= \frac{1}{2\tilde{c}} \left(\Delta(hu) - (\tilde{u} - \tilde{c})\Delta h \right), && (4.65) \\ \tilde{\alpha}^2 &= \frac{1}{\tilde{c}} \left(\Delta(hv) - \tilde{v}\Delta h \right), && (4.66) \\ \tilde{\alpha}^3 &= -\frac{1}{2\tilde{c}} \left(\Delta(hu) - (\tilde{u} + \tilde{c})\Delta h \right). && (4.67) \end{aligned}$$

The Roe numerical flux tends to perform well for flows in the transsonic and supersonic regime but, similar to the Rusanov scheme, is generally diffusive for low Mach number flows (see, for example, *Guillard and Viozat* (1999)).

4.6.3 AUSM$^+$-up

The AUSM$^+$-up approximate Riemann solver of *Liou* (2006) was recently developed with the purpose of improving numerical accuracy in the low-Mach number regime. In particular, the AUSM$^+$-up scheme works by splitting the advective component of the flux from the pressure component. We refer the reader to *Liou* (2006) for the mathematical details of this algorithm, instead giving a short overview of the implementation of this approach for the shallow-water equations.

Given left state vector $\mathbf{q}_L = (h_L, u_L, v_L)$ and right state vector $\mathbf{q}_R = (h_R, u_R, v_R)$, with orthonormal velocity components, we define the *averaged height*,

$$h_{1/2} = \tfrac{1}{2}\left(h_L + h_R\right), \tag{4.68}$$

averaged gravity wave speed,

$$a_{1/2} = \tfrac{1}{2}\left(\sqrt{gh_L} + \sqrt{gh_R}\right), \tag{4.69}$$

perpendicular Mach numbers at the interface,

$$M_L = \frac{u_L}{a_{1/2}}, \quad \text{and} \quad M_R = \frac{u_R}{a_{1/2}}, \tag{4.70}$$

and *mean local Mach number*,

$$\overline{M}^2 = \frac{u_L^2 + u_R^2}{2a_{1/2}^2}. \tag{4.71}$$

The *advective component of the flux* is then defined by

$$\dot{m}_{1/2} = a_{1/2} M_{1/2} \begin{cases} h_L & \text{if } M_{1/2} > 0, \\ h_R & \text{otherwise,} \end{cases} \tag{4.72}$$

for some appropriate choice of the *interface Mach number* $M_{1/2}$. By defining

$$\mathcal{M}^{\pm}_{(2)}(M) = \tfrac{1}{4}(M \pm 1)^2, \quad \mathcal{M}^{\pm}_{(4)}(M) = \begin{cases} \tfrac{1}{2}(M \pm |M|) & \text{if } |M| \geq 1, \\ \mathcal{M}^{\pm}_{(2)}(M)(1 \mp 16\beta \mathcal{M}^{\mp}_{(2)}(M)) & \text{otherwise,} \end{cases} \tag{4.73}$$

we can obtain an expression for $M_{1/2}$ consistent with *Liou* (2006),

$$M_{1/2} = \mathcal{M}^{+}_{(4)}(M_L) + \mathcal{M}^{-}_{(4)}(M_R) - K_p \max(1 - \sigma \overline{M}^2, 0) \frac{G(h_R^2 - h_L^2)}{2 h_{1/2} a_{1/2}^2}. \tag{4.74}$$

To obtain the *pressure-driven component of the flux*, we make use of the definition

$$\mathcal{P}^{\pm}_{(5)}(M) = \begin{cases} \tfrac{1}{2}(1 \pm \operatorname{sign}(M)) & \text{if } |M| \geq 1, \\ \mathcal{M}^{\pm}_{(2)}(M) \left[(\pm 2 - M) \mp 16\alpha M \mathcal{M}^{\mp}_{(2)}(M) \right] & \text{otherwise.} \end{cases} \tag{4.75}$$

The interface pressure-driven flux is then given by

$$p_{1/2} = \mathcal{P}^{+}_{(5)}(M_L) p_L + \mathcal{P}^{-}_{(5)}(M_R) p_R - K_u \mathcal{P}^{+}_{(5)}(M_L) \mathcal{P}^{-}_{(5)}(M_R)(h_L + h_R) a_{1/2}(u_R - u_L). \tag{4.76}$$

Combining (4.72)-(4.74) and (4.76), we obtain that the total numerical flux across the interface is then given by

$$\mathbf{F}^* = \left[\dot{m}_{1/2} \begin{cases} \mathbf{\Psi}_L & \text{if } \dot{m}_{1/2} > 0, \\ \mathbf{\Psi}_R & \text{otherwise} \end{cases} \right] + \mathbf{p}_{1/2}, \tag{4.77}$$

with

$$\mathbf{\Psi}_L = \begin{pmatrix} 1 \\ u_L \\ v_L \end{pmatrix}, \quad \mathbf{\Psi}_R = \begin{pmatrix} 1 \\ u_R \\ v_R \end{pmatrix}, \quad \mathbf{p}_{1/2} = \begin{pmatrix} 0 \\ p_{1/2} \\ 0 \end{pmatrix}. \tag{4.78}$$

Several free parameters are available in this scheme. For simplicity, we follow *Liou* (2006) by choosing

$$\alpha = \tfrac{3}{16}, \quad \beta = \tfrac{1}{8}, \quad K_p = \tfrac{1}{4}, \quad \sigma = 1. \tag{4.79}$$

The constant K_u, which governs velocity diffusivity, does not seem to play a major role in these results and is chosen to be zero.

The AUSM$^+$-up flux has been constructed with the goal of improving convergence and accuracy in the low-Mach number limit ($M \to 0$), and so we anticipate it to be less diffusive than Rusanov or Roe in this regime.

4.7 Numerical Results

For all calculations we use a normalized length scale in terms of Earth radii and use time given in days. For the Earth the physical parameters under these scalings are given by

$$G = 11489.57 \text{ Earth radii/day}^2, \quad \text{and} \quad \Omega = 6.300288 \text{ day}^{-1}. \tag{4.80}$$

Unless stated otherwise we make use of a CFL number of 1.0 for all simulations.

Error measures are calculated in the height field via the usual global error norms,

$$\begin{aligned} \ell_1(h) &= \frac{I\left[|h - h_T|\right]}{I\left[|h_T|\right]}, && (4.81) \\ \ell_2(h) &= \sqrt{\frac{I\left[(h - h_T)^2\right]}{I\left[h_T^2\right]}}, && (4.82) \\ \ell_\infty(h) &= \frac{\max|h - h_T|}{\max|h_T|}, && (4.83) \end{aligned}$$

where h_T is the height field at the initial time and I denotes an approximation to the global integral, given by

$$I[x] = \sum_{\text{all cells } k} x_k A_k, \tag{4.84}$$

with A_k denoting the area of element k. For the advection test case we also make use of the relative maximum and minimum,

$$\begin{aligned} \langle \text{Relative Maximum} \rangle &= \frac{\max h - \max h_T}{\max |h_T|}, & (4.85) \\ \langle \text{Relative Minimum} \rangle &= \frac{\min h - \min h_T}{\max |h_T|}. & (4.86) \end{aligned}$$

4.7.1 Advection of a Cosine Bell

The first test case of *Williamson et al.* (1992) simulates the advection of a cosine bell through one rotation around the sphere over a 12-day time period. The prescribed wind field is nondivergent, and so the flux-form continuity equation in orthonormal form,

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0, \tag{4.87}$$

represents an advection equation for the tracer distribution. Here, x and y denote the components of the coordinate vector within the orthonormal frame. The velocity vector is not evolved, and is instead obtained by directly evaluating the velocity field as needed. For this equation the Riemann flux solution reduces to

$$F^* = \begin{cases} h_L u & \text{if } u > 0, \\ h_R u & \text{if } u < 0. \end{cases} \tag{4.88}$$

The initial height field is given by

$$h = \begin{cases} \left(\frac{h_0}{2}\right)\left(1 + \cos\frac{\pi r}{R}\right) & \text{if } r < R, \\ 0 & \text{otherwise,} \end{cases} \tag{4.89}$$

where r is the great circle distance from the center of the height profile. The free parameters are given as $h_0 = 1000$ m $= 1.5696 \times 10^{-4}$ Earth radii and $R = \frac{1}{3}$ Earth radii. The divergence-free velocity field is given in terms of spherical coordinates as

$$u_\lambda = u_0(\cos\theta\cos\alpha + \cos\lambda\sin\theta\sin\alpha), \tag{4.90}$$

$$u_\theta = -u_0\sin\lambda\sin\alpha, \tag{4.91}$$

where $u_0 = (\pi/6)$ Earth radii/day. Here the parameter α denotes the rotation angle transcribed between the physical north pole and the center of the northern panel on the cubed-sphere grid (and should not be confused with the equiangular coordinate α).

This test case is particularly useful at verifying accuracy of the panel boundaries. We give the relative errors after one rotation in Table 4.2 using $\Delta t = 90$ minutes (CFL = 1.0) and $\Delta t = 45$ minutes (CFL = 0.5), and the corresponding time series of these errors (with CFL = 1.0) in Figure 4.5. In all cases we use a resolution of $40 \times 40 \times 6$. A graphical comparison of the reference field and results after one rotation at $\alpha = 45°$ are given in Figure 4.6 and absolute differences in Figure 4.7. These results do not show any obvious noise due to the patch boundaries and the numerical errors we observe are essentially independent of the flow direction. As expected, identical error measures at $\alpha = 0$ and $\alpha = \pi/2$ are observed and so are not repeated. Interestingly, we do observe a significant sensitivity of the method due to choice of the CFL number (see Table 4.2) with halving of the CFL number leading to a decrease of more than half in the error norms for the FV3s scheme.

Table 4.2: Relative errors in the height field h for *Williamson et al.* (1992) Test Case 1 – advection of a cosine bell (at a resolution of $40 \times 40 \times 6$ and after $t = 12$ days) for the FV3s scheme (top) and FV4 scheme (bottom).

FV3s Method

CFL	Direction	L_1 error	L_2 error	L_∞ error	Maximum	Minimum
1.0	$\alpha = 0°$	1.03060(−1)	6.68703(−2)	4.94155(−2)	−3.49156(−2)	−2.76913(−2)
	$\alpha = 45°$	1.02219(−1)	6.42548(−2)	5.01053(−2)	−3.14562(−2)	−2.42758(−2)
0.5	$\alpha = 0°$	4.87889(−2)	2.95893(−2)	2.34241(−2)	−6.66680(−3)	−1.59017(−2)
	$\alpha = 45°$	4.49184(−2)	2.55575(−2)	1.89556(−2)	−5.48348(−3)	−1.00990(−2)

FV4 Method

CFL	Direction	L_1 error	L_2 error	L_∞ error	Maximum	Minimum
1.0	$\alpha = 0°$	4.42623(−2)	2.69819(−2)	2.30115(−2)	1.50290(−4)	−2.25188(−2)
	$\alpha = 45°$	4.21728(−2)	2.36737(−2)	1.86956(−2)	1.51660(−3)	−1.48344(−2)
0.5	$\alpha = 0°$	3.83263(−2)	2.31939(−2)	1.99693(−2)	−8.35479(−5)	−1.92501(−2)
	$\alpha = 45°$	3.50956(−2)	1.96006(−2)	1.41711(−2)	3.51985(−4)	−1.25210(−2)

No attempt to include a monotonicity filter was made in these results, which leads to obvious overshoots and undershoots of the cosine bell profile and oscillations in the tracer field away from the cosine bell profile. These errors are reduced in the FV4 scheme. The error norms presented here are competitive with existing numerical methods of equivalent order-of-accuracy.

4.7.2 Steady-State Geostrophically Balanced Flow

Test case 2 of *Williamson et al.* (1992) simulates a zonally symmetric geostrophically balanced flow. The analytic height field, in terms of latitude θ and longitude λ, is given by

$$h = h_0 - \frac{1}{G}\left(\Omega u_0 + \frac{u_0^2}{2}\right)\left(-\cos\lambda\cos\theta\sin\alpha + \sin\theta\cos\alpha\right)^2, \tag{4.92}$$

with background height h_0 and background velocity u_0 chosen to be

$$h_0 = 4.7057 \times 10^{-4} \text{ Earth radii}, \quad \text{and} \quad u_0 = \frac{\pi}{6} \text{ Earth radii/day}. \tag{4.93}$$

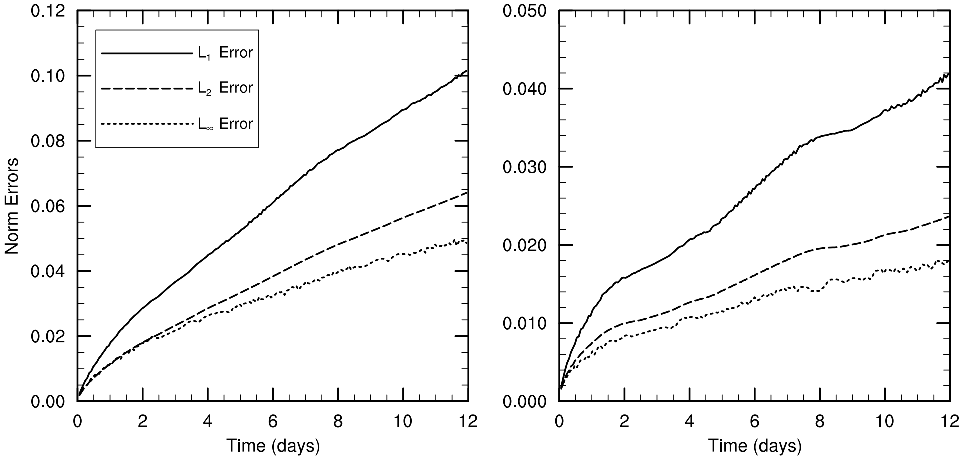


Figure 4.5: Time series of the normalized errors for the cosine bell advection test case with FV3s method (left) and FV4 method (right) in the direction $\alpha = 45°$ for one rotation (12 days) with CFL = 1.0 on a $40 \times 40 \times 6$ grid. Note the difference in the vertical scales of these plots.

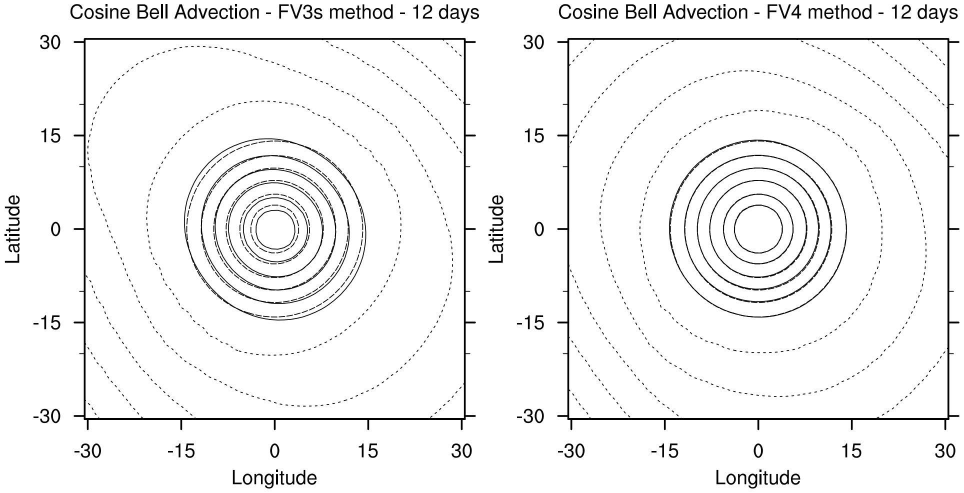


Figure 4.6: Reference height field (long-dashed line) and numerically computed height field (solid line) with FV3s method (left) and FV4 method (right) in the direction $\alpha = 45°$ after one rotation (12 days). Contours are from 0 m to 800 m in intervals of 160 m with the zero contour of the numerically computed solution shown as a dotted line so as to emphasize the numerical oscillations. The direction of motion is to the bottom-right.

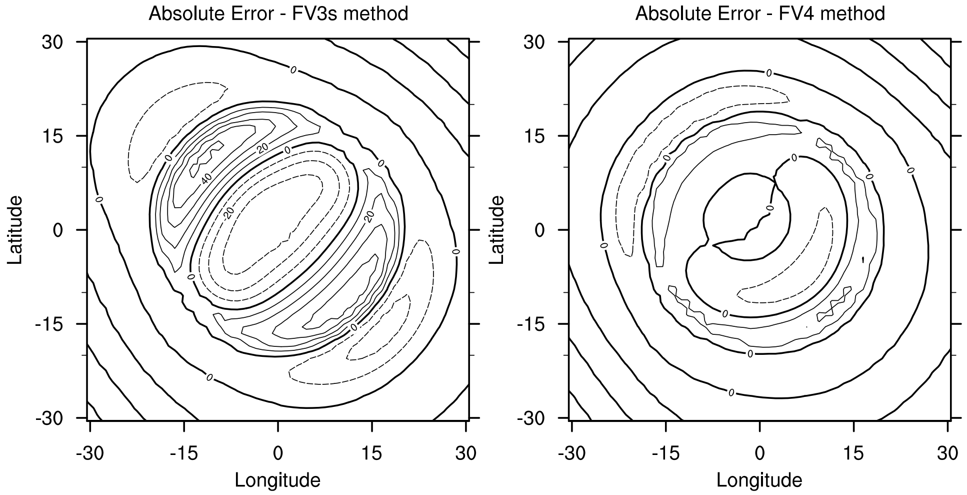


Figure 4.7: Difference between the numerically computed solution and true solution with FV3s method (left) and FV4 method (right) in the direction $\alpha = 45°$ after one rotation (12 days) and at a resolution of $40 \times 40 \times 6$. Contours are in intervals of 10 m with solid lines denoting positive contours and dashed lines denoting negative contours. The zero line is enhanced.

As in section 4.7.1, the parameter α denotes the angle transcribed between the physical north pole and the center of the northern panel. The background velocity field is the same as in (4.90) and (4.91). This test case represents an unstable equilibrium solution to the shallow-water equations, and so is not preserved in the long-term in most atmospheric models. However, it is useful to use this test case to study the convergence properties of a given numerical method.

We use high-order Gaussian quadrature to initialize the height and momentum fields in the numerical model and run the model for five days. The timestep at $40 \times 40 \times 6$ resolution for this case is $\Delta t = 16.5$ minutes. The results of the convergence study for the three schemes with all approximate Riemann solvers is given in Tables 4.3 and 4.4 using $\alpha = 45°$. We see third-order convergence for the dimension-split FV3s scheme and super-convergence above order 4 for the FV4 scheme. When looking at the approximate Riemann solvers, we see significantly different results for the FV3s scheme and the FV4 scheme. For the FV3s scheme the error norms do not differ substantially, and we actually observe the Roe solver performing slightly worse than the Rusanov solver. The FV4 scheme instead shows a significant improvement in error norms when using the Roe or AUSM$^+$-up flux over the Rusanov flux. The high order-of-accuracy for the Rusanov flux is attributed to the improvement in continuity of the reconstruction, so we do not expect the error norms from the Rusanov solver to be less than those of the Roe or AUSM$^+$-up flux. The actual error norms presented here are competitive with existing methods (see *Chen and Xiao* (2008), *Rossmanith* (2006) and *Tomita et al.* (2001)).

Error plots are given in Figures 4.8 and 4.9. The clear diffusivity of the Rusanov solver is apparent here, especially in the FV4 scheme where Rusanov demonstrates approximately five times worse errors than the other fluxes. In this case errors at the panel corners appear to be greatly enhanced. We see very little difference between the error distribution for the Roe and AUSM$^+$-up solvers.

4.7.3 Steady-State Geostrophically Balanced Flow with Compact Support

Test case 3 of *Williamson et al.* (1992) again simulates a geostrophically balanced flow, but this time with a height field that has compact support. The analytic velocity field is given in rotated latitude-longitude coordinates (θ', λ') (with rotation angle α) by

$$u'_\lambda = u_0 b(x) b(x_e - x) \exp(4/x_e), \quad \text{and} \quad u'_\theta = 0, \tag{4.94}$$

where

$$b(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ \exp(-1/x) & \text{if } 0 < x. \end{cases} \tag{4.95}$$

and

$$x = x_e \frac{(\theta' - \theta_b)}{(\theta_e - \theta_b)}. \tag{4.96}$$

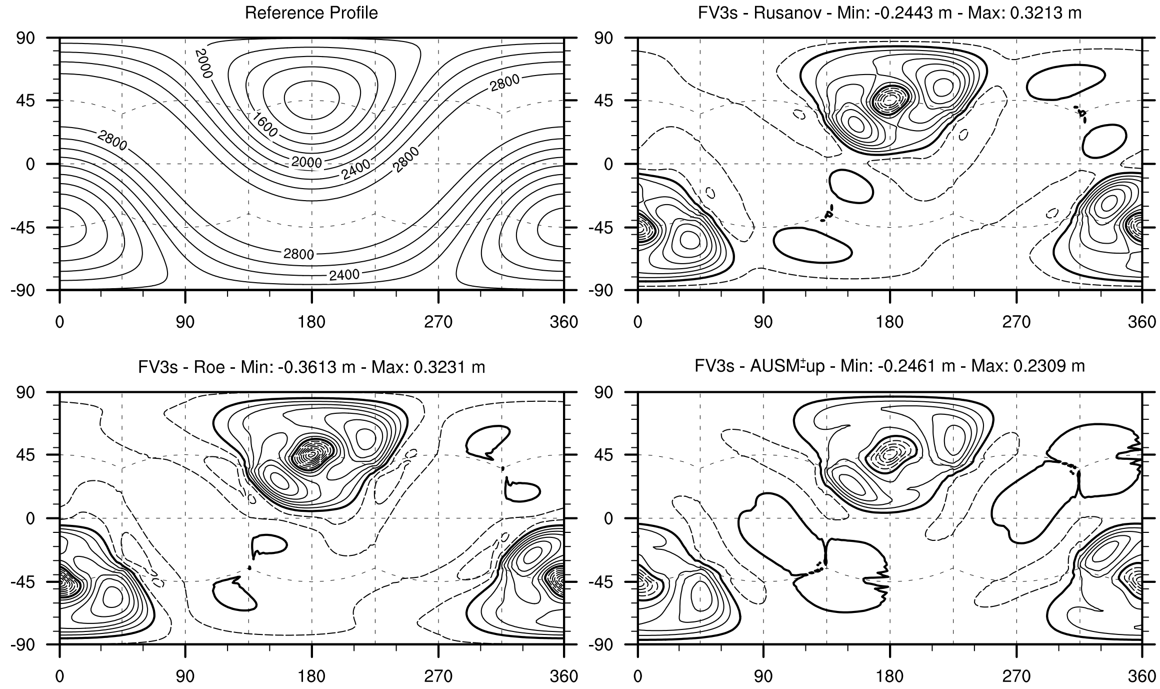


Figure 4.8: Background height field (top-left, in m) and absolute errors associated with the FV3s scheme on a 40 × 40 × 6 grid with Rusanov (top-right), Roe (bottom-left) and AUSM$^+$-up (bottom-right) solvers for *Williamson et al.* (1992) test case 2 with $\alpha = 45°$. Contour lines are in units of 5×10^{-2} m, with solid lines corresponding to positive values and long dashed lines corresponding to negative values. The thick line corresponds to zero error. The short dashed lines show the location of the underlying cubed-sphere grid.

Table 4.3: Relative errors in the height field h for *Williamson et al.* (1992) Test Case 2 – Geostrophically balanced flow (at $t = 5$ days with $\alpha = 45°$) for the FV3s scheme with Rusanov, Roe and AUSM$^+$-up Riemann solvers. The computed order of accuracy is obtained from a least squares fit through the data.

Rusanov Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	1.86793(−4)	2.65483(−4)	8.52171(−4)
40	2.42357(−5)	3.42298(−5)	1.06972(−4)
80	3.05468(−6)	4.31159(−6)	1.36274(−5)
160	3.82801(−7)	5.40185(−7)	1.83789(−6)
Order	2.978	2.981	2.954

Roe Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	1.97019(−4)	2.78152(−4)	8.28123(−4)
40	2.56497(−5)	3.65241(−5)	1.20533(−4)
80	3.23454(−6)	4.61740(−6)	1.82294(−5)
160	4.05309(−7)	5.79033(−7)	2.49317(−6)
Order	2.976	2.971	2.785

AUSM$^+$-up Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	1.29301(−4)	1.84921(−4)	5.84845(−4)
40	1.68287(−5)	2.41594(−5)	8.20936(−5)
80	2.12139(−6)	3.05250(−6)	1.22420(−5)
160	2.65731(−7)	3.82725(−7)	1.67872(−6)
Order	2.977	2.973	2.808

Table 4.4: As Table 4.3 except with the FV4 scheme.

Rusanov Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	1.23147(−5)	1.83684(−5)	5.28083(−5)
40	3.92605(−7)	6.14237(−7)	3.30766(−6)
80	1.42768(−8)	2.19348(−8)	1.94550(−7)
160	6.44784(−10)	9.23157(−10)	1.09139(−8)
Order	4.744	4.765	4.081

Roe Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	3.33670(−6)	4.71855(−6)	1.33113(−5)
40	1.56059(−7)	2.14543(−7)	5.63099(−7)
80	8.67290(−9)	1.17659(−8)	2.79641(−8)
160	5.19851(−10)	7.03778(−10)	1.62477(−9)
Order	4.211	4.232	4.333

AUSM$^+$-up Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	3.26183(−6)	4.66310(−6)	1.19600(−5)
40	1.54530(−7)	2.14661(−7)	5.14470(−7)
80	8.65658(−9)	1.18352(−8)	2.70597(−8)
160	5.19867(−10)	7.06975(−10)	1.64234(−9)
Order	4.200	4.224	4.274

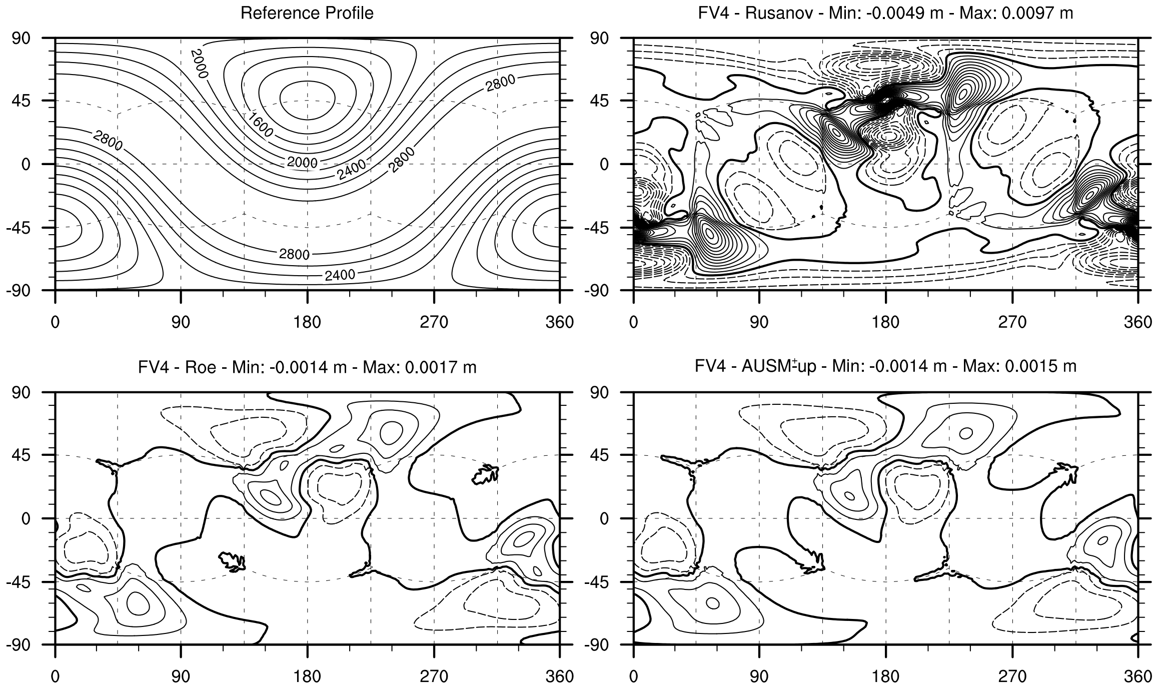


Figure 4.9: As Figure 4.8 except using the FV4 scheme. Contour lines are in units of $5 \times 10^{-4} m$.

The details of the rotated coordinate system are described in *Williamson et al.* (1992). The analytic height field is given by

$$h = h_0 - \frac{1}{G} \int_{-\pi/2}^{\theta'} \left(2\Omega \sin\tau + u'_\lambda(\tau)\tan\tau\right) u'_\lambda(\tau) d\tau, \tag{4.97}$$

which must be integrated numerically at each point where h is desired. The background height and velocity fields are again chosen to be

$$h_0 = 4.7057 \times 10^{-4} \text{ Earth radii}, \quad \text{and} \quad u_0 = \frac{\pi}{6} \text{ Earth radii/day}. \tag{4.98}$$

Further, the compact height field is determined by the parameters

$$\theta_b = -\frac{\pi}{6}, \quad \theta_e = \frac{\pi}{2}, \quad \text{and} \quad x_e = 0.3. \tag{4.99}$$

This test case again represents an unstable equilibrium solution to the shallow-water equations, and so is not preserved in the long-term in most atmospheric models. We again use high-order Gaussian quadrature to initialize the height and momentum fields in the numerical model and run the model for five days. In all cases we make use of a rotation angle of 60°. The timestep at $40 \times 40 \times 6$ resolution for this case is $\Delta t = 17.5$ minutes.

The results of the convergence study for the two schemes with all Riemann solvers is given in Tables 4.5 and 4.6. Again we see similar convergence rates to that of test case 2, except noting that we see a significant gain in accuracy (a 3× improvement in the L_1 norm) when going from the Rusanov Riemann solver to either the Roe or AUSM$^+$-up Riemann solver. For the FV3s scheme we observe slightly better performance from the Roe solver over AUSM$^+$-up, and the opposite for the FV4 scheme.

Error plots are given in Figures 4.10 and 4.11. The error plots from the Roe and AUSM$^+$-up Riemann solvers are very similar, as expected given their similar error norms. However, we see very strong error due to diffusivity in the results from the Rusanov scheme of almost an order of magnitude more than the other two approaches.

4.7.4 Zonal Flow over an Isolated Mountain

Test case 5 in *Williamson et al.* (1992) considers flow with a topographically driven source term. The wind and height fields are defined as in section 4.7.2, except with $\alpha = 0$, $h_0 = 5960$ m and $u_0 = 20$ m/s. A conical mountain is introduced into the flow, given by

$$z = z_0(1 - r/R), \tag{4.100}$$

with $z_0 = 2000$ m, $R = \pi/9$ and $r^2 = \min\left[R^2, (\lambda - \lambda_c)^2 + (\theta - \theta_c)^2\right]$. The center of the mountain is taken as $\lambda_c = 3\pi/2$ and $\theta_c = \pi/6$.

We plot the height field at day 5, 10 and 15 in Figure 4.12 on a $40 \times 40 \times 6$ grid

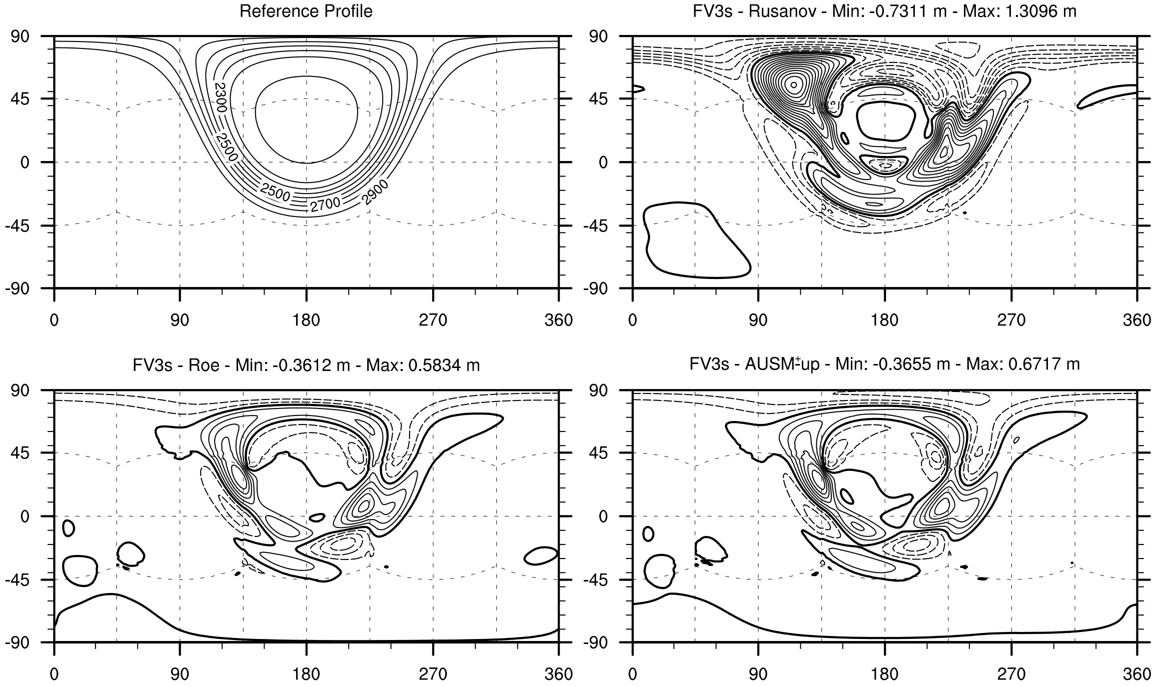


Figure 4.10: Background height field (top-left, in m) and absolute errors associated with the FV3s scheme on a 40 × 40 × 6 grid with Rusanov (top-right), Roe (bottom-left) and AUSM$^+$-up (bottom-right) Riemann solvers for *Williamson et al.* (1992) test case 3 with $\alpha = 60°$. Contour lines are in units of 10^{-1} m, with solid lines corresponding to positive values and dashed lines corresponding to negative values. The thick line corresponds to zero error.

Table 4.5: Relative errors in the height field h for *Williamson et al.* (1992) Test Case 3 – Geostrophically balanced flow with compact support (at $t = 5$ days with $\alpha = 60°$) for the FV3s scheme with Rusanov, Roe and AUSM$^+$-up Riemann solvers. The computed order of accuracy is obtained from a least squares fit through the data.

Rusanov Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	5.29489(−4)	1.04318(−3)	6.98472(−3)
40	4.14616(−5)	7.62192(−5)	4.36449(−4)
80	4.90936(−6)	8.83443(−6)	4.42438(−5)
160	6.24068(−7)	1.11900(−6)	5.91343(−6)
Order	3.226	3.270	3.392

Roe Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	1.72217(−4)	3.13719(−4)	1.60254(−3)
40	1.79322(−5)	3.39544(−5)	1.93667(−4)
80	2.41481(−6)	4.62184(−6)	2.70813(−5)
160	3.09483(−7)	5.94444(−7)	3.46030(−6)
Order	3.025	3.001	2.940

AUSM$^+$-up Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	1.47699(−4)	2.71802(−4)	1.26574(−3)
40	1.89854(−5)	3.69259(−5)	2.24091(−4)
80	2.61576(−6)	5.16111(−6)	3.13437(−5)
160	3.36013(−7)	6.64243(−7)	4.00544(−6)
Order	2.920	2.887	2.775

Table 4.6: As Table 4.5 except with the FV4 scheme.

Rusanov Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	$3.95805(-4)$	$7.38801(-4)$	$4.44070(-3)$
40	$1.65282(-5)$	$3.13826(-5)$	$1.89074(-4)$
80	$5.41290(-7)$	$1.02393(-6)$	$5.96072(-6)$
160	$1.70040(-8)$	$3.19845(-8)$	$1.79530(-7)$
Order	4.845	4.842	4.877

Roe Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	$1.22144(-4)$	$2.35850(-4)$	$1.28707(-3)$
40	$4.48290(-6)$	$8.67097(-6)$	$4.67861(-5)$
80	$1.50958(-7)$	$2.89663(-7)$	$1.50027(-6)$
160	$5.49471(-9)$	$1.04223(-8)$	$5.04890(-8)$
Order	4.821	4.830	4.888

AUSM$^+$-up Solver

Resolution (N_c)	L_1 error	L_2 error	L_∞ error
20	$1.01946(-4)$	$2.01244(-4)$	$1.22075(-3)$
40	$3.76651(-6)$	$7.45425(-6)$	$4.44324(-5)$
80	$1.29063(-7)$	$2.53563(-7)$	$1.44834(-6)$
160	$4.93269(-9)$	$9.55944(-9)$	$4.99077(-8)$
Order	4.787	4.796	4.867

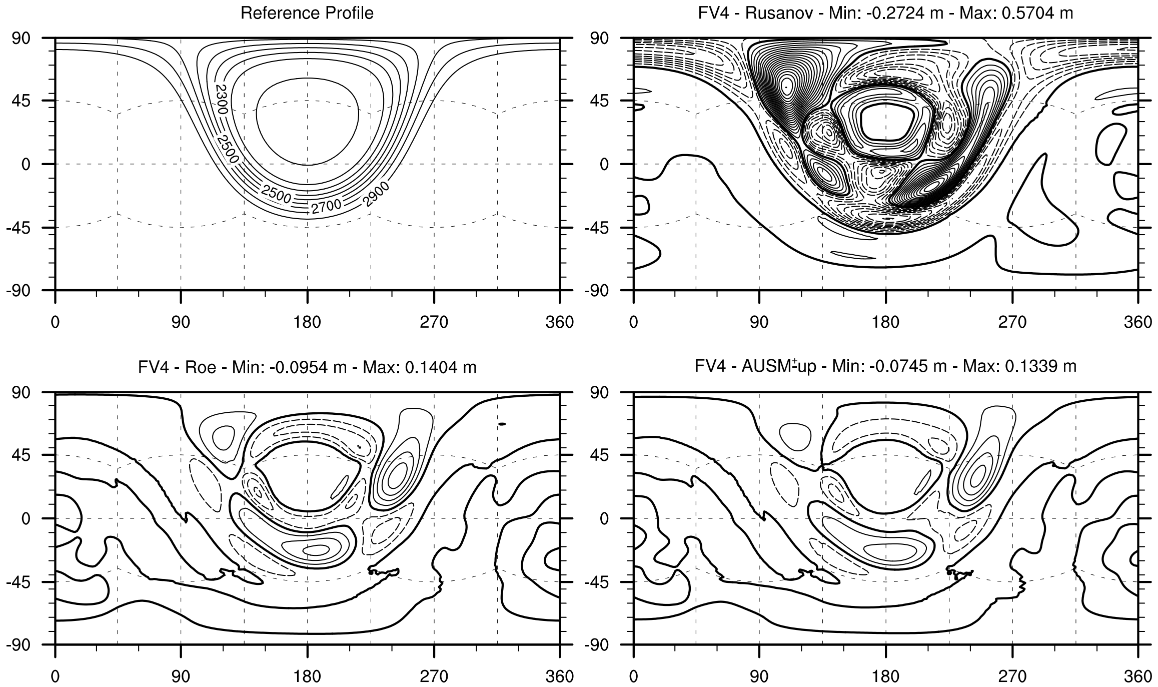


Figure 4.11: As Figure 4.10 except using the FV4 scheme. Contour lines are in units of $3 \times 10^{-2}m$.

with a timestep of 12.5 minutes.

Potential enstrophy ξ and total energy E are invariant under the shallow-water equations, and are defined by

$$\xi = \frac{(\zeta + f)^2}{2h}, \quad \text{and} \quad E = \tfrac{1}{2}h\mathbf{v}\cdot\mathbf{v} + \tfrac{1}{2}G(H^2 - z^2). \tag{4.101}$$

From the element-averages of the height and momentum field, we calculate total energy directly, weighting element-wise totals by element area. Similarly, potential enstrophy is calculated by using a central-difference approximation to the curl and then using element-averages. The resulting computed invariants are accurate up to $O(\Delta^2)$.

We compare our results with the reference solution of *Jakob-Chien et al.* (1995), which is run on the spectral transform shallow-water model (STSWM) at T426 resolution. This high resolution reference solution was computed by the German Weather Service (DWD) and is available online (http://icon.enes.org/swm/stswm/node5.html). The T426 simulation utilized a Gaussian grid with 640 $\times$ 1280 grid points in latitudinal and longitudinal direction which corresponds to a grid spacing of about 31 km at the equator. To directly compare with our model, the spectral coefficients from STSWM are sampled on the cubed-sphere grid at high-resolution Gaussian quadrature points in order to obtain element-averages of the state variables on a $40 \times 40 \times 6$ grid. Invariants are then calculated from the resampled cubed-sphere solution and our own cubed-sphere runs using standard difference operators.

The normalized total potential enstrophy and total energy difference from the initial state versus time are given for the FV3s scheme in Figure 4.13 and for the FV4 scheme in Figure 4.14. We observe very good performance of both schemes with respect to conservation of these quantities, with the best conservation properties coming from the AUSM$^+$-up Riemann solver (the Roe Riemann solver also matches very closely). The Rusanov Riemann solver performs significantly worse in all cases, due to the significant diffusion in this scheme.

4.7.5 Rossby-Haurwitz Wave

The Rossby-Haurwitz wave (test case 6 in *Williamson et al.* (1992)) is an analytic solution of the nonlinear barotropic vorticity equation on the sphere. The height and velocity field can be analytically computed with several free parameters, however for the purposes of testing the numerical model we only make use of the wave number 4 test. Our parameters are analogous to those of *Williamson et al.* (1992), who also provides expressions for the analytic fields for this test case. Again, we make use of high-order Gaussian quadrature to calculate the initial height and momentum fields.

It is well known that the wave number 4 Rossby-Haurwitz wave is susceptible to instability which can be driven by truncation error in the initial conditions (see, for

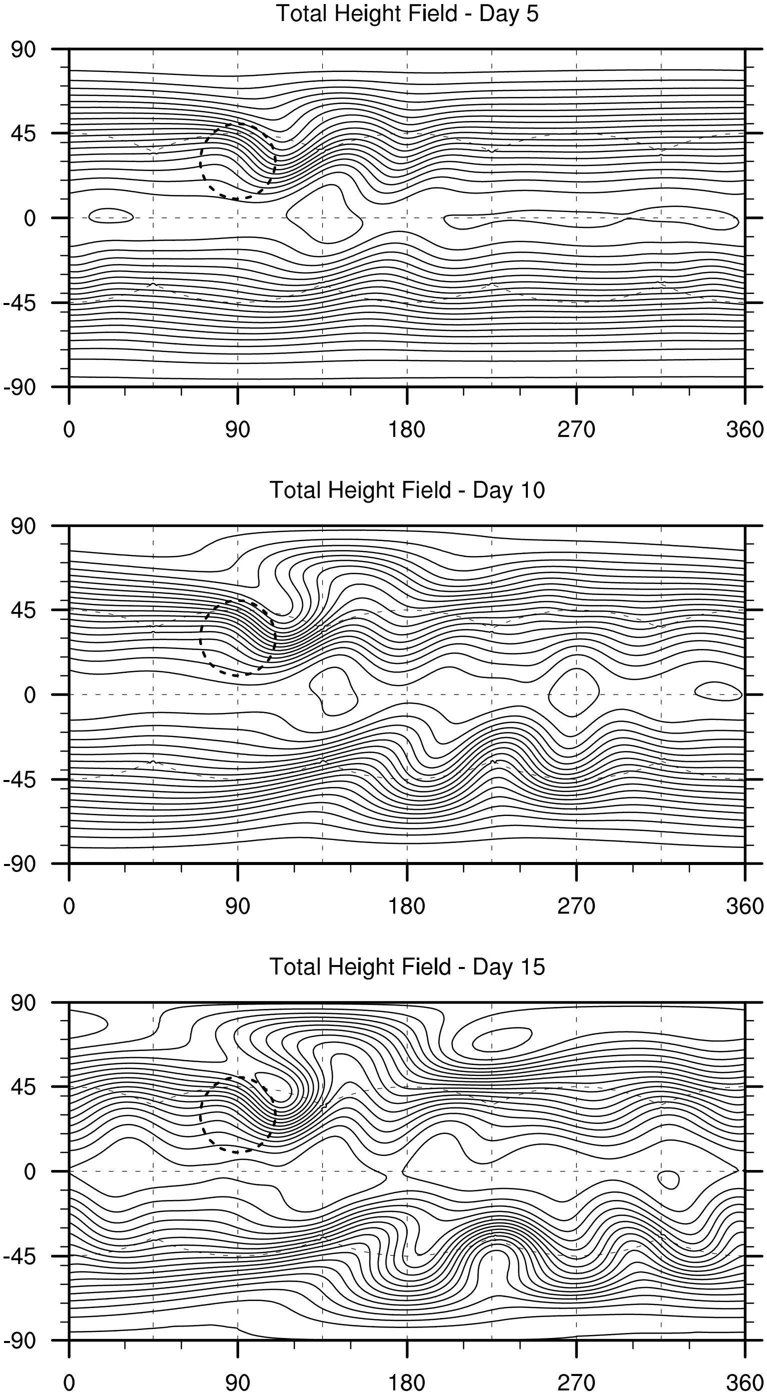


Figure 4.12: Total height field for *Williamson et al.* (1992) test case 5. We show the simulation results for the FV4 scheme with AUSM$^+$-up Riemann solver simulated on a $40 \times 40 \times 6$ grid. The dashed circle represents the location of the conical mountain. Contour levels are from 4950 m to 5950 m in intervals of 50 m, with the highest elevation being near the equator (the small enclosed contours). The results for the FV3s scheme are visually identical.

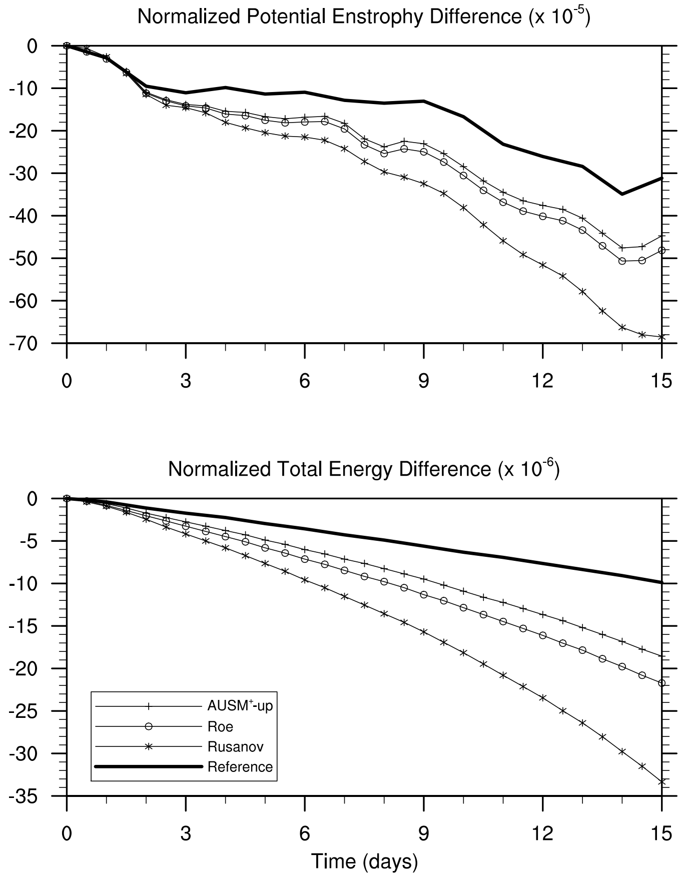


Figure 4.13: Normalized potential enstrophy (top) and total energy (bottom) difference for the flow over an isolated mountain test case using the FV3s scheme simulated on a $40 \times 40 \times 6$ grid.

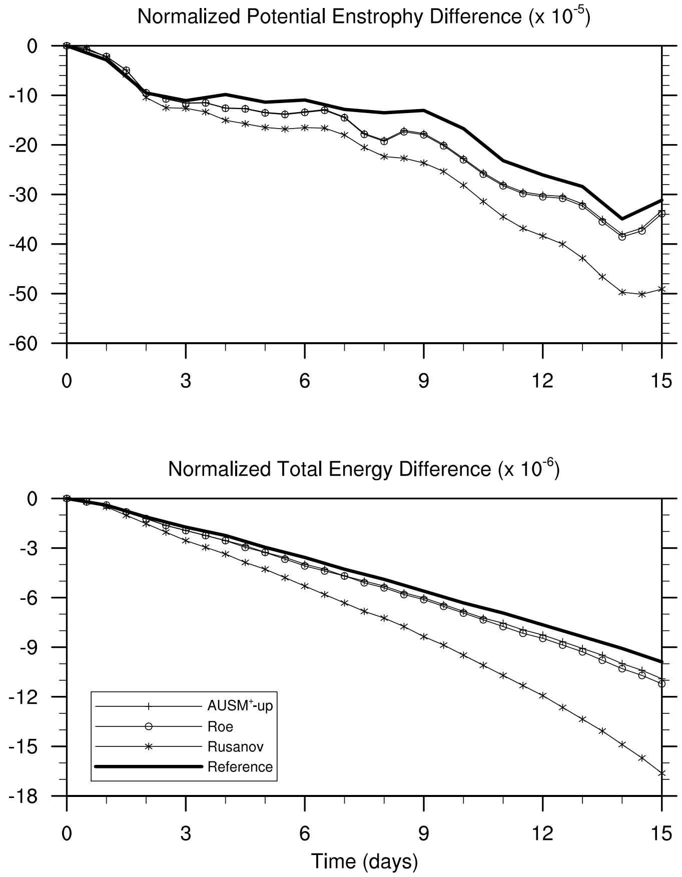


Figure 4.14: Normalized potential enstrophy (top) and total energy (bottom) difference for the flow over an isolated mountain test case using the FV4 scheme simulated on a $40 \times 40 \times 6$ grid.

example, *Thuburn and Li* (2000)), and hence will eventually collapse into a turbulent flow. The time of the breakdown varies based on the numerical scheme employed and the choice of grid resolution. For the FV3s and FV4 schemes discussed in this chapter, we begin to see breakdown at about day 80, but find that adding perturbations on the order of the scheme's truncation error can drive the collapse to as early as day 30.

We plot the height field for the FV3s scheme in Figure 4.15 and for the FV4 scheme in Figure 4.16 at day 0, 7, 14, 30, 60 and 90 at a resolution of $80 \times 80 \times 6$. The higher resolution is required for the plots to capture some of the small scale features of the wave profile. The total energy and potential enstrophy computed at each day of the simulation (up to day 14) is presented in Figures 4.17 and 4.18, compared against the STSWM reference solution at T511 (26 km) resolution truncated to the cubed-sphere at $40 \times 40 \times 6$ for consistency with Figures 4.13–4.14. The timestep is chosen to be 4.2 minutes for the $80 \times 80 \times 6$ runs and 8.4 minutes for the $40 \times 40 \times 6$ runs. Again, the Roe and AUSM^{+}-up Riemann solvers perform very well, whereas the Rusanov Riemann solver performs noticably worse. The finite-volume nature of the underlying scheme imposes fairly strong diffusivity on the energy when compared to the reference solution, but nonetheless we observe similar conservation properties to competing schemes.

4.7.6 Barotropic Instability

The barotropic instability test case of *Galewsky et al.* (2004) consists of a zonal jet with compact support at a latitude of 45°. A small height disturbance is added which causes the jet to become unstable and collapse into a highly vortical structure. The relative vorticity is used here as a comparison between the solution presented in *Galewsky et al.* (2004) (obtained via a spectral transform method) and the solution obtained by direct simulation. We present the potential vorticity at varying resolu-

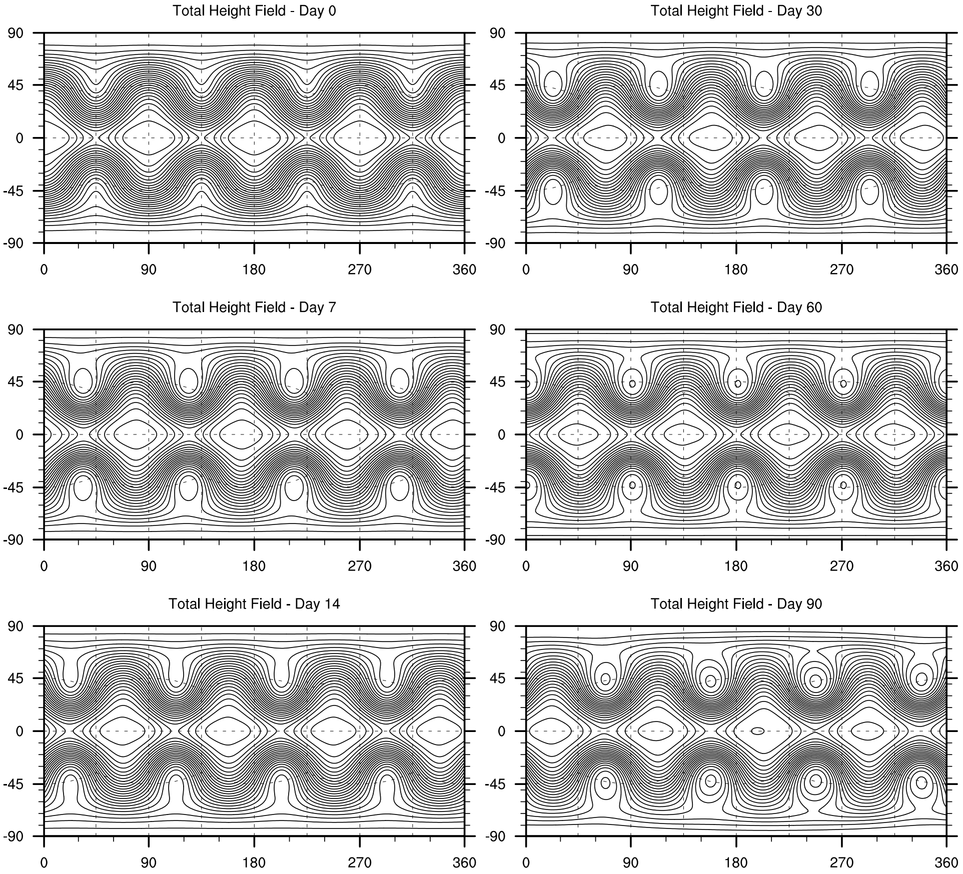


Figure 4.15: Wavenumber four Rossby-Haurwitz wave (test case 6 in *Williamson et al.* (1992)). The solution is computed on a $80 \times 80 \times 6$ grid using the FV3s scheme with AUSM$^+$-up solver on day 0, 7 and 14 (left column, from top to bottom) and day 30, 60 and 90 (right column, from top to bottom). The contour levels are from 8100 m to 10500 m in increments of 100 m, with the innermost contours being the highest.

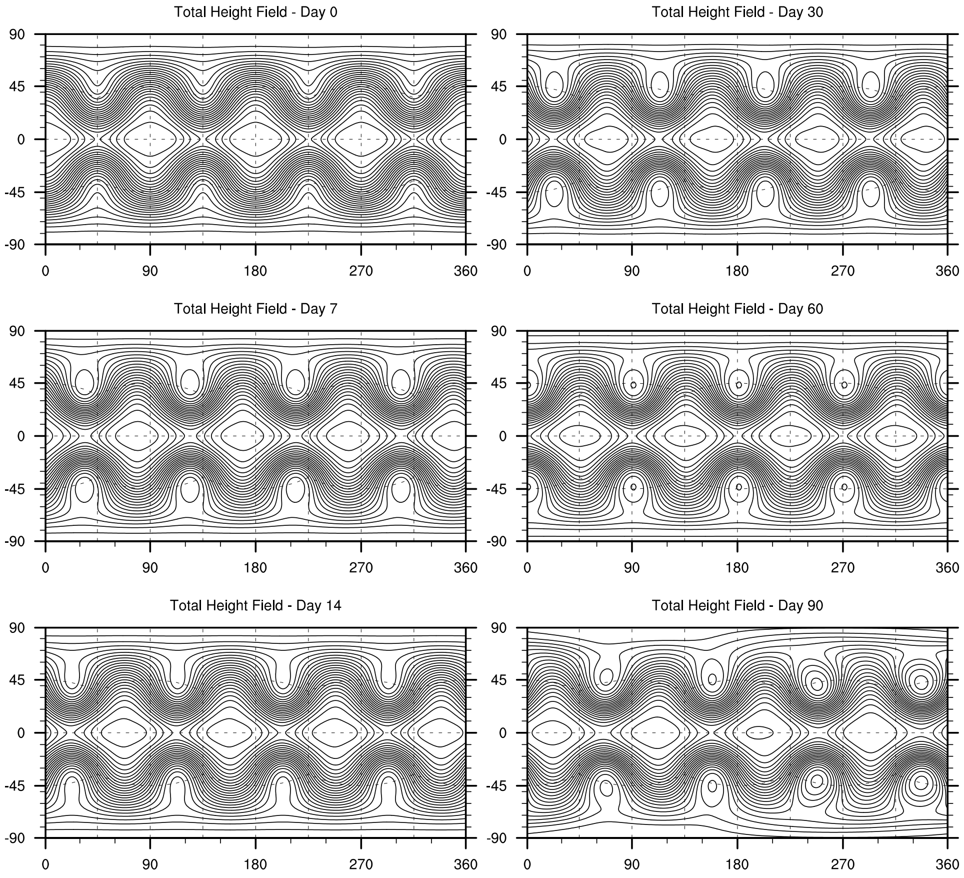


Figure 4.16: As Figure 4.15 except for the FV4 scheme.

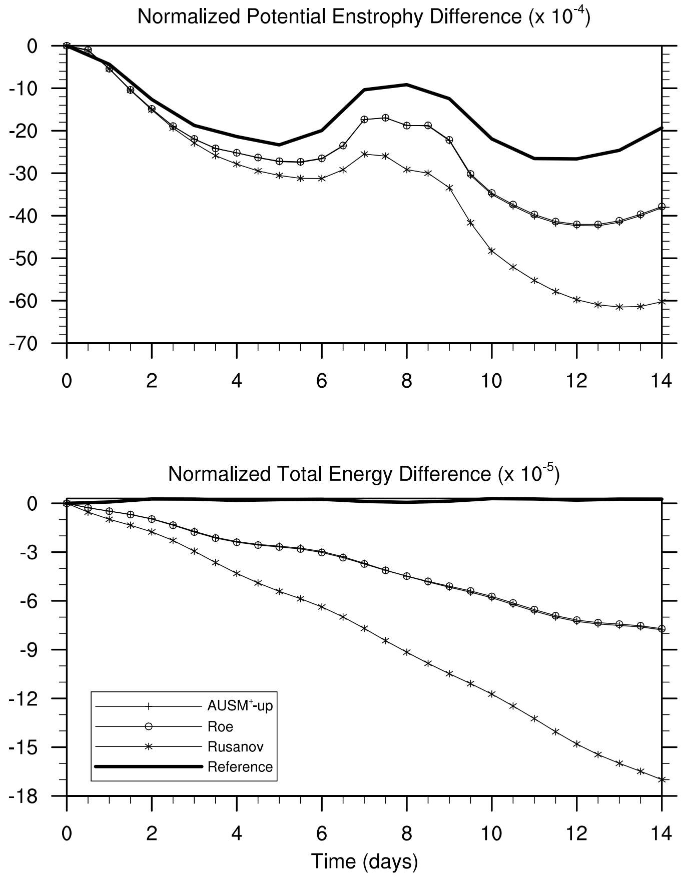


Figure 4.17: Normalized potential enstrophy (top) and potential energy (bottom) difference for the Rossby-Haurwitz wave test case using the FV3s method simulated on a 40 × 40 × 6 grid.

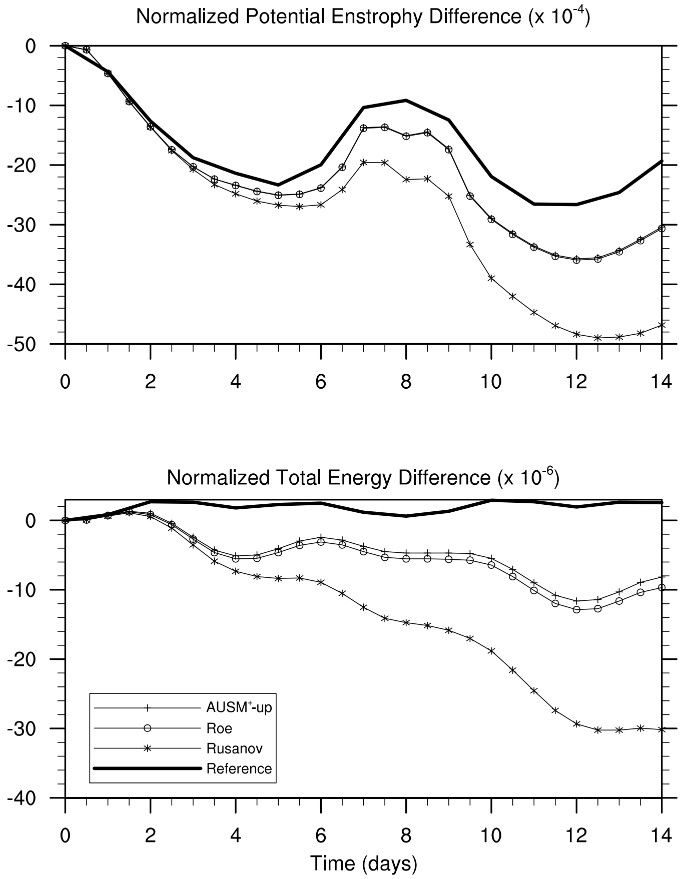


Figure 4.18: Normalized potential enstrophy (top) and potential energy (bottom) difference for the Rossby-Haurwitz wave test case using the FV4 simulated on a 40 × 40 × 6 grid.

tions for the FV3s and FV4 scheme with AUSM$^+$-up Riemann solver in Figures 4.19 and 4.20, respectively. The timestep used for this test case is 9 minutes at a resolution of $40 \times 40 \times 6$. No artificial viscosity is added in our simulation, since the numerical diffusion introduced by the finite-volume method is sufficient to ensure stability.

As observed by *St-Cyr et al.* (2008), this test case is particularly difficult for models using the cubed-sphere to handle. Since the jet passes over cubed-sphere panel edges eight times and is driven by a relatively mild perturbation, wave number four grid forcing is significant in disturbing the collapse for resolutions less than approximately $100 \times 100 \times 6$. For higher resolutions however, we observe convergence to the reference solution given by *Galewsky et al.* (2004) and similarity to the solution calculated by *Rossmanith* (2006) (except without the need to split the geostrophically balanced and unsteady modes).

4.7.7 Computational Performance

The utility of any computational scheme is a function of both accuracy and computational performance. Hence, we present the relative cost of the third- and fourth-order schemes (along with each choice of Riemann solver) in Table 4.7. As expected, we see an overall increase in cost with increasing complexity of the Riemann solver, as well as a much more significant jump in computational cost associated with going from the third- to fourth-order scheme. As is usual for explicit methods, a doubling of the resolution will lead to an eight times increase in the computational time for the scheme when keeping the CFL number constant.

Considering the significant improvement in accuracy we have observed from using the Roe or AUSM$^+$-up Riemann solvers, the added expense they incur in computation time is negligible.

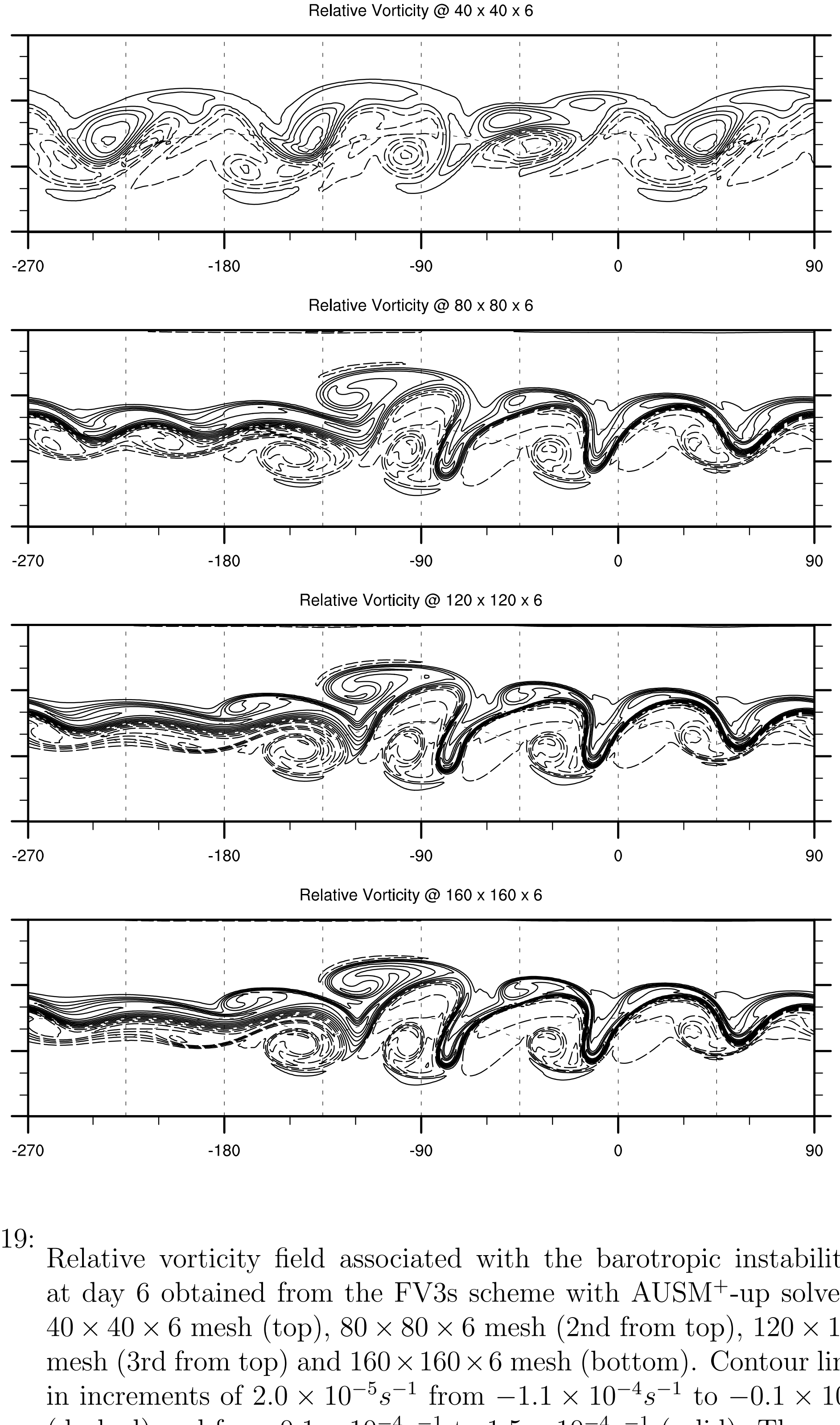


Figure 4.19: Relative vorticity field associated with the barotropic instability test at day 6 obtained from the FV3s scheme with AUSM$^+$-up solver on a $40 \times 40 \times 6$ mesh (top), $80 \times 80 \times 6$ mesh (2nd from top), $120 \times 120 \times 6$ mesh (3rd from top) and $160 \times 160 \times 6$ mesh (bottom). Contour lines are in increments of $2.0 \times 10^{-5}s^{-1}$ from $-1.1 \times 10^{-4}s^{-1}$ to $-0.1 \times 10^{-4}s^{-1}$ (dashed) and from $0.1 \times 10^{-4}s^{-1}$ to $1.5 \times 10^{-4}s^{-1}$ (solid). The zero line is omitted. Only the northern hemisphere is depicted in this plot.

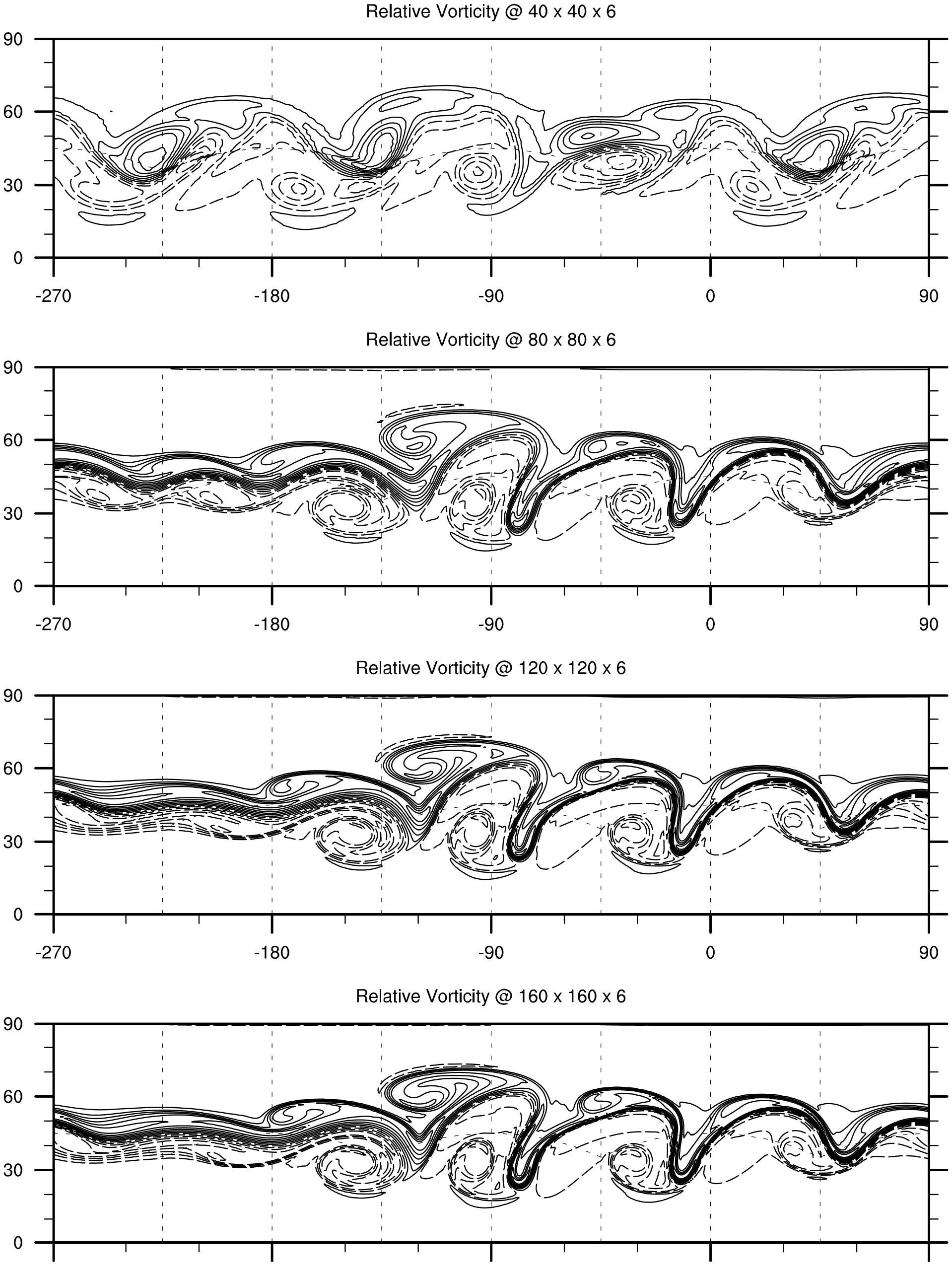


Figure 4.20: As Figure 4.19 except for the FV4 scheme.

Table 4.7: The approximate computational performance for each of the numerical schemes paired with each Riemann solver, as obtained from serial runs on a MacBook Pro with 2.4 GHz Intel Core 2 Duo. The timings correspond to the number of seconds required to simulate one day of Williamson test case 2 (described in Section 4.7.2) on a 40 × 40 × 6 grid. A CFL number of 1.0 is used in all cases.

Scheme	Rusanov	Roe	AUSM$^+$-up
FV3s	10.0 s	10.6 s	10.9 s
FV4	23.8 s	24.5 s	25.0 s

4.8 Conclusions and Future Work

In this chapter, we have successfully demonstrated both a dimension-split piecewise parabolic scheme and a fourth-order accurate piecewise cubic method for solving the shallow-water equations on the sphere. We have applied both of these schemes to a set of test problems in order to verify accuracy, stability and convergence, including the shallow-water test cases of *Williamson et al.* (1992) and the barotropic instability of *Galewsky et al.* (2004). Third- and fourth-order accuracy of these schemes is apparent in the smooth simulations tested in this chapter.

Three Riemann solvers have been considered in this analysis, including the Rusanov numerical flux, the Roe solver of *Roe* (1981), and the AUSM$^+$-up numerical flux of *Liou* (2006). Our simulations have shown that the AUSM$^+$-up flux provides the best overall accuracy when applied to various shallow-water test cases, followed very closely by the Roe flux. The Rusanov solver has demonstrated significantly worse performance in terms of accuracy and conservation of flow invariants, which we believe outweighs its simplicity. Importantly, the overall improvement in accuracy due to the Roe or AUSM$^+$-up solvers has been shown to come without a significant added computational expense.

Extension of the work described herein to a full 3D atmospheric model certainly deserves some attention. Adaption of the Riemann solvers to the full Euler equations is a trivial task, but one must be careful in the reconstruction step and in making a choice of the timestepping method. The choice of reconstruction is dependent on the vertical coordinate system, of which there are three possible routes forward: First, the high-order FV approach discussed herein is perhaps most directly applied with a semi-Lagrangian vertical coordinate, such as that described by *Lin* (2004). Second, a static terrain-following coordinate could be adopted (see, for instance, *Kasahara* (1974); *Phillips* (1957); *Wood and Staniforth* (2003)). Finally, a static height-based coordinate could be used, with topography incorporated via partial-shaved cells (see, for example, *Adcroft et al.* (1997); *Black* (1994)). For the first two choices of vertical coordinate, it is unknown what effect the vertically non-Cartesian geometry will have on the accuracy of the reconstruction step. Analysis of these approaches represents ongoing research. The second two choices of vertical coordinate also introduce issues with regard to the choice of a vertical timestepping scheme, since atmospheric models must incorporate horizontal/vertical aspect ratios that are sometimes on the order of $\sim 10^3$. In this case, the vertical CFL condition tends to be unmanagably restrictive. To overcome this problem, one could either use a modified equation set that removes vertically-propagating soundwaves (see, for example, *Arakawa and Konor* (2009); *Davies et al.* (2003)) or use a dimension-split implicit-explicit integrator (such as the IMEX-RK scheme of *Ascher et al.* (1997)). Other methods include semi-implicit treatment of sound waves or a split-explicit approach with subcycling. Research is ongoing as to the best route forward.

Acknowledgements

We would like to thank Dr. Meng-Sing Liou for his assistance with fine-tuning the AUSM$^+$-up solver for unstable shallow-water flow and Dr. Smadar Karni for her assistance with the topography parameterization. This work was supported by the NSF Collaboration in Mathematical Geosciences (CMG) grant ATM-0723440.

This work can be found in *Ullrich et al.* (2010).

CHAPTER V

Operator-Split Runge-Kutta-Rosenbrock (RKR) Methods for Non-hydrostatic Atmospheric Models

5.1 Introduction

One reason explicit time-stepping schemes are desirable in atmospheric models is locality of data, which allows the equations of motion to be evaluated with minimal communication between neighboring elements. In the context of large-scale parallel systems this benefit means that communication between processors is significantly less than with equivalent implicit methods. Nonetheless, purely explicit methods have strict time step restrictions that are required for stability. The fast moving waves in the governing equation largely determine this time step restriction, although they may possess little physical significance. In general, the maximum stable time step for explicit time discretizations is determined by the dimensionless Courant-Friedrichs-Lewy (CFL) number, which takes the form

$$\nu = \frac{c_{\max}\Delta t}{\Delta x}, \tag{5.1}$$

where $c_{\max}$ denotes the maximum wave speed of the system, Δx is the minimum grid spacing and Δt is the maximum stable time step. Most explicit time stepping methods are limited to $\nu \lesssim 1$. In atmospheric flows, sound waves are the fastest propagating wave modes, with an average speed of 340 m s^{-1} at sea level. At $\Delta x \approx$ 110 km resolution (which corresponds to about a 1° grid spacing at the equator), this leads to a time step restriction of about five minutes. Contrasting this against a vertical discretization with a minimum near-surface grid spacing of about 100 m, the maximum time step restriction is merely $\Delta t \lesssim 0.3$ seconds – 1000 times smaller than the horizontal time step. With our current computing power, atmospheric modeling would be effectively impossible at such a time step.

Various methods have been developed to deal with the computational restrictions introduced by fast waves in the atmosphere. The first method arises from a modification of the equation set to "filter" out fast-moving waves. At large horizontal scales, hydrostatic models (including *Taylor et al.* (2008) and *Lin* (2004)) are usually employed to remove the acceleration of the vertical velocity from the system. This approach eliminates vertically-propagating sound waves, but reaches its limitations when the grid size is reduced to "non-hydrostatic" scales around 10 km. Motions at this scale can be dominated by large vertical velocities, and so the vertical acceleration term cannot be neglected. Further, the dispersion relation of the hydrostatic equation set reveals that the phase speed of gravity-wave modes can be overestimated at these scales, as compared to the full non-hydrostatic equations (*Durran*, 1999). Alternatively, "sound-proof" systems of equations, including the Boussinesq equations, the anelastic system of *Ogura and Phillips* (1962), the pseudo-incompressible system of *Durran* (1989) and the unified approximation of *Arakawa and Konor* (2009), have been successfully used in models by removing sound waves from the governing equations. Nonetheless, one must be aware that these modified systems may not be valid on all scales, and therefore may not be desirable for global atmospheric models (see, for example, *Davies et al.* (2003)).

The second method for dealing with computationally fast waves relies on numerical methods which treat these modes in a stable manner. In particular, we focus on methods which integrate the full non-hydrostatic equation set, but introduce a splitting strategy to deal with fast wave modes. These approaches are generally referred to as "operator-split methods." Operator-split methods have been in use for atmospheric models for quite some time, beginning with the first semi-implicit methods of *Kwizak and Robert* (1971). Since then, semi-implicit methods have been used in atmospheric models at practically all scales (see, for example, *Bonaventura* (2000); *Thomas and Loft* (2002); *Restelli and Giraldo* (2009)). Closely related to semi-implicit methods are split-explicit and fractional step techniques. These methods are similar but instead combine explicit operators, generally splitting on the slow and fast waves. These methods were originally developed by *Gadd* (1978) for atmospheric models, but they continue to be in use today. The Weather Research and Forecasting Model (WRF, *Skamarock and Klemp* (2008)), for example, uses both semi-implicit and explicit splitting, utilizing implicit integration for vertically propagating waves and a split-explicit technique for fast waves in the horizontal.

In this chapter we introduce a new time discretization for models that split the temporal and spatial derivatives using the method of lines. The proposed method offers a simple framework for achieving high-order temporal accuracy in a semi-implicit scheme while maintaining computational efficiency. Following an operator-split Runge-Kutta-Rosenbrock (RKR) strategy, which combines an explicit Runge-Kutta (RK) method with a linearly implicit Rosenbrock step (*Rosenbrock*, 1963), we obtain a method whose maximum stable time step is constrained only by the horizontal CFL number. To maximize efficiency on parallel systems, the splitting is performed on the horizontal and vertical components of the governing equations so that the implicit solve is a local operation. This strategy is demonstrated using a high-order finite-volume method in 2D and 3D so as to verify accuracy and stability, but this approach is easily extended to other methods which independently discretize space and time, such as discontinuous Galerkin. The approach presented herein is valid for all horizontal scales, and hence may be especially well-suited for models that utilize adaptively refined meshes with scale differences.

In section 5.2 we introduce the full non-hydrostatic fluid equations and explain how to incorporate terrain-following coordinates. The RKR discretization is introduced in section 5.3, wherein we present a first-order, a second-order and a third-order temporal discretization that is stable for high-order spatial discretizations. We will demonstrate these techniques using high-order finite-volume spatial discretizations, which are explained in section 5.4, followed by numerical results in section 5.5. Our conclusions are given in section 5.6.

5.2 The non-hydrostatic fluid equations in Cartesian coordinates

We utilize the shallow-atmosphere non-hydrostatic fluid equations written in terms of the conservative variables density ρ, momentum $\rho\mathbf{u}$ (where $\mathbf{u}$ is the 3D velocity vector) and potential temperature density $\rho\theta$. In vector form, these equations are written as follows:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0, && (5.2)\\
\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p\mathcal{I}) &= -\rho g \mathbf{e}_z - f\mathbf{e}_z \times (\rho \mathbf{u}), && (5.3)\\
\frac{\partial \rho\theta}{\partial t} + \nabla \cdot (\rho\theta \mathbf{u}) &= 0. && (5.4)
\end{aligned}$$

Here $\otimes$ denotes the tensor (outer) product, $\mathcal{I}$ denotes the identity matrix, $\mathbf{e}_z$ is the basis vector in the z direction, g is gravity and f is the Coriolis parameter. A list of the constants used in this chapter and their corresponding values can be found in Table 5.1. The pressure p in the momentum equation is related to the potential

Table 5.1: List of parameters and physical constants used in this chapter.

Parameter	Description	Control Value
a	Radius of the Earth	6.37122×10^6 m
Ω	Rotational speed of the Earth	7.292×10^{-5} s^{-1}
g	Gravity	9.80616 m s^{-2}
p_0	Background surface pressure	1000 hPa
c_p	Specific heat capacity of dry air at constant pressure	1004.5 J kg^{-1} K^{-1}
c_v	Specific heat capacity of dry air at constant volume	717.5 J kg^{-1} K^{-1}
R_d	Ideal gas constant of dry air	287.0 J kg^{-1} K^{-1}

temperature density via the equation of state

$$p = p_0 \left(\frac{R_d(\rho\theta)}{p_0} \right)^{c_p/c_v}. \tag{5.5}$$

The second terms on the left-hand-side of (5.2)-(5.4) are referred to as *flux terms*, since they determine the flow rate of the conservative state variables through the edges of a spatial region. The terms on the right-hand-side of these equations are *source terms.* Nonhydrostatic mesoscale models that use a closely related equation set include the WRF model (see, for example, *Skamarock and Klemp* (2008)) and the model by *Ahmad and Lindeman* (2007).

A splitting is performed on these equations of the form

$$\begin{aligned} \rho(\mathbf{x},t) &= \overline{\rho}(\mathbf{x}) + \rho'(\mathbf{x},t), & (5.6)\\ p(\mathbf{x},t) &= \overline{p}(\mathbf{x}) + p'(\mathbf{x},t), & (5.7)\\ (\rho\theta)(\mathbf{x},t) &= \overline{\rho\theta}(\mathbf{x}) + (\rho\theta)'(\mathbf{x},t), & (5.8) \end{aligned}$$

where the mean values (denoted by the overbar) are in local hydrostatic balance, *i.e.*

$$\frac{\partial \overline{p}}{\partial z} = -\overline{\rho} g, \tag{5.9}$$

and satisfy (5.5). This choice is required to remove errors in approximating the hydrostatic state of the atmosphere that could be responsible for significant generation of spurious vertical momentum.

5.2.1 Incorporating topography

For the time stepping schemes presented in this chapter, we will be using terrain-following coordinates as introduced by *Gal-Chen and Somerville* (1975). A possible alternative to terrain-following coordinates are so-called shaved-cell methods (see, for example, *Adcroft et al.* (1997)), which remove the portions of a cell occupied by topography. Unfortunately, if terrain is accurately resolved, this approach will reduce the horizontal extent of an element, and hence the maximum allowable horizontal time step. Modified shaved-cells that do not reduce the horizontal extent of cells could also be used, but this technique may significantly degrade the accuracy of the terrain discretization.

Under an arbitrary change-of-coordinates, the non-hydrostatic governing equations (5.2)-(5.4) take the form

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{1}{J}\nabla_j (J\rho u^j) &= 0, && (5.10)\\
\frac{\partial \rho u^i}{\partial t} + \frac{1}{J}\nabla_j (J(\rho u^i u^j + pG^{ij})) &= \Gamma^i_{jk}(\rho u^j u^k + pG^{jk}) + \psi^i_g + \psi^i_c, && (5.11)\\
\frac{\partial \rho\theta}{\partial t} + \frac{1}{J}\nabla_j (J\rho\theta u^j) &= 0, && (5.12)
\end{aligned}
$$

where we have chosen to use Einstein summation notation to denote vector components. Here G^{ij} denotes the contravariant metric with covariant inverse $G_{ij} = (G^{ij})^{-1}$, $J = \det(G_{ij})$ is the volume-element and Γ^i_{jk} denotes the Christoffel symbols of the second kind, given below. The momentum equation source terms due to gravity and Coriolis are ψ^i_g and ψ^i_c, respectively.

Here we define regional terrain-following coordinates (X, Y, Z) over some orographic field $h = h(x, y)$ in terms of their Cartesian counterparts via

$$X = x, \quad Y = y, \quad Z = H\left(\frac{z-h}{H-h}\right). \tag{5.13}$$

Here H denotes the model height, at which point the terrain-following coordinates and Cartesian coordinates are identical ($Z = z$). The basis vectors $(\tilde{\mathbf{e}}_X, \tilde{\mathbf{e}}_Y, \tilde{\mathbf{e}}_Z)$ are related to the Cartesian basis vectors $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ via

$$\begin{aligned}
\tilde{\mathbf{e}}_X &= \mathbf{e}_x + \left(\frac{\partial h}{\partial x}\right)\left(1 - \frac{Z}{H}\right)\mathbf{e}_z, && (5.14)\\
\tilde{\mathbf{e}}_Y &= \mathbf{e}_y + \left(\frac{\partial h}{\partial y}\right)\left(1 - \frac{Z}{H}\right)\mathbf{e}_z, && (5.15)\\
\tilde{\mathbf{e}}_Z &= \left(1 - \frac{h}{H}\right)\mathbf{e}_z. && (5.16)
\end{aligned}$$

These relations can be inverted to obtain the Cartesian basis vectors in terms of terrain-following coordinates, as required for computing the gravitational and Coriolis forcing:

$$\begin{aligned}
\mathbf{e}_x &= \tilde{\mathbf{e}}_X - \left(\frac{\partial h}{\partial x}\right)\left(\frac{H-Z}{H-h}\right)\tilde{\mathbf{e}}_Z, && (5.17)\\
\mathbf{e}_y &= \tilde{\mathbf{e}}_Y - \left(\frac{\partial h}{\partial y}\right)\left(\frac{H-Z}{H-h}\right)\tilde{\mathbf{e}}_Z, && (5.18)\\
\mathbf{e}_z &= \left(\frac{H}{H-h}\right)\tilde{\mathbf{e}}_Z. && (5.19)
\end{aligned}$$

The contravariant metric for this choice of coordinates (again, see *Gal-Chen and Somerville* (1975)) is

$$G^{ij} = \begin{pmatrix} 1 & 0 & -\frac{(H-Z)}{(H-h)}\left(\frac{\partial h}{\partial x}\right) \\ 0 & 1 & -\frac{(H-Z)}{(H-h)}\left(\frac{\partial h}{\partial y}\right) \\ -\frac{(H-Z)}{(H-h)}\left(\frac{\partial h}{\partial x}\right) & -\frac{(H-Z)}{(H-h)}\left(\frac{\partial h}{\partial y}\right) & \left(\frac{H}{H-h}\right)^2 + \left(\frac{H-Z}{H-h}\right)^2\left[\left(\frac{\partial h}{\partial x}\right)^2 + \left(\frac{\partial h}{\partial y}\right)^2\right] \end{pmatrix}, \tag{5.20}$$

with metric determinant

$$J = 1 - \frac{h}{H}. \tag{5.21}$$

This leads to Christoffel symbols

$$\Gamma^X_{jk} = 0, \quad \Gamma^Y_{jk} = 0, \quad \Gamma^Z_{jk} = \begin{pmatrix} \frac{(H-Z)}{(H-h)}\left(\frac{\partial^2 h}{\partial x^2}\right) & \frac{(H-Z)}{(H-h)}\left(\frac{\partial^2 h}{\partial x \partial y}\right) & -\frac{1}{(H-h)}\left(\frac{\partial h}{\partial x}\right) \\ \frac{(H-Z)}{(H-h)}\left(\frac{\partial^2 h}{\partial x \partial y}\right) & \frac{(H-Z)}{(H-h)}\left(\frac{\partial^2 h}{\partial x^2}\right) & -\frac{1}{(H-h)}\left(\frac{\partial h}{\partial y}\right) \\ -\frac{1}{(H-h)}\left(\frac{\partial h}{\partial x}\right) & -\frac{1}{(H-h)}\left(\frac{\partial h}{\partial y}\right) & 0 \end{pmatrix}. \tag{5.22}$$

Under this choice of coordinates, the source terms take the form

$$\psi^X_g = 0, \quad \psi^Y_g = 0, \quad \psi^Z_g = -\rho g \frac{H}{H-h}, \tag{5.23}$$

and

$$\psi^X_c = f\rho u^Y, \quad \psi^Y_c = -f\rho u^X, \quad \psi^Z_c = f\left(\frac{z-H}{H-h}\right)\left[\left(\frac{\partial h}{\partial x}\right)(\rho u^Y) - \left(\frac{\partial h}{\partial y}\right)(\rho u^X)\right]. \tag{5.24}$$

5.3 Runge-Kutta-Rosenbrock (RKR) Schemes

The method of lines approach is one of the most popular methods for constructing high-order finite-volume methods that are applicable to general systems of partial differential equations (PDEs). Under this framework the spatial terms, including the flux and source terms, are discretized first, leading to a system of ordinary differential equations (ODEs) for the state variables within each grid cell. This system is then discretized by means of choosing an appropriate time stepping scheme. The time stepping scheme must be chosen so that the eigenvalues of the spatial operator fit within the scheme's stability region. Explicit schemes are generally computationally inexpensive, but possess a restricted stability region, whereas implicit schemes are more costly, but possess a large stability region. However, different physical processes can have eigenvalues that have dramatically different structure, and so it may not be appropriate to use a single time-stepping method to integrate all terms of the ODE system.

5.3.1 The Runge-Kutta-Rosenbrock approach

Implicit-explicit (IMEX) methods represent a category of general-purpose schemes for ODEs that couple implicit and explicit time integration methods. These methods have been in use as early as the 1970s (for example, *Crouzeix* (1980) and *Varah* (1980)). More recently, a family of implicit-explicit Runge-Kutta (IMEX-RK) schemes was collected into a general framework by *Ascher et al.* (1997) in their seminal paper. They showed that it is possible to achieve an essentially arbitrary order-of-accuracy by correctly interleaving explicit and implicit steps, although with increasing computational expense.

IMEX methods are usually applied to an ODE of the form

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{f}(\mathbf{q}) + \mathbf{g}(\mathbf{q}), \tag{5.25}$$

where $\mathbf{q}$ is some state vector. In particular, we assume that the terms $\mathbf{f}(\mathbf{q})$ are not stiff; mathematically, one can think of this as saying that the eigenvalues of the operator $\mathbf{f}$ are close to the origin. On the other hand, the terms grouped under the $\mathbf{g}(\mathbf{q})$ operator are assumed stiff, containing eigenvalues which are potentially unbounded. This stiffness can originate from short time-scale behavior, such as chemistry or, as in our case, from geometric stiffness due to a discrepancy in grid spacing in different coordinate directions. Since we focus on the dynamical aspects of the atmosphere, $\mathbf{f}(\mathbf{q})$ is assumed to be some horizontal spatial discretization of the fluid equations with grid spacing Δx and $\mathbf{g}(\mathbf{q})$ is a vertical discretization with grid spacing $\Delta z \ll \Delta x$.

The much smaller vertical scale leads to eigenvalues of $\mathbf{g}$ that are typically far from the origin. Herein, we will see that the terms $\mathbf{f}(\mathbf{q})$ and $\mathbf{g}(\mathbf{q})$ arise naturally out of some spatial discretization of the non-hydrostatic model equations (5.2)-(5.5).

To improve the performance of the IMEX methods we focus on the family of operator-split RKR methods, which are identical to IMEX schemes except replacing the computationally expensive implicit step with a so-called Rosenbrock step. Rosenbrock methods were originally developed by *Rosenbrock* (1963) in the 1960s and later refined by *Nørsett and Wolfbrandt* (1979). More recently, a framework for high-order RKR time stepping methods has been presented by *Jebens et al.* (2010) for use in atmospheric models utilizing cut-cells.

In an implicit approach, one usually ends up with a nonlinear system of equations of the form

$$\mathbf{F}(\mathbf{x}) = \mathbf{0} \tag{5.26}$$

that must be solved numerically for some vector $\mathbf{x}$. Perhaps the most well-known and robust technique for solving this system is the Newton-Krylov algorithm, which is an iterative approach defined by

$$\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} - \left[\left. \frac{d\mathbf{F}}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^{(i-1)}} \right]^{-1} \mathbf{F}(\mathbf{x}^{(i-1)}). \tag{5.27}$$

When applied to systems of ODEs obtained from time-split PDE systems, the Newton-Krylov method is usually initialized by taking $\mathbf{x}^{(0)}$ to be the value of $\mathbf{x}$ obtained at the previous time step. Although this method converges quadratically, the Jacobian $d\mathbf{F}/d\mathbf{x}$ must nonetheless be computed (numerically or analytically) at every iteration. Under the Rosenbrock method, one does not require nonlinear convergence of (5.26), and instead only takes one step of (5.27). This significantly reduces the costs associated with the implicit step and restricts the number of Jacobian evaluations to one per Rosenbrock step.

In order to obtain high-order-accuracy in time with our scheme, we consider a general Runge-Kutta method with interleaved explicit and linearly implicit Rosenbrock steps. In the general RKR framework, we define the initial data by

$$\mathbf{q}^{(0)} = \mathbf{q}^n, \tag{5.28}$$

and then each subsequent step by

$$\mathbf{q}^{(i)} = \sum_{k=0}^{i-1} \alpha_{i,k}\mathbf{q}^{(k)} + \mathcal{E}_i\beta_i\Delta t\mathbf{f}(\mathbf{q}^{(i-1)}) + (1-\mathcal{E}_i)\beta_i\Delta t\left(\mathbf{I} + \beta_i\Delta t\mathbf{Dg}(\mathbf{q}^{(i-1)})\right)^{-1}\mathbf{g}(\mathbf{q}^{(i-1)}), \tag{5.29}$$

where $\alpha_{i,k}$ and β_i are arbitrary coefficients, Δt is the time step, $\mathbf{Dg}$ denotes the Jacobian of $\mathbf{g}$, and $\mathcal{E}$ is a binary indicator variable,

$$\mathcal{E}_i = \begin{cases} 1, & \text{for an explicit step,} \\ 0, & \text{for a Rosenbrock step.} \end{cases} \tag{5.30}$$

This scheme description is interchangeable with the dual Butcher-tableau approach of *Ascher et al.* (1997), but is preferred in this chapter since it is more closely linked with the method's actual implementation.

5.3.2 A crude splitting scheme

Perhaps the simplest stable scheme for splitting the explicit and linearly implicit components of the evolution equations involves simply applying a high-order explicit Runge-Kutta operator to the explicit terms followed by the Rosenbrock operator applied at the full time step. If we use the third-order strong-stability-preserving (SSP) RK3 scheme of *Gottlieb et al.* (2001) for the explicit component, this scheme proceeds according to

$$\begin{aligned}
\mathbf{q}^{(1)} &= \mathbf{q}^n + \Delta t\mathbf{f}(\mathbf{q}^n), & (5.31)\\
\mathbf{q}^{(2)} &= \frac{3}{4}\mathbf{q}^{(1)} + \frac{1}{4}\mathbf{q}^{(1)} + \frac{\Delta t}{4}\mathbf{f}(\mathbf{q}^{(1)}), & (5.32)\\
\mathbf{q}^{(3)} &= \frac{1}{3}\mathbf{q}^n + \frac{2}{3}\mathbf{q}^{(2)} + \frac{2\Delta t}{3}\mathbf{f}(\mathbf{q}^{(2)}), & (5.33)\\
\mathbf{q}^{n+1} \equiv \mathbf{q}^{(4)} &= \mathbf{q}^{(3)} + \Delta t\left(\mathbf{I} - \Delta t\mathbf{Dg}(\mathbf{q}^{(3)})\right)^{-1}\mathbf{g}(\mathbf{q}^{(3)}) & (5.34)
\end{aligned}$$

Here the indices n and $n+1$ denote the current and future time step. This scheme is first-order accurate in time and extremely diffusive (as demonstrated in section 5.5.1). As a consequence, we do not recommend using this scheme in practice.

5.3.3 The second-order-accurate Strang-Carryover scheme

To achieve second-order accuracy in time, we suggest a splitting scheme pointed out by Steve Ruuth (2010, personal communication) and based on Strang splitting of the explicit and implicit operators. This scheme combines a third-order SSP RK3 step with one implicit solve per time step. However, at the first time step, we must perform one additional implicit operation, storing

$$\mathcal{G}^0 = \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{Dg}(\mathbf{q}^0)\right)^{-1}\mathbf{g}(\mathbf{q}^0). \tag{5.35}$$

After this initialization step, the algorithm proceeds as follows:

$$\begin{aligned}
\mathbf{q}^{(1)} &= \mathbf{q}^n + \frac{\Delta t}{2}\mathcal{G}^n, && (5.36)\\
\mathbf{q}^{(2)} &= \mathbf{q}^{(1)} + \Delta t\mathbf{f}(\mathbf{q}^{(1)}), && (5.37)\\
\mathbf{q}^{(3)} &= \frac{3}{4}\mathbf{q}^{(1)} + \frac{1}{4}\mathbf{q}^{(2)} + \frac{\Delta t}{4}\mathbf{f}(\mathbf{q}^{(2)}), && (5.38)\\
\mathbf{q}^{(4)} &= \frac{1}{3}\mathbf{q}^{(1)} + \frac{2}{3}\mathbf{q}^{(3)} + \frac{2\Delta t}{3}\mathbf{f}(\mathbf{q}^{(3)}), && (5.39)\\
\mathcal{G}^{n+1} &= \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{Dg}(\mathbf{q}^{(4)})\right)^{-1}\mathbf{g}(\mathbf{q}^{(4)}), && (5.40)\\
\mathbf{q}^{n+1} &= \mathbf{q}^{(4)} + \frac{\Delta t}{2}\mathcal{G}^{n+1}. && (5.41)
\end{aligned}$$

This scheme can achieve third-order linear and nonlinear accuracy in $\mathbf{f}$ plus second-order accuracy in $\mathbf{g}$.

5.3.4 The Ascher-Ruuth-Spiteri (2,3,3) scheme

A third-order operator-split RKR scheme can be obtained from the Ascher-Ruuth-Spiteri (2,3,3) scheme of *Ascher et al.* (1997), hereafter referred to as the ARS(2,3,3) scheme. If we simply replace the implicit solve with a Rosenbrock step, the resulting scheme takes the form

$$\begin{aligned}
\mathbf{q}^{(1)} &= \mathbf{q}^n + \gamma_c\Delta t\mathbf{f}(\mathbf{q}^n), && (5.42)\\
\mathbf{q}^{(2)} &= \mathbf{q}^{(1)} + \gamma_c\Delta t\left(\mathbf{I} - \gamma_c\Delta t\mathbf{Dg}(\mathbf{q}^{(1)})\right)^{-1}\mathbf{g}(\mathbf{q}^{(1)}), && (5.43)\\
\mathbf{q}^{(3)} &= \frac{1}{\gamma_c}\mathbf{q}^n + \frac{(3\gamma_c - 2)}{\gamma_c}\mathbf{q}^{(1)} + \frac{(1 - 2\gamma_c)}{\gamma_c}\mathbf{q}^{(2)} + 2(1-\gamma_c)\Delta t\mathbf{f}(\mathbf{q}^{(2)}), && (5.44)\\
\mathbf{q}^{(4)} &= \mathbf{q}^{(3)} + \gamma_c\Delta t\left(\mathbf{I} - \gamma_c\Delta t\mathbf{Dg}(\mathbf{q}^{(3)})\right)^{-1}\mathbf{g}(\mathbf{q}^{(3)}), && (5.45)\\
\mathbf{q}^{n+1} &= -\frac{1}{2}\mathbf{q}^n - \frac{3\gamma_c}{2}\mathbf{q}^{(1)} + \frac{3}{2}\mathbf{q}^{(2)} + \frac{3(3\gamma_c - 2)}{2}\mathbf{q}^{(3)} + \frac{1}{2\gamma_c}\mathbf{q}^{(4)} + \frac{\Delta t}{2}\mathbf{f}(\mathbf{q}^{(4)}), && (5.46)
\end{aligned}$$

where

$$\gamma_c = \frac{3+\sqrt{3}}{6}. \qquad (5.47)$$

This scheme is linearly third-order accurate in both $\mathbf{f}$, $\mathbf{g}$ and any cross-terms that arise from the integration procedure, but is only nonlinearly third-order accurate in $\mathbf{f}$. In fact, when $\mathbf{g} = 0$ the stability region for this scheme is exactly the stability region of the usual three-stage third-order-accurate Runge-Kutta operator. This scheme requires three explicit steps per time step and two Rosenbrock steps, with each Rosenbrock step consisting of a single evaluation of the Jacobian and a single linear solve. As a consequence, the overall computational cost of this method is approximately twice that of the Strang-carryover scheme.

5.4 Spatial discretization

In this section we turn our attention to the spatial discretization of the 3D non-hydrostatic governing equations (5.2)-(5.4) using a high-order finite-volume scheme. In the full finite-volume approach we first integrate the Euler equations in the form (5.2)-(5.4) over an element $\mathcal{Z}$ (with volume $|\mathcal{Z}|$) and make use of Gauss' divergence theorem to write the flux term as an integral around the boundary $\partial\mathcal{Z}$, giving

$$\frac{\partial}{\partial t}\overline{\overline{\mathbf{q}}} + \frac{1}{|\mathcal{Z}|}\iint\limits_{\partial\mathcal{Z}} \boldsymbol{\mathcal{F}} \cdot \mathbf{n} dS = \overline{\overline{\psi}}_C + \overline{\overline{\psi}}_G. \tag{5.48}$$

Here the flux integral is taken over the surface with normal vector $\mathbf{n}$ and infinitesimal area element dS. The term $\boldsymbol{\mathcal{F}} \cdot \mathbf{n}$ is a vector quantity that denotes the outward flux of each of the state variables perpendicular to the boundary. The double overline denotes a 3D average of the form

$$\overline{\overline{\phi}} = \frac{1}{|\mathcal{Z}|}\int\limits_{\mathcal{Z}} \phi dV. \tag{5.49}$$

The term $dV = JdXdYdZ$ denotes the infinitesimal volume element, which can vary depending on the geometry. Here $\overline{\overline{\mathbf{q}}}$ again denotes the averaged state vector in cell $\mathcal{Z}$.

Likewise, $\overline{\overline{\psi}}_C$ and $\overline{\overline{\psi}}_G$ respectively denote the source terms due to the Coriolis force and gravity. Note that the volume-averaged formulation (5.48) is exactly equivalent to the original non-hydrostatic equations, and it is left to us to define an appropriate discretization over each of the terms in this expression.

For simplicity we will define our numerical methods on a regular Cartesian grid. Elements are equally spaced with grid spacing ΔX in the X direction, ΔY in the Y direction and ΔZ in the Z direction. Element centroids $\mathbf{X}_{i,j,k} = (X_i, Y_j, Z_k)$ can then be written as

$$X_i = i\Delta X, \quad Y_j = j\Delta Y, \quad Z_k = k\Delta Z, \tag{5.50}$$

with spacial indices (i, j, k). Edges (or faces) are midway between neighboring element centroids, and so are defined by half-indices and denoted by $\partial\mathcal{Z}$. For example, the edge $\partial\mathcal{Z}_{i+1/2,j,k}$ is at the interface between element (i, j, k) and $(i+1, j, k)$ and defines a plane that is constant in both Y and Z. Quantities which are defined as edge averages will be denoted by a single overline. Average edge fluxes are defined at element edges, denoted here by $\overline{\mathbf{F}}_{i+1/2,j,k}$ for a flux across edge $(i + 1/2, j, k)$, and defined by

$$\overline{\mathbf{F}}_{i+1/2,j,k} = \frac{1}{|\partial\mathcal{Z}|_{i+1/2,j,k}} \iint\limits_{\partial\mathcal{Z}_{i+1/2,j,k}} \boldsymbol{\mathcal{F}} \cdot \mathbf{n} dS, \tag{5.51}$$

Hence, the volume averaged formulation (5.48) can be written as

$$\frac{\partial}{\partial t}\overline{\overline{\mathbf{q}}}_{i,j,k} = \mathbf{H}(\mathbf{q}) + \mathbf{V}(\mathbf{q}), \tag{5.52}$$

where

$$\mathbf{H}(\mathbf{q}) = \overline{\mathbf{F}}_{i-1/2,j,k} - \overline{\mathbf{F}}_{i+1/2,j,k} + \overline{\mathbf{F}}_{i,j-1/2,k} - \overline{\mathbf{F}}_{i,j+1/2,k} + \overline{\overline{\psi}}_C, \tag{5.53}$$

and

$$\mathbf{V}(\mathbf{q}) = \overline{\mathbf{F}}_{i,j,k-1/2} - \overline{\mathbf{F}}_{i,j,k+1/2} + \overline{\overline{\psi}}_G. \tag{5.54}$$

The choice of splitting into $\mathbf{H}(\mathbf{q})$ (horizontal) and $\mathbf{V}(\mathbf{q})$ (vertical) is analogous to (5.25), with $\mathbf{H}(\mathbf{q})$ denoting terms which are usually significantly less stiff than terms of $\mathbf{V}(\mathbf{q})$.

In the upwind finite-volume framework, we require a sub-grid-scale reconstruction within each element which is used to capture features which cannot be resolved at the grid scale. The sub-grid-scale reconstruction is then used for both computing the approximate averaged flux across each edge. Since reconstructions are inherently discontinuous across edges, a Riemann problem must be solved at each interface to obtain averaged edge fluxes. So-called approximate Riemann solvers that are typically used to solve each Riemann problem are pointwise operators that take as input the approximate state vector on each side of the interface (for simplicity referred to as left and right states). Hence, the simplest form for an approximate Riemann solver is

$$\mathbf{F}^* = \mathbf{F}^*(\mathbf{q}^L, \mathbf{q}^R), \tag{5.55}$$

where $\mathbf{q}^L$ is the left state and $\mathbf{q}^R$ is the right state.

Using high-order reconstruction formulae, we can directly reconstruct edge averages from cell averages in the X direction via

$$\begin{aligned}
\overline{\mathbf{q}}^L_{i+1/2,j,k} &= \frac{1}{30}\overline{\overline{\mathbf{q}}}_{i-2,j,k} - \frac{13}{60}\overline{\overline{\mathbf{q}}}_{i-1,j,k} + \frac{47}{60}\overline{\overline{\mathbf{q}}}_{i,j,k} + \frac{9}{20}\overline{\overline{\mathbf{q}}}_{i+1,j,k} - \frac{1}{20}\overline{\overline{\mathbf{q}}}_{i+2,j,k}, \quad (5.56)\\
\overline{\mathbf{q}}^R_{i+1/2,j,k} &= -\frac{1}{20}\overline{\overline{\mathbf{q}}}_{i-1,j,k} + \frac{9}{20}\overline{\overline{\mathbf{q}}}_{i,j,k} + \frac{47}{60}\overline{\overline{\mathbf{q}}}_{i+1,j,k} - \frac{13}{60}\overline{\overline{\mathbf{q}}}_{i+2,j,k} + \frac{1}{30}\overline{\overline{\mathbf{q}}}_{i+3,j,k}. (5.57)
\end{aligned}$$

The formula for the Y direction is identical except with the j indices varied. At vertical edges we make use of a more compact third-order-accurate stencil which leads to

$$\begin{aligned}
\overline{\mathbf{q}}^L_{i,j,k+1/2} &= -\frac{1}{6}\overline{\overline{\mathbf{q}}}_{i,j,k-1} + \frac{5}{6}\overline{\overline{\mathbf{q}}}_{i,j,k} + \frac{2}{6}\overline{\overline{\mathbf{q}}}_{i,j,k+1}, \quad (5.58)\\
\overline{\mathbf{q}}^R_{i,j,k+1/2} &= \frac{2}{6}\overline{\overline{\mathbf{q}}}_{i,j,k} + \frac{5}{6}\overline{\overline{\mathbf{q}}}_{i,j,k+1} - \frac{1}{6}\overline{\overline{\mathbf{q}}}_{i,j,k+2}. \quad (5.59)
\end{aligned}$$

If we use these reconstructed edge averages as inputs for a Riemann solver, we will obtain a scheme which is formally second-order accurate.

5.4.1 Fourth-order horizontal accuracy in 3D

The problem with using edge averages as inputs to the Riemann solver is that the Riemann solution operator is inherently pointwise, whereas edge averages are only second-order approximations to the pointwise edge-center value of the state vector. However, one may apply a deconvolution operator (*Barad and Colella* (2005)) to convert edge averages to pointwise values (here denoted with a subscript (0)). For example, along an edge of constant X, it can be verified that

$$\mathbf{q}_{(0)i+1/2,j,k} = \overline{\mathbf{q}}_{i+1/2,j,k} - \frac{\Delta Y^2}{24}\left(\frac{\partial^2 \mathbf{q}}{\partial Y^2}\right)_{(0)i+1/2,j,k} \tag{5.60}$$

is a fourth-order approximation to $\mathbf{q}_{(0)i+1/2,j,k}$, the value of the state vector at the center-point of edge $\partial\mathcal{Z}_{i+1/2,j,k}$. Here $\overline{\mathbf{q}}_{i+1/2,j,k}$ is the edge-average of the state vector and $(\partial^2\mathbf{q}/\partial Y^2)_{(0)i+1/2,j,k}$ is the second derivative in Y of the state vector evaluated at the edge center-point. A second-order approximation to $(\partial^2\mathbf{q}/\partial Y^2)_{(0)i+1/2,j,k}$ can also be used, such as

$$\left(\frac{\partial^2 \mathbf{q}}{\partial Y^2}\right)_{(0)i+1/2,j,k} \approx \frac{\overline{\mathbf{q}}_{i+1/2,j-1,k} - 2\overline{\mathbf{q}}_{i+1/2,j,k} + \overline{\mathbf{q}}_{i+1/2,j+1,k}}{\Delta Y^2}, \tag{5.61}$$

which, in combination with (5.60), also gives a fourth-order approximation to $\mathbf{q}_{(0)i+1/2,j,k}$. The Riemann solver is then applied pointwise to $\mathbf{q}^L_{(0)i+1/2,j,k}$ and $\mathbf{q}^R_{(0)i+1/2,j,k}$, and a convolution operation is applied to retrieve the edge-average of the flux vector $\overline{\mathbf{F}}_{i+1/2,j,k}$,

$$\overline{\mathbf{F}}_{i+1/2,j,k} = \mathbf{F}_{(0)i+1/2,j,k} + \frac{\Delta Y^2}{24}\left(\frac{\partial^2 \mathbf{F}}{\partial Y^2}\right)_{(0)i+1/2,j,k}. \tag{5.62}$$

In this case, the edge-average of the flux vector is a fourth-order-accurate estimate of the flux across the given edge.

To obtain fourth-order accuracy overall, the source terms of the horizontal momentum equations must also be evaluated with at least third-order accuracy. Simply evaluating source terms using cell averages $\overline{\overline{\mathbf{q}}}_{i,j,k}$ only leads to second-order accurate approximations of these terms. Hence, to obtain high-order accuracy we again apply a de-convolution procedure to obtain an approximation to the state vector at the element center-point,

$$\mathbf{q}_{(0)i,j,k} = \overline{\overline{\mathbf{q}}}_{i,j,k} - \frac{\Delta X^2}{24}\left(\frac{\partial^2 \mathbf{q}}{\partial X^2}\right)_{(0)i,j,k} - \frac{\Delta Y^2}{24}\left(\frac{\partial^2 \mathbf{q}}{\partial Y^2}\right)_{(0)i,j,k}. \tag{5.63}$$

Source terms are then evaluated at the element center-point ($\psi_{(0)i,j,k} = \psi(\mathbf{q}_{(0)i,j,k})$), and a convolution procedure is applied to re-average the source term over each element,

$$\overline{\overline{\psi}}_{i,j,k} = \psi_{(0)i,j,k} + \frac{\Delta X^2}{24}\left(\frac{\partial^2 \psi}{\partial X^2}\right)_{(0)i,j,k} + \frac{\Delta Y^2}{24}\left(\frac{\partial^2 \psi}{\partial Y^2}\right)_{(0)i,j,k}. \tag{5.64}$$

Again, all second derivatives are approximated to second-order accuracy using (5.61),

$$\left(\frac{\partial^2 \psi}{\partial X^2}\right)_{(0)i,j,k} \approx \frac{\overline{\overline{\psi}}_{i-1,j,k} - 2\overline{\overline{\psi}}_{i,j,k} + \overline{\overline{\psi}}_{i+1,j,k}}{\Delta X^2}, \tag{5.65}$$

and

$$\left(\frac{\partial^2 \psi}{\partial Y^2}\right)_{(0)i,j,k} \approx \frac{\overline{\overline{\psi}}_{i,j-1,k} - 2\overline{\overline{\psi}}_{i,j,k} + \overline{\overline{\psi}}_{i,j+1,k}}{\Delta Y^2}. \tag{5.66}$$

In our treatment the edge averages in the vertical direction are used as direct input to the Riemann solver. This implies that this method is only second-order accurate in ΔZ, but due to the typically small vertical velocity this often does not have any effect on the order-of-convergence of the method.

5.4.2 The AUSM$^+$-up Riemann Solver

The AUSM$^+$-up approximate Riemann solver of *Liou* (2006) was recently developed with the goal of enhancing accuracy of the Riemann solution in the low-Mach number regime. Many other commonly used Riemann solvers, including the popular solver of *Rusanov* (1961) and the solver of *Roe* (1981), do a poor job in the very low Mach number regime since they introduce a significant amount of numerical diffusion that can smear out the solution (see *Ullrich et al.* (2010)). Here we give a short overview of the algorithmic implementation of this solver without delving into the mathematical details.

In general, standalone Riemann solvers require that the velocity components of the input state vector be written in an orthogonal frame. In basic Cartesian coordinates this requirement is already met by the reconstructed edge velocity. However, in the presence of topography the coordinate basis vectors are not orthogonal and so the components of the velocity vector must be transformed into an orthogonal frame before proceeding. Orthogonalization is performed by multiplying the velocity vector by an orthogonalization matrix, which is described in *Bale* (2002) and *Ullrich et al.* (2010) and so is not repeated here. The computed momentum flux, which is computed in the orthogonal frame, must similarly be transformed back into the coordinate frame, which is simply computed by multiplying the Riemann flux by the inverse of the orthogonalization matrix.

Here we show the flux calculation for an edge of constant X. The calculation for an edge of constant Y or Z is analogous except utilizing v or w as the perpendicular velocity vector. Given a left state vector $\mathbf{q}_L = (\rho_L, (\rho u)_L, (\rho v)_L, (\rho w)_L, (\rho\theta)_L)$ and right state vector $\mathbf{q}_R = (\rho_R, (\rho u)_R, (\rho v)_R, (\rho w)_R, (\rho\theta)_R)$ assumed to be obtained from some reconstruction procedure and with orthogonal velocity components, we define the *averaged density*,

$$\rho_{1/2} = \tfrac{1}{2}\left(\rho_L + \rho_R\right), \tag{5.67}$$

averaged gravity wave speed,

$$a_{1/2} = \frac{1}{2}\left[\sqrt{\frac{\gamma p_R}{\rho_R}} + \sqrt{\frac{\gamma p_L}{\rho_L}}\right], \tag{5.68}$$

perpendicular Mach numbers at the interface,

$$M_L = \frac{u_L}{a_{1/2}}, \quad \text{and} \quad M_R = \frac{u_R}{a_{1/2}}, \tag{5.69}$$

and *mean local Mach number*,

$$\overline{M}^2 = \frac{u_L^2 + u_R^2}{2a_{1/2}^2}. \tag{5.70}$$

Here p_L and p_R are the corresponding left and right pressures, which are calculated from the state vector via (5.5). Similarly, $u_L = (\rho u)_L/\rho_L$ and $u_R = (\rho u)_R/\rho_R$ are the interfacial velocities.

The *interface Mach number* is then defined as

$$M_{1/2} = \mathcal{M}^+_{(4)}(M_L) + \mathcal{M}^-_{(4)}(M_R) - K_p \max(1 - \sigma\overline{M}^2, 0)\frac{p_R - p_L}{\rho_{1/2}a_{1/2}^2}, \tag{5.71}$$

where

$$\mathcal{M}^\pm_{(2)}(M) = \tfrac{1}{4}(M \pm 1)^2, \quad \mathcal{M}^\pm_{(4)}(M) = \begin{cases} \tfrac{1}{2}(M \pm |M|) & \text{if } |M| \geq 1, \\ \mathcal{M}^\pm_{(2)}(M)(1 \mp 16\tilde{\beta}\mathcal{M}^\mp_{(2)}(M)) & \text{otherwise,} \end{cases} \tag{5.72}$$

The *advective component of the flux* is then defined by

$$\dot{m}_{1/2} = a_{1/2}M_{1/2}\begin{cases} \rho_L & \text{if } M_{1/2} > 0, \\ \rho_R & \text{otherwise.} \end{cases} \tag{5.73}$$

To obtain the *pressure-driven component of the flux*, we make use of the definition

$$\mathcal{P}^{\pm}_{(5)}(M) = \begin{cases} \frac{1}{2}(1 \pm \text{sign}(M)) & \text{if } |M| \geq 1, \\ \mathcal{M}^{\pm}_{(2)}(M)\left[(\pm 2 - M) \mp 16\tilde{\alpha} M \mathcal{M}^{\mp}_{(2)}(M)\right] & \text{otherwise.} \end{cases} \tag{5.74}$$

The interface pressure-driven flux is then given by

$$p_{1/2} = \mathcal{P}^{+}_{(5)}(M_L)p_L + \mathcal{P}^{-}_{(5)}(M_R)p_R - K_u \mathcal{P}^{+}_{(5)}(M_L)\mathcal{P}^{-}_{(5)}(M_R)(\rho_L + \rho_R)a_{1/2}(u_R - u_L). \tag{5.75}$$

Combining (5.71)-(5.73) and (5.75), we obtain that the total numerical flux across the interface is then given by

$$\mathbf{F}^* = \left[\dot{m}_{1/2} \begin{cases} \mathbf{q}_L/\rho_L & \text{if } \dot{m}_{1/2} > 0, \\ \mathbf{q}_R/\rho_R & \text{otherwise} \end{cases}\right] + \mathbf{p}_{1/2}, \tag{5.76}$$

with

$$\mathbf{q}_L = \begin{pmatrix} \rho_L \\ (\rho u)_L \\ (\rho v)_L \\ (\rho w)_L \\ (\rho\theta)_L \end{pmatrix}, \quad \mathbf{q}_R = \begin{pmatrix} \rho_R \\ (\rho u)_R \\ (\rho v)_R \\ (\rho w)_R \\ (\rho\theta)_R \end{pmatrix}, \quad \mathbf{p}_{1/2} = \begin{pmatrix} 0 \\ p_{1/2} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{5.77}$$

Several free parameters are available in this scheme. For simplicity, we follow *Liou* (2006) by choosing

$$\tilde{\alpha} = \tfrac{3}{16}, \quad \tilde{\beta} = \tfrac{1}{8}, \quad K_u = \tfrac{3}{4}, \quad K_p = \tfrac{1}{4}, \quad \sigma = 1. \tag{5.78}$$

5.4.3 Modified AUSM$^+$-up Riemann solver

The AUSM$^+$ solver can be simplified dramatically for approximately smooth flows with $M \ll 1$. This simplification may be desirable to improve the computational performance of the method, and significantly reduce the complexity of the analytic Jacobian needed in the implicit step. This modified Riemann solver is then applied in the vertical, where computing the Jacobian requires multiple evaluations of the Riemann solver at each time step. To begin, we assume that the difference between the left and right states is small relative to the magnitude of the hydrostatic background. In this case, the speed of sound at the interface, which primarily comes into play in the diffusion terms, can be approximated as

$$a_{1/2} = \sqrt{\frac{\gamma \bar{p}_{1/2}}{\bar{\rho}_{1/2}}}, \tag{5.79}$$

where the pressure and density field are taken to be the quantities defined by the hydrostatic background. Further, by making use of (5.5) the diffusive term in (5.71), which involves the pressure difference across the interface, can be approximated as

$$\begin{aligned} p_R - p_L &= p_0 \left[\frac{R_d}{p_0} \left((\overline{\rho\theta})_{1/2} + (\rho\theta)'_R \right) \right]^{c_p/c_v} - p_0 \left[\frac{R_d}{p_0} \left((\overline{\rho\theta})_{1/2} + (\rho\theta)'_L \right) \right]^{c_p/c_v}, \\ &\approx \frac{p_0 c_p}{c_v} (\overline{\rho\theta})^{c_p/c_v - 1} \left[(\rho\theta)'_R - (\rho\theta)'_L \right], \\ &= a_{1/2}^2 \left[(\rho\theta)'_R - (\rho\theta)'_L \right]. \end{aligned}$$

The interfacial Mach number then takes the form

$$M_{1/2} = \frac{\rho_L u_L + \rho_R u_R}{a_{1/2}(\rho_L + \rho_R)} - \frac{K_p}{(\overline{\rho\theta})_{1/2}} ((\rho\theta)'_R - (\rho\theta)'_L). \tag{5.80}$$

Similarly, in the limit of small Mach number, the interfacial pressure term (5.75) takes the simplified form

$$p_{1/2} = p_0 \left(\frac{R_d}{p_0} \cdot \frac{(\rho\theta)_L + (\rho\theta)_R}{2} \right)^{c_p/c_v} - \frac{K_u a_{1/2}}{2} (\rho_R u_R - \rho_L u_L). \tag{5.81}$$

The flux calculation in this case otherwise follows (5.73) and (5.76).

5.5 Numerical Results

In this section we present a selection of numerical results in order to verify the convergence and accuracy properties of the schemes discussed in this chapter. In section 5.5.1 we look at a rising thermal bubble in order to verify that our scheme is consistent with other models, and to show the effect of the first-, second- and third-order accurate time stepping schemes. In section 5.5.2, we extend a linear hydrostatic mountain test case to a large horizontal extent so as to verify stability and accuracy of our methods, even for a large horizontal-vertical aspect ratio and horizontal CFL number near 1.0. The problem of geostrophically balanced flow in a channel is studied in section 5.5.3, again using a large horizontal-vertical aspect ratio. This test case is used to verify fourth-order horizontal convergence of our numerical method. Finally, in section 5.5.4 we look at the behavior of a baroclinic instability in a channel. These tests further evaluate our scheme on a wide range of possible scales, ranging from the microscale with the rising thermal bubble test to the global scale with the baroclinic instability.

5.5.1 Rising Thermal Bubble

The 2D (x, z) rising thermal bubble test case is essentially ubiquitous in the study of non-hydrostatic mesoscale models. This test follows the evolution of a warm bubble in a constant potential temperature environment. The warm bubble leads to a positive perturbation in the vertical velocity field, which acts to carry the bubble upward. As the bubble moves upwards shearing quickly deforms the circular bubble into a mushroom cloud. Here we follow the initialization procedure described by *Giraldo and Restelli* (2008), which is a variation of the bubble experiments of *Robert* (1993).

The background consists of a constant potential temperature field $\overline{\theta} = 300$ K, with a small perturbation of the form

$$\theta' = \begin{cases} 0 & \text{for } r > r_c, \\ \frac{\theta_c}{2}\left[1 + \cos\left(\frac{\pi r}{r_c}\right)\right] & \text{for } r \leq r_c \end{cases}, \tag{5.82}$$

where

$$r = \sqrt{(x - x_c)^2 + (z - z_c)^2}. \tag{5.83}$$

Here we choose the amplitude and radius of the perturbation to be $\theta_c = 0.5$ K and $r_c = 250$ m, respectively. The domain consists of a square region $(x, z) \in [-500, 500] \times [0, 1000]$ m with $t \in [0, 700]$s. The center-point of the bubble is located at $x_c = 500$ m and $z_c = 350$ m. The boundary conditions are no-flux along all boundaries.

We plot the potential temperature perturbation for the crude, Strang-Carryover and ARS(2,3,3) scheme in Figs. 5.1, 5.2 and 5.3 at four resolutions ($\Delta x = 20$ m, 10 m, 5 m and 2.5 m) after 700 s. The crude scheme (Fig. 5.1) performs exceptionally poorly, unable to even resolve the correct convection velocity at low resolutions or the anticipated winding of the bubble's leading edges at higher resolutions. Significant improvement can be seen in the Strang-Carryover scheme (Fig. 5.2), which is impressive as it essentially requires the same number of calculations per iteration as the crude scheme. Further, we see a rough convergence of the shape of the bubble at increasing resolution. The ARS(2,3,3) scheme (Fig. 5.3) improves these results even further, as we see sharper gradients and increased winding along the tail of the bubble. For both the Strang-Carryover scheme and ARS(2,3,3) scheme, our solutions match the results reported in *Giraldo and Restelli* (2008) very closely.

5.5.2 Wide Hydrostatic Mountain

The linear 2D (x, z) horizontal-vertical hydrostatic mountain wave test case is another very common test within the mesoscale modeling community. This test

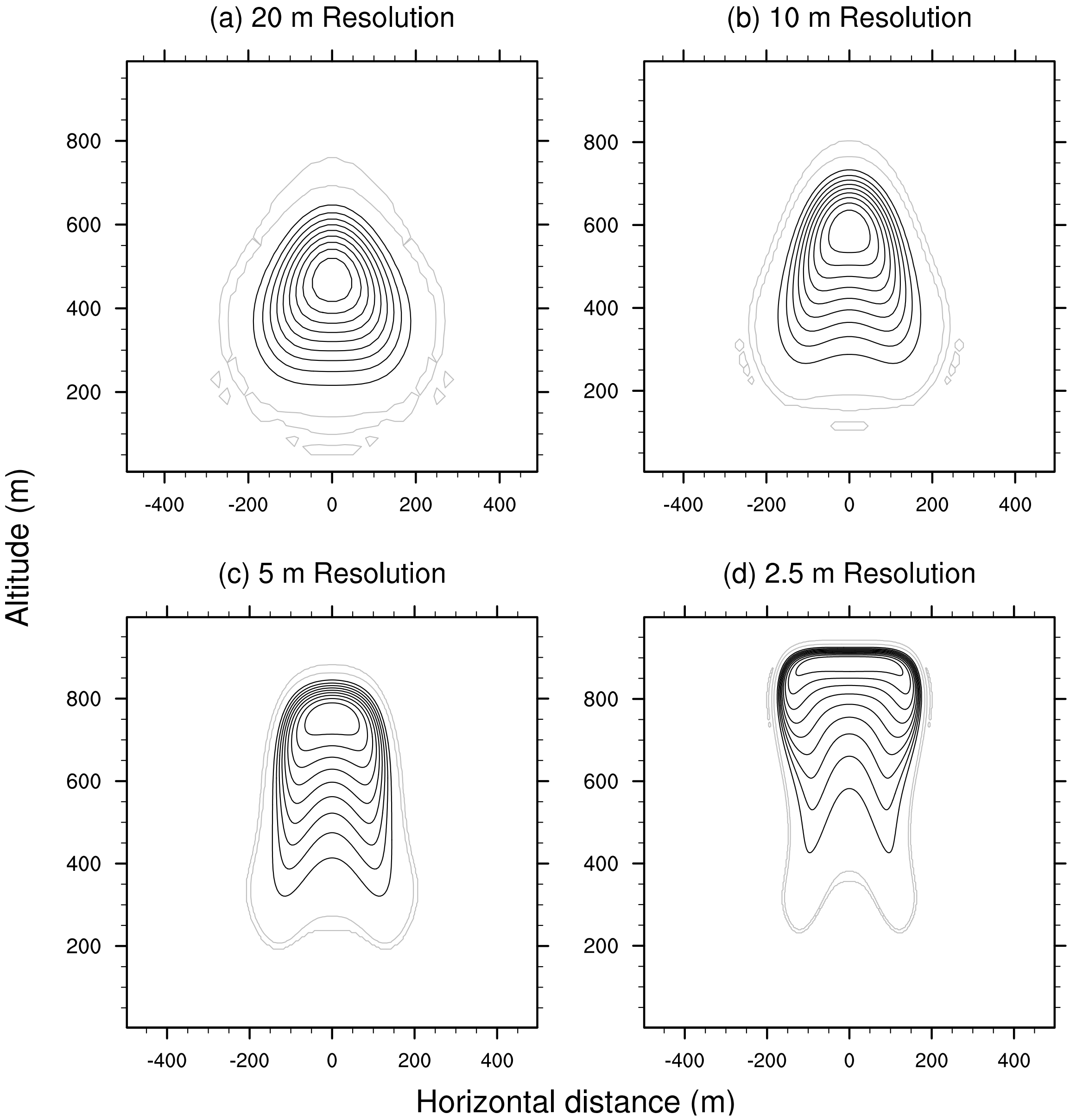


Figure 5.1: Plots of the potential temperature perturbation for the rising thermal bubble test case with crude splitting at time $t = 700$ s at four choices of resolution. The time step is chosen to be 0.06 s. Contour lines are from 300 K to 300.5 K with a contour interval of 0.05 K. The 300 K contour line is shown in light gray to emphasize oscillations due to the Gibbs' phenomenon.

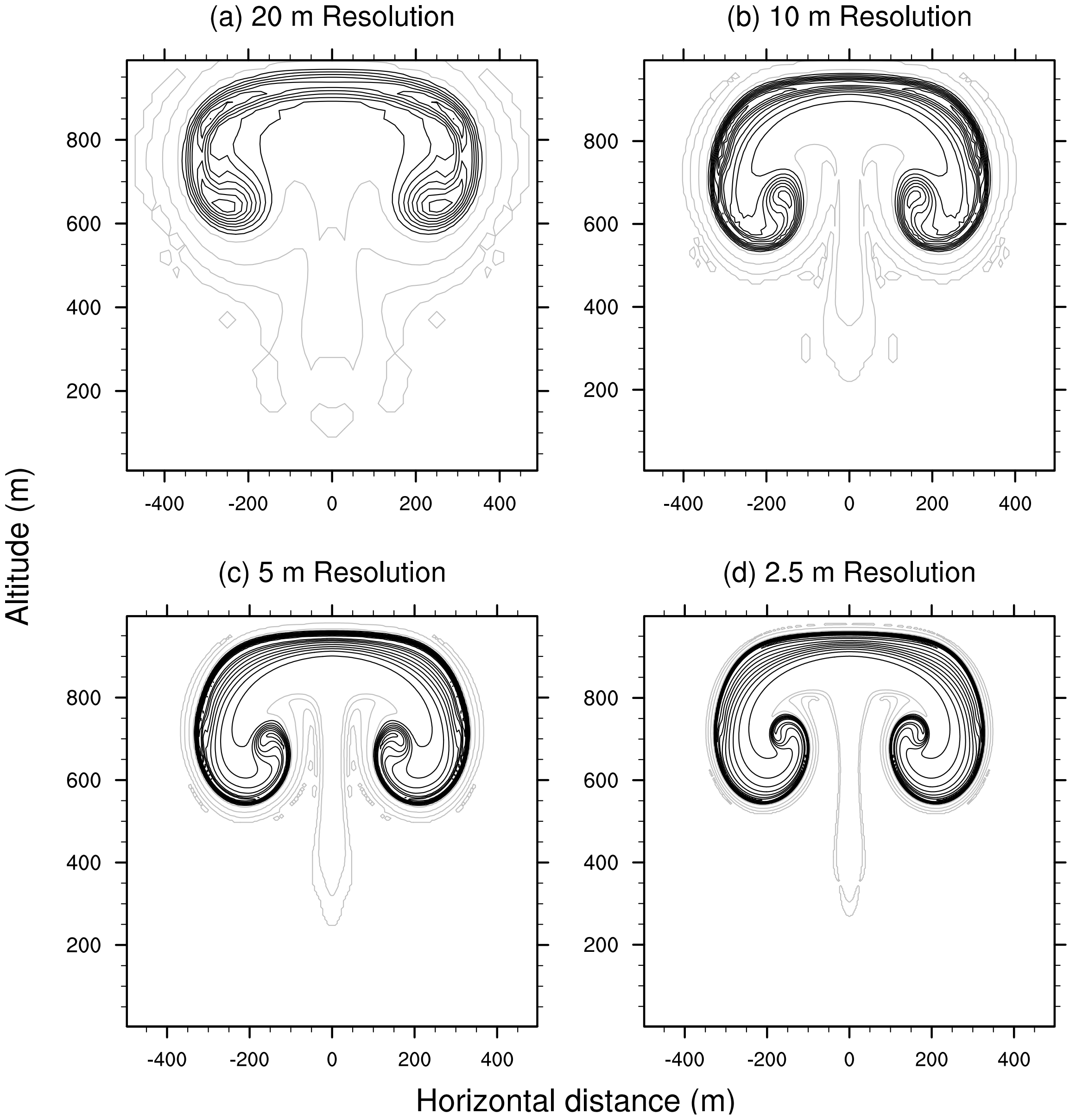


Figure 5.2: As Fig. 5.1 except with Strang-Carryover splitting.

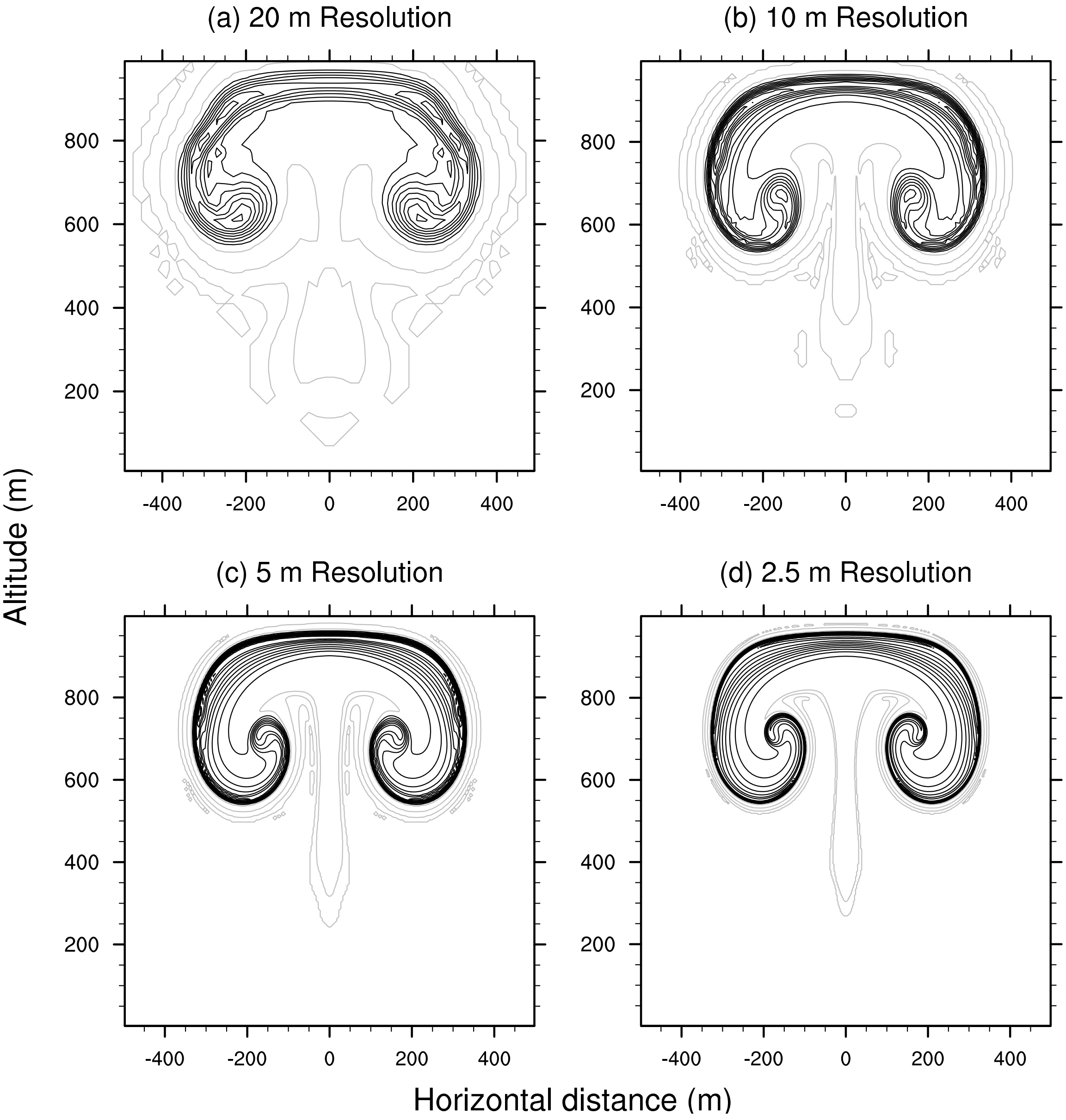


Figure 5.3: As Fig. 5.1 except with ARS(2,3,3) splitting.

case consists of a steady-state solution of linear hydrostatic flow over a single-peaked mountain with constant inflow and outflow boundary conditions. The background consists of an isothermal atmosphere with $\overline{T} = 250$ K, which implies a constant Brunt-Väisälä frequency of the form $\mathcal{N} = g/\sqrt{c_p\overline{T}}$. A constant mean flow $\overline{u} = 20$ m s^{-1} is then imposed. Topography is added in the form of a *witch of Agnesi* mountain,

$$h(x) = \frac{h_c}{1 + \left(\frac{x-x_c}{a_c}\right)^2}, \tag{5.84}$$

where the maximum height is $h_c = 1$ m, the center position is $x_c = 0$ m and the mountain half-width a_c is varied. For all values of a_c we will consider, it can be verified that $\mathcal{N}a_c/\overline{u} > 1$, so the flow is in the hydrostatic range.

Sponge-layer boundary conditions are imposed along the model top and outflow boundary by adding Rayleigh damping to the momentum and potential temperature evolution equations. This damping takes the form

$$\frac{\partial q}{\partial t} = -\tau(\overline{\overline{q}} - \overline{\overline{q}}_b), \tag{5.85}$$

where $\tau = \tau(x,z)$ is the inverse timescale of the damping, $q \in \{\rho u, \rho v, \rho w, (\rho\theta)'\}$, $(\rho u)_b = \overline{\rho} \times (20 \text{ m s}^{-1})$, $(\rho v)_b = 0$, $(\rho w)_b = 0$ and $(\rho\theta)'_b = 0$. Here $\tau = \tau(x,z)$ denotes the local strength of the Rayleigh damping. Damping is applied over a finite interval in the interior of the domain, spanning the range $[s_0 - s_T, s_0]$ where $s \in \{x, z\}$, s_0 denotes the location of the boundary and s_T denotes the thickness of the damping layer. The strength of the damping within the damping layer is determined by τ which, following *Giraldo and Restelli* (2008), is defined as

$$\tau(s) = \begin{cases} 0 & \text{if } s < s_0 - s_T, \\ \tau_0\left(\frac{s-(s_0-s_T)}{s_T}\right)^4 & \text{otherwise,} \end{cases} \tag{5.86}$$

where $\tau_0 = 2.0 \times 10^{-2}$ s^{-1}. If two boundary layers coincide, as with the outflow lateral boundary and the upper boundary, the strength of the Rayleigh friction is taken to be the maximum of the two coefficients. The layer thickness is taken to be $z_T = 18$ km in the vertical and $x_T = 2 \times a_c$ in the horizontal.

The analytic solution is known in this case, having been derived by *Alaka* (1960) and *Durran and Klemp* (1982). The steady state displacement of streamlines is given by

$$\delta(x, z) = \left(\frac{\overline{\rho}}{\rho_0}\right)^{-1/2} h_c a_c \frac{a_c \cos(\ell z) - x \sin(\ell z)}{x^2 + a_c^2}, \tag{5.87}$$

where $\overline{\rho}$ is the hydrostatic background density defined by (5.9), ρ_0 is the density of air at the surface, and ℓ is the Scorer parameter for an isothermal atmosphere defined by[1]

$$\ell^2 = \frac{g^2}{c_p \overline{T}\,\overline{u}^2} - \frac{g^2}{4R^2\overline{T}^2}. \tag{5.88}$$

The horizontal velocity is then given by

$$u = \overline{u}\left(1 - \frac{\partial \delta}{\partial z}\right), \tag{5.89}$$

and the vertical velocity by

$$w = \overline{u}\frac{\partial \delta}{\partial x}. \tag{5.90}$$

We plot the results of this test case with $a_c = 10$ km, $a_c = 100$ km and $a_c = 1000$ km in Figs. 5.4, 5.5 and 5.6 using the ARS(2,3,3) splitting. The figures show contours of the steady-state horizontal and vertical velocity perturbations as computed from the simulation (solid lines) and velocities obtained from the analytic procedure described above (dashed lines). The simulation domain is taken to be $(x, z) \in [-12a_c, 12a_c] \times [0 \text{ km}, 30 \text{ km}]$ and consists of 200 equally spaced horizontal elements and 125 equally spaced vertical elements, leading to a horizontal grid spacing

[1] Note that *Keller* (1994) shows the full formulation of this parameter for a general atmosphere.

(Δx) of 120 m, 1200 m and 12000 m and vertical grid spacing (Δz) of 240 m. Since the time required to reach an approximate steady state is proportional to a_c we also scale the final simulation time to be 10 h, 100 h and 1000 h for each of the three test cases. The time step is taken to be proportional to the simulation time, so that each simulation requires roughly equal computational time.

The results of these simulations show that the RKR procedure is stable for even very large aspect ratios (up to $\Delta x/\Delta z = 500$), and consistently produces accurate results for all three cases. In particular, the down-slope vertical velocity in each of the three cases seems to be predicted particularly well by our scheme. The deviation of the horizontal velocity contours from the analytic solution seems to match well with *Giraldo and Restelli* (2008) for the simulations with the smallest choice of a_c, suggesting that the Rayleigh damping may be consistently interfering with the computed solution.

5.5.3 Steady-state Geostrophically Balanced Flow in a Channel

The flow field for steady-state geostrophically balanced flow in a channel is based on a new test case defined by *Jablonowski et al.* (2011). The domain is a channel of dimensions $L_x \times L_y \times L_z$ with periodic boundaries in the x direction and no-flux conditions at all other interfaces. In this case we choose $L_x = 40000$ km, $L_y = 6000$ km and $L_z = 30$ km. The initial flow is comprised of a zonally-symmetric mid-latitudinal jet, defined in terms of vertical pressure-based η coordinates (see Appendix F) as

$$u(x, y, \eta) = -u_0 \sin^2\left(\frac{\pi y}{L_y}\right) \ln \eta \exp\left[-\left(\frac{\ln \eta}{b}\right)^2\right], \tag{5.91}$$

so that the wind is zero at the surface and along the y-boundary. The vertical half-width is set to $b = 2$ and u_0 is chosen to be 35 m s^{-1}. The meridional wind velocity v and vertical wind velocity w are both set to zero. We take the surface pressure to

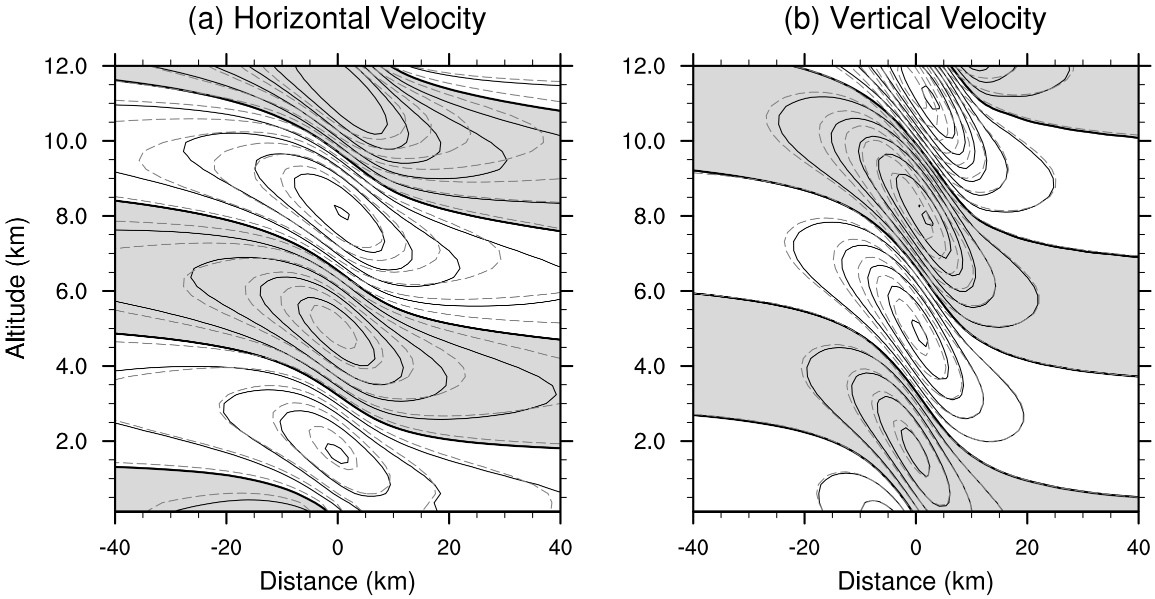


Figure 5.4: Plots of horizontal velocity perturbation (left) and vertical velocity (right) for the linear hydrostatic mountain test case with $a_c = 10$ km and ARS(2,3,3) splitting. Grid spacing is taken to be 1200 m in the horizontal and 240 m in the vertical. The simulation is run up to $t = 10$ h with a time step of 2.5 s. Contour lines in the horizontal velocity perturbation plot are from -0.025 m s^{-1} to 0.025 m s^{-1} with a contour interval of 0.005 m s^{-1}. Contours in the vertical velocity plot are from -0.005 m s^{-1} to 0.005 m s^{-1} with a contour interval of 5×10^{-4} m s^{-1}. Negative values are indicated by shaded regions. The exact solution from linear analysis is plotted as gray dashed lines.

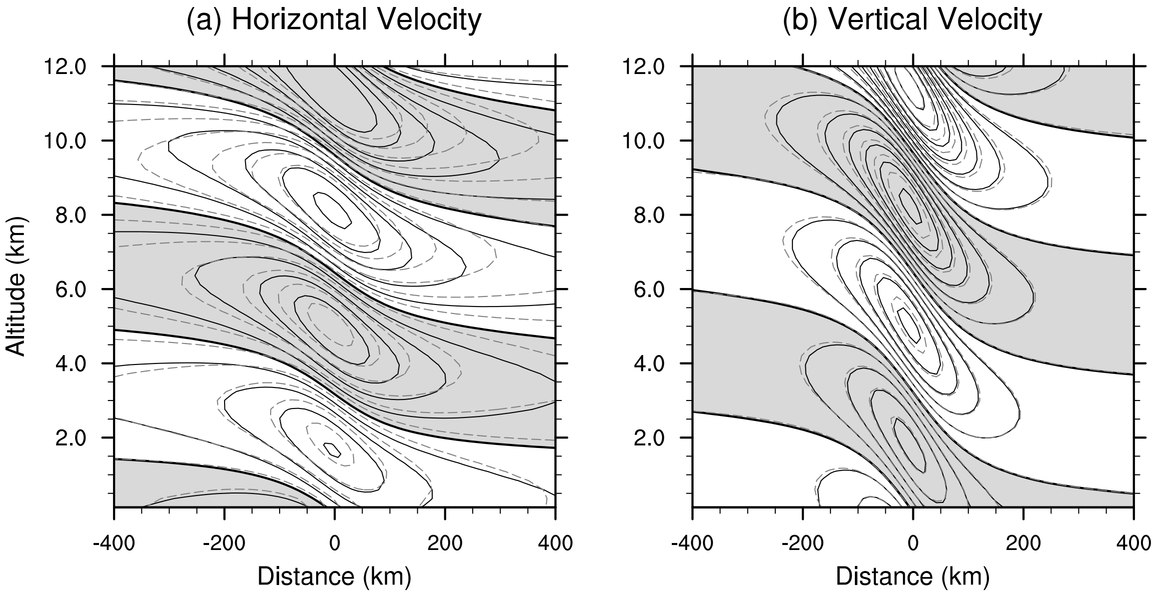


Figure 5.5: As Fig. 5.4 except with $a_c = 100$ km and a horizontal grid spacing of 12000 m. The simulation is run up to $t = 100$ h with a time step of 25 s. The vertical velocity contours are from -5×10^{-4} m s^{-1} to 5×10^{-4} m s^{-1} with a contour interval of 5×10^{-5} m s^{-1}.

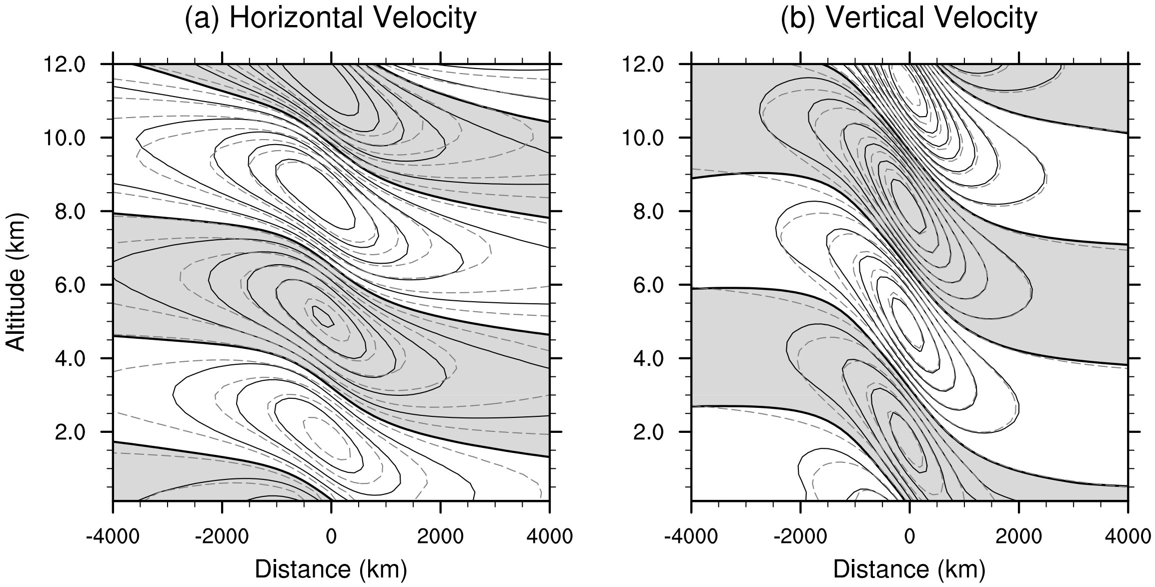


Figure 5.6: As Fig. 5.4 except with $a_c = 1000$ km and a horizontal grid spacing of 120000 m. The simulation is run up to $t = 1000$ h with a time step of 250 s. The vertical velocity contours are from -5×10^{-5} m s^{-1} to 5×10^{-5} m s^{-1} with a contour interval of 5×10^{-6} m s^{-1}.

be constant with $p_s = p_0 = 10^5$ Pa. This formulation can either be on an f-plane or β-plane, which have Coriolis parameters

$$f = f_0, \quad \text{and} \quad \beta = f_0 + \beta_0(y - y_0), \tag{5.92}$$

respectively, where $f_0 = 2\Omega \sin\varphi_0$ and $\beta_0 = 2a^{-1}\Omega\cos\varphi_0$ at latitude $\varphi_0 = 45°$N. Here, the radius of the Earth is $a = 6371.229 \times 10^3$ m, its angular velocity is $\Omega = 7.292 \times 10^{-5}$ s^{-1} and $y_0 = L_y/2$ is the center point of the domain in the y-direction. The background geopotential field is again defined in terms of η coordinates is

$$\Phi(x, y, \eta) = \langle\Phi(\eta)\rangle + \Phi'(x, y)\ln\eta \exp\left[-\left(\frac{\ln\eta}{b}\right)^2\right], \tag{5.93}$$

with the horizontal-mean geopotential

$$\langle\Phi(\eta)\rangle = \frac{T_0 g}{\Gamma}\left(1 - \eta^{\frac{R_d\Gamma}{g}}\right) \tag{5.94}$$

and variation

$$\begin{aligned}\Phi'(x, y) \quad = \quad & \frac{u_0}{2}\left\{(f_0 - \beta_0 y_0)\left[y - \frac{L_y}{2} - \frac{L_y}{2\pi}\sin\left(\frac{2\pi y}{L_y}\right)\right]\right. \\ & \left. + \frac{\beta_0}{2}\left[y^2 - \frac{L_y y}{\pi}\sin\left(\frac{2\pi y}{L_y}\right) - \frac{L_y^2}{2\pi^2}\cos\left(\frac{2\pi y}{L_y}\right) - \frac{L_y^2}{3} - \frac{L_y^2}{2\pi^2}\right]\right\}\end{aligned} \tag{5.95}$$

The reference temperature is $T_0 = 288$ K and the lapse rate is chosen to be $\Gamma = 0.005$ K m^{-1}. The corresponding temperature distribution is given by

$$T(x, y, \eta) = \langle T(\eta)\rangle + \frac{\Phi'(x, y)}{R_d}\left(\frac{2}{b^2}(\ln\eta)^2 - 1\right)\exp\left[-\left(\frac{\ln\eta}{b}\right)^2\right], \tag{5.96}$$

with horizontal mean temperature

$$\langle T(\eta)\rangle = T_0 \eta^{\frac{R_d\Gamma}{g}}. \tag{5.97}$$

This test considers the steady-state problem, where the solution is the initial state. Hence, error measures are calculated in the height field via the usual global error norms,

$$L_1(q) = \frac{I\left[|q - q_T|\right]}{I\left[|q_T|\right]}, \tag{5.98}$$

$$L_2(q) = \sqrt{\frac{I\left[(q - q_T)^2\right]}{I\left[q_T^2\right]}}, \tag{5.99}$$

$$L_\infty(q) = \frac{\max|q - q_T|}{\max|q_T|}, \tag{5.100}$$

where q_T is the field at the initial time and I denotes an approximation to the global integral, given by

$$I[x] = \sum_{\text{all cells } k} x_k V_k, \tag{5.101}$$

with V_k denoting the volume of element k.

Error norms are given in Table 5.2, for an f-plane approximation, and 5.3 for the β-plane approximation. The simulations are run with a variable horizontal resolution of 400, 200, 100 and 50 km, and a uniform vertical resolution of 1 km (30 equally spaced vertical levels). We observe convergence that is slightly less than fourth-order for the vertical momentum field and slightly better than fourth-order for the potential temperature field. Since hydrostatic balance is guaranteed by the background splitting technique described in section 5.2, errors are only accumulated due to an imbalance in the geostrophically balanced components, and hence increasing the number of vertical levels does not have a significant impact on the error norms. Discrepancies in these errors at each vertical level trigger the slight imbalances in the vertical velocity. Since this is a steady test case, the errors in this analysis are dominated by errors in the spatial reconstruction, and so similar error norms are observed with both the crude and ARS(2,3,3) time stepping schemes.

Table 5.2: Relative errors in the vertical momentum field ρw and potential temperature density field $\rho\theta$ for the geostrophically balanced flow in a channel test with an f-plane approximation and Strang-Carryover time stepping scheme. A convergence study is performed by varying the horizontal resolution. The computed order of accuracy is obtained from a least squares fit through the data.

Z momentum field ρw

Resolution	L_1 error	L_2 error	L_∞ error
400 km	1.8215215494(−8)	5.9321531434(−8)	4.2483043440(−7)
200 km	1.8193260780(−9)	6.0542924949(−9)	5.1317349168(−8)
100 km	1.2389765549(−10)	4.1249016466(−10)	3.4836674459(−10)
50 km	7.9165590539(−12)	2.6370966078(−11)	2.3812090172(−10)
Order	3.738	3.728	3.628

Potential temperature density field $\rho\theta$

Resolution	L_1 error	L_2 error	L_∞ error
400 km	4.1142264966(−6)	1.2130186602(−5)	6.6828625279(−5)
200 km	2.0397544359(−7)	5.8405303319(−7)	3.0527850754(−6)
100 km	1.2454071640(−8)	3.4742317298(−8)	1.7254831164(−7)
50 km	7.7983830011(−10)	2.1731785969(−9)	1.1131078281(−8)
Order	4.113	4.141	4.180

Table 5.3: As Table 5.2, except for the β-plane approximation.

Z momentum field ρw

Resolution	L_1 error	L_2 error	L_∞ error
400 km	1.4898989124(−8)	4.7631787283(−8)	4.4735011107(−7)
200 km	1.7994170122(−9)	5.0245100400(−9)	3.4963904410(−8)
100 km	1.3567903809(−10)	3.8515503135(−10)	2.9537028243(−9)
50 km	9.1251900583(−12)	2.6590499326(−11)	2.0542081087(−10)
Order	3.575	3.613	3.683

Potential temperature density field $\rho\theta$

Resolution	L_1 error	L_2 error	L_∞ error
400 km	1.3395442832(−5)	3.9126986059(−5)	2.1560957481(−4)
200 km	8.8035461528(−7)	2.6018038411(−6)	1.5701255563(−5)
100 km	5.5255480267(−8)	1.6427703245(−7)	1.0152684240(−6)
50 km	3.4569858976(−9)	1.0290700778(−8)	6.3671677708(−8)
Order	3.975	3.966	3.913

5.5.4 Baroclinic Instability in a Channel

This test case uses a geostrophically balanced background identical to the one described in section 5.5.3. However, we additionally introduce a confined perturbation in the zonal wind field of the form

$$u'(x, y, \eta) = u_p \exp\left[-\left(\frac{(x - x_c)^2 + (y - y_c)^2}{L_p^2}\right)\right], \tag{5.102}$$

with radius $L_p = 600$ km, maximum amplitude $u_p = 1\text{m s}^{-1}$ and center-point $(x_c, y_c) = (2000 \text{ km}, 2500 \text{ km})$. This perturbation is superimposed on the zonal wind field (5.91) so that the total zonal wind field reads

$$u_{new}(x, y, \eta) = u(x, y, \eta) + u'(x, y, \eta). \tag{5.103}$$

The setup resembles the baroclinic wave experiments on the sphere suggested by

Jablonowski and Williamson (2006). The unbalanced perturbation acts as a trigger for baroclinic waves that grow explosively over a 10-12 day simulation period. Such a flow is characteristic for the mid-latitudes. The channel test thereby assesses how well the finite volume scheme simulates large-scale flow fields with large aspect ratios. All simulations are run with the ARS(2,3,3) scheme and utilize a 100 km horizontal grid spacing with 30 equally-spaced vertical levels and a model top at 30 km.

Snapshots of the simulation for the f-plane approximation at day 12 are plotted in Fig. 5.7. The figure depicts the horizontal cross sections of the pressure, temperature and relative vorticity at 500 m. This vertical position corresponds to the height of the lowermost model level. Figure 5.7 shows that the baroclinic wave has almost broken which takes place around day 13.5. The flow has formed distinct low and high pressure systems that are associated with sharp temperature fronts and sharp gradients in the relative vorticity field.

The corresponding simulation results with the β-plane approximation are plotted in Fig. 5.8. Here we show the identical fields, but now at day 10 before wave breaking events set in. The presence of the planetary vorticity gradient has sped up the evolution of the baroclinic wave. Again, the low and high pressure systems are connected to sharp frontal zones in the temperature and vorticity fields which resemble realistic flow conditions. It is interesting to note that the β-plane simulation leads to a more confined flow field that has not spread to the northern and southern edges of the domain by day 10. These differences between the f-plane and β-plane simulations will be discussed in greater detail in *Jablonowski et al.* (2011). Here, the main focus of this test is to demonstrate that the ARS(2,3,3) scheme reliably simulates the evolution of atmospheric flow fields that are relevant for the large (mid-latitudinal) portion of global atmospheric General Circulation Models.

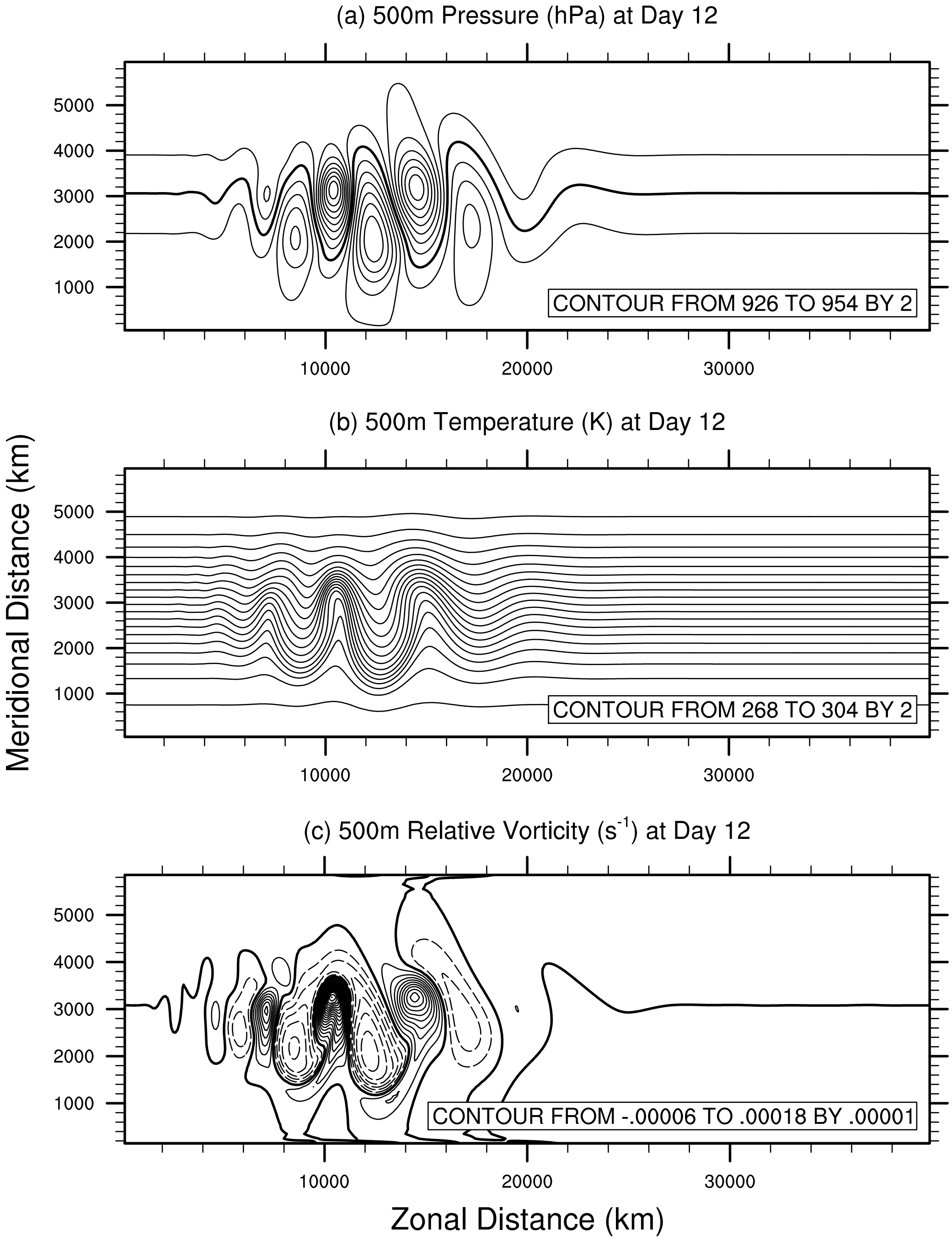


Figure 5.7: Simulation results from the baroclinic instability in a channel computed at day 12 using the ARS(2,3,3) scheme with the f-plane approximation. The simulation is run at a horizontal resolution of 100 km and a vertical resolution of 1 km with a time step of 1200 s. Contour lines are as indicated on each plot. The 942 hPa line is enhanced in the pressure plot. The zero line in the relative vorticity plot is enhanced and negative values are plotted using dashed lines.

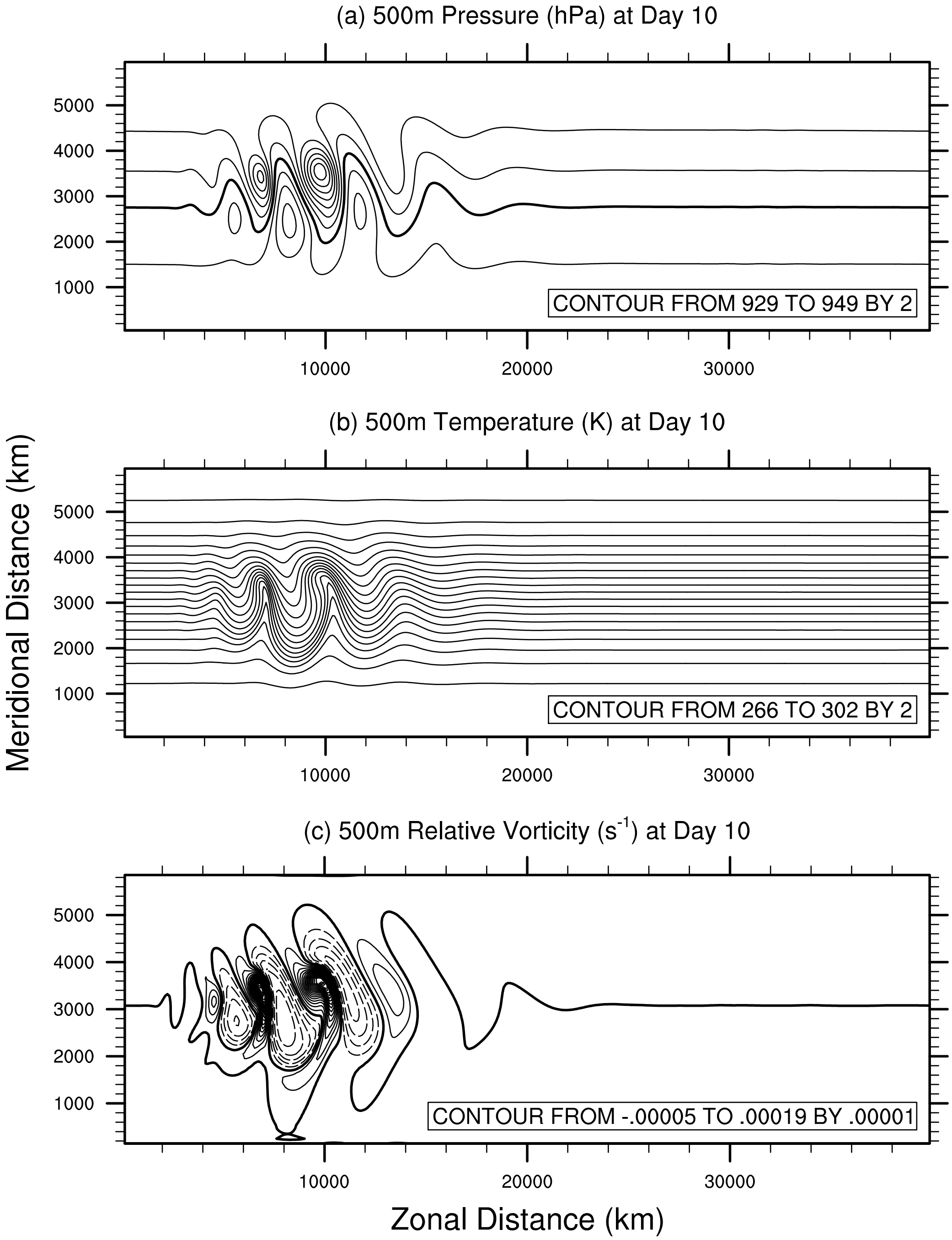


Figure 5.8: Simulation results from the baroclinic instability in a channel computed at day 10 using the ARS(2,3,3) scheme with the β-plane approximation. The simulation is run at a horizontal resolution of 100 km and a vertical resolution of 1 km. Contour lines are as indicated on each plot. The 943 hPa line is enhanced in the pressure plot. The zero line in the relative vorticity plot is enhanced and negative values are plotted using dashed lines.

5.6 Conclusions

In this chapter we have presented a vertical-horizontal splitting strategy for the full non-hydrostatic Euler equations based on Runge-Kutta-Rosenbrock (RKR) time integration schemes. For atmospheric problems where the vertical grid spacing is usually much smaller than the horizontal, this strategy allows us to simulate the full Euler equations while only constraining the time step by the horizontal grid spacing. This approach is only slightly slower than the corresponding hydrostatic approach, but remains valid and consistent at all scales. We have presented a first-order scheme based on a crude splitting, a second-order scheme that uses a Strang-splitting and carryover strategy and a third-order scheme based on an approach attributed to *Ascher et al.* (1997). The first-order scheme is shown to be highly diffusive for thermal bubble experiments, and shows no benefit over the Strang-carryover scheme, which requires the same number of explicit and implicit steps per time step. The ARS(2,3,3) scheme shows a mild improvement over the Strang-carryover approach, but requires two implicit steps per time step. However, the third-order accuracy in time this scheme affords may be desirable.

Numerical results have shown our approach to be accurate, stable and applicable to a range of atmospheric flows and horizontal-vertical aspect ratios. By using a fourth-order polynomial reconstruction in the horizontal we observe clear fourth-order convergence in the horizontal. Horizontal-vertical aspect ratios up to $500 : 1$ have been tested under our scheme and verified to be stable up to a horizontal CFL number of 1.0.

As a consequence of the results in this chapter, we believe that our horizontal-vertical dimension splitting strategy is a promising option for any high-order finite-volume or discontinuous Galerkin based method.

CHAPTER VI

MCore: A Non-hydrostatic Atmospheric Dynamical Core Utilizing High-Order Finite-Volume Methods

6.1 Introduction

In recent years, the exponential growth of computing power and trend towards massive parallelization of computing systems has had a profound influence on the atmospheric modeling community. Atmospheric cloud-resolving models are now pushing towards scales of only a few kilometers, meanwhile utilizing thousands to hundreds of thousands of processors. At these small scales many of the approximations that have been previously used in developing dynamical cores are no longer valid. As a consequence, there has been a trend towards developing atmospheric models which incorporate the full unapproximated hydrodynamic equations of motion. These developments have required a substantial paradigm shift in the way developers think about the algorithms and software behind geophysical models. Many design decisions that worked well in the past are no longer acceptable on large parallel systems, and so modifications must be made to accommodate this new generation of hardware. As a consequence, the past ten years have seen substantial innovation in the modeling community as they push forward with efforts to determine the best candidates for the next-generation of atmospheric models.

Our focus in this chapter is on non-hydrostatic modeling: That is, we are interested in models which treat the vertical velocity as a prognostic variable. In this case the vertical velocity has its own evolution equation and is not diagnosed from the other flow variables. Non-hydrostatic models are valid on essentially any horizontal scale and so can be used in cloud-resolving simulations. Several non-hydrostatic models are now in use, having been largely developed in the past ten years in response to growing availability of computing power. These include the UK Met Office unified model (*Davies et al.*, 2005; *Staniforth and Wood*, 2008), the Non-hydrostatic ICosahedral Atmospheric Model (NICAM), which was developed by *Tomita and Satoh* (2004) in cooperation with the Center for Climate System Research (CCSR, Japan), the NOAA Non-hydrostatic Icosahedral Model (*Govett et al.*, 2010, NIM) and the Ocean-Land-Atmosphere model (OLAM) (*Walko and Avissar*, 2008). Recently, GFDL has also developed a non-hydrostatic dynamical core on the cubed-sphere (*Putman and Lin*, 2009) based on the work of *Putman and Lin* (2007). These models all make use of some sort of conservative finite-difference or finite-volume formulation to ensure conservation of mass and adopt the Arakawa C-grid staggering (*Arakawa and Lamb*, 1977). Other non-hydrostatic models include the ECMWF model IFS (*Wedi et al.*, 2010) which makes use of the spectral transform method, and the semi-Lagrangian Canadian GEM model (*Yeh et al.*, 2002).

In developing models for large-scale parallel computers, the choice of grid is of particular importance. Although non-hydrostatic dynamical cores have been developed on the regular latitude-latitude (RLL) grid, including the UK Met Office model, it is well known that the RLL grid suffers from convergence of grid lines at the north and south poles. As a consequence, models using the RLL grid require the use of polar filters to remove instabilities associated with small grid elements, which can in turn severely damage performance on parallel systems. Many recently developed hydrostatic and non-hydrostatic models have tended away from this grid, instead using quasi-uniform grids such as the icosahedral or cubed-sphere grids. Several hydrostatic models are now built on the icosahedral grid, including the German Weather Service GME model (*Majewski*, 1998; *Majewski et al.*, 2002) and model of *Ringler et al.* (2000). Non-hydrostatic models that use the icosahedral grid include NICAM, NIM and OLAM. The icosahedral grid has been shown to perform well on large parallel systems, but the non-Cartesian structure of the grid leads to difficulties in organizing the grid within memory. Another choice of quasi-uniform grid is the cubed-sphere grid, which was originally developed by *Sadourny* (1972) and revived by *Ronchi et al.* (1996). It was later used as the basis for a shallow-water model by *Rančić et al.* (1996). Since then, shallow-water models have been developed using the cubed-sphere grid that utilize finite-volume methods (*Rossmanith*, 2006; *Ullrich et al.*, 2010), multi-moment finite-volume (*Chen and Xiao*, 2008), the discontinuous Galerkin method (*Nair et al.*, 2005) and the spectral element method (*Taylor et al.*, 1997). The spectral element method was successfully extended to a full hydrostatic atmospheric model (the Spectral Element Atmosphere Model, SEAM) (*Fournier et al.*, 2004), which is part of the High-Order Method Modeling Environment (HOMME). HOMME incorporates both the spectral element and discontinuous Galerkin methods, and has proven to scale efficiently to hundreds of thousands of processors. More recently, the GFDL finite-volume dynamical core has been modified to use a cubed-sphere grid (*Putman and Lin*, 2007), and has been demonstrated to also be very effective at high resolutions.

This chapter continues a series that describes the development of an atmospheric model based on unstaggered high-order finite-volume methods. In *Ullrich et al.* (2010) a shallow-water model utilizing cell-centered third- and fourth-order finite-volume methods was described. This approach was demonstrated to be robust and highly competitive with existing methods when tested against the shallow-water test cases of *Williamson et al.* (1992). The high-order finite-volume method was later extended to non-hydrostatic simulations in Cartesian geometry in *Ullrich and Jablonowski* (2011a). Therein the authors demonstrated an accuracy-preserving technique for splitting horizontal and vertical motions using interleaved explicit and implicit time steps. The work of this chapter is a combination of *Ullrich et al.* (2010) and *Ullrich and Jablonowski* (2011a), and describes the high-order finite-volume formulation in spherical geometry. MCore is a fully featured dynamical core that provides support for both the shallow-water equations and the full non-hydrostatic fluid equations. However, our emphasis in this chapter will be on the non-hydrostatic dynamical core under the shallow-atmosphere approximation. MCore maintains fourth-order-accuracy in the horizontal and second-order accuracy in the vertical, and utilizes a fully Eulerian cell-centered finite-volume formulation that has been proven to be robust for problems from several fields.

The outline of this chapter is as follows. In section 6.2 we introduce the cubed-sphere grid, which underlies the MCore model. The non-hydrostatic fluid equations under the shallow-atmosphere approximation are introduced in section 6.3. The numerical approach underlying the MCore model is presented in section 6.4. Numerical results and test cases are described in section 6.5. Finally, our conclusions and future work are discussed in section 6.6. A list of variables used in this chapter can be found in Table 6.1. A list of the constants used in this chapter and their corresponding values can be found in Table 6.2. Throughout this chapter we will make use of Einstein summation notation, especially when describing geometric relations, under which summation is implied over repeated indices.

6.2 The Cubed-Sphere

The MCore model is implemented on a cubed-sphere grid, which can be imagined as the product of projecting a cube with regularly gridded faces onto the surface of a

Table 6.1: List of variables used in this chapter.

Constant	Description	Units
ρ	Density	kg / m^3
$\mathbf{u}$	Velocity vector	rad / s
θ	Potential temperature	K
p	Pressure	kg / m / s^2
N_c	Horizontal cubed-sphere resolution	-
N_r	Radial (vertical) resolution	-
n_p	Panel indicator ($n_p \in \{1,2,3,4,5,6\}$)	-
$\mathbf{g}_\alpha$	Basis vector in the α direction	-
$\mathbf{g}_\beta$	Basis vector in the β direction	-
$\mathbf{g}_r$	Basis vector in the r direction	-
G^{ij}	Contravariant metric tensor	-
G_{ij}	Covariant metric tensor	-
J	Metric Jacobian (square root of metric determinant)	m^2
$r_s(\alpha,\beta,n_p)$	Surface elevation	m
r_T	Model height cap	m

Table 6.2: List of physical constants used in this chapter.

Constant	Description	Control Value
a	Radius of the Earth	6.37122×10^6 m
ω	Rotational speed of the Earth	7.292×10^{-5} s^{-1}
g	Gravity	9.80616 m s^{-2}
p_0	Reference pressure	1000 hPa
c_p	Specific heat capacity of dry air at constant pressure	1004.5 J kg^{-1} K^{-1}
c_v	Specific heat capacity of dry air at constant volume	717.5 J kg^{-1} K^{-1}
R_d	Ideal gas constant of dry air	287.0 J kg^{-1} K^{-1}

sphere. The cubed-sphere grid was originally suggested by *Sadourny* (1972), but was not used in developing full geophysical codes until the work of *Ronchi et al.* (1996). There are several advantages to the cubed-sphere grid, such as grid regularity on each panel. Further, the cubed-sphere grid avoids the so-called "pole-problem," which refers to the issues associated with convergence of grid lines at the poles on a latitude-longitude projection. The delay between introduction and implementation of this grid was largely due to the fact that the overhead required in storing and computationally maintaining the grid did not outweigh its potential advantages. However, in an era where computational power is increasing exponentially and more computations must utilizes large parallel systems, the inherent regularity of the cubed-sphere grid makes it an attractive option for next-generation models.

Mathematically, the cubed-sphere grid consists of six panels that form the face of a cube projected onto the surface of a concentric sphere. Several options are available for tiling each cube panel (see, for instance, *Putman and Lin* (2007) for a review of the types of cubed-sphere grids). MCore makes use of the gnomonic (equiangular) cubed-sphere grid, which consists of grid lines that have equal central angles relative to the center of the sphere (this property is also held by lines of constant longitude on a latitude-longitude projection). This choice does not lead to a perfectly uniform tiling of the sphere; instead, as resolution increases the ratio of the area of the smallest grid element to largest grid element approaches $1/\sqrt{2} \approx 0.707$. Nonetheless, one powerful advantage of this choice is that grid lines parallel to panel edges are coincident across panels. A depiction of the cubed-sphere grid and panel edges are given in Figure 6.1.

Equiangular cubed-sphere coordinates are generally given in terms of the component vector (r, α, β, n_p), with $\alpha, \beta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ denoting the horizontal coordinate on each panel, r denoting the radial distance to the center of the sphere and $n_p \in \{1, 2, 3, 4, 5, 6\}$ denoting the panel index. By convention, we choose panels $1-4$ to be along the equator, with panels 5 and 6 centered on the northern and southern pole, re-

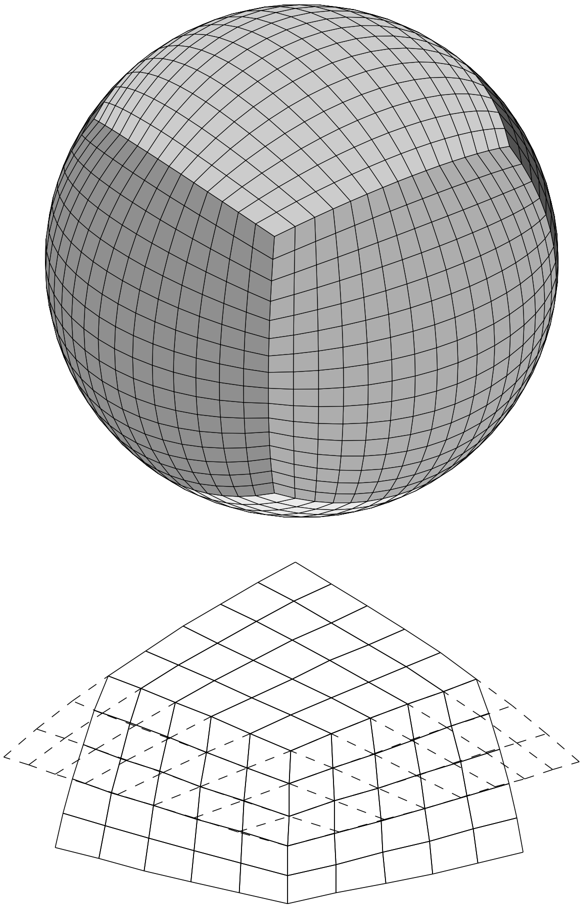


Figure 6.1: **Left:** A 3D view of the tiling of the cubed-sphere along surfaces of constant radius, shown here with a 16 × 16 tiling on each panel. **Right:** A close-up view of one of the cubed-sphere corners, also showing the "halo region" of the upper panel, which consists of elements which have been extended into neighboring panels (dashed line).

spectively. A closely related set of coordinates are gnomonic cubed-sphere coordinates (r, X, Y, n_p), which are related to equiangular coordinates via the transformation

$$X = \tan\alpha, \qquad Y = \tan\beta. \tag{6.1}$$

Gnomonic coordinates have the property that any straight line in a gnomonic projection forms the arc of a great circle. Geometric quantities are often more simply written in terms of gnomonic variables, and so we will use these two sets of coordinates interchangeably. In this chapter we will also make use of the definition

$$\delta = \sqrt{1 + X^2 + Y^2}, \tag{6.2}$$

which appears frequently in the calculation of metric quantities associated with the cubed-sphere.

Uniform grid elements on the equiangular cubed sphere form squares in the (α, β) plane, consisting of arcs of uniform angle $\Delta\alpha$. The discrete integer resolution on a uniform mesh is denoted by N_c, and related to $\Delta\alpha$ via

$$\Delta\alpha = \frac{\pi}{2N_c}. \tag{6.3}$$

Traditionally, the resolution of the cubed-sphere grid is compactly denoted by $c\langle N_c \rangle$. Grid spacing in the radial direction can be chosen arbitrarily, independent of the horizontal grid. A list of some properties of the cubed sphere grid is given in Table 6.3. The table lists the approximate equatorial spacing of grid elements, average area per element on the sphere, maximum area ratio and equivalent model resolutions of the regular latitude-longitude finite-volume and spectral transform models (under triangular truncation, as argued by *Williamson* (2008)). Details of the geometric terms used in this reconstruction are given in Appendix G.

Table 6.3: Properties of the cubed sphere grid for different resolutions. Here Δx is the grid spacing at the equator, A_{avg} is the average area of all cubed sphere grid elements, A_{min} is the minimum element area and A_{max} is the maximum element area. RLL_{equiv} denotes the equivalent grid spacing (in degrees) on the regular latitude-longitude grid with the same number of elements and T_{equiv} denotes the approximate triangular truncation of a spectral transform method.

Resolution	Δx	A_{avg}	A_{min}/A_{max}	RLL_{equiv}	T_{equiv}
$c20$	500 km	2.125×10^5 km^2	0.7359	5.2°	$T21$
$c40$	250 km	5.313×10^4 km^2	0.7213	2.6°	$T42$
$c80$	125 km	1.328×10^4 km^2	0.7141	1.3°	$T85$
$c90$	111 km	1.049×10^4 km^2	0.7133	1.15°	$T106$
$c160$	62.5 km	3.321×10^3 km^2	0.7106	0.65°	$T170$

6.3 The non-hydrostatic fluid equations in cubed-sphere coordinates

MCore utilizes the full non-hydrostatic fluid equations in terms of conserved variables density ρ, momentum $\rho\mathbf{u}$ (with 3D velocity vector $\mathbf{u}$) and potential temperature density $\rho\theta$ (with potential temperature θ). Further, MCore makes use of the shallow-atmosphere approximation, which is described in Appendix H. The differential form of the equations of motion can be written as follows:

$$\frac{\partial \rho}{\partial t} + \nabla_j(\rho u^j) = 0, \tag{6.4}$$

$$\frac{\partial \rho u^i}{\partial t} + \nabla_j(\rho u^i u^j + G^{ij} p) = -\rho g \mathbf{g}_r^i - f(\mathbf{g}_r \times (\rho \mathbf{u}))^i, \tag{6.5}$$

$$\frac{\partial \rho\theta}{\partial t} + \nabla_j(\rho\theta u^j) = 0. \tag{6.6}$$

Here G^{ij} denotes the contravariant metric, $\mathbf{g}_r$ is the basis vector in the radial direction, g is gravity and f is the Coriolis parameter. The divergence operator is denoted by $\mathsf{div}(\cdot)$. The pressure p in the momentum equation is related to the potential temperature density via the equation of state

$$p = p_0 \left(\frac{R_d(\rho\theta)}{p_0} \right)^{c_p/c_v}, \tag{6.7}$$

where $p_0 = 1000$ hPa is the reference pressure, R_d is the ideal gas constant for dry air and c_p and c_v denote the specific heat capacity of dry air at constant pressure and constant volume. The second terms on the left-hand-side of (6.4)-(6.6) are referred to as *flux terms*, since they determine the flow rate of the conservative state variables through the edges of a spatial region. The terms on the right-hand-side of these equations are *source terms.* Non-hydrostatic mesoscale models that use a closely related equation set include the WRF model (see, for example, *Skamarock and Klemp* (2008)) and the model by *Ahmad and Lindeman* (2007). Throughout this chapter we will refer to the vector of prognostic quantities as the state vector and denote it by $\mathbf{q} = (\rho, \rho\mathbf{u}, \rho\theta)$.

A splitting is performed on these equations of the form

$$\begin{aligned} \rho(\mathbf{x},t) &= \rho^h(\mathbf{x}) + \rho'(\mathbf{x},t), & (6.8)\\ p(\mathbf{x},t) &= p^h(\mathbf{x}) + p'(\mathbf{x},t), & (6.9)\\ (\rho\theta)(\mathbf{x},t) &= (\rho\theta)^h(\mathbf{x}) + (\rho\theta)'(\mathbf{x},t), & (6.10) \end{aligned}$$

where $\mathbf{x} = (r, \alpha, \beta, n_p)$ and the superscript h denotes fields which satisfy (6.7) and are in local hydrostatic balance, *i.e.*

$$\frac{\partial p^h}{\partial r} = -\rho^h g, \tag{6.11}$$

with mean acceleration due to gravity g. This choice is required to remove errors in approximating the hydrostatic state of the atmosphere that could be responsible for significant generation of spurious vertical momentum. The background state must be chosen from the space of hydrostatically balanced solutions, but can otherwise be chosen arbitrarily. For many dynamical core test cases the choice of hydrostatic background is often implicit in the formulation. When modeling idealized flows, however, the background state can be chosen to be some hydrostatically balanced mean state of the atmosphere, potentially obtained by projecting the non-hydrostatic initial data into the subspace of hydrostatic solutions.

Using the curvilinear operators defined by cubed-sphere coordinates (see Appendix G) along with the shallow-atmosphere approximation (see Appendix H) the non-hydrostatic equations (6.4)-(6.6) can be rewritten as

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J\rho u^k) &= 0, && (6.12)\\
\frac{\partial \rho u^\alpha}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J(\rho u^\alpha u^k + G^{\alpha k} p)) &= \psi_M^\alpha + \psi_C^\alpha, && (6.13)\\
\frac{\partial \rho u^\beta}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J(\rho u^\beta u^k + G^{\beta k} p)) &= \psi_M^\beta + \psi_C^\beta, && (6.14)\\
\frac{\partial \rho u^r}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J(\rho u^r u^k + G^{rk} p')) &= \psi_G^r, && (6.15)\\
\frac{\partial \rho\theta}{\partial t} + \frac{1}{J}\frac{\partial}{\partial x^k}(J\rho\theta u^k) &= 0. && (6.16)
\end{aligned}$$

Here J is the metric Jacobian, which is defined by (H.1). The source terms due to the underlying geometry are denoted by $\boldsymbol{\psi}_M = (\psi_M^\alpha, \psi_M^\beta)$, which under the shallow-atmosphere approximation takes the form

$$\begin{pmatrix} \psi_M^\alpha \\ \\ \psi_M^\beta \end{pmatrix} = \frac{2}{\delta^2} \begin{pmatrix} -XY^2 \rho u^\alpha u^\alpha + Y(1+Y^2)\rho u^\alpha u^\beta \\ \\ X(1+X^2)\rho u^\alpha u^\beta - X^2 Y \rho u^\beta u^\beta \end{pmatrix}. \qquad (6.17)$$

The second source term $\boldsymbol{\psi}_C = (\psi_C^\alpha, \psi_C^\beta)$ describes forcing due to Coriolis effect. On equatorial panels it is given by

$$\begin{pmatrix} \psi_C^\alpha \\ \\ \psi_C^\beta \end{pmatrix} = \frac{2\Omega}{\delta^2} \begin{pmatrix} -XY^2 & Y(1+Y^2) \\ \\ -Y(1+X^2) & XY^2 \end{pmatrix} \begin{pmatrix} \rho u^\alpha \\ \\ \rho u^\beta \end{pmatrix}. \tag{6.18}$$

On polar panels it is given by

$$\begin{pmatrix} \psi_C^\alpha \\ \\ \psi_C^\beta \end{pmatrix} = \frac{2s\Omega}{\delta^2} \begin{pmatrix} -XY & (1+Y^2) \\ \\ -(1+X^2) & XY \end{pmatrix} \begin{pmatrix} \rho u^\alpha \\ \\ \rho u^\beta \end{pmatrix}, \tag{6.19}$$

where s is a panel indicator defined by

$$s = \begin{cases} 1 & \text{on the northern panel } (n_p = 5), \\ -1 & \text{on the northern panel } (n_p = 6). \end{cases} \tag{6.20}$$

Finally, the gravitational source in the vertical momentum equation is denoted by ψ_G^r, and defined by

$$\psi_G^r = -\rho' g. \tag{6.21}$$

6.4 Numerical Method

In this section we present the numerical methodology used to solve the non-hydrostatic equations of motion in a discrete context. MCore uses the method-of-lines to split the spatial and temporal components of the equations. Further, it splits the horizontal and vertical component of the fluid motion, solving for the former using an explicit approach and the later using an implicit scheme. Coupling of these terms is managed via a Strang-carryover strategy, which ensures second-order-accuracy in time. In general, MCore is fourth-order-accurate for horizontal motions and second-order-accurate for vertical motions.

6.4.1 Finite-Volume Discretization

In this section we present the finite-volume discretization of the non-hydrostatic equations using a high-order finite-volume scheme. Under the finite-volume approach, we first integrate the Euler equations in the form (6.4)-(6.6) over an element $\mathcal{Z}$ (with volume $|\mathcal{Z}|$) and make use of Gauss' divergence theorem to write the flux term as an integral around the boundary $\partial\mathcal{Z}$, giving

$$\frac{\partial}{\partial t}\overline{\overline{\mathbf{q}}} + \frac{1}{|\mathcal{Z}|} \iint\limits_{\partial\mathcal{Z}} \boldsymbol{\mathcal{F}} \cdot \mathbf{n} dS = \overline{\overline{\boldsymbol{\psi}}}_M + \overline{\overline{\boldsymbol{\psi}}}_C + \overline{\overline{\boldsymbol{\psi}}}_G. \tag{6.22}$$

Here the flux integral is taken over the surface with normal vector $\mathbf{n}$ and infinitesimal area element dS. The term $\boldsymbol{\mathcal{F}} \cdot \mathbf{n}$ is a vector quantity that denotes the outward flux of each of the state variables perpendicular to the boundary. The double overline denotes a 3D average of the form

$$\overline{\overline{\phi}} = \frac{1}{|\mathcal{Z}|} \int\limits_{\mathcal{Z}} \phi dV. \tag{6.23}$$

The term $dV = Jd\alpha d\beta dr$ denotes the infinitesimal volume element, which can vary depending on the geometry. Here $\overline{\overline{\mathbf{q}}}$ denotes the averaged state vector in cell $\mathcal{Z}$. Likewise, $\overline{\overline{\boldsymbol{\psi}}}_M$ denotes the element-averaged source terms due to the underlying geometry, whereas $\overline{\overline{\boldsymbol{\psi}}}_C$ and $\overline{\overline{\boldsymbol{\psi}}}_G$ respectively denote the source terms due to the Coriolis force and gravity. Note that the volume-averaged formulation (6.22) is exactly equivalent to the original non-hydrostatic equations, and it is left to us to define an appropriate discretization over each of the terms in this expression.

In order to simplify the discussion on the computational implementation, we distinguish between the computational domain and physical domain. The physical domain on each panel consists of the region bounded by $(\alpha, \beta) \in [-\pi/4, \pi/4]^2$ in the horizontal and the region between the planetary surface, defined by $r_s(\alpha, \beta, n_p)$, and the fixed-height rigid-lid r_T in the vertical. On the other hand, the computational domain is the continuous analog of the physical domain in computer memory. On each panel it consists of a horizontal region bounded by $(\alpha, \beta) \in [-\pi/4, \pi/4]^2$, but in the vertical uses an auxiliary coordinate $\xi \in [0, 1]$. Importantly, grid elements must be uniform in computational space along each coordinate direction; that is, grid elements all have horizontal extent $\Delta\alpha$ (in α and β) and vertical extent $\Delta\xi = 1/N_r$ (where N_r is the discrete vertical integer resolution). This implies that surfaces of constant ξ are also the vertical bounding surfaces of model levels. An invertible conformal mapping is defined that takes a point in the computational domain to the physical domain via

$$\Phi : (\xi, \alpha, \beta, n_p) \to (\mathcal{R}(\xi; \alpha, \beta, n_p), \alpha, \beta, n_p). \tag{6.24}$$

Note that $\mathcal{R}(\xi) : [0, 1] \to [r_s(\alpha, \beta, n_p), r_T]$ must be bijective and monotonically increasing in ξ. In real atmospheric simulations, the function $\mathcal{R}$ can be chosen to place additional grid resolution near the surface. In section 6.5 we give one possible choice of $\mathcal{R}$, which will be used in our simulations.

For all arrays which are defined over the grid, there must exist a uniform mapping between array indices and computational space. Element centroids, for instance, are denoted by integer sub-indices, and over a single cubed-sphere panel are given by

$$\alpha_i = \left(i + \frac{1}{2}\right)\Delta\alpha - \frac{\pi}{4}, \quad \beta_j = \left(j + \frac{1}{2}\right)\Delta\alpha - \frac{\pi}{4}, \quad \xi_k = \left(k + \frac{1}{2}\right)\Delta\xi. \tag{6.25}$$

The indices span the range $(i, j) \in [1, N_c]^2$ and $k \in [1, N_r]$. The radial coordinate of element centroids is defined by mapping ξ_k to physical space,

$$r_k = \mathcal{R}(\xi_k; \alpha_i, \beta_i, n_p). \tag{6.26}$$

Note that $k = 1$ corresponds to the lowermost model level whereas $k = N_r$ corresponds to the uppermost model level, which is the opposite of most pressure-based hydrostatic model formulations.

Faces are midway between neighboring element centroids, and so are defined by half-indices and denoted by the symbol $\partial\mathcal{Z}$. For example, the edge $\partial\mathcal{Z}_{i+1/2,j,k}$ is at the interface between element (i,j,k) and $(i+1,j,k)$ and defines a plane that is constant in both α and β. Radial faces, denoted by $\partial\mathcal{Z}_{i,j,k+1/2}$ are constant in ξ, but are not necessarily constant in r due to the horizontal dependence of the mapping $\mathcal{R}$. In fact, they are defined by the two-dimensional surface $r-\mathcal{R}(\xi;\alpha,\beta,n_p)=0$ for ξ fixed.

In the finite-volume discretization we must make use of both element volumes, denoted by $|\mathcal{Z}|$, and face areas, denoted by $|\partial\mathcal{Z}|$. In terms of the metric Jacobian (H.1), the element volume takes the form

$$|\mathcal{Z}|_{i,j,k} = \int\limits_{\mathcal{Z}} dV = \int\limits_{\alpha_{i-1/2}}^{\alpha_{i+1/2}} \int\limits_{\beta_{j-1/2}}^{\beta_{j+1/2}} \int\limits_{r_{k-1/2}}^{r_{k+1/2}} J dr d\beta d\alpha. \tag{6.27}$$

The area of faces is likewise calculated by integrating the infinitesimal face areas (G.6)-(G.8). In practice these integrals are pre-computed using high-order Gaussian quadrature and stored for later use.

Quantities which are defined as edge averages will be denoted by a single overline. Average fluxes are defined at element faces and denoted here by $\overline{\mathbf{F}}_{i+1/2,j,k}$ for a flux across face $(i+1/2,j,k)$. They are defined by

$$\overline{\mathbf{F}}_{i+1/2,j,k} = \frac{1}{|\partial\mathcal{Z}|_{i+1/2,j,k}} \iint\limits_{\partial\mathcal{Z}_{i+1/2,j,k}} \boldsymbol{\mathcal{F}}\cdot\mathbf{n}dS, \tag{6.28}$$

where $|\partial\mathcal{Z}|_{i+1/2,j,k}$ denotes the area of the face. Hence, the total flux across an interface, here denoted by $\mathbf{F}_{i+1/2,j,k}$, is simply the product of the average flux and the face area

$$\mathbf{F}_{i+1/2,j,k} = \overline{\mathbf{F}}_{i+1/2,j,k}\ |\partial\mathcal{Z}|_{i+1/2,j,k}. \tag{6.29}$$

Using (6.29) the volume averaged formulation (6.22) can be rewritten as

$$\frac{\partial}{\partial t}\overline{\overline{\mathbf{q}}}_{i,j,k} = \mathbf{H}(\mathbf{q}) + \mathbf{V}(\mathbf{q}), \tag{6.30}$$

where

$$\mathbf{H}(\mathbf{q}) = \frac{1}{|\mathcal{Z}|_{i,j,k}} \left[\mathbf{F}_{i-1/2,j,k} - \mathbf{F}_{i+1/2,j,k} + \mathbf{F}_{i,j-1/2,k} - \mathbf{F}_{i,j+1/2,k}\right] + \overline{\overline{\psi}}_M + \overline{\overline{\psi}}_C, \tag{6.31}$$

and

$$\mathbf{V}(\mathbf{q}) = \frac{1}{|\mathcal{Z}|_{i,j,k}} \left[\mathbf{F}_{i,j,k-1/2} - \mathbf{F}_{i,j,k+1/2}\right] + \overline{\overline{\psi}}_G. \tag{6.32}$$

Here we utilize a splitting based on separating horizontal motions, denoted by $\mathbf{H}(\mathbf{q})$, and vertical motions, denoted by $\mathbf{V}(\mathbf{q})$. Numerically this splitting is desirable since the relatively small vertical spacing of elements (on the scale of tens to hundreds of meters) imposes a severe timestep restriction via the CFL condition. Hence, if we make use of an explicit treatment of the horizontal motions and treat the vertical motions implicitly, the model timestep will not be affected by the grid spacing in the vertical.

6.4.2 Horizontal Reconstruction

Under the finite-volume formulation only element-averaged information is known within each element. Hence, a sub-grid-scale reconstruction can only be built using information from neighboring elements. MCore uses a novel reconstruction strategy which attains fourth-order accuracy using a minimal set of neighboring elements. In general, attaining greater than second-order accuracy is difficult since a $O(\Delta\alpha^n)$ reconstruction requires that the p^{th} derivatives be approximated to $O(\Delta\alpha^{n-p})$. Standard finite-difference reconstruction formulae only guarantee $O(\Delta\alpha^2)$ accuracy, since such an approach approximates cell-centerpoint values using cell-averaged values. In a non-Cartesian domain, reconstructions must also incorporate information about the underlying geometry to recover high-order accuracy. For simplicity, we choose to drop the vertical index k in this section since the horizontal reconstruction process is applied on each vertical level separately.

The strategy we have introduced relies on the convolution and deconvolution procedure of *Barad and Colella* (2005). This approach provides a mechanism for transforming cell-averages to cell-centered point values over a sufficiently smooth data field, according to

$$\phi_{(0)} = \overline{\overline{\phi}} - \frac{\Delta\alpha^4}{12|\mathcal{Z}|}\left(\frac{\partial\phi}{\partial\alpha}\frac{\partial\tilde{J}}{\partial\alpha} + \frac{\partial\phi}{\partial\beta}\frac{\partial\tilde{J}}{\partial\beta}\right) - \frac{\Delta\alpha^2}{24}\left(\frac{\partial^2\phi}{\partial\alpha^2} + \frac{\partial^2\phi}{\partial\beta^2}\right), \tag{6.33}$$

where $\tilde{J}$ is the radially-integrated Jacobian,

$$\tilde{J}(\alpha,\beta,n_p) = \int\limits_{\mathcal{R}(\xi_{k-1/2};\alpha,\beta,n_p)}^{\mathcal{R}(\xi_{k+1/2};\alpha,\beta,n_p)} J(r,\alpha,\beta,n_p)dr. \tag{6.34}$$

If $\overline{\overline{\phi}}_{i,j}$ is known to at least fourth-order accuracy and the remaining derivative terms are known to at least $O(\Delta\alpha^2)$, this formula leads to a fourth-order-accurate approximation of $\phi_{(0)i,j}$, the cell-centered value of ϕ in element (i,j). In this formulation, we directly approximate the derivatives of the radially-integrated Jacobian by

$$\frac{\partial\tilde{J}}{\partial\alpha} \approx \frac{|\mathcal{Z}|_{i+1,j} - |\mathcal{Z}|_{i-1,j}}{2\Delta\alpha}, \tag{6.35}$$

$$\frac{\partial\tilde{J}}{\partial\beta} \approx \frac{|\mathcal{Z}|_{i,j+1} - |\mathcal{Z}|_{i,j-1}}{2\Delta\alpha}. \tag{6.36}$$

For reasons of efficiency, these derivative are pre-computed prior to the simulation and stored for later use.

The reconstruction used by MCore proceeds as follows. The stencil we will use in the reconstruction step is depicted in Figure 6.2. Second and third derivatives are first

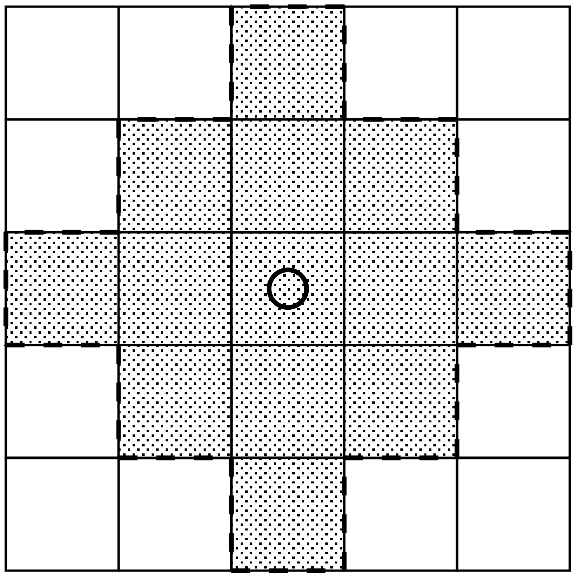


Figure 6.2: A depiction of the stencil used for computing the fourth-order sub-grid-scale reconstruction on the cubed-sphere.

calculated using standard finite-difference formulae, which leads to approximations which are $O(\Delta\alpha^2)$ accurate. Here we use q to denote one element of the state vector $\mathbf{q}$.

$$
\begin{aligned}
D_{\alpha\alpha\alpha} q_{i,j} &= \frac{\overline{\overline{q}}_{i+2,j} - 2\overline{\overline{q}}_{i+1,j} + 2\overline{\overline{q}}_{i-1,j} - \overline{\overline{q}}_{i-2,j}}{12\Delta\alpha^3}, && (6.37)\\
D_{\alpha\alpha\beta} q_{i,j} &= \frac{\overline{\overline{q}}_{i+1,j+1} - \overline{\overline{q}}_{i+1,j-1} - 2\overline{\overline{q}}_{i,j+1} + 2\overline{\overline{q}}_{i,j-1} + \overline{\overline{q}}_{i-1,j+1} - \overline{\overline{q}}_{i-1,j-1}}{4\Delta\alpha^3}, && (6.38)\\
D_{\alpha\beta\beta} q_{i,j} &= \frac{\overline{\overline{q}}_{i+1,j+1} - 2\overline{\overline{q}}_{i+1,j} + \overline{\overline{q}}_{i+1,j-1} - \overline{\overline{q}}_{i-1,j+1} + 2\overline{\overline{q}}_{i-1,j} - \overline{\overline{q}}_{i-1,j-1}}{4\Delta\alpha^3}, && (6.39)\\
D_{\beta\beta\beta} q_{i,j} &= \frac{\overline{\overline{q}}_{i,j+2} - 2\overline{\overline{q}}_{i,j+1} + 2\overline{\overline{q}}_{i,j-1} - \overline{\overline{q}}_{i,j-2}}{12\Delta\alpha^3}. && (6.40)
\end{aligned}
$$

$$\begin{aligned}
D_{\alpha\alpha} q_{i,j} &= \frac{-\overline{\overline{q}}_{i+2,j} + 16\overline{\overline{q}}_{i+1,j} - 30\overline{\overline{q}}_{i,j} + 16\overline{\overline{q}}_{i-1,j} - \overline{\overline{q}}_{i-2,j}}{24\Delta\alpha^2}, && (6.41) \\
D_{\alpha\beta} q_{i,j} &= \frac{\overline{\overline{q}}_{i+1,j+1} - \overline{\overline{q}}_{i-1,j+1} - \overline{\overline{q}}_{i+1,j-1} + \overline{\overline{q}}_{i-1,j-1}}{4\Delta\alpha^2}, && (6.42) \\
D_{\beta\beta} q_{i,j} &= \frac{-\overline{\overline{q}}_{i,j+2} + 16\overline{\overline{q}}_{i,j+1} - 30\overline{\overline{q}}_{i,j} + 16\overline{\overline{q}}_{i,j-1} - \overline{\overline{q}}_{i,j-2}}{24\Delta\alpha^2}. && (6.43)
\end{aligned}$$

To attain fourth-order accuracy in space, we require that our approximation to the first derivative terms be at least $O(\Delta\alpha^3)$. To construct such a formula, we utilize a finite-difference relation over element point-values which have been reconstructed to $O(\Delta\alpha^4)$. First, define $O(\Delta\alpha^2)$ approximations to the first derivatives by

$$\begin{aligned}
D^*_\alpha q_{i,j} &= \frac{-\overline{\overline{q}}_{i+2,j} + 8\overline{\overline{q}}_{i+1,j} - 8\overline{\overline{q}}_{i-1,j} + \overline{\overline{q}}_{i-2,j}}{12\Delta\alpha}, && (6.44) \\
D^*_\beta q_{i,j} &= \frac{-\overline{\overline{q}}_{i,j+2} + 8\overline{\overline{q}}_{i,j+1} - 8\overline{\overline{q}}_{i,j-1} + \overline{\overline{q}}_{i,j-2}}{12\Delta\alpha}. && (6.45)
\end{aligned}$$

High-order element averages of the state vector which do not incorporate geometric terms can then be computed via

$$\begin{aligned}
q^*_{i+m,j} &= \overline{\overline{q}}_{i+m,j} - \frac{\Delta\alpha^4}{12|\mathcal{Z}|_{i+m,j}} \Bigg[\left(D^*_\alpha q_{i,j} + m\Delta\alpha D_{\alpha\alpha} q_{i,j}\right) \left(\frac{\partial \tilde{J}}{\partial \alpha}\right)_{i+m,j} && (6.46) \\
&\quad + \left(D^*_\beta q_{i,j} + m\Delta\alpha D_{\alpha\beta} q_{i,j}\right) \left(\frac{\partial \tilde{J}}{\partial \beta}\right)_{i+m,j} \Bigg],
\end{aligned}$$

and

$$\begin{aligned}
q^*_{i,j+m} &= \overline{\overline{q}}_{i,j+m} - \frac{\Delta\alpha^4}{12|\mathcal{Z}|_{i,j+m}} \Bigg[\left(D^*_\alpha q_{i,j} + m\Delta\alpha D_{\alpha\beta} q_{i,j}\right) \left(\frac{\partial \tilde{J}}{\partial \alpha}\right)_{i,j+m} && (6.47) \\
&\quad + \left(D^*_\beta q_{i,j} + m\Delta\alpha D_{\beta\beta} q_{i,j}\right) \left(\frac{\partial \tilde{J}}{\partial \beta}\right)_{i,j+m} \Bigg].
\end{aligned}$$

A third-order approximation to the first derivative terms is then given by

$$\begin{aligned} D_\alpha q_{i,j} &= \frac{-q^*_{i+2,j} + 8q^*_{i+1,j} - 8q^*_{i-1,j} + q^*_{i-2,j}}{12\Delta\alpha} - \frac{\Delta\alpha^2}{24}\left(D_{\alpha\alpha\alpha} q_{i,j} + D_{\alpha\beta\beta} q_{i,j}\right), \quad (6.48) \\ D_\beta q_{i,j} &= \frac{-q^*_{i,j+2} + 8q^*_{i,j+1} - 8q^*_{i,j-1} + q^*_{i,j-2}}{12\Delta\alpha} - \frac{\Delta\alpha^2}{24}\left(D_{\alpha\alpha\beta} q_{i,j} + D_{\beta\beta\beta} q_{i,j}\right) \quad (6.49) \end{aligned}$$

A fourth-order approximation to the element centerpoint value is similarly obtained from the relation

$$q_{(0)i,j} = \bar{\bar{q}}_{i,j} - \frac{\Delta\alpha^2}{12|\mathcal{Z}|_{i,j}}\left[D_\alpha q_{i,j}\left(\frac{\partial\tilde{J}}{\partial\alpha}\right)_{i,j} + D_\beta q_{i,j}\left(\frac{\partial\tilde{J}}{\partial\beta}\right)_{i,j}\right] - \frac{\Delta\alpha^2}{24}\left[D_{\alpha\alpha} q_{i,j} + D_{\beta\beta} q_{i,j}\right]. \tag{6.50}$$

We make no attempt at introducing a limiter for enforcing monotonicity or positivity of thermodynamic quantities. Limiting is generally of greater importance for tracer transport problems where monotonicity and positivity must be guaranteed, and so it is not pursued here.

With the approximated centerpoint value and corresponding derivatives in hand, we can now write expressions for the reconstructed state vector at face centerpoints. These face centerpoint values are denoted with either a L or R, denoting whether they are left or right of the interface:

$$\begin{aligned} q_{L,i+1/2,j} &= q_{(0)i,j} + D_\alpha q_{i,j}\left(\frac{\Delta\alpha}{2}\right) + D_{\alpha\alpha} q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^2 + D_{\alpha\alpha\alpha} q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^3, \quad (6.51) \\ q_{R,i-1/2,j} &= q_{(0)i,j} - D_\alpha q_{i,j}\left(\frac{\Delta\alpha}{2}\right) + D_{\alpha\alpha} q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^2 - D_{\alpha\alpha\alpha} q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^3, \quad (6.52) \\ q_{L,i,j+1/2} &= q_{(0)i,j} + D_\beta q_{i,j}\left(\frac{\Delta\alpha}{2}\right) + D_{\beta\beta} q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^2 + D_{\beta\beta\beta} q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^3, \quad (6.53) \\ q_{R,i,j-1/2} &= q_{(0)i,j} - D_\beta q_{i,j}\left(\frac{\Delta\alpha}{2}\right) + D_{\beta\beta} q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^2 - D_{\beta\beta\beta} q_{i,j}\left(\frac{\Delta\alpha}{2}\right)^3. \quad (6.54) \end{aligned}$$

The centerpoint values, as calculated from the above formulae are then used for computing edge fluxes across the interface via a Riemann solver. This calculation will be described in more detail later.

6.4.3 Vertical Reconstruction

In the vertical we can no longer rely on uniformity of the grid spacing, so the reconstruction must take into account the variance in the height of grid elements. Nonetheless, we can still construct a second-order sub-grid-scale reconstruction within each element by fitting a parabola through the centroid of each grid element and its immediate neighbors. As a result, we obtain the following formulae for the reconstructed field at each edge:

$$\mathbf{q}_{L,k+1/2} = -\frac{\Delta r_p^2}{4\Delta r_n(\Delta r_n + \Delta r_p)}\mathbf{q}_{k-1} + \frac{(2\Delta r_n + \Delta r_p)}{4\Delta r_n}\mathbf{q}_k + \frac{(2\Delta r_n + \Delta r_p)}{4(\Delta r_n + \Delta r_p)}\mathbf{q}_{k+1}, \tag{6.55}$$

$$\mathbf{q}_{R,k-1/2} = \frac{(\Delta r_n + 2\Delta r_p)}{4(\Delta r_n + \Delta r_p)}\mathbf{q}_{k-1} + \frac{(\Delta r_n + 2\Delta r_p)}{4\Delta r_p}\mathbf{q}_k - \frac{\Delta r_n^2}{4\Delta r_p(\Delta r_p + \Delta r_n)}\mathbf{q}_{k+1}, \tag{6.56}$$

where

$$\Delta r_p = r_{k+1} - r_k, \qquad \Delta r_n = r_k - r_{k-1}. \tag{6.57}$$

Unlike in the horizontal, the subscripts L and R actually correspond to the elements immediately below and immediately above each edge, respectively.

Near the top and bottom boundaries, the stencil width is reduced by one element and so the sub-grid-scale reconstruction becomes linear. At the bottom boundary the reconstruction then reads

$$\mathbf{q}_{R,1/2} = \frac{(2\Delta r_n + \Delta r_p)}{\Delta r_n + \Delta r_p}\mathbf{q}_1 - \frac{\Delta r_n}{\Delta r_n + \Delta r_p}\mathbf{q}_2, \tag{6.58}$$

$$\mathbf{q}_{L,3/2} = \frac{\Delta r_p}{\Delta r_p + \Delta r_n}\mathbf{q}_1 + \frac{\Delta r_n}{\Delta r_p + \Delta r_n}\mathbf{q}_2, \tag{6.59}$$

where

$$\Delta r_p = r_3 - r_2, \qquad \Delta r_n = r_2 - r_1. \tag{6.60}$$

At the top boundary, the reconstruction reads

$$\mathbf{q}_{L,N_r+1/2} = -\frac{\Delta r_p}{\Delta r_n + \Delta r_p}\mathbf{q}_{N_r-1} + \frac{2\Delta r_p + \Delta r_n}{\Delta r_n + \Delta r_p}\mathbf{q}_{N_r}, \tag{6.61}$$

$$\mathbf{q}_{R,N_r-1/2} = \frac{\Delta r_p}{\Delta r_p + \Delta r_n}\mathbf{q}_{N_r-1} + \frac{\Delta r_n}{\Delta r_p + \Delta r_n}\mathbf{q}_{N_r}, \tag{6.62}$$

where

$$\Delta r_p = r_{N_r} - r_{N_r-1}, \qquad \Delta r_n = r_{N_r-1} - r_{N_r-2}. \tag{6.63}$$

6.4.4 Horizontal-Vertical Splitting and Time-stepping Scheme

To guarantee sufficient accuracy in time, MCore uses the Strang-carryover approach presented in *Ullrich and Jablonowski* (2011a) to couple explicit integration in the horizontal with implicit integration in the vertical. In the horizontal, this scheme can use either the third-order strong-stability preserving Runge-Kutta integrator of *Gottlieb et al.* (2001) or the well-known fourth-order Runge-Kutta scheme. Either choice of timestep leads to a scheme with implicit-explicit coupling terms of overall second-order accuracy.

At the initial step, we solve for $\mathbf{q}^{(1)}$ via an implicit step of duration $\Delta t/2$:

$$\frac{\mathbf{q}^{(1)} - \mathbf{q}^n}{(\Delta t/2)} - \mathbf{V}(\mathbf{q}^{(1)}) = 0. \tag{6.64}$$

MCore allows for compile-time switching between linearly implicit integration and fully implicit integration in the vertical. The implicit solve is performed using Newton-Krylov iteration using either a numerically or analytically computed Jacobian matrix. Since our choice of Riemann solver in the vertical is nearly linear, the current implementation of MCore makes use of an analytically computed matrix.

After the implicit solve, the state $\mathbf{q}^{(1)}$ is used as input for an explicit scheme. Choosing the third-order strong-stability preserving Runge-Kutta method of *Gottlieb* *et al.* (2001) leads to

$$
\begin{aligned}
\mathbf{q}^{(2)} &= \mathbf{q}^{(1)} + \Delta t \mathbf{H}(\mathbf{q}^{(1)}), && (6.65)\\
\mathbf{q}^{(3)} &= \frac{3}{4}\mathbf{q}^{(1)} + \frac{1}{4}\mathbf{q}^{(2)} + \frac{\Delta t}{4}\mathbf{H}(\mathbf{q}^{(2)}), && (6.66)\\
\mathbf{q}^{*} &= \frac{1}{3}\mathbf{q}^{(1)} + \frac{2}{3}\mathbf{q}^{(3)} + \frac{2\Delta t}{3}\mathbf{H}(\mathbf{q}^{(3)}). && (6.67)
\end{aligned}
$$

The fourth-order Runge-Kutta scheme instead leads to the sequence

$$
\begin{aligned}
\mathbf{q}^{(2)} &= \mathbf{q}^{(1)} + \frac{\Delta t}{2}\mathbf{H}(\mathbf{q}^{(1)}), && (6.68)\\
\mathbf{q}^{(3)} &= \mathbf{q}^{(1)} + \frac{\Delta t}{2}\mathbf{H}(\mathbf{q}^{(2)}), && (6.69)\\
\mathbf{q}^{(4)} &= \mathbf{q}^{(1)} + \Delta t \mathbf{H}(\mathbf{q}^{(3)}), && (6.70)\\
\mathbf{q}^{*} &= -\frac{1}{3}\mathbf{q}^{(1)} + \frac{1}{3}\mathbf{q}^{(2)} + \frac{2}{3}\mathbf{q}^{(3)} + \frac{1}{3}\mathbf{q}^{(4)} + \frac{\Delta t}{6}\mathbf{H}(\mathbf{q}^{(4)}). && (6.71)
\end{aligned}
$$

Finally, $\mathbf{q}^*$ becomes the input for a final implicit solve of size $\Delta t/2$:

$$\frac{\mathbf{q}^{n+1} - \mathbf{q}^*}{(\Delta t/2)} - \mathbf{V}(\mathbf{q}^{n+1}) = 0. \tag{6.72}$$

The forcing due to the implicit step is then stored

$$\mathcal{G} = \mathbf{q}^{n+1} - \mathbf{q}^*. \tag{6.73}$$

At the following timestep, the implicit forcing from the previous timestep is used to predict the next input for the explicit scheme,

$$\mathbf{q}^{(1)} = \mathbf{q}^n + \mathcal{G}. \tag{6.74}$$

Observe that after the initial step, the resulting scheme only uses one implicit solve per timestep.

6.4.5 Orthonormalization

MCore makes use of approximate Riemann solvers for computing the flux $\mathbf{F}(\mathbf{q})$ across each face. However, generic Riemann solvers are purely one-dimensional operators. To apply this class of solvers to multidimensional problems, we must first transform vector quantities into an orthonormal frame. In 3D, the orthonormal frame consists of one basis vector which is orthogonal to the active edge (denoted $\mathbf{e}_\perp$) and two components which are parallel to the edge (denoted $\mathbf{e}_1$ and $\mathbf{e}_2$). Hence, the non-orthogonal components define a tangent plane to the active edge. At the point where the orthogonal basis is defined, an arbitrary vector can be written in either the natural basis,

$$\mathbf{v} = v^\alpha \mathbf{g}_\alpha + v^\beta \mathbf{g}_\beta + v^r \mathbf{g}_r, \tag{6.75}$$

or in the orthonormal basis

$$\mathbf{v} = v^\perp \mathbf{e}_\perp + v^1 \mathbf{e}_1 + v^2 \mathbf{e}_2. \tag{6.76}$$

The transformation between these two systems is accomplished via the orthonormalization matrix $\mathcal{O}_d(\alpha, \beta, r)$,

$$\begin{pmatrix} v^\perp \\ v^1 \\ v^2 \end{pmatrix} = \mathcal{O}_d(\alpha, \beta, r) \begin{pmatrix} v^\alpha \\ v^\beta \\ v^r \end{pmatrix}. \tag{6.77}$$

Here d denotes the coordinate being held constant, $d \in \{\alpha, \beta, \xi\}$. Analogously, transforming from the orthonormal basis to the natural basis simply requires applying the inverse operation,

$$\begin{pmatrix} v^\alpha \\ v^\beta \\ v^r \end{pmatrix} = \mathcal{O}_d^{-1}(\alpha, \beta, r) \begin{pmatrix} v^\perp \\ v^1 \\ v^2 \end{pmatrix}. \tag{6.78}$$

Since the natural basis for the cubed-sphere consists of a radial basis vector which is already normalized and orthogonal to the basis vectors in the horizontal, at edges of constant α and constant β, we simply set $\mathbf{e}_2 = \mathbf{g}_r$. The orthonormalization matrix at edges of constant α can then be written as a 2×2 matrix,

$$\mathcal{O}_\alpha = \left(\begin{array}{cc} \dfrac{\sqrt{1+X^2}}{\delta} & 0 \\ \dfrac{-XY\sqrt{1+X^2}}{\delta^2} & \dfrac{(1+Y^2)\sqrt{1+X^2}}{\delta^2} \end{array} \right). \tag{6.79}$$

Similarly, at edges of constant β we have

$$\mathcal{O}_\beta = \left(\begin{array}{cc} 0 & \dfrac{\sqrt{1+Y^2}}{\delta} \\ \dfrac{(1+X^2)\sqrt{1+Y^2}}{\delta^2} & \dfrac{-XY\sqrt{1+Y^2}}{\delta^2} \end{array} \right). \tag{6.80}$$

In each case the matrices defined above only work on the horizontal components of the vector $\mathbf{v}$. The corresponding deorthonormalization matrices are

$$\mathcal{O}_\alpha^{-1} = \left(\begin{array}{cc} \dfrac{\delta}{\sqrt{1+X^2}} & 0 \\ \dfrac{XY\delta}{(1+Y^2)\sqrt{1+X^2}} & \dfrac{\delta^2}{(1+Y^2)\sqrt{1+X^2}} \end{array} \right), \tag{6.81}$$

$$\mathcal{O}_\beta^{-1} = \left(\begin{array}{cc} \dfrac{XY\delta}{(1+X^2)\sqrt{1+Y^2}} & \dfrac{\delta^2}{(1+X^2)\sqrt{1+Y^2}} \\ \dfrac{\delta}{\sqrt{1+Y^2}} & 0 \end{array} \right). \tag{6.82}$$

The problem of deriving an orthonormalization matrix in the vertical is more difficult since a cell's vertical bounding surface (a surface of constant ξ) is only a surface of constant r in the absence of terrain. When terrain is present, we utilize the fact that $\nabla^i f$ defines a vector perpendicular to the surface $f(\alpha, \beta, r) = r -$

$\mathcal{R}(\xi;\alpha,\beta,n_p)$ (with $\xi = \text{const.}$). Hence, the vector

$$\mathbf{e}^*_\perp = \left[\left[G^{11}\left(-\frac{\partial\mathcal{R}}{\partial\alpha}\right) + G^{12}\left(-\frac{\partial\mathcal{R}}{\partial\beta}\right)\right]\mathbf{g}_\alpha + \left[G^{21}\left(-\frac{\partial\mathcal{R}}{\partial\alpha}\right) + G^{22}\left(-\frac{\partial\mathcal{R}}{\partial\beta}\right)\right]\mathbf{g}_\beta + \mathbf{g}_r\right], \tag{6.83}$$

is orthogonal to the given terrain-following radial face (but is not of unit length). A vector of unit length can be obtained by simply scaling $\mathbf{e}^*_\perp$ according to

$$\mathbf{e}_\perp = \frac{1}{|\mathbf{e}^*_\perp|}\mathbf{e}^*_\perp, \tag{6.84}$$

where $|\mathbf{v}| = \sqrt{G_{ij}v^iv^j}$ is the magnitude of the vector in cubed-sphere coordinates. Hereafter we use $e^\alpha_\perp$, $e^\beta_\perp$ and $e^r_\perp$ to denote the components of $\mathbf{e}_\perp$ along the α, β and r directions. To obtain vectors orthogonal to $\mathbf{e}_\perp$, we simply apply the Gram-Schmidt orthonormalization procedure to the α and β components of the natural basis. For example, for the α component, we have

$$\mathbf{e}^*_1 = \mathbf{g}_\alpha - \langle\mathbf{g}_\alpha, \mathbf{e}_\perp\rangle\mathbf{e}_\perp, \tag{6.85}$$

with curvilinear inner product $\langle\mathbf{v},\mathbf{w}\rangle = G_{ij}v^iw^j$. Hence, on simplifying,

$$\mathbf{e}^*_1 = (1 - Q^\alpha e^\alpha_\perp)\mathbf{g}_\alpha - Q^\alpha e^\beta_\perp\mathbf{g}_\beta - Q^\alpha e^r_\perp\mathbf{g}_r, \tag{6.86}$$

where $Q^\alpha = G_{11}e^\alpha_\perp + G_{12}e^\beta_\perp$. Similarly,

$$\mathbf{e}^*_2 = -Q^\beta e^\alpha_\perp\mathbf{g}_\alpha + (1 - Q^\beta e^\beta_\perp)\mathbf{g}_\beta - Q^\beta e^r_\perp\mathbf{g}_r, \tag{6.87}$$

where $Q^\beta = G_{21}e^\alpha_\perp + G_{22}e^\beta_\perp$. As before, we normalize these vectors to obtain $\mathbf{e}_1$ and $\mathbf{e}_2$. The deorthonormalization matrix $\mathcal{O}^{-1}_\xi$ is then defined as the matrix whose columns are given by $\mathbf{e}_\perp$, $\mathbf{e}_1$ and $\mathbf{e}_2$. To obtain $\mathcal{O}_\xi$ at each edge, we invert the corresponding deorthonormalization matrix numerically.

In the formulation used by MCore, the orthonormalization and deorthonormalization matrices are only required at the centerpoint of each face. Hence, these matrices can be pre-computed, for reasons of efficiency, and stored for later use. After certain optimizations to the algorithmic implementation of the Riemann solver, it turns out that only three components of each of the matrices are required during runtime.

6.4.6 Riemann Solvers

Under any upwind finite-volume formulation, reconstructed edge values are inherently discontinuous – that is, the reconstructed left edge value is almost never identically equal to the right edge value. The discrepancy between left and right edge values is a measure of the roughness of the underlying fields. Godunov-type finite-volume methods (*Godunov*, 1959), such as the one in this chapter, solve the Riemann problem at interfaces so as to obtain a single-valued flux at each edge. Since computing the exact solution of the Riemann problem is generally expensive, we instead rely on so-called approximate Riemann solvers. However, dozens of such approximate solvers are now available. *Ullrich et al.* (2010) analyzed three such solvers, including the solver of *Rusanov* (1961), the solve of *Roe* (1981) and the recently introduced AUSM$^+$-up solver of *Liou* (2006). They concluded that for horizontal atmospheric motions the Rusanov solver was far too diffusive, whereas both Roe and AUSM$^+$-up yielded comparable results, with AUSM$^+$-up slightly outperforming Roe.

Based on the results of *Ullrich et al.* (2010) we have adopted the AUSM$^+$-up solver for computing fluxes. The details of this Riemann solver are given in *Liou* (2006), and so the algorithm is not repeated here. The only notable change in our implementation is the choice of $K_u = 0$, instead of Liou's suggested choice $K_u = 1/4$. In the vertical we make use of a linearized variant of the AUSM$^+$-up flux, as this method has been shown to be very effective in the extremely low Mach number regime.

The AUSM$^+$-up Riemann solver was originally designed with the low-Mach number limit in mind. Many Riemann solvers introduce diffusion that is proportional to the sound speed over all state variables, which is potentially disastrous for accurate modeling of flows in the low-Mach number limit. For general low-Mach number flows, Liou showed that a better characterizations of numerical diffusion can be obtained by asymptotic analysis. For very low-speed flows, such as vertical atmospheric motions, Liou's solver can be simplified further by linearizing the diffusive terms. This approach was used in *Ullrich and Jablonowski* (2011a) to define a quasi-linear AUSM^{+}-up variant for vertical motions. This strategy is desirable since the resulting Jacobian takes on a simple analytic form that can be efficiently implemented in an implicit solver.

The quasi-linear AUSM^{+}-up solver makes use of the hydrostatic background state to define a background sound speed $c^h_{1/2} = \sqrt{\gamma p^h/\rho^h}$ at each vertical interface (with $\gamma = c_p/c_v$). Given left and right state vectors $\mathbf{q}_L = (\rho_L, (\rho\mathbf{u})_L, (\rho\theta)_L)$ and $\mathbf{q}_R = (\rho_R, (\rho\mathbf{u})_R, (\rho\theta)_R)$, we define a modified velocity at the interface via

$$w_{1/2} = \frac{(\rho v^{\perp})_L + (\rho v^{\perp})_R}{\rho_L + \rho_R} - K_p c^h_{1/2} \frac{(\rho\theta)_R - (\rho\theta)_L}{(\rho\theta)^h}, \tag{6.88}$$

where K_p is the dimensionless pressure diffusion coefficient and $(\rho\theta)^h$ is the potential temperature density of the hydrostatic background at the interface. Similarly, we define an interface pressure using (6.7) that is given by

$$p_{1/2} = p_0 \left(\frac{R_d}{p_0} \left(\frac{(\rho\theta)_L + (\rho\theta)_R}{2} \right) \right)^{c_p/c_v} - \frac{K_u}{2} c_{1/2}((\rho v^{\perp})_R - (\rho v^{\perp})_L), \tag{6.89}$$

where p^h is the hydrostatic pressure at the interface and K_u is the dimensionless momentum diffusion coefficient. The pointwise flux across the interface is then defined by

$$\mathbf{F}^{AUSM}_{(0)} = \mathbf{p}'_{1/2} + \begin{cases} w_{1/2}\mathbf{q}_L & \text{if } w_{1/2} > 0, \\ w_{1/2}\mathbf{q}_R & \text{otherwise,} \end{cases} \tag{6.90}$$

where

$$\mathbf{p}'_{1/2} = \left(0, p_{1/2} - p^h_{1/2}, 0, 0, 0\right), \tag{6.91}$$

i.e. $\mathbf{p}_{1/2}$ only has one non-zero value, which is a contribution associated with the perpendicular component of the momentum. This solver uses tuning parameters K_u and K_p, which have been chosen as $K_u = 2$ and $K_p = 1/4$. In *Ullrich and Jablonowski* (2011a) we have observed that this larger value of K_u is required to damp spurious vertical oscillations when the aspect ratio is large (as opposed to the recommended value of 3/4 given in *Liou* (2006)). Finally, since the vertical flux is only second-order accurate, we make the approximation

$$\overline{\mathbf{F}}_{i,j,k+1/2} = \mathbf{F}_{(0)i,j,k+1/2}, \tag{6.92}$$

i.e. we take the average flux across the interface to be equal to the pointwise flux evaluated from (6.90).

6.4.7 Fourth-order Horizontal Accuracy

To solve for the flux across horizontal faces we make use of the reconstructed values of the state variables on either side of the interface. Fluxes are evaluated pointwise at element face centerpoints, and are obtained by solving a Riemann problem using left and right state vectors. The resulting pointwise flux vector is a second-order approximation to the average flux across the interface. To obtain fourth-order accuracy in computing the flux across the interface, some form of high-order quadrature is needed. In this section we again drop the vertical index k, which is assumed constant.

MCore makes use of a convolution formula for computing average face fluxes $\overline{\mathbf{F}}$

from pointwise fluxes $\mathbf{F}_{(0)}$. On faces of constant α and β, this formula takes the form

$$\overline{\mathbf{F}}_{i+1/2,j} = \mathbf{F}_{(0)i+1/2,j} + \frac{\Delta\alpha^2}{24}\left(\frac{\partial^2\mathbf{F}}{\partial\beta^2}\right)_{i+1/2,j} + \frac{\Delta\alpha^2}{12|\partial\mathcal{Z}|_{i+1/2,j}}\left(\frac{\partial F}{\partial\beta}\right)_{i+1/2,j}\left(\frac{\partial\tilde{J}_\alpha}{\partial\beta}\right)_{i+1/2,j}, \tag{6.93}$$

$$\overline{\mathbf{F}}_{i,j+1/2} = \mathbf{F}_{(0)i,j+1/2} + \frac{\Delta\alpha^2}{24}\left(\frac{\partial^2\mathbf{F}}{\partial\alpha^2}\right)_{i,j+1/2} + \frac{\Delta\alpha^2}{12|\partial\mathcal{Z}|_{i,j+1/2}}\left(\frac{\partial F}{\partial\alpha}\right)_{i,j+1/2}\left(\frac{\partial\tilde{J}_\beta}{\partial\alpha}\right)_{i,j+1/2}. \tag{6.94}$$

Here $\tilde{J}_\alpha$ and $\tilde{J}_\beta$ are the radially-integrated metric co-factors (H.2)-(H.3). Under our formulation, their derivatives are approximated in terms of neighboring face areas as

$$\left(\frac{\partial\tilde{J}_\alpha}{\partial\beta}\right)_{i+1/2,j} \approx \frac{|\partial\mathcal{Z}|_{i+1/2,j+1} - |\partial\mathcal{Z}|_{i+1/2,j-1}}{2\Delta\alpha}, \tag{6.95}$$

and

$$\left(\frac{\partial\tilde{J}_\beta}{\partial\alpha}\right)_{i+1/2,j} \approx \frac{|\partial\mathcal{Z}|_{i+1,j+1/2} - |\partial\mathcal{Z}|_{i-1,j+1/2}}{2\Delta\alpha}. \tag{6.96}$$

To maintain fourth-order-accuracy, the pointwise fluxes in (6.93) and (6.94) must be evaluated to at least $O(\Delta\alpha^4)$ and flux derivative terms must be evaluated at $O(\Delta\alpha^2)$. Hence, it is sufficient to make use of the compact second-order derivative formulae

$$\left(\frac{\partial\mathbf{F}}{\partial\beta}\right)_{i+1/2,j} \approx \frac{\mathbf{F}_{(0)i+1/2,j+1} - \mathbf{F}_{(0)i+1/2,j-1}}{2\Delta\alpha}, \tag{6.97}$$

$$\left(\frac{\partial^2\mathbf{F}}{\partial\beta^2}\right)_{i+1/2,j} \approx \frac{\mathbf{F}_{(0)i+1/2,j+1} - 2\mathbf{F}_{(0)i+1/2,j} + \mathbf{F}_{(0)i+1/2,j-1}}{\Delta\alpha^2}, \tag{6.98}$$

$$\left(\frac{\partial\mathbf{F}}{\partial\alpha}\right)_{i,j+1/2} \approx \frac{\mathbf{F}_{(0)i+1,j+1/2} - \mathbf{F}_{(0)i-1,j+1/2}}{2\Delta\alpha}, \tag{6.99}$$

$$\left(\frac{\partial^2\mathbf{F}}{\partial\alpha^2}\right)_{i,j+1/2} \approx \frac{\mathbf{F}_{(0)i+1,j+1/2} - 2\mathbf{F}_{(0)i,j+1/2} + \mathbf{F}_{(0)i-1,j+1/2}}{\Delta\alpha^2}. \tag{6.100}$$

To achieve fourth-order accuracy overall, source terms of the horizontal momentum equations must be evaluated to at least third-order accuracy. As with edge fluxes, simply evaluating the source terms using cell-averaged values $\overline{\overline{\mathbf{q}}}_{i,j,k}$ only leads to a second-order discretization. Hence, MCore follows a convolution and deconvolution strategy that leads to a fourth-order-accurate approximation of the source terms. To begin, source terms are first evaluated at element centerpoints ($\psi_{(0)i,j} = \psi(\mathbf{q}_{(0)i,j})$). Notably, the state vector $\mathbf{q}_{(0)i,j}$ has already been calculated at this point during reconstruction using (6.50). Now, to obtain a fourth-order accurate approximation to the forcing term within this element we apply a convolution operator of the form

$$\overline{\overline{\psi}}_{i,j} = \psi_{(0)i,j} + \frac{\Delta\alpha^2}{24}\left[\left(\frac{\partial^2\psi}{\partial\alpha^2}\right)_{i,j} + \left(\frac{\partial^2\psi}{\partial\beta^2}\right)_{i,j}\right] + \frac{\Delta\alpha^4}{12|\mathcal{Z}|_{i,j}}\left[\frac{\partial\psi}{\partial\alpha}\frac{\partial\tilde{J}}{\partial\alpha} + \frac{\partial\psi}{\partial\beta}\frac{\partial\tilde{J}}{\partial\beta}\right]. \qquad (6.101)$$

As with the flux operators, all first and second derivatives are approximated to second-order accuracy using standard centered finite-difference operators. The derivatives of the vertically-integrated metric Jacobian are approximated by (6.35)-(6.36).

6.4.8 Inclusion of Topography

Topography does not explicitly enter the formulation of MCore, but instead enters implicitly in two ways. First, element volumes and face areas are modified by the presence of topography. This has a direct effect on the calculation of total edge fluxes. Notably, under the shallow-atmosphere approximation the areas of vertical bounding faces are actually unmodified by the presence of topography. The second method topography enters the formulation is via pressure terms that are accounted for by the deorthonormalization matrix. Namely, when fluxes are computed across vertical surfaces the deorthonormalization procedure is responsible for applying pressure forcing appropriately to all components of the momentum. This has the effect of applying pressure forcing to the horizontal momentum components whenever the vertical bounding face is tilted, such as in the presence of topography.

6.4.9 Treatment of Panel Boundaries

One must be careful in the treatment of panel boundaries on the cubed-sphere grid. First, the underlying coordinate system is disjoint at panel boundaries so one must be certain that vector quantities are in the correct reference frame. Second, since the grid is not smooth in the vicinity of the panel boundary a direct application of the reconstruction stencil near grid edges can potentially generate significant numerical noise and reduce the accuracy of the simulation. As a consequence, we advocate the use of halo elements around panels (see Figure 6.1) in combination with an appropriate remapping algorithm.

The remapping process we apply in MCore is identical to the one described in *Ullrich et al.* (2010). Under this approach, halo elements are extended outward from each panel. Sub-grid-scale fourth-order reconstructions are built along each panel edge using one-sided reconstruction stencils and the reconstructions are sampled at Gaussian quadrature points. As a result, we obtain fourth-order-accurate approximations to element averages in halo elements. The usual horizontal reconstruction stencil can then be applied, minimizing the generation of spurious boundary noise.

6.4.10 Rayleigh Friction

It is well-known that atmospheric models that have a rigid-lid upper boundary condition typically suffer significant wave reflection at the model top. The traditional approach to deal with this problem has been to add a sponge layer that is responsible for damping out oscillations in the velocity fields high in the atmosphere. Wave reflection is particularly problematic near steep topography or in the presence of strong vertical motions.

MCore implements an optional Rayleigh friction layer. Since Rayleigh friction is a potentially stiff effect, it is included in the implicit stage of the solver as a source term of the full 3D momentum equation that takes the form

$$\boldsymbol{\psi}_R = -R_c(\alpha, \beta, \xi)(\rho\mathbf{u} - \rho\mathbf{u}_0), \tag{6.102}$$

where R_c denotes the strength of the friction term, $\mathbf{u}$ is the 3D velocity vector and $\mathbf{u}_0$ denotes some appropriate reference state for the velocity. The strength of the friction term can be chosen arbitrarily, but should transition smoothly from zero forcing at lower levels to some maximum at the model top. For simplicity, we choose

$$R_c(\alpha, \beta, \xi) = \begin{cases} 0 & \text{if } \xi < \xi_R, \\ \dfrac{1}{\tau_R}\left(\dfrac{\xi - \xi_R}{1 - \xi_R}\right)^2 & \text{otherwise.} \end{cases} \tag{6.103}$$

Here τ_R is the timescale of the damping and ξ_R is the height of the damping layer in ξ coordinates. By default, we define $\tau_R = 1$ day and $\xi_R = 0.7$, which roughly places the Rayleigh damping layer at an altitude of 20 km.

6.4.11 Design Features

The high-order finite-volume methods of MCore are designed to allow run-time switching between shallow-water, shallow-atmosphere and deep-atmosphere approximations. Further, compile-time switching is available for the choice of time-discretization, the form of the non-hydrostatic evolution equations, the choice of Riemann solver and the form of the horizontal reconstruction. The model attempts to maximize efficiency by pre-computing geometric quantities and storing the corresponding values for the duration of the simulation. To enhance modularity and readability the model is implemented in C++ using classes to isolate functionality.

MCore makes use of the GECoRe remapping package of *Ullrich et al.* (2009) for performing conservative remapping between cubed-sphere and latitude-longitude grids. The data output format can be chosen to be MATLAB-readable .mat files or

NetCDF .nc data.

6.5 Numerical Results

Several test cases have been chosen to demonstrate the robustness and accuracy of MCore, including a baroclinic instabilty, mountain-induced Rossby wave train and a 3D Rossby-Haurwitz wave. Rayleigh friction, as described in section 6.4.10, is only used for the mountain-induced Rossby wave train test case. All test runs make use of a $r_T = 30$ km model top and a vertical grid spacing which we have chosen to be

$$\mathcal{R}(\xi; \alpha, \beta, n_p) = r_s(\alpha, \beta, n_p) + (r_T - r_s(\alpha, \beta, n_p))\tilde{\phi}(\xi), \qquad (6.104)$$

where

$$\tilde{\phi}(\xi) = \frac{1}{\sqrt{b+1} - 1}\left[\sqrt{b\xi^2 + 1} - 1\right]. \qquad (6.105)$$

Recall that ξ specifies the auxiliary vertical coordinate, which is defined on the interval $[0, 1]$ at each point (α, β, n_p). This choice of vertical grid spacing is chosen to match operational conditions, where additional resolution is generally desired near the surface. Here b is a flattening parameter which we choose to be $b = 10$. It determines how quickly model levels tend towards equal spacing at higher altitudes.

All model runs have taken place on a c90 grid with 26 vertical levels. Namely, on each model level the grid consists of the six panels of the cubed-sphere each being subdivided into a 90×90 grid of elements. This choice corresponds to a RLL grid with one degree grid spacing at the equator, although overall the cubed-sphere grid has 25% fewer elements than such a RLL grid. We make use of the RK4 timestepping strategy (6.68)-(6.71) for the explicit integration with a fixed timestep of $\Delta t = 200$ s.

All of the test cases examined in this document are written in terms of spherical coordinates with longitude λ and latitude φ. Coordinate transforms are provided in Appendix G for translating these quantities to the cubed-sphere.

6.5.1 Baroclinic Instability

The baroclinic instability test of *Jablonowski and Williamson* (2006) has become an important test case for hydrostatic dynamical cores. Although this test is idealized, the background field is chosen to closely resemble the known background state of the true atmosphere. As such, this test may shed light into the treatment of real atmospheric motions by the atmospheric model. This test case was originally formulated in so-called η pressure coordinates, which are defined by $\eta = p/p_s$, and so had to be remapped to height coordinates before being used in MCore.

This test utilizes an auxiliary variable η_ν, defined by

$$\eta_\nu = (\eta - \eta_0)\frac{\pi}{2}, \tag{6.106}$$

with $\eta_0 = 0.252$. The flow field for this test consists of two symmetric zonal jets in the midlatitudes, utilizing a wind field defined in spherical coordinates via

$$\begin{aligned} u(\lambda,\varphi) &= u_0 \cos^{\frac{3}{2}} \eta_\nu \sin^2(2\varphi), && (6.107) \\ v(\lambda,\varphi) &= 0, && (6.108) \\ w(\lambda,\varphi) &= 0. && (6.109) \end{aligned}$$

Here the maximum amplitude u_0 is set to 35 m s^{-1}. The horizontally averaged temperature $\overline{T}(\eta)$ is split into representations of the lower and middle atmosphere,

$$\begin{aligned} \overline{T}(\eta) &= T_0 \eta^{\frac{R_d\Gamma}{g}} && \text{(for } \eta \geq \eta_t\text{)}, && (6.110) \\ \overline{T}(\eta) &= T_0 \eta^{\frac{R_d\Gamma}{g}} + \Delta T(\eta_t - \eta)^5 && \text{(for } \eta_{top} > \eta\text{)} && (6.111) \end{aligned}$$

with the tropopause level chosen to be $\eta_t = 0.2$, horizontal mean temperature of the surface $T_0 = 288$ K, lapse rate $\Gamma = 0.005$ K m^{-1} and empirical temperature difference $\Delta T = 4.8 \times 10^5$ K. The total temperature distribution is the sum of the mean temperature distribution and a horizontal variation at each level,

$$\begin{aligned} T(\lambda, \varphi, \eta) \; = \; & \overline{T}(\eta) + \frac{3}{4}\frac{\eta \pi u_0}{R_d} \sin \eta_\nu \cos^{\frac{1}{2}} \eta_\nu \times \\ & \left\{ \left(-2 \sin^6 \varphi \left(\cos^2 \varphi + \frac{1}{3} \right) + \frac{10}{63} \right) 2u_0 \cos^{\frac{3}{2}} \eta_\nu + \right. \\ & \left. \left(\frac{8}{5} \cos^3 \varphi \left(\sin^2 \varphi + \frac{2}{3} \right) - \frac{\pi}{4} \right) a\Omega \right\}. \end{aligned} \tag{6.112}$$

The geopotential $\Phi = gz$ completes the definition of the steady-state conditions. As with the temperature field, we define a horizontally averaged geopotential

$$\overline{\Phi}(\eta) = \frac{T_0 g}{\Gamma} \left(1 - \eta^{\frac{R_d \Gamma}{g}} \right) \qquad \text{(for } \eta \geq \eta_t\text{)} \tag{6.113}$$

$$\overline{\Phi}(\eta) = \frac{T_0 g}{\Gamma} \left(1 - \eta^{\frac{R_d \Gamma}{g}} \right) - R_d \Delta T \times \qquad \text{(for } \eta_{top} > \eta\text{)} \tag{6.114}$$

$$\left\{ \left(\ln \left(\frac{\eta}{\eta_t} \right) + \frac{137}{60} \right) \eta_t^5 - 5\eta_t^4 \eta + 5\eta_t^3 \eta^2 - \frac{10}{3} \eta_t^2 \eta^3 + \frac{5}{4} \eta_t \eta^4 - \frac{1}{5} \eta^5 \right\}.$$

The total geopotential is then determined by

$$\begin{aligned} \Phi(\lambda, \varphi, \eta) \; = \; & \overline{\Phi}(\eta) + u_0 \cos^{\frac{3}{2}} \eta_\nu \times \\ & \left\{ \left(-2 \sin^6 \varphi \left(\cos^2 \varphi + \frac{1}{3} \right) + \frac{10}{63} \right) u_0 \cos^{\frac{3}{2}} \eta_\nu + \right. \\ & \left. \left(\frac{8}{5} \cos^3 \varphi \left(\sin^2 \varphi + \frac{2}{3} \right) - \frac{\pi}{4} \right) a\Omega \right\}. \end{aligned} \tag{6.115}$$

The above formulation describes a state which is in both hydrostatic and geostrophic balance, and hence in a perfect model it should be maintained indefinitely. However, numerical errors due to the discretization will nonetheless creep into the solution and cause the steady nature of this profile to be lost. These errors eventually accumulate and result in the collapse of the flow into turbulent motions which are determined by the discrete properties of the numerical method.

In order to trigger an instability which is more deterministic in its evolution, *Jablonowski and Williamson* (2006) introduce an overlaid perturbation in the zonal velocity field. They select a Gaussian profile centered at $(\lambda_c, \varphi_c) = (\pi/9, 2\pi/9)$, which leads to an instability in the northern jet. The perturbation is Gaussian, given by

$$u'(\lambda, \varphi, \eta) = u_p \exp\left(-\left(\frac{r_c}{R}\right)^2\right), \tag{6.116}$$

with radius $R = a/10$ and maximum amplitude $u_p = 1\text{m s}^{-1}$. Here r_c is the great circle distance

$$r_c = a \arccos\left[\sin\varphi_c \sin\varphi + \cos\varphi_c \cos\varphi \cos(\lambda - \lambda_c)\right]. \tag{6.117}$$

The hydrostatic background state for these simulations is given by the initial conditions. In figure 6.3 we plot snapshots of the baroclinic wave test case at day 7 and 9 showing surface pressure, 850 hPa temperature and 850 hPa relative vorticity. All results are interpolated to pressure levels from heigh levels using a linear interpolant. An intercomparison of the performance of various dynamical cores on this test case can be found in *Lauritzen et al.* (2010). The results from MCore are very competitive with other dynamical cores, correctly capturing the location and strength of pressure minima and maintaining sharp gradients in the relative vorticity field. Of particular note is the fact that our results do not show any visually apparent grid imprinting. That is, there are no clear signs of wave number four forcing which would be triggered by anisotropy in the cubed-sphere grid. Further, we observe no signs of instability in the southern hemisphere which would suggest a significant imbalance in the numerics.

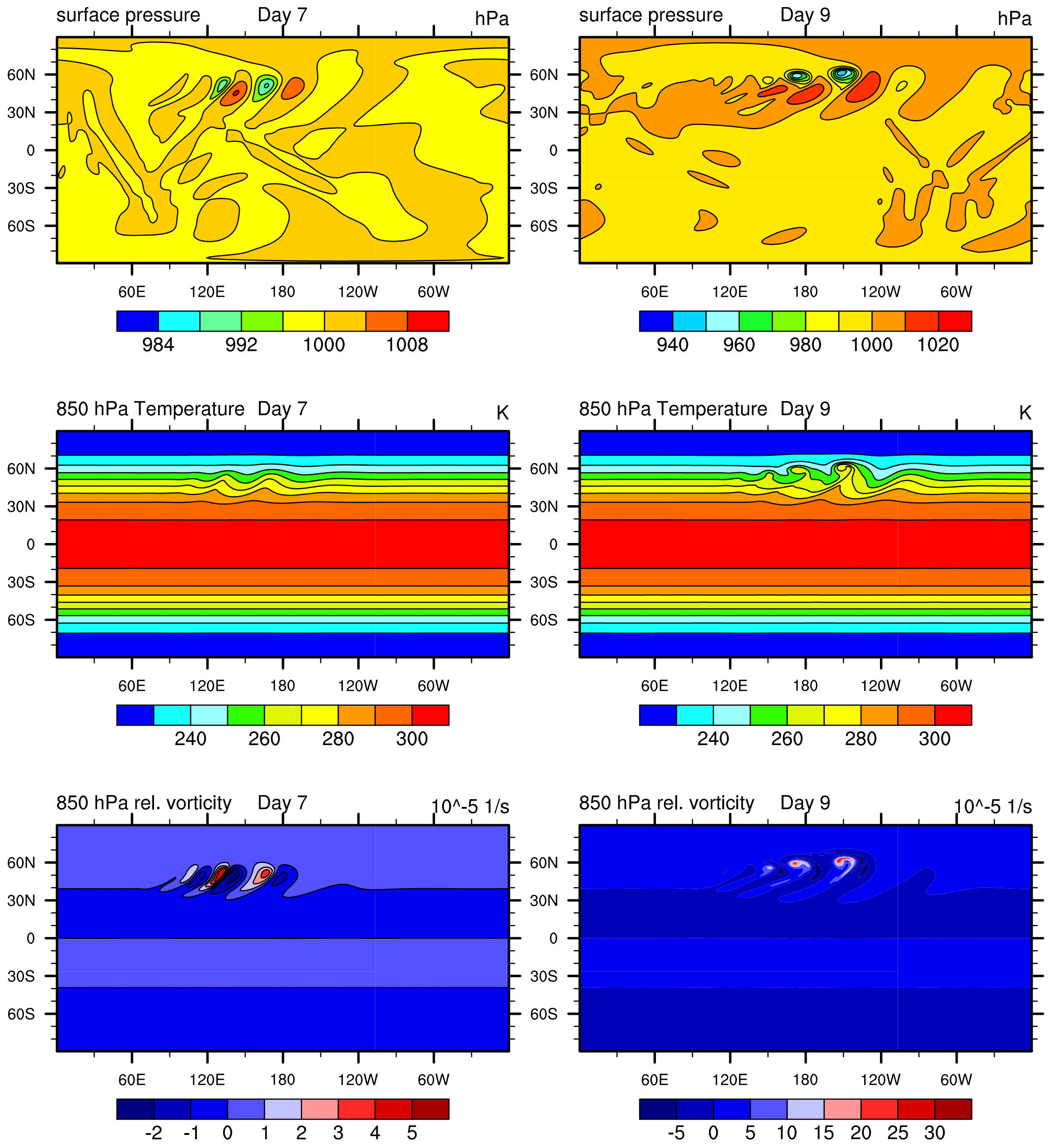

surface pressure Day 7 hPa
surface pressure Day 9 hPa
850 hPa Temperature Day 7 K
850 hPa Temperature Day 9 K
850 hPa rel. vorticity Day 7 10^-5 1/s
850 hPa rel. vorticity Day 9 10^-5 1/s
60N
30N
0
30S
60S
60E
120E
180
120W
60W
984 992 1000 1008
940 960 980 1000 1020
240 260 280 300
-2 -1 0 1 2 3 4 5
-5 0 5 10 15 20 25 30

Figure 6.3: Snapshots from the baroclinic wave test case at day 7 and 9 simulated on a c90 grid with 26 vertical levels and 30 kilometer model cap. Surface pressure is plotted in the upper row, 850 hPa temperature in the middle row and 850 hPa relative vorticity in the bottom row.

6.5.2 3D Rossby-Haurwitz Wave

The Rossby-Haurwitz wave test case is a 3D extension of the 2D shallow-water Rossby-Haurwitz wave described by *Williamson et al.* (1992). The 3D test has also been described in *Giraldo and Rosmond* (2004), although this reference contains known inaccuracies. The form of this test used here is described in *Jablonowski et al.* (2008b). The Rossby-Haurwitz wave is an analytical solution of the barotropic vorticity equation that features an unsteady wave that translates westward at a known velocity. The wave is not an analytic solution of the full non-hydrostatic equations of motion, but under these equations the solution still inherits many characteristic features. The wave still translates westward at a roughly constant velocity and approximately preserves its shape over time. As such, this test is very helpful in determining the ability of a model to maintain this shape over time.

The initial velocity field is non-divergent and defined by the streamfunction

$$\psi_s(\lambda, \varphi) = -a^2 M \sin\varphi + a^2 K \cos^n \varphi \sin\varphi \cos(n\lambda), \tag{6.118}$$

where the parameters M and K are chosen such that $M = K = u_0/(na)$ with $u_0 = 50$ m s^{-1} and $n = 4$. For a non-divergent barotropic model, *Haurwitz* (1940) showed that this streamfunction moves in the zonal direction without change of shape with an angular velocity given by

$$\nu = \frac{n(3+n)M - 2\omega}{(1+n)(2+n)}, \tag{6.119}$$

which for the given choice of parameters corresponds to westward propagation with a period of approximately 24 days. Simulations using other primitive equation models, however, instead yield motions with a period of approximately 26 days.

The horizontal velocity components are vertically uniform and given by

$$\begin{aligned} u(\lambda,\varphi) &= aM\cos\varphi + aK\cos^{n-1}\varphi\cos(n\lambda)(n\sin^2\varphi - \cos^2\varphi), && (6.120)\\ v(\lambda,\varphi) &= -aKn\cos^{n-1}\varphi\sin\varphi\sin(n\lambda). && (6.121)\end{aligned}$$

The vertical temperature profile is characterized by a constant lapse rate,

$$T = T_0 - \Gamma\tilde{z}, \tag{6.122}$$

with $T_0 = 288$ K and $\Gamma = 0.0065\text{K m}^{-1}$. Here $\tilde{z}$ is the equivalent height, defined via

$$\tilde{z} = \frac{T_0}{\Gamma}\left(1 - \left(\frac{p}{p_{\text{ref}}}\right)^{\frac{\Gamma R_d}{g}}\right), \tag{6.123}$$

where p_{ref} is a reference pressure, chosen to be $p_{\text{ref}} = 955$ hPa. The equivalent height and geopotential height are related via the formula

$$\Phi = gz = g\tilde{z} + \Phi'(\lambda,\varphi). \tag{6.124}$$

Solving (6.123) and (6.124) for pressure as a function of height then gives the functional relation

$$p = p_{\text{ref}}\left(1 - \frac{\Gamma(gz - \Phi'(\lambda,\varphi))}{gT_0}\right)^{\frac{g}{R_d\Gamma}}. \tag{6.125}$$

Here Φ' is the geopotential perturbation, defined by

$$\Phi' = a^2\left[A(\varphi) + B(\varphi)\cos(n\lambda) + C(\varphi)\cos(2n\lambda)\right], \tag{6.126}$$

where

$$\begin{aligned}
A(\varphi) &= \frac{M(2\Omega + M)}{2}\cos^2\varphi + \frac{K^2}{4}\cos^{2n}\varphi\left[(n+1)\cos^2\varphi + (2n^2 - n - 2)\right] \\
&\quad -\frac{n^2K^2}{2}\cos^{2(n-1)}\varphi, && (6.127) \\
B(\varphi) &= \frac{2(\Omega + M)K}{(n+1)(n+2)}\cos^n\varphi\left[(n^2 + 2n + 2) - (n+1)^2\cos^2\varphi\right], && (6.128) \\
C(\varphi) &= \frac{K^2}{4}\cos^{2n}\varphi\left[(n+1)\cos^2\varphi - (n+2)\right]. && (6.129)
\end{aligned}$$

The density is recovered from (6.125) via the ideal gas law $\rho = p/(R_dT)$ and potential temperature θ via (6.7).

The hydrostatic background state is chosen to be a profile with constant lapse rate Γ and constant surface temperature T_0. In figure 6.4 we plot a snapshot of the Rossby-Haurwitz wave test case at day 15 showing 850 hPa zonal and meridional wind, surface pressure, 850 hPa temperature, 500 hPa geopotential height and 850 hPa vertical velocity. Using other dynamical cores for comparison (*Lauritzen et al.*, 2010), we note that the wave speed of the Rossby-Haurwitz wave is correctly captured by our method. Further, we observe only small variations in the vertical velocity field, no signs of north-south symmetry breakage and overall observe no obvious signs of instability by day 15.

6.5.3 Mountain-Induced Rossby Wave-train

The mountain-induced Rossby wave train is again an adaptation of a similar shallow-water test case from *Williamson et al.* (1992). The test case used in this chapter most closely resemble that of *Tomita and Satoh* (2004), and is described in detail in *Jablonowski et al.* (2008b). This test begins with smooth isothermal initial conditions that are a balanced analytic solution of the primitive equation in the absence of topography. An idealized mountain then triggers the evolution of a Rossby wave train, which is modeled over a period of 25 days. This test produces significant

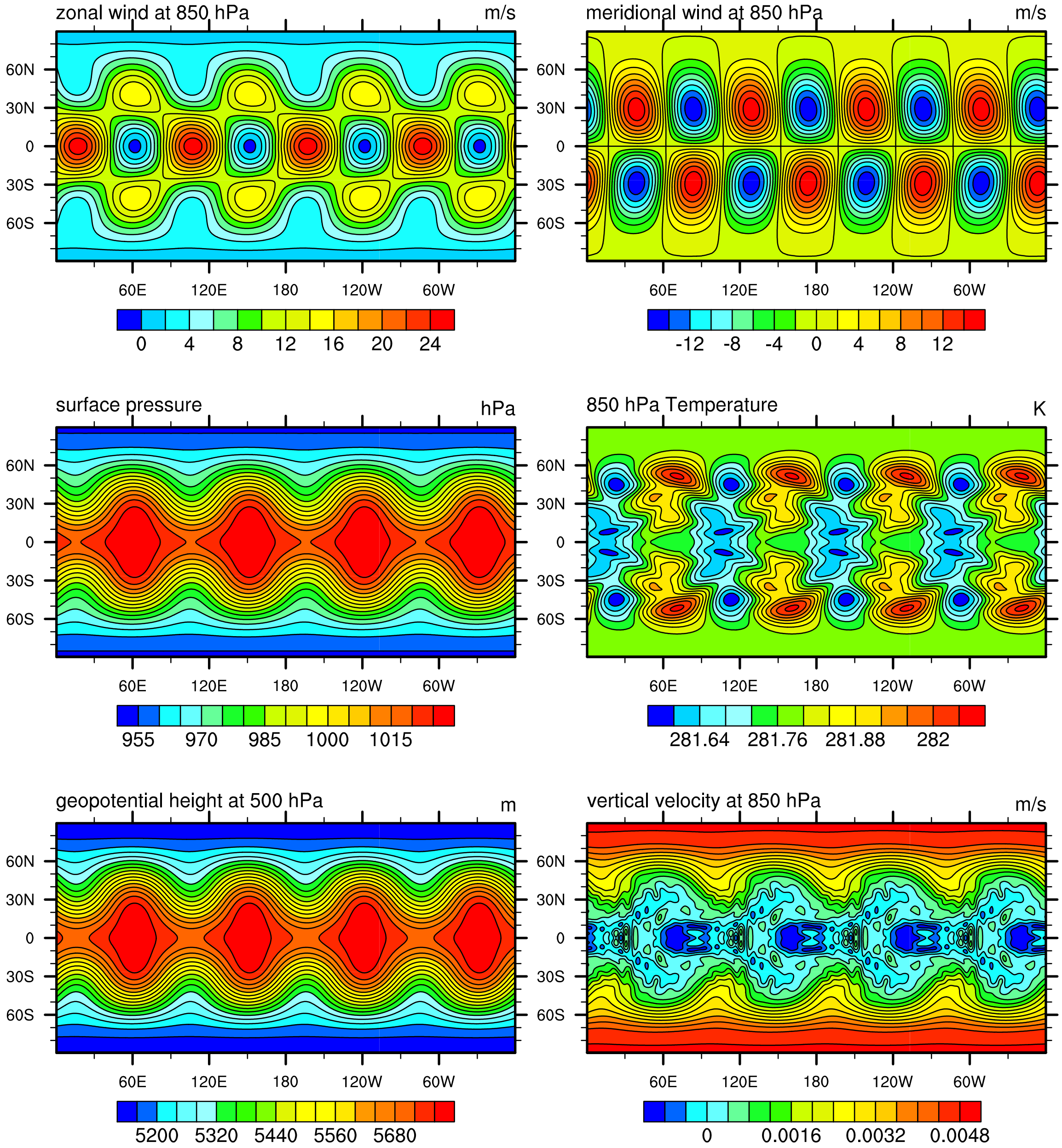


Figure 6.4: Snapshots from the Rossby-Haurwitz wave at day 15 simulated on a c90 grid with 26 vertical levels and 30 kilometer model cap. Zonal and meridional wind (both at 850 hPa) are plotted in the top row, surface pressure and temperature at 850 hPa are shown in the middle row and 500 hPa geopotential height and 850 hPa vertical velocity are plotted in the bottom row.

dynamical motions and so is useful for testing the robustness of the model and its treatment of the model top. The presence of strong vertical motions in the vicinity of the mountain leads to wave reflection in most models at the upper boundary and so a sponge layer is generally needed. There is no known analytic solution to this test case, but several known numerical solutions are available (*Jablonowski et al.*, 2008a). Model solutions tend to diverge observably after day 15 as small-scale numerical oscillations are brought to the grid scale.

The initial components of the horizontal wind in spherical coordinates are

$$u(\lambda, \varphi) = u_0 \cos\varphi, \qquad v(\lambda, \varphi) = 0, \tag{6.130}$$

where the maximum amplitude of the zonal wind u_0 is set to 20 m s^{-1} and the vertical velocity is zero. The atmosphere is initially isothermal with $T_0 = 288$ K, which gives a constant Brunt-Väisälä frequency

$$N = \sqrt{\frac{g}{c_p T_0}} \approx 0.0182 \text{ s}^{-1}. \tag{6.131}$$

An idealized bell-shape mountain is introduced with surface height

$$z_s(\lambda, \varphi) = g h_0 \exp\left[-\left(\frac{r_c}{d}\right)^2 \right], \tag{6.132}$$

where $h_0 = 2000$ m is the peak height of the mountain and $d = 1500$ km is the half-width of the Gaussian profile. Here r_c denotes the great circle distance (6.117) with centerpoint $(\lambda_c, \varphi_c) = (\pi/2, \pi/6)$. The surface pressure p_s is chosen to balance the initial conditions,

$$p_s(\lambda, \varphi) = p_p \exp\left[-\frac{a N^2 u_0}{2 g^2 \kappa} \left(\frac{u_0}{a} + 2\Omega \right) \left(\sin^2\varphi - 1 \right) - \frac{N^2}{g\kappa} z_s(\lambda, \varphi) \right], \tag{6.133}$$

with Pole pressure $p_p = 930$ hPa and $\kappa = R_d/c_p = 2/7$.

The hydrostatic background state is chosen to match the initial conditions. In Figures 6.5 and 6.6 we plot the results over a 25 day simulation period, showing 700 hPa geopotential height, 700 hPa temperature, 700 hPa zonal wind and 700 hPa meridional wind at day 5, 15 and 25. As mentioned previously, Rayleigh damping is applied near the model top. With the Rayleigh damping layer in place we observe rough agreement with known results from hydrostatic models (*Jablonowski et al.*, 2008a). After 15 days uncertainties introduced by the numerical discretization lead to divergence in model solutions and a breakdown of predictability.

6.6 Conclusions and Future Work

In this chapter we have developed a new atmospheric dynamical core which uses high-order finite-volume methods for solving the non-hydrostatic equations of motion under the shallow-atmosphere approximation. The model is built on a cubed-sphere grid with a height-based vertical coordinate. Under the upwind finite-volume methodology, a sub-grid-scale reconstruction is built within each element using neighboring element values. The reconstruction we propose is novel, incorporating geometric terms over the minimal stencil required for fourth-order accuracy. Along edges where the reconstruction is discontinuous we make use of the AUSM$^+$-up Riemann solver of *Liou* (2006) to compute fluxes in the horizontal, and a modified quasi-linear variant of the AUSM$^+$-up scheme in the vertical. To avoid restrictions due to fast vertically propagating wave modes, all terms responsible for vertical motion are computed using an iterative implicit step. The model additionally has the option for Rayleigh damping at the model top to damp out potential wave reflection at the upper boundary.

The resulting method has been tested on a variety of problems and has been shown to be stable, robust and accurate. However, all of the tests that have been applied so far have made use of idealized test cases, which do not necessarily reflect

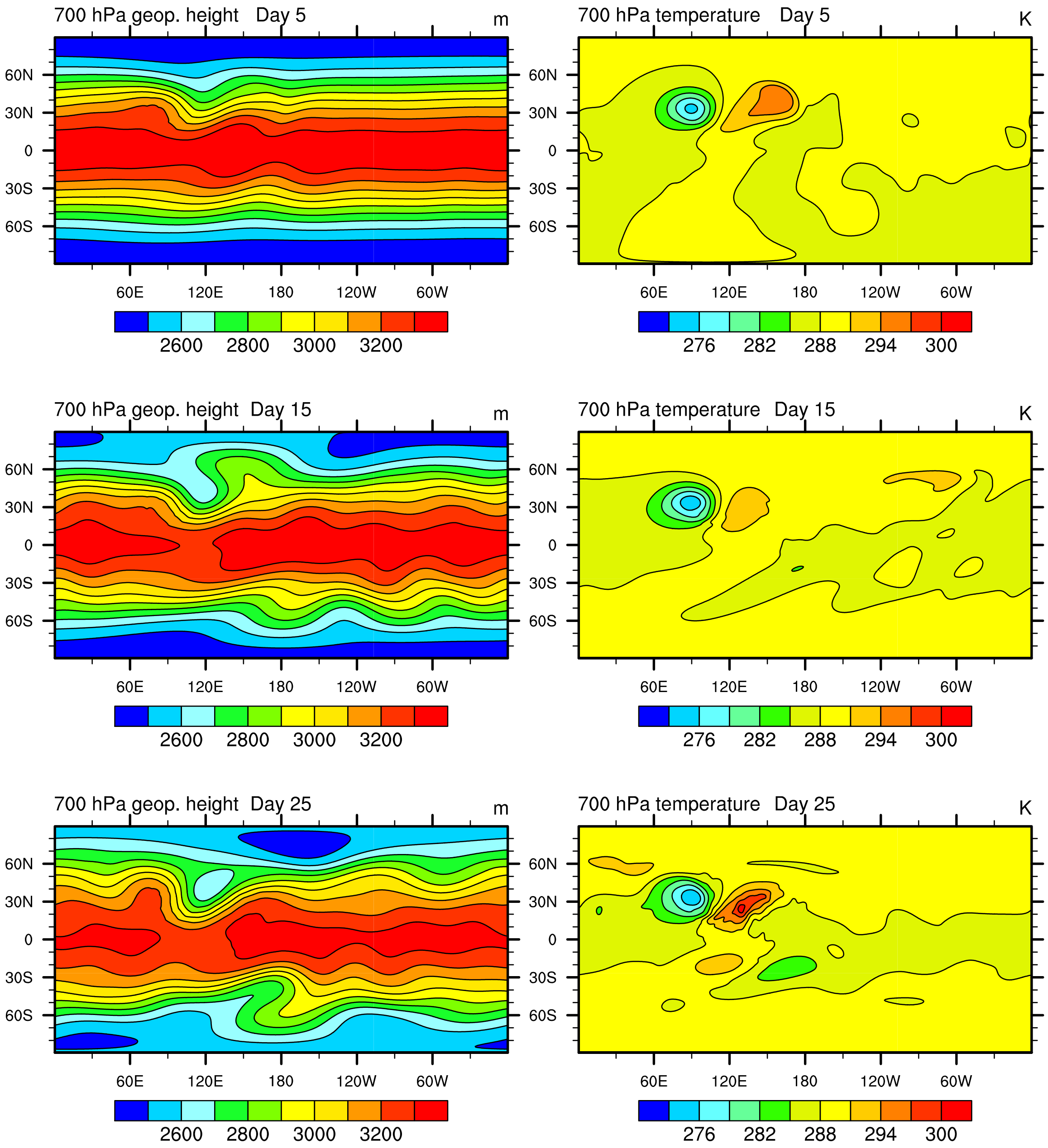


Figure 6.5: Snapshots from the mountain-induced Rossby-wave train wave at day 5 (top row), day 15 (middle row) and day 25 (bottom row) simulated on a c90 grid with 26 vertical levels and 30 kilometer model cap. Geopotential height and temperature at 700 hPa are shown in the left and right column, respectively.

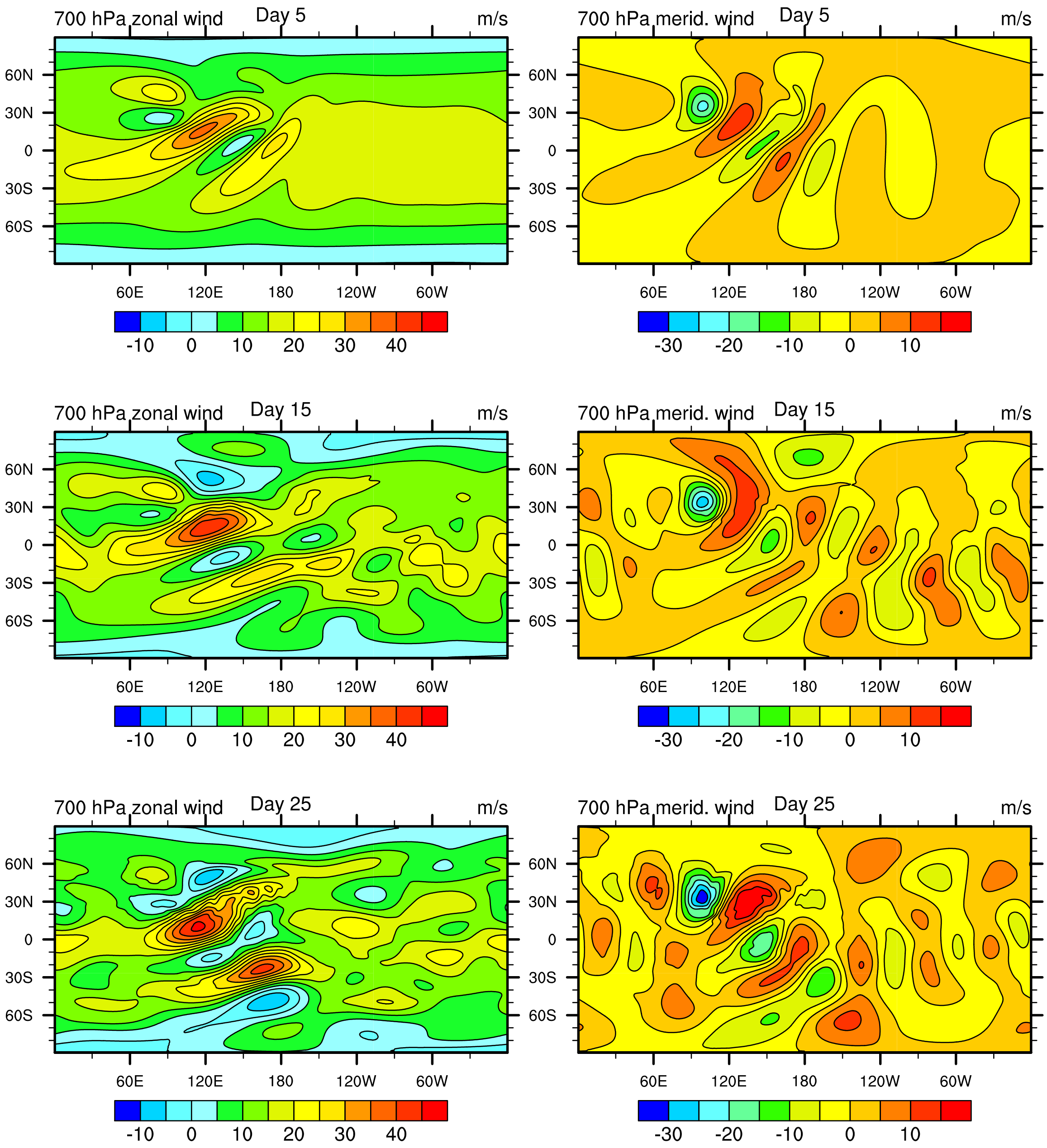


Figure 6.6: Snapshots from the mountain-induced Rossby-wave train wave at day 5 (top row), day 15 (middle row) and day 25 (bottom row) simulated on a c90 grid with 26 vertical levels and 30 kilometer model cap. Zonal and meridional wind at 700 hPa are shown in the left and right column, respectively.

real atmospheric motions. In the future, additional testing is necessary to ensure that the model is able to correctly simulate real atmospheric flows. Nonetheless, we believe that MCore has the potential to be a foundation for the development of next-generation atmospheric models.

MCore is still in the experimental stages and so significant work remains to be done before it can be used operationally. A monotonicity preserving transport scheme still remains to be included in MCore, which will require evaluation and testing of a variety of transport algorithms. Further, model physics must be incorporated in operational versions of the model.

Acknowledgments

Support for this work has been provided by the Office of Science, U.S. Department of Energy, Award No. DE-SC0003990 and a University of Michigan Rackham Predoctoral Fellowship.

CHAPTER VII

Conclusions

Current trends in computing have suggested that the next generation of supercomputers will utilize a massively parallel architecture, with hundreds of thousands of low-cost networked processors working in parallel. Many of the atmospheric models in use today are poorly designed for this hardware infrastructure. As a consequence, the past ten years have seen significant efforts directed at the design of new global atmospheric models for massively parallel hardware architectures. These models are generally referred to as next-generation models, since they have seen significant changes over models of the past decade (*Hack et al.*, 2008; *Washington et al.*, 2008; *Dennis et al.*, 2011).

Traditional models based on the regular latitude-longitude grid usually suffer from poor scalability due to the presence of grid singularities at the poles. As a consequence, mechanisms which are potentially damaging to the accuracy of numerical schemes, such as polar filters, are frequently applied. Polar filters considerably increase the parallel communication overhead on latitude-longitude grids, especially when Fast Fourier Transform (FFT) filtering techniques are employed. These issues are avoided on more uniform grids, such as the icosahedral or cubed-sphere grid. Both of these grids have been shown to be viable choices for next-generation global atmospheric models.

In addition to the choice of grid, numerical methods must also be designed with consideration for scalability on parallel systems. Traditional spectral transform approaches require global communication at each timestep, which is potentially disastrous on massively parallel systems. Hence, numerical methods that use only local information are desirable, including the families of finite-difference, finite-volume or finite-element schemes. In particular, finite-volume methods have been in use for over thirty years throughout many scientific disciplines, and have been shown to be robust, accurate and relatively easy to implement. Further, simple techniques exist for enforcement of positivity and monotonicity in finite-volume methods that simultaneously maintain high-order accuracy, which is necessary for tracer advection.

Existing operational finite-volume-based dynamical cores are formally limited to second-order accuracy for horizontal motions and do not use the full set of conservative equations. A new finite-volume approach has been presented which instead makes use of the fully conservative form of the equations, using Riemann solvers for computing fluxes along element edges. This approach closely follows the MUSCL schemes of *van Leer* (1977).

Summary

A complete software infrastructure for modeling the fluid equations on the sphere has been designed, utilizing high-order (third- and fourth-order) finite-volume methods. This infrastructure includes a set of algorithms for conservative remapping of cell-averaged data between the cubed-sphere and regular latitude-longitude grid. Further, a hierarchy of models have been presented for atmospheric motions that range from microscale to global-scale while essentially leaving the underlying numerical scheme unchanged. The hierarchy includes a 2D shallow-water model, both 2D and 3D non-hydrostatic models and a 3D non-hydrostatic dynamical core in spherical geometry. A cubed-sphere computational grid has been chosen for the global models, due to its relative uniformity as compared with the traditional regular latitude-longitude grid. The results of idealized tests have been very promising, yielding results for these schemes that are either comparable with or improve upon existing models.

In chapter 2, a new approach for conservative remapping between the cubed-sphere and latitude-longitude grids is described. This approach has been designed for finite-volume schemes, where only cell-averages of state variables are known. A reconstruction procedure for quantities on the cubed-sphere grid was also introduced in this section, making use of a primitive remapping algorithm for ghost elements at panel boundaries.

In chapter 3, an analysis of several finite-volume schemes for 1D advection has been presented. The finite-volume models in this paper will be eventually extended to use adaptive mesh refinement for capturing features of particular dynamical interest. This analysis reveals that unstaggered finite-volume methods perform significantly better near refinement boundaries than corresponding staggered schemes.

Chapter 4 introduces the high-order finite-volume model for the shallow-water equations on the sphere. Two reconstructions were tested, including a third-order dimension-split method and a fourth-order method on a wide stencil. Three Riemann solvers were tested for computing fluxes at element boundaries, and it was found that the AUSM$^+$-up solver produces the best results. This solver has been designed for low-Mach number flow, and so is a natural method to apply for atmospheric motions. The equations of motion in cubed-sphere geometry were also presented in this chapter.

One difficulty with finite-volume schemes on the cubed-sphere is in the vicinity of panel edges, where the reconstruction stencil requires information which is not defined in the local coordinate system. The use of neighboring panel information without remapping is potentially damaging to the overall accuracy of the scheme and can lead to “grid imprinting,” whereby waves at the natural frequency of the grid are unnaturally enhanced (for the cubed-sphere, this corresponds to wavenumber four).

High-order accuracy of the numerical scheme can help alleviate these issues, but an effective and accurate method for remapping information between panels is required. Such an approach has been proposed which simultaneously maintains high-order-accuracy and consequently reduces grid imprinting.

In chapter 5 a model for integrating the non-hydrostatic equations of motion in either a 2D or 3D Cartesian domain was introduced. In atmospheric modeling there is a known disparity between horizontal and vertical grid scales, with vertical grid spacing sometimes a factor of 100 or more smaller than horizontal grid spacings. As the maximum stable timestep in compact numerical methods is generally restricted by the minimum grid spacing, the vertical grid spacing is the limiting factor on the timestep. Without special treatment, atmospheric models would be limited to a timestep of only a few seconds, which would make most long-term integrations computationally infeasible. A new approach has been proposed that makes use of implicit-explicit Runge-Kutta-Rosenbrock (IMEX-RKR) integrators to separate the terms for horizontal and vertical motions in the primitive equations. As a consequence, the vertical flow terms are handled implicitly and no longer play a role in limiting the overall model timestep. The AUSM$^+$-up Riemann solver that was studied in Chapter 4 was again used for calculating horizontal fluxes. In the vertical a modified quasi-linear form of the AUSM$^+$-up Riemann solver was introduced which had a simple analytic Jacobian. The known analytic Jacobian allowed for straightforward and efficient iteration when solving the implicit equation set.

In chapter 6 the previously introduced shallow-water model was extended to a fully 3D non-hydrostatic model (MCore) by applying the implicit-explicit integrators introduced in chapter 5. At resolutions below approximately ten kilometers the hydrostatic atmosphere approximation is no longer valid and the acceleration of the vertical velocity becomes increasingly important. This is quickly becoming an issue for modern numerical models which have been run at resolutions that fall within this domain. As a consequence, numerical methods that can be seamlessly applied at both small and large scales are desirable in future models.

Accomplishments and highlights of the research project

Significance: Next-generation models for atmospheric modeling must be redesigned to preserve scalability on massively parallel architectures. With this fact in mind, a model hierarchy using high-order finite-volume methods to ensure high-order accuracy and parallel scalability on a uniform grid has been proposed. In the future, methods such as these will be used for running high-resolution simulations of the atmosphere.

Relevance and future potential: This project has established several novel numerical techniques for simulating atmospheric motions at all scales. These methods will be incorporated in future models for simulating on both uniform and non-uniform adaptive meshes. Possible extensions include upper atmosphere models with prescribed lower and upper boundary conditions, as well as applications in planetary sciences for developing general circulation models of other planets.

Collaboration: The research project has established collaboration between multiple entities, including the Department of Atmospheric, Oceanic and Space Sciences (AOSS) and Department of Aerospace Engineering at the University of Michigan, the National Center for Atmospheric Research (NCAR), and the Applied Numerical Algorithms Group (ANAG) at the Lawrence Berkeley National Laboratory. In particular, contributors to this research project have included Christiane Jablonowski[1], Peter Lauritzen and Ram Nair[2], Bram van Leer[3] and Phillip Colella[4].

[1] Atmospheric Oceanic and Space Science, University of Michigan, Ann Arbor, Michigan

[2] National Center for Atmospheric Research, Boulder, Colorado

[3] Aerospace Engineering, University of Michigan, Ann Arbor, Michigan

[4] Lawrence Berkeley National Laboratory, Berkeley, California

Interdisciplinary research: Methods and ideas used in this project have been drawn from the study of atmospheric science, computer science, applied mathematics and aerospace. It combines global climate modeling aspects with concepts from massively parallel computer systems. To implement the model on the cubed-sphere, concepts from differential equations on a manifold from differential geometry are used. Numerical analysis is further necessary to ensure long-term stable integration of these methods.

Future work

Designing new atmospheric models that achieve parallelism on massively parallel computers has recently become a topic of significant interest. In particular, the upcoming version of the Community Earth System Model (CESM) will use a version of the HOMME dynamical core as its default dynamical core (*Dennis et al.*, 2011). This model has been designed with massive parallelism in mind and, similar to the work of this thesis, is implemented on a cubed-sphere grid. This dissertation proposes a dynamical core that achieves high-order-accuracy on massively parallel systems, and further has the potential for exceptional performance on adaptively refined grids. Incorporation of physical parameterizations into the non-hydrostatic model will be one of the next steps needed to bring this model towards operational status. Further, the approach used for this model will be incorporated into the Chombo software library (*Colella et al.*, 2009), which implements data structures and algorithms for adaptive mesh refinement. Both testing of different finite-volume reconstructions and the design of an effective tracer transport scheme that works in 3D remains to be done. The models proposed in this dissertation should also be compared with existing high-order-accurate schemes in order to determine differences that arise from our approach. A rigorous assessment of the computational efficiency and its parallel computing characteristics should also be performed to verify scalability of the models.

APPENDICES

APPENDIX A

Calculation of anti-derivatives

In general, for interpolation between the RLL grid and ABP grid, we must evaluate the contour integral (2.17) along four types of line segments:

(1) **Lines of constant α.** Since $d\alpha = 0$, any integral along a line of constant α will always evaluate to zero. Hence lines of constant α can be ignored in the computation.

(2) **Lines of constant β.** In this case the integral can be evaluated directly from (2.18), on taking $\beta =$ constant.

(3) **Lines of constant longitude (λ).** In this case we can rewrite the integrand of (2.18) in terms of α and λ via the RLL and ABP coordinate relations given in (D.1) and (D.3) and then integrate to obtain a closed-form expression. Alternatively, transforming these integrals to make use of θ as the dummy variable and then rewriting all α and β components in terms of λ and θ may be helpful when performing the integration. Further, observe that lines of constant λ are lines of constant α on panels 1-4, and hence the resulting integral will evaluate to zero.

(4) **Lines of constant latitude (θ).** In this case we can rewrite the integrand of (2.18) in terms of α and θ via the RLL and ABP coordinate relations given in (D.1) and (D.3) and then integrate to obtain a closed-form expression. Alternatively, it may be helpful to make use of λ as the dummy variable and rewrite all α and β components in terms of λ and θ prior to integrating.

On performing the integration for each of cases (2)-(4), we then obtain antiderivatives $I_s^{(p,q)}$ that can be evaluated at the endpoints of each line segment in order to give a numerically computed line integral. In Cartesian geometry computing exact line integrals of polynomials is straightforward (see *Bockman* 1989), however, on the sphere the integration is non-trivial and is hence performed using the computational mathematics software Maple and then simplified by hand. In the following sections we provide $I_s^{(p,q)}$ for $p + q < 3$, as required for constructing remapping schemes up to third-order accuracy as well as the associated sub-grid-cell reconstruction functions. Extensions to fourth order and beyond can be obtained via the process of integration also described below.

A.1 First-order scheme

We now turn our attention to the piecewise-constant GECoRe scheme as an example of the requisite calculations. Under this scheme, the sub-grid cell reconstruction for each cell on the source grid is given by the constant value $a_n^{(0,0)}$, *i.e.*

$$\overline{f}_n = a_n^{(0,0)}, \tag{A.1}$$

and hence the remapping scheme (2.8) and (2.9) reduces to

$$\overline{f}_k = \frac{1}{A_k} \sum_{n=1}^{N} a_n^{(0,0)} \int\limits_{A_{nk}} dA. \tag{A.2}$$

One can quickly observe that for this scheme we only require knowledge of the area covered by the overlapping regions, which is given by the interior integral term. Hence, in order to apply the divergence theorem and rewrite the area integral in terms of line integrals we require knowledge of the potential $\mathbf{\Psi}^{(0,0)}$ associated with the constant field $\phi = 1$. A simple calculation using (2.11) gives

$$\frac{\Psi_\beta}{\rho \cos\alpha} = \frac{\tan\beta}{\rho}. \tag{A.3}$$

The area integral is then evaluated via (2.17), where the antiderivatives $I_s^{(0,0)}$ follow from (2.18), and hence are given by

$$I_s^{(0,0)} = \int \frac{\tan\beta}{\rho} d\alpha. \tag{A.4}$$

Integration of this quantity is then performed over each of the line segments described previously. For this constant field, we obtain the following closed-form relations:

Lines of constant β

$$I_s^{(0,0)} = -\arccos(\sin\alpha \sin\beta). \tag{A.5}$$

Lines of constant λ (Panel 5 and 6)

$$I_s^{(0,0)} = \text{sign}(\sin\lambda \sin\theta) \arcsin(\cos\lambda \cos\alpha). \tag{A.6}$$

Lines of constant θ (Panel 1 - 4)

$$I_s^{(0,0)} = \alpha \sin\theta. \tag{A.7}$$

Lines of constant θ (Panel 5 and 6)

$$I_s^{(0,0)} = \text{sign}(\sin\theta)\arctan\left(\frac{\tan\alpha}{\tan\beta\sin\theta}\right) - \lambda\sin\theta. \tag{A.8}$$

A.2 Second-order scheme

We now turn our attention to the higher-order GECoRe schemes, beginning with the second-order scheme, which is based on a piecewise-linear reconstruction. First, the anti-derivatives associated with the background fields used in these high-order schemes are presented without derivation. Second, we provide a derivation of the reconstruction coefficients for this scheme.

A.2.1 Anti-derivatives

For the second-order-accurate reconstruction, following (2.16), we must find potentials $\mathbf{\Psi}^{(1,0)}$ and $\mathbf{\Psi}^{(0,1)}$ that satisfy

$$\nabla\cdot\mathbf{\Psi}^{(1,0)} = \tan\alpha, \quad \text{and} \quad \nabla\cdot\mathbf{\Psi}^{(0,1)} = \tan\beta, \tag{A.9}$$

where the divergence operator is defined in accordance with (2.11). From (2.16) we obtain

$$\frac{\Psi_\beta^{(1,0)}}{\rho\cos\alpha} = \frac{\tan\alpha\tan\beta}{\rho}, \quad \text{and} \quad \frac{\Psi_\beta^{(0,1)}}{\rho\cos\alpha} = -\frac{1}{\rho\cos^2\alpha}. \tag{A.10}$$

The antiderivatives of these expressions then lead to $I_s^{(1,0)}$ and $I_s^{(0,1)}$, when evaluated along lines of constant β, constant latitude and constant longitude (recall the antiderivatives along lines of constant α are zero):

Lines of constant β

$$I_s^{(1,0)} = -\text{arcsinh}(\tan\beta\cos\alpha), \tag{A.11}$$

$$I_s^{(0,1)} = -\text{arcsinh}(\tan\alpha\cos\beta). \tag{A.12}$$

Lines of constant λ (Panels 5 and 6)

$$I_s^{(1,0)} = \text{sign}(\sin\theta)\left[\text{arctanh}(\cos\lambda\cos\theta) - \cos\lambda\text{arctanh}(\cos\theta)\right], \tag{A.13}$$

$$I_s^{(0,1)} = -\sin\lambda\text{arctanh}(\cos\theta). \tag{A.14}$$

Lines of constant θ (Panels 1 - 4)

$$I_s^{(1,0)} = -\sin\theta\ln(\cos\alpha), \tag{A.15}$$

$$I_s^{(0,1)} = -\cos\theta\ln(\sec\alpha + \tan\alpha). \tag{A.16}$$

Lines of constant θ (Panels 5 and 6)

$$I_s^{(1,0)} = \text{sign}(\sin\theta)\left[-\cos\lambda\cos\theta + \text{arctanh}(\cos\lambda\cos\theta)\right], \tag{A.17}$$

$$I_s^{(0,1)} = -\sin\lambda\cos\theta. \tag{A.18}$$

A.2.2 Piecewise linear (second-order) reconstruction

We can extend upon the first-order reconstruction by including a linear term in the sub-grid scale reconstruction. That is, for each cell n on the source grid, the field f_n, from (2.5), takes the form

$$f_n(x,y) = a_n^{(0,0)} + a_n^{(1,0)}(x - x_0) + a_n^{(0,1)}(y - y_0). \tag{A.19}$$

For the second-order method, the reconstruction coefficients $a_n^{(i,j)}$ obtained from the Taylor series expansion are clearly the most natural choice. Recall that we can write the Taylor series expansion about $\mathbf{x}_0$ as[1]

$$f_n = \overline{f}_n + \left(\frac{\partial f}{\partial x}\right)_n (x - x_0) + \left(\frac{\partial f}{\partial y}\right)_n (y - y_0). \tag{A.20}$$

Hence, on comparing (A.19) and (A.20) we are inclined to choose the reconstruction coefficients according to

$$a_n^{(0,0)} = \overline{f}_n, \quad a_n^{(1,0)} = \left(\frac{\partial f}{\partial x}\right)_n, \quad a_n^{(0,1)} = \left(\frac{\partial f}{\partial y}\right)_n. \tag{A.21}$$

It can quickly be verified that this choice satisfies the area-averaged field constraint (2.7), and hence is a valid choice of reconstruction coefficients. Computationally, each of these coefficients can then be easily approximated via a discretized derivative operator. Numerous possibilities exist for the choice of discretized derivative operator that vary in both order and stencil size. We refer the reader to *Chung* (2002) for a list of possible discrete operators. The discretized derivative operator must be at least first-order accurate so as to obtain a second-order method, and must correspondingly increase in order for increasingly higher order schemes.

Note that the second-order scheme requires knowledge of the area integrals (or line segment weights) for the fields $x = \tan\alpha$ and $y = \tan\beta$, in addition to the constant field, for a total of three weights per line segment.

A.3 Third-order scheme

In this section we provide the anti-derivatives and reconstruction coefficients required for the third-order-accurate piecewise-parabolic scheme.

[1] Observe that (A.20) is analogous to *Jones* (1999) equations (5).

A.3.1 Anti-derivatives

For the third-order-accurate reconstruction, following (2.16), we must find potentials $\mathbf{\Psi}^{(2,0)}$, $\mathbf{\Psi}^{(0,2)}$ and $\mathbf{\Psi}^{(1,1)}$ that satisfy

$$\nabla \cdot \mathbf{\Psi}^{(2,0)} = \tan^2\alpha, \qquad \nabla \cdot \mathbf{\Psi}^{(0,2)} = \tan^2\beta, \qquad \nabla \cdot \mathbf{\Psi}^{(1,1)} = \tan\alpha\tan\beta. \tag{A.22}$$

From (2.16) we find

$$\frac{\Psi_\beta^{(2,0)}}{\rho\cos\alpha} = \frac{\tan^2\alpha\tan\beta}{\rho}, \tag{A.23}$$

$$\frac{\Psi_\beta^{(0,2)}}{\rho\cos\alpha} = \frac{1}{\cos^2\alpha}\left[-\frac{\tan\beta}{\rho} + \operatorname{arcsinh}\left(\tan\beta\cos\alpha\right)\right], \tag{A.24}$$

$$\frac{\Psi_\beta^{(1,1)}}{\rho\cos\alpha} = -\frac{\tan\alpha}{\rho\cos^2\alpha}. \tag{A.25}$$

The antiderivatives of these expressions then lead to $I_s^{(2,0)}$, $I_s^{(0,2)}$ and $I_s^{(1,1)}$, when evaluated along the appropriate line segments:

Lines of constant β

$$I_s^{(2,0)} = \tan\beta\ \operatorname{arcsinh}(\cos\beta\tan\alpha) + \arccos(\sin\alpha\sin\beta), \tag{A.26}$$

$$I_s^{(0,2)} = \tan\alpha\ \operatorname{arcsinh}(\cos\alpha\tan\beta) + \arccos(\sin\alpha\sin\beta), \tag{A.27}$$

$$I_s^{(1,1)} = -\rho. \tag{A.28}$$

Lines of constant λ (Panels 5 and 6)

$$I_s^{(2,0)} = -\frac{\sin\lambda\cos\lambda}{\sin\theta} + \arctan\left(\frac{\tan\lambda}{\sin\theta}\right), \tag{A.29}$$

$$I_s^{(0,2)} = -\frac{\sin\lambda\cos\lambda}{\sin\theta} - \sin\lambda\cot\theta\ \operatorname{arctanh}(\cos\lambda\cos\theta) - \arctan(\cot\lambda\sin\theta), \tag{A.30}$$

$$I_s^{(1,1)} = -\frac{\sin^2\lambda}{|\sin\theta|}. \tag{A.31}$$

Lines of constant θ (Panels 1 - 4)

$$
\begin{aligned}
I_s^{(2,0)} &= \sin\theta(\tan\alpha - \alpha), && \text{(A.32)}\\
I_s^{(0,2)} &= \tan\alpha(-\sin\theta + \operatorname{arcsinh}(\tan\theta)), && \text{(A.33)}\\
I_s^{(1,1)} &= -\frac{\cos\theta}{\cos\alpha}. && \text{(A.34)}
\end{aligned}
$$

Lines of constant θ (Panels 5 and 6)

$$
\begin{aligned}
I_s^{(2,0)} &= -\frac{\cos^2\theta}{2\sin\theta}\sin\lambda\cos\lambda - \lambda\sin\theta\left(\tfrac{1}{2}\cos^2\theta + 1\right) + \arctan\left(\frac{\tan\lambda}{\sin\theta}\right), && \text{(A.35)}\\
I_s^{(0,2)} &= \frac{\cos^2\theta}{2\sin\theta}\sin\lambda\cos\lambda + \frac{\lambda}{\sin\theta}\left(\tfrac{1}{2}\cos^2\theta - 1\right) \\
&\quad - \cot\theta\sin\lambda\ \operatorname{arctanh}(\cos\lambda\cos\theta) + \arctan\left(\frac{\tan\lambda}{\sin\theta}\right), && \text{(A.36)}\\
I_s^{(1,1)} &= -\tfrac{1}{2}|\sin\theta|\tan^2\alpha. && \text{(A.37)}
\end{aligned}
$$

A.3.2 Piecewise parabolic (third-order) reconstruction

We can devise a third-order scheme by including parabolic terms in the sub-grid scale reconstruction. That is, for each cell n on the source grid, the field f_n takes the form

$$
\begin{gathered}
f_n(x,y) = a_n^{(0,0)} + a_n^{(1,0)}(x-x_0) + a_n^{(0,1)}(y-y_0) + \\
a_n^{(2,0)}(x-x_0)^2 + a_n^{(1,1)}(x-x_0)(y-y_0) + a_n^{(0,2)}(y-y_0)^2. \qquad \text{(A.38)}
\end{gathered}
$$

As with the second-order method, we begin by writing the Taylor series expansion of f_n about the centroid $\mathbf{x}_0$, obtaining

$$
\begin{aligned}
f_n(x,y) &= f_n(\mathbf{x}_0) + \left(\frac{\partial f}{\partial x}\right)_n (x-x_0) + \left(\frac{\partial f}{\partial y}\right)_n (y-y_0) && \text{(A.39)}\\
&\quad + \tfrac{1}{2}\left(\frac{\partial^2 f}{\partial x^2}\right)_n (x-x_0)^2 + \left(\frac{\partial^2 f}{\partial x \partial y}\right)_n (x-x_0)(y-y_0) + \tfrac{1}{2}\left(\frac{\partial^2 f}{\partial y^2}\right)_n (y-y_0)^2.
\end{aligned}
$$

Observe that in this expansion we have not fixed the value of the source field at $\mathbf{x}_0$ to be equal to $\overline{f}_n$. In fact, one can quickly verify that the choice $f_n(\mathbf{x}_0) = \overline{f}_n$ does not lead to a method consistent with the constraint (2.7). The "correct" choice for $f_n(\mathbf{x}_0)$ is instead obtained by integrating (A.39) over the source volume and rewriting the left-hand side in terms of the area-averaged field $\overline{f}_n$. Following this approach, we find[2]

$$\begin{aligned} a_n^{(0,0)} = f_n(\mathbf{x}_0) = \overline{f}_n + \tfrac{1}{2}\left(\frac{\partial^2 f}{\partial x^2}\right)_n (x_0^2 - m_n^{(2,0)}) \\ + \left(\frac{\partial^2 f}{\partial x \partial y}\right)_n (x_0 y_0 - m_n^{(1,1)}) + \tfrac{1}{2}\left(\frac{\partial^2 f}{\partial y^2}\right)_n (y_0^2 - m_n^{(0,2)}), \end{aligned} \tag{A.40}$$

where $m_n^{(p,q)}$ are the area-averaged moments, defined via

$$m_n^{(p,q)} = \frac{1}{A_n} \int\limits_{A_n} x^p y^q dA. \tag{A.41}$$

The remaining reconstruction coefficients are obtained from the Taylor expansion (A.39),

$$a_n^{(1,0)} = \left(\frac{\partial f}{\partial x}\right)_n, \quad a_n^{(0,1)} = \left(\frac{\partial f}{\partial y}\right)_n, \tag{A.42}$$

$$a_n^{(2,0)} = \tfrac{1}{2}\left(\frac{\partial^2 f}{\partial x^2}\right)_n, \quad a_n^{(1,1)} = \left(\frac{\partial^2 f}{\partial x \partial y}\right)_n, \quad a_n^{(0,2)} = \tfrac{1}{2}\left(\frac{\partial^2 f}{\partial y^2}\right)_n. \tag{A.43}$$

Each of these coefficients must be constructed from a discretized derivative operator of at least second-order, and must increase in order correspondingly for higher order schemes.

The third-order scheme requires knowledge of the area integrals, or line potentials, of the fields $x^2 = \tan^2 \alpha$, $xy = \tan\alpha \tan\beta$ and $y^2 = \tan^2 \beta$, in addition to all potentials from the first and second-order schemes.

[2] This choice reduces our reconstruction to the well-known Piecewise Parabolic Method (PPM) devised by *Colella and Woodward* (1984).

APPENDIX B

High-order bisected element reconstruction

Observe that for each of the high-order schemes described in Appendix A the reconstruction is performed over the Gnomonic coordinates, which are inherently discontinuous between panels. Hence, special consideration must be taken when performing remapping over a RLL source volume which covers two or more panels. The simplest solution to this problem is to divide the RLL cell into two or more cells, bisected by the panel edge. All high-order reconstruction coefficients can be maintained in this case, but the first-order cell-average must be re-evaluated in each sub-cell.

B.1 A second-order accurate bisected element reconstruction

One can obtain a simple second-order approximation that conserves the scalar field by imposing

$$A_1\overline{f}_1 + A_2\overline{f}_2 = A_n\overline{f}_n, \tag{B.1}$$

where $\overline{f}_1$ and $\overline{f}_2$ are the new area-averaged field values in the sub-cells spanning the panel edge and $\overline{f}_n$ is the area-averaged field in the original source cell. In order to solve this equation uniquely for $\overline{f}_1$ and $\overline{f}_2$, we also impose that the reconstructions in each sub-cell must be identical in RLL coordinates, *i.e.*

$$\overline{f}_1 - \overline{f}_2 = \left(\frac{\partial f}{\partial \lambda}\right)_n (\lambda_1 - \lambda_2) + \left(\frac{\partial f}{\partial \theta}\right)_n (\theta_1 - \theta_2). \tag{B.2}$$

These two conditions then lead to a simple second-order approximation that conserves the scalar field, given by

$$\overline{f}_1 = \overline{f}_n + \left(\frac{A_2}{A_n}\right)\left[\left(\frac{\partial f}{\partial \lambda}\right)_n (\lambda_1 - \lambda_2) + \left(\frac{\partial f}{\partial \theta}\right)_n (\theta_1 - \theta_2)\right], \tag{B.3}$$

$$\overline{f}_2 = \overline{f}_n - \left(\frac{A_1}{A_n}\right)\left[\left(\frac{\partial f}{\partial \lambda}\right)_n (\lambda_1 - \lambda_2) + \left(\frac{\partial f}{\partial \theta}\right)_n (\theta_1 - \theta_2)\right], \tag{B.4}$$

Since the reconstruction in Gnomonic coordinates requires that derivatives are aligned along x and y coordinate axes, additional work must be performed in order to rotate the reconstructed derivative in spherical coordinates to the Gnomonic coordinate system. Appendix C gives equations for translating the reconstructed derivatives in Gnomonic coordinates from a reconstruction in RLL coordinates for both the second- and third-order schemes.

B.2 A third-order accurate bisected element reconstruction

A third-order accurate reconstruction for a bisected cell can be obtained as an extension of the method discussed in the previous section, and is necessary for remapping from the RLL grid to CS grid using the PPM scheme. Henceforth, we will use the notation (x_i, y_i) to denote the centroid of each sub-cell ($i \in \{1, 2\}$) in Gnomonic coordinates and (λ_i, θ_i) to denote the centroid in RLL coordinates.

In this case, the second derivatives of the field are inherited from the parent cell in each sub-cell, but the cell-averaged values and first derivatives must be computed separately. Using a simple Taylor series expansion in RLL coordinates, we can write the sub-cell first derivatives as

$$\begin{aligned}
\left(\tfrac{\partial f}{\partial \lambda}\right)_i &= \left(\tfrac{\partial f}{\partial \lambda}\right)_n + \left(\tfrac{\partial^2 f}{\partial \lambda \partial \theta}\right)_n (\theta_i - \theta_n) + \left(\tfrac{\partial^2 f}{\partial \lambda^2}\right)_n (\lambda_i - \lambda_n), \qquad \text{(B.5)} \\
\left(\tfrac{\partial f}{\partial \theta}\right)_i &= \left(\tfrac{\partial f}{\partial \theta}\right)_n + \left(\tfrac{\partial^2 f}{\partial \lambda \partial \theta}\right)_n (\lambda_i - \lambda_n) + \left(\tfrac{\partial^2 f}{\partial \theta^2}\right)_n (\theta_i - \theta_n). \qquad \text{(B.6)}
\end{aligned}$$

However, one cannot simply use a Taylor series to obtain $f(\mathbf{x}_0)$, since we have no guarantee that it would satisfy the conservation constraint given in (B.1). Instead, we impose (B.1), which, in conjunction with the area-averaged constraint (A.40), gives

$$\begin{aligned}
&A_1 a_1^{(0,0)} + A_2 a_2^{(0,0)} = \overline{f}_n (A_1 + A_2) \qquad \text{(B.7)} \\
&+A_1 \left[\tfrac{1}{2} \left(\tfrac{\partial^2 f}{\partial x^2}\right)_1 (x_1^2 - m_1^{(2,0)}) + \left(\tfrac{\partial^2 f}{\partial x \partial y}\right)_1 (x_1 y_1 - m_1^{(1,1)}) + \tfrac{1}{2} \left(\tfrac{\partial^2 f}{\partial y^2}\right)_1 (y_1^2 - m_1^{(0,2)}) \right] \\
&+A_2 \left[\tfrac{1}{2} \left(\tfrac{\partial^2 f}{\partial x^2}\right)_2 (x_2^2 - m_2^{(2,0)}) + \left(\tfrac{\partial^2 f}{\partial x \partial y}\right)_2 (x_2 y_2 - m_2^{(1,1)}) + \tfrac{1}{2} \left(\tfrac{\partial^2 f}{\partial y^2}\right)_2 (y_2^2 - m_2^{(0,2)}) \right].
\end{aligned}$$

A second equation can be obtained by imposing that the sub-cells have the same reconstruction in RLL coordinates, which can be expressed as

$$\begin{aligned}
a_2^{(0,0)} - a_1^{(0,0)} &= \left[\left(\tfrac{\partial f}{\partial \lambda}\right)_2 \lambda_2 - \left(\tfrac{\partial f}{\partial \lambda}\right)_1 \lambda_1 \right] + \left[\left(\tfrac{\partial f}{\partial \theta}\right)_2 \theta_2 - \left(\tfrac{\partial f}{\partial \theta}\right)_1 \theta_1 \right] \qquad \text{(B.8)} \\
&- \tfrac{1}{2} \left(\tfrac{\partial^2 f}{\partial \lambda^2}\right) (\lambda_2^2 - \lambda_1^2) - \left(\tfrac{\partial^2 f}{\partial \lambda \partial \theta}\right) (\lambda_2 \theta_2 - \lambda_1 \theta_1) - \tfrac{1}{2} \left(\tfrac{\partial^2 f}{\partial \theta^2}\right) (\theta_2^2 - \theta_1^2).
\end{aligned}$$

If we define the right-hand side of (B.7) as c_A and the right-hand side of (B.8) as c_B, then the $a_i^{(0,0)}$ in each sub-cell take the form

$$a_1^{(0,0)} = \frac{c_A - A_2 c_B}{A_1 + A_2}, \quad \text{and} \quad a_2^{(0,0)} = \frac{c_A + A_1 c_B}{A_1 + A_2}. \qquad \text{(B.9)}$$

The reconstruction coefficients $a_i^{(0,0)}$ can then be converted back to area-averaged values $\overline{f}_i$ on applying (2.7).

APPENDIX C

The gnomonic cubed-sphere projection

A point on the cubed-sphere in the gnomonic projection is normally given in terms of (x, y, n_p) coordinates, where $x, y \in [-1, 1]$ and $n_p \in \{1, 2, 3, 4, 5, 6\}$. By convention, we choose panels $1-4$ to be along the spherical equator, with panels 5 and 6 centered on the southern and northern pole, respectively. As in (2.1), Gnomonic coordinates are related to equiangular coordinates via the relations

$$x = a \tan \alpha, \quad \text{and} \quad y = a \tan \beta, \tag{C.1}$$

where, without loss of generality, we have chosen $a = 1$. Since Gnomonic coordinates are panel-dependent, the change of coordinates relations are dependent on the choice of panel.

C.1 Panels 1-4 (equatorial panels)

In terms of spherical coordinates, x and y take the form

$$x = \tan \lambda^*, \qquad y = \tan \theta \sec \lambda^*, \tag{C.2}$$

where λ^* is the panel-centric longitude coordinate, defined in terms of the panel k by

$$\lambda^* = \lambda - \tfrac{\pi}{2}(k-1). \tag{C.3}$$

Inverting (C.2) yields

$$\lambda^* = \arctan x, \qquad \theta = \arctan\left(\frac{y}{\sqrt{1+x^2}}\right). \tag{C.4}$$

C.2 Panels 5 and 6 (polar panels)

For simplicity, we define a panel indicator variable

$$k = \mathrm{sign}(\theta). \tag{C.5}$$

Observe that on the south polar panel (5) and the north polar panel (6), the indicator variables takes on the values -1 and $+1$, respectively, over the entire panel.

In terms of spherical coordinates, x and y take the form

$$x = k \sin\lambda \cot\theta, \qquad y = -\cos\lambda\cot\theta. \tag{C.6}$$

Inverting (C.6) yields

$$\lambda = -k\arctan(x/y), \qquad \theta = k \arctan\left(\frac{1}{\sqrt{x^2+y^2}}\right). \tag{C.7}$$

C.3 Change-of-coordinates matrices for high-order schemes

When performing interpolation between RLL and cubed-sphere grids, we are required to obtain values for the reconstruction coefficients in terms of Gnomonic coordinates. Since obtaining the reconstructed derivatives in RLL coordinates is relatively simple, a quick application of the chain rule leads to a change-of-coordinates matrix of the form

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \lambda}{\partial x} & \frac{\partial \theta}{\partial x} \\ \frac{\partial \lambda}{\partial y} & \frac{\partial \theta}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial \lambda} \\ \frac{\partial f}{\partial \theta} \end{bmatrix}. \tag{C.8}$$

Hence, the following matrices allow us to rotate the reconstructed derivatives in RLL coordinates, which can be easily obtained via a discretized derivative operator, to Gnomonic coordinates.

Panels 1 - 4 (Equatorial panels)

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} \cos^2 \lambda & -\frac{1}{4}\sin(2\lambda)\sin(2\theta) \\ 0 & \cos\lambda\cos^2\theta \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial \lambda} \\ \frac{\partial f}{\partial \theta} \end{bmatrix}, \tag{C.9}$$

Panels 5 and 6 (Polar panels)

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} k\cos\lambda\tan\theta & -k\sin\lambda\sin^2\theta \\ \sin\lambda\tan\theta & \cos\lambda\sin^2\theta \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial \lambda} \\ \frac{\partial f}{\partial \theta} \end{bmatrix}. \tag{C.10}$$

For third and higher order schemes, we also require equations that express the second-order Gnomonic derivatives

$$\frac{\partial^2 f}{\partial x^2}, \qquad \frac{\partial^2 f}{\partial x \partial y}, \qquad \frac{\partial^2 f}{\partial y^2},$$

in terms of derivatives in RLL coordinates. Again, applying the chain rule, we find

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x^2} \\ \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial^2 \lambda}{\partial x^2} & \frac{\partial^2 \theta}{\partial x^2} \\ \frac{\partial^2 \lambda}{\partial x \partial y} & \frac{\partial^2 \theta}{\partial x \partial y} \\ \frac{\partial^2 \lambda}{\partial y^2} & \frac{\partial^2 \theta}{\partial y^2} \end{bmatrix}}_{\mathbf{A}^{(1,2)}} \begin{bmatrix} \frac{\partial f}{\partial \lambda} \\ \frac{\partial f}{\partial \theta} \end{bmatrix} + \underbrace{\begin{bmatrix} \left(\frac{\partial \lambda}{\partial x}\right)^2 & 2\frac{\partial \lambda}{\partial x}\frac{\partial \theta}{\partial x} & \left(\frac{\partial \theta}{\partial x}\right)^2 \\ \frac{\partial \lambda}{\partial x}\frac{\partial \lambda}{\partial y} & \frac{\partial \theta}{\partial x}\frac{\partial \lambda}{\partial y} + \frac{\partial \theta}{\partial x}\frac{\partial \lambda}{\partial y} & \frac{\partial \theta}{\partial y}\frac{\partial \theta}{\partial x} \\ \left(\frac{\partial \lambda}{\partial y}\right)^2 & 2\frac{\partial \lambda}{\partial y}\frac{\partial \theta}{\partial y} & \left(\frac{\partial \theta}{\partial y}\right)^2 \end{bmatrix}}_{\mathbf{A}^{(2,2)}} \begin{bmatrix} \frac{\partial^2 f}{\partial \lambda^2} \\ \frac{\partial^2 f}{\partial \lambda \partial \theta} \\ \frac{\partial^2 f}{\partial \theta^2} \end{bmatrix}. \tag{C.11}$$

Evaluating the matrix $\mathbf{A}^{(1,2)}$ on panels 1-4 gives

$$\mathbf{A}^{(1,2)} = \begin{bmatrix} -\cos^2\lambda\sin(2\lambda) & -\frac{1}{2}\cos^2\lambda\sin(2\theta)\left[\cos(2\lambda) - \sin^2\lambda\cos(2\theta)\right] \\ 0 & -\frac{1}{2}\cos\lambda\sin(2\lambda)\cos^2\theta\cos(2\theta) \\ 0 & -\cos^2\lambda\sin(2\lambda)\cos^2\theta \end{bmatrix}. \tag{C.12}$$

On panels 5 and 6 we obtain

$$\mathbf{A}^{(1,2)} = \begin{bmatrix} -\sin(2\lambda)\tan^2\theta & -\sin^2\theta\left[\cos^2\lambda\tan\theta - \sin^2\lambda\sin(2\theta)\right] \\ k\cos(2\lambda)\tan^2\theta & -\frac{1}{2}k\sin(2\lambda)\sin^2\theta\left[\tan\theta + \sin(2\theta)\right] \\ \sin(2\lambda)\tan^2\theta & -\sin^2\theta\left[\sin^2\lambda\tan\theta - \cos^2\lambda\sin(2\theta)\right] \end{bmatrix}. \tag{C.13}$$

The matrix $\mathbf{A}^{(2,2)}$ is trivial to calculate, given (C.9) and (C.10), and hence is not provided here.

APPENDIX D

The equiangular cubed-sphere projection

In this appendix we briefly provide details on coordinate relations and the metric associated with the equiangular cubed-sphere projection. For a more thorough treatment of this material, we refer to *Nair et al.* (2005).

A point on the cubed-sphere in the equiangular projection is normally given in terms of (α, β, n_p) (ABP) coordinates, where $\alpha, \beta \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$ and $n_p \in \{1, 2, 3, 4, 5, 6\}$. By convention, we choose panels $1 - 4$ to be along the spherical equator, with panels 5 and 6 centered on the southern and northern pole, respectively.

D.1 Panels 1 - 4 (equatorial panels)

In terms of spherical coordinates, α and β take the form

$$\alpha = \lambda^*, \qquad \beta = \arctan(\tan\theta \sec\lambda^*), \tag{D.1}$$

where λ^* is the panel-centric longitude coordinate, defined earlier in (C.3). Inverting (D.1) yields

$$\lambda^* = \alpha, \qquad \theta = \arctan(\tan\beta \cos\alpha). \tag{D.2}$$

D.2 Panels 5 and 6 (polar panels)

In terms of spherical coordinates, α and β take the form

$$\alpha = k \arctan(\sin\lambda \cot\theta), \qquad \beta = -\arctan(\cos\lambda \cot\theta). \tag{D.3}$$

Inverting (D.3) yields

$$\lambda = -k \arctan\left(\frac{\tan\alpha}{\tan\beta}\right), \qquad \theta = -\arctan\left(\frac{\cos\lambda}{\tan\beta}\right). \tag{D.4}$$

A useful identity that follows from these relationships is

$$\tan^2\alpha + \tan^2\beta = \cot^2\theta. \tag{D.5}$$

D.3 The equiangular cubed-sphere metric

The equiangular cubed-sphere metric is given by

$$g_{ij} = \frac{1}{\rho^4 \cos^2\alpha \cos^2\beta} \begin{bmatrix} 1+\tan^2\alpha & -\tan\alpha\tan\beta \\ -\tan\alpha\tan\beta & 1+\tan^2\beta \end{bmatrix}, \tag{D.6}$$

where ρ is defined by

$$\rho^2 = 1 + \tan^2\alpha + \tan^2\beta. \tag{D.7}$$

The volume element for this metric is then

$$\sqrt{g} = \sqrt{\det(g_{ij})} = (\rho^3 \cos^2\alpha \cos^2\beta)^{-1}. \tag{D.8}$$

We can use the metric to define unit basis vectors, such that $\mathbf{e}\cdot\mathbf{e} = g_{ij}e^ie^j = 1$.

In terms of the natural basis $\hat{\alpha}$ and $\hat{\beta}$, these can be written as

$$(\mathbf{e}_\alpha) = (\rho^2 \cos^2 \alpha \cos \beta)\hat{\alpha}, \qquad (\mathbf{e}_\beta) = (\rho^2 \cos \alpha \cos^2 \beta)\hat{\beta}. \tag{D.9}$$

The non-orthogonality parameter, which determines the degree or non-orthogonality of the basis vectors at each point on the manifold, is then given by

$$\cos(\phi) = \mathbf{e}_\alpha \cdot \mathbf{e}_\beta = -\sin \alpha \sin \beta. \tag{D.10}$$

APPENDIX E

Treatment of panel boundaries

In this appendix we give a detailed description of the high-order boundary reconstruction process we apply at panel edges. For sake of brevity, we will focus on the third-order reconstruction, but note that a fourth-order reconstruction can be easily obtained using a similar method. The basic steps in our reconstruction process are summarized in Figure E.1.

The first step in the treatment of panel boundaries requires a reconstruction of the form (4.44) to be built on the source panel. For the third-order boundary reconstruction we make use of minimal 3×3 stencils so as to obtain a piecewise-parabolic reconstruction. Increasing the accuracy of the boundary reconstruction via larger stencils or a higher-order reconstruction will generally lead to an increase in the model's global accuracy, but the effect on the overall accuracy of the scheme is generally minor. Since information in ghost regions is not known, elements that are immediately adjacent to panel edges must approximate derivatives perpendicular to the panel edge using one-sided stencils. One such set of stencils used for these reconstructions are depicted in Figure E.2.

In Figure E.2(a) and (c), we make use of one-sided reconstructions for the first

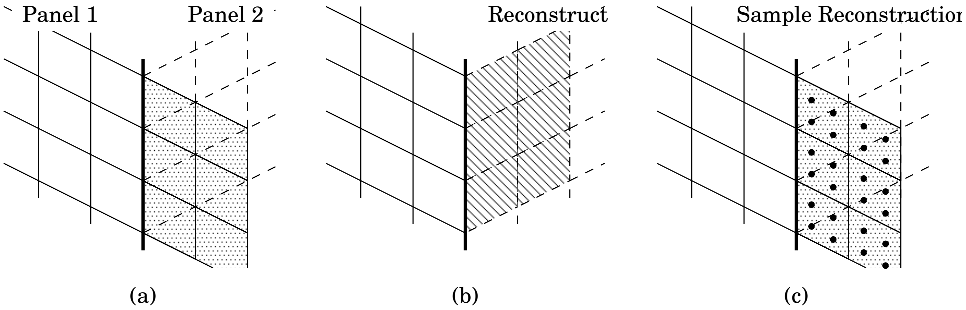


Figure E.1: (a) Reconstruction at panel boundaries is necessitated by the fact that the ghost elements of one panel (Panel 1) do not correspond exactly to elements on a neighboring panel (Panel 2) where element-averages are known exactly. (b) The first step in reconstruction requires one-sided derivative approximations to be calculated on Panel 2 so as to develop a sub-grid-scale reconstruction of the form (4.44). (c) The one-sided reconstructions are then sampled over four Gauss points (per element on Panel 1) so as to ensure high-order accuracy.

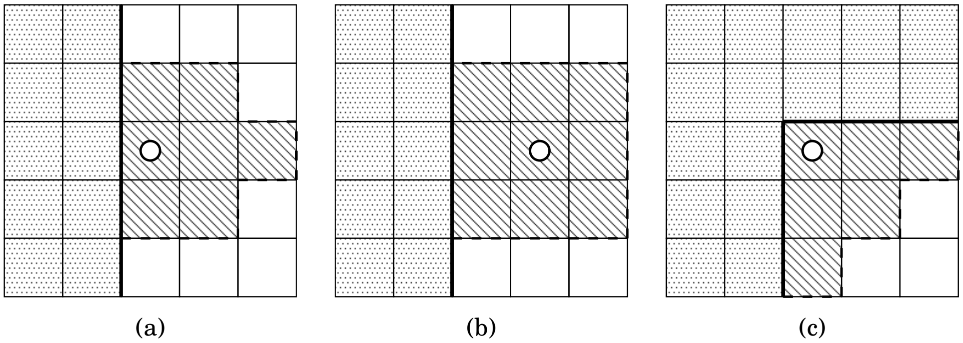


Figure E.2: A set of one-sided stencils for the third-order boundary reconstruction along the left edge. Shading indicates ghost elements, where information is unavailable. Elements used for computing a reconstruction in the specified element are shown with diagonal hatching. Reconstructions along other panel edges can be obtained via a straightforward rotation of the given stencils.

and second derivatives in the α direction,

$$\left(\frac{\delta q}{\delta \alpha}\right)_i = \frac{-3\overline{q}_i + 4\overline{q}_{i+1} - \overline{q}_{i+2}}{2\Delta\alpha} + O(\Delta\alpha^2, \Delta\beta^2), \tag{E.1}$$

$$\left(\frac{\delta^2 q}{\delta \alpha^2}\right)_i = \frac{\overline{q}_i - 2\overline{q}_{i+1} + \overline{q}_{i+2}}{\Delta\alpha^2} + O(\Delta\alpha, \Delta\beta^2). \tag{E.2}$$

In (c) these one-sided approximations must also be utilized for derivatives in the β direction. Cross-derivatives are approximated in reconstruction (a) via

$$\left(\frac{\delta^2 q}{\delta \alpha \delta \beta}\right)_{i,j} = \frac{\overline{q}_{i+1,j+1} - \overline{q}_{i,j+1} - \overline{q}_{i,j-1} + \overline{q}_{i,j-1}}{2\Delta\alpha\Delta\beta} + O(\Delta\alpha, \Delta\beta^2), \tag{E.3}$$

and in (c) by

$$\left(\frac{\delta^2 q}{\delta \alpha \delta \beta}\right)_{i,j} = \frac{\overline{q}_{i+1,j} - \overline{q}_{i,j} - \overline{q}_{i+1,j-1} + \overline{q}_{i,j-1}}{\Delta\alpha\Delta\beta} + O(\Delta\alpha, \Delta\beta). \tag{E.4}$$

Note that in Figure E.2(b) we may utilize standard central reconstructions on a 3×3 stencil to approximate all derivatives.

Once the one-sided reconstruction is established, we can sample each ghost element on the source grid using four-point Gaussian quadrature so as to obtain element averages on the destination grid. Since the reconstruction obtained in the first step is at least third-order accurate, the reconstructed element-averages in each ghost cell will also be third-order accurate.

The process described in this section must be applied whenever boundary information is needed from an adjacent panel. Hence, under a RK3 timestepping scheme, this process must be applied three times per timestep.

APPENDIX F

Converting between η and z coordinates

The initial conditions required for the geostrophically balanced flow (section 5.5.3) and baroclinic instability test cases (section 5.5.4) are given in terms of pressure-based vertical coordinates with $\eta = p/p_0$. To convert these coordinates to height-based z-coordinates, we must implicitly solve a nonlinear equation relating η and z. Here we choose to use the iterative Newton-Raphson strategy, which is given by

$$\eta^{n+1} = \eta^n - \left[\frac{\partial F}{\partial \eta}(x, y, \eta^n)\right]^{-1} F(x, y, \eta^n), \tag{F.1}$$

where $n = 0, 1, 2, \ldots$ is the iteration count. Here the functions F and $\partial F/\partial \eta$ are determined by

$$F(x, y, \eta^n) = -gz + \Phi(x, y, \eta^n), \tag{F.2}$$

$$\frac{\partial F}{\partial \eta}(x, y, \eta^n) = -\frac{R_d}{\eta^n} T(x, y, \eta^n). \tag{F.3}$$

Here Φ and T are given by equations (5.93) and (5.96), respectively. The starting value of $\eta^0 = 10^{-7}$ is used for all Newton iterations, corresponding to a model top of about 100 km. If a higher model top is required, the value of η^0 needs to be decreased. Convergence is deemed to have occurred if $|\eta^{n+1} - \eta^n| \leq 10^{-14}$, and usually takes about 10 iterations in most cases.

To compute pressure p, density ρ and potential temperature θ as a function of position (x, y, z) we first solve for $\eta(x, y, z)$ via the iterative technique and then apply

$$\begin{aligned} p(x,y,\eta) &= \eta(x,y,z)p_0, && \text{(F.4)} \\ \rho(x,y,\eta) &= \frac{p(x,y,\eta)}{R_d T(x,y,\eta)}, && \text{(F.5)} \\ \theta(x,y,\eta) &= T(x,y,\eta)\left(\frac{p_0}{p(x,y,\eta)}\right)^{R_d/c_p}. && \text{(F.6)} \end{aligned}$$

APPENDIX G

Geometric properties of cubed-sphere coordinates

In this appendix we present some of the geometric properties of cubed-sphere coordinates. Further, we provide the mathematical formulae required for transforming point values and vectors between cubed-sphere coordinates and spherical coordinates. We will make ample use of Einstein summation notation, where repeated indices imply summation over that index. Under cubed-sphere coordinates the indices take on values α, β and r, which correspond to the first, second and third coordinate direction.

G.1 The Metric

The metric is identical on each panel of the cubed-sphere grid, but varies depending on the coordinate within each panel. Using either covariant and contravariant indices, the 2D metric on a surface of constant r is defined as follows (note that

$G^{ij} = (G_{ij})^{-1}$).

$$\hat{G}_{ij} = \frac{r^2(1+Y^2)(1+X^2)}{\delta^4} \begin{pmatrix} 1+X^2 & -XY \\ -XY & 1+Y^2 \end{pmatrix}, \tag{G.1}$$

$$\hat{G}^{ij} = \frac{\delta^2}{r^2(1+X^2)(1+Y^2)} \begin{pmatrix} 1+Y^2 & XY \\ XY & 1+X^2 \end{pmatrix}. \tag{G.2}$$

The radial basis vector is everywhere orthogonal to surfaces of constant r, as with spherical coordinates, and has unit length by construction. Hence, the complete metric in 3D can be decomposed into a 2D component along with a unit radial component,

$$G_{ij} = \begin{pmatrix} \hat{G}_{ij} & 0 \\ 0 & 1 \end{pmatrix}, \qquad G^{ij} = \begin{pmatrix} \hat{G}^{ij} & 0 \\ 0 & 1 \end{pmatrix}. \tag{G.3}$$

In curvilinear coordinates, the metric is responsible for determining the length of basis vectors as well as the orthogonality properties of the coordinate system. The inner product of two vectors is defined as

$$\langle \mathbf{v}, \mathbf{w} \rangle = G_{ij} v^i w^j, \tag{G.4}$$

where v^i and w^j denote the components of $\mathbf{v}$ and $\mathbf{w}$ in the cubed-sphere basis. The magnitude of a vector in the cubed-sphere basis can be constructed via the inner product, $|\mathbf{v}| = \langle \mathbf{v}, \mathbf{v} \rangle$. The determinant of the covariant metric determines the infinitesmal volume element via

$$J = \sqrt{\det G_{ij}} = \frac{r^2(1+X^2)(1+Y^2)}{\delta^3}, \qquad d\mathbf{V} = J d\alpha d\beta dr. \tag{G.5}$$

Similarly, the determinants of the cofactor matrices of G_{ij} determine the infinitesmal areas along surfaces where one variable is held constant:

$$J_\alpha = \frac{r(1+Y^2)\sqrt{1+X^2}}{\delta^2} \qquad dA_\alpha = J_\alpha d\beta dr, \tag{G.6}$$

$$J_\beta = \frac{r(1+X^2)\sqrt{1+Y^2}}{\delta^2} \qquad dA_\beta = J_\beta d\alpha dr, \tag{G.7}$$

$$J_r = \frac{r^2(1+X^2)(1+Y^2)}{\delta^3} \qquad dA_r = J_r d\alpha d\beta. \tag{G.8}$$

Other vector operations are similarly defined via the metric and its byproducts. The cross-product of two vectors is defined as

$$(\mathbf{u} \times \mathbf{v})^i = JG^{ij}\epsilon_{jk\ell}u^k v^\ell, \tag{G.9}$$

with third-order permutation symbol $\epsilon_{jk\ell}$, defined via

$$\epsilon_{jk\ell} = \begin{cases} +1 & \text{if } (j,k,\ell) \text{ is } (\alpha,\beta,r),\ (r,\alpha,\beta) \text{ or } (\beta,r,\alpha), \\ -1 & \text{if } (j,k,\ell) \text{ is } (\alpha,r,\beta),\ (r,\beta,\alpha) \text{ or } (\beta,\alpha,r), \\ 0 & \text{otherwise.} \end{cases} \tag{G.10}$$

Under the natural basis the gradient operator is

$$\nabla^i \phi = G^{ij}\frac{\partial \phi}{\partial x^j}, \tag{G.11}$$

and the divergence operator takes the form

$$\nabla_j F^j = \frac{1}{J}\frac{\partial}{\partial x^k}(JF^k), \tag{G.12}$$

with $x^k = (x^\alpha, x^\beta, x^r) = (\alpha, \beta, r)$.

G.2 Christoffel Symbols of the second kind

The Christoffel symbols of the second kind represent the effect of transport of a vector field along coordinate lines. They appear in certain derivative operations applied to tensor fields, such as the divergence of the two-index tensor flux of the momentum. In terms of the metric, they are defined via

$$\Gamma^{i}{}_{jk} = \frac{1}{2} g^{im} \left(\frac{\partial g_{jm}}{\partial x^k} + \frac{\partial g_{km}}{\partial x^j} - \frac{\partial g_{jk}}{\partial x^m} \right). \tag{G.13}$$

In equiangular cubed-sphere coordinates we obtain the following expressions.

$$\Gamma^{\alpha} = \begin{pmatrix} \dfrac{2XY^2}{\delta^2} & \dfrac{-Y(1+Y^2)}{\delta^2} & \dfrac{1}{r} \\ \dfrac{-Y(1+Y^2)}{\delta^2} & 0 & 0 \\ \dfrac{1}{r} & 0 & 0 \end{pmatrix}, \tag{G.14}$$

$$\Gamma^{\beta} = \begin{pmatrix} 0 & \dfrac{-X(1+X^2)}{\delta^2} & 0 \\ \dfrac{-X(1+X^2)}{\delta^2} & \dfrac{2X^2Y}{\delta^2} & \dfrac{1}{r} \\ 0 & \dfrac{1}{r} & 0 \end{pmatrix}, \tag{G.15}$$

$$\Gamma^{r} = \frac{r(1+X^2)(1+Y^2)}{\delta^4} \begin{pmatrix} -(1+X^2) & XY & 0 \\ XY & -(1+Y^2) & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{G.16}$$

Using these definitions, we can write the divergence of a two-index tensor T_{ij} as

$$\nabla_j \boldsymbol{T}^{ij} = \frac{1}{J} \frac{\partial}{\partial x^j} (J \boldsymbol{T}^{ij}) + \Gamma^i_{jk} \boldsymbol{T}^{jk}. \tag{G.17}$$

G.3 Transformation matrices from spherical coordinates

The change of variables matrices are defined via the relation

$$\begin{bmatrix} v^\alpha \\ v^\beta \end{bmatrix} = \mathsf{A} \begin{bmatrix} v^\lambda \\ v^\varphi \end{bmatrix}, \qquad \begin{bmatrix} v^\lambda \\ v^\varphi \end{bmatrix} = \mathsf{A}^{-1} \begin{bmatrix} v^\alpha \\ v^\beta \end{bmatrix}, \tag{G.18}$$

where λ is the longitude (chosen so that $\lambda = 0$ corresponds to $\alpha = 0$ on panel 1) and φ is the latitude ($\varphi = 0$ here corresponds to the equator). Both sets of equations are defined in the natural coordinate basis.

G.4 Equatorial panels

Equatorial panels are denoted by an index $n_p \in \{1, 2, 3, 4\}$, where $n_p = 1$ corresponds to the panel containing the longitude line $\lambda = 0$. The point coordinate transformation between cubed-sphere coordinates and spherical coordinates on these panels is defined by the following relations:

$$\alpha = \lambda - \frac{\pi}{2}(n_p - 1), \qquad \beta = \arctan\left(\frac{\tan\varphi}{\cos\lambda}\right), \tag{G.19}$$

$$\lambda = \alpha + \frac{\pi}{2}(n_p - 1), \qquad \varphi = \arctan\left(\tan\beta\cos\alpha\right). \tag{G.20}$$

On equatorial panels, the change of variables matrices are defined in terms of the gnomonic coordinate (X, Y) as follows:

$$\mathsf{A} = \begin{bmatrix} 1 & 0 \\ \dfrac{XY}{1+Y^2} & \dfrac{\delta^2}{(1+Y^2)\sqrt{1+X^2}} \end{bmatrix}, \tag{G.21}$$

$$\mathsf{A}^{-1} = \begin{bmatrix} 1 & 0 \\ \dfrac{-XY\sqrt{1+X^2}}{\delta^2} & \dfrac{(1+Y^2)\sqrt{1+X^2}}{\delta^2} \end{bmatrix}. \tag{G.22}$$

G.4.0.1 Polar panels

Polar panels are denoted by an index $n_p \in \{5, 6\}$ where the index $p = 5$ corresponds to the north polar panel and the index $n_p = 6$ corresponds to the south polar panel. We define a panel indicator s as

$$s = \begin{cases} 1 & \text{if } n_p = 5, \\ -1 & \text{if } n_p = 6. \end{cases} \tag{G.23}$$

The pointwise coordinate transforms then take the form:

$$\alpha = s \arctan\left(\cot\varphi \sin\lambda\right), \qquad \beta = -\arctan\left(\cot\varphi\cos\lambda\right), \tag{G.24}$$

$$\lambda = -\arctan\left(\frac{\tan\alpha}{\tan\beta}\right), \qquad \varphi = s \ \mathrm{arccot}\left(\sqrt{\tan^2\alpha + \tan^2\beta}\right). \tag{G.25}$$

The pointwise change of variables matrices are defined by:

$$\mathsf{A} = \begin{bmatrix} \dfrac{-sY}{1+X^2} & \dfrac{-s\delta^2 X}{(1+X^2)\sqrt{X^2+Y^2}} \\ \dfrac{sX}{1+Y^2} & \dfrac{-s\delta^2 Y}{(1+Y^2)\sqrt{X^2+Y^2}} \end{bmatrix}. \tag{G.26}$$

$$\mathsf{A}^{-1} = \begin{bmatrix} -\dfrac{sY(1+X^2)}{X^2+Y^2} & \dfrac{sX(1+Y^2)}{X^2+Y^2} \\ -\dfrac{sX(1+X^2)}{\delta^2\sqrt{X^2+Y^2}} & -\dfrac{sY(1+Y^2)}{\delta^2\sqrt{X^2+Y^2}} \end{bmatrix}. \tag{G.27}$$

G.4.1 Coriolis source term

The Coriolis source terms of the momentum equation takes the form

$$\boldsymbol{\psi}_C = -2\boldsymbol{\Omega} \times (\rho \mathbf{u}), \qquad \boldsymbol{\Omega} = \omega \left[\cos\varphi \mathbf{e}_\varphi + \sin\varphi \mathbf{e}_r\right], \tag{G.28}$$

at latitude φ and with latitudinal unit basis vector $\mathbf{e}_\varphi$ and radial basis vector $\mathbf{e}_r$. In terms of cubed-sphere coordinates, $\boldsymbol{\Omega}$ is

$$\boldsymbol{\Omega} = \begin{cases} \dfrac{\omega}{\delta}\left[\dfrac{1+X^2}{r}\left[\mathbf{g}_\alpha + \dfrac{XY}{1+Y^2}\mathbf{g}_\beta\right] + Y\mathbf{g}_r\right] & \text{for equatorial panels } (n_p < 5), \\ \dfrac{s\omega}{\delta}\left[\dfrac{\sqrt{X^2+Y^2}}{r}\left[\dfrac{-Y}{1+X^2}\mathbf{g}_\alpha + \dfrac{X}{1+Y^2}\mathbf{g}_\beta\right] + \mathbf{g}_r\right] & \text{for polar panels } (n_p \geq 5) \end{cases} \tag{G.29}$$

APPENDIX H

The shallow-atmosphere approximation

The shallow-atmosphere approximation has the effect of reducing the vertically varying structure of the atmosphere to a set of stacked layers. Layers are approximated to have radius $r = a$, which removes geometric terms associated with increasing area as altitude increases. This approximation was first introduced by *Phillips* (1966). It is described in conjuction with other consistent approximations of the full non-hydrostatic primitive equations in *White et al.* (2005).

Following the metric formulation described in Appendix G, the shallow-atmosphere approximation follows by simply replacing all instances of r in the deep-atmosphere metric (G.1)-(G.2) with the radius of the Earth a. As a result, the infinitesmal volume element becomes

$$J = \sqrt{\det G_{ij}} = \frac{a^2(1+X^2)(1+Y^2)}{\delta^3}, \qquad d\mathbf{V} = J d\alpha d\beta dz, \tag{H.1}$$

where z is the height above the orography. Infinitesmal areas along surfaces (G.6)-(G.8) likewise become

$$J_\alpha = \frac{a(1+Y^2)\sqrt{1+X^2}}{\delta^2} \qquad dA_\alpha = J_\alpha d\beta dr, \tag{H.2}$$

$$J_\beta = \frac{a(1+X^2)\sqrt{1+Y^2}}{\delta^2} \qquad dA_\beta = J_\beta d\alpha dr, \tag{H.3}$$

$$J_r = \frac{a^2(1+X^2)(1+Y^2)}{\delta^3} \qquad dA_r = J_r d\alpha d\beta. \tag{H.4}$$

The Christoffel symbols are significantly affected by this change, now taking the form

$$\Gamma^\alpha = \begin{pmatrix} \frac{2XY^2}{\delta^2} & \frac{-Y(1+Y^2)}{\delta^2} & 0 \\ \frac{-Y(1+Y^2)}{\delta^2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{H.5}$$

$$\Gamma^\beta = \begin{pmatrix} 0 & \frac{-X(1+X^2)}{\delta^2} & 0 \\ \frac{-X(1+X^2)}{\delta^2} & \frac{2X^2Y}{\delta^2} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{H.6}$$

$$\Gamma^r = 0. \tag{H.7}$$

Finally, conservation of energy requires that so-called “cosine Coriolis terms” be dropped from the momentum evolution equation. As a consequence, all $\mathbf{g}_\alpha$ and $\mathbf{g}_\beta$ dependence of (G.29) is dropped and (G.28) simplifies to

$$\boldsymbol{\psi}_C = -f\mathbf{g}_r \times (\rho\mathbf{u}), \tag{H.8}$$

where

$$f = \frac{2\omega}{\delta}\begin{cases} Y & \text{for equatorial panels } (n_p < 5) \\ s & \text{for polar panels } (n_p \geq 5) \end{cases} \tag{H.9}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

Adcroft, A., C. Hill, and J. Marshall (1997), Representation of Topography by Shaved Cells in a Height Coordinate Ocean Model, *Monthly Weather Review*, *125*, 2293–2315, doi:10.1175/1520-0493(1997)125⟨2293:ROTBSC⟩2.0.CO;2.

Adcroft, A., C. Hill, J. M. Campin, J. Marshall, and P. Heimbach (2004), Overview of the Formulation and Numerics of the MIT GCM, in *Proceedings of the ECMWF seminar series on Numerical Methods, Recent developments in numerical methods for atmosphere and ocean modelling.*, pp. 139–149, Reading, U.K., 6-10 September, 2004.

Ahmad, N., and J. Lindeman (2007), Euler solutions using flux-based wave decomposition, *Int. J. Num. Meth. Fluids*, *54*, 47–72, doi:10.1002/fld.1392.

Alaka, M. (1960), The airflow over mountains, *WMO Tech. Note 34*, 135 pp.

Alcrudo, F., and P. Garcia-Navarro (1993), A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations, *International Journal for Numerical Methods in Fluids*, *16*, 489–505, doi:10.1002/fld.1650160604.

Arakawa, A. (1970), Numerical Simulation of Large-Scale Atmospheric Motions, in *Numerical Solution of Field Problems in Continuum Physics (SIAM-AMS Conference)*, vol. 2, pp. 24–40, American Mathematical Society, Providence, RI.

Arakawa, A., and C. S. Konor (2009), Unification of the Anelastic and Quasi-Hydrostatic Systems of Equations, *Monthly Weather Review*, *137*, 710–726, doi:10.1175/2008MWR2520.1.

Arakawa, A., and V. Lamb (1977), Computational design and the basic dynamical processes of the UCLA general circulation model, *Methods in Computational Physics*, *17*, 173–265.

Ascher, U., S. Ruuth, and R. Spiteri (1997), Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, *Applied Numerical Mathematics*, *25*(2-3), 151–167, doi:10.1016/S0168-9274(97)00056-1.

Bale, D. S. (2002), Wave propagation algorithms on curved manifolds with applications to relativistic hydrodynamics, Ph.D. thesis, University of Washington, 154 pp.

Barad, M., and P. Colella (2005), A fourth-order accurate local refinement method for Poisson's equation, *Journal of Computational Physics*, *209*, 1–18, doi:10.1016/j.jcp.2005.02.027.

Barth, T., and D. Jespersen (1989), The design and application of upwind schemes on unstructured meshes, in *Proc. AIAA 27th Aerospace Sciences Meeting, Reno, NV*, January 9-12, 1989.

Berger, M. J., and P. Colella (1989), Local adaptive mesh refinement for shock hydrodynamics, *Journal of Computational Physics*, *82*, 64–84, doi:10.1016/0021-9991(89)90035-1.

Berger, M. J., and J. Oliger (1984), Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations, *Journal of Computational Physics*, *53*, 484–512, doi:10.1016/0021-9991(84)90073-1.

Bergthorsson, P., B. R. Döös, S. Fryklund, O. Haug, and R. Lindquist (1955), Routine Forecasting with the Barotropic Model, *Tellus*, *7*, 272–76.

Black, T. L. (1994), The New NMC Mesoscale Eta Model: Description and Forecast Examples, *Weather and Forecasting*, *9*, 265–284, doi:10.1175/1520-0434(1994)009⟨0265:TNNMEM⟩2.0.CO;2.

Bockman, S. (1989), Generalizing the formula for areas of polygons to moments, *Amer. Math. Mon.*, *96*(2), 131–132.

Bonaventura, L. (2000), A semi-implicit semi-Lagrangian scheme using the height coordinate for a nonhydrostatic and fully elastic model of atmospheric flows, *Journal of Computational Physics*, *158*, 186–213, doi:10.1006/jcph.1999.6414.

Bonaventura, L., and T. Ringler (2005), Analysis of Discrete Shallow-Water Models on Geodesic Delaunay Grids with C-Type Staggering, *Monthly Weather Review*, *133*, 2351–2373, doi:10.1175/MWR2986.1.

Boris, J. P., and D. L. Book (1973), Flux-Corrected Transport. I. SHASTA, A Fluid Transport Algorithm That Works, *Journal of Computational Physics*, *11*, 38–69, doi:10.1016/0021-9991(73)90147-2.

Brillouin, L. (1960), *Wave Propagation and Group Velocity*, 154 pp., Academic Press (New York).

Cathers, B., and S. Bates (1995), Spurious numerical refraction, *International Journal for Numerical Methods in Fluids*, *21*, 1049–1066, doi:10.1002/fld.1650211104.

Charney, J. G., R. Fjørtoft, and J. von Neumann (1950), Numerical Integration of the Barotropic Vorticity Equation, *Tellus*, *2*, 237–254.

Chen, C., and F. Xiao (2008), Shallow water model on cubed-sphere by multi-moment finite volume method, *Journal of Computational Physics*, *227*, 5019–5044, doi:10.1016/j.jcp.2008.01.033.

Chin, R. C. Y., and G. W. Hedstrom (1982), Scattering of Waves from a Staggered Difference Scheme on a Variable Grid, in *Proc. 10th IMACS World Congress, Montreal, Que.*, pp. 49–51, August 8-13, 1982.

Chung, T. J. (2002), *Computational Fluid Dynamics*, 1012 pp., Cambridge University Press.

Colella, P., and P. R. Woodward (1984), The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations, *Journal of Computational Physics*, *54*, 174–201, doi:10.1016/0021-9991(84)90143-8.

Colella, P., et al. (2009), Chombo Software Package for AMR Applications Design Document, *Tech. rep.*, Lawrence Berkeley National Laboratory.

Collins, W., et al. (2006), The Community Climate System Model version 3 (CCSM3), *Journal of Climate*, *19*, 2122–2143.

Collins, W. D., P. J. Rasch, B. A. Boville, J. J. Hack, J. R. McCaa, D. L. Williamson, J. T. Kiehl, and B. Briegleb (2004), Description of the NCAR Community Atmosphere Model (CAM 3.0). NCAR Tech. Note, NCAR/TN-464+STR.

Côté, J., and A. Staniforth (1990), An accurate and efficient finite-element global model of the shallow-water equations, *Monthly Weather Review*, *118*, 2707–2717.

Côté, J., S. Gravel, A. Méthot, A. Patoine, M. Roch, and A. Staniforth (1998), The Operational CMC MRB Global Environmental Multiscale (GEM) Model. Part I: Design Considerations and Formulation, *Monthly Weather Review*, *126*, 1373–1395, doi:10.1175/1520-0493(1998)126⟨1373:TOCMGE⟩2.0.CO;2.

Courant, R., K. Friedrichs, and H. Lewy (1928), Über die partiellen Differenzengleichungen der mathematischen Physik, *Mathematische Annalen*, *100*(1), 32–74.

Crouzeix, M. (1980), Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques, *Numerische Mathematik*, *35*, 257–276, doi:10.1007/BF01396412.

Davies, T., A. Staniforth, N. Wood, and J. Thuburn (2003), Validity of anelastic and other equation sets as inferred from normal-mode analysis, *Quart. J. Royal Meteor. Soc.*, *129*, 2761–2775, doi:10.1256/qj.02.1951.

Davies, T., M. J. P. Cullen, A. J. Malcolm, M. H. Mawson, A. Staniforth, A. A. White, and N. Wood (2005), A new dynamical core for the Met Office's global and regional modelling of the atmosphere, *Quart. J. Royal Meteor. Soc.*, *131*(608), 1759–1782, doi:10.1256/qj.04.101.

Dennis, J., J. Edwards, K. J. Evans, O. Guba, P. H. Lauritzen, A. A. Mirin, A. St-Cyr, M. A. Taylor, and P. H. Worley (2011), CAM-SE: A scalable spectral element dynamical core for the community atmosphere model, *Int. J. High. Perform. C. Appl.*, submitted.

Dukowicz, J. K., and J. W. Kodis (1987), Accurate conservative remapping (rezoning) for arbitrary Lagrangian-Eulerian computations, *SIAM J. Sci. Stat. Comput.*, *8*(3), 305–321, doi:http://dx.doi.org/10.1137/0908037.

Durran, D. R. (1989), Improving the anelastic approximation., *Journal of the Atmospheric Sciences*, *46*, 1453–1461, doi:10.1175/1520-0469(1989)046⟨1453:ITAA⟩2.0.CO;2.

Durran, D. R. (1999), *Numerical methods for wave equations in geophysical fluid dynamics*, 489 pp., Springer-Verlag.

Durran, D. R., and J. B. Klemp (1982), The effects of moisture on trapped mountain lee waves, *Journal of the Atmospheric Sciences*, *39*(11), 2490–2506, doi:10.1175/1520-0469(1982)039⟨2490:TEOMOT⟩2.0.CO;2.

Edwards, P. N. (2010), *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming*, 528 pp., MIT Press.

Fels, S. B., J. D. Mahlman, M. D. Schwarzkopf, and R. W. Sinclair (1980), Stratospheric Sensitivity to Perturbations in Ozone and Carbon Dioxide: Radiative and Dynamical Response., *Journal of Atmospheric Sciences*, *37*, 2265–2297, doi:10.1175/1520-0469(1980)037⟨2265:SSTPIO⟩2.0.CO;2.

Fournier, A., M. A. Taylor, and J. J. Tribbia (2004), The spectral element atmosphere model (SEAM): High-resolution parallel computation and localized resolution of regional dynamics, *Monthly Weather Review*, *132*(3), 726–748, doi:10.1175/1520-0493(2004)132⟨0726:TSEAMS⟩2.0.CO;2.

Fox-Rabinovitz, M., J. Côté, B. Dugas, M. Déqué, and J. L. McGregor (2006), Variable resolution general circulation models: Stretched-grid model intercomparison project (SGMIP), *Journal of Geophysical Research (Atmospheres)*, *111*, 16,104–16,124, doi:10.1029/2005JD006520.

Fox-Rabinovitz, M. S. (1991), Computational Dispersion Properties of Horizontal Staggered Grids for Atmospheric and Ocean Models, *Monthly Weather Review*, *119*, 1624–1639, doi:10.1175/1520-0493(1991)119⟨1624:CDPOHS⟩2.0.CO;2.

Fox-Rabinovitz, M. S., G. L. Stenchikov, M. J. Suarez, and L. L. Takacs (1997), A Finite-Difference GCM Dynamical Core with a Variable-Resolution Stretched Grid, *Monthly Weather Review*, *125*, 2943–2968, doi:10.1175/1520-0493(1997)125⟨2943:AFDGDC⟩2.0.CO;2.

Fox-Rabinovitz, M. S., L. L. Takacs, R. C. Govindaraju, and M. J. Suarez (2001), A Variable-Resolution Stretched-Grid General Circulation Model: Regional Climate Simulation, *Monthly Weather Review*, *129*, 453–469, doi:10.1175/1520-0493(2001)129⟨0453:AVRSGG⟩2.0.CO;2.

Frank, J., and S. Reich (2004), On Spurious Reflections, Non-uniform Grids and Finite Difference Discretizations of Wave Equations, *Tech. rep.*, Center for Mathematics and Computer Science, CWI Report MAS-E0406.

Fromm, J. E. (1968), A Method for Reducing Dispersion in Convective Difference Schemes, *Journal of Computational Physics*, *3*, 176–189, doi:10.1016/0021-9991(68)90015-6.

Gadd, A. J. (1978), A split explicit integration scheme for numerical weather prediction, *Quart. J. Roy. Meteor. Soc.*, *104*, 569–582, doi:10.1002/qj.49710444103.

Gal-Chen, T., and R. C. J. Somerville (1975), On the use of a coordinate transformation for the solution of the Navier-Stokes equations, *Journal of Computational Physics*, *17*, 209–228, doi:10.1016/0021-9991(75)90037-6.

Galewsky, J., R. K. Scott, and L. M. Polvani (2004), An initial-value problem for testing numerical models of the global shallow-water equations, *Tellus Series A*, *56*, 429–440, doi:10.1111/j.1600-0870.2006.00192.x.

Gaßmann, A. (2010), Non-hydrostatic modelling with the ICON model, Presentation at ECMWF Non-Hydrostatic Workshop, Reading, U.K., Nov. 8-10, 2010.

Giraldo, F. X., and M. Restelli (2008), A study of spectral element and discontinuous Galerkin methods for the Navier Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases, *Journal of Computational Physics*, *227*, 3849–3877, doi:10.1016/j.jcp.2007.12.009.

Giraldo, F. X., and T. E. Rosmond (2004), A Scalable Spectral Element Eulerian Atmospheric Model (SEE-AM) for NWP: Dynamical Core Tests, *Monthly Weather Review*, *132*, 133–153, doi:10.1175/1520-0493(2004)132⟨0133:ASSEEA⟩2.0.CO;2.

Giraldo, F. X., J. S. Hesthaven, and T. Wartburton (2002), Nodal high-order discontinous galerkin methods for the shallow water equations, *Journal of Computational Physics*, *181*, 499–525.

Godunov, S. K. (1959), A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations, *Math. Sbornik*, *47*, 271–306, translated US Joint Publ. Res. Service, JPRS 7226, 1969.

Gottlieb, S., C.-W. Shu, and E. Tadmor (2001), Strong stability-preserving high-order time discretization methods, *SIAM Review*, *43*(1), 89–112, doi:10.1137/S003614450036757X.

Govett, M. W., J. Middlecoff, and T. Henderson (2010), Running the NIM next-generation weather model on GPUs, *Cluster Computing and the Grid, IEEE International Symposium on*, *0*, 792–796, doi:10.1109/CCGRID.2010.106.

Grotjahn, R., and J. J. O'Brien (1976), Some Inaccuracies in Finite Differencing Hyperbolic Equations, *Monthly Weather Review*, *104*, 180–194, doi:10.1175/1520-0493(1976)104⟨0180:SIIFDH⟩2.0.CO;2.

Guillard, H., and C. Viozat (1999), On the behaviour of upwind schemes in the low mach number limit, *Comput. Fluids*, *28*, 63–86.

Hack, J. J., et al. (2008), Report on computational and informational technology rate limiters to the advancement of climate change science, *Tech. rep.*, Advanced Scientific Computing Advisory Committee and Biological and Environmental Advisory Committee, Department of Energy.

Harper, K. C. (2008), *Weather by the Numbers: The Genesis of Modern Meteorology*, 328 pp., MIT Press, Cambridge, MA.

Harten, A., P. Lax, and B. van Leer (1983), On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws, *SIAM Review*, *25*(1), 35–61.

Haurwitz, B. (1940), The motion of atmospheric disturbances on the spherical earth, *Journal of Marine Research*, *3*, 254–267.

Heikes, R., and D. A. Randall (1995), Numerical integration of the shallow water equations on a twisted icosahedral grid. Part I: Basic design and results of tests, *Monthly Weather Review*, *123*, 1862–1880.

Henderson, T. B., S. Benjamin, R. Bleck, J. Brown, S. Sun, J. Bao, S. Sahm, G. A. Grell, and M. Fiorino (2010), Progress in development of the Flow-following finite-volume Icosahedral Model (FIM) toward improving NCEP global ensemble forecasts and toward a chemistry-coupled global, *AGU Fall Meeting Abstracts*, pp. A41B–0088.

Jablonowski, C., and D. L. Williamson (2006), A baroclinic instabilitiy test case for atmospheric model dynamical cores, *Quart. J. Roy. Meteor. Soc.*, *132*(621C), 2943–2975.

Jablonowski, C., M. Herzog, J. E. Penner, R. C. Oehmke, Q. F. Stout, B. van Leer, and K. G. Powell (2006), Block-Structured Adaptive Grids on the Sphere: Advection Experiments, *Monthly Weather Review*, *134*, 3691–3713, doi:10.1175/MWR3223.1.

Jablonowski, C., P. H. Lauritzen, M. A. Taylor, and R. D. Nair (2008a), An Intercomparison of 10 Atmospheric Model Dynamical Cores, in *Eos Trans. AGU, 89(53), Fall Meet. Suppl.*, Abstract A33A-0214.

Jablonowski, C., P. H. Lauritzen, M. A. Taylor, and R. D. Nair (2008b), Idealized test cases for the dynamical cores of Atmospheric General Circulation Models: A proposal for the NCAR ASP 2008 summer colloquium, *Tech. rep.*, NCAR, available at http://esse.engin.umich.edu/admg/publications.php.

Jablonowski, C., R. C. Oehmke, and Q. F. Stout (2009), Block-structured Adaptive Meshes and Reduced Grids for Atmospheric General Circulation Models, *Phil. Trans. R. Soc. A*, *367*, 4497–4522.

Jablonowski, C., P. Ullrich, F. Giraldo, J. Kelly, T. Hundertmark, and S. Reich (2011), The baroclinic instability in a channel: A test case for atmospheric limited area models, in preparation.

Jakob-Chien, R., J. J. Hack, and D. L. Williamson (1995), Spectral transform solutions to the shallow water test set, *Journal of Computational Physics*, *119*, 164–187.

Jebens, S., O. Knoth, and R. Weiner (2010), Linearly implicit peer methods for the compressible Euler equations, *Journal of Computational Physics*, in press.

Jones, P. W. (1999), First- and second-order conservative remapping schemes for grids in spherical coordinates, *Monthly Weather Review*, *127*, 2204–2210.

Karni, S. (1994), On the group velocity of symmetric and upwind numerical schemes, *International Journal for Numerical Methods in Fluids*, *18*, 1073–1081, doi:10.1002/fld.1650181105.

Kasahara, A. (1974), Various Vertical Coordinate Systems Used for Numerical Weather Prediction, *Monthly Weather Review*, *102*, 509–522, doi:10.1175/1520-0493(1974)102⟨0509:VVCSUF⟩2.0.CO;2.

Kasahara, A., and W. M. Washington (1967), NCAR Global General Circulation Model of the Atmosphere, *Monthly Weather Review*, *95*, 389–402, doi:10.1175/1520-0493(1967)095.

Keller, T. L. (1994), Implications of the Hydrostatic Assumption on Atmospheric Gravity Waves., *Journal of Atmospheric Sciences*, *51*, 1915–1929, doi:10.1175/1520-0469(1994)051⟨1915:IOTHAO⟩2.0.CO;2.

Kurihara, Y. (1965), Numerical Integration of the Primitive Equations on a Spherical Grid, *Monthly Weather Review*, *93*, 399–415, doi:10.1175/1520-0493(1965)093⟨0399:NIOTPE⟩2.3.CO;2.

Kwizak, M., and A. J. Robert (1971), A semi-implicit scheme for grid point atmospheric models of the primitive equations, *Monthly Weather Review*, *99*, 32–36, doi:10.1175/1520-0493(1971)099⟨0032:ASSFGP⟩2.3.CO;2.

Laprise, J. P. R., and A. Plante (1995), A Class of Semi-Lagrangian Integrated-Mass (SLIM) Numerical Transport Algorithms, *Monthly Weather Review*, *123*, 553–565, doi:10.1175/1520-0493(1995)123⟨0553:ACOSLI⟩2.0.CO;2.

Lauritzen, P. H., and R. D. Nair (2007), Monotone and conservative cascade remapping between spherical grids (CaRS): Regular latitude-longitude and cubed-sphere grids, *Monthly Weather Review*, *136*, 1416–1432.

Lauritzen, P. H., P. A. Ullrich, and R. D. Nair (2008), A finite-volume multi-tracer transport/advection scheme on the cubed sphere, *Journal of Computational Physics*, in prep.

Lauritzen, P. H., C. Jablonowski, M. A. Taylor, and R. D. Nair (2010), Rotated versions of the Jablonowski steady-state and baroclinic wave test cases: A dynamical core intercomparison, *J. Adv. Model. Earth Syst.*, *2*, doi:10.3894/JAMES.2010.2.15, art. #15, 34 pp.

Lauritzen, P. H., R. D. Nair, and P. A. Ullrich (2010), A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid., *Journal of Computational Physics*, *229*, 1401–1424.

Lax, P., and B. Wendroff (1960), Systems of conservation laws, *Communications on Pure and Applied Mathematics*, *13*(2), 217–237, doi:10.1002/cpa.3160130205.

Lax, P. D. (1954), Weak solutions of non-linear hyperbolic equations and their numerical computation, *Comm. Pure Appl. Math.*, *VII*, 159–193.

Lee, J., R. Bleck, A. E. MacDonald, J. Bao, and S. Benjamin (2006), A flow-following finite-volume icosahedral model, *Eos Trans. AGU*, *87*(52), Fall Meet. Suppl., Abstract A33F–04.

LeVeque, R. (1997), Wave propagation algorithms for multidimensional hyperbolic systems, *Journal of Computational Physics*, *131*, 327–353, doi:10.1006/jcph.1996.5603.

Lin, S.-J. (2004), A "Vertically Lagrangian" Finite-Volume Dynamical Core for Global Models, *Monthly Weather Review*, *132*, 2293–2307, doi:10.1175/1520-0493(2004)132⟨2293:AVLFDC⟩2.0.CO;2.

Lin, S.-J., and R. B. Rood (1996), Multidimensional flux-form semi-Lagrangian transport schemes, *Monthly Weather Review*, *124*, 2046–2070.

Lin, S.-J., and R. B. Rood (1997), An explicit flux-form semi-Lagrangian shallow-water model on the sphere, *Quart. J. Royal Meteor. Soc.*, *123*, 2477–2498, doi:10.1256/smsqj.54415.

Liou, M.-S. (2006), A sequel to AUSM, Part II: AUSM+-up for all speeds, *Journal of Computational Physics*, *214*, 137–170, doi:10.1016/j.jcp.2005.09.020.

Liou, M.-S., and C. Steffen (1993), A new flux splitting scheme, *Journal of Computational Physics*, *107*, 23–39.

Liu, Y., C. Shu, E. Tadmor, and M. Zhang (2007), Central discontinuous Galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction, *SIAM J. Numer. Anal*, *45*, 2442–2467.

Long, D. (2009), Spurious propagation of hyperbolic waves on in-homogeneously staggered grids, Master's thesis, University of Exeter, U.K.

Long, D., and J. Thuburn (2011), Numerical wave propagation on non-uniform one-dimensional staggered grids, *Journal of Computational Physics*, *230*, 2643–2659, doi:10.1016/j.jcp.2010.12.040.

Lorenz, E. N. (1963), Deterministic Nonperiodic Flow, *Journal of Atmospheric Sciences*, *20*, 130–148, doi:10.1175/1520-0469(1963)020⟨0130:DNF⟩2.0.CO;2.

MacCormack, R. (2003), The Effect of Viscosity in Hypervelocity Impact Cratering, *Journal of Spacecraft and Rockets*, *40*, 757–763, doi:10.2514/2.6901.

Majewski, D. (1998), The new global icosahedral-hexagonal gridpoint model GME of the Deutscher Wetterdienst., in *Proc. Seminar on Numerical Methods for Weather Prediction*, vol. 1, pp. 172–201, ECMWF, Reading, United Kingdom.

Majewski, D., D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanisch, G. Paul, W. Wergen, and J. Baumgardner (2002), The Operational Global Icosahedral Hexagonal Gridpoint Model GME: Description and High-Resolution Tests, *Monthly Weather Review*, *130*, 319–338, doi:10.1175/1520-0493(2002)130⟨0319:TOGIHG⟩2.0.CO;2.

Manabe, S., and J. Smagorinsky (1965), Simulated Climatology of General Circulation with a Hydrologic Cycle, *Monthly Weather Review*, *93*, 769–798.

Margolin, L. G., and M. Shashkov (2003), Second-order sign-preserving conservative interpolation (remapping) on general grids, *Journal of Computational Physics*, *184*, 266–298.

Marshall, J., A. Adcroft, C. Hill, L. Perelman, and C. Heisey (1997), A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers, *Journal of Geophysical Research*, *102*, 5753–5766, doi:10.1029/96JC02775.

Mintz, Y. (1965), Very Long-Term Global Integration of the Primitive Equations of Atmospheric Motion, in *WMO-IUGG Symposium on Research and Development Aspects of Long-Range Forecasting, Boulder, Colo., 1964. (WMO Technical Note No. 66)*, edited by W. M. Organization.

Nair, R. D., and B. Machenhauer (2002), The mass-conservative cell-integrated semi-Lagrangian advection scheme on the sphere, *Monthly Weather Review*, *130*(3), 649–667.

Nair, R. D., S. J. Thomas, and R. D. Loft (2005), A discontinuous Galerkin transport scheme on the cubed sphere, *Monthly Weather Review*, *133*, 814–828.

Nair, R. D., S. J. Thomas, and R. D. Loft (2005), A discontinuous Galerkin global shallow water model, *Monthly Weather Review*, *133*, 876–888, doi:10.1175/MWR2903.1.

National Academy of Sciences (1966), *Weather and Climate Modification: Problems and Prospects (2 Vols.)*, National Academy of Sciences, Washington, DC, Committee on Atmospheric Sciences Panel on Weather and Climate Modification.

Noelle, S., N. Pankratz, G. Puppo, and J. R. Natvig (2006), Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows, *Journal of Computational Physics*, *213*, 474–499, doi:10.1016/j.jcp.2005.08.019.

Nørsett, S. P., and A. Wolfbrandt (1979), Order conditions for Rosenbrock type methods, *Numerische Mathematik*, *32*, 1–15, 10.1007/BF01397646.

Ogura, Y., and N. A. Phillips (1962), Scale analysis of deep and shallow convection in the atmosphere., *Journal of the Atmospheric Sciences*, *19*, 173–179, doi:10.1175/1520-0469(1962)019.

Osher, S., and J. A. Sethian (1988), Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *Journal of Computational Physics*, *79*, 12–49, doi:10.1016/0021-9991(88)90002-2.

Phillips, N. A. (1956), The General Circulation of the Atmosphere: A Numerical Experiment, *Quart. J. Roy. Met. Soc.*, *82*, 123–64.

Phillips, N. A. (1957), A Coordinate System Having Some Special Advantages for Numerical Forecasting., *Journal of the Atmospheric Sciences*, *14*, 184–185, doi:10.1175/1520-0469(1957)014⟨0184:ACSHSS⟩2.0.CO;2.

Phillips, N. A. (1966), The Equations of Motion for a Shallow Rotating Atmosphere and the 'Traditional Approximation', *Journal of Atmospheric Sciences*, *23*, 626–627, doi:10.1175/1520-0469(1966)023⟨0626:TEOMFA⟩2.0.CO;2.

Prather, M. J. (1986), Numerical advection by conservation of second-order moments, *Journal of Geophysical Research*, *91*, 6671–6681, doi:10.1029/JD091iD06p06671.

Putman, W. M., and S.-J. Lin (2007), Finite-volume transport on various cubed-sphere grids, *Journal of Computational Physics*, *227*, 55–78, doi:10.1016/j.jcp.2007.07.022.

Putman, W. M., and S.-J. Lin (2009), A finite-volume dynamical core on the cubed-sphere grid, in *Numerical Modeling of Space Plasma Flows: Astronum-2008*, vol. 406, edited by N. V. Pogorelov, E. Audit, P. Colella, and G. P. Zank, pp. 268–276, Astronomical Society of the Pacific Conference Series.

Putman, W. M., and M. Suarez (2009), Mesoscale weather and climate modeling with the global non-hydrostatic Goddard Earth Observing System Model (GEOS-5) at cloud-permitting resolutions, *AGU Fall Meeting Abstracts*, abs. # A42B-08.

Randall, D. A. (1994), Geostrophic Adjustment and the Finite-Difference Shallow-Water Equations, *Monthly Weather Review*, *122*, 1371–1377, doi:10.1175/1520-0493(1994)122⟨1371:GAATFD⟩2.0.CO;2.

Randall, D. A. (2000), *General Circulation Model development: Past, present, and future*, Academic Press, 807 pp.

Rančić, M. (1992), Semi-Lagrangian piecewise biparabolic scheme for two-dimensional horizontal advection of a passive scalar, *Monthly Weather Review*, *120*, 1394–1405.

Rančić, M., J. Purser, and F. Mesinger (1996), A global shallow water model using an expanded spherical cube, *Quart. J. Roy. Meteor. Soc.*, *122*, 959–982.

Restelli, M., and F. X. Giraldo (2009), A conservative discontinuous Galerkin semi-implicit formulation for the Navier-Stokes equations in nonhydrostatic mesoscale modeling, *SIAM Journal of Scientific Computing*, *31*(3), 2231–2257, doi:10.1137/070708470.

Richardson, L. F. (1922), *Weather Prediction by Numerical Process*, 250 pp., Cambridge University Press, Cambridge, MA, reprinted NY, Dover, 1965.

Ringler, T. D., R. P. Heikes, and D. A. Randall (2000), Modeling the Atmospheric General Circulation Using a Spherical Geodesic Grid: A New Class of Dynamical Cores, *Monthly Weather Review*, *128*, 2471–2490, doi:10.1175/1520-0493(2000)128⟨2471:MTAGCU⟩2.0.CO;2.

Robert, A. (1993), Bubble convection experiments with a semi-implicit formulation of the Euler equations, *Journal of the Atmospheric Sciences*, *50*, 1865–1873, doi:10.1175/1520-0469(1993)050⟨1865:BCEWAS⟩2.0.CO;2.

Roe, P. L. (1981), Approximate Riemann solvers, parameter vectors, and difference schemes, *Journal of Computational Physics*, *43*, 357–372, doi:10.1016/0021-9991(81)90128-5.

Ronchi, C., R. Iacono, and P. S. Paolucci (1996), The "cubed sphere": A new method for the solution of partial differential equations in spherical geometry, *Journal of Computational Physics*, *124*(1), 93–114.

Rood, R. B. (1987), Numerical advection algorithms and their role in atmospheric transport and chemistry models, *Reviews of Geophysics*, *25*, 71–100, doi:10.1029/RG025i001p00071.

Rosenbrock, H. (1963), Some general implicit processes for the numerical solution of differential equations., *The Computer Journal*, *5*, 329–330.

Rossmanith, J. A. (2006), A wave propagation method for hyperbolic systems on the sphere, *Journal of Computational Physics*, *213*, 629–658, doi:10.1016/j.jcp.2005.08.027.

Rusanov, V. V. (1961), Calculation of intersection of non-steady shock waves with obstacles, *J. Comput. Math. Phys. USSR*, *1*, 267–279.

Sadourny, R. (1972), Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids, *Monthly Weather Review*, *100*, 136–144.

Sadourny, R., A. Arakawa, and Y. Mintz (1968), Integration of the nondivergent barotropic vorticity equation with an icosahedral-hexagonal grid for the sphere, *Monthly Weather Review*, *96*, 351–356, doi:10.1175/1520-0493(1968)096⟨0351:IOTNBV⟩2.0.CO;2.

Schmidt, G. A., et al. (2006), Present-Day Atmospheric Simulations Using GISS ModelE: Comparison to In Situ, Satellite, and Reanalysis Data., *Journal of Climate*, *19*, 153–192, doi:10.1175/JCLI3612.1.

Simmons, A. J., and D. M. Burridge (1981), An energy and angular-momentum conserving vertical finite-difference scheme and hybrid vertical coordinates, *Mon. Wea. Rev.*, *109*, 758–766.

Skamarock, W. C., and J. B. Klemp (1993), Adaptive Grid Refinement for Two-Dimensional and Three-Dimensional Nonhydrostatic Atmospheric Flow, *Monthly Weather Review*, *121*, 788–804, doi:10.1175/1520-0493(1993)121⟨0788:AGRFTD⟩2.0.CO;2.

Skamarock, W. C., and J. B. Klemp (2008), A time-split nonhydrostatic atmospheric model for weather research and forecasting applications, *Journal of Computational Physics*, *227*, 3465–3485, doi:10.1016/j.jcp.2007.01.037.

Skamarock, W. C., J. Oliger, and R. L. Street (1989), Adaptive Grid Refinement for Numerical Weather Prediction, *Journal of Computational Physics*, *80*, 27–60, doi:10.1016/0021-9991(89)90089-2.

Skamarock, W. C., J. Klemp, M. Duda, S.-H. Park, L. Fowler, T. Ringler, J. Thuburn, M. Gunzburger, and L. Ju (2010), Global Non-Hydrostatic Modeling Using Voronoi Meshes: The MPAS Model, Presentation at ECMWF Non-Hydrostatic Workshop, Reading, U.K. Nov. 8-10, 2010.

Smagorinsky, J. (1983), The Beginnings of Numerical Weather Prediction and General Circulation Modeling: Early Recollections, *Advances in Geophysics*, *25*, 3–37.

St-Cyr, A., C. Jablonowski, J. M. Dennis, H. M. Tufo, and S. J. Thomas (2008), A comparison of two shallow-water models with nonconforming adaptive grids, *Monthly Weather Review*, *136*, 1898–1922, doi:10.1175/2007MWR2108.1.

Staniforth, A., and N. Wood (2008), Aspects of the dynamical core of a nonhydrostatic, deep-atmosphere, unified weather and climate-prediction model, *Journal of Computational Physics*, *227*, 3445–3464, doi:10.1016/j.jcp.2006.11.009.

Staniforth, A. N., and H. L. Mitchell (1978), A Variable-Resolution Finite-Element Technique for Regional Forecasting with the Primitive Equations, *Monthly Weather Review*, *106*, 439–447, doi:10.1175/1520-0493(1978)106⟨0439:AVRFET⟩2.0.CO;2.

Taylor, M., J. Tribbia, and M. Iskandarani (1997), The spectral element method for the shallow water equations on the sphere, *Journal of Computational Physics*, *130*, 92–108.

Taylor, M. A., J. Edwards, and A. St.Cyr (2008), Petascale atmospheric models for the community climate system model: New developments and evaluation of scalable dynamical cores, *J. Phys. Conf. Ser.*, *125*, 012,023, doi:10.1088/1742-6596/125/1/012023.

Thomas, S. J., and R. D. Loft (2002), Semi-implicit spectral element atmospheric model, *Journal of Scientific Computing*, *17*, 339–350, doi:10.1023/A:1015129420882.

Thomas, S. J., and R. D. Loft (2005), The NCAR spectral element climate dynamical core: Semi-implicit Eulerian formulation, *Journal of Scientific Computing*, *25*(1), 307–322, doi:http://dx.doi.org/10.1007/s10915-004-4646-2.

Thuburn, J., and Y. Li (2000), Numerical simulations of Rossby Haurwitz waves, *Tellus Series A*, *52*, 181–189, doi:10.1034/j.1600-0870.2000.00107.x.

Thuburn, J., T. D. Ringler, W. C. Skamarock, and J. B. Klemp (2009), Numerical representation of geostrophic modes on arbitrarily structured C-grids, *Journal of Computational Physics*, *228*(22), 8321–8335, doi:10.1016/j.jcp.2009.08.006.

Tomita, H., and M. Satoh (2004), A new dynamical framework of nonhydrostatic global model using the icosahedral grid, *Fluid Dynamics Research*, *34*(6), 357 – 400, doi:10.1016/j.fluiddyn.2004.03.003.

Tomita, H., M. Tsugawa, M. Satoh, and K. Goto (2001), Shallow water model on a modified icosahedral geodesic grid by using spring dynamics, *Journal of Computational Physics*, *174*, 579–613, doi:10.1006/jcph.2001.6897.

Toro, E. F. (1999), *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Second ed., 624 pp., Springer, ISBN-10: 3540659668.

Trefethen, L. N. (1982), Group velocity in finite difference schemes, *SIAM Review*, *24*(2), 113–136, doi:10.1137/1024038.

Ullrich, P., and C. Jablonowski (2011a), Operator-Split Runge-Kutta-Rosenbrock (RKR) Methods for Non-hydrostatic Atmospheric Models, *Monthly Weather Review*, in review.

Ullrich, P. A., and C. Jablonowski (2011b), An analysis of 1D finite-volume methods for geophysical problems on refined grids, *Journal of Computational Physics*, *230*, 706–725, doi:10.1016/j.jcp.2010.10.014.

Ullrich, P. A., P. H. Lauritzen, and C. Jablonowski (2009), Geometrically exact conservative remapping (GECoRe): Regular latitude-longitude and cubed-sphere grids, *Monthly Weather Review*, *137*, 1721–1741.

Ullrich, P. A., C. J. Jablonowski, and B. L. van Leer (2010), High-order finite-volume models for the shallow-water equations on the sphere, *Journal of Computational Physics*, *229*, 6104–6134.

van Leer, B. (1974), Towards the Ultimate Conservation Difference Scheme. II. Monotonicity and Conservation Combined in a Second-Order Scheme, *Journal of Computational Physics*, *14*, 361–370, doi:10.1016/0021-9991(74)90019-9.

van Leer, B. (1977), Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection, *Journal of Computational Physics*, *23*, 276–299, doi:10.1016/0021-9991(77)90095-X.

van Leer, B. (1979), Towards the ultimate conservative difference scheme. V - A second-order sequel to Godunov's method, *Journal of Computational Physics*, *32*, 101–136, doi:10.1016/0021-9991(79)90145-1.

van Leer, B., J. Thomas, P. Roe, and R. Newsome (1987), A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations, *AIAA Paper No. 87-1104-CP*, Computational Fluid Dynamics Conference, 8th, Honolulu, HI, June 9-11, 1987.

Varah, J. M. (1980), Stability restrictions on second order, three level finite difference schemes for parabolic equations, *SIAM J. Numer. Anal.*, *17*(2), 300–309, doi:10.1137/0717025.

Vichnevetsky, R. (1987), Wave propagation and reflection in irregular grids for hyperbolic equations, *Applied Numerical Mathematics*, *3*, 133–166.

Vichnevetsky, R., and L. H. Turner (1991), Spurious scattering from discontinuously stretching grids in computational fluid dynamics, *Applied Numerical Mathematics*, *8*(3), 289–299, doi:http://dx.doi.org/10.1016/0168-9274(91)90058-8.

Walko, R. L., and R. Avissar (2008), The Ocean-Land-Atmosphere Model (OLAM). Part II: Formulation and Tests of the Nonhydrostatic Dynamic Core, *Monthly Weather Review*, *136*, 4045–4062, doi:10.1175/2008MWR2523.1.

Wan, H. (2009), Developing and testing a hydrostatic atmospheric dynamical core on triangular grids, *Tech. Rep. 65*, *Reports on Earth System Science*, Max-Planck Institute for Meteorology, Hamburg, Germany, iSSN 1614-119.

Washington, W., D. B. Bader, B. Collins, J. Drake, M. A. Taylor, B. Kirtman, D. N. Williams, and D. Middleton (2008), Challenges in climate change science and the role of computing at the extreme scale, *Tech. rep.*, U.S. Department of Energy, Office of Biological and Environmental Research and the Office of Advanced Scientific Computing Research, Nov. 6-7, 2008, Washington D.C.

Wedi, N., P. Benard, K. Yessad, A. Untch, S. Malardel, M. Hamrud, G. Mozdzynski, M. Fisher, and P. Smolarkiewicz (2010), Non-hydrostatic modeling with IFS: current status, Presentation at ECMWF Non-Hydrostatic Workshop, Reading, U.K., Nov. 8-10, 2010.

White, A. A., B. J. Hoskins, I. Roulstone, and A. Staniforth (2005), Consistent approximate models of the global atmosphere: shallow, deep, hydrostatic, quasi-hydrostatic and non-hydrostatic, *Quart. J. Royal Meteor. Soc.*, *131*, 2081–2107, doi:10.1256/qj.04.49.

Williamson, D., J. Drake, J. Hack, R. Jakob, and P. Swarztrauber (1992), A standard test set for numerical approximations to the shallow water equations in spherical geometry, *Journal of Computational Physics*, *102*, 211–224.

Williamson, D. L. (1968), Integration of the barotropic vorticity equation on a spherical geodesic grid, *Tellus*, *20*(4), 642–653, doi:10.1111/j.2153-3490.1968.tb00406.x.

Williamson, D. L. (2008), Equivalent finite volume and Eulerian spectral transform horizontal resolutions established from aqua-planet simulations, *Tellus Series A*, *60*, 839–847, doi:10.1111/j.1600-0870.2008.00340.x.

Wood, N., and A. Staniforth (2003), The deep-atmosphere Euler equations with a mass-based vertical coordinate, *Quart. J. Royal Meteor. Soc.*, *129*, 1289–1300, doi: 10.1256/qj.02.153.

Yeh, K., J. Côté, S. Gravel, A. Méthot, A. Patoine, M. Roch, and A. Staniforth (2002), The CMC MRB Global Environmental Multiscale (GEM) Model. Part III: Nonhydrostatic Formulation, *Monthly Weather Review*, *130*, 339–356, doi:10.1175/1520-0493(2002)130⟨0339:TCMGEM⟩2.0.CO;2.

Zalesak, S. T. (1979), Fully multidimensional flux-corrected transport algorithms for fluids, *Journal of Computational Physics*, *31*, 335–362.

Zerroukat, M., N. Wood, and A. Staniforth (2005), A monotonic and positive-definite filter for a semi-Lagrangian inherently conserving and efficient (SLICE) scheme, *Quart. J. Royal Meteor. Soc.*, *131*(611), 2923–2936.