Algorithm development for vertical Lagrangian coordinate based non-hydrostatic and hydrostatic models

by

Xi Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Atmospheric and Space Sciences)
in The University of Michigan
2013

Doctoral Committee:

Professor Joyce E. Penner, Chair
Professor Ken Powell
Physical Scientist Shian-Jiann Lin
Research Scientist Natalia G Andronova
Associate Professor Christiane Jablonowski

To my wife Yuan, and my parents LIU Mengqun and CHEN Shangfang

ACKNOWLEDGEMENTS

Over the past six years I have received support and encouragement from a great number of people. First of all, I would like to say thank you to my committee, my advisor Joyce E. Penner, Ken Powell, Shian-Jiann Lin, Natalia Andronova and Christiane Jablonowski for all the support and advice you provided. Especially I would like to thank my advisor, Joyce E. Penner for her support and great vision and scientific excitement that inspire me all the time during my research. In addition, I would like to extend a special thank you to Bram van Leer for his unbounded experience and knowledge on computational fluid dynamics. He could always use humorous language to enlighten me on any problem that I have.

Many thanks goes to Ken Powell and John Boyd, both of who have offered unique insights and perspectives during the development of this project. I also wish to thank Natalia Andronova, who spent countless hours with me in numerous project discussions, brain-storming talks and provided needed encouragement and insights. Furthermore, I would like to thank Christiane Jablonowski, who gave me a very good big picture on climate modeling and offered me all the help she could provide in every detailed way. Special thanks to Shian-Jiann Lin at GFDL, who always replied my emails with full enthusiasm. I would never forget the super long discussion during my two-day visit with him. I would like to thank Denny VandenBerg for many useful discussion on the AMR library. I would also thank Paul Ullrich for many "numerical discussions" and he helped me to tune the AUSM$^+$-up Riemann Solver to work. I would also like to thank Xianglei Huang for the unexpected meetings during the night

hours and he introduced me many details on various aspects of the GCMs.

In addition, thank you, Margaret Reid, Sue Griffin, Sandy Pytlinski and Kristi Hansen for the excellent administrative support and thank you Bryan White, Faye Ogasawara and Melissa Terwilliger for the IT support. You helped us to keep everything running smooth and made my life a lot easier.

My big research group also gave me endless supports and have been a big part of my life in Ann Arbor. So, thank you Yang Chen, Huan Guo, Minghuai Wang, Li Xu, Roy Chen, Yuxing Yun, Erika Roesler and Guangxing Lin.

Most importantly, I would like to thank my family for all of their support and unconditional love. Especially I would like to thank my mom and dad. I would not be able to finish this project without their continuous encouragement during the years. Finally, I would like to thank my wife Yuan for her support, patience and understanding in every step of the way.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF APPENDICES

Appendix

ABSTRACT

Algorithm development for vertical Lagrangian coordinate based non-hydrostatic and hydrostatic models

by

Xi Chen

Chair: Joyce E. Penner

This thesis demonstrates a new non-staggered-grid finite-volume method for dynamical cores in atmospheric general circulation models. Finite volume methods have been proved to be robust and accurate in many research areas. These numerical methods have many properties, which make them desirable for modeling atmospheric dynamics. However, many non-staggered-grid finite-volume methods are numerically expensive because they implement Riemann solvers. To bypass the requirement of using a Riemann solver, current finite volume dynamical cores in the General Circulation Models use less stable staggered grids. We developed a low-cost Riemann solver to implement with the non-staggered-grid schemes, which matches the numerical efficiency of the current staggered-grid schemes. It is much easier to develop a scheme with a high-order of accuracy and which has better performance when using adaptive mesh refinement (AMR) using a non-staggered grid finite volume scheme.

Adaptive mesh refinement is a method that allows one to adaptively capture the features of small-scale motions of particular dynamical interest. However, non-uniform grids are required in adaptive mesh refinement techniques. The varying

resolution can cause artificial reflections of waves due to incompatible mechanisms at fine-grid and coarse-grid interfaces. We tested our models using the full set of the conservative Euler equations with non-uniform horizontal grids, in both hydrostatic and non-hydrostatic formulations. Our algorithm for the interface is fully two-way interactive. There is almost no reflection observed in our results and due to the high-order accurate interface, the waves are not damped when passing the interfaces.

Since the vertical distribution of pressure in the atmosphere is a non-polynomial monotonic function, the polynomial based 3D AMR might create pressure gradients at the interface of two grids, even if the atmosphere is initialized statically as a steady state. We introduced an alternative way to find the pressure at any vertical altitude within a control volume using an iterative method. We have shown that in the isentropic atmosphere, the theoretical value of the pressure at any altitude can be determined using this method. Using this method in the vertical volume refinement can prevent creating pressure gradients at the interface between two grids.

CHAPTER I

Introduction: algorithm development in weather and climate modeling

1.1 Atmospheric models and their limitations

An atmospheric model is a mathematical model construct around the full set of primitive dynamical equations, which govern atmospheric motions. It may be necessary to supplement these equations with parameterizations for turbulent diffusion, radiation, moist processes (clouds and precipitation), heat exchange, soil, vegetation, surface water, the kinematic effects of terrain, and convection. Atmospheric models, if properly initialized, can predict microscale phenomena such as tornadoes and boundary layer eddies, sub-microscale turbulent flow over buildings, as well as synoptic and global flows. The horizontal domain of a model is either global, covering the entire Earth, or regional, covering only part of the Earth. An atmospheric model of the general circulation on a rotating sphere of earth with thermodynamic terms for various energy sources is called atmospheric general circulation model (GCM). Atmospheric and oceanic GCMs (AGCM and OGCM) are key components of global climate models along with sea ice and land surface components. GCMs and global climate models are widely applied for weather forecasting, understanding the climate and projecting climate change. Climate refers to the average of weather conditions.

It varies on timescales ranging from seasonal to centennial. Climate models use quantitative methods to simulate the interactions between the ocean, the atmosphere, the land, the ice, and changes in the Earth's energy balance resulting from volcanic eruptions and variations in the sun's intensity. A minimal GCM consists of a *dynamical core* that relates material properties such as temperature to dynamical properties such as pressure and velocity. This thesis focuses on the *dynamical core* component of the GCMs.

Like all kinds of numerical simulations, the equations of motion are discretized in GCMs, and discretization introduces errors. The precision of the discretization in GCMs is primarily determined or limited by computational power. Global climate is produced through a variety of processes and interactions that operate on a wide range of scales, including molecular, regional, continental and global. Changes in climate occur from physical interactions that take place on any or all of these scales. The changes, and the resulting weather patterns, can occur nearly instantaneously or they can take decades or millennia to develop. Unfortunately, the computers and programs that run the GCMs are limited to gross representations of the geographic, geologic and atmospheric details that they use to run climate simulations. Thus, many small-scale features, such as a shift in the prevailing winds or unusually dry surface conditions due to increased evaporation from forest fires and high winds cannot be properly resolved, but must be represented by empirical parameterizations.

Since it is almost impossible to represent physical processes in the atmosphere of all scales in GCMs, the dynamical core of GCMs is engineered to satisfy a delicate balance between numerical stability, an accurate representation of the equations of motion, and computational cost. Over the years, the range of scales of atmospheric motions that could be simulated with proper resolution has been expanding with the increasing speed and memory of supercomputers. The development of dynamical cores in GCMs during the past 50 years is also a history of designing the most suitable

numerical methods to simulate atmospheric flows of the scales corresponding to the available computer power.

1.2 A brief history of the development of numerical weather prediction

Hundreds of algorithms for numerical modeling have been developed during the past 50 years. It is not practical to cover every aspect of the development of numerical weather prediction in this section. We target instead the most important milestones during the years. A more complete introduction written by Spencer Weart could be found online from: http://www.aip.org/history/climate/GCM.htm.

We use the word "simulate" to indicate that we use the computer to solve numerically the systems of differential equations derived from the basic laws of physics, fluid motions, and chemistry. However, the idea of predicting the weather was presented far before the introduction of the computer. Vilhelm Bjerknes suggested the possibility of deterministic weather prediction as early as 1904 (*Gramelsberger*, 2009). Around the time of the First World War, Lewis Richardson actually attempted to produce such a forecast by manually integrating a finite-difference approximation to the equations governing atmospheric motion (*Richardson*, 1922). Unfortunately, his calculations did not yield a reasonable forecast. Moreover, the human labor required to obtain this disappointing result was so great that subsequent attempts at deterministic weather prediction had to await the introduction of a high-speed computational aid. The first computer-generated weather forecast was conducted by a team of researchers under the direction of Jule Charney and John von Neumann at the Institute for Advanced Study, at Princeton, from the first general-purpose electronic computer, the ENIAC (Electronic Numerical Integrator and Calculator) (*Charney et al.*, 1950). This computer-generated one-day weather forecast was surprisingly good, and this

success led to the rapid growth of a new meteorological sub-discipline, "numerical weather prediction".

When Jule Charney was developing the first computer-generated weather forecast, he used Richardson's equations as the starting-point. However, Charney had to simplify them to run large-scale calculations in weeks rather than centuries. Some unwanted solutions, such as the sound waves, are filtered out. Charney's model was a regional model. They divided the atmosphere over North America into 270 grid points with grid spacing of roughly 700 km, and used a time step of approximately 3 hours. The simulation was purely in 2D. The calculation time for a 24-hour forecast was about 24 hours.

Inspired by Charney's work, a meteorology group at the University of Stockholm started delivering forecasts to the Royal Swedish Air Force Weather Service in December 1964. The American Weather Bureau and units also established a Joint Numerical Weather Prediction Unit to issue real-time forecasts in advance of the weather. With limited computer power available, the teams had to simplify the full "primitive equations" of Bjerknes and Richardson. These models, like Charney's, are all regional models, and could give fairly good forecasts up to three days ahead.

Norman Phillips at Princeton University developed the first atmospheric general circulation model in 1955 (*Phillips*, 1956). Phillips had developed improved equations for a two-layer atmosphere modeled circulation on a cylinder of 17 cells high and 16 in circumference. The calculations developed a plausible jet stream and the evolution of a realistic-looking weather disturbance over as long as a month.

In 1958, Joseph Smagorinsky at the U.S. Weather Bureau invited Syukuro Manabe to join his lab to create a general circulation model of the entire three-dimensional global atmosphere. By 1965, Manabe and Smagorinsky's group had built a reasonably complete three-dimensional global model (*Manabe et al.*, 1965). As computer power grew, this model could directly solve the primitive equations. During the same

period of time, Yale Mintz at the University of California at Los Angeles (UCLA) recruited Akio Arakawa to develop another general circulation model. Their model also performed on an entire globe. But unlike the Manabe model, their work incorporated realistic orography using a two-layer model. The Mintz-Arakawa and Smagorinsky-Manabe models were the best models during 1950s and early 1960s. However, in order to improve the models so that they could be solved within a few weeks, computers that were ten or even a hundred times more powerful were needed.

The early development of the algorithms for solving the primitive equations in the GCMs proceeded by dividing the planet's surface into a grid of elements. With increasing computer power, a wider range of the scale of fluid motions could be solved using smaller grid elements. However, the grids were based on the natural choice of latitudes and longitudes. Near the Earth's poles, the meridians converge in a point and the mathematics gets difficult. In the 1970s, the equations of motion were reformulated in the forms of spherical harmonics, which got around the trouble of the pole problems. This spectral method simplified many of the computations but required much faster computers. Nonetheless, the spectral transform technique had many advantages, such as a high degree of accuracy, and it became a very popular method in the GCM community and is still in use in models today. However, the spectral methods were based on Fourier transforms, and had some problems when discontinuity existed in the solution. One problem is that the monotonicity and positivity are not guaranteed for the flow variables. Significant spurious overshoots and undershoots may result in negative tracer concentrations. Another problem is the so-called "Gibbs ringing" when solutions are not smooth. High-frequency waves must be explicitly damped. Finally, spectral methods require Fourier transforms of the global range. The calculations of the Fourier transforms are not easily decomposed into small calculating units. Thus modern parallel supercomputing architectures would not work efficiently with spectral methods.

In 1987, Richard Rood compared many algorithms for modeling advection processes, including spectral methods (*Rood*, 1987). Although Spectral and pseudo-spectral techniques consistently provided the highest degree of accuracy, monotonicity was poorly handled. However, positive definite solutions and monotonicity are perfectly handled by the finite volume methods, while high accuracy could also be maintained. At that time, finite volume methods were very popular in other fields such as aerospace and astrophysics, which often dealt with high-speed flows with strong discontinuities. The finite volume method was first introduced by Godunov in 1959 (*Godunov*, 1959) and later extended by Bram van Leer in a series of papers (*van Leer et al.*, 1973; *van Leer*, 1974; *Van Leer*, 1977; *van Leer*, 1979) which extending Godunov's method to second-order accuracy. *Colella and Woodward* (1984) developed the piecewise-parabolic method (PPM), which achieved a third-order accurate finite volume method. Different from the spectral methods, finite-volume methods are similar to finite difference methods, which calculate values on a meshed geometry. The advection of the value in one grid only requires information from limited number of neighboring grids, which is useful when using the modern parallel supercomputer architectures. Besides, conservation, which is important in many aspects of both climate models and weather prediction models (*Thuburn*, 2008), is preserved out-of-the-box in finite volume methods.

Guided by Rood's work, a finite-volume dynamic core was developed by Shian-Jiann Lin (*Lin and Rood*, 1996, 1997; *Lin*, 2004), which used a staggered grid and the third-order PPM reconstruction procedure in a vertical Lagrangian coordinate. This dynamical core suited perfectly with the distributed memory parallel computing architecture, and is perhaps one of the most well-know dynamic cores today. Today, it is still well used in many major GCMs such as the Geophysical Fluid Dynamics Laboratory (GFDL) GCM, NASA's Goddard Earth Observing System Model (GEOS) and NCAR's Community Atmosphere Model (CAM).

1.3 New algorithm challenges for the dynamical core in GCMs

The smallest scale of atmospheric motions that can be resolved in an Atmospheric numerical model is determined by the smallest grid spacing used in the models. The grid spacing in the first numerical model on a computer developed by Charney was about 700 km, which could resolve planetary waves. With the exponential increase in computational power, the massively parallel supercomputing system, which consists of thousands of interconnected processors, allows us to run our models with resolutions on the order of 10 km. Many small scale atmospheric flows such as synoptic cyclones and hurricanes can be simulated in today's models. However, since global climate is determined by the nonlinear integration of a variety of processes and interactions that include a wide range of scales, the finest resolution a GCM could provide is never sufficient to explicitly represent all processes. Nonetheless, with the finer resolution available in today's GCMs, we can improve the algorithms in the GCMs by replacing some parameterizations, which were approximations since they were under-resolved features, with the equations that describe the underlying physical processes.

However, not all the parameterizations of the unresolved-scale processes served at a satisfactory level. One of the most uncertain aspects of today's climate modeling are the feedbacks involving clouds. The cloud related processes, such as cloud/aerosol interactions, caused the biggest uncertainty in the calculations of the climate change. Obviously, GCMs were not the only way to understand the clouds. Regional Cloud Resolving Models (CRMs) could use much smaller time steps and finer resolutions to explicitly simulate the underlying physics equations and provide us with a better understanding of the cloud processes and the cloud related feedbacks. The results simulated from the CRMs could serve as a reference to judge the quality of the results generated by the GCMs. But, implementing GCMs at the resolution level of the regional CRMs is far beyond computing capability available today.

One promising way for allowing clouds to be followed within global models is the

so-called "super-parameterization" (*Khairoutdinov and Randall*, 2001; *Khairoutdinov et al.*, 2005; *Wang et al.*, 2011). Super-parameterizations insert a two-dimensional slice of a higher resolution model inside each bigger GCM box. Though the embedded model is very rough and does not really resolve clouds as in full CRMs, it does use appropriate fundamental equations to describe basic cloud motions rather than the simpler formulations used in traditional GCMs. Super-parameterizations are used as the basis for increasing the accuracy of climate models that can be run more efficiently to simulate climate at a lower cost than a true global CRM, but more realistically than a traditional climate model. But even for these models aerosol/cloud interactions may be poorly resolved if grid resolutions are limited to those above 250 m.

Another alternative approach to better represent clouds in climate models in a computationally efficient and accurate manner is to develop methods that use non-uniform grids to resolve the 3-dimensional flow field and physics in different regions within a climate model. Substantial savings in computer time could result if a code had the capability to resolve water species transport and removal as well as precipitation development on the scales relevant to cloud formation in the local fine grids. The possible savings in computer time becomes even more compelling when additional trace species such as aerosols are incorporated into the climate model. This is because some of the most important removal processes for trace species are those involving precipitation.

In our work, the finite volume dynamical core was adopted to develop the means to solve the problem of resolving small-scaled processes such as cloud formation within a larger scale GCM. Our work supports the development of future climate models based on high performance, parallel computing paradigms to address questions related to cloud-aerosol interactions. Our aim is to refine the model resolution on a global scale in regions of interest to the solution of the aerosol/cloud interaction problem. This requires that non-hydrostatic regions be embedded within a hydrostatic model

(*Côté et al.*, 1998; *Yeh et al.*, 2002; *Lee and Penner*, 2010). Previous work explored adaptive grid techniques in a Lagrangian hydrostatic model (*Jablonowski et al.*, 2006). Algorithms to embed non-hydrostatic regions into the hydrostatic model need to be developed.

1.4 Hydrostatic and non-hydrostatic models

As discussed above, the flow motions that are explicitly resolved in atmospheric models are highly dependent on the grid spacing in the model. Most of the current general circulation models resolve processes whose horizontal scale is significantly larger than the vertical scale. Thus, vertical acceleration is negligible compared to vertical pressure gradients and vertical buoyancy forces, and these models are based on equations that assume hydrostatic equilibrium. Hydrostatic models have been successfully applied with horizontal resolutions as small as about 10 km, resolving some mesoscale circulations. Global and regional weather prediction models have traditionally been hydrostatic models.

Pure hydrostatic flows are highly stratified. In the hydrostatic primitive equations, which are formulated using a vertical Lagragian coordinate system, both left-hand-side and right-hand-side of the vertical momentum equation vanish. Thus, the 3D governing equations are reduced to a 2D form. The vertical Lagrangian coordinate was first formulated in *Starr* (1945), and was reviewed and discussed in *Kasahara* (1974). The Lin-Rood finite volume dynamic core took full advantage of the vertical Lagrangian coordinate, and constructed a terrain-following Lagrangian control-volume coordinate system. To close the coordinate system, the model top (at a prescribed constant pressure) is also assumed to be a Lagrangian surface. With the help of the vertical Lagrangian coordinate, the time step is not constrained by the small vertical grid sizes.

However, on the other hand, the embedded regions that are needed in order to

resolve cloud-scale processes in a GCM is of much smaller grid spacing in both horizontal and vertical directions. Thus, the hydrostatic assumption is not valid in the embedded regions. However, discretizing the non-hydrostatic formulations in a vertical Lagrangian coordinate is not straightforward based on the existing discretization of the hydrostatic formulations. The Lin-Rood hydrostatic finite volume dynamical core was based on a two-step advection scheme of a staggered Arakawa C- and D-grid approach (*Arakawa and Lamb*, 1977), which was originally designed for 2D applications. Additionally, when a wave propagates through a resolution discontinuity, staggered schemes would cause more reflections compared to non-staggered schemes (*Ullrich and Jablonowski*, 2011), which is a big disadvantage for non-hydrostatic embedded models. On the other hand, when solving the full set of non-hydrostatic primitive equations in non-staggered grids, the fluxes at the finite-volume interfaces are usually calculated using a Riemann solver in order to provide stability for the numerical schemes. Unfortunately, the traditional Riemann solvers were mostly developed in field, which often dealt with high-speed flows and strong discontinuities. The computational costs of implementing the Riemann solvers in a non-staggered-grid system would be significantly larger than solutions based on the Lin-Rood hydrostatic staggered-grid system.

Fortunately, although the primitive equations governing the atmospheric flows are often nonlinear, their solutions almost never develop energetic shocks or discontinuities. A Riemann solver to be implemented in atmospheric models does not need to deal with supersonic flows or strong discontinuities, and it is applicable to assume that the temperature is a constant at the finite volume interfaces where the fluxes are to be calculated. In this thesis, we will build a computationally efficient approximate Riemann solver based on these assumptions, and implement this technique in a non-staggered grid non-hydrostatic model using a Lagrangian vertical coordinate.

1.5 An introduction of models with non-uniform grids

The non-uniform grids with increased resolution over areas of interest have been implemented in regional models for decades (*Fox-Rabinovitz et al.*, 1997). Methods that use stretched grids, nested grids and adaptive mesh refinement (AMR) methods have been discussed.

Grid nesting is a kind of non-conformal mesh set up. It is very common in mesoscale and limited-area modeling but is not often used in global modeling. In models that implement a nested grid, a smaller area, where small-scale features are better resolved, is covered using a smaller grid spacing, while its "parent" or the rest of the domain is of coarse grid. The boundary conditions for the nested grid are then initialized from coarse-grid data using refinement techniques at the beginning of each time step. The nested-grid data can also feed back onto the coarse grid through "two-way" nesting, in which the boundary conditions of the coarse grid are formed from nested-grid data.

Grid stretching is a kind of conformal mesh. It uses a smooth mapping between the original grid and the desired mesh. It is less commonly used than grid nesting, but holds some advantages over nesting. The primary benefit of a stretched grid is that the grid resolution is smoothly changed, so there are no clear boundaries between the coarse grid and refined grid. Thus no reflection is caused due to an abrupt change of grid resolution.

Adaptive mesh refinement refers to the addition of grid elements to regions, where small-scale features are explicitly simulated with sufficient resolution. The goal of adaptive mesh refinement is not to move the grid, rather to refine the grid in advance of any important physical course that needs additional grid resolution. It allows the grid to change over the process of the simulation.

1.6 Outline of the thesis

The goal of this thesis is to design an efficient numerical scheme for small-scale non-hydrostatic models. Since the purpose of this exploration is to ultimately be able to join hydrostatic and non-hydrostatic formulations as grid resolution is decreased, this non-hydrostatic approach should follow some basic design features of the existing finite volume hydrostatic dynamic core, such as implementing a vertical Lagrangian coordinate.

This thesis is organized as follows. In Chapter II, we introduce the non-staggered-grid-based 2-dimensional finite volume dynamical model using a vertical Lagrangian coordinate. In order to take advantage of the low-speed and smooth atmospheric flow, we also developed a Low Mach number Approximate Riemann Solver (LMARS) to calculate the fluxes and velocity at the interface of the control volumes. This numerical scheme is available with both hydrostatic and non-hydrostatic formulations. Chapter III is a natural extension of Chapter II, and will discuss how to connect the computational blocks of hydrostatic and non-hydrostatic formulations. The computational blocks are 2D (x-z) and aligned in the horizontal direction. The quality of the numerical algorithm is judged by observing reflections at the interface between different models. Additionally, joining blocks with different horizontal grid spacing, which is denoted as two-dimensional horizontal nesting, will also be tested. However, joining horizontally aligned blocks with different vertical grid spacing, which is denoted as vertical nesting, is a much more complex problem. Chapter IV will take a first step towards this application by examining the existing techniques that are used in dynamical cores. A "stationary rule", which indicates that no motion should be produced if a model implements the vertical nesting in a statically initialized atmosphere, is introduced. In the vertical nesting algorithms, vertical grid refinement and coarsening would be performed at the boundaries, which connect blocks of different vertical grid spacing. The polynomial interpolation based refinement will break the

stationary rule. In order to keep the refinement from breaking the stationary rule, an iteration-based vertical interpolation technique is introduced. Finally, conclusions and future work are presented in Chapter V. A substantial portion of some validation and detailed tests to the non-staggered finite volume non-hydrostatic scheme can be found in Appendix A. A polynomial interpolation of Newton's form is explained for reference in Appendix B.

CHAPTER II

A control-volume model of the compressible Euler equations with a vertical Lagrangian Coordinate

2.1 Introduction

Most of the current global climate models are based on equations that assume hydrostatic equilibrium. These models resolve processes whose horizontal scale is significantly larger than the vertical scale. However, if a global model aims to resolve motions whose horizontal and vertical scales are similar, the model must also include non-hydrostatic effects (*Daley*, 1988). One difficulty with the non-hydrostatic equations is that the fast sound waves, generated by the model's equations, can travel in all directions, vertically and horizontally, and thus, require special computational approaches, e.g. implicit methods (*Skamarock and Klemp*, 1992) and/or small time steps in explicit time stepping schemes. Therefore, the main question is how to formulate an efficient numerical scheme for small-scale non-hydrostatic models, which has the ability to correctly and stably represent the important small atmospheric interactions at the model's resolution limits (*Smolarkiewicz et al.*, 2001; *Skamarock and Klemp*, 2008).

A large number of hydrostatic and non-hydrostatic models use pressure or pressure-based terrain-following sigma or hybrid coordinates as the vertical coordinate (*Phillips*,

1956; *Smagorinsky*, 1963; *Kasahara*, 1974; *Bates et al.*, 1993; *Miller and Pearce*, 1974; *Miller and White*, 1984; *Xue and Thorpe*, 1991; *Juang*, 1992; *Skamarock and Klemp*, 2008). *Laprise* (1992) suggested that hydrostatic-pressure coordinates could be used advantageously in non-hydrostatic atmospheric models. Since the mass in the layers between adjacent coordinate surfaces is proportional to the increment in the vertical coordinate across the layer, this coordinate is often referred to as a mass coordinate. A "floating" mass coordinate, which does not allow mass to flow across vertical layers, is called a vertical Lagrangian coordinate. A few hydrostatic climate models have been developed using this formulation, including the NCAR Community Atmosphere Model (CAM) version 4 and 5 (*Neale et al.*, 2010) and NOAA Geophysical Fluid Dynamics Laboratory (GFDL) models (*Putman and Lin*, 2009; *Donner et al.*, 2011) based on the finite-volume dynamical core by *Lin* (2004). A major advantage of applying a vertical Lagrangian coordinate is that the 3D motion can be reformulated into pure horizontal 2D flow within the floating Lagrangian layers, with the resulting system closely resembling that of the shallow water system (*Lin*, 2004). Developing a non-hydrostatic model based on a vertical Lagrangian coordinate could allow the dynamics component of General Circulation Models (GCMs) to switch between different representations with hydrostatic dynamics in some regions and non-hydrostatic dynamics in other regions where higher resolution is desired. *Klemp et al.* (2007) implemented a vertical mass coordinate in a non-hydrostatic model. However, their vertical mass coordinate was not implemented using a Lagrangian formulation. The Geophysical Fluid Dynamics Laboratory Finite Volume Cubed Sphere (GFDL FVcubed) model (*Donner et al.*, 2011) which is based on a vertical Lagrangian coordinate has a nonhydrostatic option. However, an explicit divergence damping term was required to maintain its stability. Numerical damping in atmospheric modeling in general, whether accomplished implicitly through the numerical scheme or explicitly through the addition of specific terms to the equations, should only be large enough

to maintain a smooth and stable integration.

In this paper we present the development of a 2D (x-z) non-hydrostatic dynamical model based on the use of a generalized Lagrangian vertical coordinate, which was also adopted in the well-known Lin-Rood hydrostatic dynamical core (*Lin and Rood*, 1996, 1997; *Lin*, 2004). The multidimensional Flux-Form Semi-Lagrangian (FFSL) Lin-Rood dynamical core simulates the conservative, monotonic advection of the prognostic variables, and uses a floating vertically Lagrangian finite-volume (FV) representation of the model equations with a conservative remapping algorithm in the vertical direction. The Lagrangian coordinate requires periodic remapping to a reference grid in order to avoid severe deformation of the vertical mesh which would occur, for example, if layers with overlapping interfaces develop. The horizontal numerical algorithm of the Lin-Rood dynamical core is based on a staggered C-D grid approach (*Arakawa and Lamb*, 1977). This FFSL FV algorithm has been adopted in several atmospheric GCMs (e.g., CAM, GFDL).

The development of a global non-hydrostatic climate model represents a computational challenge. However, it has become feasible to embed non-hydrostatic regions within a hydrostatic model (*Yeh et al.*, 2002). While an FV model formulation based on the flux form of the equations is favorable for maintaining mass and momentum conservation when merging the hydrostatic and non-hydrostatic regions, grid staggering in a C-D fashion for the non-hydrostatic atmosphere may limit the ability of the model to perform in a stable and accurate fashion (*Skamarock*, 2008; *Ullrich et al.*, 2010; *Whitehead et al.*, 2011).

Here, we explore the use of a vertically Lagrangian nonhydrostatic model. The purpose of this exploration is to ultimately be able to join hydrostatic and nonhydrostatic formulations as grid size is decreased. Because a large number of general circulation models currently use a Lagrangian hydrostatic formulation, and our previous work explored adaptive grid techniques in a Lagrangian hydrostatic model

(*Jablonowski et al.*, 2006), it is more straightforward and easy to do this using the Lagrangian formulation in both models. This will hopefully lead to the ability to seamlessly treat both hydrostatic and non-hydrostatic regimes and allow adaptive mesh refinement using this framework. Here, we develop a method for solving the 2D (x-z) non-hydrostatic equations in Cartesian geometry using a finite-volume approach based on an unstaggered grid together with a generalized Lagrangian vertical coordinate. We do not filter acoustic waves in order to minimize any added numerical diffusion caused by filtering. Since the non-hydrostatic equations are nonlinear, no analytical solution can be used to validate the accuracy of our results. Thus we also applied our scheme using an Eulerian coordinate system to act as a reference for comparison purposes. We test the method based on both an Eulerian and a Lagrangian formulation, using the 2D warm bubble tests of *Robert* (1993) and propagating gravity waves.

Several advantages of the current scheme are:

1. No divergence damping is needed;

2. The use of an unstaggered grid simplifies the numerical representation of the advection equations;

3. We developed a fast method for evaluating the fluxes using a new approximate Riemann solver for low speed flow;

4. A Lagrangian vertical coordinate was introduced to facilitate the switching between a Lagrangian hydrostatic and non-hydrostatic treatment, since the variables in both treatments share the same definition. This approach allows us to easily join the efficient Lagrangian hydrostatic approach with the nonhydrostatic approach in the GCM;

Some advantages inherited from the finite volume framework:

5. Its built-in physical conservation laws;

6. Free of Gibbs oscillations.

One disadvantage of our current scheme is that we implement a vertically-explicit scheme. In weather or climate models where the ratio of horizontal to vertical grid spacing might be of order 10 to 100, it might be more efficient to implement a vertically-implicit scheme in order to allow longer time steps. However, in this work, in order to examine the numerical properties of our non-staggered, Riemann solver-based algorithm, we kept the numerical treatments simple and applied them to a problem with similar vertical and horizontal grid spacing. We also do not use a limiter for the reconstruction of the conservative variables profiles in order to avoid the added diffusion when limiters are adopted. This approach represents a first step towards a more generally applicable formulation.

The paper is organized as follows: in Section 2.2 we present a 2D (x-z) version of the fully compressible Euler equations. Section 2.3 introduces the numerical technique for their solution. In Section 2.4 the results of the model tests are discussed. Section 2.5 presents the conclusions. We present a converged solution for the Eulerian and Lagrangian formations at high resolution in the appendix, section A.1, and examine which formulation is more accurate formulation at low spatial resolution. This test is used to demostrate that the Eulerian solution should be considered the more accurate of the two formulations. Section A.2 of the appendix discusses the order of accuracy of our model and shows our model is fourth-order accurate in one-dimensional formulations and second-order accurate in two-dimensional formulations. Section A.3 shows that at shorter time scales, prior to the development of a significant amount of turbulence in our Guassian bubble tests, and in our gravity wave tests, the Eulerian and Lagrangian formulations without the addition of diffusion produce similar results.

2.2 Model Equations

In order to be able to embed our non-hydrostatic model into a hydrostatic model with a vertical Lagrangian coordinate, we focus on developing the non-hydrostatic model with a similar vertical coordinate. However, the Eulerian vertical coordinate can also be used based on a natural extension of the technique we introduce. Thus, we also include some results using the non-hydrostatic model with an Eulerian coordinate configuration. We introduce the control equations using the vertical Lagrangian coordinate in section 2.2.1, and the equations using the Eulerian coordinate in section 2.2.2.

2.2.1 The finite volume equations in a vertical Lagrangian coordinate

2.2.1.1 The non-hydrostatic formulation

The model equations are the fully compressible 2D (x, z) Euler conservation equations in flux form with a vertical Lagrangian coordinate. Because of the latter, the model layers are material impenetrable surfaces and the bottom surface is terrain following. This eliminates the need for the vertical advection terms and renders the equations one-dimensional. Instead, vertical transport is represented by the remapping mechanism.

The mass conservation law is written in the form:

$$\frac{\partial \pi}{\partial t} + \frac{\partial \pi u}{\partial x} = 0 \tag{2.1}$$

where π is interpreted as a pseudo-density, which is the density multiplied by the vertical geopotential gradient within the Lagrangian FV, and has the units of pressure. u is the horizontal wind. π is defined as:

$$\pi = \frac{\partial p^*}{\partial s} = -\rho g \frac{\partial z}{\partial s} = -\rho \frac{\partial \Phi}{\partial s} \tag{2.2}$$

where ρ, g, Φ, p^* are the nonhydrostatic density, gravity, geopotential, and hydrostatic pressure respectively, and s is a generalized vertical coordinate which is the integer index of each of the layer interfaces, numbered in the top-down direction. $\partial/\partial s$ thus denotes the difference of the value of any parameter between two Lagrangian layer interfaces.

The horizontal momentum equation is:

$$\frac{\partial \pi u}{\partial t} + \frac{\partial}{\partial x}\left(\pi u u + \Psi\right) = -\frac{\partial}{\partial s}\left(p \frac{\partial \Phi}{\partial x}\right) \tag{2.3}$$

where Ψ is defined as the nonhydrostatic pressure p multiplied by the vertical geopotential gradient:

$$\Psi = -p \frac{\partial \Phi}{\partial s} \tag{2.4}$$

Since π is interpreted as a pseudo-density, the left hand side of equation (2.3) is consistent with the general 1D flux form momentum equation.

The vertical momentum equation is:

$$\frac{\partial \pi w}{\partial t} + \frac{\partial \pi w u}{\partial x} = g \frac{\partial p'}{\partial s} \tag{2.5}$$

where w is the vertical velocity, $p' = p - p^*$ is the deviation from the hydrostatic

pressure, and $\partial p'/\partial s = \partial p/\partial s - \pi$ is the perturbation of the pressure from hydrostatic balance between layers.

The first law of thermodynamics provides the conservation of the potential temperature equation:

$$\frac{\partial \Theta}{\partial t} + \frac{\partial \Theta u}{\partial x} = 0 \tag{2.6}$$

where $\Theta = \pi\theta/p_0^{\kappa}$, is a scaled pseudo-potential temperature density with θ the potential temperature, p_0 a constant reference pressure at the surface with $p_0 = 1000$ hPa and κ the ratio of the gas constant R_d and the heat capacity at constant pressure c_p for dry air.

In these equations, the variables π, πu, πw and Θ are treated as prognostic variables. Two additional equations are needed to predict the full set of non-hydrostatic variables, and we add equations for the geopotential Φ and the nonhydrostatic pressure p. The geopotential advection equation comes from the definition of the vertical velocity in z-coordinates:

$$\frac{\partial \Phi}{\partial t} + u\frac{\partial \Phi}{\partial x} = wg \tag{2.7}$$

The equation for the non-hydrostatic pressure is derived from the equation of state:

$$p = \left(-\frac{R_d \Theta}{\partial \Phi/\partial s}\right)^{\gamma} \tag{2.8}$$

where $\gamma = 1/(1-\kappa)$ Since the index s is numbered in top-down direction, $\partial\Phi/\partial s$ is

negative.

Equations (2.1), (2.3), and (2.5) - (2.8) define our non-hydrostatic system.

2.2.1.2 The hydrostatic formulation

In the hydrostatic system the vertical velocity w is not treated as a prognostic variable. Our system of equations is fully consistent with a hydrostatic system if we add the assumption that $p' = 0$, i. e. p is simply the hydrostatic pressure p^*. In addition, equation (2.5) is not used, and the vertical velocity may be diagnostically derived from equation (2.7). The pressure p^* at the layer interfaces is calculated by (for level index $k > 1$):

$$(p_I^*)_k = p^*_{k-\frac{1}{2}} = p^*_{top} + \sum_{n=1}^{n=k-1} \pi_n \tag{2.9}$$

where $(p_I^*)_{k=1} = p^*_{k=\frac{1}{2}} = p^*_{top}$ is the pressure at the model top. The layer mean pressure p^* is calculated by:

$$p^* = \left(\kappa \frac{\partial p_I^* / \partial s}{\partial p_I^{*\kappa} / \partial s} \right)^{\gamma} \tag{2.10}$$

and the equation of state (2.8) is modified to calculate geopotential:

$$\frac{\partial \Phi}{\partial s} = -c_p \Theta \frac{\partial p_I^{*\kappa} / \partial s}{\partial p_I^* / \partial s} \tag{2.11}$$

Equations (2.9), (2.10) and (2.11) are the auxiliary equations needed to calculate Ψ and $-\partial \left(p \partial \Phi / \partial x \right) / \partial s$ in equation (2.3). Thus, equations (2.1), (2.3), (2.6) (2.9), (2.10) and (2.11) complete the hydrostatic system.

2.2.2 The finite volume equations in an Eulerian coordinate system

The layers in the Eulerian coordinate are stationary, so that vertical fluxes across layer boundaries take place at the layer interfaces. The prognostic variables in the Eulerian coordinate are ρ, ρu, ρw, $\tilde{\Theta}$, with the scaled potential temperature density $\tilde{\Theta} = \rho\theta/p_0^\kappa$, and the corresponding set of equations are:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho w}{\partial z} = 0 \tag{2.12}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x}\left(\rho u u + p\right) + \frac{\partial \rho u w}{\partial z} = 0 \tag{2.13}$$

$$\frac{\partial \rho w}{\partial t} + \frac{\partial \rho w u}{\partial x} + \frac{\partial}{\partial z}\left(\rho w w + p'\right) = 0 \tag{2.14}$$

$$\frac{\partial \tilde{\Theta}}{\partial t} + \frac{\partial \tilde{\Theta} u}{\partial x} + \frac{\partial \tilde{\Theta} w}{\partial z} = 0 \tag{2.15}$$

$$p = \left(R_d \tilde{\Theta}\right)^\gamma \tag{2.16}$$

Note that the position of the layers are prescribed, so there is no equation for the geopotential.

2.3 Solution Technique

In this section, we mainly discuss the discretization technique using the vertical Lagrangian coordinate configuration. The discretization using the Eulerian coordinate system is analogous.

Our dynamical core consists of six equations (2.1), (2.3), (2.5), (2.6), (2.7) and (2.8) for six variables: $Q = (\pi, u, w, \Theta, p, \Phi)$. Since Φ is not a conservative variable, equation (2.7) is updated using the advective form. We use the flux form for the equations (2.1), (2.3), (2.5) and (2.6), which has the general form:

$$\frac{\partial \mathbf{R}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \frac{\partial \mathbf{S}}{\partial s} \tag{2.17}$$

where $\mathbf{R} = (\pi, \pi u, \pi w, \Theta)$ is a vector of the π-weighted variables, $\mathbf{F} = (\pi u, \pi uu + \Psi, \pi wu, \Theta u)$ is the flux vector and $\mathbf{S} = (0, -p\partial\Phi/\partial x, gp', 0)$ is a source vector. The use of the equations in the flux form assures conservation of mass, momentum, and potential temperature.

In our numerical representation of the model equations, we use an unstaggered grid and place the variables π, πu , πw , Θ , p and $\partial\Phi/\partial s$ at the center of the cell (the A grid) so that they represent the volume mean values of these variables. The volume mean values of u and w are calculated as: $u = (\pi u)/\pi$ and $w = (\pi w)/\pi$ respectively.

An upwind method for determining the flux is used. The flux vector $\mathbf{F}$ is placed at the FV horizontal interfaces and is split according to:

$$\mathbf{F} = \left(\pi_b u_{\frac{1}{2}}, (\pi u)_b u_{\frac{1}{2}} + \Psi_{\frac{1}{2}}, (\pi w)_b u_{\frac{1}{2}}, \Theta_b u_{\frac{1}{2}}\right) = \mathbf{R_b} u_{\frac{1}{2}} + \Psi_{\frac{1}{2}}\,(0, 1, 0, 0) \tag{2.18}$$

with $u_{\frac{1}{2}}$ as the interface velocity and the index b as an upwind indicator: $b = l$, if $u_{\frac{1}{2}} > 0$, and the scalar advection terms are chosen from the left (l) side of the interface; and $b = r$, if $u_{\frac{1}{2}} \leqslant 0$, and the scalar advection terms are chosen from the right (r) side of the interface. *Lin* (2004) used a similar form to transport moisture in a general circulation model, with the horizontal velocity defined on the interface of the FV using a C-D grid and achieved a second-order overall accuracy. However, in a non-hydrostatic model, the vertical velocity needs to be taken into account, and the expansion of the staggered grid approach to three dimensions in the treatment of vertical velocity is not intuitive. *Lin* (2007) used an A-grid to treat the vertical velocity. However, the use of a staggered grid for the horizontal velocity and an

unstaggered grid for the vertical velocity introduces an inconsistency.

Liou (2006) invented an Advection Upstream Splitting Method (AUSM$^+$-up) for accurate flux calculations for all flow speeds. This method was also based on an upwind method with the flux as presented in equation (2.18). *Ullrich et al.* (2010) applied this method to a shallow water model and achieved 3rd- and 4th-order accuracy.

However, to make the AUSM$^+$-up method suitable for all flow speeds (i.e. Mach number $M \sim 1$ and $M >> 1$), the calculation of the interface flux vector is relatively complicated, and as a result, is computationally intensive. For most atmospheric phenomena, the Mach number is small ($M << 1$) and it is acceptable to assume the acoustic speed is locally constant. Thus, in order to achieve a computationally economic scheme, we invented the Low Mach number Approximate Riemann Solver (LMARS) to solve the system of equations (2.17) in the work described here. In the following sections, we show that the LMARS method has a simple form, which saves a substantial number of computational steps, but retains the accuracy of the AUSM$^+$-up method. Similar to the AUSM$^+$-up method, the LMARS consists of two steps. For the first step, the interface velocity $u_{\frac{1}{2}}$ and the pseudo-pressure $\Psi_{\frac{1}{2}}$ are calculated by solving a Riemann problem. Note that in the Eulerian coordinate the pseudo-pressure $\Psi_{\frac{1}{2}}$ is replaced by the real pressure $p_{\frac{1}{2}}$. For the second step, the fluxes are updated using equation (2.18). $\mathbf{R_b}$ can be acquired by any kind of interpolation scheme. As an aside, the AUSM$^+$-up method is slightly different in that it does not calculate the interface velocity in the first step but calculates the interface mass flux instead. To achieve low diffusivity, we use a conservative 5-point central polynomial interpolation scheme. Here we provide the expression of the interface values of η, where η could be velocity, pseudo-density or pseudo-pressure, etc., at the cell i, where i could be either the horizontal or vertical index.

$$\eta_{i-\frac{1}{2}} = -\frac{1}{20}\eta_{i-2} + \frac{9}{20}\eta_{i-1} + \frac{47}{60}\eta_i - \frac{13}{60}\eta_{i+1} + \frac{1}{30}\eta_{i+2} \tag{2.19}$$

$$\eta_{i+\frac{1}{2}} = \frac{1}{30}\eta_{i-2} - \frac{13}{60}\eta_{i-1} + \frac{47}{60}\eta_i + \frac{9}{20}\eta_{i+1} - \frac{1}{20}\eta_{i+2} \tag{2.20}$$

On the horizontal boundaries, we have 4 "ghost cells", which allow us to use the expression above for boundary cells; however, we do not use "ghost cells" for the vertical boundaries; thus, equations (2.19) and (2.20) cannot be applied at the boundaries. Here we use the top boundary cells 1 and 2 as an example to describe the top boundary condition; the 2 cells at the bottom boundary can be treated analogously with reversing the index number.

We should point out that the interpolated profile of η by the polynomials is more stable when it is evaluated near the middle of the cells, i. e. the 3rd cell's control volume interface values $\eta_{2.5}$, $\eta_{3.5}$ are evaluated using cells 1 to 5. However, if the 1st cell's control volume interface values $\eta_{0.5}$, $\eta_{1.5}$ are evaluated using cells 1 to 5, unpredictable behavior might appear. On the other hand, although a lower order of accuracy of the interpolation scheme at the boundaries does not affect the overall accuracy in the full domain, if cell 1 uses a uniform distribution of η, such that $\eta_{0.5} = \eta_1 = \eta_{1.5}$, the numerical diffusion might be too large and it might mask or filter out the waves at the boundaries. Conservatively, we use a central 3-point interpolation for cell 2:

$$\eta_{1.5} = \frac{1}{3}\eta_1 + \frac{5}{6}\eta_2 - \frac{1}{6}\eta_3 \tag{2.21}$$

$$\eta_{2.5} = -\frac{1}{6}\eta_1 + \frac{5}{6}\eta_2 + \frac{1}{3}\eta_3 \tag{2.22}$$

and a one-side interpolation for cell 1:

$$\eta_{0.5} = \frac{3}{2}\eta_1 - \frac{1}{2}\eta_2 \tag{2.23}$$

$$\eta_{1.5} = \frac{1}{2}\eta_1 + \frac{1}{2}\eta_2 \tag{2.24}$$

This treatment is not the least diffusive, but it is more generally applicable.

We derive the LMARS for the vertical Lagrangian coordinate in sub-sections 2.3.1 and 2.3.2, provide the calculation using LMARS for the Eulerian coordinate and some tuning techniques in section 2.3.3 and 2.3.4. The boundary conditions, time integration and the remapping scheme are discussed in sections 2.3.5, 2.3.6 and 2.3.7.

2.3.1 Horizontal Riemann solver

In order to derive the interface flux vector $\mathbf{F}$ in the first step, we need $u_{\frac{1}{2}}$ and $\Psi_{\frac{1}{2}}$.

When evaluating the fluxes at each cell face, only one flux vector at each cell face, namely, the vector of normal fluxes is needed. The vector of normal fluxes can be obtained by evaluating the normal speed $u_{1/2}$ and the pseudo-pressure $\Psi_{1/2}$ using a local 1-dimensional form of the Euler equation. Equation (2.3) can be written in the form of an advection-type equation, assuming a zero RHS (right hand side) for purely horizontal flow:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{1}{\pi}\frac{\partial \Psi}{\partial x} \tag{2.25}$$

Assuming a local isothermal condition, we used the gas pressure equation in the form generally defined in compressible flow (*Laprise*, 1992):

$$\frac{dp}{dt} + a^2 \rho \frac{\partial u}{\partial x} = 0 \tag{2.26}$$

where $a = \sqrt{\gamma p/\rho} = \sqrt{\gamma \Psi/\pi}$ is the Eulerian speed of sound. Taking the total derivative of equation (2.4) and using equations (2.26) and (2.2) we have:

$$\frac{\partial \Psi}{\partial t} + u \frac{\partial \Psi}{\partial x} = -a^2 \pi \frac{\partial u}{\partial x} \tag{2.27}$$

Note that the equations (2.25) and (2.27) do not need to be solved using the conservation form of the equations, since they are only used for deriving an expression to calculate the values of $u_{\frac{1}{2}}$ and $\Psi_{\frac{1}{2}}$ on the interfaces of the FV (finite volume) cells in the x-direction. In general, these two equations can be represented as:

$$\mathbf{U_t} + \mathbf{A}\mathbf{U_x} = 0, \tag{2.28}$$

$$\text{where } \mathbf{U} = \begin{pmatrix} u_{\frac{1}{2}} \\ \Psi_{\frac{1}{2}} \end{pmatrix} ; \text{ and } \mathbf{A} = \begin{pmatrix} u_{\frac{1}{2}} & 1/\pi \\ a^2\pi & u_{\frac{1}{2}} \end{pmatrix} \tag{2.29}$$

Solving this system, $\mathbf{AU} = \lambda \mathbf{U}$, for its eigenvalues, we find that $\lambda_{1,2} = u_{\frac{1}{2}} \mp a$.

We assume that we have a discontinuity at the interface of two FV cells, which comes from the interpolation of u and Ψ in different FV cells. Using the interpolation scheme given in equations (2.19), (2.20), we define the pair of the variables u and Ψ to the left of the discontinuity as $\mathbf{U_l} = [u_l, \Psi_l]$, to the right of it as $\mathbf{U_r} = [u_r, \Psi_r]$, and at the interface as $\mathbf{U} = \left[u_{\frac{1}{2}}, \Psi_{\frac{1}{2}}\right]$. Then we use the Rankine-Hugoniot "jump" conditions (*Hirsh*, 2007) to write:

$$\lambda_1 \left(\mathbf{U} - \mathbf{U_l}\right) = \mathbf{A}\left(\mathbf{U} - \mathbf{U_l}\right) \text{ and } \lambda_2 \left(\mathbf{U} - \mathbf{U_r}\right) = \mathbf{A}\left(\mathbf{U} - \mathbf{U_r}\right) \tag{2.30}$$

For $\lambda_1 = u_{\frac{1}{2}} - a$ we have:

$$\begin{pmatrix} u_{\frac{1}{2}} & 1/\pi \\ a^2\pi & u_{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} u_{\frac{1}{2}} - u_l \\ \Psi_{\frac{1}{2}} - \Psi_l \end{pmatrix} = \left(u_{\frac{1}{2}} - a\right) \begin{pmatrix} u_{\frac{1}{2}} - u_l \\ \Psi_{\frac{1}{2}} - \Psi_l \end{pmatrix}$$

$$\text{or } \Psi_{\frac{1}{2}} + \pi a u_{\frac{1}{2}} = \Psi_l + \pi a u_l \tag{2.31}$$

Similarly, for $\lambda_2 = u_{\frac{1}{2}} + a$:

$$\begin{pmatrix} u_{\frac{1}{2}} & 1/\pi \\ a^2\pi & u_{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} u_{\frac{1}{2}} - u_r \\ \Psi_{\frac{1}{2}} - \Psi_r \end{pmatrix} = \left(u_{\frac{1}{2}} + a\right) \begin{pmatrix} u_{\frac{1}{2}} - u_r \\ \Psi_{\frac{1}{2}} - \Psi_r \end{pmatrix}$$

$$\text{or } \Psi_{\frac{1}{2}} - \pi a u_{\frac{1}{2}} = \Psi_r - \pi a u_r \tag{2.32}$$

Rearranging equations (2.31) and (2.32), we obtain the values of velocity u and Ψ at interfaces of the cells:

$$\Psi_{\frac{1}{2}} = \frac{1}{2}\left(\Psi_r + \Psi_l\right) - \frac{\pi a}{2}\left(u_r - u_l\right) \tag{2.33}$$

$$u_{\frac{1}{2}} = \frac{1}{2}\left(u_r + u_l\right) - \frac{1}{2\pi a}\left(\Psi_r - \Psi_l\right) \tag{2.34}$$

The flux $\mathbf{F}$ is updated in the second step according to equation (2.18).

2.3.2 Vertical Riemann solver

To fully solve the system of equations represented by equation (2.17), we need to calculate the source vector $\mathbf{S} = (0, -(p\partial\Phi/\partial x)_{\frac{1}{2}}, gp'_{\frac{1}{2}}, 0)$. The source terms are discretized in the vertical direction, and the discretization terms $-p\partial\Phi/\partial x$ and gp' are defined at the layer interfaces. Here, the subscript $1/2$ denotes to the value at the vertical layer interfaces. Thus, we need to find the values of $p'_{\frac{1}{2}}$, $w_{\frac{1}{2}}$ and $\Phi_{\frac{1}{2}}$ at the vertical interfaces of the Lagrangian layers. Although there is no vertical flux across the Lagrangian coordinate, we can still create a Riemann problem to derive an expression to calculate the values of $w_{\frac{1}{2}}$ and $p'_{\frac{1}{2}}$ at the vertical interfaces by performing the first step of LMARS, so that $\Phi_{\frac{1}{2}}$ can be updated using equation (2.7).

Similar to the treatment for the horizontal flux evaluation, the local normal velocity and pressure at the control volume interface can be evaluated using the 1-dimensional Euler equations. We start by differentiating the state equation (2.8) along the time axis. Then taking into account equations (2.1) and (2.6), where, for pure vertical flow, π and Θ are both constant, we derive:

$$\frac{\partial p}{\partial t} = \gamma\rho p \frac{\partial w/\partial s}{\partial m/\partial s} \tag{2.35}$$

where $\partial m/\partial s = -\rho\partial z/\partial s$. Using the definition $p' = p - p^*$, we assume $\partial p^*/\partial t = 0$. This is valid because mass is conserved in each control volume, and we can write:

$$\frac{\partial p'}{\partial t} - C^2 \frac{\partial w/\partial s}{\partial m/\partial s} = 0; \text{ or } \frac{\partial p'}{\partial t} - C^2 \frac{\partial w}{\partial m} = 0 \tag{2.36}$$

where $C = \sqrt{\gamma\rho p}$ is the sound speed along the m axis in the Lagrangian vertical coordinate. From equation (2.1) and (2.5) and considering only vertical movements,

$(\partial/\partial x = 0)$, we obtain:

$$\frac{\partial w}{\partial t} - \frac{\partial p'/\partial s}{\partial m/\partial s} = 0; \text{ or } \frac{\partial w}{\partial t} - \frac{\partial p'}{\partial m} = 0; \tag{2.37}$$

Equation (2.36) and (2.37) will be used to solve the Riemann problem at the interface of the Lagrangian layers. Analogous to the horizontal case, we have:

$$\mathbf{U_t} + \mathbf{A}\mathbf{U_m} = 0, \tag{2.38}$$

$$\text{where } \mathbf{U} = \begin{pmatrix} w \\ p' \end{pmatrix}; \text{ and } \mathbf{A} = \begin{pmatrix} 0 & -1 \\ -C^2 & 0 \end{pmatrix} \tag{2.39}$$

As before, we find the eigenvalues: $\lambda_{1,2} = \mp C$. Also, we assume that we have a discontinuity of w and p' at the interface of two Lagrangian layers, and we define the pair of the variables w and p' above the discontinuity as $\mathbf{U_u} = [w_u, p'_u]$, below it as $\mathbf{U_d} = [w_d, p'_d]$, (where u, d denote up and down), and at the interface as $\mathbf{U} = \left[w_{\frac{1}{2}}, p'_{\frac{1}{2}}\right]$. Then, the Rankine-Hugoniot conditions are:

$$\lambda_1 (\mathbf{U} - \mathbf{U_u}) = \mathbf{A} (\mathbf{U} - \mathbf{U_u}) \text{ and } \lambda_2 (\mathbf{U} - \mathbf{U_d}) = \mathbf{A} (\mathbf{U} - \mathbf{U_d}) \tag{2.40}$$

Equation (2.40) can be expanded for both eigenvalues as:

$$\begin{pmatrix} 0 & -1 \\ -C^2 & 0 \end{pmatrix} \begin{pmatrix} w_{\frac{1}{2}} - w_u \\ p'_{\frac{1}{2}} - p'_u \end{pmatrix} = -C \begin{pmatrix} w_{\frac{1}{2}} - w_u \\ p'_{\frac{1}{2}} - p'_u \end{pmatrix}$$

$$\text{or } p'_{\frac{1}{2}} - C w_{\frac{1}{2}} = p'_u - C w_u \tag{2.41}$$

$$\begin{pmatrix} 0 & -1 \\ -C^2 & 0 \end{pmatrix} \begin{pmatrix} w_{\frac{1}{2}} - w_d \\ p'_{\frac{1}{2}} - p'_d \end{pmatrix} = C \begin{pmatrix} w_{\frac{1}{2}} - w_d \\ p'_{\frac{1}{2}} - p'_d \end{pmatrix}$$

$$\text{or } p'_{\frac{1}{2}} + C w_{\frac{1}{2}} = p'_d + C w_d \tag{2.42}$$

Rearranging equations (2.41) and (2.42), we obtain:

$$p'_{\frac{1}{2}} = \frac{1}{2}\left(p'_d + p'_u\right) + \frac{C}{2}\left(w_d - w_u\right) \tag{2.43}$$

$$w_{\frac{1}{2}} = \frac{1}{2}\left(w_d + w_u\right) + \frac{1}{2C}\left(p'_d - p'_u\right) \tag{2.44}$$

The overall stability is not sensitive to the representation of the horizontal velocity at the interface. The simplest representation $u_{\frac{1}{2}} = (u_d + u_u)/2$, is sufficient. With $u_{\frac{1}{2}}$ and $w_{\frac{1}{2}}$, the geopotential $\Phi_{\frac{1}{2}}$ can be updated using equation (2.7). These values can then be used to calculate the source vector $\mathbf{S}$.

2.3.3 The LMARS in Eulerian coordinate

The system of equations in the Eulerian coordinate is:

$$\frac{\partial \mathbf{R}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{H}}{\partial z} = 0 \tag{2.45}$$

where $\mathbf{R} = (\rho, \rho u, \rho w, \tilde{\Theta})$, $\mathbf{F} = (\rho u, \rho u u + p, \rho w u, \tilde{\Theta} u)$, and $\mathbf{H} = (\rho w, \rho w u, \rho w w + p', \tilde{\Theta} w)$, with $\tilde{\Theta} = \rho\theta / p_0^{\kappa}$. Similar to the representation in the Lagrangian system, we derive the left and right flux vectors as:

$$\mathbf{F} = \mathbf{R}_b u_{\frac{1}{2}} + p_{\frac{1}{2}} (0, 1, 0, 0) \tag{2.46}$$

$$\mathbf{H} = \mathbf{R}_c w_{\frac{1}{2}} + p'_{\frac{1}{2}} (0, 0, 1, 0) \tag{2.47}$$

where,

$$b = \begin{cases} l, & \text{if } u_{\frac{1}{2}} > 0 \\ r, & \text{if } u_{\frac{1}{2}} \leqslant 0 \end{cases} \text{ and } c = \begin{cases} d, & \text{if } w_{\frac{1}{2}} > 0 \\ u, & \text{if } w_{\frac{1}{2}} \leqslant 0 \end{cases} \tag{2.48}$$

The derivation of the LMARS is very similar to that given in section 2.3.1. Here we provide the result:

$$p_{\frac{1}{2}} = \frac{1}{2} (p_r + p_l) - \frac{\rho a}{2} (u_r - u_l) \tag{2.49}$$

$$u_{\frac{1}{2}} = \frac{1}{2} (u_r + u_l) - \frac{1}{2\rho a} (p_r - p_l) \tag{2.50}$$

$$p'_{\frac{1}{2}} = \frac{1}{2} (p'_d + p'_u) + \frac{\rho a}{2} (w_d - w_u) \tag{2.51}$$

$$w_{\frac{1}{2}} = \frac{1}{2} (w_d + w_u) + \frac{1}{2\rho a} (p'_d - p'_u) \tag{2.52}$$

Although the system represented by equations (2.17) and (2.45) are only described in 2D, the extension of the solution to multi-dimensions follows the standard procedure using directional splitting.

2.3.4 The tuning of LMARS

The equations (2.33) (2.34) (2.43) (2.44) and (2.49) to (2.52) are in similar form, if we denote $\tilde{p}$, $\tilde{\rho}$, $\tilde{u}$, and $\tilde{a}$ as the pressure/pseudo-pressure, density/pseudo-density, velocity perpendicular to the flux interface, and speed of sound. The equations for

the velocity and pressure at the flux interface can be expressed as:

$$\tilde{p}_{\frac{1}{2}} = \frac{1}{2}(\tilde{p}_{upwind} + \tilde{p}_{downwind}) + \frac{\tilde{\rho}\tilde{a}}{2}(\tilde{u}_{upwind} - \tilde{u}_{downwind}) \tag{2.53}$$

$$\tilde{u}_{\frac{1}{2}} = \frac{1}{2}(\tilde{u}_{upwind} + \tilde{u}_{downwind}) + \frac{1}{2\tilde{\rho}\tilde{a}}(\tilde{p}_{upwind} - \tilde{p}_{downwind}) \tag{2.54}$$

The second terms on the right hand side are associated with implicit diffusion effects. This diffusion can be minimized under conditions, which do not have strong vertical convection and have continuous physical variables with small perturbations, by introducing the form:

$$\tilde{p}_{\frac{1}{2}} = \frac{1}{2}(\tilde{p}_{upwind} + \tilde{p}_{downwind}) + \beta\frac{\tilde{\rho}\tilde{a}}{2}(\tilde{u}_{upwind} - \tilde{u}_{downwind}) \tag{2.55}$$

$$\tilde{u}_{\frac{1}{2}} = \frac{1}{2}(\tilde{u}_{upwind} + \tilde{u}_{downwind}) + \beta\frac{1}{2\tilde{\rho}\tilde{a}}(\tilde{p}_{upwind} - \tilde{p}_{downwind}) \tag{2.56}$$

with β a variable diffusion parameter, $\beta \leqslant 1$. Using $\beta = 1$ provides stability for most situations, but introduces more diffusion. Using a smaller value of β, we are able to achieve smaller diffusion in the LMARS scheme, but should be tested on a case by case basis.

2.3.5 Boundary Conditions

2.3.5.1 Horizontal boundary conditions

Since all the prognostic variables are defined on an unstaggered grid, we can apply standard boundary conditions in the horizontal direction (the position of Φ is vertically staggered, however, it is unstaggered in the horizontal direction). In the bubble tests shown in section 2.4, a reflective boundary condition is applied by mirroring the ghost cells. The ghost cells are extra grid cells on the boundaries used

for the interpolation of the variables.

In the gravity wave test shown in section 2.4, a periodic boundary condition is applied.

2.3.5.2 Vertical boundary conditions

In the vertical Lagrangian coordinate, the top/bottom boundary conditions are derived from the equations (2.41) and (2.42). At the bottom of the model we adopt a reflective surface by setting:

$$w = 0 \tag{2.57}$$

$$p' = p'_u - Cw_u \tag{2.58}$$

At the top of the model, we have two options: either a "rigid lid" condition or an "open boundary" condition to allow waves and disturbances originating within the model domain to leave the domain without affecting the interior solution.

For the "rigid lid" condition, we set reflective conditions similar to the bottom boundary condition:

$$w = 0 \tag{2.59}$$

$$p' = p'_d + Cw_d \tag{2.60}$$

For the open boundary condition we set:

$$w = w_d \tag{2.61}$$

$$p' = p'_d \tag{2.62}$$

Although we do not have any vertically-oriented ghost cells, however, using equations (2.43) (2.44) and (2.61) (2.62) we can derive that the upper and lower side values at the boundary interface are identical, which has an effect similar to that of building a ghost cell using the non-reflective boundary condition.

The vertical boundary condition in the Eulerian coordinate are all set as reflective boundary conditions.

2.3.6 Time integration

The application of a two step prediction-correction time marching scheme with the conservative 5-point polynomial interpolation scheme, will lead to different results if different time-steps are used, because the two-step prediction-correction time marching scheme is only 2nd-order accurate in time (unless very small time steps are used). However, when paring the conservative 5-point polynomial interpolation scheme with a 4-step Runge-Kutta method, the choice of time step would not affect the result as long as the $CFL < 1$ condition is met. In two dimensions, the interface values may be regarded as having been averaged along the interface. The interface flux based on such an average, though, is not the proper flux average along the interface, because the flux is not a linear function of the state quantities. So fourth-order accuracy is downgraded to second-order accuracy. However, the higher-order interpolation scheme does achieve low diffusivity of the overall scheme so that we are able to observe small-scale structures if we use a small grid size configuration.

We use a 4-step Runge-Kutta method to integrate the equation for the prognostic

variables in time. In the nonhydrostatic Lagrangian version, the prognostic variables are $\mathbf{U} = (\pi, u, w, \Theta, \Phi)$ (equation (2.7) is included, which is not in flux form). The prognostic variable vectors in the Eulerian version and the hydrostatic version are similar. The differential form of U is:

$$\frac{d\mathbf{U}}{dt} = RHS\left(\mathbf{U}\right) \tag{2.63}$$

where RHS stands for "right hand side" of the system of differential equations (2.1), (2.3), (2.5), (2.6) and (2.7), and these functions are independent with variable of time . The 4th-order in time 4-step Runge-Kutta method is:

$$\mathbf{U}^1 = \mathbf{U}(t) + \frac{\Delta t}{2} RHS\left(\mathbf{U}(t)\right) \tag{2.64}$$

$$\mathbf{U}^2 = \mathbf{U}(t) + \frac{\Delta t}{2} RHS\left(\mathbf{U}^1\right) \tag{2.65}$$

$$\mathbf{U}^3 = \mathbf{U}(t) + \Delta t RHS\left(\mathbf{U}^2\right) \tag{2.66}$$

$$\mathbf{U}(t+\Delta t) = \mathbf{U}(t) + \frac{\Delta t}{6}\left(RHS\left(\mathbf{U}(t)\right) + 2RHS\left(\mathbf{U}^1\right) + 2RHS\left(\mathbf{U}^2\right) + RHS\left(\mathbf{U}^3\right)\right) \tag{2.67}$$

where Δt is the time step. More standard 4-step Runge-Kutta methods are described in *Durran* (2010).

2.3.7 Vertical Remap

When using the vertical Lagrangian coordinate, the Lagrangian surfaces that bound an atmospheric layer deform and need to be re-mapped onto the original coordinates. The volume mean prognostic variables π, πu, πw, and Θ are remapped. Taking π as an example, our procedure is as follows: 1. Integrate π in the top-down direction to build a continuous profile. If we denote the vertical integral to grid level $k + 0.5$ as $m_{k+0.5} = \sum_{i=1}^{k} \pi_i$, with $m_{0.5} = 0$, then $m_{k+0.5}$ is defined at the layer in-

terface with vertical location represented by the geopotential $\Phi = gz$, where z is the height. 2. The profile of $m_{k+0.5}$ is acquired using a 7-point polynomial interpolation scheme, with a Newton form polynomial interpolation (*Yang*, 2001), i. e. to get m at location $\Phi_{ref_{k+0.5}}$ i. e. $(m_{new_{k+0.5}}, \Phi_{ref_{k+0.5}})$, we use the following inputs to the polynomial interpolation $(m_{k-2.5}, \Phi_{k-2.5})$, $(m_{k-1.5}, \Phi_{k-1.5})$, $(m_{k-0.5}, \Phi_{k-0.5})$, $(m_{k+0.5}, \Phi_{k+0.5})$, $(m_{k+1.5}, \Phi_{k+1.5})$, $(m_{k+2.5}, \Phi_{k+2.5})$, $(m_{k+3.5}, \Phi_{k+3.5})$. Near the top and bottom of the domain, the top most and bottom most 7 points are used for the input to the polynomial interpolation, i. e. $(m_{new_{1.5}}, \Phi_{ref_{1.5}})$, $(m_{new_{2.5}}, \Phi_{ref_{2.5}})$, $(m_{new_{3.5}}, \Phi_{ref_{3.5}})$ are all calculated using the profile built by the points $(m_{0.5}, \Phi_{0.5})$, $(m_{1.5}, \Phi_{1.5})$, $(m_{2.5}, \Phi_{2.5})$, $(m_{3.5}, \Phi_{3.5})$, $(m_{4.5}, \Phi_{4.5})$, $(m_{5.5}, \Phi_{5.5})$, $(m_{6.5}, \Phi_{6.5})$ using a polynomial interpolation. 3. Using the profile $(m_{new_{k+0.5}}, \Phi_{ref_{k+0.5}})$, the remapped π is calculated from $\pi_{new_k} = m_{new_{k+0.5}} - m_{new_{k-0.5}}$. Steps 1 to 3 complete the remap for π. Since the values of $m_{k+0.5}$ at the top and bottom remain unchanged during the remapping process, the total mass $\sum \pi/g$ is automatically conserved. Similarly, the total momentum is also conserved (conservation of $\sum \pi u$ and $\sum \pi w$), and no extra heat is introduced into the system (conservation of $\sum \Theta$). For simplicity, there is no limiting mechanism used in step 2.

2.4 Tests and Results

2.4.1 Robert's warm bubble tests

We test our non-hydrostatic approach for solving the equations for the 2D (x, z) non-hydrostatic atmosphere using two standard tests from *Robert* (1993). These tests are for two different types of warm bubbles: the "Gaussian" and "uniform" bubbles, which rise in an isentropic atmosphere (303.15 K) within a closed box.

The Gaussian bubble is placed in a 1 km wide by 1.5 km high box and is represented by a perturbation of the potential temperature of the form:

$$\theta' = \begin{cases} A, & \text{if } r \leqslant a \\ Ae^{-(r-a)^2/s^2}, & \text{if } r > a \end{cases} \tag{2.68}$$

where $r^2 = (x - x_0)^2 + (z - z_0)^2$ is the distance from the bubble center; $x_0 = 500$ m; $z_0 = 260$ m; $A = 0.5$ K; $a = 50$ m; and $s = 100$ m. The uniform bubble with a radius of 250 m and with an initial 303.65 K "flat" potential temperature was positioned at $(x_0, z_0) = (500, 260)$ m in a 1 km by 1 km box.

2.4.1.1 Eulerian framework vs. Lagrangian framework

Although the Lagrangian framework is useful in reducing the 2D flow to 1D flow, and hence enhancing the computational efficiency of the solution, it allows the finite volume cells to deform from their rectangular shape, which will introduce some geometric errors when the Lagrangian interfaces of the layers are significantly distorted. Using the Eulerian framework avoids this problem because the rectangular shape of the grid cells is fixed. The LMARS solver can calculate fluxes in both Lagrangian coordinates and Eulerian coordinates using a similar framework, so the results in the Eulerian framework can serve as the reference solution.

Figure 2.1 shows the potential temperature perturbation in the experiments. The output times selected are the same as those shown in *Robert* (1993). The first row shows the Gaussian bubble results from the Eulerian configuration, while the second row shows the same results from the Lagrangian configuration. The output times for the Gaussian bubble results are 0, 6, 12 and 18 min. The last row shows the uniform bubble test, with the left two sub-plots from the Eulerian configuration while the right two are from the Lagrangian configuration. The grid size for all the results was $\Delta x = \Delta z = 5$ m, and the time step $\Delta t = 0.007$ s scales with the grid spacing. The color range is presented from 0 to 0.5 K for comparison purposes with the results

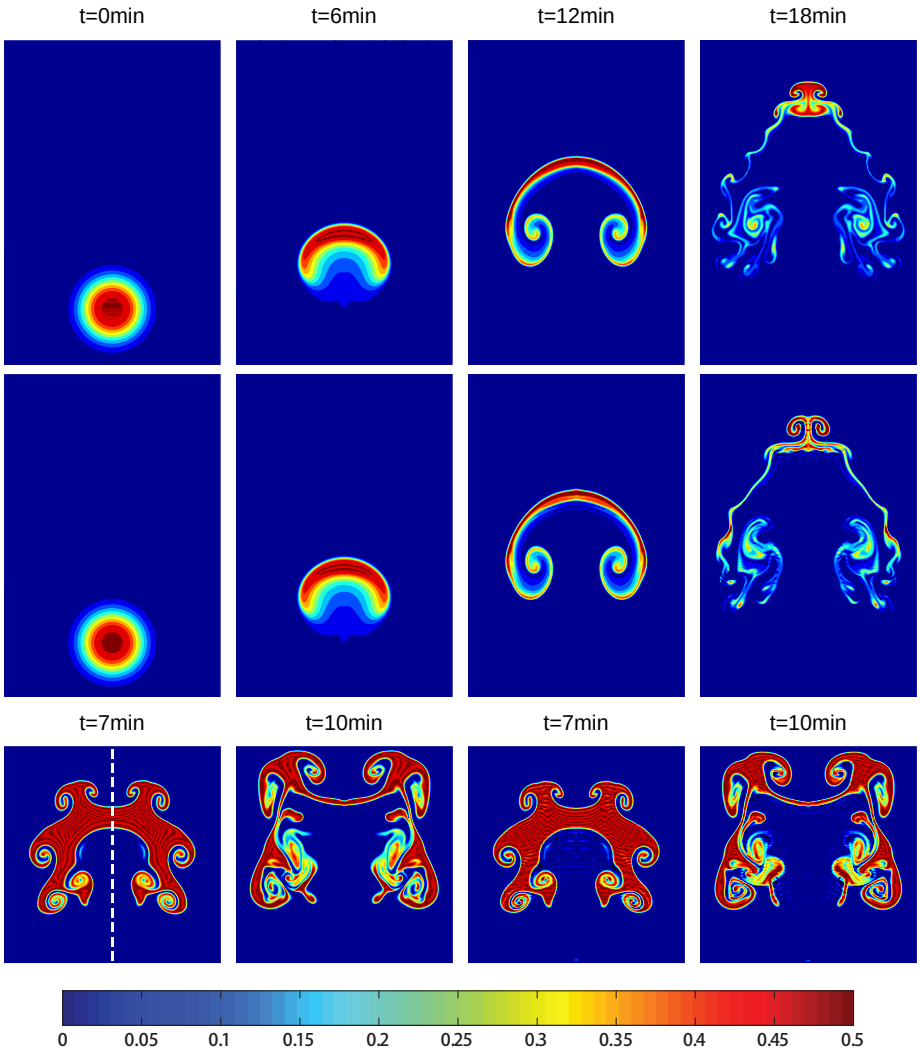

Figure 2.1: The upper two rows show the potential temperature (PT) (in K) for a Gaussian bubble perturbation in a 1 km by 1.5 km domain using the Eulerian coordinate (row 1) and the Lagrangian coordinate (row 2). The bottom row shows an initial uniform bubble perturbation in a 1 km by 1 km domain using the Eulerian coordinate (left 2) and the Lagrangian coordinate (right 2). The grid spacing of all results is 5 m. The cross-section of the PT perturbation along the dashed line in the lower left panel is presented in figure 2.2

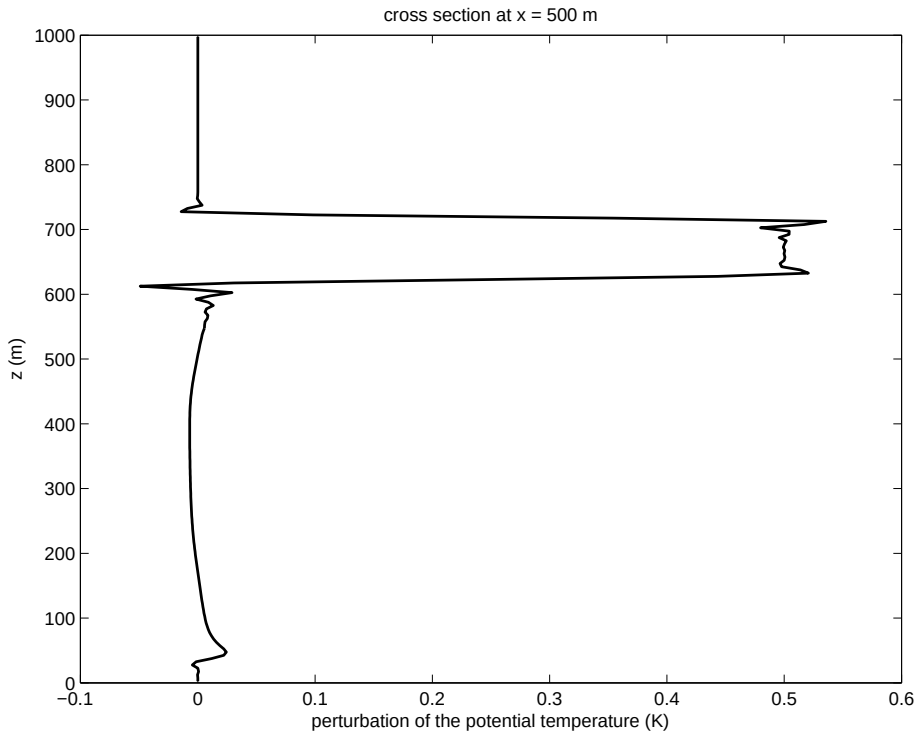

Figure 2.2: Cross-section of the potential temperature perturbation (in K) at x = 500 m of the uniform bubble test after 7 min using the Eulerian coordinate.

shown by *Robert* (1993). The output times for the uniform bubble results are at 7 and 10 min. The AUSM$^+$-up method was also tested with the Eulerian configuration and shows almost identical results (figures omitted), but requires about 50% more computer time compared to the LMARS method.

Since no limiter is applied in either interpolation scheme for the variables or remapping, 2-grid-size waves can be observed. These waves do not grow or cause instability. Figure 2.2 provides a clearer picture of these small-scale oscillations. It shows the cross-section at the center of the uniform bubble test for the 7 min plot (the dashed line in the lower left plot in figure 2.1). The oscillations are especially present near the sharp edges of the rising uniform bubble.

In figure 2.1, the 6 and 12 min results for the two different coordinate configurations agree well with each other and with the corresponding solutions by *Robert* (1993). However, at 18 minutes, the Kelvin-Helmholtz instability starts to appear,

and the results are different between the 2 different schemes, especially for the shape of the bubble head. Since we have not applied any diffusion or limiters in our experiments, the numerical solutions are only expected to agree at finite times and may diverge depending on the implicit damping and dispersion characteristics of the chosen algorithms. Similar behavior is also found when comparing the results of the uniform bubble: the general shapes of the uniform bubbles are similar; however, the Kelvin-Helmholtz instability is resolved differently between different schemes at longer times. The bubble head and bubble's width show a slight difference between our results and those of *Robert* (1993), which could be due to different applications of the reflective boundary conditions.

In this particular case, because the Eulerian framework is able to maintain a rectangular shape for the control volume at each time step, the fluxes are always perpendicular to the control volume interfaces. Moreover, we note that the error induced by the remapping scheme is not included in the Eulerian framework. Therefore, we consider the Eulerian version to be the more accurate solution and use it as the reference below. In the appendix, Section A.1, we also conducted a sensitivity test in which we added strong viscosity to both the Lagrangian and Eulerian formulations. With the addition of viscosity, the results for both formulations converge at high resolution. But at coarse resolution, the plots using the Eulerian formulation are slightly closer to the high-resolution solution. In general, if sub-grid turbulence is added to a model using a parameterization, both Eulerian and Lagrangian frameworks should provide converged results. At shorter time scales, before any turbulence develops in the Guassian bubble tests (i.e. at 6 and 12 minutes) or for the results of the gravity wave tests in the next section, the Eulerian and Lagrangian results are similar. The difference between the Eulerian and Lagrangian results without explicit diffusion, which related to the plots mentioned above are compared in Section A.3 of the Appendix.

2.4.1.2 Grid spacing versus numerical diffusion

We do not need to apply any diffusion such as divergence damping to stabilize our numerical scheme because our LMARS solver provides the necessary stability. Since LMARS is applied in both the horizontal and vertical directions, any implicit numerical diffusion is consistent in all directions. As the diffusion is determined by the difference of the pressure and velocities at the interface of the FV cells, which are acquired by interpolation, a high-order interpolation scheme will lead to less diffusion.

The numerical diffusion is not a linear function of the grid size. As a result, when the grid spacing is decreased, the result may not converge. Instead, numerical viscosity will be decreased strongly and an ultra-low diffusive result will be found. Figure 2.3 shows a plot at 18 min of the Gaussian bubble using the Eulerian configuration and the result of the uniform bubble at 7 min in the Lagrangian configuration with a grid size of 10 m and 2.5 m, respectively. The 10 m run is actually more diffusive than the results of *Robert* (1993), however, the 2.5 m results are of high quality. With smaller grid size, the smaller scale Kelvin-Helmholtz waves can be resolved in the Gaussian bubble. In addition, in the uniform bubble test, the discontinuity at the bubble edge is very sharp.

These tests show that we are able to resolve the warm bubble tests with small numerical diffusion using a high-order interpolation scheme for the prognostic variables and a small grid size. Although the amount of numerical diffusion that can be tolerated in the solution depends on the specific application, and for some applications, an economic computational performance is preferred, we have shown that LMARS provides stable solutions with a small amount of implicit numerical diffusion. Our algorithm can be "downgraded" by using a lower order interpolation scheme for the prognostic variables without any special treatment such as introducing a divergence damping term for stability. An implicit time marching scheme might also be used and equipped with the LMARS numeric solver to filter out acoustic waves and achieve a

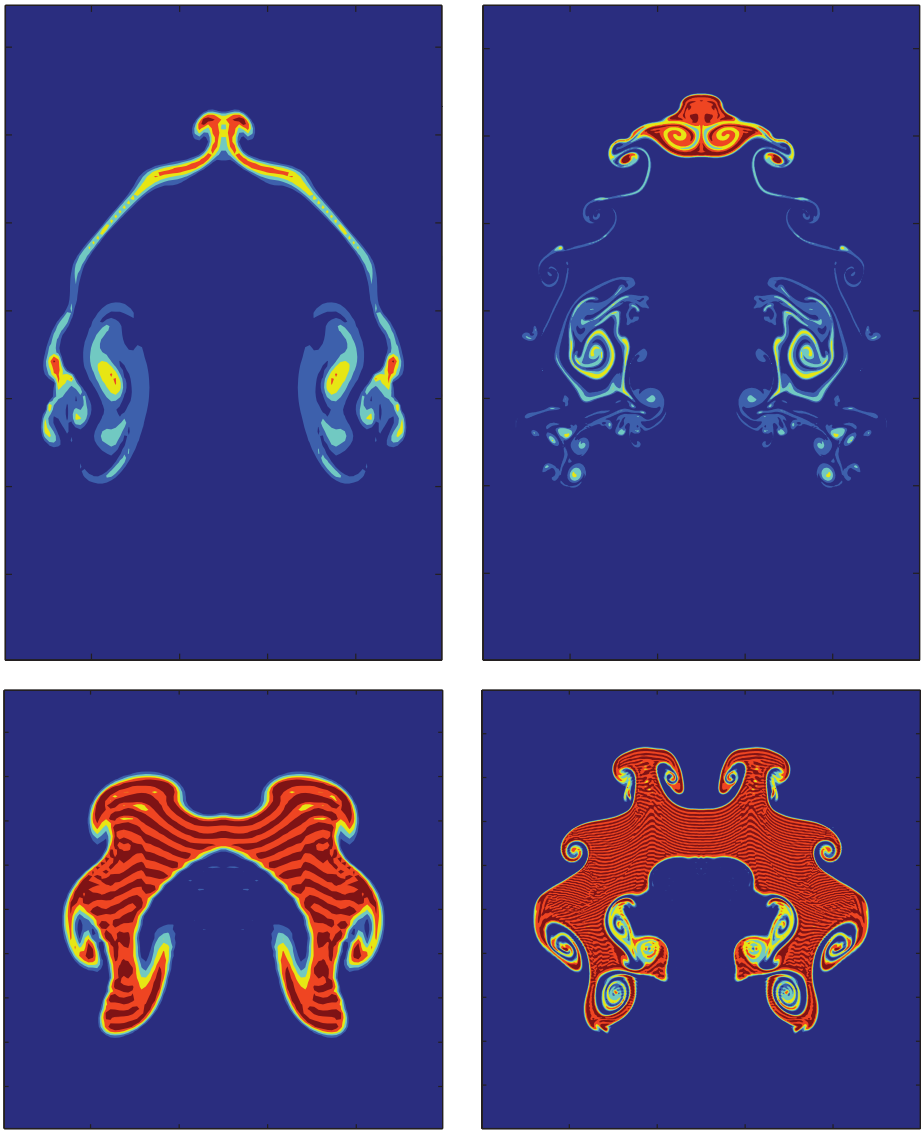

Figure 2.3: Plot of the Gaussian bubble at 18 min using the Eulerian configuration (row 1) and the result of the uniform bubble at 7 min in the Lagrangian configuration (row 2) with a grid size of 10 m (left column) and 2.5 m (right column).

CFL number larger than one, but this technique will introduce additional diffusion into the solution. The discussion of time-marching schemes is beyond the scope of this work.

2.4.2 Gravity wave test

Figure 2.4 presents the results from a gravity wave test using the vertical Lagrangian coordinate, Eulerian coordinates, with the nonhydrostatic and hydrostatic setups, which is very similar to the test case developed by *Skamarock and Klemp* (1994). The vertical domain size is 10 km and the horizontal domain size is 300 km. The background atmosphere has a constant Brunt-Väisälä frequency of 10^{-2} s^{-1} with a surface temperature of 300 K, and the surface pressure is 10^5 Pa. The grid spacings are $\Delta x = \Delta z = 1$ km and time step is 2 s. The initial horizontal wind is 20 ms^{-1} and a periodic horizontal boundary condition is used. The waves are excited by an initial θ perturbation of the form:

$$\theta' = \Delta\theta_0 \frac{\sin \frac{\pi z}{H}}{1 + (x - x_c)^2/a^2} \tag{2.69}$$

where $\Delta\theta_0 = 10^{-2}$ K, $H = 10$ km, $a = 5$ km, and $x_c = 100$ km.

The original test published in *Skamarock and Klemp* (1994) used a Boussinesq model and a rigid lid boundary condition. A rigid lid boundary condition is used in the test using the Eulerian coordinates. However, the way we apply the hydrostatic approximation requires that we use a free surface at the top of the model, so an open boundary condition is applied in all Lagrangian coordinate tests. Additionally, since the Coriolis force would bring in one more equation for v, and this would break the momentum conservation, we did not include this in our test, and set the Coriolis parameter to zero. This is different from the approach in *Skamarock and Klemp*

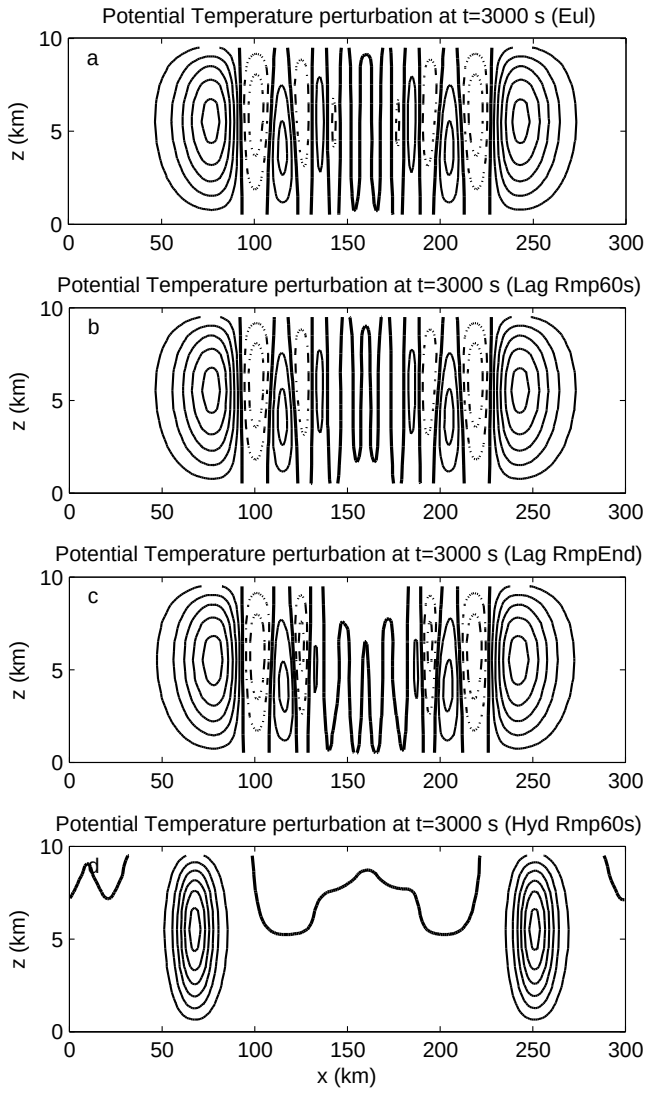

Figure 2.4: The potential temperature perturbation (in K) at $t = 3000$ s in the Gravity wave test. The configurations are a) nonhydrostatic with Eulerian coordinates. b) nonhydrostatic with vertical Lagrangian coordinate and a remap frequency of 60 s. c) same as b), but only a single remap at the end of the simulation. d) hydrostatic with vertical Lagrangian coordinates with 60 s remap frequency. The contour interval is 0.0005 K, the bold line is the zero contour, the solid lines are positive, dash-dotted lines are negative. The grid spacing is $dx = dz = 1$ km.

(1994). However, *Giraldo and Restelli* (2008) conducted a similar test, without the Coriolis force, with the full Euler equations. The perturbation potential temperature at $t = 3000$ s is shown in figure 2.4 for comparison with the nonhydrostatic results of *Giraldo and Restelli* (2008). Both the hydrostatic and the nonhydrostatic simulations start from identical initial conditions.

The upper subplot in figure 2.4 is the potential temperature perturbation using the nonhydrostatic configuration in the Eulerian coordinates. This result is in good agreement with the results of *Giraldo and Restelli* (2008), who used a local spectral method. Our result is slightly more damping, which may due to the fact that *Giraldo and Restelli* (2008) used a 250 m resolution and 10th-order polynomials. Rows b and c present the results of the nonhydrostatic configuration in the vertical Lagrangian coordinate. The remap frequency of the result in row b is every 60 s, while that in row c is only remapped at the end of the simulation. The lower subplot in figure 2.4 uses a hydrostatic configuration with remapping every 60 s.

This test is dominated by the evolution of the gravity wave. The horizontal background velocity of the fluid, which is close to the gravity wave speed, is much greater than the vertical velocity. So even with different boundary conditions at the top of the model, there is no visible deformation at the model top, and the reflection effect is small. For the same reason, the FV vertical deformation is very small and the difference due to different remapping frequencies in this test is negligible. The only differences are the zero-contour differences in the middle of the domain. The model's Eulerian version does not involve remapping, so it is least diffusive. However, since the model's Lagrangian version translates the vertical motion into the finite volume deformation, and it has no vertical flux terms to calculate, it is more computationally efficient than the Eulerian version.

Row d is the gravity wave simulation using the hydrostatic formulations in the vertical Lagrangian coordinate. The same grid spacing and time step is used. Since

the hydrostatic formulations does not permit vertical acoustic wave, the dispersive wave trains observed in the non-hydrostatic model are not generated in the hydrostatic mode.

2.5 Conclusions

The finite volume scheme using a vertical Lagrangian coordinate has proven to be very useful in the GCM modeling community and has been applied in both the CAM and GFDL GCMs. Here, we used a similar structure for the development of a nonhydrostatic dynamical core in 2D (x, z) Cartesian geometry.

In this work, we developed the equation sets for a generalized vertical coordinate, and present these in the nonhydrostatic Lagrangian and Eulerian form. We also show their hydrostatic variant. The Arakawa A-gridding is used in our approach to keep density, velocities and temperature all volume mean variables. With the A-grid, the fluxes between the FVs or the vertical movement of the Lagrangian layer interfaces are calculated by Riemann solvers.

The Low Mach number Approximate Riemann Solver is designed for atmospheric fluid motions, and is extremely efficient when compared with the traditional (approximate/exact) Riemann solvers. With the introduction of LMARS, the numerical treatment of the fluxes is decoupled from the governing equations. The algorithm developer can chose variables or equations based on the specific physical requirement without changes to the numerical properties of the system.

No limiters or divergence damping are included in the numerical algorithm. The treatment for the interpolation and remapping are all based on polynomial interpolation. All plots present the "pure" effect of LMARS, and the 2-grid-size wave does not grow or cause any instability. In real applications however, limiters are desirable to prevent negative values of density, pressure or tracers, like water-vapor mixing ratios. Also, a better remap scheme will be required when simulating physical phenomena at

the boundaries of the domain. But both limiters and remapping schemes cause extra diffusion in the numerical system, which, however, do not introduce any instability properties. In future studies, we will introduce limiters and an improved remapping scheme into our system.

LMARS is a flexible way to ensure stability for finite volume numerical schemes both in Eulerian and Lagrangian configurations. Of course for the simulations of clouds in a model, some form of explicit diffusion needs to be added to parameterize the sub-grid scale turbulence. However, with the built-in stabilizing mechanism in LMARS, there is no need to add an explicit diffusion term to the numerical scheme in for stability purpose. The numerical diffusion of the scheme can be decreased by using a high-order interpolation scheme or a smaller grid size or by reducing the diffusion factor in the LMARS scheme. Although our approach utilizes the A-grid, the LMARS technique can also be used to provide the FV cell interface velocities and pressure for the C-grid and D-grids. As a result, it is possible to integrate this method into the Lin-Rood scheme which is applied in many GCMs. However, taking this next step would also entail exploring vertical implicit methods, which we have not yet done. In the future, we will explore an application which couples the nonhydrostatic regime and algorithm to a hydrostatic regime using a vertical Lagrangian coordinate.

CHAPTER III

Connecting hydrostatic and non-hydrostatic models in a vertical Lagrangian coordinate system

3.1 Introduction

Atmospheric flows can be simulated using a numerical model based on Navier-Stokes equations. Although the full system of Navier-Stokes equations could describe most general fluid flows, it is almost impossible to use one single numerical model to simulate all the atmospheric processes. For example, a typical cumulonimbus cloud has a horizontal scale of 1 km, while some large systems such as hurricanes or synoptic cyclones might have scales of order 10^5 to 10^6 m (*Holton and Hakim*, 2012). Cloud resolving models (CRMs) can be used to simulate cloud processes using a time step less than 1 s, and horizontal resolution of about 40 m. However, extending CRMs of such high resolution to the global domain size is far beyond today's computational capability. A typical global resolving model (GCM) uses a time-step of 30 min and horizontal resolution of at least 10 km. The cloud processes are described using empirically tuned parameterizations in the GCMs. It is widely agreed that feedbacks involving clouds are among the most uncertain aspects of climate modeling. The parameterizations used in most climate models do not accurately resolve the interactions of aerosols, radiation, microphysics and dynamics at least if judged by

the ability of these models to reproduce the effects in cloud resolving models or large eddy simulations (*Johnson*, 2005; *Lee and Penner*, 2010).

One promising way for allowing clouds to be followed within global models is the so-called "super-parameterization" (*Khairoutdinov and Randall*, 2001; *Khairoutdinov et al.*, 2005; *Wang et al.*, 2011). Super-parameterizations are used as the basis for increasing the accuracy of climate models that can be run more efficiently to simulate climate at a lower cost than a true global CRM, but more realistically than a traditional climate model. But even for these models aerosol/cloud interactions may be poorly resolved if grid resolutions are limited to those above 250 m. Thus, the fidelity of these parameterizations for the treatment of aerosol/cloud interactions is not yet known.

Alternatively, we could use non-uniform grids in climate modeling that allow us to use the non-approximate model to locally resolve the small-scale features of the underlying physics principles with increased resolution, so that the corresponding parameterizations can locally be dropped. Non-uniform grids are commonly applied using nested and stretched grid techniques, both of which can be implemented in a statically or dynamically adaptive way (see also *Fox-Rabinovitz et al.* (1997) for an overview). The adaptive mesh refinement (AMR) technique is similar to the nested grid technique, except it adds and removes grid points locally without affecting the resolution in distant model domains and does not require a priori knowledge of future refinement regions. Non-uniform grids are extensively used in numerical modeling of the atmosphere for a wide range of applications (*St-Cyr et al.*, 2008; *Prusa and Smolarkiewicz*, 2003; *Mass et al.*, 2002; *Giorgi and Mearns*, 1999; *Krol et al.*, 2005).

Like the super-parameterization, non-uniform grid techniques allow us to resolve sub-scale physical processes within climate models without applying high resolutions on a global scale. We prefer the more flexible AMR techniques because the model regions at high resolution are kept at a minimum. Compared to the grid stretching

techniques, nested grids only requires a local modification of the mesh when additional resolution is added or removed, so it is more suited to adaptive mesh refinement. Thus, this work is more focused on the nested grid type non-uniform grid applications. One issue is that, the varying resolution of the nested grids can cause artificial reflections. As an example, a traveling wave may undergo false reflections or aliasing when propagating from the fine grid to the coarse domain. One way to remedy this problem is to use one-way (parasitic) nesting between the fine and coarse grid interface. One-way nesting does not allow any nested-to-coarse-grid communication, and the solution on the coarse grid is independent of that on the fine grid. However, *Harris and Durran* (2010) showed that two-way nesting, in which the solution on the coarse grid is continually replaced by that on the nested grid wherever the two grids coincide, was found to be generally superior to one-way nesting. Thus in this work, we pursue the non-uniform grid application using a two-way nested grid approach.

Many GCMs have the option to use finite-volume based dynamical cores. The reason for using finite-volume methods is because it preserves positive-definite results, guarantees conservation of mass and maintains high accuracy (*Rood*, 1987). One finite volume based dynamical core is developed in the series of papers (*Lin and Rood*, 1996, 1997; *Lin*, 2004). This dynamical core is perhaps one of the most well-know dynamical models today, and well-used within Geophysical Fluid Dynamics Laboratory (GFDL) model, NASA's Goddard Earth Observing System Model, Version 5 (GEOS-5) and NCAR's Community Atmosphere Model (CAM). This dynamic core used a staggered grid or the Arakawa C- and D-grid (*Arakawa and Lamb*, 1977). However, *Ullrich and Jablonowski* (2011) showed that when a wave propagates through a resolution discontinuity, the numerical algorithms that use staggered grids might cause more reflections than the numerical algorithms that use a non-staggered grid. Additionally, most of the current GCMs are based on equations that assume hydrostatic equilibrium. These models resolve processes whose horizontal scale is significantly

larger than the vertical scale. However, one reason for using non-uniform grids in GCMs is to resolve physical processes, in which the horizontal and vertical scales are similar, e.g., strong convections and cloud processes. The hydrostatic formulation does not correctly simulate the vertical motions, and instead, a non-hydrostatic formulation should be used in the refined grids. Thus, in order to build an algorithm that is suited for AMR techniques in the GCMs, we have developed a non-staggered grid finite-volume approach for solving both hydrostatic and non-hydrostatic Euler equations with either a Lagrangian vertical coordinate or Eulerian coordinates in *Chen et al.* (2012).

In this work, we present a technique that joins the hydrostatic and non-hydrostatic models in an accurate manner without reflection. In addition, the resolution at the interface between hydrostatic and non-hydrostatic models can be changed and the results are still accurate. Because the "floating" vertical Lagrangian coordinate does not allow material to penetrate between the Lagrangian layers, it eliminates the calculation of fluxes between the vertical control volume interfaces and reduces the 3-dimensional problem to a 2-dimensional problem, and thus increases the computational efficiency of the dynamic core. For this reason, we use a vertical Lagrangian coordinate in both the hydrostatic and non-hydrostatic models.

In order to validate our algorithm, we modified the gravity wave test, which was first introduced in *Skamarock and Klemp* (1994). The original tests are well suited to benchmark the ability of the dynamic models in representing the propagation of a gravity wave in a 2D (x-z) channel in both a hydrostatic or non-hydrostatic environment. The Coriolis force is set to zero for simplicity, which is different from the original test. We keep the domain size of the channel as in *Skamarock and Klemp* (1994) but divide it equally from the center into two blocks. Since the channel has a background horizontal eastward velocity of 20 m/s, it is easy to see that the west block is the upwind block and the east block is the downwind block. Different resolutions for

each block with hydrostatic or non-hydrostatic settings are assigned. We denote the results of simulations using different resolutions or hydrostatic and non-hydrostatic formulations between the upwind and downwind blocks as the "results of connected blocks". Since both hydrostatic and non-hydrostatic Euler equations are non-linear, we do not have analytical solutions to validate our results. Therefore, we estimate the reflection by comparing the difference between results of connected blocks and results of upwind blocks in pure hydrostatic or non-hydrostatic channels.

Our model might need to implement a vertically-implicit scheme to implement it in more realistic GCMs, in order to allow longer time steps. Nevertheless in this work, in order to introduce the least diffusion into our system, we use a vertically-explicit scheme in the non-hydrostatic equations. Both our hydrostatic and non-hydrostatic models introduce no explicit diffusion terms. It is more challenging to build a non-reflective two-way interface between blocks for different dynamical cores or resolutions if one does not introduce additional diffusion.

This chapter is organized as follows: in Section 3.2, we briefly review the governing equations to be used in our simulation. We also review the numerical implementation of the governing equations in Section 3.3. Section 3.4 describes the technique to be used in building the interface between blocks of different governing equations or different resolutions. In Section 3.5 the results of the model tests are discussed. Section 3.6 presents the conclusion.

3.2 Model equations

The 2-dimensional (x-z) system is used in our tests. The location of the discretized control volume is denoted using the index (i, k), where i is incremented from the left to right or west to east, and k is incremented in a top-down direction. We use η to denote any scalar variable or vector of scalar variables. The volume integration of η at control volume (i, k) is denoted by $\iint_{i,k} \eta \, dx \, dz$. The expressions: $\int_{i\pm0.5,k} \eta \, dz$

and $\int_{i,k\pm0.5}\eta\,dx$ are used to denote the area integration at the left/right surface and lower/upper surface of control volume (i,k).

3.2.1 Hydrostatic equations

The governing hydrostatic Euler equations in the integral form are:

$$\frac{\partial}{\partial t}\boldsymbol{U}_{i,k} = -\delta_i \boldsymbol{F} - \delta_k \boldsymbol{H} \tag{3.1}$$

where

$$\boldsymbol{U}_{i,k} = \begin{pmatrix} \iint_{i,k}\rho\,dx\,dz \\ \iint_{i,k}\rho u\,dx\,dz \\ \iint_{i,k}\rho\theta\,dx\,dz \end{pmatrix} \tag{3.2}$$

$$\delta_i\eta = \eta_{i+0.5,k} - \eta_{i-0.5,k} \tag{3.3}$$

$$\delta_k\eta = \eta_{i,k+0.5} - \eta_{i,k-0.5} \tag{3.4}$$

$$\boldsymbol{F}_{index} = \begin{pmatrix} \int_{index}\rho u\,dz \\ \int_{index}(\rho uu + p)\,dz \\ \int_{index}\rho\theta u\,dz \end{pmatrix} \tag{3.5}$$

$$\boldsymbol{H}_{index} = \begin{pmatrix} 0 \\ \int_{index} p\frac{\partial z}{\partial x}\,dx \\ 0 \end{pmatrix} \tag{3.6}$$

where ρ is the density, u is the horizontal velocity, p is the pressure and θ is the potential temperature. *index* could be replaced by $(i, k \pm 0.5)$ or $(i \pm 0.5, k)$.

We assume a constant model top pressure $p_{i,0.5} = p_{i,top}$, then the pressure at the vertical interfaces of each control volumes are:

$$p_{i,k+0.5} = p_{i,k-0.5} + \int_k \rho g \, dz \tag{3.7}$$

where $\int_k \eta \, dz = \int_{z_{i,k+0.5}}^{z_{i,k-0.5}} \eta \, dz$.

With the hydrostatic assumption, the control volume's vertical thickness $\delta_k z$ (note that k is indexed in top-down direction, $\delta_k z < 0$) is calculated by:

$$\delta_k z = -\frac{c_p \theta_{i,k}}{g p_0^\kappa} \delta_k p^\kappa \tag{3.8}$$

where c_p is the heat capacity at constant pressure, κ is the ratio of gas constant R_d and c_p. $p_0 = 10^5$ Pa is the reference surface temperature.

The vertical velocity is diagnostically derived by:

$$w_{i,k} = \frac{\partial}{\partial t} z_{i,k} + u_{i,k} \frac{\partial}{\partial x} z_{i,k} \tag{3.9}$$

The value of the hydrostatic pressure at the center of the control volume $p_{i,k}$ is calculated from:

$$p_{i,k} = \left(\kappa \frac{\delta_k p}{\delta_k p^\kappa} \right)^\gamma \tag{3.10}$$

where $\gamma = 1/(1 - \kappa)$.

3.2.2 Non-hydrostatic equations

The governing non-hydrostatic Euler equations in the integral form can also be described in the form of Eq. 3.1. However, vertical velocity is a prognostic variable in non-hydrostatic models. Thus one more term is needed in each of the vectors in Eq. 3.1:

$$\boldsymbol{U}_{i,k} = \begin{pmatrix} \iint_{i,k} \rho \, dx \, dz \\ \iint_{i,k} \rho u \, dx \, dz \\ \iint_{i,k} \rho w \, dx \, dz \\ \iint_{i,k} \rho\theta \, dx \, dz \end{pmatrix} \tag{3.11}$$

$$\boldsymbol{F}_{index} = \begin{pmatrix} \int_{index} \rho u \, dz \\ \int_{index} (\rho u u + p) \, dz \\ \int_{index} \rho w u \, dz \\ \int_{index} \rho\theta u \, dz \end{pmatrix} \tag{3.12}$$

$$\boldsymbol{H}_{index} = \begin{pmatrix} 0 \\ \int_{index} p \frac{\partial z}{\partial x} \, dx \\ \int_{index} p' \, dx \\ 0 \end{pmatrix} \tag{3.13}$$

where the pressure p is the full pressure, the hydrostatic pressure is denoted by p^*, and there difference between the full pressure and the hydrostatic pressure is $p' = p - p^*$. The hydrostatic pressure p^* at the control volume vertical interface and control volume

center are acquired from Eq. 3.7 and 3.10. The full pressure p is calculated by:

$$p = \left(\frac{R_d \rho \theta}{p_0^{\kappa}} \right)^{\gamma} \tag{3.14}$$

The location of the vertical interfaces between the control volumes in the non-hydrostatic simulations is also a prognostic variable. Eq. 3.9 is rearranged into the advection form of z:

$$\frac{\partial}{\partial t} z_{i,k\pm0.5} = -u_{i,k\pm0.5} \frac{\partial}{\partial x} z_{i,k\pm0.5} + w_{i,k\pm0.5} \tag{3.15}$$

3.3 Numerical solutions

The numerical discretization of the prognostic Eq. 3.1 is described in Chap. II. Here a brief description is presented. Take the hydrostatic formulation for example, if we denote $\int_i \eta \, dx = \int_{x_{i-0.5,k}}^{x_{i+0.5,k}} \eta \, dx$, and define:

$$\boldsymbol{R}_{i,k} = \begin{pmatrix} \int_{i,k} \rho \, dz \\ \int_{i,k} \rho u \, dz \\ \int_{i,k} \rho \theta \, dz \end{pmatrix} \tag{3.16}$$

then

$$\boldsymbol{U}_{i,k} = \begin{pmatrix} \iint_{i,k} \rho \, dx \, dz \\ \iint_{i,k} \rho u \, dx \, dz \\ \iint_{i,k} \rho \theta \, dx \, dz \end{pmatrix} = \int_i \boldsymbol{R} \, dx \tag{3.17}$$

The values within the control volume are reconstructed by interpolation schemes. We approximate the flux vector F , e.g., fluxes at the right interface of volume i, k,

by:

$$\boldsymbol{F}_{i+0.5,k} = u_{i+0.5,k}\boldsymbol{R}_{i+0.5,k} + \boldsymbol{P}_{i+0.5,k} \tag{3.18}$$

where $\boldsymbol{P}_{i+0.5,k}$ is $(0, \int_{i+0.5,k} p\, dz, 0)^T$ in hydrostatic equations, and is $(0, \int_{i+0.5,k} p\, dz, 0, 0)^T$ in non-hydrostatic equations. We denote the interpolated value of η at the control volume horizontal interfaces of the volume index (i, k) to be $\eta^W_{i,k}$ and $\eta^E_{i,k}$, where W and E denotes the values at west and east of control volume surfaces. $R_{i+0.5,k}$ is defined with upwind values:

$$\boldsymbol{R}_{i+0.5,k} = \begin{cases} \boldsymbol{R}^W_{i+1,k} & \text{if } u_{i+0.5,k} < 0 \\ \boldsymbol{R}^E_{i,k} & \text{if } u_{i+0.5,k} \geq 0 \end{cases} \tag{3.19}$$

and $u_{i+0.5,k}$ and $p_{i+0.5,k}$ are calculated using the Low Mach number Approximate Riemann Solver or LMARS, (Chap. II):

$$u_{i+0.5,k} = \frac{1}{2}\left(u^E_{i,k} + u^W_{i+1,k}\right) + \frac{1}{\left(\rho^E_{i,k} + \rho^W_{i+1,k}\right) c_{si+0.5,k}}\left(p^E_{i,k} - p^W_{i+1,k}\right) \tag{3.20}$$

$$p_{i+0.5,k} = \frac{1}{2}\left(p^E_{i,k} + p^W_{i+1,k}\right) + \frac{1}{4}\left(\rho^E_{i,k} + \rho^W_{i+1,k}\right) c_{si+0.5,k}\left(u^E_{i,k} - u^W_{i+1,k}\right) \tag{3.21}$$

note the second terms in the right hand side of Eq. 3.20 and 3.21 account for the implicit numerical diffusion, thus, the sound speed $c_{si+0.5,k}$ is not required to be calculated at every step. The diffusion terms can also be scaled using a parameter β, with $0 < \beta \leq 1$ (Eq. 53 to 56 in Chap. II), in order to decrease the implicit numerical diffusion. However, we do not recommend using this tuning technique because it might cause instability. The full derivation of Eq. 3.20 and 3.21 is described in Chap. II.

In the non-hydrostatic formulation, the vertical interface values of $w_{i,k+0.5}$ and

$p'_{i,k+0.5}$ are required in Eq. 3.13 and 3.15. We can also use LMARS to calculate these values:

$$w_{i,k+0.5} = \frac{1}{2}\left(w^B_{i,k} + w^T_{i,k+1}\right) - \frac{1}{\left(\rho^B_{i,k} + \rho^T_{i,k+1}\right) c_{si,k+0.5}}\left(p'^B_{i,k} - p'^T_{i,k+1}\right) \tag{3.22}$$

$$p'_{i,k+0.5} = \frac{1}{2}\left(p'^B_{i,k} + p'^T_{i,k+1}\right) - \frac{1}{4}\left(\rho^B_{i,k} + \rho^T_{i,k+1}\right) c_{si,k+0.5}\left(w^B_{i,k} - w^T_{i,k+1}\right) \tag{3.23}$$

where T and B denote top and bottom. Note that the vertical indices are in top-down direction, thus the second terms in the right hand side have a negative sign.

The values at the cell interfaces, e.g. horizontal interfaces $\eta^W_{i,k}$ and $\eta^E_{i,k}$, are calculated by:

$$\eta^W_{i,k} = -\frac{1}{20}\eta_{i-2,k} + \frac{9}{20}\eta_{i-1,k} + \frac{47}{60}\eta_{i,k} - \frac{13}{60}\eta_{i+1,k} + \frac{1}{30}\eta_{i+2,k} \tag{3.24}$$

$$\eta^W_{i,k} = -\frac{1}{20}\eta_{i+2,k} + \frac{9}{20}\eta_{i+1,k} + \frac{47}{60}\eta_{i,k} - \frac{13}{60}\eta_{i-1,k} + \frac{1}{30}\eta_{i-2,k} \tag{3.25}$$

In the horizontal direction, 4 ghost cells are appended at the boundaries of the calculation domain or blocks. However, no vertical ghost cell is used in our model. Thus the interpolation schemes near the vertical boundaries, e.g. top boundary, are one-sided:

$$\eta^T_{i,2} = \frac{1}{3}\eta_{i,1} + \frac{5}{6}\eta_{i,2} - \frac{1}{6}\eta_{i,3} \tag{3.26}$$

$$\eta^B_{i,2} = \frac{1}{3}\eta_{i,3} + \frac{5}{6}\eta_{i,2} - \frac{1}{6}\eta_{i,1} \tag{3.27}$$

$$\eta_{i,1}^{T} = \frac{3}{2}\eta_{i,1} - \frac{1}{2}\eta_{i,2} \tag{3.28}$$

$$\eta_{i,1}^{B} = \frac{1}{2}\eta_{i,1} + \frac{1}{2}\eta_{i,2} \tag{3.29}$$

The bottom boundary can be treated analogously by reversing the index numbers.

The standard 4-step Runge-Kutta method (*Durran*, 2010) is used as the time marching scheme. The floating vertical Lagrangian coordinate requires periodical remapping. The vertical remap scheme and vertical reflective boundary conditions are also described in Chap. II. With the hydrostatic formulation, the top boundary is always a free surface.

3.4 Building the interface that connects the hydrostatic and non-hydrostatic blocks

The control volume is the basic element in the finite volume scheme. Since we use the A-grid, each control volume simply carries the volume mean values of each physical variable. Additionally, because the floating vertical coordinate is not a constant, the value of the coordinate at the top surface of each control volume is also recorded. In summary, one control volume at index (i, k) records independent physical variables: volume mean values of the density $\rho_{i,k}$, horizontal velocity $u_{i,k}$, the vertical velocity $w_{i,k}$, potential temperature $\theta_{i,k}$ and the scalar value of the vertical coordinate $z_{i,k-0.5}$. The topography and domain top pressure are considered as block-wise constants. We denote these variables primary variables and constants. Other variables such as pressure can be derived from these primary variables and constants.

3.4.1 Ghost cell communication between blocks of uniform grid spacing

The computational domain is divided into blocks. In the horizontal direction, each block uses 4 ghost cells to extend the west and east boundaries. If all blocks are of uniform horizontal resolution, the ghost cells are updated using the 4 inside columns of nearest-to-the-interface cells of the neighbor block. Note that, each control volume is defined the same way regardless of whether it is in the hydrostatic block or in the non-hydrostatic block. In other words, the definition of control volume is independent of the choice of the dynamic cores. Thus, there is no conflict to copy values from a control volume in the hydrostatic block into the control volumes of the non-hydrostatic block as a ghost cell or vise versa. Thus, if the neighbor blocks are of the same horizontal grid spacing, no special treatment is required when communicating the boundary ghost cells between the two blocks. The interface between blocks of different governing dynamic formulations is a full two-way interface.

3.4.2 Ghost cell communication between blocks of different grid spacing

When communicating ghost cells between blocks with different horizontal grid size, refinement and coarsening of the control volumes are required. Here we only discuss the most basic case. The ratio of the horizontal grid size between the coarse and refined blocks is always 2. Only one-dimensional (x) grid refinement and coarsening is involved in our tests.

The flux form independent physical variables are described by the vector $\boldsymbol{U}_{i,k}$ in Eq. 3.11. $z_{i,k-0.5}$ is recorded as a point value at the top of the control volume. Assume two blocks are connected, the left block has n horizontal cells, and the ghost cells are labeled from $n+1$ to $n+4$. These four columns of ghost cells are to be constructed using the innermost several columns of cells in the right block. More general cases are analogous to this process, e.g. simply applied by reversing the indices.

If the right block is of finer grid resolution, then a coarsening process is needed.

Since the ratio of the horizontal grid size between the blocks is 2, every two columns of ghost cells in the left block are perfectly aligned to one column of cells in the inner side of the right block. We denote the target ghost cells to be $\boldsymbol{U}^L_{n+1,k}$ to $\boldsymbol{U}^L_{n+4,k}$ and $z^L_{n+1,k-0.5}$ to $z^L_{n+4,k-0.5}$, the preprocessed cells in the right block to be $\boldsymbol{U}^R_{i,k}$ and $z^R_{i,k-0.5}$. The cells are aligned: $x^L_{n+0.5+2i,k} = x^R_{i+0.5,k}$. The flux form physical variables are built from:

$$\boldsymbol{U}^L_{n+i,k} = \boldsymbol{U}^R_{2i-1,k} + \boldsymbol{U}^R_{2i,k} \tag{3.30}$$

where $i = 1$ to 4. The coarsening of vertical coordinate is simply using:

$$z^L_{n+i,k-0.5} = \frac{1}{2}(z^R_{2i-1,k-0.5} + z^R_{2i,k-0.5}) \tag{3.31}$$

If the right block is of coarser grid resolution, then a refinement process is needed. The Newton interpolation scheme (*Yang*, 2001) is used for the reconstruction of the horizontal profiles of each flux form physical variable U or vertical location z. The reconstruction of the horizontal profile of U is performed using the following procedure: 1) define $\boldsymbol{A}_{-0.5,k} = 0$, $\boldsymbol{A}_{i-0.5,k} = \boldsymbol{A}_{i-1.5,k} + \boldsymbol{U}^R_{i,k}$, $i = 1$ to 4; 2) we perform the Newton interpolation with the points $(-0.5, \boldsymbol{A}_{-0.5,k})$, $(0.5, \boldsymbol{A}_{0.5,k})$, $(1.5, \boldsymbol{A}_{1.5,k})$, and $(2.5, \boldsymbol{A}_{2.5,k})$ to reconstruct the profile of $\boldsymbol{A}_k(x)$; 3) build the refined cells as:

$$\boldsymbol{U}^L_{n+i,k} = \boldsymbol{A}_k(\frac{i+1}{2}) - \boldsymbol{A}_k(\frac{i}{2}) \tag{3.32}$$

For the refinement of z, simply get the profile of $z_{k-0.5}(x)$ by polynomial interpolation using $(0, z^R_{0,k-0.5})$, $(1, z^R_{1,k-0.5})$, $(2, z^R_{2,k-0.5})$ and $(3, z^R_{3,k-0.5})$. The values of z of the refined cells are:

$$z^L_{n+i,k-0.5} = z_{k-0.5}(\frac{i-0.5}{2}) \tag{3.33}$$

Chap. II demonstrated that the advection scheme within the block is 4th order accurate. The coarsening of the flux form variables $\boldsymbol{U}$ does not involve any interpolation, thus will not affect the accuracy of the numerical scheme. The coarsening of the vertical coordinate is 2nd order accurate. This treatment is for simplicity, because z is not involved in the flux calculation, and the diffusion introduced by this low order algorithm is very limited. The refinement algorithm is 4th order accurate. Thus, the fluxes calculated near the block interfaces do not introduce diffusion caused by low order numerical schemes.

3.5 Simulations and results

3.5.1 Vertical velocities and the sensitivity test

We have conducted the gravity wave test formulated by *Skamarock and Klemp* (1994) with hydrostatic and non-hydrostatic formulations in Chap. II. The vertical domain size is 10 km and the horizontal domain size is 300 km. The surface pressure is 10^5 Pa. The background atmosphere has a constant Brunt-Väisälä frequency of 10^{-2} s^{-1} with a surface temperature of 300 K. The initial horizontal wind is 20 ms^{-1} and a periodic horizontal boundary condition is used. For simplicity, the Coriolis force is set to zero. The waves are excited by an initial potential temperature perturbation of the form:

$$\theta' = \Delta\theta_0 \frac{\sin\frac{\pi z}{H}}{1 + (x - x_c)^2/a^2} \tag{3.34}$$

where $\Delta\theta_0 = 10^{-2}$ K, $H = 10$ km, $a = 5$ km, and $x_c = 100$ km.

The contour of the potential temperature perturbation is usually used as a benchmark for non-hydrostatic simulations. However, since our purpose is to apply a non-hydrostatic formulation within a hydrostatic model using non-uniform grids, and it is crucial to correctly resolve the vertical motions, we plotted the contours of the vertical

velocity in Fig. 3.1. Vertical velocity is more closely connected to some atmospheric processes, such as cloud formation. (i.e. in a humid atmosphere, a rising motion will transport the air parcel into colder environment, where water vapor condensation is more likely to take place. Thus, cloud is more likely to form with the rising motion.) Similarly, sinking motion is more likely to suppress cloud formation.

Rows a, b and c in Fig. 3.1 are results from the non-hydrostatic formulation. The numerical scheme using the vertical Lagrangian coordinate is slightly more diffusive than the scheme using Eulerian coordinate. According to this figure, we can see the rising-sinking vertical motions appear periodically in the non-hydrostatic scheme, and the wavelength is around 20 km to 30 km. However, the hydrostatic scheme cannot permit such wave trains even at this high resolution of $dx = dz = 1$ km. According to the different vertical velocity patterns between hydrostatic and non-hydrostatic models, the periodic rising-sinking train in the non-hydrostatic formulation is more likely to generate cloud formation with a periodical pattern between the two gravity wave fronts. The hydrostatic atmosphere would have a different response to the gravity waves propagation.

Comparing the non-hydrostatic results in Rows a, b and c from Fig. 3.1, the differences between each plot are small. The remap frequency of every 60 s used in Lagrangian simulation in Row b provides almost identical results compared to its Eulerian counterpart. Since our purpose is to connect the hydrostatic and non-hydrostatic regimes using a vertical Lagrangian coordinate, the rest of our tests are using the vertical Lagrangian coordinate, vertical grid spacing $dz = 1$ km, and a remap frequency of 60 s by default. All results are taken at $t = 3000$ s and the time step is $\Delta t = 1$ s. The contour step of the potential temperature perturbation is 0.0005 K, and the contour step of the vertical velocity is 0.0005 m/s.

In many GCMs, the horizontal grid spacing is significantly larger than the vertical grid spacing. Although the vertical grid spacing is already sufficient for simulating

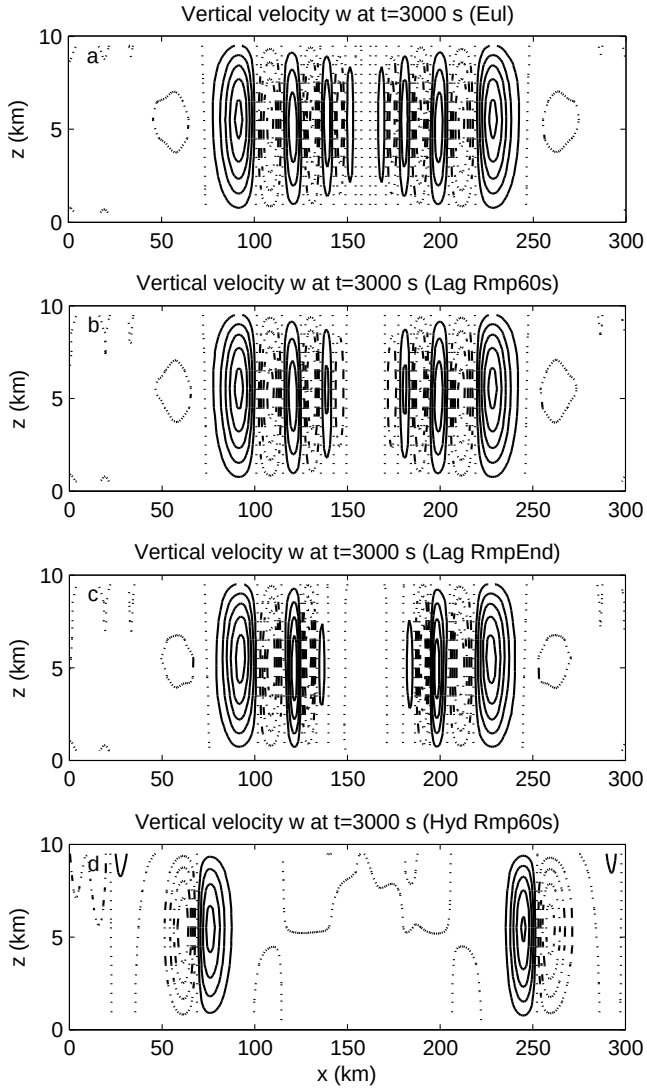

Figure 3.1: The vertical velocity contours of the gravity wave test at $t = 3000$ s. The configurations are a) nonhydrostatic with Eulerian coordinates. b) nonhydrostatic with vertical Lagrangian coordinate and a remap frequency of 60 s. c) same as b), but only a single remap at the end of the simulation. d) hydrostatic with vertical Lagrangian coordinates with 60 s remap frequency. The contour interval is 0.0005 m/s, the dotted lines are the zero contours, the solid lines are positive, dash-dotted lines are negative. The grid spacing is $dx = dz = 1$ km.

some non-hydrostatic processes, these processes might not be correctly resolved even with the non-hydrostatic formulations since the horizontal grid spacing is too large. The impact of changing the ratio of the horizontal-vertical grid size on the validity of the hydrostatic assumption is investigated in Fig. 3.2 and 3.3. The next two figures show the vertical velocity using hydrostatic and non-hydrostatic formulations under different ratio of horizontal to vertical grid spacing, i.e. $dx/dz = 2, 3, 5$.

In the hydrostatic plots of Fig. 3.2, since non-hydrostatic patterns are not resolved, the wave patterns of the vertical velocity are not sensitive to the ratio of horizontal to vertical grid spacing. The intensity of the wave amplitude decreases as the horizontal grid spacing increases. This is because that larger grid spacing results to larger numerical diffusivity. However, in the non-hydrostatic plots of Fig. 3.3, when $dx/dz = 2$, the non-hydrostatic rising-sinking wave train is resolved in the non-hydrostatic scheme, but the intensity is decreased compared with the result of $dx = dz$. When $dx/dz = 3$, the wave train pattern becomes much less clear in the non-hydrostatic model. When $dx/dz = 5$, the non-hydrostatic wave patterns are almost indistinguishable compared to the results using the hydrostatic formulation.

According to the sensitivity test by changing the ratio of the horizontal-vertical grid spacing, resolving the non-hydrostatic wave train pattern requires high horizontal resolution, e.g., $dx = 1$ or 2 km. In order to properly resolve the small-scaled non-hydrostatic motions, the wavelength of the middle wave train should be as large as at least 10 grid cells. As mentioned in the introduction, a typical cumulonimbus cloud has the horizontal scale of 1 km. However, it is very expensive to keep the global horizontal resolution smaller than a few kilometers in GCMs. Thus, we want to develop a model that allows waves to be transported from hydrostatic regimes to non-hydrostatic regimes. The hydrostatic regimes could have larger horizontal grid spacing than the non-hydrostatic regimes.

In the following simulations, the *Skamarock and Klemp* (1994) gravity wave test

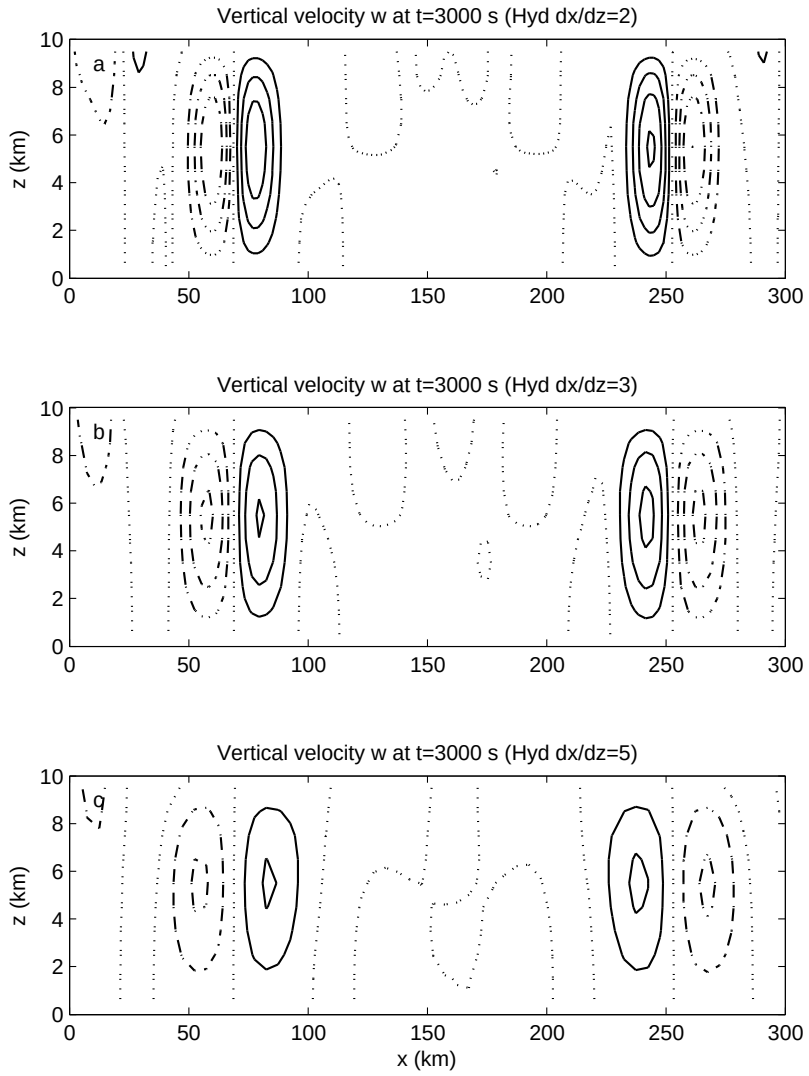

Figure 3.2: The vertical velocity contours of the gravity wave test at $t = 3000$ s with the hydrostatic formulations. The contour setups are identical with Figure 3.1. $dz = 1$ km. a) ratio of the horizontal and vertical resolution $dx/dz = 2$. b) $dx/dz = 3$. c) $dx/dz = 5$.

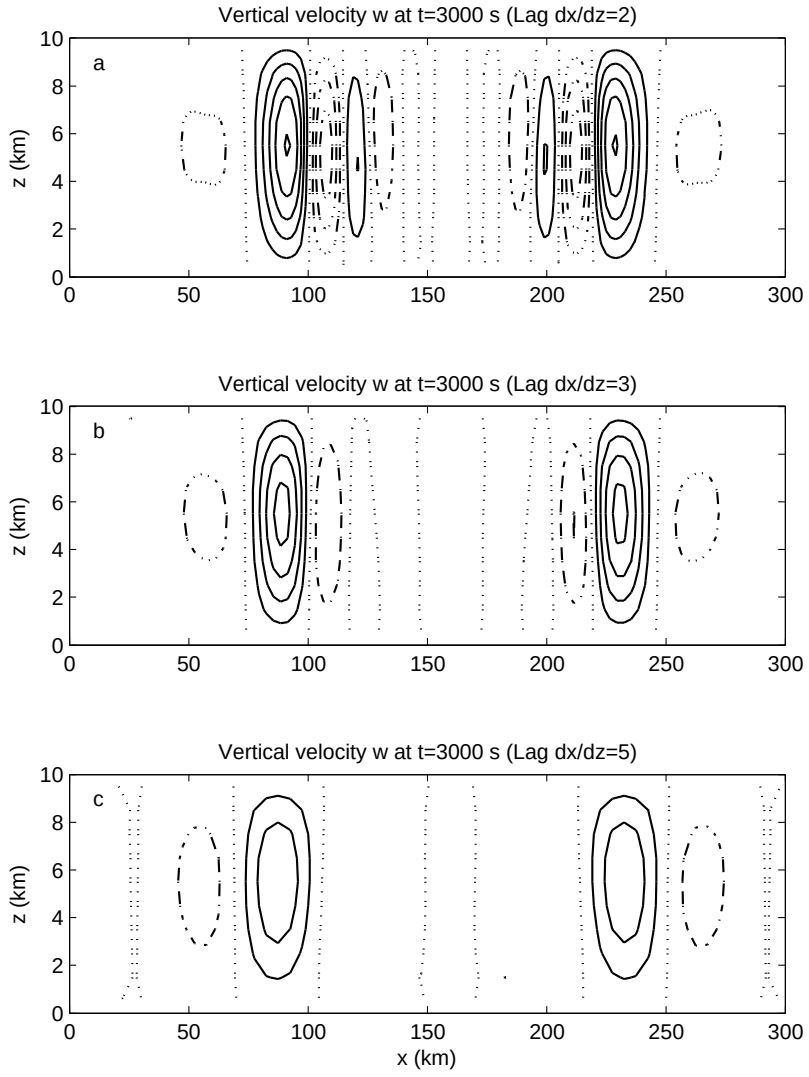

Figure 3.3: Same as figure 3.2, but the non-hydrostatic formulations are used.

is modified for connecting the hydrostatic regimes with non-hydrostatic regimes. We display our results in three steps. The first step is to test the gravity wave propagating between blocks of different hydrostatic and non-hydrostatic formulations without changing the grid spacing. In these simulations, only uniform grids are involved, thus no refinement or coarsening is needed at the interfaces of the different blocks. The second step is to connect blocks using the same governing dynamic formulations with non-uniform grid spacing. And in the last step, these two representations are integrated. Blocks of different governing dynamic formulations are connected and non-uniform grids are applied.

3.5.2 The propagation of gravity waves between blocks of different governing equations

No control volume refinement or coarsening is applied on the block interface. Each block just copies four outermost-to-the-interface columns of control volumes as the ghost cells, and sends the ghost cells to its neighbor. No other special treatment is required in this process.

3.5.2.1 The propagation of gravity waves from the hydrostatic block to the non-hydrostatic block

In Fig. 3.4, the channel is equally divided into two blocks. The initialization of the gravity wave is exactly the same as the one block version. However, hydrostatic formulations are used in the left upwind block, while the non-hydrostatic formulations are used in the right downwind block. The contours are snapshots of the potential temperature perturbation and vertical velocity at the end of the simulation $t = 3000$ s. The vertical dashed line at $x = 150$ km indicates the interface between the two blocks.

The results from Fig. 3.4 indicate that the waves initialized in the left block are

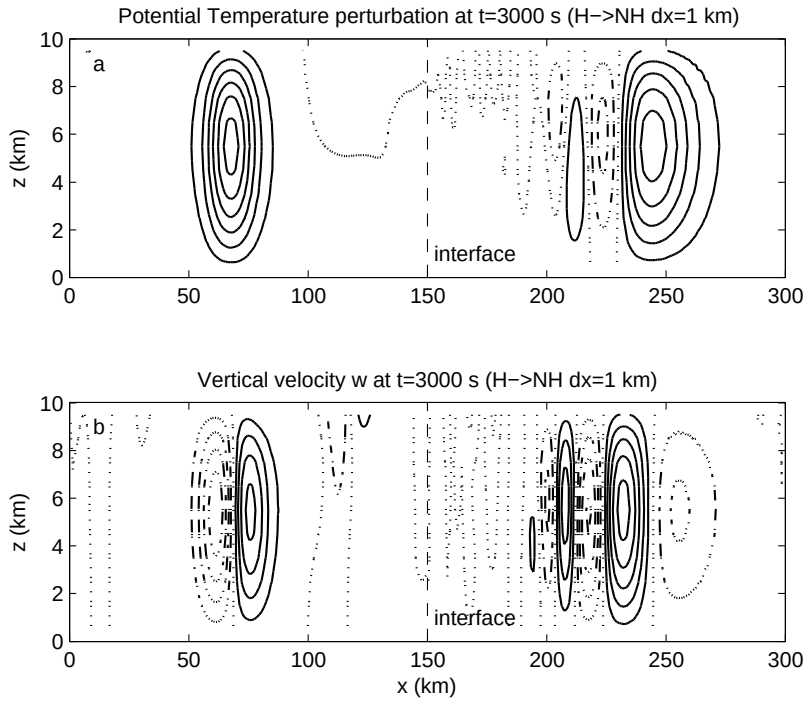

Figure 3.4: Wave transported from the hydrostatic block to the non-hydrostatic block. a) Contours of potential temperature perturbation. The contour interval is 0.0005 K. b) Contours of the vertical velocity. The contour interval is 0.0005 m/s. The vertical dashed line at $x = 150$ km indicates the block interface. In both subplots, the dotted lines are the zero contours, the solid lines are positive, dash-dotted lines are negative. The grid spacing is $dx = dz = 1$ km.

transported smoothly to the right block. No obvious reflection is observed. The zero contours are naturally connected between the interfaces in the results of both potential temperature perturbation and vertical velocity. Both left (upstream) and right (downstream) blocks present certain similarities to the pure hydrostatic or non-hydrostatic counterparts in Fig 2.4d, 2.4b and Fig. 3.1d, 3.1b. Note that even the gravity wave generated in the upstream hydrostatic block does not produce mid wave-trains, a wave train is produced in the downwind non-hydrostatic block when the gravity wave front is transported into this regime.

3.5.2.2 The interface quality control

If we compare the hydrostatic block of Fig. 3.4 to the left panel of the result of the pure hydrostatic simulation in 2.4d and Fig. 3.1d, the differences are the combination of numerical reflections and the impact from the downwind regime. The numerical reflection is highly undesired and should be eliminated as much as possible. In order to investigate the amount of the numerical reflection, we sampled the potential temperature and the vertical velocity at the height of $z = 4.5$ km at the horizontal range from 0 to 150 km in the pure hydrostatic channel and the channel that connects the hydrostatic and the non-hydrostatic blocks in Fig. 3.5.

If we assume the impact from the downwind non-hydrostatic regime is not coupled with the numerical reflection, the differences of the sampled values in Fig. 3.5 are the linear combination of these two factors. The result of the vertical velocity shows that the main wave feature of the single rising-sinking vertical velocity pattern, which is located from 50 to 90 km, is not affected whether the downwind block is hydrostatic or non-hydrostatic, because Fig. 3.5b indicates almost zero difference at this area. However, some weak motions located from 10 to 30 km and from 110 to 130 km might come from the downwind non-hydrostatic regime. The sampled values of potential temperature indicate very good agreement between the pure hydrostatic channel and

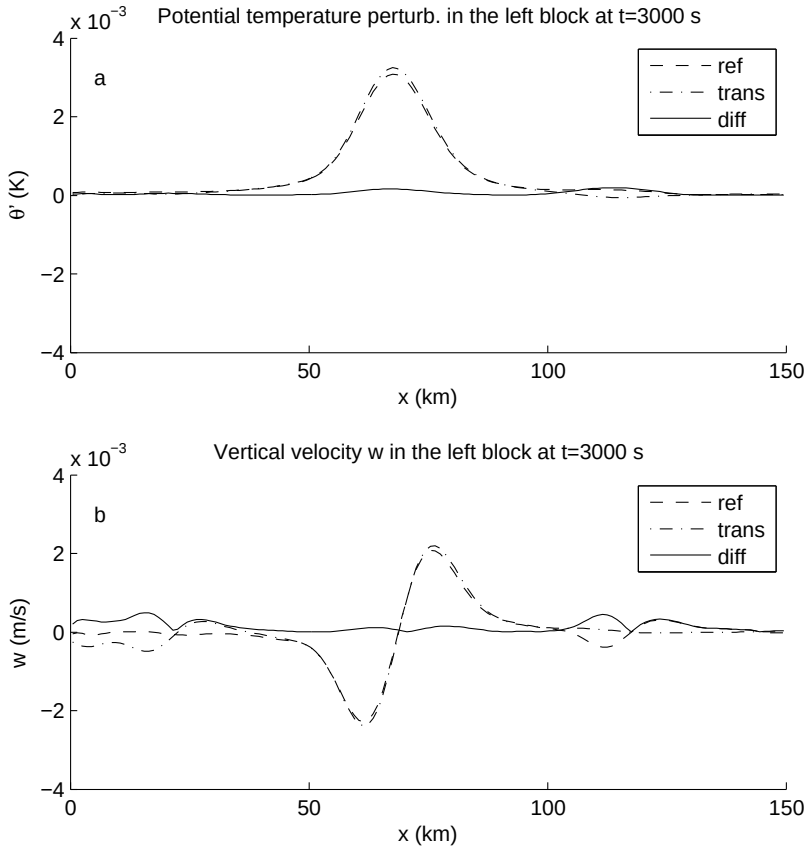

Figure 3.5: a) The potential temperature perturbation sampled at $z = 4.5$ km in range 0 to 150 km. b) The vertical velocity sampled at $z = 4.5$ km in range 0 to 150 km. In the legends: ref indicates the values in the pure hydrostatic channel; $trans$ indicates the values in the channel that connects hydrostatic and non-hydrostatic block; $diff$ indicates the absolute difference between the two sampled values mentioned above

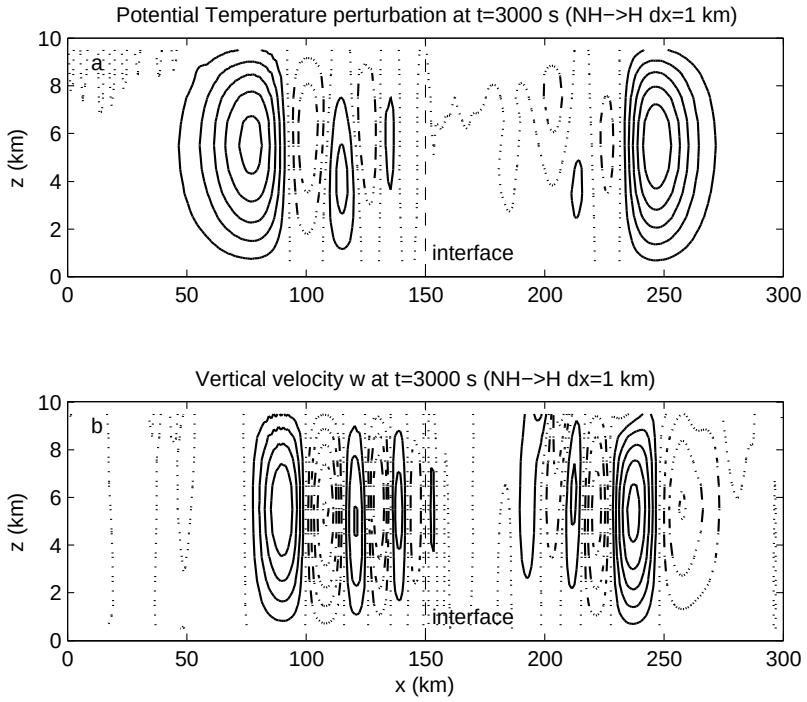

Figure 3.6: Same with figure 3.4, except the left block is non-hydrostatic and the right block is hydrostatic.

the channel connected different blocks, since the difference is almost zero. No spurious waves are generated from either numerical reflection or downwind block impact.

3.5.2.3 The propagation of gravity waves from the hydrostatic block to the non-hydrostatic block

Simply reversing the two blocks in Fig. 3.4 can do the similar wave transportation from non-hydrostatic block to the hydrostatic block. The results are presented in Fig. 3.6. The non-hydrostatic formulations are applied in the left upwind block, while the hydrostatic formulations are applied the right downwind block.

Comparing Fig. 3.6 with Fig. 3.1b and Fig 2.4b, significant similarities are observed among the upwind non-hydrostatic blocks, whether or not the downwind block is hydrostatic or non-hydrostatic. However, the downwind hydrostatic block of Fig. 3.6 is quite different from Fig. 3.1d or Fig 2.4d. If a wave train of the rising-

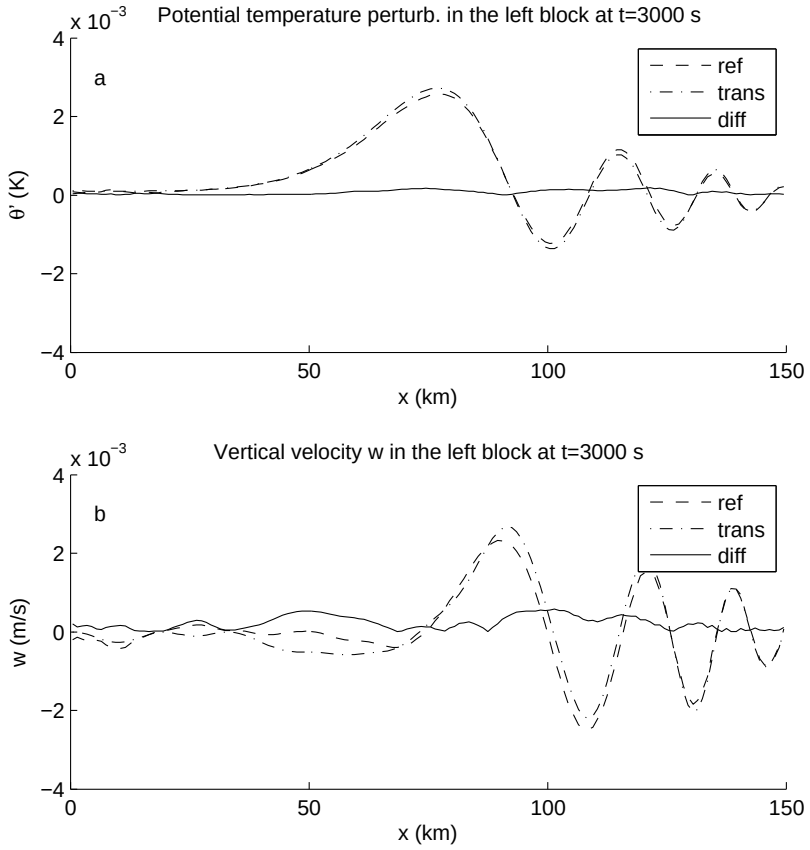

Figure 3.7: Same with figure 3.5, except the upwind blocks being sampled are non-hydrostatic.

sinking pattern is formed in the upwind non-hydrostatic block, it will be preserved when it is transported into the downwind hydrostatic block. However, this wave train would not develop in the hydrostatic block, according to the intensity of the wave is much less than the left block.

3.5.2.4 The interface quality control

Similar to Fig. 3.5, we sampled the vertical velocity and the potential temperature perturbation in the left non-hydrostatic block in Fig. 3.7.

Fig. 3.7b shows unstructured difference of the vertical velocity, which indicates the downwind hydrostatic block has some nonlinear impacts to the upwind non-

hydrostatic block. These downwind impacts have little effect to the potential temperature field, because the values of the differences in Fig. 3.7a are almost zero. Note that both the potential temperature perturbation and the vertical velocity near the interface are non-zero, however, almost zero differences are observed near the interface.

It is not easy to isolate the numerical reflection and the impact from the downwind block from the difference between the pure hydrostatic or non-hydrostatic channel and the channel that connects different blocks. The bottom line is that the numerical reflection would not be larger than the differences presented by Fig. 3.5 and 3.7. No wave structures that might destroy the stability are observed in all plots.

3.5.2.5 Summary to this part

The terminology "upwind" and "downwind" is used because of the 20 m/s background horizontal velocity and the initial potential temperature perturbation is always formed on the left panel of the channel. The interface applied between the two blocks is fully two-way. No filter is applied, thus all waves that reach the interface are transported to the neighbor block. However, the governing dynamic formulations are fully compressible, thus acoustic wave is permitted. In the hydrostatic regime, the acoustic wave is not permitted in vertical direction. Due to these different physical properties between different regimes, the incoming information develops differently in the downwind hydrostatic and non-hydrostatic block. We observed that the single rising-sinking vertical velocity developed in the hydrostatic left block triggered multiple rising-sinking wave trains in the non-hydrostatic right block. The wave trains developed in the non-hydrostatic upwind block would be preserved when they were transported into the hydrostatic block.

Our model uses no diffusive treatment at the interface between the hydrostatic regime and the non-hydrostatic regimes. We connect these two regimes by simply

exchange the ghost cells at the interface. Thus the numerical diffusion is kept to a minimum. At the same time, no obvious reflection that might harm the stability or quality of the results is observed. Since the interface ghost cell exchange is independent of whether the block is hydrostatic or non-hydrostatic, each of the blocks "blindly" prepares the ghost cells for its neighbors and "blindly" receives ghost cells from its neighbors. This property is also valid when refinement or coarsening of control volume is performed, and will be discussed in the following subsections.

3.5.3 The propagation of gravity waves between regimes of different horizontal grid spacing

In order to isolate the impact from grid refinement or coarsening at the block interface from the dynamic core change between the blocks, the tests of the gravity wave propagation between same governing equations but with different horizontal grid spacing are performed.

As described in section 3.4, the refinement algorithm is 4th order accurate, which is consistent with the order of accuracy of the hydrostatic or non-hydrostatic dynamic schemes. The consistent order of accuracy prevents wave filtering at the block interface, and thus the amplitude of the wave should be retained when it is transported between the blocks.

Fig. 3.8 shows how the gravity wave travels in the pure hydrostatic channel. The full domain is also equally divided at $x = 150$ km. The horizontal resolution in the left block is $dx = 2$ km, while in the right block is $dx = 1$ km and the vertical grid spacing is 1 km in both blocks. In row a, the contours of the potential temperature perturbation are plotted. In row b, the contours of the vertical velocity are plotted. Fig. 3.9 is the same as Fig. 3.8, except both left and right blocks are non-hydrostatic. The results presented in Fig. 3.8 and 3.9 indicate very smooth connection between the blocks. Both the contours of zero potential temperature perturbation and vertical

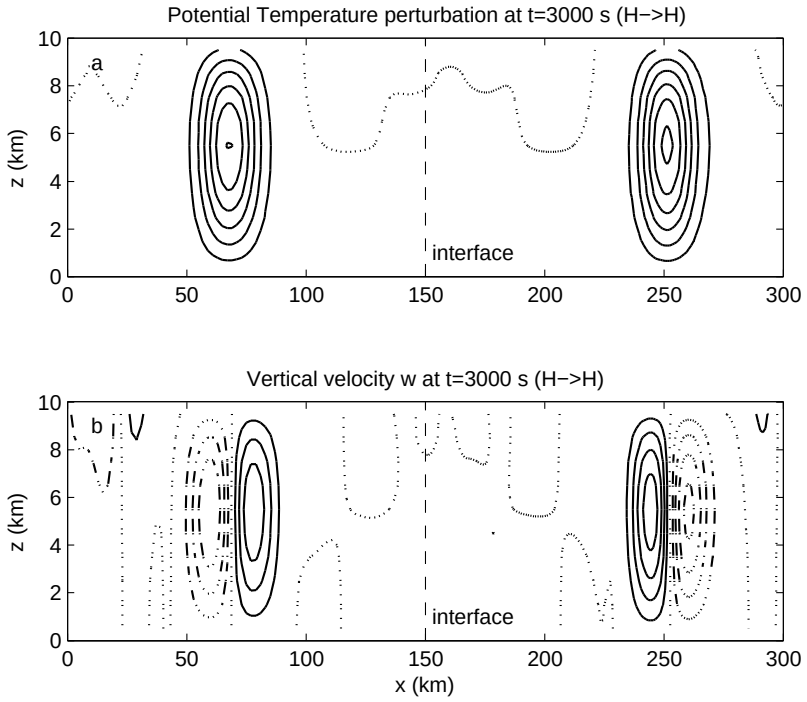

Figure 3.8: Wave transported in the hydrostatic channel from coarse upwind block of horizontal grid spacing $dx = 2$ km to refined downwind block of horizontal grid spacing $dx = 1$ km at time $t = 3000$ s. a) Contours of potential temperature perturbation. b) Contours of the vertical velocity.

velocity are closely connected at the interfaces.

Because the refined right blocks have less numerical diffusivity than the left blocks, the amplitudes of the waves are all stronger in the right blocks. This property indicates that the block interfaces do not have strong damping effect when the gravity waves were transported through blocks of different horizontal resolutions.

3.5.3.1 The interface quality control

We performed the same quality control procedure as we have done in Fig. 3.5 and 3.7 in order to monitor the numerical reflection at the interfaces. The results are shown in Fig. 3.10 and 3.11. Note that no dynamic core changes were involved in the tests, the downwind blocks should produce much less impact to the upwind

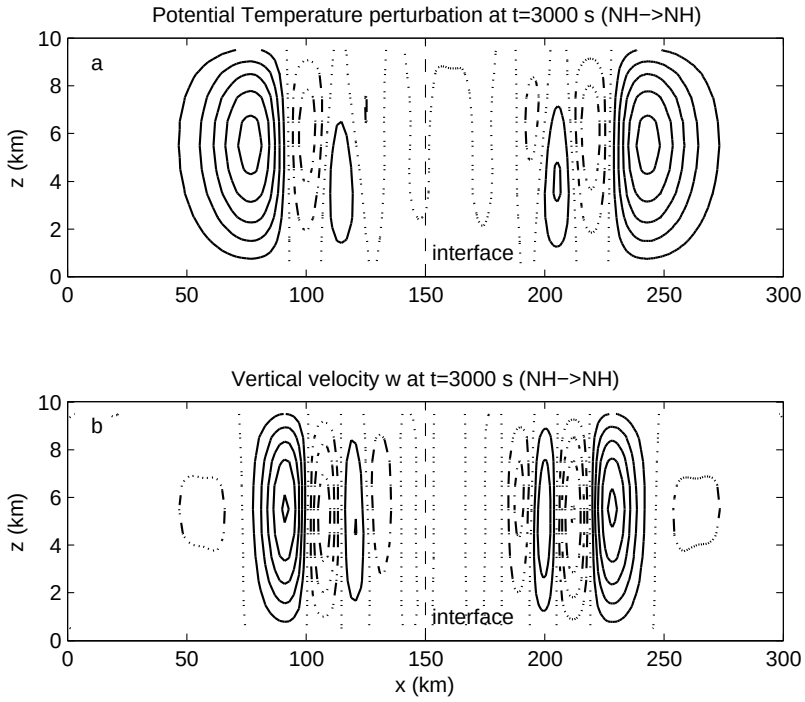

Figure 3.9: Same as figure 3.8, except both left and right blocks are non-hydrostatic.

blocks comparing to the tests in previous section. The results from Fig. 3.10 and 3.11 indicate very little numerical reflection since the differences of both the potential temperature perturbation and the vertical velocities between the channel of uniform grid and non-uniform grid in all figures are almost zero.

3.5.4 Connecting blocks of different governing dynamic formulations and different grid spacing

As we discussed above, each of the blocks do not requires information of the neighbor block types to prepare or receive the ghost cells. No additional treatment to the refinement and coarsening process is required. This property is desirable in the large parallel system since the communication between each of computing threads is minimum. Each of the blocks performs control volume refinement or coarsening at the interfaces just as described in section 3.4, then send these ghost cells to its

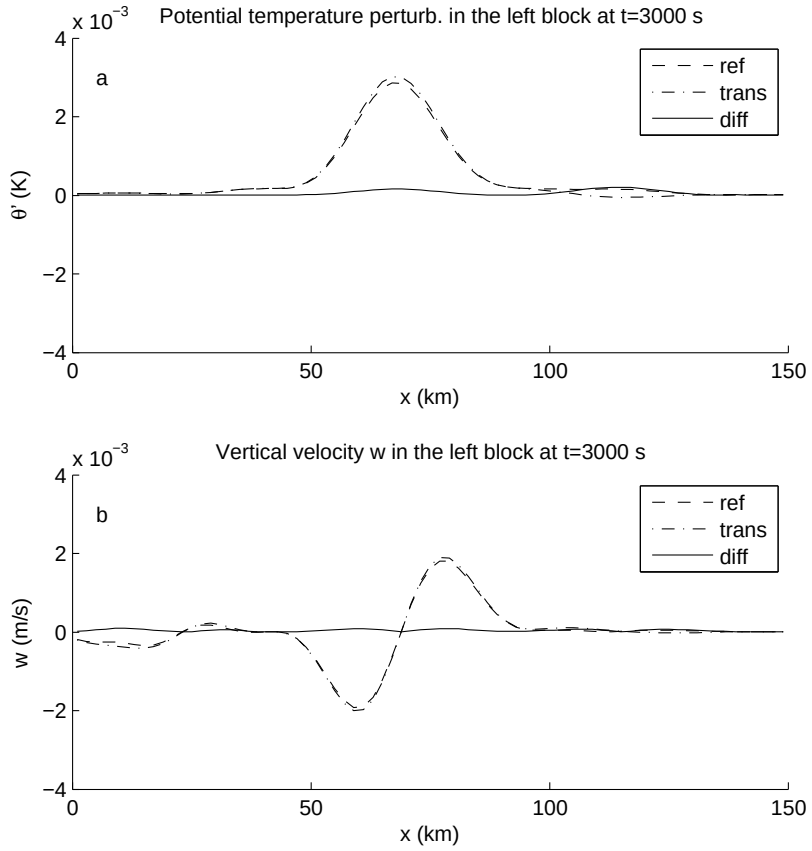

Figure 3.10: a) The potential temperature perturbation sampled at $z = 4.5$ km in range 0 to 150 km. b) The vertical velocity sampled at $z = 4.5$ km in range 0 to 150 km. In the legends: ref indicates the values in the hydrostatic channel of uniform grids; $trans$ indicates the values in the channel that connects blocks from coarse grid to refined grid; $diff$ indicates the absolute difference between the two sampled values mentioned above.

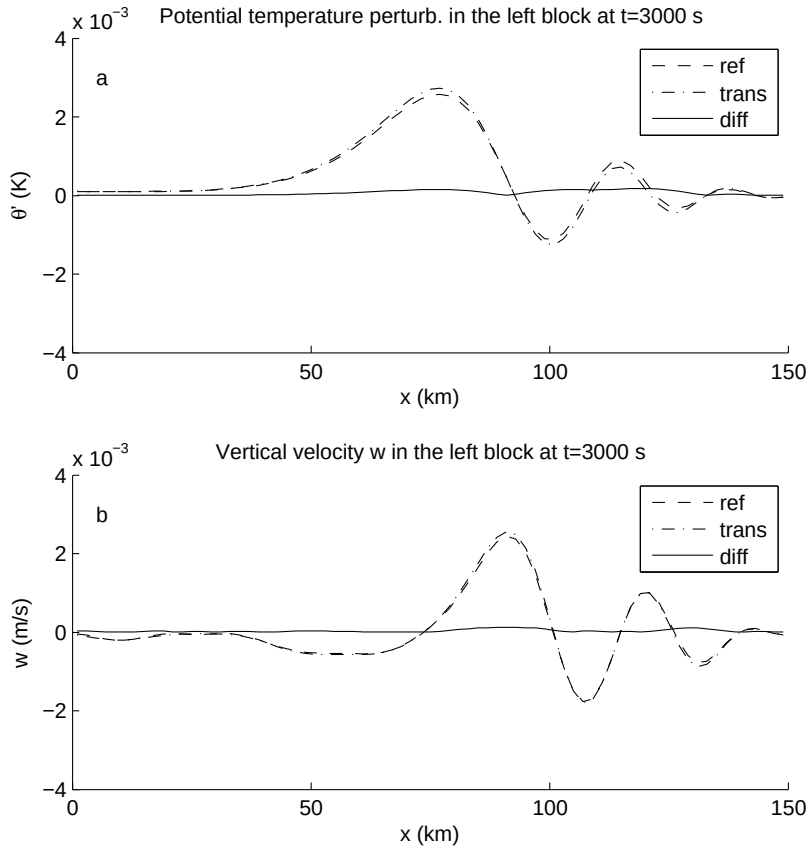

Figure 3.11: Same as figure 3.10, except that the non-hydrostatic formulations are used in the simulations.

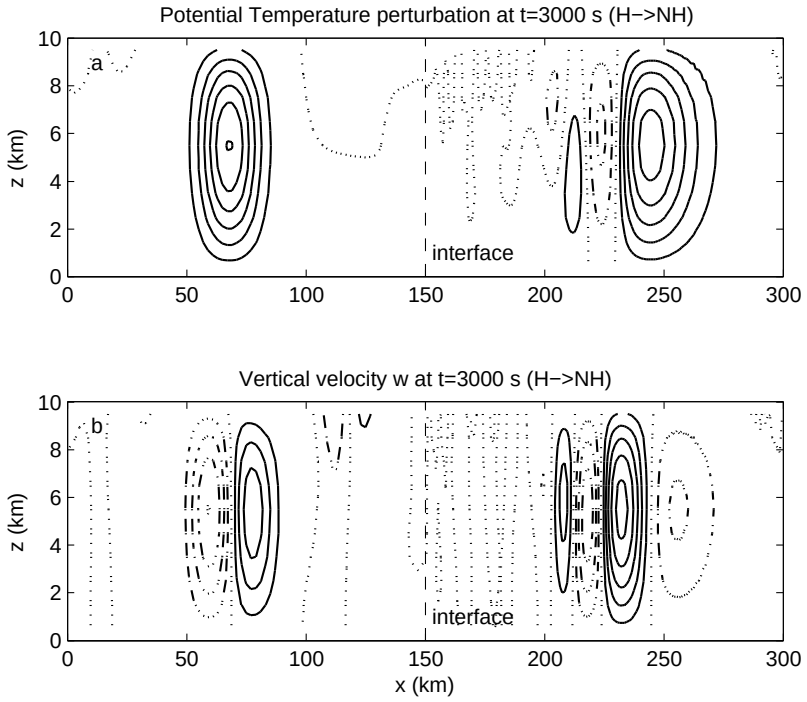

Figure 3.12: The gravity waves travel from hydrostatic block on the left to the non-hydrostatic block on the right at $t = 3000$ s. The horizontal grid spacing in the hydrostatic block on the left is $dx = 2$ km, in the non-hydrostatic block on the right is $dx = 1$ km. a) Contours of the potential temperature perturbation. b) Contours of the vertical velocity.

neighbor and receive the ghost cells prepared by its neighbors. Fig. 3.12 shows results of gravity waves travels from hydrostatic block to the non-hydrostatic block. Fig. 3.13 shows results of gravity waves travels from non-hydrostatic block to the hydrostatic block. In these two figures, the horizontal grid spacing of the hydrostatic block is $dx = 2$ km, and the horizontal grid spacing of the non-hydrostatic block is $dx = 1$ km. The vertical grid spacing of both blocks is $dz = 1$ km.

Similar to previous results, the zero contours of both the potential temperature perturbation and the vertical velocity are closely connected at the block interfaces. The 4th order refinement and coarsening algorithms created no distorted contour at the interfaces even both governing dynamic equations and resolution of the two blocks are different. Fig. 3.12 and 3.13 shows significant similarities to Fig. 3.4 and

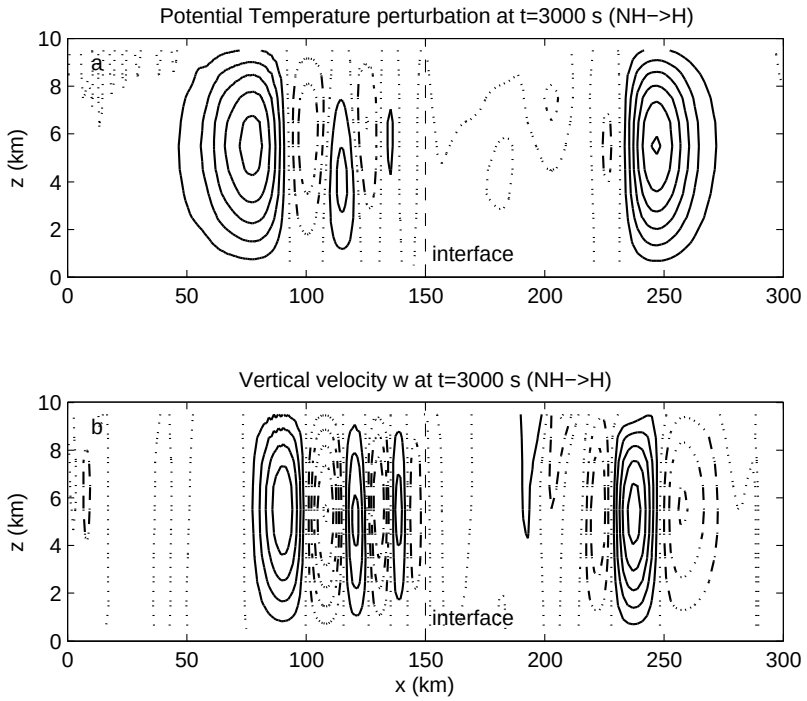

Figure 3.13: The gravity waves travel from non-hydrostatic block on the left to the hydrostatic block on the right at $t = 3000$ s. The horizontal grid spacing in the non-hydrostatic block on the left is $dx = 1$ km, in the hydrostatic block on the right is $dx = 2$ km. a) Contours of the potential temperature perturbation. b) Contours of the vertical velocity.

3.6. The only difference is that the intensity of the waves in Fig. 3.12 and 3.13 is slightly smaller. This is because the coarser hydrostatic block has more numerical diffusion.

3.5.4.1 The interface quality control

We performed the same quality control procedure as we have done in Fig. 3.5, 3.7 and 3.10, 3.11 in order to monitor the numerical reflection at the interfaces. The results are illustrated in Fig. 3.14 and 3.15.

The results plotted in Fig. 3.14 and 3.15 are similar to results in Fig. 3.5 and 3.7. Note that the grid spacing in Fig. 3.14 is twice as large as the grid spacing in Fig. 3.5, the sampled values of the amplitude of the waves in Fig. 3.14 are slightly weaker than the ones from Fig. 3.5. However the wave patterns are similar between the results from different horizontal resolutions. The resolution of Fig. 3.15 is the same with the resolution of Fig. 3.7, while the downwind hydrostatic blocks in these two simulations are of different horizontal resolutions. Since few differences between Fig. 3.15 and Fig. 3.7 are observed, the downwind block has very limited impact to the upwind block.

3.6 Conclusions

The finite volume scheme using a vertical Lagrangian coordinate has proven to be efficient and useful for hydrostatic formulation-based GCMs and has been applied in several popular GCMs. Here we developed a two-way interface between models using hydrostatic formulations and non-hydrostatic formulations with vertical Lagrangian coordinates. This interface allows the finite volumes to be exchanged between different blocks as the ghost cells without knowing if the neighboring blocks are using the same governing dynamic formulations. The interface can be implemented between blocks of different horizontal resolutions with 4th order refinement algorithms. The coarsening

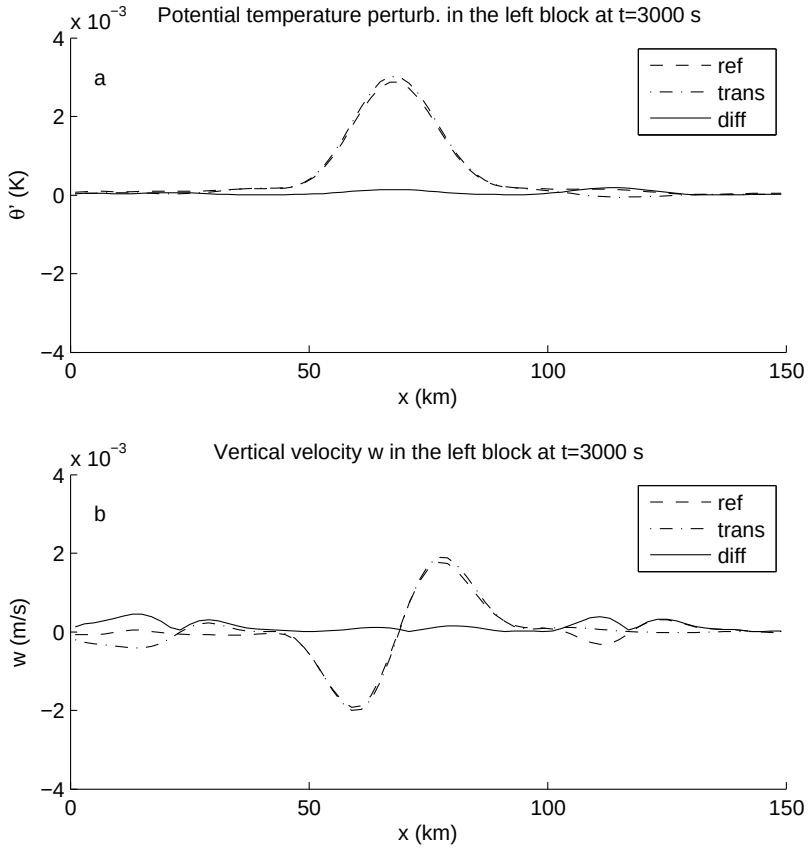

Figure 3.14: a) The potential temperature perturbation sampled at $z = 4.5$ km in range 0 to 150 km. b) The vertical velocity sampled at $z = 4.5$ km in range 0 to 150 km. In the legends: ref indicates the values in the hydrostatic channel of uniform grids; $trans$ indicates the values in the channel that connects upwind-coarse-hydrostatic block and downwind-refined-non-hydrostatic grid; $diff$ indicates the absolute difference between the two sampled values mentioned above.

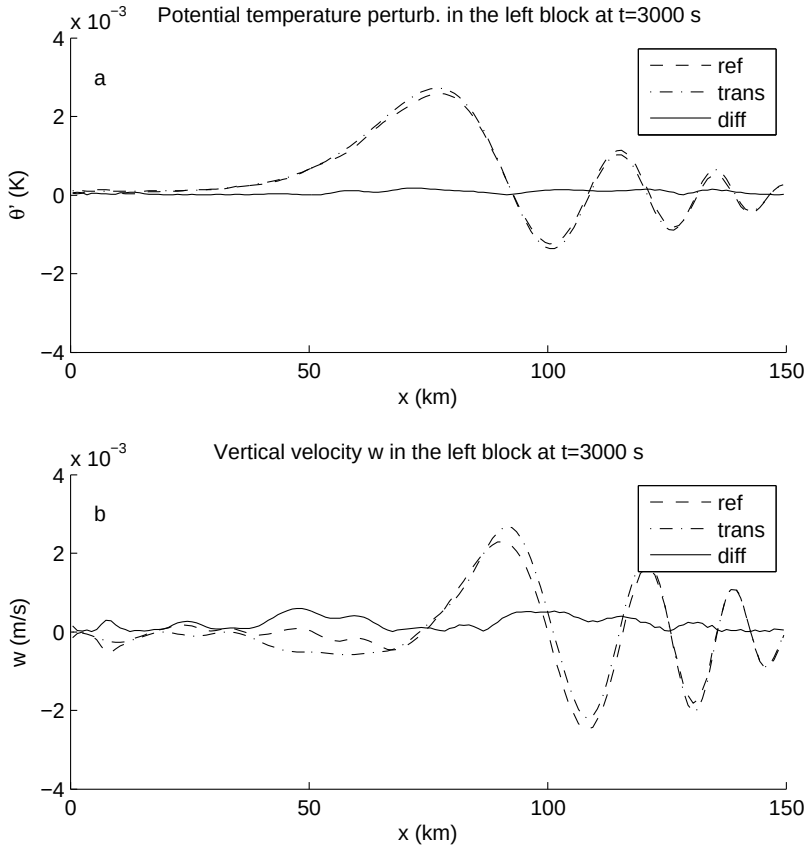

Figure 3.15: a) The potential temperature perturbation sampled at $z = 4.5$ km in range 0 to 150 km. b) The vertical velocity sampled at $z = 4.5$ km in range 0 to 150 km. In the legends: ref indicates the values in the hydrostatic channel of uniform grids; $trans$ indicates the values in the channel that connects upwind-refined-non-hydrostatic block and downwind-coarse-hydrostatic grid; $diff$ indicates the absolute difference between the two sampled values mentioned above.

algorithm and the 4th order refinement algorithms create few numerical damping at the interface.

The nonlinear hydrostatic and non-hydrostatic governing formulations do not have analytical solutions. It is also difficult to determine the correctness of the wave propagation between different types of hydrostatic and non-hydrostatic regimes. However, our interface has still shown very good smoothness and low diffusivity in two aspects: 1. The contours of both the potential temperature perturbation and the vertical velocity are smoothly connected at the interfaces. 2. The intensities of the waves in the downwind blocks are not damped. However, considering the numerical reflection is an important measurement to determine the quality of a numerical interface between non-uniform grids, it is still necessary to quantitatively determine the numerical reflection at the interface. We compared the propagation of waves between blocks of different governing dynamic equations or resolutions with their pure hydrostatic or non-hydrostatic counterparts in the upwind half channel. The numerical reflection should be smaller than the differences. Our results indicate that no obvious reflection is observed.

Our scheme uses the formulations and numerical discretization introduced by *Chen et al.* (2012), which is a low diffusive framework using A-grid. The numerical representation of each conservative variable, such as density, velocities, or potential temperature, is relatively simpler than the application using C-grid or D-grid. Our interface implementation is also benefit from the simplicity of A-grid. Thus, the low diffusivity at the interface is easier to maintain.

The vertical Lagrangian coordinate assumes the vertical coordinate is "floating" as the simulation processes. As a result, even though the vertical coordinates from different regimes are not aligned at the interfaces, they are treated as the natural coordinate deformation, thus no special treatment is required. A reasonable remap frequency is still required in order to prevent severe coordinate deformation, which

might cause overlapped vertical coordinates. We apply remap every 60 seconds.

The low Mach number approximate Riemann solver (LMARS) has a tuning parameter β. When extra less diffusivity is required, β can be set to a number less than 1. However, we recommend not modifying this parameter from 1 in order to maintain stability in most applications. Alternatively, higher order of interpolation schemes can also achieve lower diffusivity. Note that no explicit numerical diffusion is applied in either advection or interface algorithms. However, certain forms of explicit diffusion could be applied in order to parameterize the sub-grid scale turbulence or to achieve grid-independent results or convergence as the grid spacing decreases.

In a model that uses Adaptive Mesh Refinement techniques, large-scale weather systems such as mid-latitude cyclones could be properly resolved by using the hydrostatic formulations, while the small-scale weather system such as cloud formation or deep convection system could be properly resolved by using the non-hydrostatic formulations. Using the non-hydrostatic formulations for regimes using large horizontal grid spacing will introduce extra numerical efforts, e.g., techniques that conquer the small time step problem caused by the small vertical grid spacing. It also means extra computational resource requirement. However, the non-hydrostatic implementation would not provide results with significant enhanced quality as shown in our grid spacing sensitivity tests. Our work provides a method that allows a model to take full advantage of the AMR techniques and avoids implementing the non-hydrostatic formulations through the full calculation domain. Additionally, our framework uses finite volume based discretization, which is similar to the Lin-Rood framework. Thus, many existing hydrostatic based dynamic codes or parameterizations of other physical processes are still valid or only needs small modifications to be adopted into our numerical schemes. In this work, in order to examine the interface connecting different blocks with minimum diffusion introduced by the numerical algorithms, we use vertical-explicit scheme. In the future, we will explore an application with a

vertically-implicit scheme with the non-hydrostatic equations in order to allow large time steps.

CHAPTER IV

A first step towards connecting horizontally aligned computational blocks with different vertical grid spacing

4.1 Introduction

In Chapter II, we implemented the non-staggered grid based finite volume method with a vertical Lagrangian coordinate, and developed a computationally efficient Low Mach number Approximate Riemann Solver (LMARS) to calculate the fluxes or interface physical states between the control volumes. In Chapter III, we developed a grid-nested algorithm that would allow us to connect horizontally aligned two computational blocks, which are of different horizontal resolutions. Since both hydrostatic and non-hydrostatic blocks are simulated using the same numerical discretization, the algorithm that connects the blocks would be independent of the dynamical governing equations used in each block. One major issue with nested grid techniques is that, the various resolution of the nested grid can cause artificial reflections. We solved the full compressible Euler equation with and without the hydrostatic assumption, using nested grids but without any explicit diffusion. We modified the *Skamarock and Klemp* (1994) gravity wave test to validate our algorithm. The results showed almost no reflection when gravity waves propagated between blocks of non-uniform

grids or governing dynamical equations.

The algorithm developed in Chapter III might be a good start to develop a full 2D (x-y) nested-grid technique for non-uniform horizontal grid spacing. The horizontal grid spacing is usually significantly larger than the vertical grid spacing in GCMs. Some physical processes, such as hurricanes, whose vertical scale of resolution matches the GCM's vertical grid spacing, would be better simulated with refined horizontal grid spacing. The nested-grid technique of non-uniform horizontal grid spacing could be useful when these physical processes are adaptively simulated in a GCM. However, many physical processes, such as deep convection, or cloud/aerosol interactions will require much finer resolution in both horizontal and vertical directions. Then the nested-grid technique of non-uniform vertical grid spacing would be required to adaptively simulate these processes in a GCM.

Oehmke developed an Adaptive Mesh Refinement (AMR) grid library for parallel computer architectures (*Oehmke*, 2004) using a block-structured data layout in both 2D and 3D adaptive grid techniques on the sphere. This library was incorporated with a 2D shallow water model (*Jablonowski et al.*, 2006) and idealized 3D advection test (*Andronova et al.*, 2010) using a prescribed circulation. The 3D AMR library has not yet been fully integrated with 3D dynamical cores because the nesting-grid technique for non-uniform vertical grid spacing was not ready.

In this chapter, we will try to address some of the new challenges in developing the nesting-grid technique for non-uniform vertical grid spacing. As discussed in Chapter III, when waves are transported between blocks of various resolutions, artificial reflections might be caused at the block interfaces. Eliminating unrealistic reflections is the major consideration when developing nesting techniques. The nesting-grid technique of non-uniform horizontal grid refinement is based on the Newton's form of the polynomial interpolation (*Yang*, 2001). The polynomial based interpolation might not work smoothly in non-uniform vertical grid refinement, because some of the basic

numerical model requirements would be broken. For example, if a hydrostatically balanced atmospheric domain is initialized in a dynamic model, there should be no motion created. For simplicity, we will denote this property as the "stationary rule".

It is very easy to prove that the stationary rule is valid in nested grid with only non-uniform horizontal grid spacing. In this case, the vertically discretized layers of control volumes between the nested blocks and the parent blocks are perfectly connected in a one-to-one correspondence. If the computational domain is initialized statically, the values of the mean control-volume pressure within each layer are constant, since the pressure is constant at the same altitude. At the interface wherever the two grids coincide, the two grids would have overlapped boundaries if finite volume methods were used. At the boundary, the values in the control-volumes of the coarse grid are continually replaced by the values of the overlapped control-volumes in the nesting grid and vice versa. We call the control-volumes at the block boundaries that need to be replaced "ghost cells". The ghost cells of the coarse grids are updated by coarsening the overlapped control volumes in the nested grids. The ghost cells of the nested grids are updated by refining the overlapped control volumes in the coarse grids. In a static atmosphere, since the value of the pressure (as well as the temperature and density) is a constant in the horizontal direction, most grid refinement and coarsening algorithms would keep values of the pressure in the refined or coarsened volumes unchanged, thus no motion would be created since the pressure gradient will always be zero. However, when applying nesting blocks with non-uniform vertical grid spacing, the layers from different blocks would not connect in a one-to-one correspondence. One layer from the coarse grid might overlap multiple layers from the nesting grids. For example, if we choose the simplest case, in which the horizontal grid spacing is a constant and the vertical grid spacing in the nested grids is half of the size in the coarse grids, one ghost cell from the coarse grids would overlap two control volumes in the nested grids. The volume-mean values of the pressure would be different among the three cells,

which are at three different altitudes. Unfortunately, even with a static atmosphere, the vertical profile of the pressure is a monotonic non-polynomial function of the altitude. A polynomial interpolation could not guarantee to reconstruct an accurate vertical distribution of the pressure. The vertical refinement of coarsening of the ghost cells might cause pressure gradients between the interfaces, where two grids coincide. Thus, the stationary rule might be broken in nested grids with non-uniform vertical grid spacing.

The goal of this chapter is to maintain the stationary rule in grid nesting with non-uniform vertical grid spacing. Section 4.2 will first look into the model initialization process, which may break the stationary rule with static grid nesting. We will show different responses of the model when it is initialized with various vertical grid spacings. In Section 4.3 we will show that in the hydrostatically balanced isentropic atmosphere column, the vertical structure of the thermal dynamical properties such as the pressure and the density can be uniquely determined by values of the pressure at the top and the bottom of the domain and the column-mean value of the product of the density and potential temperature. An iterative method will be introduced to reconstruct the vertical distribution of pressure. Although this method would accurately reconstruct the vertical pressure distribution in isentropic environment, it is not accurate in a non-isentropic environment. In order to estimate the error when applying this method in non-isentropic atmosphere, we will compare this pressure profile reconstruction method with several polynomial interpolations. The Newton's form of polynomial interpolation is used to find the value of pressure, and is explained in Appendix B. Section 4.4 summarizes the findings and outlines the future potential of the iterative method interpolation techniques.

4.2 Discretization and initialization of the vertical atmospheric thermal dynamic profiles

The nested grids could be implemented in a statically or dynamically adaptive way. If it is implemented statically, the nested blocks and the parent blocks will be prescribed with finer and coarser grids from the initialization process. When applying nested grids with only non-uniform vertical grid spacing, the vertical distribution of the values of the density and the pressure in nested grids and coarse grids are initialized with different vertical grid spacing. The stationary rule requires that no pressure gradient should be created at the horizontal interfaces, wherever two grids coincide. For example, the pressure at the top of the model should be a constant between both grids. We will examine this property in the following sections.

4.2.1 Theoretical derivation

We denote density, pressure and temperature or potential temperature to be the thermal dynamical variables. One way to initialize a one-dimensional vertical distribution of density, temperature and pressure in a hydrostatically balanced atmosphere in a numerical model is by building the profile with the surface pressure p_{surf} and a prescribed vertical potential temperature profile $\theta(z)$. A finite volume vertical discretization is used. For example, we could label each of the control volumes using index k from the model top to bottom direction. The vertical grid spacing at level k is Δz_k. Using the equation of hydrostatic balance:

$$dp = -\rho g \, dz \tag{4.1}$$

and ideal gas law:

$$p = \rho R_d T \tag{4.2}$$

we have:

$$\frac{dp^{\kappa}}{dz} = -\frac{gp_{surf}^{\kappa}}{c_p\theta} \tag{4.3}$$

or:

$$\Delta p_k^{\kappa} = -\frac{gp_0^{\kappa}}{c_p\bar{\theta}_k}\Delta z_k \tag{4.4}$$

where c_p is the heat capacity at constant pressure, κ is the ratio of gas constant R_d and c_p, and typically we set $p_{surf} = 10^5$ Pa. $\bar{\theta}_k$ is the volume mean potential temperature in the kth control volume from the top. Denote the value of the pressure at the bottom and top surfaces of the control volume k to be $p_{k\pm0.5}$, we have:

$$p_{k-0.5} = \left(p_{k+0.5}^{\kappa} - \Delta p_k^{\kappa}\right)^{\frac{1}{\kappa}} \tag{4.5}$$

Given the prescribed surface pressure p_{surf}, we can calculate the values of the pressure at the control volume vertical interfaces from bottom to top using Eq. (4.5). And the mean density of control volume k is derived from Eq. (4.1):

$$\bar{\rho}_k = \frac{p_{k+0.5} - p_{k-0.5}}{g\Delta z} \tag{4.6}$$

The mean pressure of each control volume is calculated by:

$$\bar{p}_k = \left(\kappa\frac{p_{k+0.5} - p_{k-0.5}}{p_{k+0.5}^{\kappa} - p_{k-0.5}^{\kappa}}\right)^{\gamma} \tag{4.7}$$

where $\gamma = 1/(1-\kappa)$

The temperature profile or potential temperature profile is prescribed. The thermal dynamical variables in each control volume are then initialized.

Assuming the atmosphere column has N control volumes, the surface pressure

is $p_{surf} = p_{N+0.5}$, the pressure at the top is $p_{top} = p_{0.5}$, and the total mass in an atmosphere column above the unit surface area is:

$$M = \frac{p_{N+0.5} - p_{0.5}}{g} = \frac{p_{surf} - p_{top}}{g} \tag{4.8}$$

Given a constant surface pressure, the total mass in an atmosphere column above a unit area is determined by the pressure at the top of the column.

One approximation is made in this approach. Although the potential temperature $\theta(z)$ is prescribed, the volume-mean value of the potential temperature $\bar{\theta}_k$ used in Eq. (4.4) is approximated using $\theta(z_k)$, where z_k is the height of the central point of control volume k. However, the potential temperature at the central point of the control volume does not necessarily equal to the mean value of the potential temperature. When the atmosphere is isentropic, i.e. $\theta(z)$ is a constant, $\bar{\theta}_k = \theta(z_k)$, the initialization is accurate, and the value of the pressure at the top of the column p_{top} would be a constant no matter what vertical grid spacing Δz is used. In a non-isentropic atmosphere, smaller grid spacing of Δz would lead to more accurate value of p_{top}.

In the following subsection, we will test this initialization procedure using two kinds of potential temperature distributions.

4.2.2 Isentropic atmosphere initialization

A 1D (z) isentropic environment as an atmospheric column is initialized. The surface pressure is $p_{surf} = 10^5$ Pa. The vertical distribution of the potential temperature is constant $\theta(z) = 300$ K. The height of the atmosphere is $H = 10$ km, and the column is vertically discretized in N equal control volumes. We performed the initialization process described in Section 4.2.1. Fig. 4.1 shows the vertical distribution of pressure using $N = 1000$.

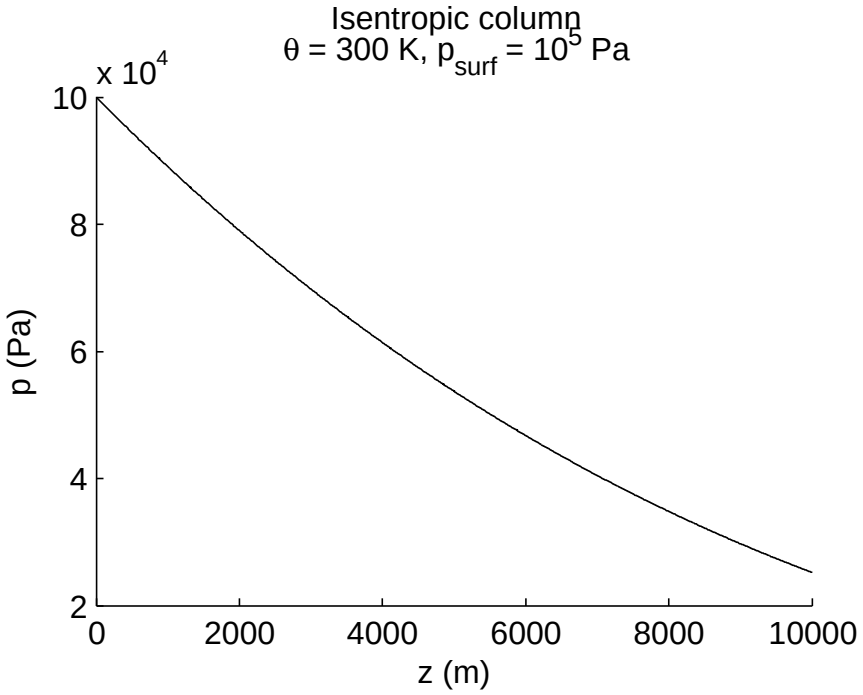

Figure 4.1: The vertical distribution of pressure in isentropic atmosphere with surface pressure of 10^5 Pa and potential temperature of 300 K

The value of the pressure at the model top p_{top} is a constant of 25220 Pa with different N of 1, 10, 100 and 1000. Note that when $N = 1$, the total column of the atmosphere is just a single control volume.

In this subsection, we have shown that with the initialization of a hydrostatic atmosphere column based on uniform vertical potential temperature profile, the value of the pressure at the top of the domain is a constant and is independent of the number of discretized volume used.

4.2.3 Initialization of the atmosphere with constant Brunt-Väisälä frequency

Similar to the isentropic atmosphere initialization, the surface pressure is $p_{surf} = 10^5$ Pa. The column of the atmosphere is of the height $H = 10$ km, and to be discretized in N control volumes. The background atmosphere has a constant Brunt-Väisälä frequency of 10^{-2} s^{-1} with a surface temperature of 300 K. Fig. 4.2 shows

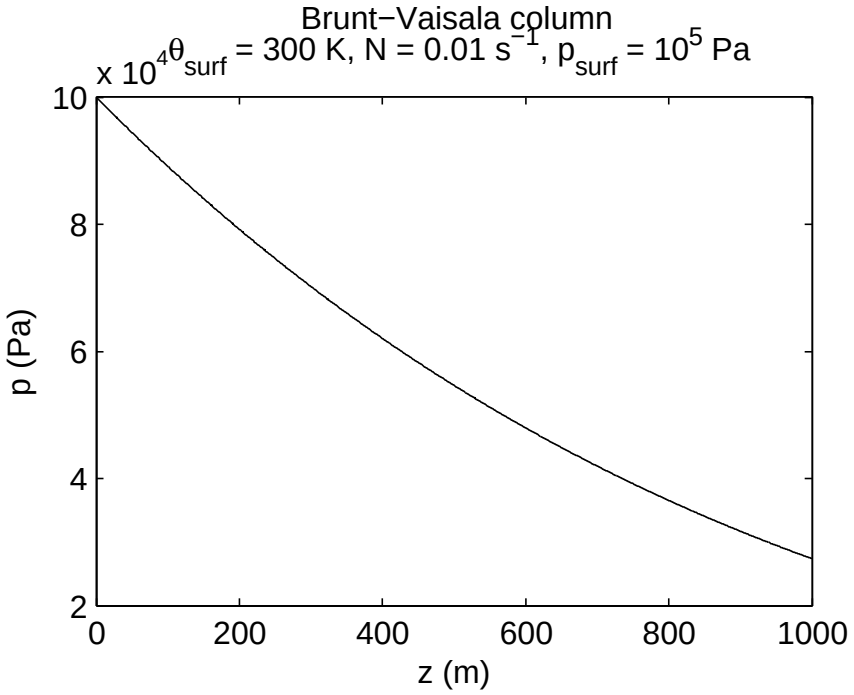

Figure 4.2: The vertical distribution of pressure in atmosphere with a constant Brunt-Väisälä frequency of 0.01 s^{-1}, surface pressure of 10^5 Pa and surface potential temperature of 300 K

the vertical distribution of pressure using $N = 1000$.

The value of the pressure at the top of the column p_{top} is not a constant when applying different vertical grid spacing. We have a converged value of $p_{top} = 27382$ Pa when $N = 1000$. If we use $p_{top}|_{N=1000}$ as the reference value, and denote the errors to be the absolute differences between $p_{top}|_{N=1000}$ and p_{top} calculated using other values of N. Then we could plot $\ln(err) vs. \ln(dz)$ in Fig. 4.3: in which the slope of the solid line is 2, indicating the value of the pressure at the column top is second order accurate with respect to dz. If the value of the pressure is not a constant when different vertical spacing are used, then the total mass of columns with different vertical grid spacing will not be the same value, although the physical properties of the atmosphere, such as the potential temperature profile, surface pressure, are the same. Additionally, the pressure of the same altitude would not be the same between columns with different vertical grid spacing, e.g. the p_{top} would not be the same. Pressure gradients would

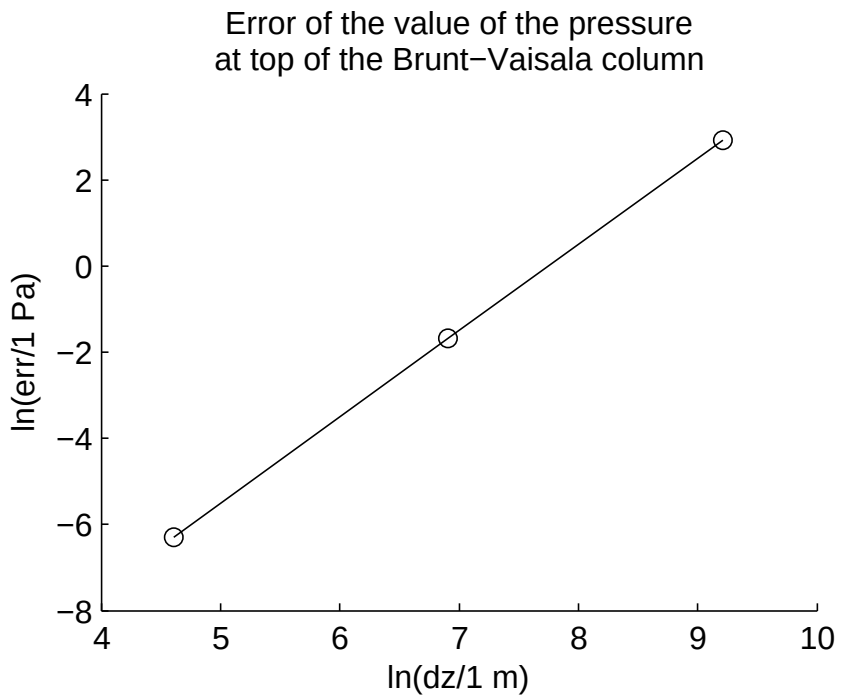

Figure 4.3: $\ln(err) vs. \ln(dz)$ where err is the error of the value of the pressure at the column top. The slope of the solid line is 2.

be formed, wherever two grids with different vertical resolution coincided, and the stationary rule would be broken.

Thus, in order to keep the stationary rule when initializing the non-isentropic atmosphere, it would be better to use a relatively finer vertical grid spacing to determine the vertical pressure profile, when the value of the pressure at the model top converges. Then we can discretize the atmosphere using the relatively continuous vertical pressure distribution to calculate the volume-mean density using Eq. (4.6).

4.2.4 Summary of this section

In this section, we reviewed some initialization techniques in idealized simulations. We have shown that the isentropic atmosphere can be accurately initialized. However, the initialization of an atmosphere column based on non-uniform vertical potential temperature profile will introduce second order errors. The inaccurate initialization of the non-isentropic atmosphere would break the stationary rule when nested grid with

non-uniform vertical grid spacing is used. The alternative initialization of the non-isentropic atmosphere should use a finer grid to build a converged vertical pressure profile before vertically discretize the atmosphere into control volumes.

4.3 Reconstruction of the vertical pressure distribution using an iterative function

Grid nesting with non-uniform vertical grid spacing requires vertical control volume coarsening and refinement procedures. For simplicity, we assume that when two grids coincide, one control volume from the coarse grids exactly overlaps two control volumes from the nested grids, and the vertical grid spacing of the nested grids is a constant within each type of grids. The coarsening process is to update values, such as density, potential temperature and momentum, of the coarse control volume using the two nested control volumes. The values are calculated:

$$\overline{\rho}^{coarse} = \frac{1}{2}\left(\overline{\rho}^{nest}_{up} + \overline{\rho}^{nest}_{down}\right) \tag{4.9}$$

$$\overline{\rho \boldsymbol{v}}^{coarse} = \frac{1}{2}\left(\overline{\rho \boldsymbol{v}}^{nest}_{up} + \overline{\rho \boldsymbol{v}}^{nest}_{down}\right) \tag{4.10}$$

$$\overline{\rho\theta}^{coarse} = \frac{1}{2}\left(\overline{\rho\theta}^{nest}_{up} + \overline{\rho\theta}^{nest}_{down}\right) \tag{4.11}$$

where *up* or *down* indicates the overlapped nesting grids at higher or lower position within the coarse control volume.

However, control volume refinement requires the knowledge of the pressure at the location where the coarse-grid control volume is divided. In the following subsections, we will discuss how to rebuild a pressure profile within a coarse-grid control volume.

4.3.1 Theoretical derivation

In a column of the hydrostatically balanced atmosphere above a unit area, given any two points at the height of z_1 and z_2 ($z_1 > z_2$) with values of the pressure at these two points p_1 and p_2, we define the weight of the air mass segment of the column between the two levels:

$$\bar{\rho} g(z_1 - z_2) = p_2 - p_1 \tag{4.12}$$

where $\bar{\rho}$ is the average density within the column segment of air:

$$\bar{\rho} = \frac{1}{z_1 - z_2} \int_{z_2}^{z_1} \rho \, dz \tag{4.13}$$

We also denote the flux form volume mean potential temperature times density as $\overline{\rho\theta}$.

$$\overline{\rho\theta} = \frac{1}{z_1 - z_2} \int_{z_2}^{z_1} \rho\theta \, dz \tag{4.14}$$

Given the two pairs (z_1, p_1), (z_2, p_2) and $\overline{\rho\theta}$, our algorithm is to develop the function of the pressure $p(z)$.

With hydrostatic balance, the averaged potential temperature between two levels of the height z_A and z_B is given by:

$$\overline{\theta}_{AB} = -\frac{z_A - z_B}{p_A^\kappa - p_B^\kappa} \frac{g p_0^\kappa}{c_p} \tag{4.15}$$

where $p_0 = 10^5$ Pa is the value of the pressure at the surface.

For any given z between z_A and z_B, the mean potential temperature between z_1 and z is $\bar{\theta}_1$, and the mean potential temperature between z and z_2 is $\bar{\theta}_2$. Then we

have:

$$\overline{\theta}_1 = -\frac{z_1 - z}{p_1^\kappa - p^\kappa}\frac{gp_0^\kappa}{c_p} \tag{4.16}$$

$$\overline{\theta}_2 = -\frac{z - z_2}{p^\kappa - p_2^\kappa}\frac{gp_0^\kappa}{c_p}. \tag{4.17}$$

The volume mean density between z_1 and z is $\bar{\rho}_1$, and the mean density between z and z_2 is $\bar{\rho}_2$. From the potential temperature conservation we get:

$$(z_1 - z)\left(\overline{\rho\theta}\right)_1 + (z - z_2)\left(\overline{\rho\theta}\right)_2 = (z_1 - z_2)\overline{\rho\theta}. \tag{4.18}$$

Note that the volume mean potential temperature times the density $\overline{\rho\theta}$ is not necessarily equal to $\bar{\rho}\bar{\theta}$, but here we make the approximation that:

$$\overline{\rho\theta} = \bar{\rho}\bar{\theta}. \tag{4.19}$$

Then we have:

$$(z_1 - z)\bar{\rho}_1\bar{\theta}_1 + (z - z_2)\bar{\rho}_2\bar{\theta}_2 = (z_1 - z_2)\overline{\rho\theta}. \tag{4.20}$$

With Eq. (4.12), we have:

$$(p_1 - p)\bar{\theta}_1 + (p - p_2)\bar{\theta}_2 = -(z_1 - z_2)g\overline{\rho\theta}. \tag{4.21}$$

Eliminating $\bar{\rho}_1$ and $\bar{\rho}_2$ using Eq. (4.16) and (4.17). Denote $h_1 = z_1 - z$, $h_2 = z - z_2$, we have:

$$f(p) = \frac{gp_0^\kappa}{c_p}h_1\frac{p_1 - p}{p_1^\kappa - p^\kappa} + \frac{gp_0^\kappa}{c_p}h_2\frac{p_2 - p}{p_2^\kappa - p^\kappa} - (z_1 - z_2)g\overline{\rho\theta} = 0. \tag{4.22}$$

$f(p)$ can not be solved analytically. However, it can be solved using a secant iteration method with two initial values for p.

The secant method is defined by the recurrence relation:

$$p^{(n)} = p^{(n-1)} - f(p^{(n-1)}) \frac{p^{(n-1)} - p^{(n-2)}}{f(p^{(n-1)}) - f(p^{(n-2)})} \tag{4.23}$$

and one initial value of p could be found using a first order linear interpolation:

$$p^{(0)} = \frac{h_2 p_1 + h_1 p2}{H} \tag{4.24}$$

where

$$h_1 = z_1 - z \tag{4.25}$$

$$h_2 = z - z_2 \tag{4.26}$$

$$H = z_1 - z_2 \tag{4.27}$$

The second initial value of p could be found using an exponential interpolation of p:

$$p^{(1)} = p_2 \exp\left(-\frac{h_2}{H_s}\right) \tag{4.28}$$

where H_s is the scale height, defined as $p_1 = p_2 \exp(-H/H_s)$, and calculated from:

$$H_s = \frac{H}{\ln\left(\frac{p_2}{p_1}\right)} \tag{4.29}$$

and the pressure $p(z)$ can be determined by the converged value of Eq. (4.22).

This method could be used in vertical refinement techniques. Consider a control volume in the coarse vertical grid, of which the top and bottom surfaces are located

at z_1 and z_2, which is to be refined into two smaller control volumes at a point in the middle of the height z. Since we can derive the hydrostatic pressure $p(z)$ using Eq. (4.22), the values of the volume mean density of the refined control volumes are:

$$\bar{\rho}_1 = \frac{p(z) - p(z_1)}{g(z - z_1)} \tag{4.30}$$

$$\bar{\rho}_2 = \frac{p(z) - p(z_2)}{g(z - z_2)} \tag{4.31}$$

Note that the approximation we made is to assume $\overline{\rho\theta} = \bar{\rho}\bar{\theta}$. This approximation will be exact in an isentropic atmosphere since the potential temperature is a constant. So, we would expect an accurate reconstruction of pressure in an isentropic atmosphere column, but there might be some errors when reconstructing non-isentropic control volumes.

4.3.2 The reconstruction of the vertical profile of pressure in an isentropic atmosphere

In this test, the vertical one-dimensional isentropic atmosphere column initialized in Section 4.2.2 using $N = 1000$ is used as the reference. In order to determine the vertical distribution of the pressure $p(z)$, we will use the pressure at the top of the atmosphere $p_{top} = 25220$ Pa at $z = 10$ km, and the surface pressure $p_{surf} = 100000$ Pa, at $z = 0$ km and we use the volume mean value of the product of the density and the potential temperature 228.774 kgK/m^3. The column will be discretized into 10 control volumes with equal vertical grid spacing of $dz = 1$ km. The values of pressure at each control volume interface will be calculated, and compared to the values of pressure in the column that we initialized in Section 4.2.2. The results are shown in Fig. 4.4.

If we denote the error as the absolute difference between the reconstructed values

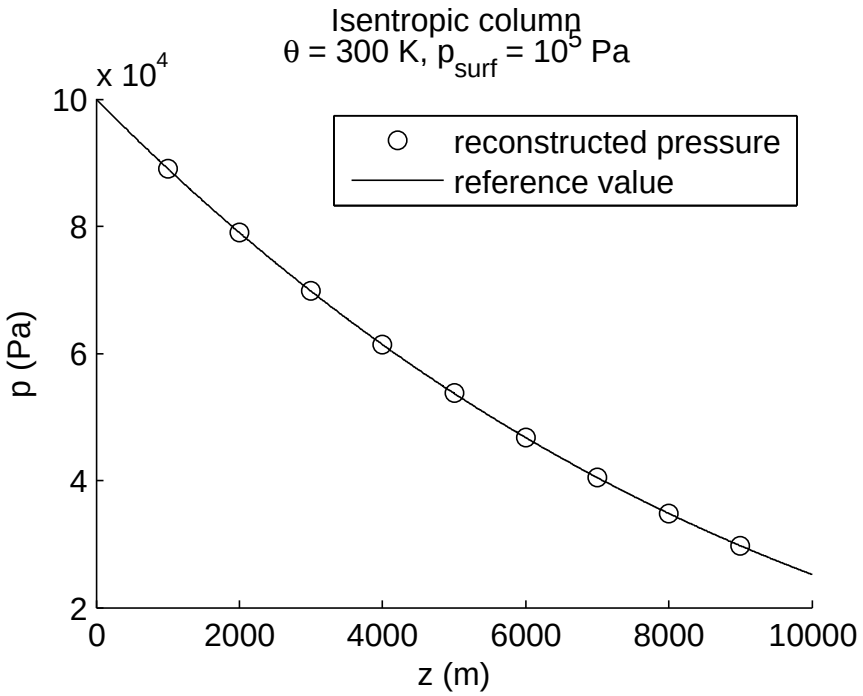

Figure 4.4: The reconstructed vertical distribution of the pressure in an isentropic column. The circles are the reconstructed pressure, and the solid line is the value of the vertical profile of the hydrostatic pressure built in Section 4.2.2 with N=1000

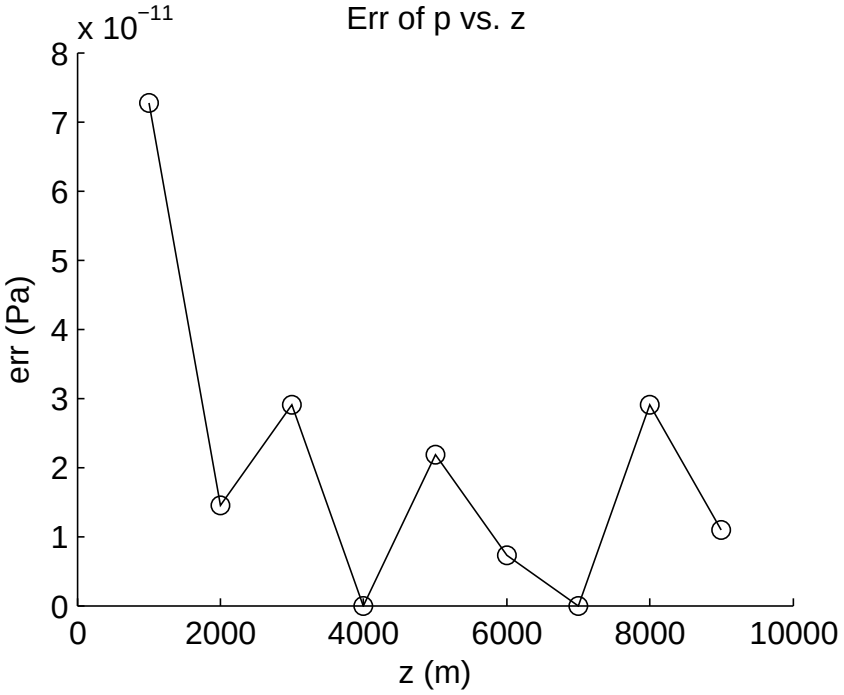

Figure 4.5: The absolute error of the reconstructed pressure with respect to the vertical location

of the pressure and the reference values of the pressure, we could plot the $errvs.z$ in Fig. 4.5.

Note that the convergence condition we used in Eq. (4.22) is $abs(p^{(n)} - p^{(n-1)}) < 10^{-10}$ Pa, and the order of magnitude of the error is 10^{-10} Pa. Thus, we consider the reconstruction of the pressure is sufficiently accurate.

4.3.3 The reconstruction of the vertical profile of the pressure in the atmosphere with constant Brunt-Väisälä frequency

We will use the atmosphere column with constant Brunt-Väisälä frequency initialized in Section 4.2.3 as the reference. The initial top pressure is $p_{top} = 27382$ Pa at $z = 10$ km, and the initial surface pressure is $p_{surf} = 100000$ Pa, at $z = 0$ km, the volume mean value of the product of the density and the potential temperature is 231.901 kgK/m^3. The column is discretized into 10 control volumes with equal vertical grid spacing of $dz = 1$ km. The values of pressure at each control volume

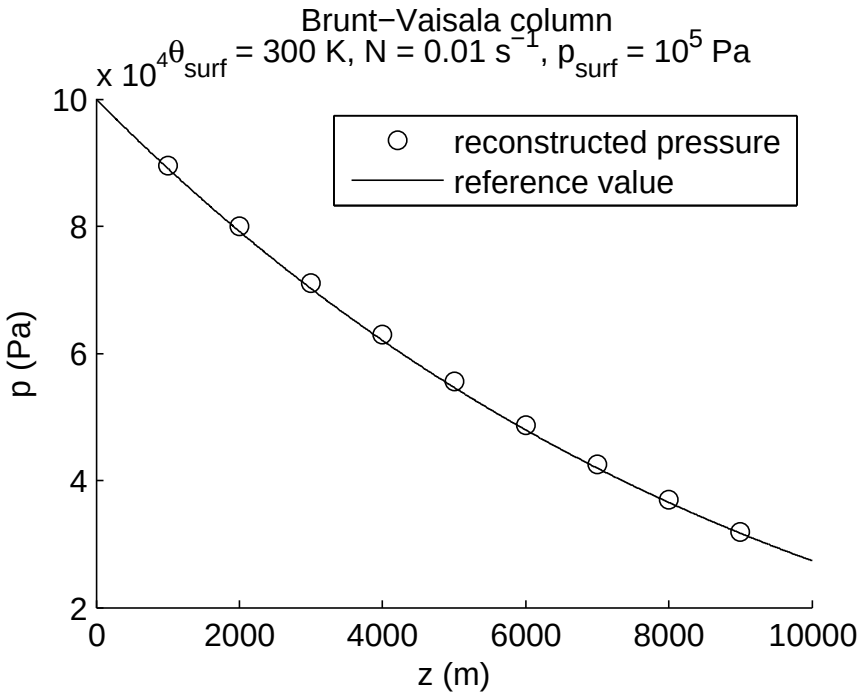

Figure 4.6: The reconstructed vertical distribution of the pressure in the atmosphere with constant Brunt-Väisälä frequency. The circles are the reconstructed pressure, and the solid line is the value of the vertical profile of the pressure built in Section 4.2.3 using $N = 1000$

interface will be calculated, and compared to the reference values. The results are shown in Fig. 4.6.

We could calculate the errors in the same way as we did in the previous subsection. Since the results are not accurate, the absolute errors are a function of the grid spacing. Instead of the plotting absolute error, we plot the relative error $err/p \times 100\%$ in Fig. 4.7.

As illustrated in Fig. 4.7, we observe that in the reconstructed pressure profile, the minimum relative error is 0.55% at height $z = 5$ km. The reconstruction of vertical distribution of the pressure is not accurate because of the approximation $\overline{\rho\theta} = \bar{\rho}\bar{\theta}$.

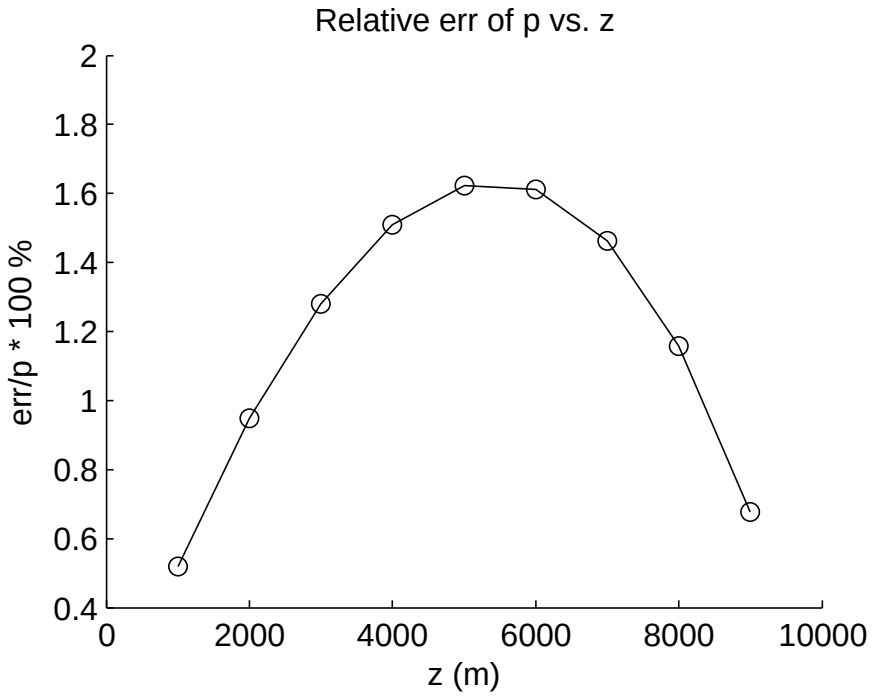

Figure 4.7: The relative error of the reconstructed pressure with respect to the vertical location

4.3.4 Comparison between the iteration-based interpolation scheme and the polynomial-based interpolation schemes

Next we will compare this method to a polynomial based interpolation scheme. In this test, we will also use the atmospheric column with a constant Brunt-Väisälä frequency initialized in section 4.2.3 as the reference. The column is discretized into 5 equal control volumes. Assume we need to refine the volume at the location from $z = 4$ to 6 km into two volumes by the mid-point at $z = 5$ km. The reference value of pressure at $z = 5$ km is 54639 Pa.

We label the sets (p_k, z_k) from the top down with index $k = 1, 2, ..., 6$ and $z_k = (12 - 2k)$ km. We denote the iterative based vertical pressure interpolation to be vPrsBuilder. The input values for vPrsBuilder are the sets of (p_3, z_3), (p_4, z_4) and the volume mean $\overline{\rho\theta}$ between z_3 and z_4, which is calculated from the reference profiles of $\rho(z)$ and $\theta(z)$. We use Newton's form of the polynomial interpolation to

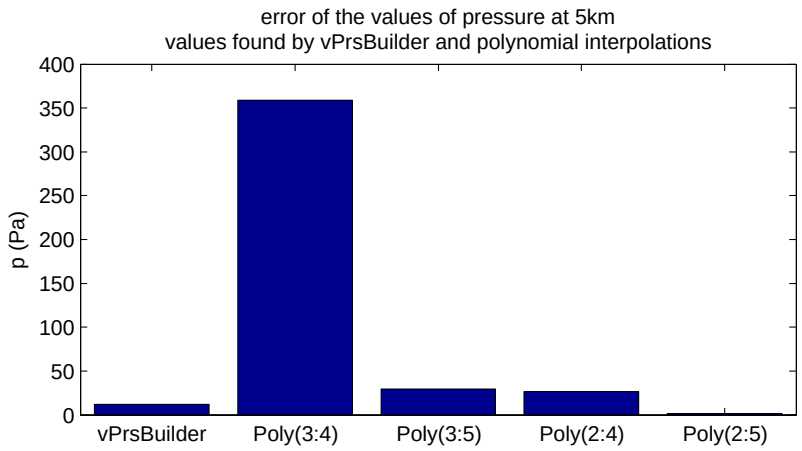

Figure 4.8: The error of the values of pressure at 5 km. $Poly(i:j)$ denotes the value of pressure is calculated from polynomial interpolation, which taken the sets from (p_i, z_i) to (p_j, z_j) as the input

interpolate the value of p at $z = 5$ km. We denote the values calculated by polynomial interpolations to be $Poly(i:j)$, which takes the sets from (p_i, z_i), to (p_j, z_j) as the inputs.

The errors in the estimates of p at $z = 5$ km is shown in Fig. 4.8. We can see that the vPrsBuilder is more accurate than the linear interpolation of $Poly(3:4)$ and the parabolic interpolations of $Poly(3:5)$ and $Poly(2:4)$, but is less accurate than the 3rd order interpolation of $Poly(2:5)$.

4.3.5 Summary of this subsection

In order to maintain the stationary rule in the vertical control volume refinement, we need accurate values of the pressure at the interfaces of the refined volumes. Unfortunately, the value of the pressure could not be accurately found by the polynomial interpolations because the vertical distribution of the pressure is a non-polynomial function of the height. This section introduced an alternative method to find the pressure at any location between two given points of pressure and the altitude. If the control volume is isentropic, the iteration method will find an accurate value of the pressure, and the stationary rule is maintained. However, this method does not work

well when the control volume is not isentropic.

4.4 Conclusions

In this chapter, we discussed the challenges when applying nested grid techniques using non-uniform vertical spacing at the fine-coarse grid boundaries. Particularly, if a static atmospheric domain is initialized in a dynamic model, there should be no motion created by the model. For simplicity, we denote this property as the "stationary rule". We explained that the stationary rule is not satisfied when applying grid-nesting techniques with a non-uniform vertical spacing.

We discussed the initialization methods in building a vertical distribution of the density and pressure with the surface pressure and a prescribed vertical distribution of potential temperature. We showed that the profile of pressure or density would not be accurately initialized unless the atmosphere is isentropic, i.e., the prescribed potential temperature is a constant. When the potential temperature is not a constant with height, the volume central point value of the potential temperature is considered to be the volume mean potential temperature, which introduced errors. The initialized vertical distribution of pressure is second order accurate with respect to the vertical grid spacing dz. The inaccuracy introduced by interpolation would cause pressure gradients when two grids with different vertical grid spacing coincide, and this would break the stationary rule.

We have also proved that the vertical distribution of the pressure in the hydrostatically balanced atmosphere column could be uniquely determined by the pressure at the top and bottom of the column and the column mean value of the product of the density and the potential temperature. The value of the pressure at any altitude within the column can be calculated using an iterative method. With the isentropic atmospheric column, the vertical distribution can be accurately rebuilt to the accuracy of the convergence condition of the iterative method. Thus, if the refinement

of the pressure is calculated using our iterative-based method in an isentropic atmosphere, the stationary rule will be maintained for grid nesting with non-uniform vertical grid spacing. However, since we made an assumption that the volume mean value of the product of the density and potential temperature is equal to the product of the individual volume mean value of the density and potential temperature, we introduced errors in the reconstruction of the vertical pressure profile in non-isentropic atmosphere.

This work is still under development. To fully implement this scheme with the nested grid, we need three steps: 1. Test the "stationary rule" with an isentropic atmosphere with blocks in different vertical different resolutions. 2. Improve the formulation to better represent the volume mean value of the product of the density and the potential temperature to improve Eq. (4.22). 3. Then test the "stationary rule" with a non-isentropic atmosphere with blocks in different vertical different resolutions.

CHAPTER V

Summary and future research directions

The dynamical core of GCMs is built to evolve. Modelers always borrow ideas from earlier models or ideas from other fields in devising new techniques of their own to better represent the equations of motion with a suitable computational cost. The increasing speed and memory of supercomputers have been driving the evolution of development of the dynamical cores. Today's supercomputer is of a massively parallel architecture, with hundreds of thousands of interconnected processors working in parallel. Current development of the dynamic core would require the ability to divide the computational workload into small computational units, which could be distributed to the low-cost networked processors.

The traditional spectral method, which requires global communication at each time step, is difficult to implement in parallel computers, even though it is one the fastest methods for obtaining solutions at high accuracy. The most widely applied method today in computational fluid dynamics (CFD) is the finite volume method. Its popularity is due to its generality, its conceptual simplicity and the relative ease of application to any kind of grids. It is also relatively easy to update each finite volume using only a few neighbors, thus the finite volume method is well suited for parallel computing. Additionally, it is also easy to enforce the conservation, positivity and monotonicity with finite-volume method, which is necessary for accurate tracer

advection. All these advantages indicate that finite volume method is suited to the evolution of the dynamical cores.

One of the most popular finite-volume dynamical cores is the so-called Lin-Rood core. There are two signature features of the Lin-Rood framework. First, the staggered C- and D-grid helps to bypass the expensive Riemann solvers when calculating the fluxes between the finite volumes. Second, the application using the vertical Lagrangian coordinate takes full advantage of the stratification property of the atmospheric flows, and eliminates the calculation of fluxes between the vertical control volume interfaces. This model has proved effective for various applications to resolve atmosphere processes in hydrostatic models. It is also possible to use adaptive mesh refinement (AMR) techniques to implement this dynamical core to locally resolve the small-scale motions. However, staggered grid systems cause much more inconvenience than non-staggered grids when applying the finite-volume method with non-hydrostatic formulations. So there is a big challenge in the evolution of the finite-volume dynamical cores: how can we build a non-staggered-grid-based finite-volume dynamical core, which could match the performance of the existing Riemann-solver-free staggered-grid-based dynamical core? In this thesis, a new scheme using non-staggered finite volume methods to solve the full set of conservative equations was designed. In order to match the efficiency of the staggered scheme, we developed a high performance Riemann solver. This method was validated in both hydrostatic and non-hydrostatic atmospheres, and produced good results. We also developed some techniques to apply this non-staggered dynamical core in non-uniform grids, which is required for AMR applications.

5.1 Summary

In Chapter II, we described how to formulate both the hydrostatic and non-hydrostatic flux-form conservative primitive equations with a vertical Lagrangian co-

ordinate. The primitive equations has been designed for non-staggered grids finite-volume schemes, where the state variables, such as density, velocity and potential temperature are all volume-mean. The floating vertical Lagrangian coordinate would eliminate all the fluxes between the interfaces of the Lagrangian layers, thus simplify the system into pure horizontal flows. Although we only discussed the 2D (x-z) system of equations, the numerical discretization is horizontally isotropic due to the non-staggered grids. So the extension of full 3D equation in the Cartesian coordinate would be straightforward. Finite volume methods that implemented with non-staggered grids typically use Riemann solvers for computing fluxes along volume edges. The traditional Riemann solvers were designed for the fields, which dealt with high-speed flows and strong discontinuities, thus are computationally expensive or too diffusive for low-speed flows. In order to match the Riemann-solver-free staggered finite-volume scheme, we took the advantage that atmospheric flows are low-speed (low Mach number) and smooth, and we could safely assume that the volume interfaces are locally isothermal. Based on this assumption, we invented a new Low Mach number Approximate Riemann Solver (LMARS) which is in a surprisingly simple form and provide "just enough" implicit diffusion for stability.

Chapter III is an extension from Chapter II. Since one of our ultimate purpose is to adaptively resolve cloud using non-hydrostatic schemes within a GCM, we need to develop algorithms to connect the hydrostatic regimes and non-hydrostatic regimes with non-uniform grids. We connected the hydrostatic and non-hydrostatic models in two computational blocks with non-uniform horizontal resolution, which is a kind of nested grids. However, one major issue with grid nested is that varying resolution can cause artificial reflections and refractions of waves due to incompatible mechanisms at coarse-grid interfaces. This problem is even harder to solve with two-way nesting, in which the solution on the coarse grid is continually updated by that on the nested grid wherever the two grids coincide. Since both hydrostatic and non-hydrostatic

models shares the same discretization, the control volumes are defined the same way in both models, and the interface information exchange process is independent to whether the hydrostatic or non-hydrostatic model is used. We used the polynomial interpolation to rebuild the profile of state variables within the finite volumes and use this technique for horizontal volume refinement. Our results shows both very few reflections, and low diffusivity at the interfaces between the computation blocks.

To fully implement a 3D AMR technique within the hydrostatic GCM, we have to implement the techniques to connect blocks with different vertical grid spacing. We discussed this application in Chapter IV. Although using the polynomial interpolation to rebuild the profile of state variables is successful with grid nesting with non-uniform horizontal resolution, it is would almost inevitably create reflections with grid nesting with non-uniform vertical resolution. We denote a "stationary rule" that a nested grids should not create any motion if the atmosphere is initialized statically. We examined model initialization process and found that with non-isentropic atmosphere, even with identical surface pressure and back ground potential temperature profiles, different values of model-top pressure will be initialized with different vertical grid spacing and the error is in 2nd order accuracy. We also developed an iterative interpolation technique to accurately reconstruct the vertical distribution of pressure. This new technique can find the theoretical value of the pressure at any given altitude within a control volume. Using this method in the volume refinement would prevent any pressure gradient at the interfaces, where two grids coincide, and maintain the stationary rule. Unfortunately, this technique only works with isentropic atmosphere, and is still under development.

5.2 Accomplishments and future work

The biggest accomplishment is the simplicity of all the new numerical techniques we have developed. Simplicity translates into less computational steps when evaluat-

ing the fluxes in finite volume methods. It also allows more elegant coding.

The finite volume method has a history of use of over thirty years and has proven to be robust, accurate and relatively easy to implement. When borrowing ideas from other fields into atmospheric numerical flow simulation, we should optimize these ideas with the properties of atmospheric flows. For example, the implementation of the vertical Lagrangian coordinate in a finite volume scheme took advantage of the fact that the atmospheric flows are highly stratified. However, in order to prevent the expensive computational cost of the Riemann solver, alternative approaches, such as staggered grids, were applied. In this work, we have reexamined the non-staggered finite volume scheme using Riemann solvers. Based on the assumption of local isothermal finite volume interface, which is acceptable in atmospheric flows, we derived a new simple form of the Riemann solver. We implement robust non-staggered grid dynamic core without the need for any explicit diffusion, and could easily extend this method to high-order accuracy with high-order polynomial interpolation and a Runge-Kutta time marching scheme.

This dissertation proposes a dynamical core that has the potential to achieve high-order-accuracy on massively parallel systems. At the same time, this dynamical core maintains a high efficiency when used with the hydrostatic formulation because it adopts a vertical Lagrangian coordinate. Our next step is to extend the two-dimensional formulations on Cartesian coordinates into three-dimensional formulations on a sphere. Since the primitive equations used in this work are similar to the primitive equations used in the Lin-Rood dynamical core, it does not require significant modification to implement our numerical scheme into the current popular finite-volume dynamic cores. Incorporating water into the model is also one of our next steps. Our ultimate purpose is to implement this 3D dynamical core with our AMR library (*Oehmke*, 2004) to adaptively capture features of particular dynamical interest.

APPENDICES

APPENDIX A

Validation tests on the non-staggered grid finite volume algorithm

A.1 Comparison of the Eulerian and Lagrangian formulations with large viscosity for the Gaussian warm bubble test

The low Mach number approximate Riemann solver (LMARS) can be used in both the Eulerian and Lagrangian frameworks. In order to demonstrate that both methods converge to a similar result, we added a 2nd-order explicit damping term to the momentum equation.

We repeated the rising Gaussian warm bubble test from Robert (1994), with a strong diffusion term of the form $\pi K(u_{xx} + u_{zz})$ and $\pi K(w_{xx} + w_{zz})$ added on the right hand side of the horizontal and vertical momentum equations to represent the addition of a term with large viscosity. The diffusion coefficient K was set to a value of 75 m^2/s. We compared the results at different resolutions: $dx = dz = 5$, 10, 20 and 33 m and at the time $t = 18$ min. The results are illustrated in Fig. A.1. Because the

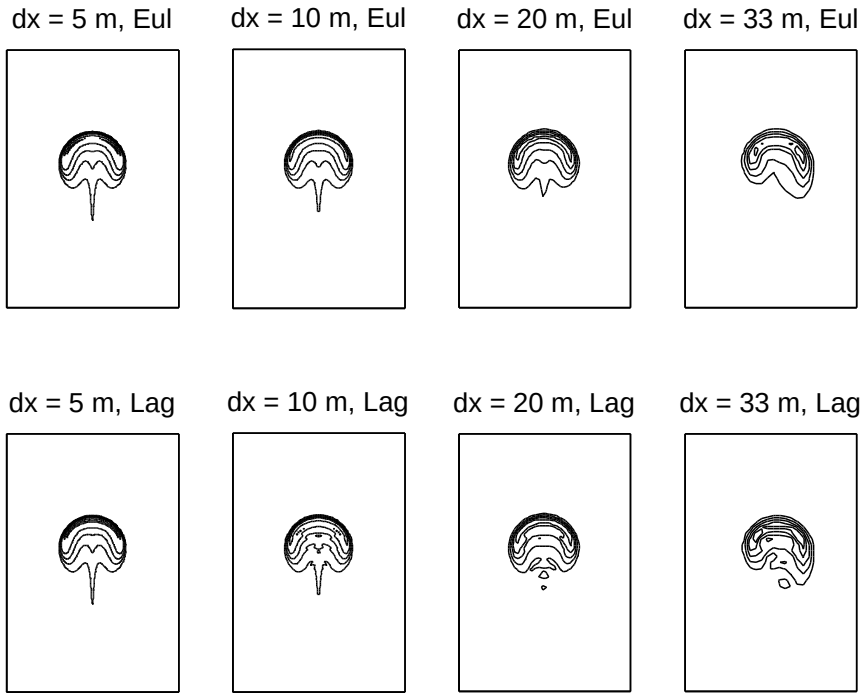

Figure A.1: The rising Gaussian bubble test for different $dx = dz$ resolutions simulated with the Eulerian (top row) and Lagrangian (bottom row) approach. A 2nd-order explicit diffusion term with the coefficient set to 75 m^2/s is added to the system. The domain of the simulation is 1 km by 1.5 km. The contour interval is 0.1 K, and starts at 0.05 K.

diffusion term changes the viscosity of the air, the Kelvin-Helmholtz instability is not observed. Additionally in the momentum equations, $\partial u/\partial x$ is horizontally symmetric with respect to the central vertical line, but $\partial^2 u/\partial x^2$ is symmetric with opposite signs. As a result, the symmetry is broken when the diffusion term is added into the system.

The Eulerian and Lagrangian formulations started to have converged results at a resolution of $dx = dz = 10$ m. When the grid spacing increases, the quality of the results is degraded in both the Eulerian and Lagrangian versions. In particular, the Eulerian version maintains "smoother" results than does the Lagrangian version. Thus, we judge that the Eulerian formulation produces results that are in better agreement with the converged solution at high resolution than is the Lagrangian version. In our simulations, the results calculated using the Eulerian formulation are used as the reference solution.

A.2 Order of accuracy analysis

The advection terms in our numerical scheme are discretized using a 4th-order accurate method, and the 4-step Runge-Kutta method provides 4th-order accuracy in time. Thus, the one-dimensional simulations are of 4th-order accuracy. In two-dimensional simulations, however, the advection fluxes are based on averaged values along the interface. Because the flux is not a linear function of the state quantities, fourth-order accuracy is downgraded to second-order accuracy. However, the use of a high-order interpolation scheme on the advection terms is able to ensure low diffusivity. This appendix reports several simulations to validate the order of the accuracy of our numerical scheme.

A.2.1 1D simulation error analysis

The full set of 1D compressible Euler equations permits acoustic waves. A 1D (x) domain of 3 km length with periodic boundary conditions is used to demonstrate the

errors inherent in our scheme. The background pressure is set to $1e5$ Pa, and in order to keep the acoustic speed at $c_s = 300$ m/s, the background temperature is set to:

$$T = \frac{c_s^2}{\gamma R_d} \tag{A.1}$$

where γ is the adiabatic index, the ratio of specific heats of the gas at constant-pressure to the gas at a constant-volume (c_p/c_v), and R_d is the gas constant for dry air. The background density is calculated from:

$$\rho = \frac{p}{R_d T} \tag{A.2}$$

An initial temperature perturbation with a Gaussian distribution is added to the background temperature to initiate the formation of acoustic waves:

$$T' = Ae^{-\frac{(x-x_0)^2}{s^2}} \tag{A.3}$$

where $A = 0.5$ K, $x_0 = 1500$ m, $s = 100$ m. The acoustic waves will travel in both directions at the speed of $c_s = 300$ m/s after the wave is fully developed. The result was sampled at $t = 48$ s to evaluate the order of accuracy of the numerical scheme. The grid spacing is varied from $dx = dz = 5$ up to 40 m. The horizontal velocity at $t = 48$ s is shown in Fig. A.2.

Since no analytical solution is available for this test, we treat the result with the finest resolution as the reference, and compare the results from coarser resolutions to this reference. In order to compare the results at different resolutions, the reference solution was averaged to the coarser grids. Because finite volume grids are used in our

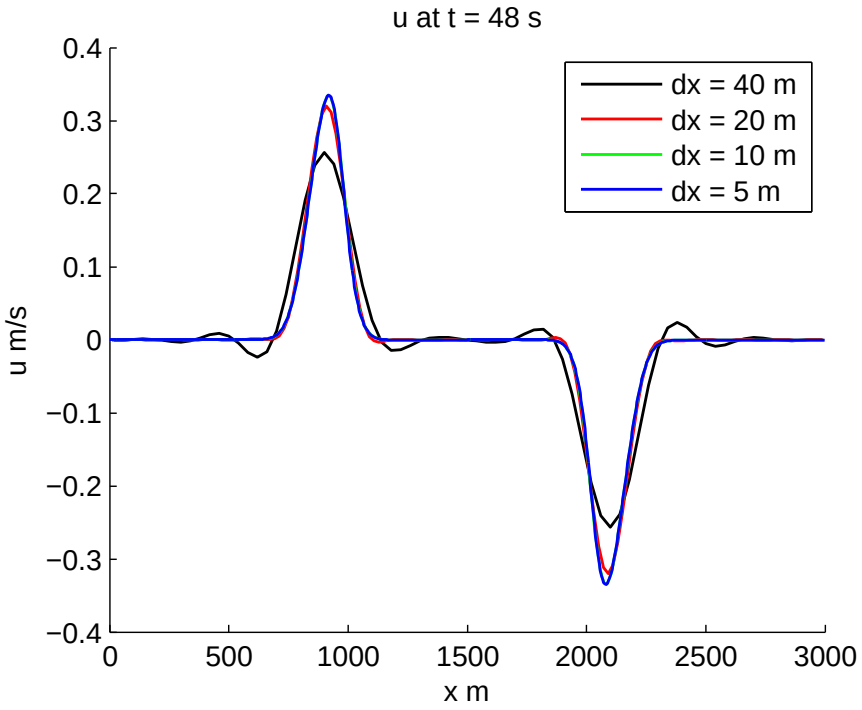

Figure A.2: Horizontal velocity at t = 48 s for the one-dimensional acoustic wave test at varying resolutions.

simulation, each grid point corresponds to the average value of a grid cell of width of dx, and for each coarsening step, since we doubled the grid spacing, we averaged the reference results to compare to these coarser resolutions, i. e. the result of cell size $dx = 10$ m was compared to the averaged value of 2 grid cells from the result with $dx = 5$ m, etc. The averaged reference result is aligned to the results from coarser resolutions, the $l2$ norm error is calculated using the difference between the results from the coarser resolution and the averaged reference result. The logarithm plot of dx vs. $error$ is shown in figure A.3.a. The scattered points are in good agreement with the solid line, which has the slope of 4, so the one-dimensional test of our scheme is 4th-order accurate.

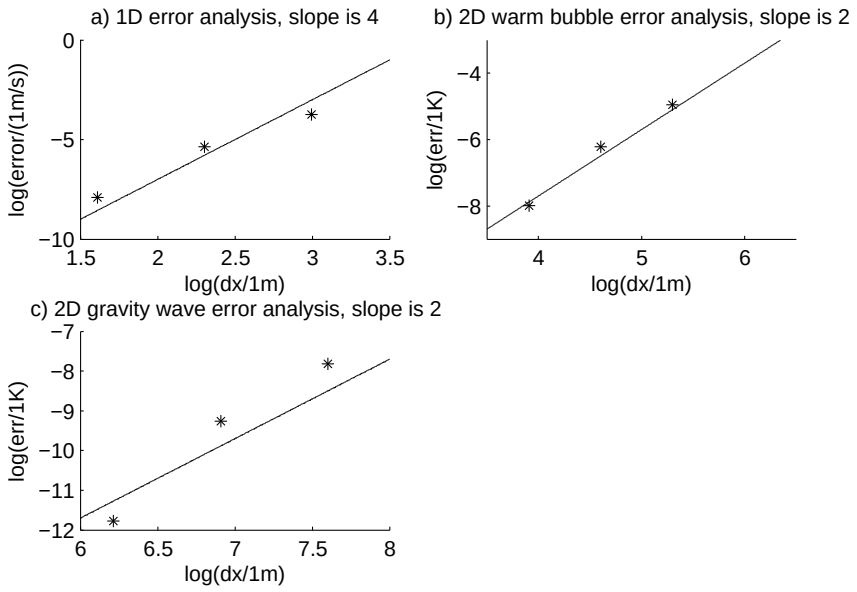

Figure A.3: L2 error analysis to determine the order of accuracy of each of the schemes. a) 1D acoustic wave test after 48 s, b) 2D Gaussian bubble test with Eulerian vertical coordinate after 6 min, c) 2D gravity wave test with Lagrangian vertical coordinate after 3000 s.

A.2.2 2D simulation error analysis

We performed the rising bubble test using different resolutions. The Gaussian bubble tests with the Eulerian vertical coordinate were selected for the error analysis, because the profiles of the perturbation are more continuous. The sampled results are taken at a finite time of 6 minutes. Similar to the 1D simulation error analysis, the results from different resolutions using $dx = dz = 25$ m, 50 m, 100 m, and 200 m were used, and the result from $dx = dz = 25$ m were used as the reference result. Analogously to the 1-dimensional test, the results from $dx = dz = 25$ m were averaged in order to compare them with results at different resolutions.

The $l2$ norm error is calculated using differences between the averaged reference result and results using coarser resolutions. The logarithm plot of dx vs. $error$ is shown in figure A.3.b. The scattered points are in good agreement with the solid line, which has the slope of 2, so we conclude our two-dimensional test is 2nd-order

accurate.

A.2.3 Error analysis with vertical Lagrangian coordinates and remapping

In the previous two sections we discussed the errors and order of accuracy of our model using a simple horizontal 1D and 2D Eulerian framework. In this section, the 2D Lagrangian framework together with remapping is examined. We used the gravity wave test described in section 2.4.2 of the paper. This analysis allows us to examine the accuracy of the Lagrangian framework for the transport of gravity waves. The gravity wave test with the Lagrangian vertical coordinate and a remap frequency of 60 seconds was selected for our error analysis. The sampled results are taken at a 3000 seconds. Similar to the warm bubble simulation error analysis, the results from different resolutions at $dx = dz = 500$ m, 1000 m, and 2000 m were used. Since we consider the Eulerian simulations as the reference simulations, the pure Eulerian simulation using a resolution of $dx = dz = 500$ m were used as the reference result. Other methods of analysis are similar to those described in the warm bubble simulation error analysis.

The logarithm plot of dx vs. $l2error$ is shown in figure A.3.c. The scattered points are in good agreement with the solid line, which has the slope of 2, so we conclude our 2-dimensional tests using the vertical Lagrangian coordinate and remapping is of 2nd-order accuracy.

A.3 Comparison of the Eulerian and Lagrangian formulations without viscosity

If the physical processes to be resolved do not involve the development of turbulence, the simulations of both Eulerian and Lagrangian formulations produce similar results. This is shown here by comparing the difference between early-time results

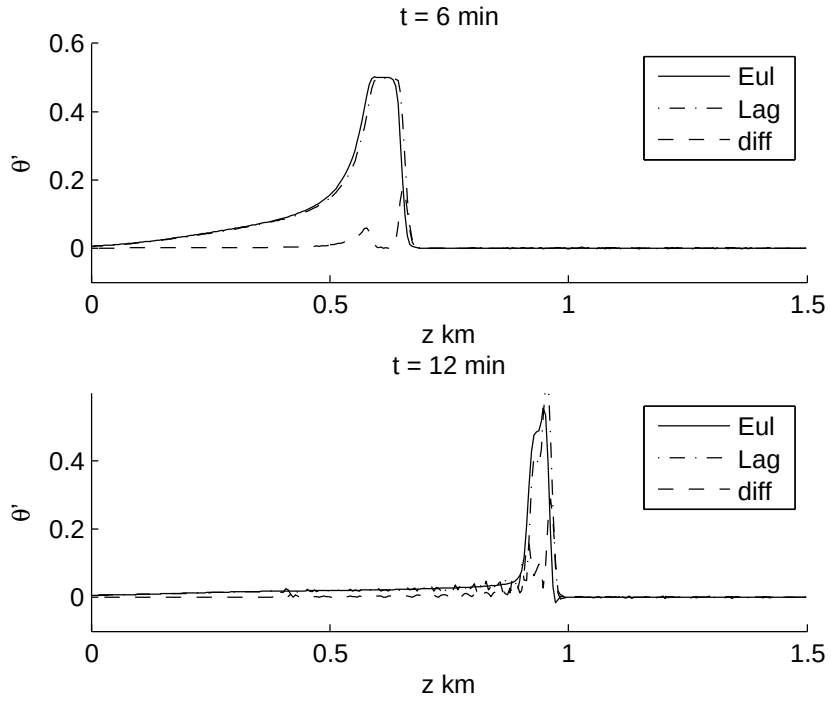

Figure A.4: Potential temperature perturbation in the Gaussian bubble test after a) 6 min and b) 12 min sampled along the central vertical line at $x = 497.5$ m. In the legend, diff means the difference between the Eulerian and Lagrangian results (absolute value).

from the rising bubble test described in section 2.4.1 of the paper and the gravity wave test described in section 2.4.2.

For the rising bubble test, we compare the plots of the Gaussian bubble at $t =$ 6 and 12 min using a resolution of 5 m. The value of the potential temperature perturbation θ' is sampled along the central vertical line (i.e. at $x = 497.5$ m). The plot of θ'_{Eul}, θ'_{Lag} and $abs(\theta'_{Lag} - \theta'_{Eul})$ is shown in Fig. A.4: The plot shows a slight shift between the Lagrangian bubble and the Eulerian bubble. However, the shape of the bubbles is almost identical as shown by the difference plot.

We conducted a similar examination of the gravity wave test, using a remap frequency of every 60 s for the Lagrangian formulation. The potential temperature is sampled along the central horizontal line (at $y = 5.5$ km) and at time $t = 3000$ s. The results are illustrated in Fig. A.5: Although the Lagrangian result is slightly more

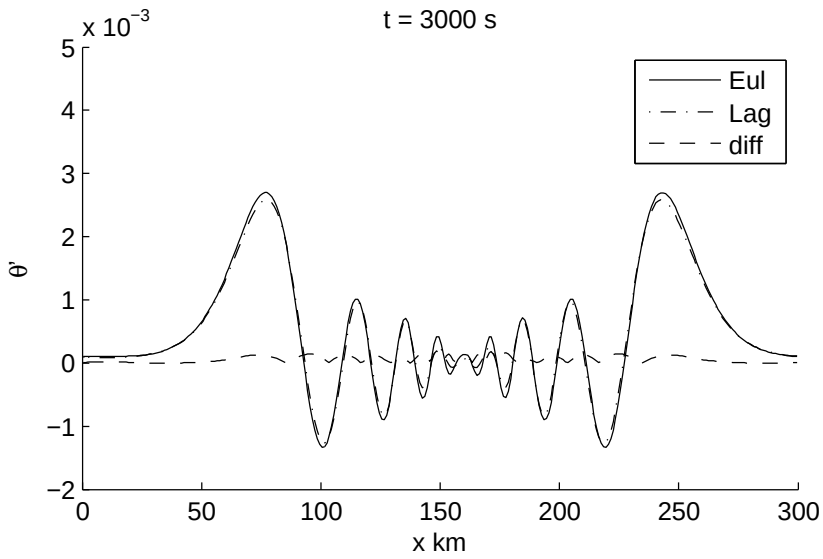

Figure A.5: Potential temperature perturbation in the gravity wave test after 3000 s sampled along the horizontal line at $x = 5.5$ km. In the legend, diff means the difference between the Eulerian and Lagrangian results (absolute value).

damped than the Eulerian result, the potential temperature perturbations calculated by both formulations are almost identical.

APPENDIX B

The Newton form of the polynomial Interpolation

Given a set of $n+1$ values y_i of a function $f(x)$ at $n+1$ distinct points x_i, where $i = 0, 1, ..., n$ and n is a positive integer. The function $f(x)$ could be approximated with a polynomial $p_n(x)$ of degree n constructed using polynomial interpolation that satisfies the conditions $f(x_i) = p_n(x_i), i = 0, 1, ..., n$. Newton form is an efficient polynomial interpolation scheme.

B.1 The Newton form of the polynomial interpolation scheme

The approximation polynomial $p_n(x)$ can be written in the Newton form

$$p_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + ... + c_n(x - x_0) \cdots (x - x_{n-1}) \quad \text{(B.1)}$$

where $c_0, c_1, ..., c_n$ are constants to be determined.

Below is the procedure to find the coefficients c_i. The requirement $p_n(x_i) = f(x_i)$

leads to

$$\begin{aligned} f(x_0) = p_n(x_0) &= c_0, \\ f(x_1) = p_n(x_1) &= c_0 + c_1(x_1 - x_0), \\ f(x_2) = p_n(x_2) &= c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1), \\ &\vdots \end{aligned} \tag{B.2}$$

Thus $c_0 = f(x_0)$, $c_1 = (f(x_1) - f(x_0))/(x_1 - x_0)$. Define the divided differences of order 0 as

$$f[x_i, x_i] = f(x_i) \tag{B.3}$$

and of order j as

$$f[x_i, x_{i+j}] = \frac{f[x_{i+1}, x_{i+j}] - f[x_i, x_{i+j-1}]}{x_{i+j} - x_i} \tag{B.4}$$

Then

$$c_i = f[x_0, x_i] \tag{B.5}$$

Given the value of c_i, Newton's form can be evaluated efficiently. Write $p_n(x)$ into a nested multiplication form:

$$p_n(x) = d_0(c_1 + d_1(c_2 + ... + d_{n-2}(c_{n-1} + d_{n-1}(c_n))...)) \tag{B.6}$$

where $d_0 = x - x_0$, $d_1 = x - x_1$,...,$d_{n-1} = x - x_{n-1}$, and introduce the notation:

$$\begin{aligned} u_n &= c_n; \\ u_{n-1} &= c_{n-1} + d_{n-1}u_n; \\ u_{n-2} &= c_{n-2} + d_{n-2}u_{n-1}; \\ &\vdots \\ u_1 &= c_1 + d_1u_2; \\ u_0 &= c_0 + d_0u_1; \end{aligned} \tag{B.7}$$

Then $p_n(x) = u_0$.

Newton's form of the polynomial interpolation is very efficient and the coefficients c_i can be reused when later there are more interpolation points available. For example, if later one more point $(x_{n+1}, f(x_{n+1}))$ is given, then $p_{n+1}(x) = p_n(x) + c_{n+1}(x - x_0)\cdots(x - x_n)$ gives $c_{n+1} = (f(x_{n+1} - p_n(x_{n+1}))/((x_{n+1} - x_0)\cdots(x_{n+1} - x_n))$.

BIBLIOGRAPHY

BIBLIOGRAPHY

Andronova, N. G., D. R. VandenBerg, Q. F. Stout, R. Oehmke, J. E. Penner, and V. Zubov (2010), Application of 3-d spherical shell adaptive mesh refinement to an atmospheric model with a vertical lagrangian coordinate, in *Proceedings of the 2010 Scientific Discovery through Advanced Computing (SciDAC) Conference*.

Arakawa, A., and V. R. Lamb (1977), *Methods of Computational Physics, volume 17.*, 173–265 pp., Academic Press, New York.

Bates, J., S. Moorthi, and R. Higgins (1993), A global multilevel atmospheric model using a vector semi-lagrangian finite-difference scheme. Part I: Adiabatic formulation, *Mon. Wea. Rev.*, *121*, 244–244.

Charney, J., R. Fjörtoft, and J. Neumann (1950), Numerical integration of the barotropic vorticity equation, *Tellus A*, *2*(4).

Chen, X., N. Andronova, B. V. Leer, J. E. Penner, J. Boyd, C. Jablonowski, and S.-J. Lin (2012), A control-volume model of the compressible Euler equations with a vertical Lagrangian Coordinate, *Mon. Wea. Rev.*, *Accepted*.

Colella, P., and P. R. Woodward (1984), The Piecewise Parabolic Method (PPM) for gas-dynamical simulations, *J. Comput. Phys.*, *54*(1), 174 – 201, doi:10.1016/0021-9991(84)90143-8.

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, *Mon. Wea. Rev.*, *126*(6), 1373–1395.

Daley, R. (1988), The normal modes of the spherical non-hydrostatic equations with applications to the filtering of acoustic modes, *Tellus A*, *40*(2), 96–106.

Donner, L. J., et al. (2011), The dynamical core, physical parameterizations, and basic simulation characteristics of the atmospheric component AM3 of the GFDL Global Coupled Model CM3, *J. Climate*, doi:doi:10.1175/2011JCLI3955.1.

Durran, D. R. (2010), *Numerical Methods for Fluid Dynamics: With Applications in Geophysics*, Springer.

Fox-Rabinovitz, M., G. Stenchikov, M. Suarez, and L. Takacs (1997), A finite-difference GCM dynamical core with a variable-resolution stretched grid, *Mon. Wea. Rev.*, *125*(11), 2943–2968

Giorgi, F., and L. Mearns (1999), Introduction to special section: Regional climate modeling revisited, *Journal of Geophysical Research*, *104*(D6), 6335–6352

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, *J. Comput. Phys.*, *227*, 3849–3877, doi:10.1016/j.jcp.2007.12.009.

Godunov, S. K. (1959), A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations, *Math. Sbornik*, *47*, 271–306.

Gramelsberger, G. (2009), Conceiving meteorology as the exact science of the atmosphere: Vilhelm bjerknes's paper of 1904 as a milestone, *Meteorologische Zeitschrift*, *18*(6), 669–673.

Harris, L., and D. Durran (2010), An idealized comparison of one-way and two-way grid nesting, *Mon. Wea. Rev.*, *138*(6), 2174–2187 1520–0493.

Hirsh, C. (2007), *Numerical Computation of Internal and External Flows: Fundamentals of Computational Fluid Dynamics*, 552 pp., Butterworth-Heinemann, Burlington.

Holton, J., and G. Hakim (2012), *An introduction to dynamic meteorology*, Academic press.

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, *Mon. Wea. Rev.*, *134*(12), 3691–3713, doi:10.1175/MWR3223.1.

Johnson, B. (2005), The semidirect aerosol effect: Comparison of a single-column model with large eddy simulation for marine stratocumulus, *Journal of Climate*, *18*(1), 119–130.

Juang, H. (1992), A spectral fully compressible nonhydrostatic mesoscale model in hydrostatic sigma coordinates: Formulation and preliminary results, *Meteorology and Atmospheric Physics*, *50*(1), 75–88.

Kasahara, A. (1974), Various vertical coordinate systems used for numerical weather prediction, *Mon. Wea. Rev.*, *102*(7), 509–522.

Khairoutdinov, M., D. Randall, and C. DeMott (2005), Simulations of the atmospheric general circulation using a cloud-resolving model as a superparameterization of physical processes, *J. Atmos. Sci.*, *62*(7), 2136–2154.

Khairoutdinov, M. F., and D. A. Randall (2001), A cloud resolving model as a cloud parameterization in the ncar community climate system model: Preliminary results, *Geophys. Res. Lett.*, *28*(18), 3617–3620, doi:10.1029/2001GL013552.

Klemp, J. B., W. C. Skamarock, and J. Dudhia (2007), Conservative split-explicit time integration methods for the compressible non-hydrostatic equations, *Mon. Wea. Rev.*, *135*, 2897–2913.

Krol, M., S. Houweling, B. Bregman, M. van den Broek, A. Segers, P. van Velthoven, W. Peters, F. Dentener, and P. Bergamaschi (2005), The two-way nested global chemistry-transport zoom model TM5: algorithm and applications, *Atmos. Chem. Phys*, *5*, 417–432.

Laprise, R. (1992), The Euler Equations of Motion with Hydrostatic Pressure as an Independent Variable, *Mon. Wea. Rev.*, *120*, 197–207.

Lee, S., and J. Penner (2010), Comparison of a global-climate model to a cloud-system resolving model for the long-term response of thin stratocumulus clouds to preindustrial and present-day aerosol conditions, *Atmos. Chem. Phys*, *10*, 6371–6389.

Lin, S.-J. (2004), A "vertically Lagrangian" finite-volume dynamical core for global models, *Mon. Wea. Rev.*, *132*, 2293–2307.

Lin, S.-J. (2007), Vertically Lagrangian control-volume discretization of the compressible Euler equations, unpublished manuscript.

Lin, S.-J., and R. B. Rood (1996), Multidimensional flux-form semi Lagrangian transport schemes, *Mon. Wea. Rev.*, *124*, 2046–2070.

Lin, S.-J., and R. B. Rood (1997), An explicit flux-form semi-Lagrangian shallow-water model on the sphere, *Q. J. R. Meteor. Soc.*, *123*, 2477–2498.

Liou, M.-S. (2006), A sequel to AUSM, Part II: AUSM$^+$-up for all speeds, *J. Comput. Phys.*, *214*, 137–170.

Manabe, S., J. Smagorinsky, and R. Strickler (1965), Simulated climatology of a general circulation model with a hydrologic cycle, *Mon. Wea. Rev.*, *93*(12), 769–798.

Mass, C., D. Ovens, K. Westrick, and B. Colle (2002), Does increasing horizontal resolution produce more skillful forecasts, *Bull. Amer. Meteor. Soc*, *83*(3), 407–430.

Miller, M. J., and R. P. Pearce (1974), A three-dimensional primitive equation model of cumulonimbus convection, *Quart. J. Roy. Meteor. Soc.*, *100*, 133–154.

Miller, M. J., and A. A. White (1984), On the non-hydrostatic equations in pressure and sigma coordinates, *Quart. J. Roy. Meteor. Soc.*, *110*, 515–533.

Neale, R. B., et al. (2010), Description of the NCAR Community Atmosphere Model (CAM 5.0), *NCAR Technical Note NCAR/TN-486+STR*, National Center for Atmospheric Research, Boulder, Colorado, 268 pp., available from http://www.cesm.ucar.edu/models/cesm1.0/cam/.

Oehmke, R. C. (2004), High performance dynamic array structures, Ph.D. thesis, University of Michigan, Ann Arbor, MI, USA.

Phillips, N. (1956), The general circulation of the atmosphere: A numerical experiment, *Quarterly Journal of the Royal Meteorological Society*, *82*(352), 123–164.

Prusa, J. M., and P. K. Smolarkiewicz (2003), An all-scale anelastic model for geophysical flows: dynamic grid deformation, *J. Comput. Phys.*, *190*(2), 601–622.

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, pp. 268–276, Astronomical Society of the Pacific Conference Series.

Richardson, L. (1922), *Weather prediction by numerical process*, University Press.

Robert, A. (1993), Bubble convection experiments with a semi-implicit formulation of the Euler equations, *J. Atmos. Sci.*, *50*(13), 1865–1865.

Rood, R. (1987), Numerical advection algorithms and their role in atmospheric transport and chemistry models, *Reviews of geophysics*, *25*(1), 71–100.

Skamarock, W. C. (2008), A linear analysis of the NCAR CCSM finite-volume dynamical core, *Mon. Wea. Rev.*, *136*, 2112–2119.

Skamarock, W. C., and J. B. Klemp (1992), The stability of time-split numerical methods for the hydrostatic and non-hydrostatic elastic equations, *Mon. Wea. Rev.*, *120*, 2109–2127.

Skamarock, W. C., and J. B. Klemp (1994), Efficiency and accuracy of the Klemp-Wilhelmson time-splitting technique, *Mon. Wea. Rev.*, *122*(11), 2623–2630.

Skamarock, W. C., and J. B. Klemp (2008), A time-split nonhydrostatic atmospheric model for weather research and forecasting applications, *J. Comput. Phys.*, *227*, 3465–3485, doi:10.1016/j.jcp.2007.01.037.

Smagorinsky, J. (1963), General circulation experiments with the primitive equations: I. The basic experiment, *Mon. Wea. Rev.*, *91*, 99–164.

Smolarkiewicz, P. K., L. G. Margolin, and A. A. Wyszogrodzki (2001), A class of non-hydrostatic global models, *J. Atmos. Sci.*, *58*, 349–364.

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, *Mon. Wea. Rev.*, *136*(6), 1898–1922.

Starr, V. P. (1945), A quasi-Lagrangian system of hydrodynamical equations, *2*, 227–237.

Thuburn, J. (2008), Some conservation issues for the dynamical cores of NWP and climate models, *J. Comput. Phys.*, *227*(7), 3715–3730.

Ullrich, P., and C. Jablonowski (2011), An analysis of 1d finite-volume methods for geophysical problems on refined grids, *J. Comput. Phys.*, *230*(3), 706–725

Ullrich, P. A., C. Jablonowski, and B. van Leer (2010), Riemann-solver-based high-order finite-volume models for the shallow-water equations on the sphere, *J. Comput. Phys.*, *229*, 6104–6134.

van Leer, B. (1974), Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme, *J. Comput. Phys.*, *14*(4), 361–370.

Van Leer, B. (1977), Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection, *J. Comput. Phys.*, *23*(3), 276–299.

van Leer, B. (1979), Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *J. Comput. Phys.*, *32*(1), 101–136.

van Leer, B., H. Cabannes, and R. Temam (1973), *Towards the ultimate conservative difference scheme I. The quest of monotonicity*, vol. 18, pp. 163–168, Springer Berlin / Heidelberg, doi:10.1007/BFb0118673.

Wang, M., S. Ghan, M. Ovchinnikov, X. Liu, R. Easter, E. Kassianov, Y. Qian, and H. Morrison (2011), Aerosol indirect effects in a multi-scale aerosol-climate model PNNL-MMF, *Atmospheric Chemistry and Physics*, *11*(11), 5431–5455, doi: 10.5194/acp-11-5431-2011.

Whitehead, J., C. Jablonowski, R. B. Rood, and P. H. Lauritzen (2011), A Stability Analsis of Divergence Damping on a Latitude-Longitude Grid, *Mon. Wea. Rev.*, *139*, 2976–2993, doi:http://dx.doi.org/10.1175/2011MWR3607.1.

Xue, M., and A. J. Thorpe (1991), A mesoscale numerical model using the non-hydrostatic pressure-based coordinate equations: Model experiments with dry mountain flows, *Mon. Wea. Rev.*, *119*, 1168–1185.

Yang, D. (2001), *C++ and Object-Oriented Numeric Computing for Scientists and Engineers*, Springer.

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, *Mon. Wea. Rev.*, *130*(2), 339–356.