ABSTRACT

NUMERICAL TECHNIQUES FOR IMPROVING SIMULATIONS OF TROPICAL CYCLONES

by
Yassine Tissaoui

The increasing frequency and intensity of tropical cyclones (TCs) due to climate change pose significant challenges for forecasting and mitigating their impacts. Despite advancements, accurately predicting TC rapid intensification (RI) remains a challenge. Large eddy simulation (LES) allows for explicitly resolving the large eddies involved in TC turbulence, thus providing an avenue for studying the mechanisms behind their intensification and RI. LES of a full tropical cyclone is very computationally expensive and its accuracy will depend on both explicit and implicit dissipation within an atmospheric model. This dissertation presents two novel numerical methodologies with the potential to improve the efficiency of tropical cyclone LES in the future. The first is a pioneering non-column based implementation of the Kessler warm rain microphysics parametrization, a method which would allow for the use of three-dimensional (3D) adaptive mesh refinement (AMR) in the simulation of moist tropical cyclones. The second is an implementation of Laguerre-Legendre semi-infinite elements for use in the damping layers of atmospheric models, a method which was shown to be capable of improving the efficiency of benchmark atmospheric simulations. Finally, the dissertation presents a study of two-dimensional (2D) AMR applied to simulations of a rapidly intensifying dry tropical cyclone and shows that AMR is able to accurately reproduce the results of simulations using static grids while demonstrating considerable cost savings.

NUMERICAL TECHNIQUES FOR IMPROVING SIMULATIONS OF TROPICAL CYCLONES

by
Yassine Tissaoui

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mechanical Engineering

Department of Mechanical and Industrial Engineering

August 2024

APPROVAL PAGE

NUMERICAL TECHNIQUES FOR IMPROVING SIMULATIONS OF TROPICAL CYCLONES

Yassine Tissaoui

Dr. Simone Marras, Dissertation Co-Advisor — Date
Assistant Professor of Mechanical and Industrial Engineering, NJIT

Dr. Stephen R. Guimond, Dissertation Co-Advisor — Date
Associate Professor of Atmospheric and Planetary Sciences,
Hampton University, Hampton Virginia

Dr. Shawn Chester, Committee Member — Date
Associate Professor of Mechanical and Industrial Engineering, NJIT

Dr. Samaneh Farokhirad, Committee Member — Date
Assistant Professor of Mechanical and Industrial Engineering, NJIT

Dr. David G. Shirokoff, Committee Member — Date
Associate Professor of Mathematics, NJIT

BIOGRAPHICAL SKETCH

Author: Yassine Tissaoui

Degree: Doctor of Philosophy

Date: August 2024

Undergraduate and Graduate Education:

- Doctor of Philosophy in Mechanical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2024
- Master of Science in Modeling for Industry and Services,
  National Engineering School of Tunis, Tunisia, 2018
- Bachelor of Science in Modeling for Industry and Services,
  National Engineering School of Tunis, Tunisia, 2018

Major: Mechanical Engineering

Presentations and Publications:

A. Sridhar, Y. Tissaoui, S. Marras, Z. Shen, C. Kawczynski, S. Byrne, K. Pamnany, M. Waruszewski, T. H. Gibson, J. E. Kozdon, V. Churavy, L. C. Wilcox, F. X. Giraldo, T. Schneider, "Large-eddy simulations with ClimateMachine v0.2.0: a new open-source code for atmospheric simulations on GPUs and CPUs," *Geoscientific Model Development*, p 335, 2022.

Y. Tissaoui, S. Marras, A. Quaini, F. A. V. De Bragança Alves, and F. X. Giraldo, "A non-column based, fully unstructured implementation of Kessler microphysics with warm rain using continuous and discontinuous spectral elements," *Journal of Advances in Modeling Earth Systems*, 2023.

Y. Tissaoui, J. F. Kelly, and S. Marras, "Efficient Spectral Element Method for the Euler Equations on Unbounded Domains in Multiple Dimensions" *Journal of Applied Mathematics and Computation*, 2024, in review.

Reddy, S., Tissaoui, Y., F. A. V. De Bragança Alves, Marras, S., Giraldo, F. X., "Comparison of Sub-Grid Scale Models for Large-Eddy Simulation using a High-Order Spectral Element Approximation of the Compressible Navier-Stokes Equations at Low Mach Number" *Arxiv*, 2021, preprint.

Y. Tissaoui, S. R. Guimond, S. Marras, and F. X. Giraldo, "Accelerating Simulations of Tropical Cyclones using Adaptive Mesh Refinement", 2024, in preparation.

To climbing, a sport where falling down and getting back up is expected, encouraged and also necessary (how else are we supposed to get off the wall). Putting this together was not unlike solving a climbing problem, occasionally very frustrating and resulting in a few mental scrapes and bruises. I still love climbing and I love science, the challenge was always part of the appeal.

To my wife Emna, you're the only reason I haven't lost my mind yet.

ACKNOWLEDGMENTS

I would like thank my dissertation advisor, Dr. Simone Marras for his guidance, support and patience throughout the years it took to complete this thesis. I would also like to thank him for the enriching discussions we've had throughout the years. I would also like to thank Dr. Stephen Guimond my co-advisor for his insight and guidance, this work would not have been possible without his unique expertise.

I would like to express my gratitude to my PhD thesis committee members Dr. Shawn Chester, Dr. Samaneh Farokhirad, and Dr. David G. Shirokoff. Thank you for your time and effort.

I would also like to thank Dr. Shawn Chester and Dr. Vejlko Samardzic for their advice and help throughout my first years at NJIT. I would like to thank Dr. Abhishek Mukherjee for showing me the ropes when I first started the program and for always offering reliable support whenever it was needed and I would like to thank Dr. Keven Alkhoury at NJIT for the many fruitful discussions we've had as colleagues and for providing this thesis template.

I would like to thank Dr. Tapio Schneider, Dr. Akshay Sridhar, and the Climate Modeling Alliance team for their mentorship, help, and support. I would like to express my sincere gratitude to Dr. Francis X. Giraldo and Dr. James Kelly for their support and for their help the many times I needed their assistance with technical problems.

I would like to acknowledge the partial support by the National Science Foundation through Grant PD-2121367.

I want to express my deepest gratitude to my wife, Emna Sahraoui for always believing in me, for always making my day brighter, for never letting me give up on myself and for reminding me that I'm human and that I too need breaks. I would like

to thank my sibling, Yassmine Tissaoui for always providing interesting conversation and for being excited about the work I do even when I'm in a rut. Finally I would like to thank my parents, Luz Tissaoui and Jassem Tissaoui, I wouldn't be here without their emotional support and them indulging my curiosity and my interest in science while I was child.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

It is a matter of fact that the climate is warming and it is expected that this will lead to an increase in the number of highly intense weather events across the globe. An example of such events are tropical cyclones for which the most intense season on record was in 2020 only a few years ago. Even with modern forecasting systems recent examples such as Hurricane Harvey in 2017, Hurricane Micheal in 2018 and Hurricane Ian in 2022 show the devastation that tropical cyclones can leave in their wake. The warming climate makes tropical cyclones more likely to occur, more likely to cause more intense flooding rainfall, and more likely to progress deeper inland which can cause further devastation through the appearance of tornadoes even after the tropical cyclone itself clears [67, 58, 57].

With these potential disasters becoming more probable it is important to be appropriately prepared and to respond to incoming tropical cyclones as effectively as possible. Preparation and timely evacuation has the potential to save many lives and prevent billions of dollars in damages. To this end, accurate forecasts of tropical cyclone track, storm surge, intensity and rainfall is of the utmost importance. One of the most significant issues with current and past operational weather forecast models is the ability to predict tropical cyclone rapid intensification (RI). This area of research had been stagnant for a significant length of time [69, 89], and according to DeMaria et al., (2014) [18] the ability to forecast RI was completely absent from deterministic operational models up to around 2015. Current models do present the ability to forecast rapid intensification, but this phenomenon still remains under-predicted [17].

Tropical cyclone intensification and RI, depend not only on large scale environmental factors such as high sea surface temperatures, but also on a multitude of dynamical and physical processes occurring over a range of scales extending from tens of kilometers to a few meters. Intensification is also highly dependent on the latent heating generated from the phase changes occurring in the moist atmosphere. Tropical cyclones are affected by vortex Rossby waves [32, 36, 108, 80], barotropic instabilities along the eyewall and the turbulence caused by them [94, 62, 39, 31, 16], boundary-layer roll vortices [35, 22], air-sea interaction [114, 20], eyewall replacement cycles [113, 46] and deep convective bursts [90, 77, 38, 32]. Three dimensional deep rotating convection in tropical cyclone intensification has drawn special interest from the tropical cyclone modeling community [33, 78, 84, 86, 88, 79], with some recent studies highlighting its effect on intensification[88, 33, 78]. Tropical cyclone intensity is also affected by planetary boundary layer (PBL) turbulence. While energy can be dissipated by surface friction and turbulent eddies in the PBL, recent work has shown that energy can also be "backscattered" through coherent turbulent structures in the hurricane boundary layer [104]. Furthermore, turbulence in the eyewall region has been shown to significantly impact intensification [88, 91, 21, 12]. Most simulation of tropical cyclones are done with horizontal resolutions around $1-2$ km which makes these simulations highly sensitive to the turbulence parametrizations being used and can lead to potential inaccuracies [42, 10].

The simplest approach to avoid relying on parametrizations is to increase the span of scales being explicitly resolved by the numerical model. To this effect, performing high-order large eddy simulation (LES) of TCs with grids of around 100 m resolution would allow for explicitly resolving the large eddies involved in TC turbulence and thus explicitly take into account more of the dynamics involved [11, 64]. These types of simulations are very computationally demanding particularly

when done at resolutions less than 100 m and when they have been performed in the past it's only over short periods of time, with some groups performing LES on the entire inner-core of a TC (50 km radius or larger) [91, 47, 34].

Two main challenges come to mind when thinking of simulating an entire tropical cyclone domain at LES resolutions: The expense of performing the simulation, and having its results be as accurate as possible. This work focuses on studying and developing numerical methods that can improve the efficiency of atmospheric models in general, but with a special interest given to their applicability in speeding up the simulations of hurricanes.

Simulations of tropical cyclones have been shown to be sensitive to the implicit numerical dissipation (IND) of the models being used [29, 37]. This dissipation primarily stems from the order of the discretization being used and causes simulations of tropical cyclones to exhibit weaker responses to asymmetric heating perturbations when compared to less dissipative models. As such, this work uses high order Galerkin spectral elements methods to discretize the compressible Euler equations in space. These methods' ability to extend to arbitrarily high-order while possessing minimal dissipation and dispersion errors, and their scalability and efficiency on massively parallel architectures [2, 1, 3, 71] make them a suitable testing bed for developing numerical techniques for tropical cyclone simulations.

Adaptive mesh refinement (AMR) is a powerful tool with the potential to improve the time to solution for otherwise much more costly simulations. AMR also allows for obtaining the desired resolution where it is required by making the computational grid finer around specific flow features characterized by predefined criteria. AMR in atmospheric modeling has existed since the seminal works of Berger and Oliger [8], Skarmarock et al., (1989) [99] and Skamarock and Klemp (1993) [98] proposed adaptive meshes for hyperbolic equations with applications

to atmospheric problems. Bacon et al., (2000) [4] developed the first operational model that used horizontally adaptive meshes and used to successfully simulate tropical cyclone storm tracks. AMR has been shown to be effective for simulations of the shallow water equations [7, 65, 76] and AMR has been tested with Galerkin methods for meteorological applications [81, 60, 15]. Recent advances with AMR have allowed for improving and unstructured meshes have allowed for improvements in modeling atmospheric flows around topography [115, 66]. In terms of tropical cyclone simulations Hendricks et al., (2016) [40] demonstrated that idealized simulations of tropical cyclones could be sped up by $4 - 15$ times using adaptive mesh refinement with a shallow water spectral elements model.

AMR would make it possible to simulate a large tropical cyclone domain at lower costs while obtaining LES resolution in regions of interest. To this effect, it would be advantageous to be able simulate tropical cyclones with three dimensional AMR which allows for both the horizontal and vertical resolutions in different areas of the domain to adapt over time. However, a challenge that prevents us from being able to do this is that more realistic tropical cyclone simulations require a microphysical parametrization for moisture and these parametrizations require the presence of vertically structured grids with coherent column data structures whereas 3D AMR would generate vertically non-conforming elements and make a column data-structure impossible on the native dynamics grid. These parametrizations have relied on grids with column-based structures for their implementation since the first microphysical parametrizations were utilized for early simulations of clouds and precipitation in the 1960s and 1970s [53, 102, 52, 109]. As part of this dissertation the first non-column based implementation of Kessler microphysics was developed. This novel approach which was first introduced by the author and collaborators in [105]. This approach

allows for solving moist atmospheric problems on unstructured grids and was tested on thunderstorm problems.

While not directly tied to AMR, another challenge to the efficiency of atmospheric simulations in general and tropical cyclone simulations in particular is the need for damping layers to prevent the reflection of outgoing waves (gravity waves especially). This approach involves augmenting the computational domain by adding absorbing layer designed to damp these outgoing waves towards a reference state [19, 55, 54, 63]. This presents a significant drawback in terms of computational efficiency since it enlarges the computational domain outside the region of interest for a given simulation. This cost is further compounded in large multidimensional high-resolution problems. As a cost-effective approach to Rayleigh damping, Benacchio and Bonaventura [5] introduced a scaled Laguerre spectral collocation approach to prevent the reflection one-dimensional(1D) shallow water waves. This approach was extended to 1D discontinuous Galerkin spectral elements by [6], then to two dimensions for the advection-diffusion equation in [107] and to general hyperbolic problems in [106]. As part of this dissertation, this approach is extended to the compressible Euler equations using continuous Galerkin spectral elements on a Laguerre-Legendre basis and demonstrates the ability of this methodology to be cost effective for benchmark atmospheric problems.

Finally, a study of the application of 2D adaptive mesh refinement to a dry but rapidly intensifying tropical cyclone is presented in this work. 2D AMR is shown to be able to reproduce the results of much more costly static grid simulations at a fraction of the computational cost. AMR is also demonstrated to be able to effectively refine around areas of interest in tropical cyclone simulations and detect important flow features like large intense updrafts which could play a role in intensification.

The remainder of this dissertation is organized as follows: Chapter 2 describes the continuous and discontinuous Galerkin methods as well the Laguerre-Legendre semi-infinite element approach. Chapter 3 describes the different atmospheric models used in this work and highlights the differences between them. Chapter 4 shows the potential improvements in efficiency from using Laguerre-Legendre semi-infinite elements for the damping layers of atmospheric flows. Chapter 5 showcases a novel method for the simulation of precipitating atmospheres without the need for column based data structures and presents its results for simulations of squall lines and supercells. Chapter 6 presents adaptive mesh refinement and highlights its ability to reproduce accurate simulations of hurricanes at a fraction of the cost. This is followed by a conclusion. Appendix A presents the pseudo code to compute the element right-hand side on semi-infinite elements, Appendix B provides a description of how to extend the Laguerre-Legendre semi-infinite element method to three dimensions, and Appendix C presents two soundings, the first of which is used for the tropical cyclone simulations and the second of which is used for the squall line and supercell simulations.

CHAPTER 2

SPECTRAL ELEMENTS METHODS

The work being presented here relies heavily on the spectral elements method for the spatial discretization of the governing equations. As such, this chapter presents the reader with a brief overview of the method and outlines how to derive the spectral element approximation for a given system of partial differential equations (PDEs).

To make the description of the numerical method easy to follow, let us consider a generic equation of the form:

$$\frac{\partial f}{\partial t} + G(f) = 0, \tag{2.1}$$

where f is the unknown variable and G is a linear functional that may contain first and second derivatives of f. If the equations to be solved are written in conservation form, then G is the divergence of a flux.

2.1 The Spectral Elements Method on Finite Domains

Let us subdivide the domain Ω into a set of conforming [1] N_e hexahedral elements Ω_e of arbitrary orientation to create the discrete domain Ω^h as

$$\Omega \approx \Omega^h = \bigcup_{e=1}^{N_e} \Omega_e. \tag{2.2}$$

Figure 2.1 shows examples of a structured and unstructured grid in 2D. Using a fully unstructured grid means that structures such as the rows or columns that are seen on the left side of Figure 2.1 are no longer present. Let us define the reference

[1]The condition of conformity is not strictly necessary, although it simplifies the discussion of the method. For results with non-conforming grids, the reader is referred to, e.g., [59].

element Ω_{ref}: $(\xi, \eta) \in [-1, 1]^2$ in 2D and $(\xi, \eta, \zeta) \in [-1, 1]^3$ in 3D. Regardless of whether the mesh is structured or unstructured, let us introduce a mapping from a generic element in the global system of coordinates, i.e., (x, y) in 2D and (x, y, z) in 3D, to the reference element. Let $\mathbf{J}$ be the Jacobian matrix of this mapping. The construction of the Jacobian for a 3D mapping is given in Equation (2.3)

$$\mathbf{J}(\xi) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \tag{2.3}$$

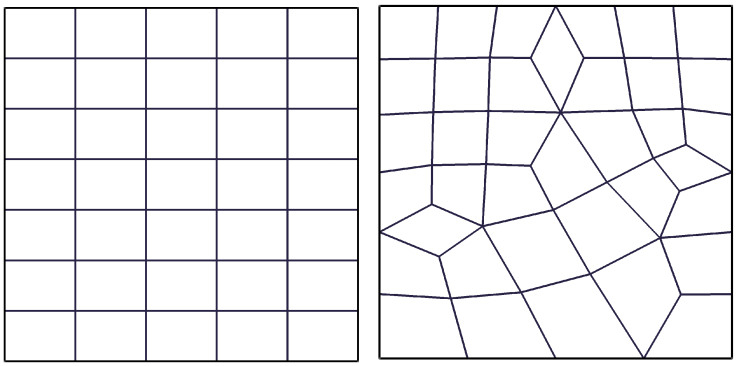


Figure 2.1 Examples of a structured (left) and an unstructured grid (right) made of quadrilateral elements.

Let h_i, $i = 1, \ldots, N + 1$, be the Lagrange polynomials of degree N:

$$h_i(\xi) = \frac{1}{N(N+1)} \frac{(1 - \xi_i^2) P_N'(\xi)}{(\xi - \xi_i) P_N(\xi)},$$

where P_N is the Legendre polynomial of order N, and P_N' its derivative evaluated at the point ξ. The polynomials in multiple dimensions are built via a tensor product of the 1D bases, as shown below. The remainder of this section is written for a 3D case.

For every element, we seek an approximation f^h of a variable f of the form:

$$f^h(\boldsymbol{\xi}, t) = \sum_{l=1}^{(N+1)^3} \psi_l(\boldsymbol{\xi}) \hat{f}_l(t), \tag{2.4}$$

where $\boldsymbol{\xi} = (\xi, \eta, \zeta)$, $\hat{f}_l$ are the expansion coefficients, and ψ_l are nodal basis functions defined as tensor products of the Lagrange polynomials

$$\psi_l = h_i[\xi(\mathbf{x})] \otimes h_j[\eta(\mathbf{x})] \otimes h_k[\zeta(\mathbf{x})], \quad l = i + 1 + j(N+1) + k(N+1)(N+1), \tag{2.5}$$

where $\mathbf{x} = (x, y, z)$. The Legendre-Gauss-Lobatto (LGL) points are used as the interpolation points on each element. LGL points are not equidistant and represent the solutions of the following equation:

$$(1 - \xi^2) P_N^{'}(\xi) = 0,$$

where $P_N(x)$ is the Legendre polynomial of order N. These polynomials and their first derivative are computed recursively via

$$P_0(\xi) = 1 \tag{2.6a}$$

$$P_1(\xi) = \xi \tag{2.6b}$$

$$P_k(\xi) = \frac{2k-1}{k} \xi P_{k-1}(\xi) - \frac{k-1}{k} P_{k-2}(\xi), \qquad \forall k \geq 2 \tag{2.6c}$$

$$P_k'(\xi) = (2k-1) P_{k-1}(\xi) + P_{k-2}'(\xi). \tag{2.6d}$$

See, e.g., [26, 50]. Finally, the corresponding quadrature weights are given by:

$$\omega(\xi_i) = \frac{2}{N(N+1)} \left[\frac{1}{P_N(\xi_i)} \right]^2$$

The weights are used to approximate the integrals with a Gauss quadrature rule of accuracy $\mathcal{O}(2N-1)$. Over a generic element Ω_e, this is done as follows:

$$\int_{\Omega_e} f(\mathbf{x})d\mathbf{x} = \int_{\Omega_{ref}} f(\boldsymbol{\xi})|\mathbf{J}(\boldsymbol{\xi})|d\boldsymbol{\xi} \approx \sum_{i,j,k=1}^{N+1} \omega(\xi_i)\omega(\eta_j)\omega(\zeta_k)f(\xi_i,\eta_j,\zeta_k)|\mathbf{J}(\xi_i,\eta_j,\zeta_k)|, \tag{2.7}$$

where $|\mathbf{J}|$ is the determinant of the Jacobian matrix.

To approximate the solution of Equation (2.1), let $(\cdot,\cdot)$ be the Legendre inner product on a given element Ω_e:

$$(f,g)_e = \int_{\Omega_e} f(\mathbf{x})g(\mathbf{x})d\mathbf{x}.$$

If in Equation (2.1) we replace f with f^h as defined in Equation (2.4), we will obtain the following residual:

$$R = \frac{\partial f^h}{\partial t} + G(f^h), \tag{2.8}$$

which is orthogonal to the expansion functions in Galerkin methods, i.e.,:

$$(R,\psi_k)_e = 0,\ \ k = 1,\ldots,(N+1)^3. \tag{2.9}$$

Taking Equation (2.9) into account, let us now write an approximation of Equation (2.1) on each element Ω_e as follows :

$$\int_{\Omega_e} \psi_i(\mathbf{x})\frac{\partial f^h(\mathbf{x},t)}{\partial t}d\mathbf{x} = -\int_{\Omega_e} \psi_i(\mathbf{x})G(f^h(\mathbf{x},t))d\mathbf{x},\ \ i = 1,\ldots,(N+1)^3. \tag{2.10}$$

Let us first consider the case where $G(f) = \boldsymbol{\nabla}\cdot\mathbf{f}$, where $\boldsymbol{\nabla} = \left(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z}\right)$ and $\mathbf{f} = (f,f,f)$.

We can use the polynomial expansion to write Equation (2.10) as follows:

$$\int_{\Omega_e} \psi_i(\mathbf{x}) \sum_{j=1}^{(N+1)^3} \psi_j(\mathbf{x})\frac{\partial \hat{f}_j^e(t)}{\partial t}d\mathbf{x} = -\int_{\Omega e} \psi_i(\mathbf{x}) \sum_{j=1}^{(N+1)^3} \boldsymbol{\nabla}\psi_j(\mathbf{x})\cdot\hat{\mathbf{f}}_\mathbf{j}^\mathbf{e}(\mathbf{t})d\mathbf{x},\ \ i = 1,\ldots,(N+1)^3, \tag{2.11}$$

where the superscript e is used to denote that the expansion is defined on an element basis and $\hat{\mathbf{f}}^{\mathbf{e}}_{\mathbf{j}}(\mathbf{t}) = (\hat{f}^e_j(t), \hat{f}^e_j(t), \hat{f}^e_j(t))$. We can now define the mass matrix $\mathbf{M}^e$ and the differentiation matrix $\mathbf{D}^e$ on each element through their components:

$$M^e_{ij} = \int_{\Omega_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x})d\mathbf{x} = \int_{\Omega_{ref}} \psi_i(\boldsymbol{\xi})\psi_j(\boldsymbol{\xi})|\mathbf{J}(\boldsymbol{\xi})|d\boldsymbol{\xi}, \tag{2.12}$$

$$\mathbf{D}^e_{ij} = \int_{\Omega_e} \psi_i(\mathbf{x})\boldsymbol{\nabla}\psi_j(\mathbf{x})d\mathbf{x} = \int_{\Omega_{ref}} \psi_i(\boldsymbol{\xi}) \left(\boldsymbol{\nabla}_{\boldsymbol{\xi}}\psi_j(\boldsymbol{\xi})\mathbf{J}^{-1}(\boldsymbol{\xi})\right) |\mathbf{J}(\boldsymbol{\xi})|d\boldsymbol{\xi}, \tag{2.13}$$

with $i, j = 1, \ldots, (N+1)^3$ and $\boldsymbol{\nabla}_{\boldsymbol{\xi}} = \left(\frac{\partial}{\partial\xi}, \frac{\partial}{\partial\eta}, \frac{\partial}{\partial\zeta}\right)$. By approximating the integrals with a quadrature rule, we obtain:

$$M^e_{ij} \approx \sum_{k=1}^{N+1}\sum_{m=1}^{N+1}\sum_{n=1}^{N+1} \omega(\xi_k, \eta_m, \zeta_n)\psi_i(\xi_k, \eta_m, \zeta_n)\psi_j(\xi_k, \eta_m, \zeta_n)|\mathbf{J}(\xi_k, \eta_m, \zeta_n)|, \tag{2.14}$$

$$\mathbf{D}^e_{ij} \approx \sum_{k=1}^{N+1}\sum_{m=1}^{N+1}\sum_{n=1}^{N+1} \omega(\xi_k, \eta_m, \zeta_n)\psi_i(\xi_k, \eta_m, \zeta_n)\boldsymbol{\nabla}\psi_j(\xi_k, \eta_m, \zeta_n)|\mathbf{J}(\xi_k, \eta_m, \zeta_n)|. \tag{2.15}$$

Remark 2.1.1 *The difference in notation between M^e_{ij} and D^e_{ij} is due to the components of the mass matrix being scalars while those of the differentiation matrix are vectors.*

Note that $\boldsymbol{\nabla}\psi_j(\xi_k, \eta_m, \zeta_n) = \boldsymbol{\nabla}_{\boldsymbol{\xi}}\psi_j(\xi_k, \eta_m, \zeta_n)\mathbf{J}^{-1}(\xi_k, \eta_m, \zeta_n)$. Then, the matrix form of Equation (2.11) is:

$$M^e_{ij}\frac{\partial \hat{f}^e_j(t)}{\partial t} = -\mathbf{D}_{ij} \cdot \hat{\mathbf{f}}^e_j(t), \;\; i, j = 1, \ldots, (N+1)^3. \tag{2.16}$$

Let us now consider $G(f) = \boldsymbol{\nabla}\cdot\mathbf{f} - \boldsymbol{\nabla}^2 f$ in Equation (2.1), where $\boldsymbol{\nabla}^2 = \boldsymbol{\nabla}\cdot\boldsymbol{\nabla}$. In this case, Equation (2.10) becomes:

$$\begin{aligned}\int_{\Omega_e} \psi_i(\mathbf{x}) \sum_{j=1}^{(N+1)^3} \psi_j(\mathbf{x})\frac{\partial \hat{f}^e_j(t)}{\partial t}d\mathbf{x} = &- \int_{\Omega e} \psi_i(\mathbf{x}) \sum_{j=1}^{(N+1)^3} \boldsymbol{\nabla}\psi_j(\mathbf{x}) \cdot \hat{\mathbf{f}}^{\mathbf{e}}_{\mathbf{j}}(\mathbf{t})d\mathbf{x} \\ &+ \int_{\Omega_e} \psi_i \boldsymbol{\nabla}\cdot \left[\sum_{j=1}^{(N+1)^3} \boldsymbol{\nabla}\psi_j(\mathbf{x})\hat{f}^e_j(t)\right] d\mathbf{x},\end{aligned} \tag{2.17}$$

where $i, j = 1, \ldots, (N+1)^3$. After integrating by parts the second term on the right-hand side, we can rewrite Equation (2.17) as:

$$M_{ij}^e \frac{\partial \hat{f}_j^e(t)}{\partial t} = -\mathbf{D}_{ij}^e \cdot \hat{\mathbf{f}}_j^e(t) + \left[\psi_i \mathbf{n}^e(\mathbf{x}) \cdot \sum_{j=1}^{N+1} \boldsymbol{\nabla} \psi_j(\mathbf{x}) \hat{f}_j^e(t) \right]_{\Gamma_e} - \int_{\Omega_e} \boldsymbol{\nabla} \psi_i(\mathbf{x}) \cdot \sum_{j=1}^{N+1} \boldsymbol{\nabla} \psi_j(\mathbf{x}) \hat{f}_j^e(t) d\Omega_e \;\; i, j = 1, \ldots, (N+1)^3, \quad (2.18)$$

where Γ_e represents the element boundary and $\mathbf{n^e}$ is the boundary normal vector. For the sake of brevity, let us assume that the boundary term, i.e., the second term on the right-hand side in Equation (2.18), vanishes at all element boundaries. We refer the reader to, e.g., [26] and [51] for a detailed explanation of how this term is handled when it is not zero, as is the case for DG. Under the assumption of vanishing boundary terms, Equation (2.18) becomes:

$$M_{ij}^e \frac{\partial \hat{f}_j^e(t)}{\partial t} = -\mathbf{D}_{ij}^e \cdot \hat{\mathbf{f}}_j^e(t) - \int_{\Omega_e} \boldsymbol{\nabla} \psi_i(\mathbf{x}) \cdot \boldsymbol{\nabla} \psi_j(\mathbf{x}) d\mathbf{x} \hat{f}_j^e, \;\; i, j = 1, \ldots, (N+1)^3. \quad (2.19)$$

We define the Laplacian element matrix $\mathbf{L^e}$ through its components as follows:

$$L_{ij}^e = \int_{\Omega_e} \boldsymbol{\nabla} \psi_i(\mathbf{x}) \cdot \boldsymbol{\nabla} \psi_j(\mathbf{x}) d\mathbf{x} \equiv \int_{\Omega_{ref}} (\boldsymbol{\nabla}_\xi \psi_i(\boldsymbol{\xi}) \mathbf{J}^{-1}(\boldsymbol{\xi})) \cdot (\boldsymbol{\nabla}_\xi \psi_j(\boldsymbol{\xi}) \mathbf{J}^{-1}(\boldsymbol{\xi})) |\mathbf{J}(\boldsymbol{\xi})| d\boldsymbol{\xi} \quad (2.20)$$

$$\equiv \int_{\Omega_{ref}} \mathbf{J}^{-T}(\boldsymbol{\xi}) \left(\boldsymbol{\nabla}_\xi \psi_i(\boldsymbol{\xi}) \right)^T \left(\boldsymbol{\nabla}_\xi \psi_j(\boldsymbol{\xi}) \right) \mathbf{J}^{-1}(\boldsymbol{\xi}) |\mathbf{J}(\boldsymbol{\xi})| d\boldsymbol{\xi},$$

where the superscript T denotes the transpose operator and $i, j = 1, \ldots, (N+1)^3$. By approximating the integral in Equation (2.20) with a quadrature rule, we obtain:

$$L_{ij}^e \approx \sum_{k=1}^{N+1} \sum_{m=1}^{N+1} \sum_{n=1}^{N+1} \omega(\xi_k, \eta_m, \zeta_n) \boldsymbol{\nabla} \psi_i(\xi_k, \eta_m, \zeta_n) \cdot \boldsymbol{\nabla} \psi_j(\xi_k, \eta_m, \zeta_n) |\mathbf{J}(\xi_k, \eta_m, \zeta_n)|. \quad (2.21)$$

Then, we write Equation (2.19) as:

$$M_{ij}^e \frac{\partial \hat{f}_j^e(t)}{\partial t} = -D_{ij}^e \cdot \hat{\mathbf{f}}_j^e(t) - L_{ij}^e \hat{f}_j^e(t) \ , i,j = 1,\ldots,(N+1)^3. \tag{2.22}$$

Next, we present briefly how the global solution is calculated depending on the choice of continuous Galerkin (CG) or discontinuous Galerkin (DG) spectral elements. The reader interested in more details on Galerkin spectral element methods is referred to, e.g., [26, 41, 61, 97].

2.1.1 Continuous Galerkin

Let $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{L}$ be the global mass matrix, global differentiation matrix, and global Laplacian matrix. These matrices are, assembled using Direct Stiffness Summation (DSS):

$$\mathbf{M} = \bigwedge_{e=1}^{N_e} \mathbf{M}^e, \ \mathbf{D} = \bigwedge_{e=1}^{N_e} \mathbf{D}^e, \ \mathbf{L} = \bigwedge_{e=1}^{N_e} \mathbf{L}^e$$

where $\mathbf{M}^e$ is the element mass matrix from Equation (2.14), $\mathbf{D}^e$ is the element differentiation matrix from Equation (2.15), and $\mathbf{L}^e$ is the element weak Laplacian matrix from Equation (2.21). Since the same set of Legendre-Gauss-Lobatto (LGL) points are used for both interpolation and integration, the global mass matrix $\mathbf{M}$ is diagonal and thus easy to invert. This is only the case if we integrate using $N+1$ LGL points as shown in Equation (2.7). This type of integration is known as inexact numerical integration, since the number of LGL quadrature points necessary to integrate a polynomial of order $2N$ (such as is the case for the mass matrix) up to machine precision is $N+2$. We choose to sacrifice accuracy in favor of obtaining an easily invertible mass matrix, which allows us to save considerable computational time. Additionally, it has been shown that when using polynomials of order $N \geq 4$ this type of integration has a minimal impact on accuracy, with the impact decreasing

as the polynomial order is increased [26]. It should be noted, however, that no global matrix is actually constructed (except for the diagonal mass matrix); the differentiation and Laplacian global matrices are never stored, only the action of these matrices on the solution vector is computed (see, e.g., [26]). The global form associated with Equation (2.1) for $G(f) = \boldsymbol{\nabla} \cdot \mathbf{f} + \boldsymbol{\nabla}^2 f$ can be written as:

$$\frac{\partial \mathbf{f}^h}{\partial t} + \mathbf{M}^{-1}(\mathbf{D}\mathbf{f}^h + \mathbf{L}\mathbf{f}^h) = 0, \tag{2.23}$$

where $\mathbf{f}^h$ is a global vector containing the nodal values of f^h.

2.1.2 Discontinuous Galerkin

For this kind of approximation, the global matrices are not constructed since an element communicates only with the neighboring elements through inter-element numerical fluxes. Thus, we write a local approximation of Equation (2.1), instead of a global one as in Equation (2.23).

Let us apply integration by parts to the entries of the differentiation matrix:

$$\mathbf{D}^e_{ij} = \int_{\Omega_e} \psi_i(\mathbf{x}) \boldsymbol{\nabla} \psi_j(\mathbf{x}) d\mathbf{x} = \int_{\Gamma_e} \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) \mathbf{n}^{(F,e)} d\Omega_e - \int_{\Omega_e} \boldsymbol{\nabla} \cdot \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) d\mathbf{x}, \tag{2.24}$$

where $i, j = 1, \ldots, (N+1)^3$, $\mathbf{n}^{(F,e)}$ is the outwards facing normal of inter-element face F of the element e. The first term of the right-hand side in Equation (2.24) represents an inter-element flux or a boundary flux, if the element is a boundary element and it enforces the continuity of the global solution. Notice that in a CG discretization this term vanishes at the interior faces (but not along boundary faces) as continuity is enforced via DSS. We define the corresponding element flux matrix $\mathbf{F}^e$ as follows:

$$\begin{aligned} \mathbf{F}^e_{ij} &= \int_{\Gamma_e} \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) \mathbf{n}^{(F,e)} d\mathbf{x} \\ &\approx \sum_{F=1}^{N_F} \sum_{k=1}^{N+1} \sum_{m=1}^{N+1} \omega(\boldsymbol{\xi}_{F,km}) \psi_i(\boldsymbol{\xi}_{F,km}) \psi_j(\boldsymbol{\xi}_{F,km}) |\mathbf{J}(\boldsymbol{\xi}_{F,km})| \mathbf{n}^{(F,e)}, \end{aligned} \tag{2.25}$$

where $i, j = 1, \ldots, (N+1)^3$, N_F is the number of faces for element e and $\boldsymbol{\xi}_{F,km}$ denotes an integration point on the face F of the element. The second term on the right-hand side in Equation (2.24) is called the weak differentiation matrix and is approximated as follows:

$$\begin{aligned}\hat{\mathbf{D}}^e_{ij} &= \int_{\Omega_e} \boldsymbol{\nabla}\psi_i(\mathbf{x})\psi_j(\mathbf{x})d\mathbf{x} \\ &\approx \sum_{k=1}^{N+1}\sum_{m=1}^{N+1}\sum_{n=1}^{N+1} \omega(\xi_k,\eta_m,\zeta_n)|\mathbf{J}(\xi_k,\eta_m,\zeta_n)|\boldsymbol{\nabla}\psi_i(\xi_k,\eta_m,\zeta_n)\psi_j(\xi_k,\eta_m,\zeta_n),\end{aligned} \tag{2.26}$$

where $i, j = 1, \ldots, (N+1)^3$.

We can now rewrite Equation (2.22) for a DG discretization taking $G(f) = \boldsymbol{\nabla}\cdot\mathbf{f} + \boldsymbol{\nabla}^2 f$, which holds on each element as follows:

$$\mathbf{M}^e_{ij}\frac{\partial \hat{f}^e_j(t)}{\partial t} = -\hat{\mathbf{D}}^e_{ij}\cdot\hat{\mathbf{f}}^e_j(t) + \mathbf{F}^e_{ij}\cdot\mathbf{f}^*_j(t) - \mathbf{L}^e_{ij}\hat{f}^e_j(t) = 0, \quad i,j = 1,\ldots,(N+1)^3,$$

where $\mathbf{f}^*$ represents the inter-element interface values of $\hat{f}^e_j$. We define $\mathbf{f}^*$ as follows:

$$\mathbf{f}^*_j = \mathbf{C}(\hat{f}^e_j) - \mathbf{P}(\hat{f}^e_j)$$

where $\mathbf{P}$ is a penalty term and the central term $\mathbf{C}$ is defined as follows:

$$\mathbf{C}(\hat{f}^e_j) = (g(\hat{f}^{e,R}_j) + g(\hat{f}^{e,L}_j))/2,$$

where L and R refer to the left and right sides of a given inter-element interface. The function g is dependent on the first derivative component of G in Equation (2.1) where, in this case, $G(f) = \boldsymbol{\nabla}\cdot\mathbf{f} + \boldsymbol{\nabla}^2 f$ and $g(f) = \mathbf{f}$. The definition of P depends on the choice of numerical flux. The simplest and most commonly used flux for DG is the Rusanov flux [26] and [72], which gives:

$$\mathbf{P}(\hat{f}^e_j) = \mathbf{n}^{(F,e)} w_s(\hat{f}^{e,R}_j - \hat{f}^{e,L}_j)/2,$$

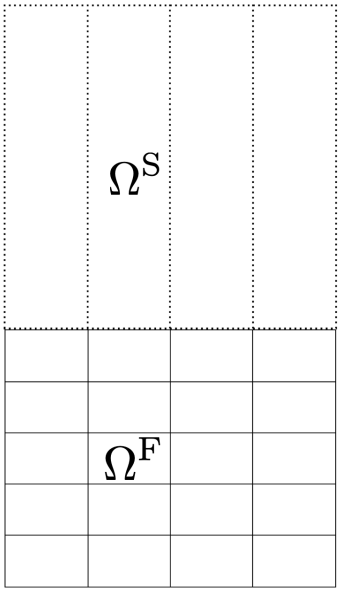


Figure 2.2 Example of a finite spectral element domain Ω^F with $N_F = 16$ connected to a semi-infinite element domain Ω^S with $N_S = 4$.

where w_s is the wave speed across the interface, which depends on the specific equation to be solved. This gives the following equation for $\mathbf{f}^*$:

$$\mathbf{f}_j^* = \frac{1}{2}\left(\hat{\mathbf{f}}_j^{e,R} + \hat{\mathbf{f}}_j^{e,L} - \mathbf{n}^{F,e} w_s (\hat{f}_j^{e,R} - \hat{f}_j^{e,L})\right), \quad j = 1, \ldots, (N+1)^3, \tag{2.27}$$

where $\hat{\mathbf{f}}_j^e = (\hat{f}_j^e, \hat{f}_j^e, \hat{f}_j^e)$. We note that in the DG formulation for $G(f) = \boldsymbol{\nabla} \cdot \mathbf{f} + \boldsymbol{\nabla}^2 f$ the boundary term in Equation (2.18) does not vanish and needs to be evaluated. Such term is treated in a similar fashion as the boundary term in Equation (2.24). For the details, we refer the interested reader to [26, 41].

2.2 A Galerkin Method for Modeling Semi-infinite Spaces

It is also possible to use the Galerkin methodology to discretize semi-infinite domains. In our case, this is done using semi-infinite elements. An example of this type of element is presented in Figure 2.2 The semi infinite domain uses both a different set of basis functions and integration points than its finite domain counterpart. First, define the Laguerre polynomials using their three term recurrence relation [96, Equation

(3.3)]:

$$L_0(\xi) = 1 \tag{2.28a}$$

$$L_1(\xi) = 1 - x \tag{2.28b}$$

$$L_k(\xi) = \frac{2k-1-\xi}{k} L_{k-1}(\xi) - \frac{k-1}{k} L_{k-2}(\xi), \forall k \geq 2. \tag{2.28c}$$

Additionally the first derivative of the k-th Laguerre polynomial is [96, Equation (3.5)]:

$$L_k'(\xi) = -\sum_{n=0}^{k-1} L_n(\xi). \tag{2.29}$$

The Laguerre polynomials are orthogonal on the semi-infinite interval $[0, \infty)$ with respect to an exponentially decaying weight. This orthogonality property can be written as follows:

$$\int_0^\infty L_i(\xi) L_j(\xi) e^{-\xi} d\xi = \delta_{ij}, \forall i, j \geq 0. \tag{2.30}$$

The Laguerre-Gauss-Radau (LGR) points are the roots of $\xi L_{N+1}'(\xi)$ for a fixed integer N. The LGR points $\{\xi_j\}_{j=0,N}$ will be used to construct a nodal spectral element on the semi-infinite elements. We compute the LGR points using the Eigenvalue Method [28, 96], which forms a tridiagonal matrix using the coefficients in the three-term recurrence relationship Equation (2.28) and solves an eigenvalue problem. This method is stable and robust at very high-order and we have tested it for orders up to 60. Following [95], the scaling factor λ adjusts the LGR nodes, such that the physical nodes on the semi-infinite element are $\{x\}_{i=1}^{N+1} = \lambda\{\xi\}_{i=1}^{N+1}$. This scaling factor allows us to adjust the effective length of the semi-infinite element for a given problem.

Next, we construct the Lagrange-Laguerre interpolating polynomials associated with the LGR points $\{\xi_j\}_{j=0,\ldots,N}$ points following [95]:

$$h_j^{Lag}(\xi) = -\frac{\xi L_{N+1}^{'}(\xi)}{(N+1)L_{N+1}(\xi_j)(\xi-\xi_j)}. \tag{2.31}$$

We can then write their derivatives as follows:

$$h_j^{'Lag}(\xi_i) = \begin{cases} \frac{L_{N+1}(\xi_i)}{L_{N+1}(\xi_j)(\xi_i-\xi_j)} & \text{if } i \neq j \\ \frac{1}{2} & \text{if } i = j \neq 0 \\ \frac{N}{2} & \text{if } i = j = 0. \end{cases} \tag{2.32}$$

We now introduce the scaled Laguerre function (SLF) [95, 96, 5]

$$\hat{L}_i(\xi) = e^{-\frac{\xi}{2\lambda}} L_i\left(\frac{\xi}{\lambda}\right), \tag{2.33}$$

where λ is a scaling factor and represents a characteristic length. Note that this notation is equivalent to the notation in [5] for $\lambda = \beta^{-1}$. Applying Equation (2.30) yields

$$\int_0^\infty \hat{L}_i(\xi)\hat{L}_j(\xi)d\xi = \lambda\delta_{ij}, \forall i,j \geq 0, \tag{2.34}$$

indicating that the SLFs form an orthogonal basis on $L^2(\mathbb{R}^+)$. Each SLF decays exponentially as $\xi \to \infty$ for any $\lambda > 0$. This property is illustrated in figure 2.3, which shows the first six Laguerre functions $\hat{L}_i(\xi)$ for $\lambda = 1$. Thanks to this property, the SLFs Equation (2.33) are ideal for approximating functions in an absorbing layer; the layer damps any outgoing perturbations by enforcing the exponential decay property.

Remark: We limit the use of damping terms to Ω^S and essentially overlap the sponge layer with the semi-infinite elements. In the results, we will describe the damping coefficients used for each test and we will show that minimal reflection can be obtained while relying on this approach. We can now construct the Lagrange-

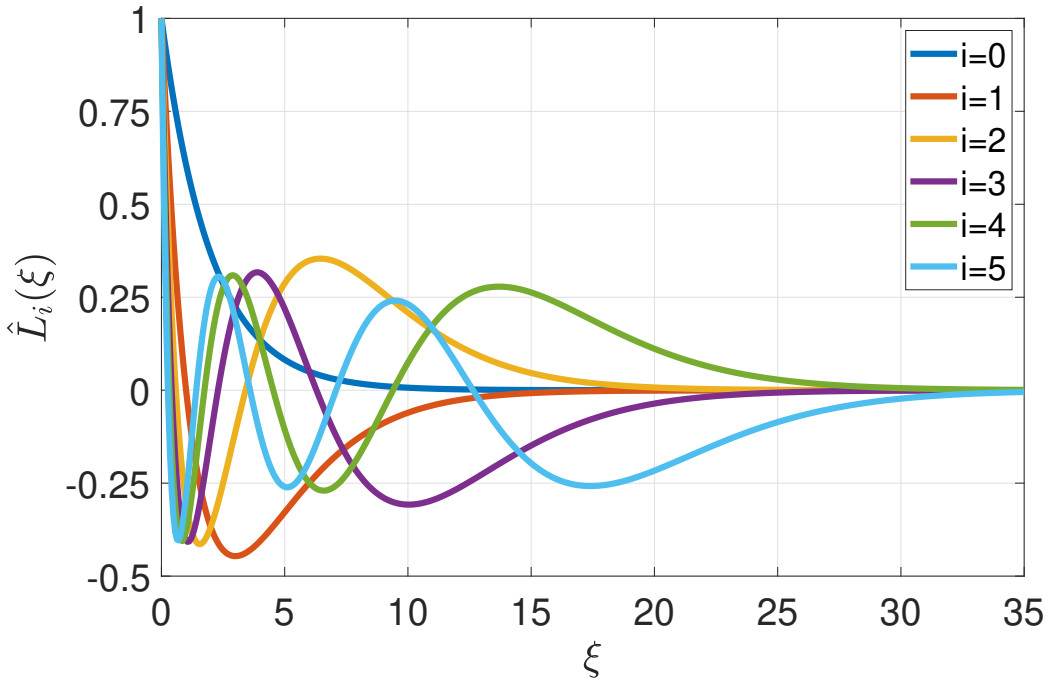


Figure 2.3 First six scaled Laguerre functions (SLFs) specified by Equation (2.33).

Laguerre interpolant associated with LGR points $\{\hat{h}_j^{Lag}\}_{j=0}^{N}$ such that $\hat{h}_j^{Lag}(\xi_i) = \delta_{ij}$, and

$$\hat{h}_j^{Lag}(\xi) = \frac{\exp(-\xi/2)}{\exp(-\xi_j/2)} h_j^{Lag}(\xi). \tag{2.35}$$

Their derivatives are:

$$\hat{h}_j'^{Lag}(\xi_i) = \begin{cases} \frac{\hat{L}_{N+1}(\xi_i)}{\hat{L}_{N+1}(\xi_j)(\xi_i - \xi_j)} & \text{if } i \neq j \\ 0 & \text{if } i = j \neq 0 \\ -\frac{N+1}{2} & \text{if } i = j = 0. \end{cases} \tag{2.36}$$

The quadrature weights $\{\hat{\omega}(\xi_i)\}_{i=0}^{N}$ associated with the LGR points are defined as

$$\hat{\omega}(\xi_i) = \frac{\exp(\xi_i)}{(N+1)[L_N(\xi_i)]^2}. \tag{2.37}$$

Let us now consider the reference 2D semi-infinite element Ω_{ref}^{S} such that it is only semi-infinite in the direction of the outgoing waves. In its finite direction, the

LGL nodes and their associated bases are used, and in its semi-infinite direction the LGR nodes and their associated SLF bases are used. As such, for an integration point $\xi \in \Omega^S_{ref}$, we can write $\xi = (\xi_{LGL}, \eta_{LGR})$. The nodal basis functions ψ^S_l on the semi-infinite element are the tensor product of the 1D Lagrange polynomials associated with the LGL nodes and the 1D Lagrange-Laguerre interpolating functions associated with the LGR nodes giving us the following:

$$\psi^S_l(\mathbf{x})_{\Omega^S_{ref}} = h_i[\xi(\mathbf{x})] \otimes \hat{h}_j[\eta(\mathbf{x})], \quad l = i + 1 + j(N_{LGL}), \tag{2.38}$$

where $i \in \{1, \ldots, N_{LGL}\}$, $j \in \{1, \ldots, N_{LGR}\}$, N_{LGL} is the number of LGL nodes and N_{LGR} is the number of LGR nodes. This makes it so that Equation (2.7) remains valid on semi-infinite elements provided the appropriate substitutions of nodes and weights is performed.

2.2.1 Constructing element matrices

In this section, we present the reader with a template for how an element matrix is constructed for elements of the finite and semi-infinite domains. In what follows we will discuss the construction of the mass matrix. We refer the reader to [26] for constructions of the differentiation or laplacian matrices on the finite domain. The extension of these constructions to the semi-infinite domain is done similarly to the construction of the mass matrix. Let us first define every component of the mass matrix on a given spectral or semi-infinite element $\Omega_e \in \Omega^h = \Omega^F \cup \Omega^S$:

$$\mathbf{M}^e_{ij} = \int_{\Omega_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x})d\mathbf{x} = \int_{\Omega_{ref}} \psi_i(\boldsymbol{\xi})\psi_j(\boldsymbol{\xi})|\mathbf{J}(\boldsymbol{\xi})|d\boldsymbol{\xi} \ \forall i, j = 1, \ldots, N_\xi N_\eta, \tag{2.39}$$

where ψ_i is defined using a generalized form of Equation (2.5) as follows:

$$\psi_l(\mathbf{x}) = h_i[\xi(\mathbf{x})] \otimes \overline{h}_j[\eta(\mathbf{x})], \quad l = i + (j-1)N_\eta, \tag{2.40}$$

and where,

$$\Omega_{ref}, N_\xi, N_\eta, \overline{h}_j = \begin{cases} \Omega^F_{ref}, N_{LGL}, N_{LGL}, h_j & \text{if } \Omega_e \in \Omega^F \\ \Omega^S_{ref}, N_{LGL}, N_{LGR}, \hat{h}^{Lag}_j & \text{if } \Omega_e \in \Omega^S \end{cases}. \tag{2.41}$$

By approximating Equation (2.40) using an inexact quadrature rule (i.e., the quadrature and interpolation points coincide) we obtain the following:

$$\mathbf{M}^e_{ij} = \sum_{k=1}^{N_\xi} \sum_{m=1}^{N_\eta} \overline{\omega}(\xi_k, \eta_m) \psi_i(\xi_k, \eta_m) \psi_j(\xi_k, \eta_m) |\mathbf{J}(\xi_k, \eta_m)|, \tag{2.42}$$

$$\overline{\omega}(\xi_k, \eta_m) = \omega_\xi(\xi_k) \omega_\eta(\eta_m), \tag{2.43}$$

$$\omega_\xi, \omega_\eta = \begin{cases} \omega, \omega & \text{if } \Omega_e \in \Omega^F \\ \omega, \hat{\omega} & \text{if } \Omega_e \in \Omega^S \end{cases}, \tag{2.44}$$

$$N_\xi, N_\eta = \begin{cases} N_{LGL}, N_{LGL} & \text{if } \Omega_e \in \Omega^F \\ N_{LGL}, N_{LGR} & \text{if } \Omega_e \in \Omega^S \end{cases}. \tag{2.45}$$

Next, we present the pseudo-code for constructing the mass matrix of an element in the semi-infinite domain using inexact integration. Let us define $\mathbf{M}^{Lag}$ as the mass matrix of an element of the semi-infinite domain, η^{LGR} as the LGR nodes, and ξ as the LGL nodes:

2.2.2 Direct stiffness summation

In order to couple the element local Galerkin expansion given by Equation (2.4) between adjacent elements, we need to construct a direct stiffness summation (DSS)

Algorithm 1 Construction of the mass matrix of a semi-infinite element

$\mathbf{M}^{Lag} = \text{zeros}(N_{LGL}N_{LGR}, N_{LGL}N_{LGR})$

for $l = 1, N_{LGR}$ **do**

  for $k = 1, N_{LGL}$ **do**

    $I = k + (l-1)(N_{LGL})$

    $\overline{\omega} = \omega(\xi_k)\hat{\omega}(\eta_l^{LGR})$

    $\mathbf{x} = (\xi_k, \eta_l^{LGR})$

    for $j = 1, N_{LGR}$ **do**

      for $i = 1, N_{LGL}$ **do**

        $J = i + (j-1)(N_{LGL})$

        $\mathbf{M}_{IJ}^{e,Lag} = \mathbf{M}_{IJ}^{e,Lag} + \overline{\omega}\psi_I(\mathbf{x})\psi_J(\mathbf{x})|\mathbf{J}(\xi_k, \eta_l^{LGR})|$

      end for

    end for

  end for

end for

operator. The DSS operator enforces the continuity of the global solution by averaging the state variable on nodes shared by multiple elements. As shown in the next two paragraphs, this DSS operator couples the finite domain Ω^F and the semi-infinite domain Ω^S illustrated in Figure 2.2 in a straight-forward manner.

First, we must define mappings from local elements to global nodes. Let $I = H^F(e, i)$ be the map from the local element-wise node i on the e-th finite element Ω_e^F and let $I = H^S(e, i)$ be the corresponding map from the e-th semi-infinite element Ω_e^S. For H^F, i runs from 1 to N_{LGL}^2, while for H^S, i runs from 1 to $N_{LGL}N_{LGR}$. These mappings contain the connectivity information in the finite and semi-infinite grids, respectively.

We can now illustrate how the global problem is assembled by considering the mass matrix. Let $\mathbf{M}_i^e$ be a local diagonal mass matrix corresponding to element e (either finite or semi-infinite). We construct the global mass matrix via a DSS operator

$$\mathbf{M}_{IJ} = \bigwedge_{e=1}^{N_e} \mathbf{M}_{ij}^e, \tag{2.46}$$

where the DSS operator $\bigwedge_{e=1}^{N_e}$ consists of a local-global mapping and appropriate summation. For additional details, see Section 5.8 in [26] or [51] for the parallel MPI implementation. This DSS operator may be decomposed into two independent DSS operators, the first over the collection of N^f finite elements, and the second over the collection of N^s semi-infinite elements via

$$\mathbf{M}_{IJ} = \left[\bigwedge_{e_f=1}^{N_f} \mathbf{M}_{ij}^{e_f} \right] \bigwedge \left[\bigwedge_{e_s=1}^{N_s} \mathbf{M}_{ij}^{e_s} \right]. \tag{2.47}$$

Hence, the DSS operation consists of three stages: 1) perform a DSS over the finite domain Ω^F using the local to global mapping H^F, 2) perform a DSS over the semi-infinite domain Ω^S using the mapping H^S, and finally 3) DSS the nodes shared by both the finite and semi-infinite domains $\Omega^F \cap \Omega^S$ using both H^F and H^S. Since the only coupling between the finite and semi-infinite grids is via this final DSS ope rator, the proposed semi-infinite approach may be retrofitted to an existing spectral element solver with only minor modifications.

Low pass spectral filter To control unresolved grid-scale noise and aliasing, low-pass spectral filters are typically used with the spectral element method (SEM) [Sec. 18.3][26]. Spectral filters employ a three-step process to damp/remove unphysical high-frequency components: 1) the element-local nodal solution is transformed into modal space, 2) a low-pass filter is applied in modal space, 3) the filtered modal

representation is inverse transformed to nodal space. The nodal to modal transform requires defining a set of modal basis functions. As discussed in Section 3d of [27], the low-pass filter should not violate the continuity requirement of SEM. One way to enforce this is by choosing modal functions such that most of the modes are zero at the element boundary. For a 1D element using LGL points, an appropriate choice is $\phi_k(\xi) = P_k(\xi)$ for $k = 0$ or 1 and $\phi_k(\xi) = P_k(\xi) - P_{k-2}(\xi)$ for $k \geq 2$. Since $\phi_k(\pm 1) = 0$ for $k \geq 2$, these higher-order modes do not effect the boundary of the element, and hence may be damped by an appropriate filter function. The resulting transform (or Legendre) matrix is given by Equation (31) in [27].

For a 1D semi-infinite element using LGR points, a similar choice of modal functions is $\phi_0(\xi) = e^{-\xi/2}$ and $\phi_k(\xi) e^{-\xi/2} [L_k(\xi) - L_{k-1}(\xi)]$. Since $L_k(0) = 1$ for all $k \geq 0$, we have $\phi_k(0) = 0$ for $k \geq 1$. The Legendre matrix is then constructed by evaluating $\phi_k(\xi_j)$, where ξ_j are the LGR points. Since we employ a tensor product of LGL and LGR points within semi-infinite element, we construct a corresponding 2D tensor product of modal functions in order to transform the nodal representation. A Boyd-Vandeven filter [9] is applied to the modal representation, and then the solution is inverse transformed to nodal space.

CHAPTER 3

GOVERNING EQUATIONS

3.1 Problem Definition

Moist air is a mixture of dry air with density ρ, water vapor with density ρ_v, and suspended cloud condensate with density ρ_c. The mass fractions of water vapor and cloud water are defined as $q_v = \rho_v/\rho$ and $q_c = \rho_c/\rho$, respectively. In addition, let ρ_r be the rain density and $q_r = \rho_r/\rho$ the rain mass fraction. Warm rain is assumed (no ice formation or precipitation take place). We denote by c_p and c_v the specific heat capacities at constant pressure and volume for dry air. The specific gas constants of dry air and vapor are denoted by R_d and R_v and set $\epsilon = \frac{R_d}{R_v}$. Let:

$$\theta = (1 + \epsilon q_v)\frac{T}{\pi}, \quad \text{with } \pi = \left(\frac{p}{p_s}\right)^{\frac{R_d}{c_p}}, \tag{3.1}$$

be the virtual potential temperature, where T is the absolute temperature and $p_s = 10^5$ Pa is the ground surface pressure. Finally, let $\mathbf{u}$ be the wind velocity.

We consider a fixed spatial domain Ω and a time interval of interest $(0, t_f]$. Balance of mass, momentum, and potential temperature for moist air in terms of *prognostic variables* ρ, $\mathbf{u}$, and θ in conservative form are given by:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \qquad \text{in } \Omega \times (0, t_f], \tag{3.2}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \rho \mathbf{b} \qquad \text{in } \Omega \times (0, t_f], \tag{3.3}$$

$$\frac{\partial (\rho \theta)}{\partial t} + \nabla \cdot (\rho \theta \mathbf{u}) = \rho \mathcal{S}_\theta \qquad \text{in } \Omega \times (0, t_f]. \tag{3.4}$$

where $\otimes$ denotes the tensor-product operator (i.e., the Kronecker product) and $\mathbf{b}$ is the total buoyancy. We have $\mathbf{b} = -(1 + \epsilon q_v - q_c - q_r) g \widehat{\mathbf{k}}$, where $g = 9.81$ m/s^2 is the

magnitude of the acceleration of gravity, and $\widehat{\mathbf{k}}$ is the unit vector aligned with the vertical axis z. Finally, the source/sink term $\mathcal{S}_\theta$ in Equation (3.4) describes latent heat release–uptake during phase changes of moisture variables and is detailed in Section 3.2. Equation (3.3) and Equation (3.4) can be rewritten in non-conservative form as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \mathbf{b} \qquad \text{in } \Omega \times (0, t_f], \tag{3.5}$$

$$\frac{\partial \theta}{\partial t} + \mathbf{u} \cdot \nabla \theta = \mathcal{S}_\theta \qquad \text{in } \Omega \times (0, t_f]. \tag{3.6}$$

A thermodynamics equation of state for the pressure of moist air p is needed for closure. We assume that p is the sum of the partial pressures of dry air and vapor (p_d and p_v, respectively), both taken to be ideal gases. Thus, neglecting the volume of the condensed phase, the equation of state relating p to ρ and T is given by:

$$p = p_d + p_v = \rho R_d T + \rho q_v R_v T = \rho R_d T (1 + \epsilon q_v). \tag{3.7}$$

To facilitate the numerical solution of the system of Equations (3.2)-(3.4) or (3.2), (3.5)-(3.6), we write density, pressure, and potential temperature as the sum of their mean hydrostatic values and fluctuations:

$$\rho(x, y, z, t) = \rho_0(z) + \rho'(x, y, z, t), \tag{3.8}$$

$$\theta(x, y, z, t) = \theta_0(z) + \theta'(x, y, z, t), \tag{3.9}$$

$$p(x, y, z, t) = p_0(z) + p'(x, y, z, t). \tag{3.10}$$

Note that the hydrostatic reference states are functions of the vertical coordinate z only. Hydrostatic balance relates p_0 to ρ_0 as follows:

$$\frac{dp_0}{dz} = -\rho_0 g. \tag{3.11}$$

Plugging Equations (3.8)- (3.10) into Equations (3.2)-(3.4) and accounting for Equation (3.11) leads to:

$$\frac{\partial \rho'}{\partial t} + \nabla \cdot ((\rho_0 + \rho')\mathbf{u}) = 0, \tag{3.12}$$

$$\frac{\partial ((\rho_0 + \rho')\mathbf{u})}{\partial t} + \nabla \cdot ((\rho_0 + \rho')\mathbf{u} \otimes \mathbf{u}) + \rho' g \widehat{\mathbf{k}} = -\nabla p' + (\rho_0 + \rho')\widetilde{\mathbf{b}}, \tag{3.13}$$

$$\frac{\partial ((\rho_0 + \rho')(\theta_0 + \theta'))}{\partial t} + \nabla \cdot ((\rho_0 + \rho')\theta'\mathbf{u}) + \nabla \cdot ((\rho_0 + \rho')\theta_0\mathbf{u}) = (\rho_0 + \rho')\mathcal{S}_\theta, \tag{3.14}$$

where $\widetilde{\mathbf{b}} = -\left(\frac{\rho'}{\rho_0+\rho'} + \epsilon q_v - q_c - q_r\right) g\widehat{\mathbf{k}}$ is a modified total buoyancy. Following a similar procedure for Equations (3.5)-(3.6), we obtain

$$\frac{\partial \rho'}{\partial t} + \nabla \cdot ((\rho_0 + \rho')\mathbf{u}) = 0, \tag{3.15}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho_0 + \rho'} \nabla p' + \widetilde{\mathbf{b}}, \tag{3.16}$$

$$\frac{\partial \theta'}{\partial t} + \mathbf{u} \cdot \nabla \theta_0 + \mathbf{u} \cdot \nabla \theta' = \mathcal{S}_\theta. \tag{3.17}$$

Remark 3.1.1 *To preserve numerical stability of the solution, we add an artificial diffusion term with a diffusivity coefficient β to Equations* (3.12)-(3.14) *and* (3.15)-(3.17)*; the units of β are given consistently with the equations at hand. The term $\beta\nabla^2\mathbf{u}$ is added to the right-hand side of the momentum equation, while the term $\beta\nabla^2\theta'$ is added to the right-hand side of the equation of the potential temperature.*

Next, we write the balance equations for q_v and q_c in conservative form:

$$\frac{\partial (\rho q_v)}{\partial t} + \nabla \cdot (\rho q_v \mathbf{u}) = \rho \mathcal{S}_v \qquad \text{in } \Omega \times (0, t_f], \tag{3.18}$$

$$\frac{\partial (\rho q_c)}{\partial t} + \nabla \cdot (\rho q_c \mathbf{u}) = \rho \mathcal{S}_c \qquad \text{in } \Omega \times (0, t_f], \tag{3.19}$$

and non-conservative form:

$$\frac{\partial q_v}{\partial t} + \mathbf{u} \cdot \nabla q_v = \mathcal{S}_v \qquad \text{in } \Omega \times (0, t_f], \tag{3.20}$$

$$\frac{\partial q_c}{\partial t} + \mathbf{u} \cdot \nabla q_c = \mathcal{S}_c \qquad \text{in } \Omega \times (0, t_f]. \tag{3.21}$$

The source/sink terms on the right-hand side in the equations above are related to conversion rates. In particular, we have:

$$\mathcal{S}_v = C(q_c \to q_v) + C(q_r \to q_v), \quad \mathcal{S}_c = C(q_v \to q_c) + C(q_r \to q_c), \quad \mathcal{S}_t = \mathcal{S}_v + \mathcal{S}_c, \tag{3.22}$$

where the terms $C(q_\phi \to q_\psi) = -C(q_\psi \to q_\phi)$ represent the conversion of species ϕ to species ψ. All of these terms, which account for processes such as evaporation of cloud condensate, are provided by the microphysics equations reported in Section 3.2.

Precipitating water (rain) is treated in the same manner. Letting w_r be the fall speed of rain (provided by the microphysics equations), we can write the conservation law for rain in conservative form:

$$\frac{\partial(\rho q_r)}{\partial t} + \nabla \cdot (\rho q_r(\mathbf{u} - w_r\widehat{\mathbf{k}})) = \rho \mathcal{S}_r \quad \text{in } \Omega \times (0, t_f], \tag{3.23}$$

and non-conservative form:

$$\frac{\partial q_r}{\partial t} + \mathbf{u} \cdot \nabla q_r = \mathcal{S}_r + \frac{1}{\rho}\frac{\partial}{\partial z}(\rho q_r w_r) \quad \text{in } \Omega \times (0, t_f], \tag{3.24}$$

with

$$\mathcal{S}_r = C(q_v \to q_r) + C(q_c \to q_r). \tag{3.25}$$

In summary, the conservative form of the atmospheric model considered in this work is given by Equations (3.12)-(3.14), (3.18)-(3.19), (3.23) and (3.7), while its non-conservative form is given by Equations (3.15)-(3.17), (3.20)-(3.21), (3.24) and (3.7). In both cases, the problem has to be supplemented with proper initial and boundary conditions that are problem dependent.

3.2 Microphysical Parameterization

The terms on the right-hand sides of Equations (3.14), (3.18), (3.19), and (3.23), and their respective non-conservative counterparts are defined according to [53]. Let q_{vs} be the saturation water vapor fraction. To determine q_{vs} we use Teten's formula following [53]. The evaporation of cloud water is given by:

$$C(q_c \to q_v) = -C(q_v \to q_c) = \frac{\partial q_{vs}}{\partial t}. \tag{3.26}$$

This is computed with the saturation adjustment approach of [102]. The evaporation of rain, i.e., conversion rate $C(q_r \to q_v) = -C(q_v \to q_r)$, is taken directly from [53], which use an approach similar to [87]. We have

$$C(q_c \to q_r) = -C(q_r \to q_c) = A_r + C_r, \tag{3.27}$$

where A_r and C_r represent rain auto-conversion and rain accretion [52], respectively. The source/sink term in Equation (3.17) is given by:

$$S_\theta = -\gamma \left(\frac{\partial q_{vs}}{\partial t} + C(q_r \to q_v) \right), \quad \gamma = \frac{L}{c_p \pi}, \tag{3.28}$$

where L is the latent heat of vaporization and π is the Exner pressure defined in Equation (3.1).

Finally, we define the terminal velocity of rain following [102, 52, 53]:

$$w_r = 3634(\rho q_r^{0.1346}) \left(\frac{\rho}{\rho_g} \right)^{-\frac{1}{2}}, \tag{3.29}$$

where ρ_g is the reference density at the surface.

CHAPTER 4

LAGUERRE SEMI-INFINITE ELEMENTS AS A POTENTIAL SOURCE OF COMPUTATIONAL EFFICIENCY FOR ATMOSPHERIC FLOWS

This chapter presents the results of applying the Laguerre-Legendre semi-infinite elements approach to benchmark atmospheric test cases with and without topography. All the tests in this chapter are done in a dry atmosphere where $q_v = 0.0$, $q_c = 0.0$, and $q_r = 0.0$ everywhere and at all times. These tests were run on `Jexpresso` [75], a new open-source spectral element code written in performant Julia.

4.1 A Semi-infinite Rising Thermal Bubble Test

We validate the Laguerre-Legendre semi-infinite element approach on the compressible Euler equations by using it to simulate a classic rising thermal bubble case. In this test a perturbation $\Delta\theta$ is introduced to a neutral atmosphere with uniform potential temperature $\theta_0 = 300$ K. The finite domain is $\Omega^F = [-5, 5]$ km $\times$ $[0, 5]$ km, which is subdivided into $N_z = N_x = 20$ elements of order $N = 4$. This yields a horizontal resolution $\Delta x = 125$ m and a vertical resolution $\Delta z = 62.5$ m. The potential temperature perturbation and initial pressure are defined as follows:

$$\Delta\theta = \theta_c \left(1 - \frac{r}{r_0}\right) \tag{4.1a}$$

$$r = \sqrt{(x - x_c)^2 + (z - z_c)^2} \tag{4.1b}$$

$$p = p_0 \left(1 - \frac{gz}{c_p\theta}\right)^{cp/R}, \tag{4.1c}$$

where $\theta_c = 2$ K, $r_0 = 2000$ m, $x_c = 0$ m, $z_c = 2500$ m, and $p_0 = 1000$ hPa. The initial potential temperature is $\theta = \theta_0 + \Delta\theta$, and the initial density can be deduced through the ideal gas law (3.7).

A set of 20 Laguerre-Legendre semi-infinite elements of order 24 is added on top of Ω^F to build the semi-infinite domain $\Omega^S = [-5, 5]$ km $\times$ $[5, \infty)$ km. To verify that this approach is able to solve the Euler equations without adversely affecting the solution, we do not use a damping layer for this test and simply verify that the rising thermal crosses the $\Omega^F - \Omega^S$ boundary without being affected by the change in element type and drastic change in resolution. To stabilize the solution past $t = 500$ s a viscosity coefficient $\nu = 30$ m^{2}s^{-1} is used in conjunction with a constant thermal diffusivity $\kappa = 2\nu$. This is standard and would be required even for simulations without a semi-infinite element. Free-slip type boundary conditions are used at all domain edges.

The initial potential temperature perturbation generates positive buoyancy, which causes the bubble to rise. Figure 4.1 shows the solution at $t = 1000$ s after a significant portion of it has transported through the interface between Ω^F and Ω^S at $z = 5$ km. The figure shows that the solution is correctly transported and diffused through the interface. The symmetry of the bubble in the x direction is maintained, and the interface does not introduce spurious noise or discontinuities. The figure also highlights the improvement in solution quality as the order of the semi-infinite element is increased. The sharpest solution is obtained using a semi-infinite element of order 48 which is displayed in the right panel of Figure 4.1. With this approach validated for the Euler equations, we move on to demonstrating that we can use it for effective absorbing layers in atmospheric flows.

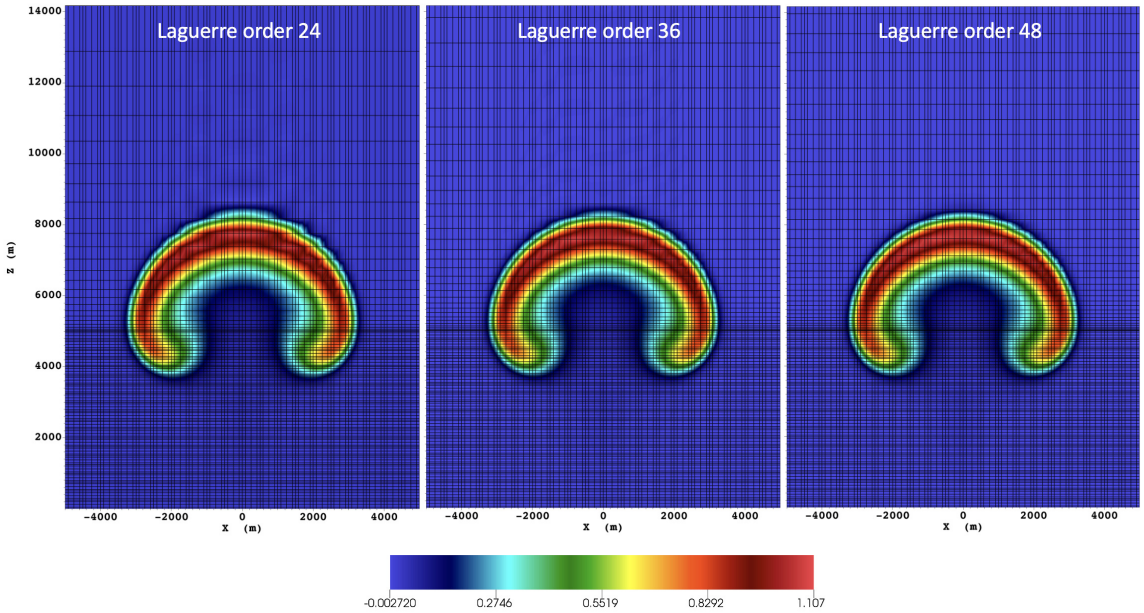


Figure 4.1 Potential temperature perturbations for the rising thermal bubble at $t = 1000$ s using semi-infinite elements of order 24 (left), 36 (middle), and 48 (right). In all three cases the bubble rises correctly and is able to pass through the interface between Ω^F and Ω^S at $z = 5$ km without any issues. Note that a higher order semi-infinite element yields a sharper, higher resolution solution. The grid used for this test is fully visible to visually show the transition between the two domains. Note that no absorbing layer is used in this test since the goal is not that of testing non-reflecting conditions, but to assess the sanity of the Legendre-Lagerre discretization.

Linear hydrostatic mountain waves In this test, a constant horizontal flow with velocity $U = 20$ ms^{-1} impinges on a mountain in a stratified atmosphere. These flow conditions and the mountain's size determines the structure of the resulting waves. As soon as the flow encounters the mountain, gravity waves quickly propagate both horizontally and vertically. In a proper implementation, these waves should leave the domain with no reflection.

The background state for this test is a hydrostatically balanced atmosphere whose pressure and potential temperature are:

$$p = p_0 \left[1 + \frac{g^2}{c_p \theta_0 N^2} \left(\exp\left(\frac{-z\mathcal{N}^2}{g} \right) - 1 \right) \right]^{c_p/R} \tag{4.2a}$$

$$\theta = \theta_0 \exp\left(\frac{z\mathcal{N}^2}{g} \right), \tag{4.2b}$$

where $p_0 = 1000$ hPa and $\theta_0 = 250$ K are the sea level values of pressure and potential temperature, and

$$\mathcal{N} = \frac{g}{\sqrt{c_p \theta_0}} = 0.0196 \text{ s}^{-1}$$

is the Brunt-Väisälä frequency. An Agnesi mountain with height $h = 1$ m and half-width $a = 10,000$ m is located at the center of the domain $x_c = 0$ m with shape

$$z = \frac{ha^2}{(x - x_c)^2 + a^2}. \tag{4.3}$$

For this test as well as for all other cases involving topography, a terrain following sigma coordinate [24] is used. We consider a finite domain $\Omega^F = [-120, 120]$ km $\times$ $[0, 15]$ km, and subdivided it into $N_x \times N_z = 120 \times 21$ elements of order 4, which lead to the resolution $(\Delta x, \Delta z) = (500 \text{ m}, 178 \text{ m})$. A set of 120 Laguerre-Legendre semi-infinite elements are added on top of Ω^F yielding the semi-infinite domain $\Omega^S = [-120, 120]$ km $\times$ $[15, \infty)$ km. In the horizontal direction,

each semi-infinite element uses an order four spectral element discretization using LGL nodes, and in the vertical direction it uses an order 14 Laguerre function basis on LGR nodes and a scaling factor $\lambda = 300$ m which yields an end point $Z_{end} = 28853$. Due to prevalence of both acoustic waves and gravity waves, as well as the sensitivity of high order numerical methods, we make use of the spectral low pass Boyd-Vandeven filter [9] to help insure the stability of the simulation (this approach is standard for spectral element simulations of the atmosphere). The filter on the elements in Ω^S is different from that in Ω^F and the continuity of the solution at shared interface nodes is insured by the DSS operation.

To trigger the hydrostatic waves, a uniform eastward wind with speed $U = 20.0$ ms^{-1} impinges the mountain. We use a sine squared function to define the damping coefficient in the vertical absorbing layer:

$$\gamma(z) = \Delta\gamma \sin^2\left(\frac{\pi}{2}\frac{z - z_s}{z_{max} - z_s}\right), \tag{4.4}$$

where $z_s = 15000$ m, $\Delta\gamma = 0.1\text{s}^{-1}$ and z_{max} is the top of the absorbing layer. The lateral boundaries are periodic and lateral damping layers are used on each side (these damping layers are in Ω^F and are not on semi-infinite elements). These boundary conditions are also used for the remaining other mountain tests presented in this paper.

Figure 4.2 shows the contours of the vertical velocity at $t = 30,000$ s. The solution is stable, the outgoing waves are effectively damped within Ω^S, and the solution is physically meaningful when compared to other numerical solutions of atmospheric models using spectral elements see e.g., [103].

Table 4.1 shows the time per time step for different configurations of the linear hydrostatic mountain test case. A fixed domain end point $Z_{end} = 30,000$ m is

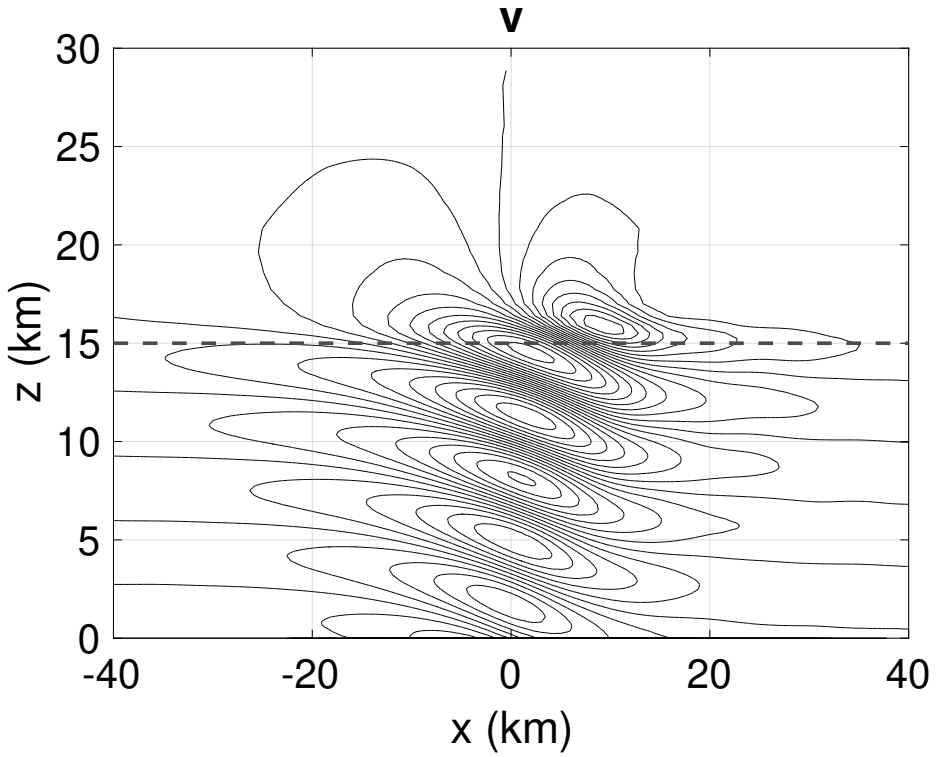


Figure 4.2 Time-converged numerical vertical velocity w for the linear hydrostatic mountain. The numerical solutions are displayed at t=30,000 s for $\Delta x = 500$ m, $\Delta z = 178$ m. The contours of the vertical velocity range between -0.005 and +0.005 with interval 0.0005. The dashed line shows where Ω^F meets Ω^S. In Ω^S there is only one element of order 14 along the vertical direction (see Figure. 2.2 for reference).

maintained for the cases where no semi-infinite element is used and instead only the number of vertical elements is changed. Even while only using ten additional order 4 spectral elements in the vertical direction to replace the Laguerre semi-infinite element the time per time step remains about 10% higher than running a single order 20 Laguerre element. If we would seek to run the entire vertical domain at the same resolution as the finite domain of these simulations utilizing a Laguerre semi-infinite element, the cost increase would be nearly 50% higher. Considerably few additional vertical spectral elements (we estimate no more than three) would have to be used to obtain a similar time per time step as using a single semi-infinite element in this case and this would also yield a much lower order discretization within the absorbing layer.

Table 4.1 Timings of Linear Hydrostatic Mountain Simulations With and Without Laguerre Semi-infinite Elements in the Absorbing Layer.

Absorbing layer type	T^*	Z_{end}	T^*_{Finite} %	$T^*_{Laguerre}$ %	N_z
Semi-infinite elements of order 14	1	28853	90.41	9.59	21
Semi-infinite elements of order 18	1.02	33279	87.97	12.03	21
Semi-infinite elements of order 20	1.04	35513	86.49	13.51	21
Extended finite domain of order four	1.14	30000	100	N/A	31
”	1.29	”	”	”	35
”	1.47	”	”	”	40

Note: A fixed vertical domain end point is used for simulation using only a finite domain while the simulations using semi-infinite elements are allowed to have varying Z_{end} depending on the order of the semi-infinite elements. In this case the number of elements in the x-direction N_x remains the same but the number of elements in the z-direction N_z is adjusted. The additional elements are all of order $N = 4$ but the vertical resolution Δz is allowed to change.

Linear non-hydrostatic mountain waves For this test, we modify the background parameters and the mountain profile of the previous test such that we obtain $\frac{Na_c}{u} = 1$ corresponding to nonhydrostatic flow conditions. As such, the mountain profile described in Equation (4.3) from the previous test is modified by making $a = 1000$ m. The background state is changed by taking $\mathcal{N} = 0.01$ s$^{-}1$, $\theta_0 = 280$ K, and $U = 10$ ms^{-1}. All other parameters remain the same as the previous test.

We consider a finite domain $\Omega^F = [-72, 72]$ km $\times$ $[0, 15]$ km and subdivide it with $N_x \times N_z = 40 \times 5$ elements of order 10, leading to the effective resolution $(\Delta x, \Delta z) = (360 \text{ m}, 300 \text{ m})$. A set of 40 Laguerre-Legendre semi-infinite elements are added on top of Ω^F yielding the semi-infinite domain $\Omega^S = [-72, 72]$ km$\times[15, \infty)$ km. In the horizontal direction each semi-infinite element has order 10 on LGL nodes with an order 14 Laguerre function basis on LGR nodes and a scaling factor of $\lambda = 300$ m which yields an end point $Z_{end} = 28853$. Similarly to the previous mountain wave test a Boyd-Vandeven filter [9] is used to help insure the stability of the simulation and the damping layer uses the same sine squared function as the previous test case.

Figure 4.3 shows the contours of vertical velocity for the time converged numerical solution at $t = 18,000$ s. The solution is stable, the outgoing waves are

effectively damped within Ω^S, and the structure and frequency of the wave is properly captured by the simulation. The solution is physically meaningful and comparable to other numerical solutions for the same test, see e.g [103].

Table 4.2 Timings of Linear Non-hydrostatic Mountain Wave Simulations With and Without Laguerre Semi-infinite Elements in the Absorbing Layer.

Absorbing layer type	T^*	Z_{end}	T^*_{Finite} %	$T^*_{Laguerre}$ %	N_z
Semi-infinite elements of order 14	1	28853	85.57	14.43	5
Semi-infinite elements of order 18	1.03	33279	82.49	17.51	5
Extended finite domain of order 10	1.01	30000	40	N/A	7
”	1.24	”	”	”	8
”	1.58	”	”	”	10

Note: A fixed vertical domain end point is used for simulation using only a finite domain while the simulations using semi-infinite elements are allowed to have varying Z_{end} depending on the order of the semi-infinite elements. In this case the number of elements in the x-direction N_x remains the same but the number of elements in the z-direction N_z is adjusted. The additional elements are all of order $N = 10$ but the vertical resolution Δz is allowed to change.

Table 4.2 shows the time per time step for different configurations of the linear non-hydrostatic mountain wave problem. A fixed domain end point $Z_{end} = 30,000$ m is maintained for the cases where no semi-infinite element is used and instead only the number of vertical elements is changed. For this case, even while using only three additional elements to replace the Laguerre semi-infinite element and resulting in nearly equivalent resolutions within the finite domain, the cost per time step is nearly 25% higher than using semi-infinite elements of order 18 while also having a smaller domain height. This becomes more than 50% more costly, if five additional elements are used instead of 3. Only two additional elements of order 10 can be used to replace the semi-infinite elements, if a similar time to solution is desired, but this will come at the cost of the order of accuracy within the damping layer.

In order to complete the validation of the hydrostatic and non-hydrostatic mountain wave simulations, we constructed a linear Fourier solution using the approach outlined in [101]. The vertical velocity is expressed as a Fourier integral

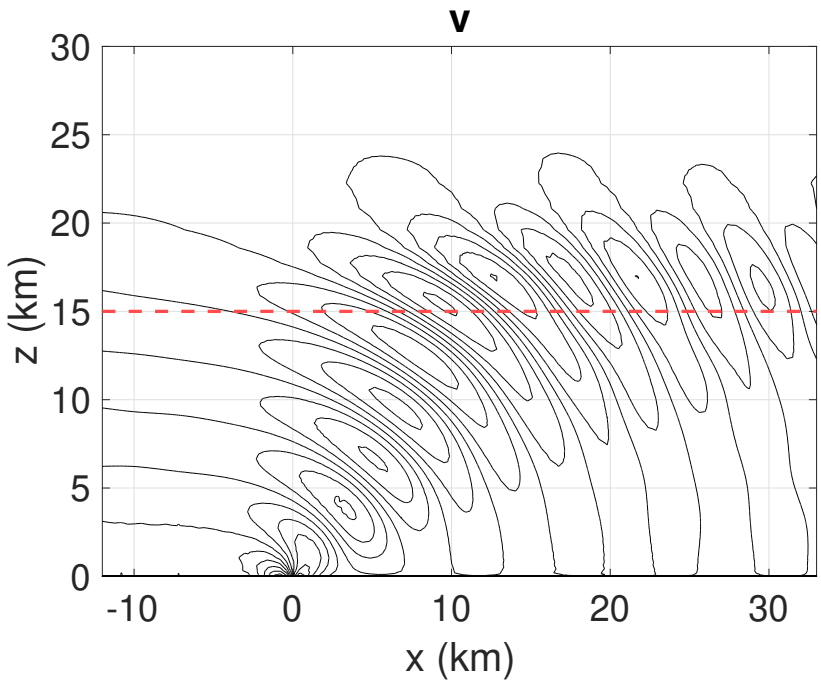


Figure 4.3 Linear non-hydrostatic mountain. Time-converged numerical solution at t=18,000 s for $\Delta x = 360$ m, $\Delta z = 300$ m. The contours are in the range between -0.005 m/s and +0.005 m/s with interval 0.001 m/s. The dashed line shows where Ω^F meets Ω^S. In Ω^S there is only one element of order 14 along the vertical direction (see Figure 2.2 for reference).

under the Boussinesq approximation that is then evaluated using adaptive Gauss quadrature in wavenumber space. Since this solution neglects vertical variations in density, it is only valid for heights less than a scale height (∼9 km). Figure 4.4 compares vertical velocity profiles of the numerically obtained solution (in blue) with the linear Fourier solution (in red). The figure shows a good overlap of the two solutions and the existing deviations are expected. This is given that the two models deviate due to the analytical solution relying on the linearization of the Euler equations and the Boussinesq approximation. Furthermore, this deviation is comparable to other atmospheric models.

Schär mountain waves The Schär test [93] consists of a uniform flow with a reference horizontal velocity $U = 10$ ms^{-1} in a stratified atmosphere with Brunt-

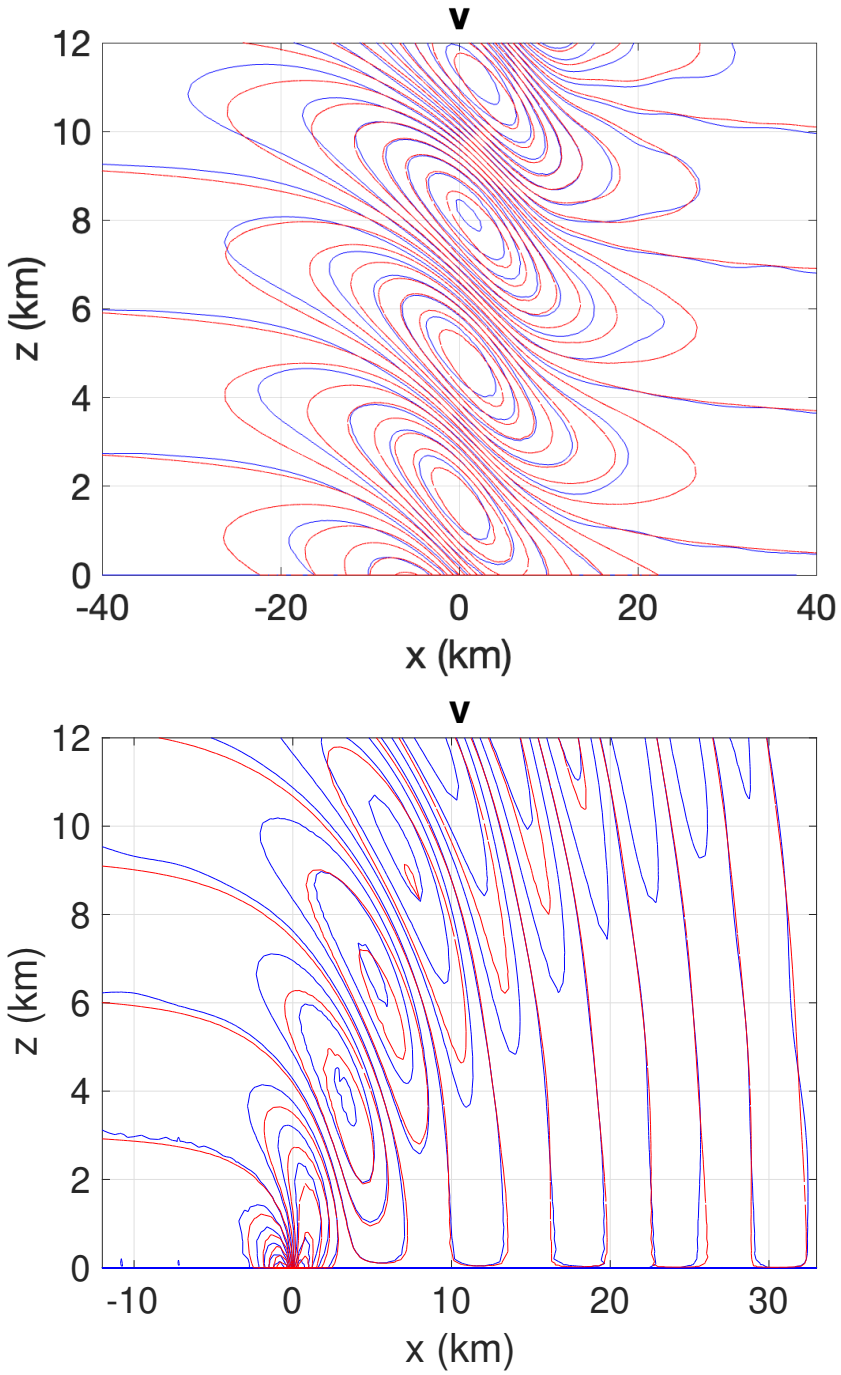


Figure 4.4 Comparison of vertical velocity contours of the numerical solution (blue) of the linear hydrostatic (top) and nonhydrostatic (bottom) mountain wave problems with an approximate linear Fourier solution (red). The contours are the same contours used in the vertical velocity plot of Figure 4.2. The numerical solutions compare favorably to the analytical approximation and display the expected vertical wavelength $\lambda_z = 2\pi U/N \sim 6.4$ km.

Väisälä frequency $\mathcal{N} = 0.01$ 1/s, sea-level pressure $p_0 = 1000$ hPa and and potential temperature at sea-level $\theta_0 = 280$ K. The flow impinges the five-peak mountain defined as

$$z = he^{-\left(\frac{x}{a}\right)^2} \cos^2\left(\frac{\pi x}{\lambda_c}\right)$$

with parameters $h = 250$ m, $a = 5000$ m, and $\lambda_c = 4000$ m.

We consider a finite domain $\Omega^F = [-25, 25]$ km $\times$ $[0, 15]$ km, and discretize it using $N_x = 20$ elements in the x direction and $N_z = 7$ elements in the z direction. These elements are of polynomial order 10 in both directions, leading to an effective resolution $(\Delta x, \Delta z) = (214 \text{ m}, 250 \text{ m})$. A set of 20 Laguerre-Legendre semi-infinite elements are added on top of Ω^F which translates to the semi-infinite domain $\Omega^S = [-25, 25]$ km $\times$ $[15, \infty)$ km. In the horizontal direction, each semi-infinite element uses an order 10 spectral element discretization using LGL nodes, and in the vertical direction it uses an order 14 Laguerre basis function on LGR nodes and a scaling factor $\lambda = 300$ m which yields an end point $Z_{end} = 28853$ m. Similarly to the previous mountain wave tests a Boyd-Vandeven filter [9] is used to help insure the stability of the simulations and the a sine squared function is used in the damping coefficient of the Rayleigh damping layer.

Figure 4.5 shows the vertical velocity of the time converged numerical solution at $t = 36,000$ s. The solution is stable and outgoing waves are effectively damped withing Ω^S. The solution is also physically meaningful and comparable to other numerical solutions of atmospheric models, see e.g [93]. Figure 4.6 shows an overlap of the numerical vertical velocity with a linear Fourier solution under the anelastic approximation [56, Equation (A10)]. The figure shows a good overlap of the two solution and the existing deviations are within expectations given the

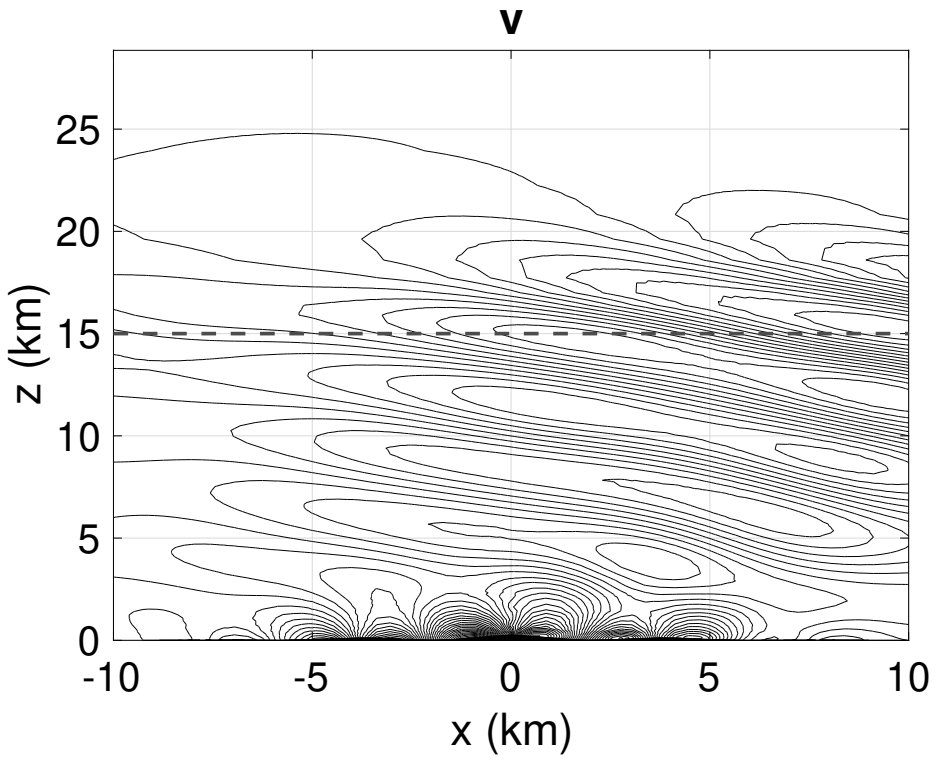


Figure 4.5 Schär mountain waves. Time-converged vertical velocity at t=36,000 s for $\Delta x = 250$ m, $\Delta z = 220$ m. The contours of the vertical velocity range between -2.0 m/s and +2.0 m/s with interval 0.1 m/s. The dashed line shows where Ω^F meets Ω^S. In Ω^S there is only one element of order 14 along the vertical direction (see Figure 2.2 for reference).

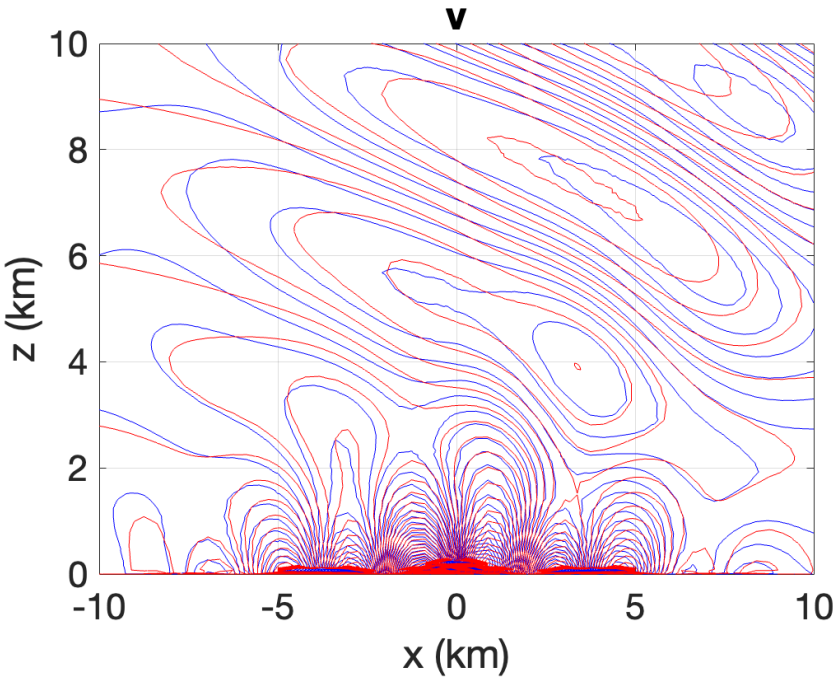


Figure 4.6 Comparison of vertical velocity contours of the numerical solution (blue) and semi-analytical Fourier solution (red) for the Schär mountain test. As for those shown in Figure 4.4, the discrepancy is due to the linearization of the Euler equations and the anelastic approximation used to calculate the semi-analytic Fourier solution.

Table 4.3 Timings of Schär Mountain Wave Simulations With and Without Laguerre Semi-Infinite Elements in the Absorbing Layer.

Absorbing layer type	T^*	Z_{end}	T^*_{Finite} %	$T^*_{Laguerre}$ %	N_z
Semi-infinite elements of order 14	1	28853	88.62	11.38	7
Semi-infinite elements of order 18	1.03	33279	86.02	13.98	7
Extended finite domain of order 10	0.92	21000	100	N/A	8
”	1.03	”	”	”	9
”	1.14	”	”	”	10
”	1.27	”	”	”	11

Note A fixed vertical domain end point is used for simulation using only a finite domain while the simulations using semi-infinite elements are allowed to have varying Z_{end} depending on the order of the semi-infinite elements. In this case the number of elements in the x-direction N_x remains the same but the number of elements in the z-direction N_z is adjusted. The additional elements are all of order $N = 10$ but the vertical resolution Δz is allowed to change.

differences between the models. Furthermore, these deviations are comparable to other atmospheric models.

Table 4.3 shows the time per time step for different configurations of the schär mountain problem. A fixed domain end point $Z_{end} = 21,000$ m is maintained for the cases where no semi-infinite element is used and instead only the number of vertical elements is changed. For this case, even while using only four additional elements to replace the Laguerre semi-infinite element and resulting in nearly equivalent resolutions within the finite domain, the cost per time step is nearly 25% higher than using semi-infinite elements of order 18 while also having a smaller domain height. Only two additional elements of order 10 can be used to replace the semi-infinite elements, if a similar time to solution is desired, but this will come at the cost of the order of accuracy within the damping layer.

CHAPTER 5

NON-COLUMN BASED SIMULATIONS OF THUNDERSTORMS

In this chapter, we first present the approach utilized to obtain a non-column based implementation of the Kessler warm rain microphysics. We then show how this approach performs when used for simulating squall lines and a fully three-dimensional super-cell thunderstorm. Simulations in this chapters are performed on a moist atmosphere using the Non-hydrostatic Unified model for the atmosphere (NUMA) [51].

5.1 Non-column Based Rain Sedimentation

The main novelty of the study presented in this chapter lies in the computation of the sedimentation term for the rain equation (i.e., the last term on the right-hand side in Equation (3.24)) which differs from the methods in, e.g., [52, 53, 102, 87, 45]. The typical column-based approach to handle the sedimentation term is by computing the spatial derivative along each individual column starting from the top of the domain and descending. See, e.g., [23, 74] for a spectral element implementation of this approach.

Although widely used, the traditional column-based implementation has a main drawback: it requires the availability of column-aware data structures that may not serve other purposes in the numerical method, thereby forcing the use of structured grids. Unstructured grids are highly advantageous around topography. By forgoing the use of columns, our approach to compute sedimentation could help yield more accurate predictions for storm behavior in mountainous regions.

Computing the sedimentation term is done separately from the other micro-physics calculations, and is done after solving the compressible Euler and moisture

advection equations. This term is included by solving the following equation:

$$\frac{\partial q_r}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}(\rho q_r w_r) \tag{5.1}$$

in non-conservative form and

$$\frac{\partial(\rho q_r)}{\partial t} = \boldsymbol{\nabla}\cdot(\rho q_r w_r \hat{\mathbf{k}}) \tag{5.2}$$

in conservation form. Given that $\hat{\mathbf{k}} = (0,0,-1)^T$ for the domains we consider, Equation (5.2) can be written as follows:

$$\frac{\partial(\rho q_r)}{\partial t} = \frac{\partial}{\partial z}(\rho q_r w_r).$$

This makes it so that for either the conservative or non-conservative form, solving the sedimentation equation essentially amounts to calculating the term $\frac{\partial}{\partial z}(\rho q_r w_r)$.

We can rewrite the sedimentation equation in the form of Equation (2.1) by taking $G(f) = -c\frac{\partial F_{\text{sed}}}{\partial z}$, where $F_{\text{sed}} = (\rho q_r w_r)$, $c = 1$ and $f = \rho q_r$ in conservation form, while $c = \frac{1}{\rho}$ and $f = q_r$ in non-conservative form. By multiplying by the expansion functions and integrating, we get:

$$\int_{\Omega_e} \psi_i(\mathbf{x})\frac{\partial f^h(\mathbf{x},t)}{\partial t}d\mathbf{x} = \int_{\Omega_e}\psi_i(\mathbf{x})\sum_{j=1}^{(N+1)^3}\frac{\partial\psi_j(\mathbf{x})}{\partial z}c\hat{F}^e_{j,sed}(t)(\mathbf{x})d\mathbf{x}, \quad i=1,\ldots,(N+1)^3,$$

where $\hat{F}_{j,sed}$ are the expansion coefficients of $F_{j,sed}$. Moving to the reference element and identifying the mass matrix yields

$$\mathbf{M}^e_{ij}\frac{\partial\hat{f}^e_j(t)}{\partial t} = \int_{\Omega_{ref}}\psi(\boldsymbol{\xi})\left[\boldsymbol{\nabla}_\xi\psi_j(\boldsymbol{\xi})\cdot\left(\frac{\partial\xi}{\partial z},\frac{\partial\eta}{\partial z},\frac{\partial\zeta}{\partial z}\right)(\boldsymbol{\xi})\right]c\hat{F}^e_{j,sed}(t)|\mathbf{J}(\boldsymbol{\xi})|d\boldsymbol{\xi}, \tag{5.3}$$

where $i,j = 1,\ldots,(N+1)^3$. Let us call $\mathbf{D}^e_{\text{sed}}$ the element-wise differentiation matrix for Equation (5.1) and write Equation (5.3) in matrix form:

$$M^e_{ij}\frac{\partial\hat{f}^e_j(t)}{\partial t} = \mathbf{D}^e_{ij,\text{sed}}\cdot c\hat{\mathbf{F}}^e_{j,sed}(t), \quad i,j=1,\ldots,(N+1)^3. \tag{5.4}$$

We can write $\mathbf{D}^e_{\text{sed}}$ discretely as follows:

$$\begin{aligned}\mathbf{D}^e_{ij,\text{sed}} = \sum_{k=1}^{N+1}\sum_{m=1}^{N+1}\sum_{n=1}^{N+1} \omega(\xi_k,\eta_m,\zeta_n)\psi_i(\xi_k,\eta_m,\zeta_n)\boldsymbol{\nabla}_\xi\psi_j(\xi_k,\eta_m,\zeta_n) \\ \cdot\left(\frac{\partial\xi}{\partial z},\frac{\partial\eta}{\partial z},\frac{\partial\zeta}{\partial z}\right)(\xi_k,\eta_m,\zeta_n)|\mathbf{J}(\xi_k,\eta_m,\zeta_n)|,\end{aligned} \tag{5.5}$$

where $i,j = 1,\ldots,(N+1)^3$. From this point, if CG is used, then the global equation can be solved using DSS as follows:

$$\frac{\partial \mathbf{f}^h}{\partial t} - \mathbf{M}^{-1}\mathbf{D}_{\text{sed}}(\mathbf{c}\odot\mathbf{F}^h_{\text{sed}}) = 0, \tag{5.6}$$

where $\mathbf{D}_{\text{sed}} = \sum_{e=1}^{N_e}\mathbf{D}^e_{\text{sed}}$, $\mathbf{c}$ is the vector containing the nodal values of c, $\mathbf{F}_{\text{sed}}$ is the vector containing the nodal values of F_{sed}, and $\odot$ denotes a component-wise multiplication (i.e., the Hadamard product). The local DG problem is given by:

$$\frac{\partial \mathbf{f}^h}{\partial t} - \mathbf{M}^{-1(e)}(\hat{\mathbf{D}}^e_{\text{sed}}(\mathbf{c}\odot\mathbf{F}^h_{\text{sed}}) - \mathbf{F}^{\mathbf{e}}(\mathbf{c}^*\odot\mathbf{F}^*_{\text{sed}})), \tag{5.7}$$

where $\hat{\mathbf{D}}^e_{\text{sed}}$ is the weak form of $\mathbf{D}^e_{\text{sed}}$, $\mathbf{F}^e$ is the flux matrix at each element, $\mathbf{F}^*_{\text{sed}}$ is the interface value of $\mathbf{F}_{\text{sed}}$, and $\mathbf{c}^*$ is the interface value of $\mathbf{c}$.

In what follows, we present the procedure we use to solve the fully compressible Euler equations with moisture, including rain. Algorithm 1 summarizes the entire procedure. The algorithm makes use of the following quantities: $N_{LGL} = N+1$ is the number LGL points in each element, Δt is the time step, $f^{h,n}$ is the approximation of f^h at the time $t^n = n\Delta t$, N_{points} the total number of points the domain has been discretized into including repeating nodes at element edges and faces, t_n the current discrete time, and $t_{n+1} = t_n + \Delta t$. We also define the sedimentation Courant number, which we use to determine the time sub-step for the sedimentation problem:

$$Cr = w_r\frac{\Delta t}{\Delta z}. \tag{5.8}$$

This number is used to determine the appropriate sedimentation time step as follows:

$$\Delta t_{\text{sed}} = \frac{\Delta t}{\max(1, 0.5 + Cr_{\max}/Cr_{\text{limit}})}, \tag{5.9}$$

where:

$$Cr_{\max} = \max([Cr_i]_{i=1}^{N_{points}}), \tag{5.10}$$

is the maximum sedimentation Courant number among all points in the domain and Cr_{limit} is the maximum allowable Courant number for the sedimentation problem. The rest of the notation is defined in Section 2.1.

Next, we report on the results obtained with this algorithm and fully unstructured grids.

5.2 Results

We assess the method presented in Section 5.1 with an idealized squall line test from [23] and a fully 3D supercell problem. All the simulations are run with the Nonhydrostatic Unified Model of the Atmosphere (NUMA) [51], which is designed to solve the dry Euler equations, with the addition of artificial viscosity as described in Section 2.1, on unstructured grids of hexahedra with arbitrary orientation. NUMA enables the use of both CG and DG spectral elements and has been shown to scale exceptionally well on CPUs and GPUs in [1, 2, 82].

5.2.1 2.5D squall line

The first benchmark we consider is an idealized test presented in [23]. While the computational domain in [23] is two-dimensional, we run the same test in a 2.5 D domain $\Omega = [150 \times 12 \times 24]$ km^3. The domain is discretized with a single element in the y direction and a resolution dependent number of elements in the x and z directions. Periodic boundary conditions are applied to the lateral boundaries, a

Algorithm 2 Simulation of moist-air and rain sedimentation with unstructured grids.

for $time = 0, \Delta t, \ldots, t_f$ **do**
  for $e = 1, 2, \ldots, N_e$ **do**
    for $node = 1, 2, \ldots, N_{LGL}$ **do**
      Calculate contributions to element-wise derivatives from each LGL point along
      the reference element.
    end for
    Compute these local derivatives in physical space.
  end for
  Perform DSS for CG or calculate numerical fluxes for DG.
  Solve the discrete version of the Euler equations: (3.2), (3.3) and (3.4), if using conservation
  form, and (3.2), (3.5) and (3.6), if using non-conservative form.
  Solve the advection equations for q_v, q_c and q_r by the flow velocity u: Equations (3.18), (3.19) and
  Equations (3.23), if using conservation form, and (3.20), (3.21) and (3.24), if using non-conservative form.
  for $i = 1, 2, \ldots, N_{points}$ **do**
    Determine w_r using Equation (3.29)
    Determine $Cr_{\max}$ using Equation (5.10)
    Determine Δt_{sed} using Equation (5.9)
  end for
  for ts = t_n,$t_n + \Delta t_{\text{sed}}$,. . . ,$t_{n+1}$ **do**
    for $e = 1, 2, \ldots, N_e$ **do**
      if space method == CG
        Compute $\mathbf{D}^e_{\text{sed}}$
      else if space method == DG
        Compute $\hat{\mathbf{D}}^e_{\text{sed}}$
      end if
    end for
    if space method == CG
      Perform DSS.
    else if space method == DG
      Apply inter-element fluxes for the sedimentation equation using w_r as the wave
      speed.
    end if
    Solve Equation (5.1)
  end for
  for $e = 1, 2, \ldots, N_e$ **do**
    Update moisture variables and potential temperature to account for phase changes
    following Equations (3.28)-(3.27)
  end for
end for

free-slip type boundary condition is applied at the domain bottom and the domain top utilizes a Rayleigh sponge for gravity wave damping. In this domain, a squall line forms in a weakly stable atmosphere with Brunt-Väisälä frequency $N = 0.01$ s^{-1} below the tropopause and a more stable atmosphere with $N = 0.02$ s^{-1} above 12 km. The cloud begins to form around $t \approx 500$ s, while rain starts to form and fall at approximately $t \approx 900$ s. The initial condition consists of a saturated boundary layer typical of mid-latitude storms that has been used in several numerical studies (see, e.g, [92] and [110]). A low altitude wind shear in the x direction is imposed to break the cloud symmetry and allow for a continuous storm evolution. The initial background sounding is tabulated in the Appendix.

The storm is triggered by a thermal perturbation of the background state [92] centered at $(x_c, z_c) = (75000, 2000)$ m and defined by:

$$\Delta\theta = \begin{cases} \theta_c \cos\left(\frac{\pi r}{2}\right) & \text{if } r \leq r_c, \\ 0 & \text{if } r \geq r_c, \end{cases} \tag{5.11}$$

where

$$r = \sqrt{\frac{(x - x_c)^2}{r_x^2} + \frac{(z - z_c)^2}{r_z^2}}, \quad \theta_c = 3 \text{ K}, \quad r_c = 1, \quad r_x = 10000 \text{ m}, \quad r_z = 1500 \text{ m}.$$

We generated seven grids using GMSH [25]. Table 5.1 lists the total number of hexahedral elements and the effective resolution Δx for each mesh. We choose to report the effective resolution because the LGL points for an element are not equidistant [26, 41, 61]. NUMA relies on P4est [14] to read unstructured meshes and perform the graph partitioning for the parallel application.

Figure 5.1 shows an example of clouds and precipitation calculated on a fully unstructured grid of hexahedra for an effective resolution of 150 m in both spatial directions.

Table 5.1 Total Number of Hexahedral Elements, Effective Resolutions, and Time Steps for All the Meshes Used for the Squall Line Simulations.

# elements	473	1078	3181	4134	6485	11447	25863
Δx	750 m	500 m	290 m	250 m	200 m	150 m	100 m
Δt	0.9 s	0.6 s	0.35 s	0.3 s	0.24 s	0.18 s	0.12 s

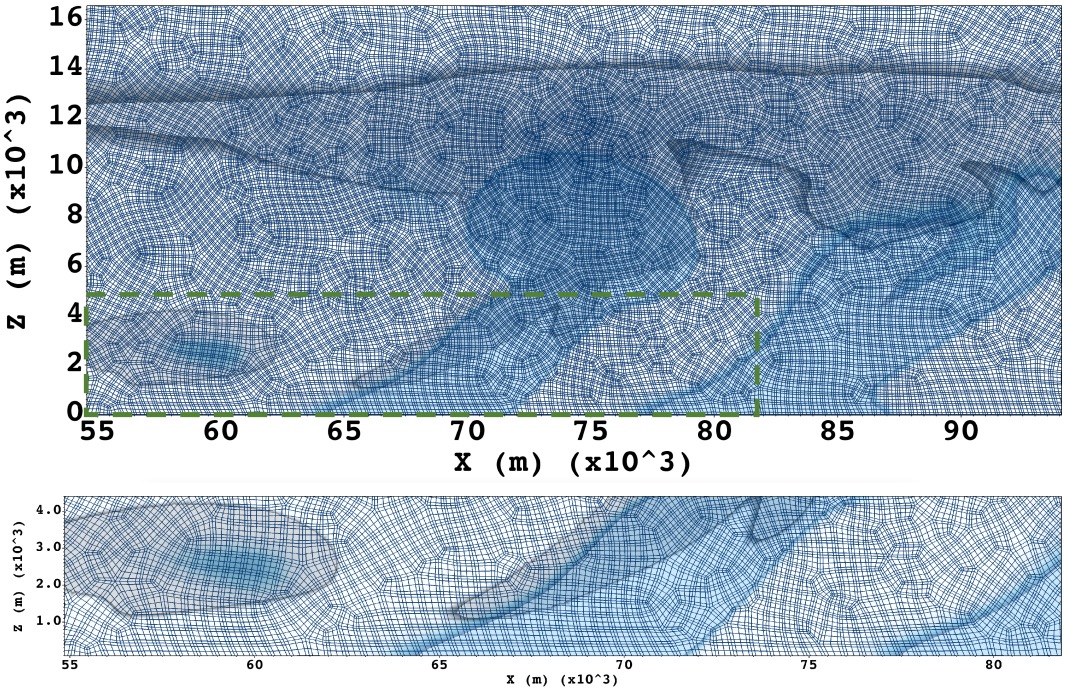


Figure 5.1 Top: q_c and q_r over unstructured grid $\Delta x = 150$ m. Cloud water is shaded in grey for values of $q_c > 1 \times 10^{-5}$ kg/kg whereas rain is shaded in blue for values of $q_r > 1 \times 10^{-4}$ kg/kg. Bottom: close-up view corresponding to the dashed rectangle in the top figure.

For all the simulations, we use an Additive Runge Kutta third order (ARK3) semi-implicit time integrator and elements of polynomial order 4. We maintain the acoustic Courant number $C \leq 1$ for all the simulations. While the ARK3 time integrator allows for larger acoustic Courant numbers, we limit the time step for the purposes of obtaining a greater deal of accuracy for the higher-resolution simulations.

We run this test using both the CG approach with the governing equations in non-conservation form and the DG approach with the governing equations in conservation form. Consistently with [23], a constant artificial viscosity of $\beta = 200$ (for the units see Remark 3.1.1) is used to stabilize the simulations.

Let us examine the results obtained with the finest mesh, i.e., the one with $\Delta x =$ 100 m. Figures 5.2 and 5.3 show the stages of the storm evolution given by the CG and DG simulations, respectively. Both simulations yield similar plots at $t = 1500$ s. Additionally, in both cases we observe a downwind tilt of the convective tower, which is caused by the horizontal wind-shear, and the eventual development of the anvil cloud near the tropopause where the atmosphere offers higher stability. For the sake of brevity, we do not show the plots associated with other meshes, but a similar early storm evolution is observed in all the simulations at all resolutions with both the CG and DG approaches. The differences between the CG and DG simulations remain minimal even up to about $t = 6000$ s. This is a rather long period of time since by then the storm has fully developed. Starting from $t = 6000$ s till the end of the time interval of interest, some differences in the CG and DG simulations arise, as can been seen by comparing Figures 5.2 and 5.3. At $t = 9000$ s, when additional convective towers are observed, the DG simulation generates multiple convective towers, some of which are significantly downwind. This is not as pronounced in the CG simulation. Compare the bottom right panels in Figures 5.2 and 5.3.

Figures 5.2 and 5.3 also show the rain accumulated on the ground. At $t =$ 1500 s, no rain has accumulated yet in either the DG or CG simulations. This is confirmed by the rain contour plots, where we see that the contour lines have yet to reach the ground (see top left panel in Figures 5.2 and 5.3). At $t = 3000$ s, the accumulated rain is primarily near the center of the domain for both methods. Indeed, from the top right panel in Figures 5.2 and 5.3 we see that rain accumulates at the

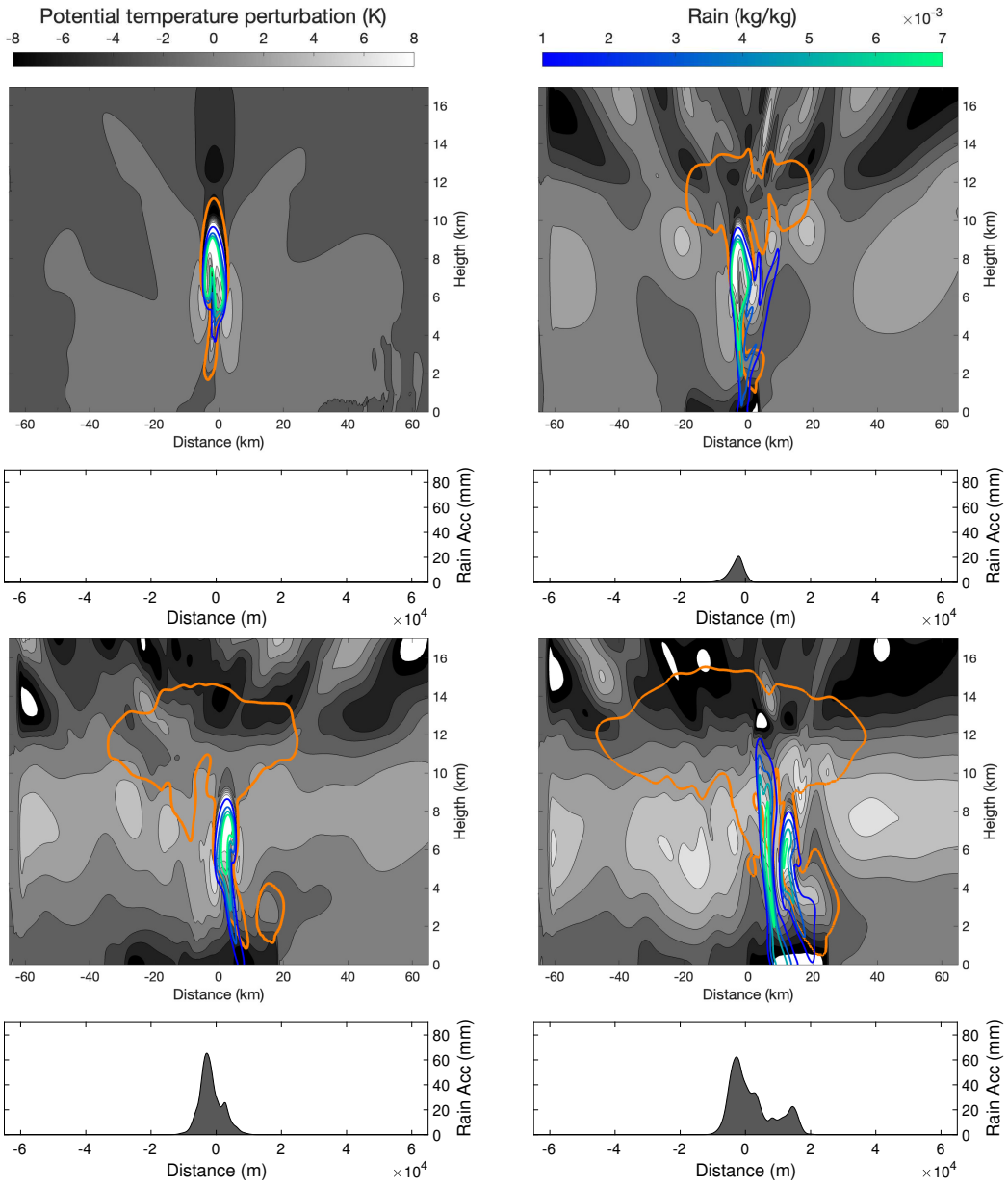


Figure 5.2 Storm evolution obtained with a CG approximation and mesh with resolution $\Delta x = 100$ m at t =1500 s (top-left), 3000 s (top-right), 6000 s (bottom-left) and 9000 s (bottom-right). In the top portion of each panel, the thick orange contour line ($q_c = 10^{-5}$ kgkg^{-1}) represents the outline of the cloud. The white and gray contours represent the perturbation potential temperature, and the blue and green contours represent q_r. The bottom portion of each panel shows the rain accumulated at the surface for each time as a function of horizontal distance from the point $x = 0$ m.

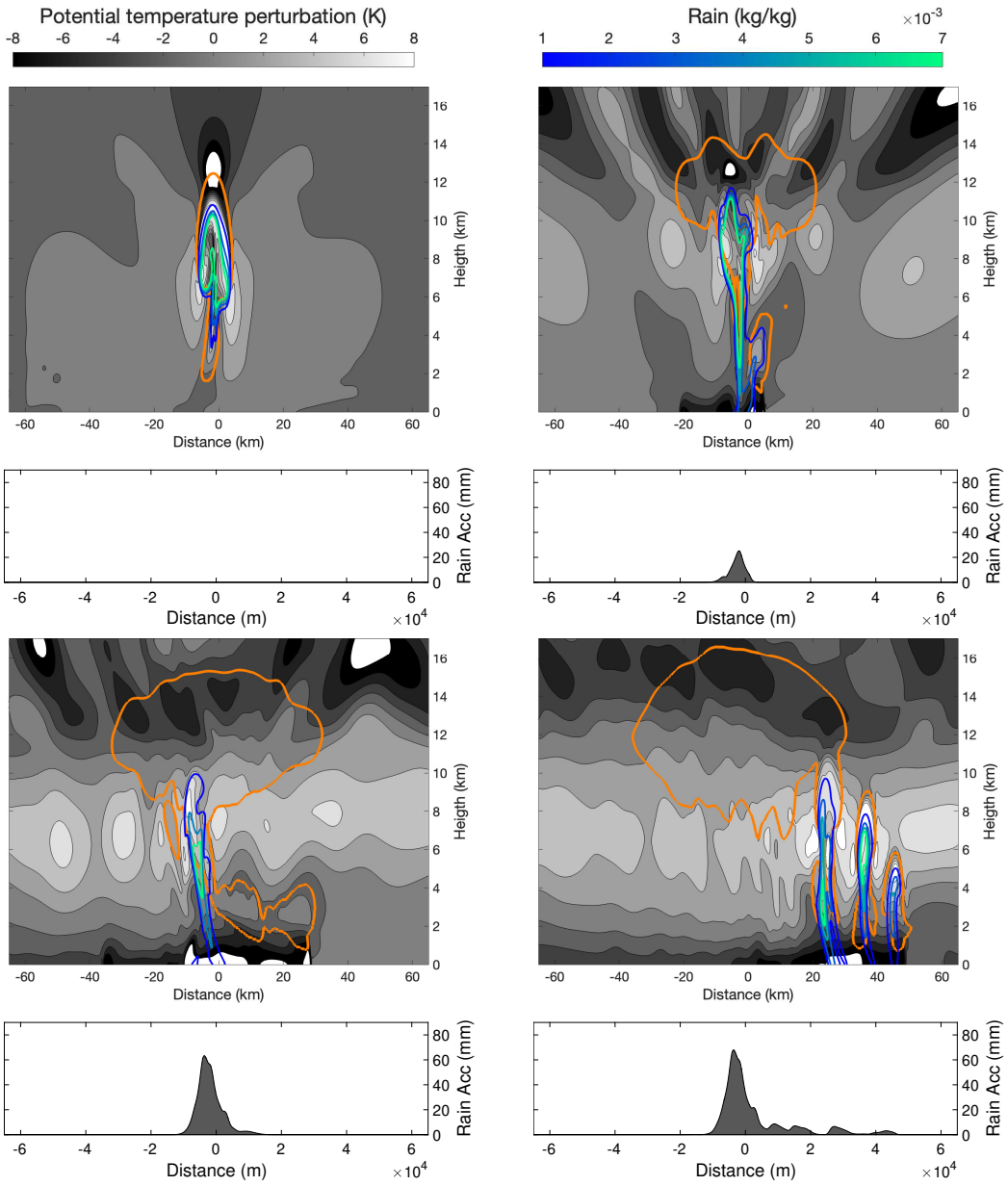


Figure 5.3 Storm evolution obtained with a DG approximation and mesh with resolution $\Delta x = 100$ m at t =1500 s (top-left), 3000 s (top-right), 6000 s (bottom-left) and 9000 s (bottom-right). In the top portion of each panel, the thick orange contour line ($q_c = 10^{-5}$ kgkg^{-1}) represents the outline of the cloud. The white and gray contours represent the perturbation potential temperature and the blue and green contours represent q_r. The bottom portion of each panel shows the rain accumulated at the surface for each time as a function of horizontal distance from the point $x = 0$ m.

location of the convective tower, with a slight asymmetry that follows the asymmetry of the convective tower seen at $t = 1500$ s. As time progresses, the convective tower tilts. An early stage of this is visible at $t = 3000$ s, but the tilting becomes more pronounced at $t = 6000$ s when the effect of the wind shear is more noticeable. The rain accumulation reflects the tilting and location of the convective tower in both the CG and DG simulations, as shown in the bottom left panel of Figures 5.2 and 5.3. By $t = 9000$ s, we observe once again some differences in the results given by the two methods. For the CG simulation, in the bottom right panel of Figure 5.2 we see a much wider distribution of accumulated rain with a secondary peak below the new location of the convective tower and a third peak appearing below the location of the secondary convective tower. As for the DG simulation, in the bottom right panel of Figure 5.3, we notice that the rain accumulation matches the downwind shifting of the main column and small peaks appear where secondary convective towers are present.

Regardless of the space discretization method, we see that once rain appears within the convective tower it is correctly transported downward without the need for a vertically structured grid. This holds true also when multiple, possibly disconnected, sources of rain are present in the domain. In both sets of simulations, the rain falls to the ground following the location of the convective towers and the effects of the wind-shear. This gives us confidence that our algorithm is able to correctly transport rain despite the lack of a vertically structured grid and regardless of the space discretization method.

The results obtained with the $\Delta x = 250, 200, 150, 100$ m meshes at $t = 9000$ s are compared in Figure 5.4 for the CG approximation and in Figure 5.5 for the DG approximation. In Figure 5.4, we observe the same cloud structure (anvil extent, downwind tilt of the convective tower) and similar profiles of perturbation potential

temperature for all the meshes under consideration. However, the spatial distributions of the rainfall accumulated at the ground show some differences: the simulations with resolutions $\Delta x = 250$ m and $\Delta x = 200$ m have smaller peaks of rain accumulation near the domain center than the simulations with $\Delta x = 150$ m and $\Delta x = 100$ m. The simulations with the $\Delta x = 290, 500, 750$ m meshes (not shown for brevity) give even more intense rainfall than the $\Delta x = 250$ m and $\Delta x = 200$ m simulations. A similar observation on rain accumulation and mesh resolution for this benchmark can be found in [112, 23], where it is shown that higher resolutions are correlated with faster storm development, weaker storm circulation and less overall precipitation over the length of the simulation. The DG simulations also show similar tilt in the convective tower, similar anvil extents and similar profiles of perturbation potential temperature at $t = 9000$ s for all the meshes; see Figure 5.5. Concerning the rain accumulation, the DG simulation with the $\Delta x = 250$ m mesh gives a large primary and secondary peak near the center of the domain. The amount of rain falling at the domain center decreases with increasing resolution. Indeed, the $\Delta x = 200, 150$ m simulations give a smaller amount of accumulated rain in the domain center and slightly larger peaks downwind and away from the center, reflecting the availability of more moisture for the secondary convective tower. Once again, we observe a decrease in precipitation with increasing resolution as expected [23, 112, 73, 70].

We conclude by reporting the maximum vertical velocity obtained over the course of the CG and DG simulation as a function of the resolution in Figure 5.6. We see that for $\Delta x \geq 290$ m the maximum vertical velocity for both DG and CG simulations lies between 20 ms^{-1} and 30 ms^{-1}, as in [13, 111, 23]. Increasing the resolution yields an increase in the maximum velocity, as shown in [23]. We note that the CG and DG simulations give similar values of the maximum vertical velocity for a given mesh, with the values getting closer as the resolution increases.

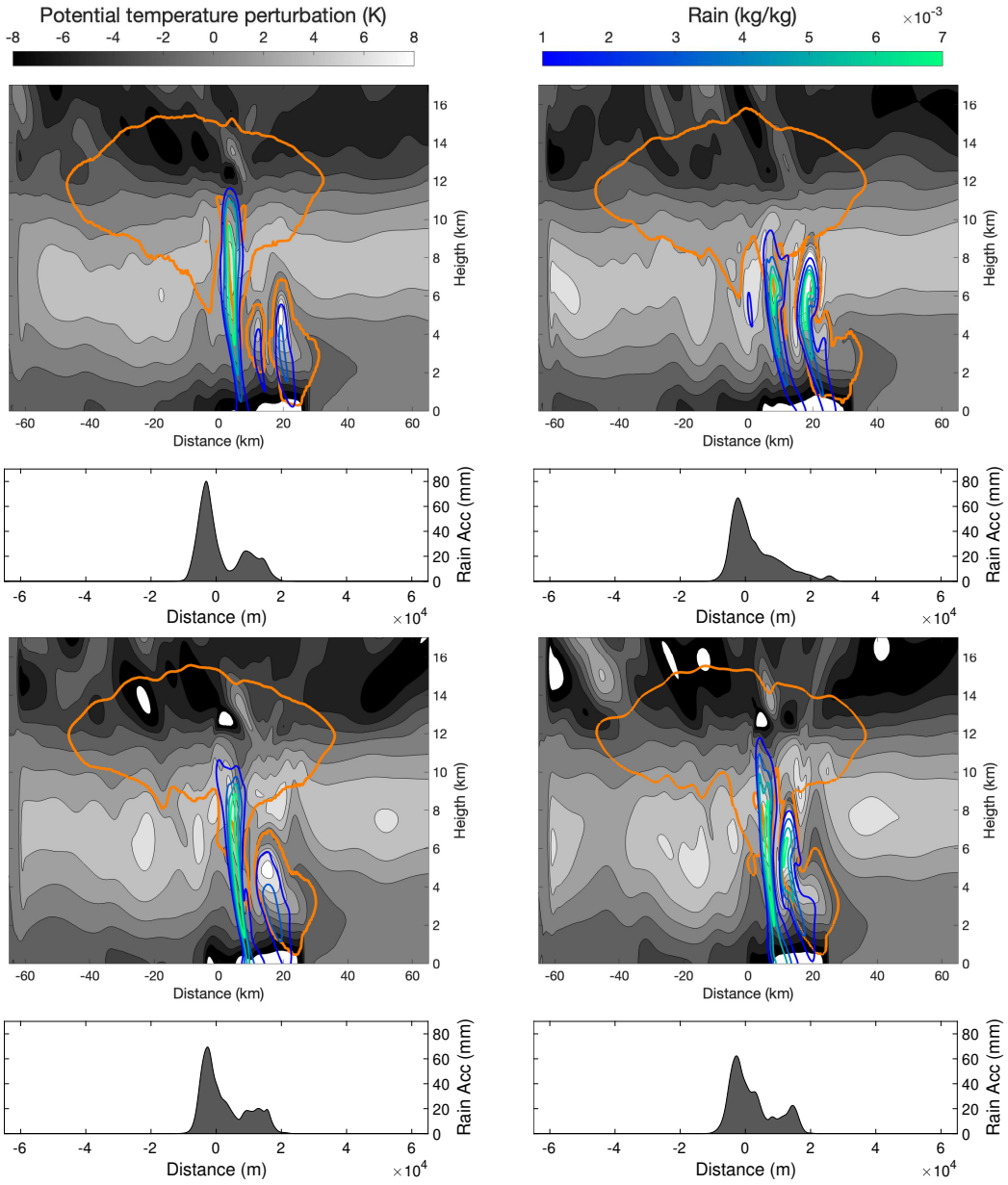


Figure 5.4 Storm at $t = 9000$ s computed with the CG method and meshes $\Delta x = 250$ m (top-left), $\Delta x = 200$ m (top-right), $\Delta x = 150$ m (bottom-left), and $\Delta x = 100$ m (bottom-right). The thick orange contour line ($q_c = 10^{-5}$ kgkg^{-1}) represents the outline of the cloud. The white and gray contours represent the perturbation potential temperature and the blue and green contours represent q_r. The bottom portions of each panel show the rain accumulated at the surface as a function of horizontal distance from the point $x = 0$ m.

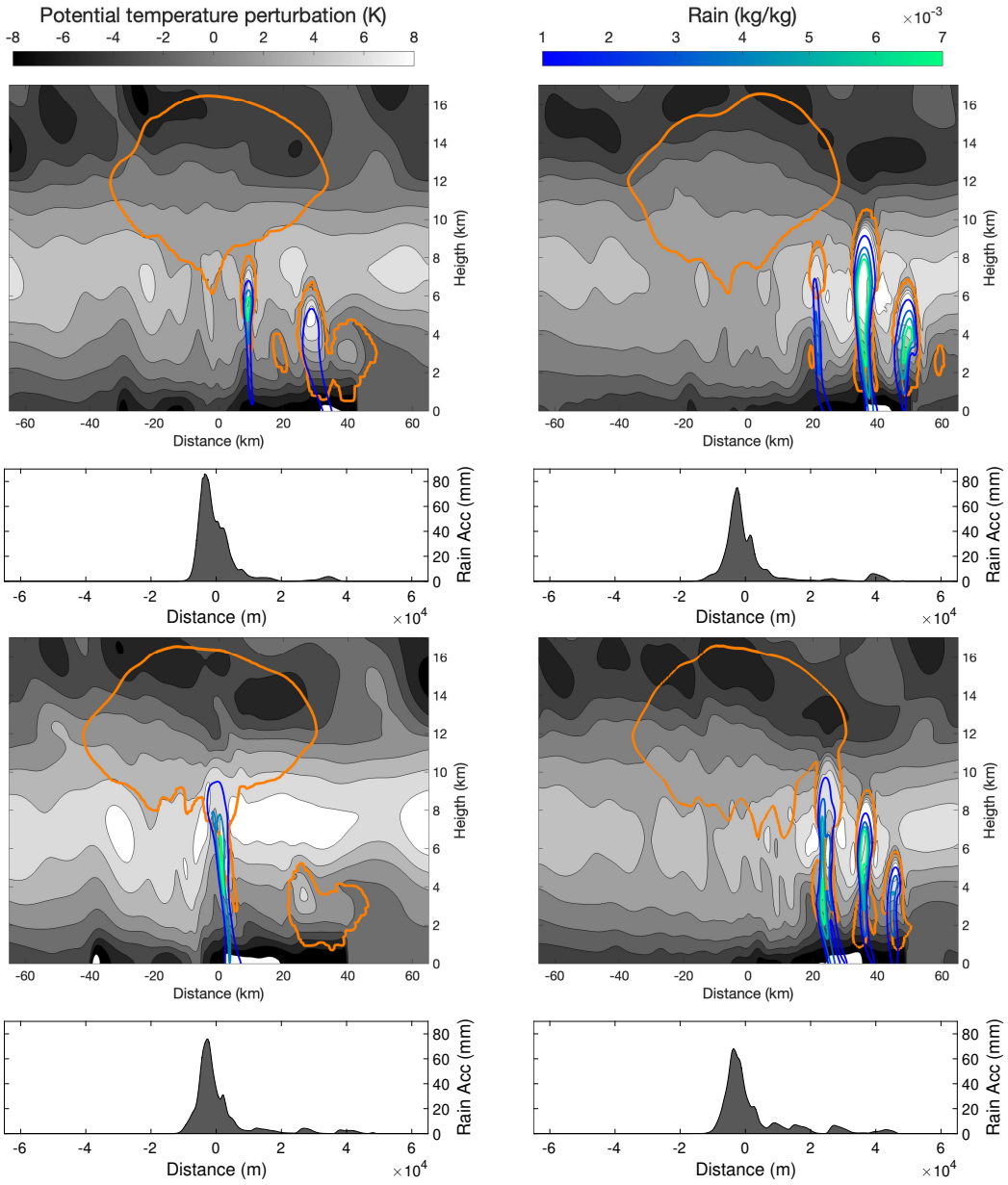


Figure 5.5 Storm at $t = 9000$ s computed with the DG method and meshes $\Delta x = 250$ m (top-left), $\Delta x = 200$ m (top-right), $\Delta x = 150$ m (bottom-left), and $\Delta x = 100$ m (bottom-right). The thick orange contour line ($q_c = 10^{-5}$ kgkg^{-1}) represents the outline of the cloud. The white and gray contours represent the perturbation potential temperature and the blue and green contours represent q_r. The bottom portions of each panel show the rain accumulated at the surface as a function of horizontal distance from the point $x = 0$ m.

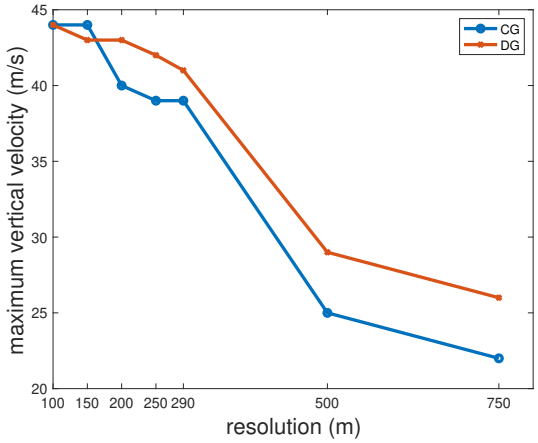


Figure 5.6 Maximum vertical velocity obtained over the course of the CG and DG simulations as a function of the resolution.

The results in this section demonstrate that, in two dimensions, our algorithm successfully transports the rain downwards along the convective towers without the need for a vertically structured grid.

5.2.2 3D supercell

In this section, we test our algorithm for a fully three-dimensional supercell. The convective cell develops within a domain $\Omega = [150 \times 100 \times 24]$ km^3. The storm is initiated by a thermal perturbation of the background state defined by Equation (5.11), with center $(x_c, y_c, z_c) = (75000, 50000, 2000)$ m and

$$r = \sqrt{\frac{(x - x_c)^2}{r_x^2} + \frac{(y - y_c)}{r_y^2} + \frac{(z - z_c)^2}{r_z^2}}, \quad \theta_c = 3 \text{ K}, \quad r_c = 1,$$

where:

$$r_x = r_y = 10000 \text{ m}, \quad r_z = 2000 \text{ m}.$$

The domain is discretized using a grid of unstructured hexahedra of order four in all directions for an approximate effective resolution $\Delta\mathbf{x} \approx 250$ m. The grid is partially shown in Figure 5.7.

We use periodic boundary conditions for the lateral boundaries, a free-slip boundary at the domain bottom and a Rayleigh sponge at the domain top. Like for the squall line test described above, we use the ARK3 3D semi-implicit time integrator to advance the simulation in time and keep the acoustic Courant number $C \leq 1$ (We use $\Delta t = 0.3$ s for the test shown here). An artificial viscosity $\beta = 200$ (see Remark 3.1.1 for the units) is used to provide stabilization. The wind shear in the x direction is the same as the one used for the squall-line. The cloud begins to form at $t \approx 500$ s while rain forms and starts to precipitate at $t \approx 900$ s.

A 3D view of the fully developed storm at $t = 7200$ s is shown in Figure 5.7, along with a partial view of the three-dimensional grid. The semi-transparent blue shading is the iso-surface $q_r = 10^{-4}$ kg/kg. The blue shading is the perturbation potential temperature (blue is negative) showing the cold pools due to rain evaporation. All of the convective towers exhibit tilting due to wind-shear, with the parts closer to the ground experiencing a greater wind-shear and thus trailing the rest of the convective tower. An anvil cloud is also observed near the top of the troposphere.

Figure 5.8 shows the state of the storm at $t = 7200$ s. The right side of the figure shows the existence of three distinct convective towers in the supercell. One in the center of the Y axis at $y = 50000$ m and two columns symmetric about the $y = 50000$ m plane. The three towers merge into the anvil cloud near the tropopause. Figure 5.8 (left) shows the rain distribution at the ground at $t = 7200$ s. The position of the rain concentration follows the location of the convective towers, falling below them. The largest amount of rain is present below the larger central tower as indicated by the maximum over $y = 50000$ m. Additionally we can see the presence

of some rain slightly separated from the main rain distribution which corresponds to the small low clouds that are shown symmetric to the $y = 50000$ m plane in the right side of the figure.

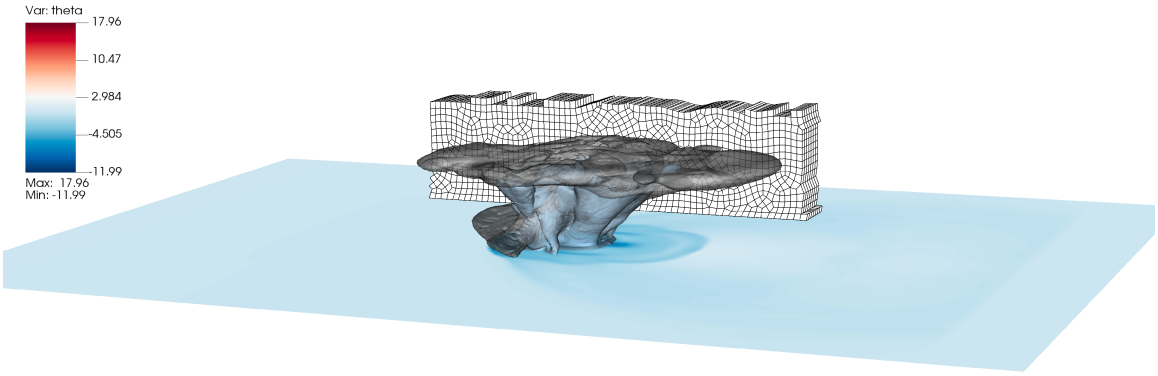


Figure 5.7 3D mature supercell at $t = 7200$ s. The grey shading is the iso-surface $q_c = 10^{-5}$ kg/kg. The semi-transparent blue shading is the iso-surface $q_r = 10^{-4}$ kg/kg. The blue shading is the perturbation potential temperature (blue is negative) showing the cold pools due to rain evaporation. A small sample of the three-dimensional unstructured grid is shown in the background.

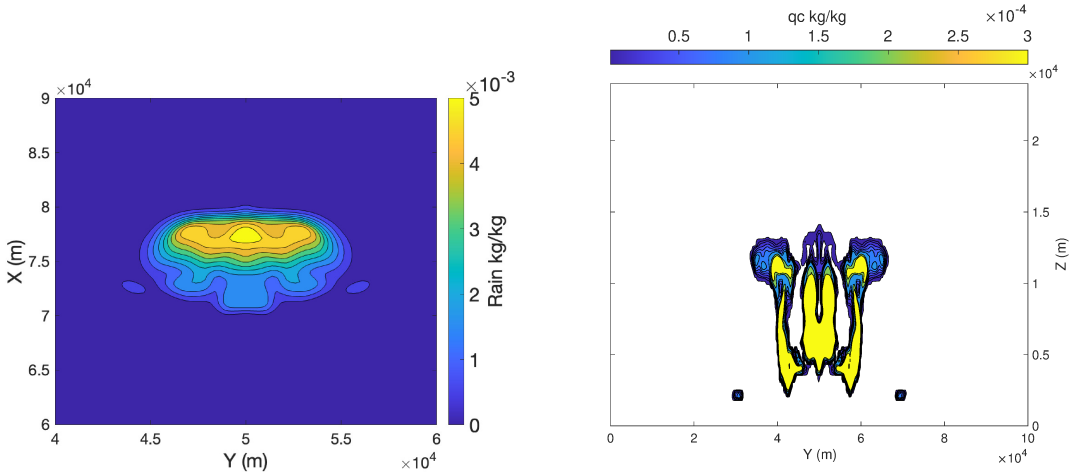


Figure 5.8 3D mature supercell at $t = 7200$ s. Left: Horizontal cross-section of the instantaneous distribution of rain along the surface ($z = 0$ m) at $t = 7200$ s. Right: Vertical cross section taken at $x = 75000$ m of the cloud fraction $t = 7200$ s.

The results presented in this section show that the storm develops in a symmetrical manner and the rain falls correctly following the location of the convective towers, as is expected. This is accomplished without a column based grid.

This demonstrates that, in three dimensions, our algorithm successfully transports the rain downward along the convective towers without the need for a vertically structured grid.

CHAPTER 6

ACCELERATING SIMULATIONS OF TROPICAL CYCLONES USING ADAPTIVE MESH REFINEMENT

This chapter focuses on simulations of tropical cyclones with adaptive mesh refinement. The AMR algorithm is presented and tested on two sets of dry tropical cyclone simulations. We will show that AMR can replicate the results of simulations using static grids and provide significant speedup at the same time. All the tests performed in this chapter are performed with the Non-hydrostatic Unified Model of the Atmosphere (NUMA) [51].

6.1 Adaptive Mesh Refinement (AMR)

We rely on the **P4est** [14] library to handle domain decomposition and load balancing of our parallel applications. This library also allows for refining and coarsening grids and we make use of this capability to perform adaptive mesh refinement.

6.1.1 The AMR procedure

Thanks to **P4est** the AMR procedure is straightforward to perform. The process only requires that **P4est** know if an element should be refined, coarsened or left untouched. Let us define an array *adapt* of size Ne, the total number of elements. This array is sent to **P4est**. The values of this array for each element e are as follows:

$$adapt(e) = \begin{cases} -1 & \text{if the element is to be marked for coarsening} \\ 0 & \text{if the element is to be left as is} \\ 1 & \text{if the element is to be marked for refinement} \end{cases} \tag{6.1}$$

Refining an element: In this work, we are only considering adaptive mesh refinement in the horizontal directions. As such, if a hexahedral element is marked

for refinement, it is split into four elements of equal size along the horizontal plane. Since we depend on column data-structures for the planetary boundary layer package being used in our simulations, if an element belonging to a column is refined, the entire column must be refined with it to maintain this structure.

Coarsening an element: If the refinement process involves splitting an element into smaller elements, then the coarsening process is the opposite. Four neighboring elements that are marked for coarsening are merged together to form a single larger element. As such, coarsening cannot take place unless there are four adjacent elements that all share a corner.

Conditions for refinement and coarsening: **P4est** also stores the current level of refinement of each element in an array we will call lvl. Initially, all elements have the level $lvl(e) = 0$ indicating no refinement has taken place. This is also the maximum size of a given element, meaning that it cannot be coarsened if $lvl(e)$ is not strictly positive. In other words, an element cannot be coarsened, if it is currently at the maximum size. An element also cannot be further refined, if it is at the maximum allowed level of refinement lvl_{max}.

A criterion is set for each level of refinement. If one of the nodes belonging to an element verifies the refinement criterion and $lvl(e) < lvl_{max}$, then it is marked for refinement. If an element no longer verifies the refinement criterion for its current level of refinement, then it is marked for coarsening.

The refinement/coarsening criterion Consider that the refinement level l depends on the value of a flow variable C^l, and consider $C^l_{k,e}$ the value of this variable at the kth node of element e. We consider a threshold type criterion for refinement. This means that if any node k belonging to element e verifies

$C^{l}_{k,e} > threshold(l)$, then the element is marked for refinement. It also means that if $\forall k\ C^{lvl(e)}_{k,e} \leq threshold(lvl(e))$, then the element is marked for coarsening.

Frequency of the adaptive mesh refinement procedure: Because AMR is not without cost, it should not be performed at every simulation time step. We define a time interval t_{amr} that is a multiple of the time step Δt and allow the AMR procedure to take place at every instance that the current simulation time is a multiple of t_{amr}. This interval should be large enough to avoid needlessly executing the AMR procedure (a needless AMR procedure would be one where no elements are coarsened or refined), and small enough to be able to adapt to substantial changes in the flow.

Remark: An element can only be refined once per AMR iteration, even if it verifies the criterion for a higher level of refinement. The same applies for coarsening. The level of a given element can only increase or decrease by a value of one at the most every time the AMR procedure is executed.

6.1.2 The AMR algorithm and workflow

We present the reader with an algorithmic representation of the adaptive mesh refinement process through Algorithm 3, where t_{final} is the final time of a given simulation, *threshold* is an array storing the threshold criteria for each level of refinement, and *mod* is the remainder operator. We also present the reader with a workflow diagram of the AMR procedure in order to illustrate the sequence of operations taking place. This diagram is shown in Figure 6.1

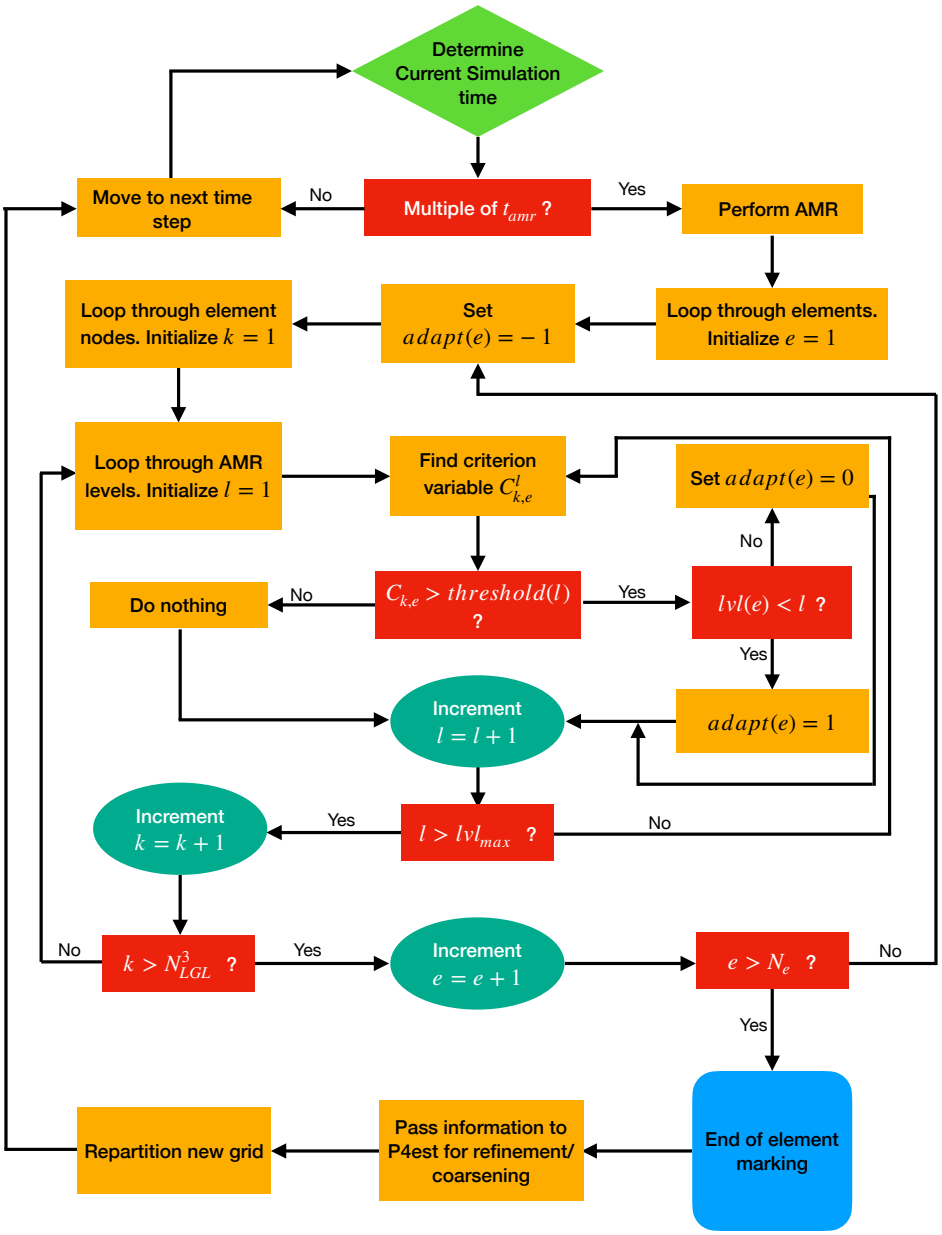


Figure 6.1 Workflow of the adaptive mesh refinement procedure.

Algorithm 3 Algorithm of the Adaptive Mesh Refinement Procedure

for $t = 0, t_{final}, \Delta t$ **do**
  Check if the current time is a multiple of t_{amr}.
  if(mod $(t, t_{amr}) == 0$) **then**
  for $e = 1, N_e$ **do**
    $adapt(e) = -1$
    elements are marked for coarsening unless they pass at least one AMR threshold.
    for $k = 1, N^3_{LGL}$ **do**
      Compute local AMR criterion $C_{k,e}$.
      for $l = 1, lvl_{max}$ **do**
        if $C^l_{k,e} > threshold(kv)$ **then**
          if $lvl(e) < l$ **then**
            $adapt(e) = 1$
          else
            $adapt(e) = 0$
          end if
        end if
      end for
    end for
  end for
  end if
  Refine and coarsen marked elements through P4est
  Re-partition new mesh
end for

6.2 Simulations and Results

The initial conditions for this test are similar to [29]. A dry tropical storm-like vortex is initialized using the following profile for azimuthal-mean tangential velocity:

$$\overline{v}(r, z) = V(r) exp\left[-\frac{z^\sigma}{\sigma D_1^\sigma}\right] exp\left[-\left(\frac{r}{D_2}\right)^6\right], \tag{6.2}$$

where V is the surface tangential velocity, $\sigma = 2$, $D_1 = 5,823$ m, and $D_2 =$ 200 km. The surface tangential velocity can be found by following the procedure described in [85, 86] and integrating a specified Gaussian distribution with a vorticity peak of 1.5×10^{-3} s^{-1} and maximum winds of 21.5 ms^{-1} at a radius of 50 km from the center. The vertical velocity is initially taken to be $w = 0$ s everywhere.

The density and potential temperature are initialized by an iterative procedure that oscillates between satisfying the gradient wind balance and the hydrostatic balance until a specified criterion is met. This procedure is described in detail in [83]. The background state is defined by vertically interpolating the Jordan [49] mean hurricane-season sounding onto the spectral element grid.

Observational heating: The time evolution of the vortex is driven by 4-dimensional source term in the energy equation. This source term represents latent heating/cooling rates derived from Doppler radar measurements in hurricane Guillermo (1997). The latent heating/cooling is described in detail in [30]. Hurricane Guillermo (1997) was a rapidly intensifying tropical cyclone and the heating derived from its observational data should provide a good testing ground for how AMR responds to a rapidly intensifying storm simulation. As described in [37], the heating is computed on a grid covering the inner core of the system out to a radius of $r = 60$ km from the domain center. This grid has a resolution of 2 km in the horizontal direction and 0.5 km in the vertical direction. The heating observations are split into 10 snapshots covering a 5.7 hour period in intervals of 34 minutes. The largest heating rates are present at a radius of $25 - 30$ km from the domain center, well within the radius of maximum winds for the initial conditions. Outside of $r = 60$ km radius the heating term is zero. The heating rates are interpolated in space onto the spectral elements grid. After initialization, the first heating snapshot gradually introduced over the first 30 minutes of the simulation, by way of a hyperbolic tangent function. The snapshots are then linearly interpolated to the next observation time over the course of the remainder of the simulation. Past $t = 5.7$ hours the heating is maintained constant until the simulations finish at $t = 6$ hours.

Boundary conditions At the lower boundary the surface layer model described in [48] is active, while the sea-surface temperature is maintained constant at its reference value. The lateral boundaries are doubly periodic, and a Rayleigh absorbing layer is used to damp gravity waves at the domain top and is described by:

$$R(z) = \gamma sin^2 \left[\frac{\pi}{2} \left(1 - \frac{z_{top-z}}{z_d} \right) \right], \tag{6.3}$$

where $\gamma = 1.0$, $z_{top} = 20$ km and $z_d = 4$ km.

Sub-grid models The Smagorinsky-Lilly model [100, 68] is used to model sub-grid scale turbulence in the horizontal direction. This model is a turbulent viscosity model and contributes to the right hand side through the divergence of the turbulent stress tensor $\nabla \cdot \tau$. The turbulent stress tensor is defined as:

$$\tau = (-2\nu_t \mathbf{S}), \tag{6.4}$$

where $\mathbf{S} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ is the strain rate tensor, and ν_t is turbulent viscosity and is defined as follows:

$$\nu_t = (\Delta C_s)^2 \sqrt{2|\mathbf{S}|^2}, \tag{6.5}$$

where C_s is the constant Smagorinsky coefficient and is taken to $C_s = 0.21$ for our simulations. Δ is the filter width of the Smagorinsky model and is taken to be the mean horizontal resolution of a given element.

The vertical turbulent sub-grid diffusion is modeled by the planetary boundary layer(PBL) scheme of [43, 44] and is described by:

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial z} \left[K_c \left(\frac{\partial C}{\partial z} - \gamma_c \right) - \overline{(w'c')}_h \left(\frac{z}{h^3} \right) \right], \tag{6.6}$$

where C is a prognostic variable, K_c is the eddy diffusivity coefficient, $\overline{w'c'}_h$ is the flux at the inversion layer, and γ_c is a correction to the local gradient. For a more detailed description of the PBL scheme we refer the reader to [43, 44].

Computational domain All of our simulations are done on a $[-400, 400]$ km $\times$ $[-400, 400]$ km $\times$ $[0, 20]$ km domain using continuous Galerkin spectral elements of order 4.

6.2.1 Time to solution comparison

A set of one hour long simulations are performed to compare the time to solution of simulations with and without AMR at varying resolutions. The vertical resolution is maintained constant at $\Delta z = 312$ m. A simulation using a static horizontal resolution of $\Delta x = \Delta y = 4$ km is used as the baseline and its time to solution $T_{4\text{km}}$ is used to obtain a normalized time to solution $T^* = T/T_{4\text{km}}$, where T is the time to solution for a given simulation.

Six simulations in total are performed for this comparison. Three constant horizontal resolution simulations are performed with $\Delta x = \Delta y\,(4, 2, 1)$ km respectively. Three simulations using AMR are performed with respective maximum horizontal resolutions of $\Delta x = \Delta y = 2, 1, 0.5$ km, corresponding to one, two, and three levels of refinement. The criterion for refinement in the AMR simulations is that the velocity magnitude must pass a set of predefined thresholds $threshold_k$, where $k = 1, lvl_{max}$ and lvl_{max} is the maximum refinement level. For these tests we perform one test with $threshold_1 = [7.5]$ ms^{-1}, one test with $threshold_2 = [7.5, 15]$ ms^{-1}, and one test with $threshold_3 = [7.5, 15, 22.5]$ ms^{-1}. All simulations are performed on the same machine with the same number cores, and we will refer to these tests as the first set of test from here on out. The normalized time to solution for these tests is presented in Figure 6.2. For the static grids, the increase in horizontal resolution

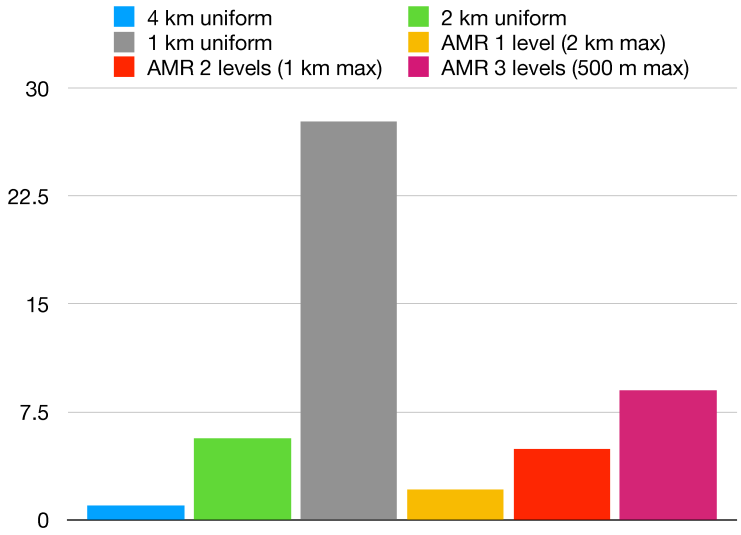


Figure 6.2 Time to solution comparison of static and AMR simulations in normalized time for the first set of tests.

results in ≈ 4 times increase in time to solution from 4 km to 2 km and then similarly from 2 km to 1 km. This is expected as the number of grid cells quadruples when doubling the horizontal resolution in a 3D simulation. With the set of criteria being used, the time to solution for the AMR simulations with up to 1 km horizontal resolution remain lower than a uniform 2 km horizontal resolution simulation. The addition of additional refinement levels results in a ≈ 2 times increase in cost for the AMR simulations with the sets of criteria being used in these tests. The AMR simulation with a horizontal resolution of up to 500 m is still much cheaper than a static 1 km resolution simulation and barely more expensive than a static 2 km resolution simulation.

A second set of tests is performed to observe the effect of changing the refinement criterion on the time to solution and to see how AMR performs over the full six hour simulation period. This set of tests includes four simulations:

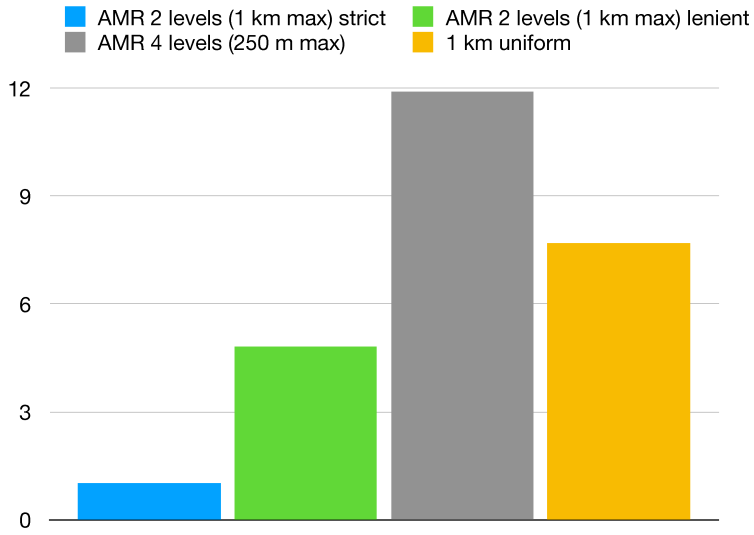


Figure 6.3 Time to solution comparison of static and AMR simulations in normalized time for the second set of tests. The strict 2-level AMR uses $threshold_{st}$, while the lenient 2-level AMR uses $threshold_{ln}$

- A simulation using a static grid with a horizontal resolution $\Delta x = \Delta y = 1$ km.
- A simulation with two levels of refinement (up to 1 km horizontal resolution) with the same refinement criterion $threshold_{st} = threshold_2$ as the previous 2-level simulation, this will be referred to as the strict 2-level simulation.
- A simulation with two levels of refinement but with a more lenient refinement criterion $threshold_{ln} = [2, 5]$ ms^{-1}, this will be referred to as the lenient 2-level simulation.
- A simulation with four levels of refinement(up to 250 m horizontal resolution) and a vertical resolution $\Delta z = 250$ m. The thresholds for the simulation are $threshold_4 = [7.5, 15, 22.5, 30]$ ms^{-1}.

Remark 6.2.1 *All simulations except the simulation with four levels of AMR maintain a vertical resolution* $\Delta z = 312$ *m.*

Figure 6.3 presents the time to solution for the second set of tests in normalized time. For these tests, the baseline is the strict 2-level AMR simulation as it is the cheapest to perform. We can see that while the two simulations using two levels of

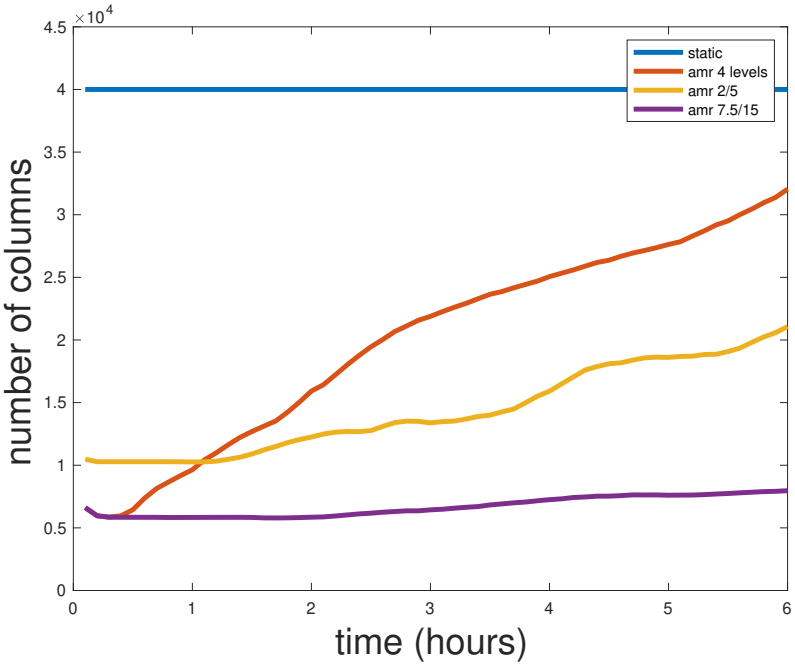


Figure 6.4 Number of element columns over time for the static grid simulation (blue), the 4 level amr simulation (red) with $threshold_4 = [7.5, 15, 22.5, 30]$ ms^{-1}, the 2 level amr simulation with $threshold_{ln} = [2, 5]$ ms^{-1} (yellow), and the 2 level amr simulation with $threshold_{st} = [7.5, 15]$ ms^{-1} (purple).

AMR remain cheaper than the static alternative, the more lenient criterion incurs a substantial increase in the cost of the simulation.

Figure 6.4 shows the number of vertical columns over time for this set of simulations. We can see that with the stricter AMR criteria and two levels of refinement (purple line), the number of columns does not vary substantially over the course of the simulation. With the more lenient criteria and two levels of refinement (yellow line), the number of columns at the end of the simulation is double what it was at the beginning which explains the significant increase in cost. The simulation using four levels of AMR shows a substantial increase in the number of columns over the course of the simulation, the number of columns at the end is five times what it was at the beginning. As the hurricane intensifies more areas of high velocity magnitude appear and the mesh adapts by refining around these areas and generating

more columns. While the total number of columns for the 4 level AMR simulation is less than the total number of columns for the static simulation, it still requires a longer time to complete. The main reason for this is the increased vertical resolution which increases the total number of degrees of freedom and further restricts the time required for stability. For the remainder of this section we will continue referring to this set of simulations.

6.2.2 Accuracy of AMR simulations

Figure 6.5 shows the maximum values of the horizontally averaged horizontal velocity over time for $(x, y) \in [-50, 50]$ km $\times$ $[-50, 50]$ km. These values are found by finding the average value of horizontal velocity at each vertical level and then finding the maximum among these. This horizontal sub-domain is chosen as it should allow for focusing on where the observational heating takes place. Figure 6.6 shows the values of maximum horizontal velocity over time. Both figures compare these values for the static simulation and the two 2-level AMR simulations. Both figures show an essentially perfect overlap for these quantities and demonstrate the AMR's ability to capture the intensification of the storm even with relatively strict criteria.

Figure 6.7 shows the horizontal velocity of the storm at different times for a horizontal slice taken at $z = 1,000$ m. We can see that the left column (2-level AMR with strict criterion) and middle column (static grid) are identical at all the displayed times. The right column (4-level AMR) allows us to see the additional details and structures that can be captured with additional refinement. Figure 6.8 shows the velocity magnitude of the storm at different for a vertical slice taken at $x = 0$ m. Once again the left and middle columns are identical and demonstrate the ability of AMR to obtain high fidelity results at a fraction of the cost of a simulation using a static grid. Figure 6.9 is similar to Figure 6.7 but presents the vorticity magnitude.

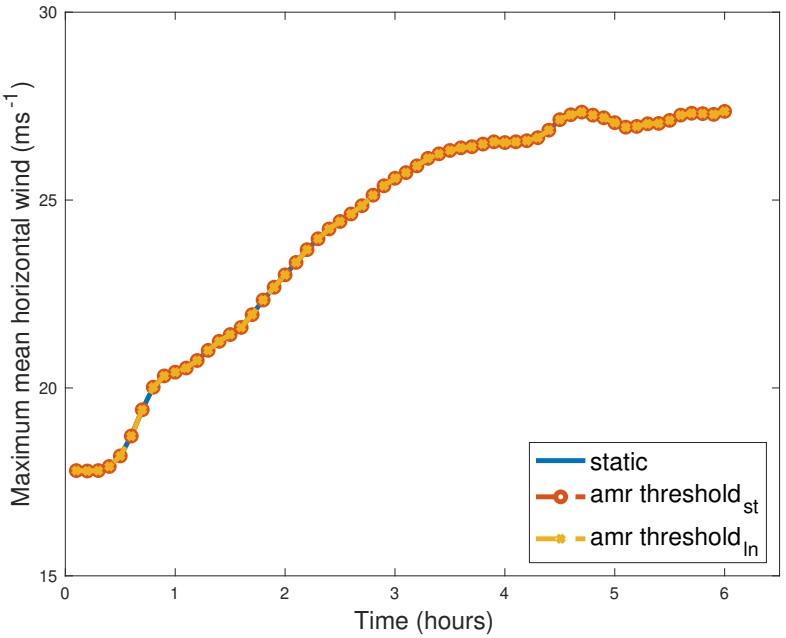


Figure 6.5 Maximum of horizontally averaged horizontal velocity within the $[-50, 50]$ km^2 sub domain over the course of the 6 hour simulation period. The blue solid line (static simulation) overlaps perfectly with the dashed line with red circles (stricter criterion AMR simulation) and the dashed line with yellow crosses(more lenient criterion AMR simulation).

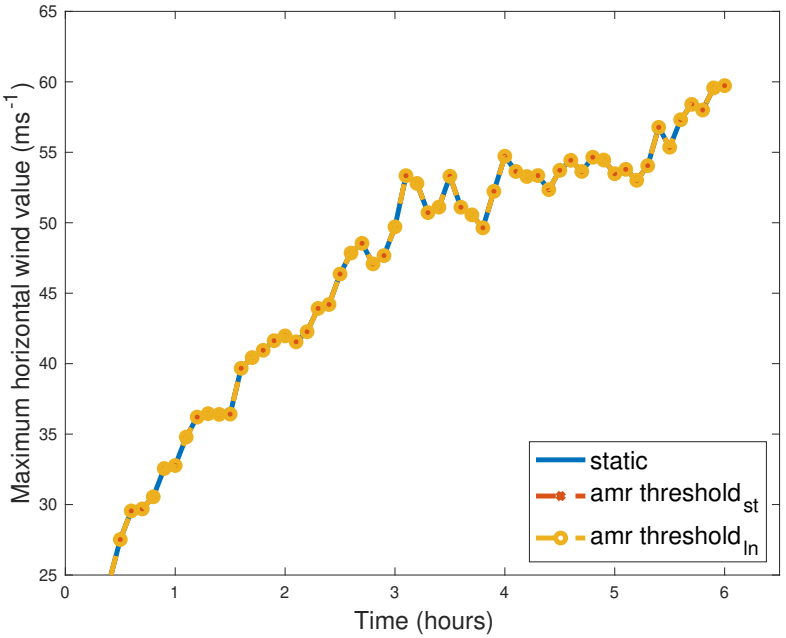


Figure 6.6 Maximum value of horizontal velocity within the $[-50, 50]$ km $\times$ $[-50, 50]$km $\times$ $[0, 20]$ km sub domain over the course of the 6 hour simulation period. The blue solid line (static simulation) overlaps perfectly with the dashed line with red crosses (stricter criterion AMR simulation) and the dashed line with yellow circles(more lenient criterion AMR simulation).

Even for this derived quantity the left and center columns are still identical. The right column showcases what additional resolution has to offer in terms of resolving turbulent structures. The simulations with resolutions of up to 250 m of horizontal resolution show an increase in both vorticity and velocity compared to their lower resolution counterparts. This reflects the role of the additional resolution in capturing the intensification of the storm. Vorticity is especially important when considering features like strong updrafts and convective towers. Pushing this resolution even further could allow for studies of how these features affect TC intensification and RI through numerical experiments.

We now focus on the simulation using four levels of refinement. Figure 6.10 shows the first instance we detect of the grid refining to a horizontal resolution $\Delta x = \Delta y = 500$ m. This refinement takes place at $t \approx 0.3$ hours, near the beginning of the simulation. The top two plots show areas of higher velocity magnitude (dark red shades) this corresponds to the criterion for refining to 500 m being met and the bottom two plots of the figure show how the grid responds to the intensification. Figure 6.11 shows the first instance we detect of the grid refining to a horizontal resolution $\Delta x = \Delta y = 250$ m. This refinement takes at $t \approx 0.4$ hours, still near the beginning of the simulation. As shown in the top two plots, high values of velocity magnitude trigger the refinement process and grid adapts (middle two plots). At this time the high value of velocity magnitude is associated with the occurrence of a strong updraft, as evidenced by the high vertical velocity values shown in the bottom two plots of the figure.

Figure 6.12 Shows the 4 level AMR simulation at $t = 6$ hours and at the $z = 1000$ m horizontal plane within the boundaries of $(x, y) \in [-200, 200]$ km $\times$ $[-200, 200]$ km. All the levels of the AMR grid are visible and as we would expect, the grid is finer near the storm center than it is farther away.

6.3 Discussion

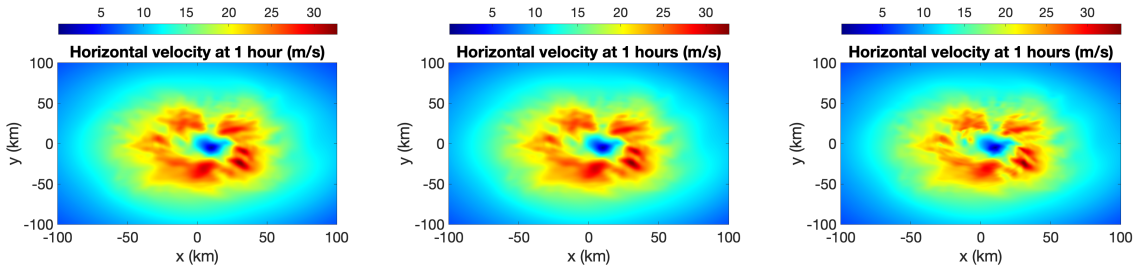
6.3.1 Criteria for adaptive mesh refinement

The efficiency of AMR depends very closely on the choice of criterion chosen for refinement. A criterion that is too strict and hard to fulfill can result in insufficient refinement and lead to less accurate simulations. A criterion that is too lenient and easy to fulfill can results in less time savings and perhaps unnecessary costs. This work does not offer an in depth study of different criteria for AMR, but we do demonstrate how it can affect the efficiency of a simulation. Deciding on a criterion requires knowledge of specific features that the AMR is meant to capture, velocity magnitude was chosen for the simulations here as it allows for AMR to detect regions where intensification takes place. However, it could be argued that this criterion can only be activated after intensification has already begun taking place and it might not be able to capture the onset of intensification. Potential vorticity has been studied by [40] as a potential criterion for tropical cyclone AMR and showed promise. Perhaps a criterion that combines multiple flow variables should be considered for future AMR simulations of tropical cyclones.

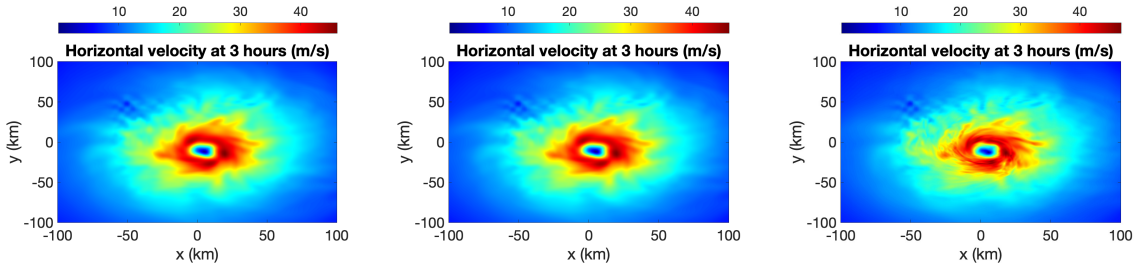
6.3.2 LES simulations of tropical cyclones with AMR

Two approaches come to mind when thinking of extending tropical cyclone simulations with AMR to LES resolutions. The first is to use LES resolutions within the entire domain thereby completely eliminating the need for a PBL scheme. This approach means the minimum resolution must be high enough to resolve most of the large eddies at all time. AMR would be used to decrease the dependence on the sub-grid-scale model and resolve more of the small scale turbulence. Though this method makes use of AMR, simulating a tropical cyclone over its life cycle would still be very expensive. Nevertheless, as this approach would not need a typically column-dependent PBL

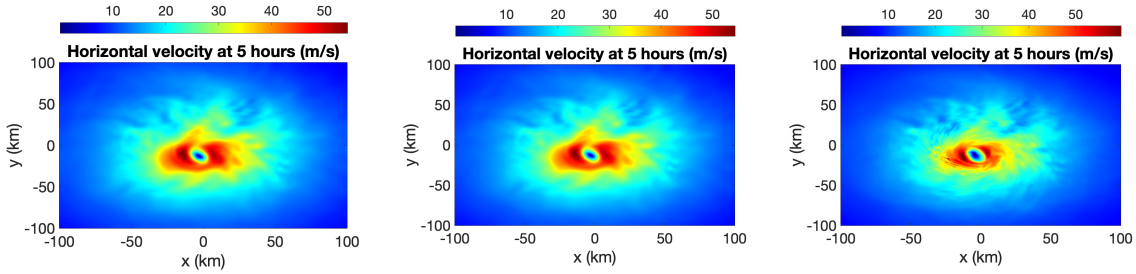
scheme, it would make 3D AMR simulations of moist tropical cyclones straightforward thanks to the non-column microphysics approach [105]. The second approach is much less expensive and involves using the same coarse base grid tested in this work. To reach resolutions below 100 m, six levels of refinement would be required. But, this approach poses a complication as it pertains to modeling vertical turbulent fluxes. At high resolutions a fully 3D LES turbulence model should be used, while at coarse resolutions a PBL scheme should be used. It remains unclear at what resolutions a transition should be made between the two turbulence models. It is also unclear how exactly a transition between the two models would look. The need for a PBL scheme also makes the use of 3D AMR difficult given the need for column-based data structures. Perhaps if a suitable criterion was designed such that all areas of interest of a tropical cyclone could be resolved at LES resolutions, it might be possible to circumvent the need for a PBL scheme. In this case the assumption would be made that the PBL scheme's contribution to intensification would have to be minimal at a sufficient distance from the storm center. We have begun experimenting with this approach for moist tropical cyclone simulations and have obtained some early results. Figure 6.13 presents a moist tropical cyclone simulation with six levels of adaptive mesh refinement for a horizontal resolution of up to $\Delta x = \Delta y = 62.5$ m. This figure presents a render of the cloud cover at $t = 17.3$ hours of a simulation with a cold start (No observational heating is involved). This work is still only its early stages and hopefully that an in depth study can be done in the future.



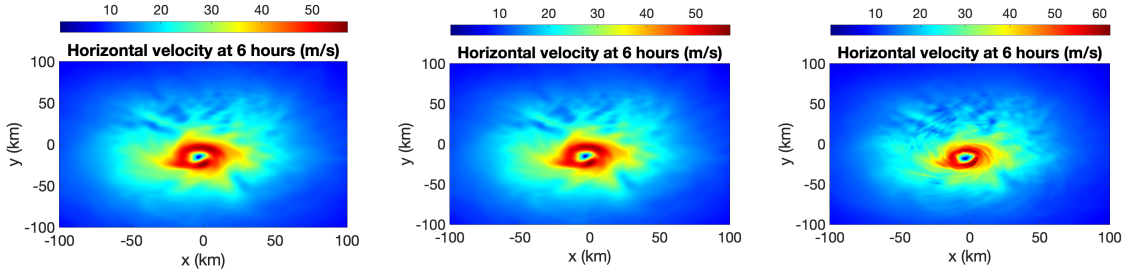
(a) AMR with two levels **(b)** Static 1km **(c)** AMR with four levels



(d) AMR with two levels **(e)** Static 1km **(f)** AMR with four levels

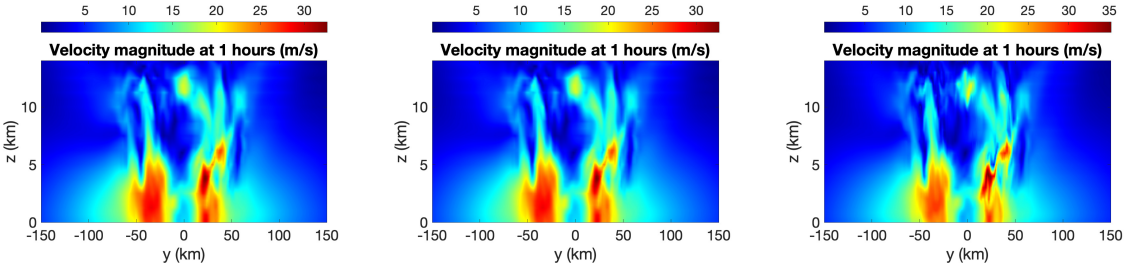


(g) AMR with two levels **(h)** Static 1km **(i)** AMR with four levels

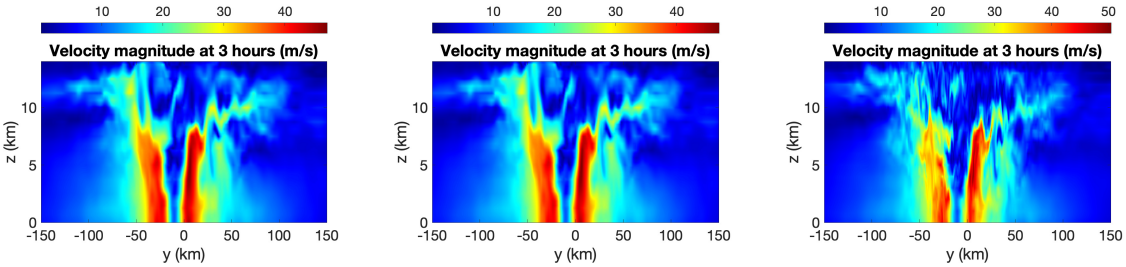


(j) AMR with two levels **(k)** Static 1km **(l)** AMR with four levels

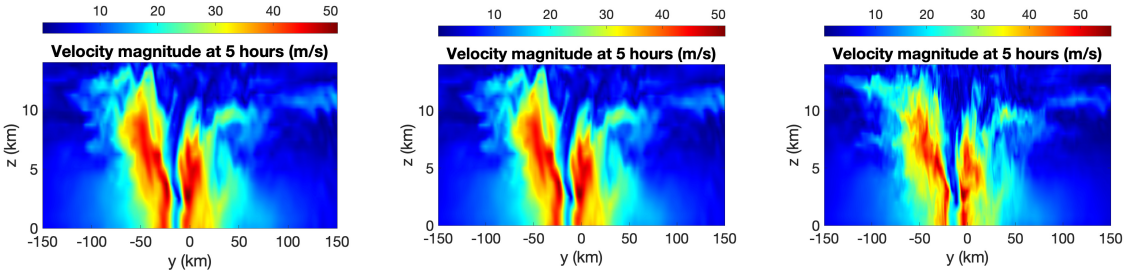
Figure 6.7 Horizontal winds taken at height $z = 1000$ m above sea level at different stages of the simulation. The left column presents results for a simulation with two levels of AMR and a maximum horizontal resolution of 1 km. The center column presents the results for a static grid with 1 km constant horizontal resolution. The right column presents the results for a simulation with four levels of AMR and a maximum horizontal resolution of 250 m.



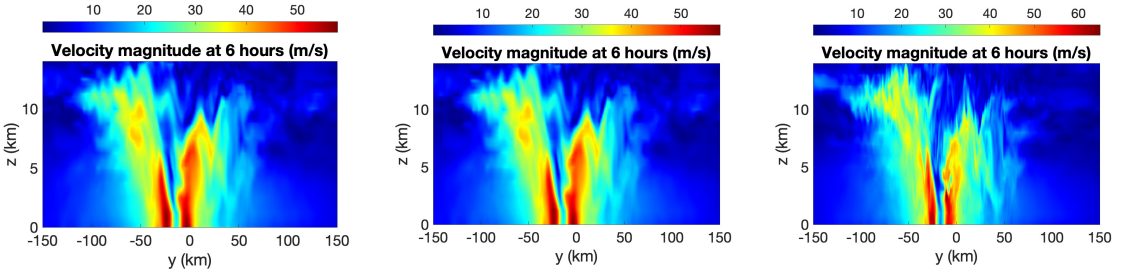
(a) AMR with two levels **(b)** Static 1km **(c)** AMR with four levels



(d) AMR with two levels **(e)** Static 1km **(f)** AMR with four levels

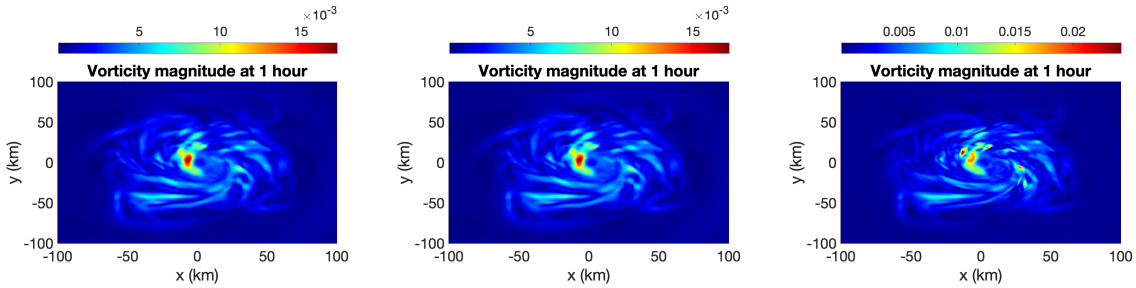


(g) AMR with two levels **(h)** Static 1km **(i)** AMR with four levels



(j) AMR with two levels **(k)** Static 1km **(l)** AMR with four levels

Figure 6.8 Velocity Magnitude taken at $x = 0$ m at different stages of the simulation. The left column presents results for a simulation with two levels of AMR and a maximum horizontal resolution of 1 km. The center column presents the results for a static grid with 1 km constant horizontal resolution. The right column presents the results for a simulation with four levels of AMR and a maximum horizontal resolution of 250 m.



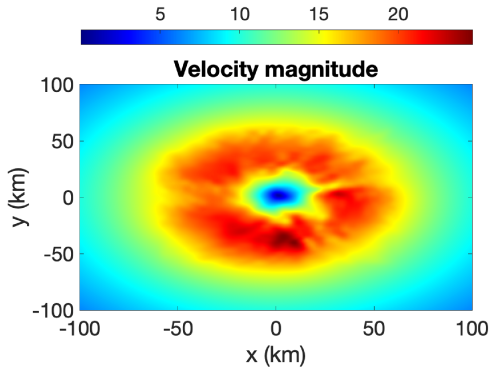
(a) AMR with two levels **(b)** Static 1km **(c)** AMR with four levels

(d) AMR with two levels **(e)** Static 1km **(f)** AMR with four levels

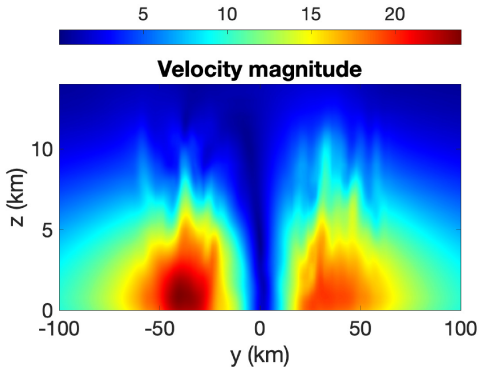
(g) AMR with two levels **(h)** Static 1km **(i)** AMR with four levels

(j) AMR with two levels **(k)** Static 1km **(l)** AMR with four levels

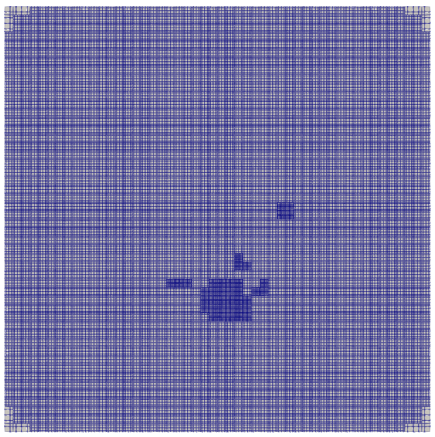
Figure 6.9 Vorticity magnitude taken at height $z = 1000$ m above sea level at different stages of the simulation. The left column presents results for a simulation with two levels of AMR and a maximum horizontal resolution of 1 km. The center column presents the results for a static grid with 1 km constant horizontal resolution. The right column presents the results for a simulation with four levels of AMR and a maximum horizontal resolution of 250 m.



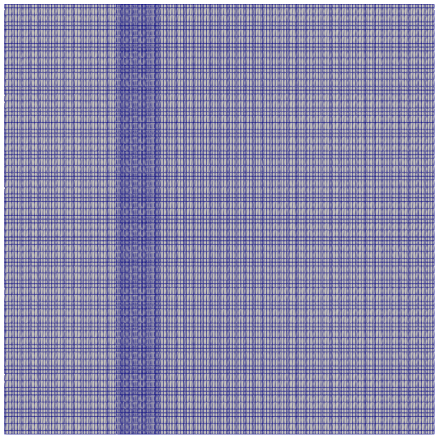
(a) Horizontal slice at $z = 1,000$ m



(b) vertical slice at $x = 0$ m

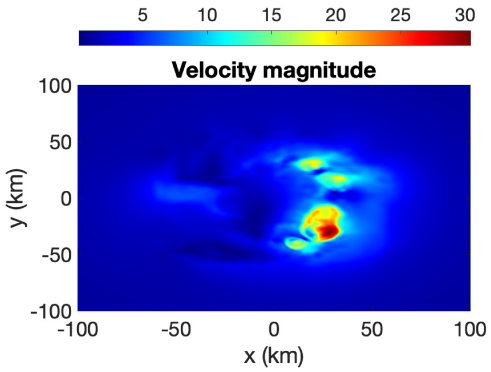


(c) Horizontal profile of the grid

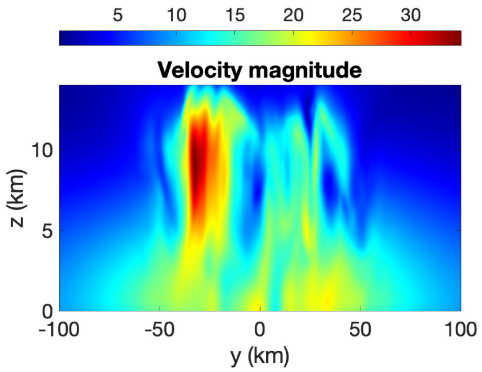


(d) Vertical profile of the grid

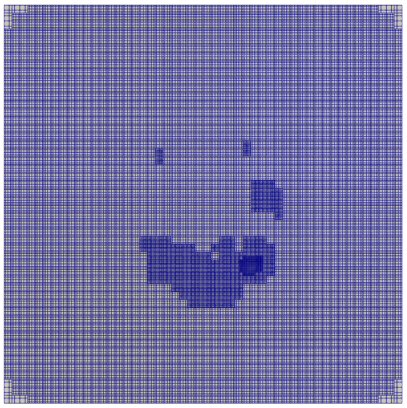
Figure 6.10 Velocity magnitude and grid at the first instance of reaching the third level of refinement for the 4 level AMR simulation. a) Velocity magnitude at $z = 1000$ m and $t = 0.3$ hours. b) Velocity magnitude at $x = 0$ m and $t = 0.3$ hours. c) Horizontal grid at $z = 1000$ m, $t = 0.3$ hours, and for $(x, y) \in [-100, 100]$ km $\times$ $[-100, 100]$ km. d) Vertical grid at $x = 0$ m, $t = 0.3$ hours, and for $(y, z) \in [-100, 100]$ km $\times$ $[-0, 20]$ km. Note that the lowest resolution visible in c) is 2 km (seen in the corners) and the highest is 500 m. The baseline 4 km resolution is not visible as it present farther away from the domain center. In d) the lowest visible horizontal resolution is 1 km and the highest is 500 m. The lower resolutions are present in other areas of the simulated domain not pictured here. Areas with a velocity magnitude larger than 22.5 ms^{-1} trigger the criterion for the third level of refinement and cause grid to reach 500 m of horizontal resolution wherever the criterion is met. As this refinement is done on a column basis, the entire vertical column is refined as shown in d).



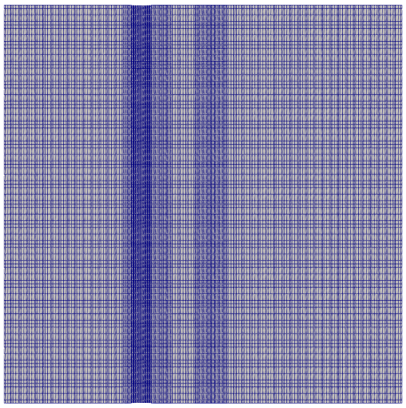
(a) Horizontal slice at $z = 12,000$ m



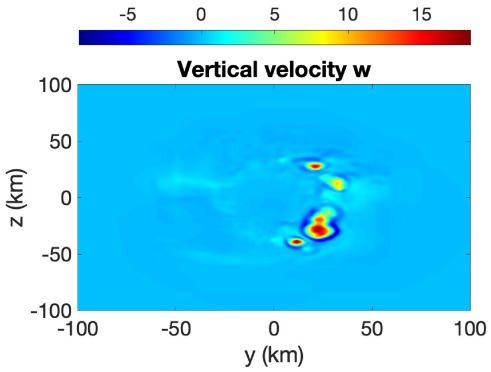
(b) vertical slice at $x = 24,000$ m



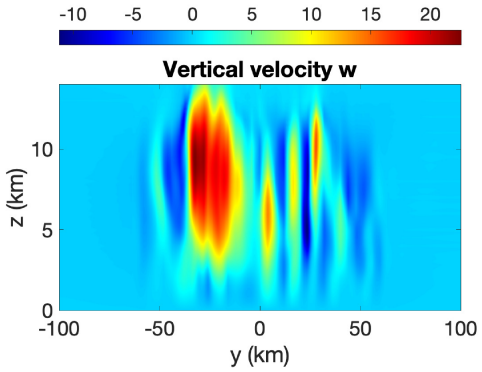
(c) Horizontal profile of the grid



(d) Vertical profile of the grid

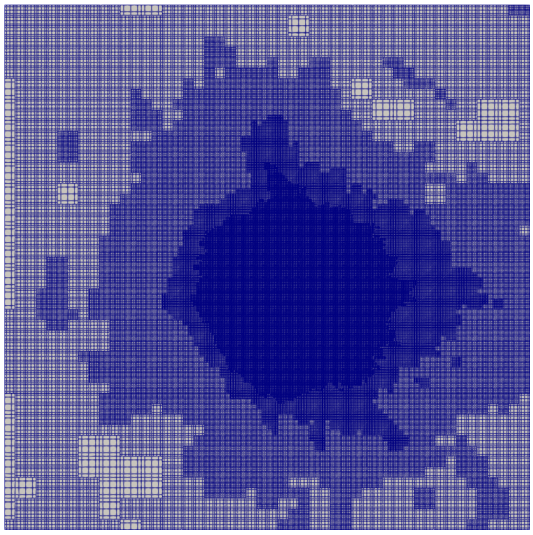


(e) Horizontal slice at $z = 12,000$ m

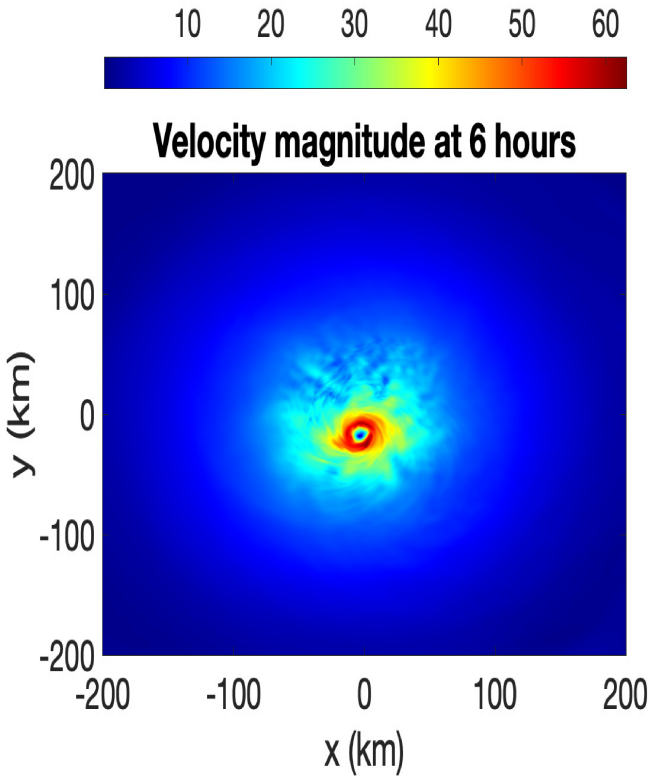


(f) vertical slice at $x = 24,000$ m

Figure 6.11 Velocity magnitude and grid at $t = 0.4$ hours, the first instance of reaching the fourth level of refinement for the 4 level AMR simulation. a) Velocity magnitude at $z = 12$ km. b) Velocity magnitude at $x = 24$ km. c) Horizontal grid at $z = 12$ km and for $(x, y) \in [-100, 100]$ km $\times$ $[-100, 100]$ km. d) Vertical grid at $x = 12$ km and for $(y, z) \in [-100, 100]$ km $\times$ $[-0, 20]$ km. e) Vertical velocity at $z = 12$ km. f) Vertical velocity at $x = 24$ km. In c) and d) the lowest horizontal resolution is 2 km and 1 km respectively and the highest is 250 m. The baseline 4 km resolution is not visible but is present farther away from the center. A substantial updraft indicated by the elevated vertical velocity shown in e) and f) allows the velocity magnitude to pass the 30 ms^{-1} threshold required for the 4th level of AMR.



(a) AMR grid at $t = 6$ hours



(b) Tropical cyclone at $t = 6$ hours and $z = 1000$ m.

Figure 6.12 Example of 4-level refinement at $t = 6$ hours. The criterion used is $threshold_4 = [7.5, 15, 22.5, 30]$ ms^{-1} on velocity magnitude. a) Horizontal cross section of the AMR grid. b) Velocity magnitude at $t = 6$ hours and $z = 1000$ m.

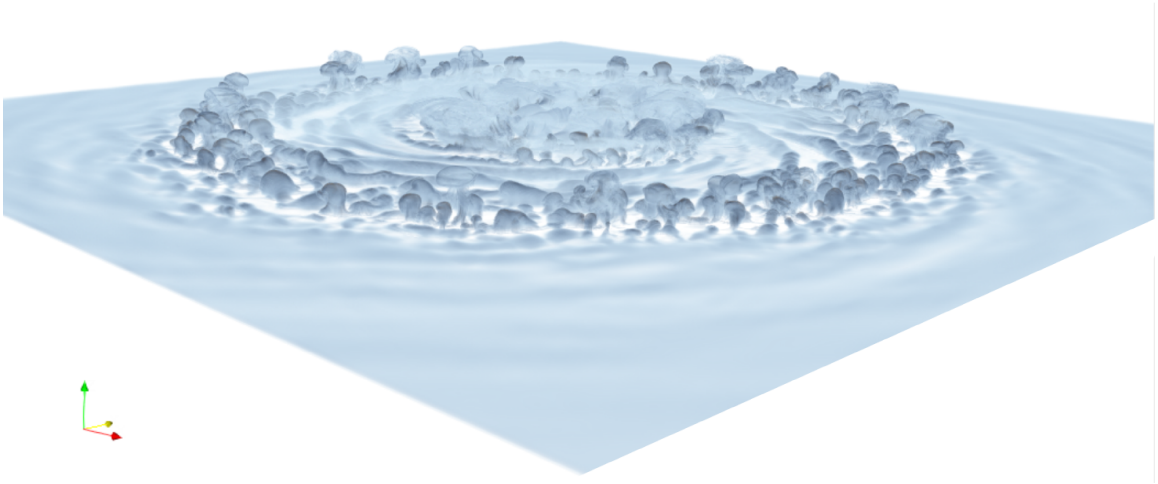


Figure 6.13 Cloud cover of a moist tropical cyclone simulation with up to 6 levels of AMR, a vertical resolution of $\Delta z = 100$ m and a maximum horizontal resolution of $\Delta x = \Delta y = 62.5$ m.

CHAPTER 7

CONCLUSION

This dissertation demonstrated the efficacy of adaptive mesh refinement (AMR) in accelerating simulations of tropical cyclones during rapid intensification. It showed that AMR can replicate the results of static grid simulations while significantly reducing simulation costs. Additionally, the dissertation introduced a novel column-free method for modeling microphysics in atmospheric simulations, which enables the use of 3D AMR for simulating tropical cyclones. This approach holds promise for greater efficiency compared to 2D AMR. Furthermore, the dissertation proposed a new and efficient method for the Rayleigh damping layer, commonly utilized to damp outgoing gravity waves in atmospheric flows. Thi1s advancement has the potential to further enhance the efficiency of tropical cyclone simulations and pave the way for large eddy simulations of their entire life cycle. This in turn, could lead to a better understanding of tropical cyclones and their rapid intensification through the use of numerical experiments.

APPENDIX A

RIGHT HAND SIDE CONSTRUCTION FOR SEMI-INFINITE ELEMENTS USING THE LAGUERRE FUNCTION BASIS

The construction of the right hand side for a semi-infinite element with a scaled Laguerre function basis is presented here. Consider a general system of PDEs, written in conservation form, on the semi-infinite domain $\Omega \; \forall \; t = 0, ...T_{\text{end}}$:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{q})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{q})}{\partial z} = \mathbf{S}(\mathbf{q}) + \mathbf{V}(\mathbf{q}), \tag{A.1}$$

where the state vector $\mathbf{q}$, flux vectors $\mathbf{F}$ and $\mathbf{G}$, source vector $\mathbf{S}$, and diffusion vector $\mathbf{V}$ are problem-dependent. We provide the reader with a pseudo-code for the use of inexact integration in computing the element right hand sides of a PDE on semi-infinite elements, where $\mathbf{rhs}^{Lag}$ is the right hand side of a semi-infinite element.

With the element right hand sides determined, the same DSS operation described in § 2.2.2 can be used to construct the global right hand. Similarly to applying DSS to the mass matrix, this enforces the continuity of the global solution and is the only coupling between the finite and semi-infinite domains.

Algorithm 4 Construction of the right hand side for an element of the semi-infinite domain

$\mathbf{rhs}^{Lag} = \text{zeros}(N_{LGL}, N_{LGR})$

for $j = 1, N_{LGR}$ **do**

  for $i = 1, N_{LGL}$ **do**

    $\overline{\omega} = \omega(\xi_i)\hat{\omega}(\eta_j)$

    $dFd\xi = dFd\eta = dGd\xi = dGd\eta = 0$

    for $k = 1, N_{LGL}$ **do**

      $dFd\xi = dFd\xi + h'_k(\xi_i)F(\mathbf{x}(\xi_k, \eta_j, e))$

      $dGd\xi = dGd\xi + h'_k(\xi_i)G(\mathbf{x}(\xi_k, \eta_j, e))$

    end for

    for $k = 1, N_{LGR}$ **do**

      $dFd\eta = dFd\eta + \hat{h}'^{Lag}_k(\eta_j)F(\mathbf{x}(\xi_i, \eta_k, e))$

      $dGd\eta = dGd\eta + \hat{h}'^{Lag}_k(\eta_j)G(\mathbf{x}(\xi_i, \eta_k, e))$

    end for

    $dFdx = dFd\xi \cdot d\xi dx + dFd\eta \cdot d\eta dx$

    $dGdz = dGd\xi \cdot d\xi dz + dGd\eta \cdot d\eta dz$

    $\mathbf{rhs}^{Lag}_{ij} = \mathbf{rhs}^{Lag}_{ij} - \overline{\omega}|J(\xi_i, \eta_j)|(dFdx + dGdz)$

  end for

end for

APPENDIX B

EXTENDING THE LAGUERRE-LEGENDRE SEMI-INFINITE ELEMENT METHOD TO 3D

This appendix presents the procedure for constructing 3D semi-infinite elements to help the reader interested in extending 2D semi-infinite elements to three-dimensions. The extension can be done as easily as extending a standard CG approach on a tensor product basis. We give an explanation here although this paper is limited to two-dimensions. Let $\xi = (\xi, \eta, \zeta)$ be the coordinate of a point on the three-dimensional reference element. Extending Equation (2.38) to three dimensions is done as follows:

$$\psi_l(\mathbf{x}) = h_i[\xi(\mathbf{x})] \otimes h_j[\eta(\mathbf{x})] \otimes \overline{h}_k[\eta(\mathbf{x})], \quad l = i + (j-1)N_{LGL} + (k-1)N_{LGL}^2, \quad \text{(B.1)}$$

where $i \in \{1, \ldots, N_{LGL}\}$, $j \in \{1, \ldots, N_{LGL}\}$, and $k \in \{1, \ldots, N_{LGR}\}$. It is then simple to extend Equation (2.7) to three dimensions by adding an additional sum over the LGL nodes and including their corresponding weights in the product:

$$\int_{\Omega_e} f(\mathbf{x}) d\mathbf{x} = \int_{\Omega_{ref}} f(\boldsymbol{\xi}) |\mathbf{J}(\boldsymbol{\xi})| d\boldsymbol{\xi} \approx \sum_{i,j=1}^{N_{LGL}} \sum_{k=1}^{N_{LGR}} \omega(\xi_i)\omega(\eta_j)\hat{\omega}(\zeta_k) f(\xi_i, \eta_j, \zeta_k) |\mathbf{J}(\xi_i, \eta_j, \zeta_k)|. \quad \text{(B.2)}$$

The remainder of the extension, such as the construction of the mass matrix can be done by following the same approach.

APPENDIX C

SOUNDINGS FOR TROPICAL CYCLONE SIMULATIONS AND FOR THE SQUALL LINE AND SUPER-CELL TEST CASES

This appendix presents the soundings used to initialize the background state for the simulations presented in Chapter 5 and in Chapter 6.

Table C.1 Jordan 1958 [49] Mean Hurricane Season Sounding Used to Initialize the Background State for the Tropical Cyclone Simulations presented in Chapter 6.

z (m)	q_v (g/kg)	θ (K)
10.0	18.62	299.94
125.0	18.47	299.64
810.0	15.259	295.04
1541.0	11.96	290.07
3178.0	6.729	282.04
4438.0	4.120	274.64
5887.0	2.409	266.54
7596.0	1.119	256.04
9690.0	0.329	240.84
10949.0	0.04	230.74
12417.0	0.01	218.74
14202.0	0.01	205.94
16589.0	0.0	198.74
20727.0	0.0	210.14
22139.0	0.0	215.84
23971.0	0.0	219.14

Table C.2 Sounding Used for the Squall Line and Super-Cell Simulations of Chapter 5

z (m)	θ (K)	q_v (g/kg)	u (m/s)	v (m/s)	p (Pa)
0.0	303.025079	14.000	12.0	0.0	100000.0
480.0	303.337272	14.000	9.696000	0.0	94697.28
960.0	304.402985	14.000	7.392000	0.0	89609.81
1440.0	305.397187	12.796	5.088000	0.0	84736.79
1920.0	306.306214	10.556	2.784000	0.0	80070.30
2400.0	307.365269	8.678	0.540000	0.0	75604.36
2880.0	308.550318	7.104	0.0	0.0	71334.51
3360.0	309.845257	5.788	0.0	0.0	67255.79
3840.0	311.235047	4.691	0.0	0.0	63362.95
4320.0	312.708238	3.777	0.0	0.0	59650.49
4800.0	314.255743	3.020	0.0	0.0	56112.80
5280.0	315.869985	2.396	0.0	0.0	52744.15
5760.0	317.544512	1.885	0.0	0.0	49538.82
6240.0	319.273784	1.469	0.0	0.0	46491.09
6720.0	321.052868	1.134	0.0	0.0	43595.27
7200.0	322.877588	0.866	0.0	0.0	40845.73
7680.0	324.744235	0.653	0.0	0.0	38236.93
8160.0	326.649534	0.487	0.0	0.0	35763.41
8640.0	328.590559	0.357	0.0	0.0	33419.84
9120.0	330.565013	0.259	0.0	0.0	31200.99
9600.0	332.571020	0.184	0.0	0.0	29101.75
10080.0	334.606102	0.129	0.0	0.0	27117.17
10560.0	336.668475	0.088	0.0	0.0	25242.39
11520.0	340.869535	0.038	0.0	0.0	21803.59
12000.0	343.712008	0.026	0.0	0.0	20232.15
12480.0	350.647306	0.026	0.0	0.0	18763.71
12960.0	358.453724	0.029	0.0	0.0	17401.15
13440.0	366.433620	0.031	0.0	0.0	16138.11
13920.0	374.591035	0.034	0.0	0.0	14967.29
14400.0	382.929618	0.037	0.0	0.0	13881.93
15360.0	400.170355	0.044	0.0	0.0	11942.99
15840.0	409.081924	0.049	0.0	0.0	11078.24
16320.0	418.191751	0.053	0.0	0.0	10276.53
16800.0	427.504224	0.058	0.0	0.0	9533.23
17280.0	437.023716	0.063	0.0	0.0	8844.07
17760.0	446.755038	0.069	0.0	0.0	8205.09
18720.0	466.871821	0.083	0.0	0.0	7063.24
19200.0	477.267160	0.091	0.0	0.0	6553.82
19680.0	487.891998	0.094	0.0	0.0	6081.42
20160.0	498.742611	0.094	0.0	0.0	5643.35
20640.0	509.643457	0.094	0.0	0.0	5237.00
21120.0	520.544304	0.094	0.0	0.0	4859.92
21600.0	531.445151	0.094	0.0	0.0	4509.85
22560.0	553.246845	0.094	0.0	0.0	3882.66
23040.0	564.147692	0.094	0.0	0.0	3601.93
23520.0	575.048539	0.094	0.0	0.0	3340.96
24000.0	585.949386	0.094	0.0	0.0	3098.30

REFERENCES

[1] D. S. Abdi, F. X. Giraldo, E. M. Constantinescu, L. E. Carr III, L. C. Wilcox, and T. C. Warburton. Acceleration of the Implicit-Explicit Non-Hydrostatic Unified Model of the Atmosphere (NUMA) on Manycore Processors. *International Journal of High Performance Computing Applications*, 2017.

[2] D. S. Abdi, L. C. Wilcox, T. C. Warburton, and F. X. Giraldo. A GPU-accelerated continuous and discontinuous Galerkin non-hydrostatic atmospheric model. *International Journal of High Performance Computing Applications*, 2017.

[3] Daniel S. Abdi and Francis X. Giraldo. Efficient construction of unified continuous and discontinuous galerkin formulations for the 3d euler equations. *Journal of Computational Physics*, 320:46–68, 2016.

[4] D. P. Bacon, N. N. Ahmad, Z. Boybeyi, T. J. Dunn, M. S. Hall, P. C. S. Lee, R. A. Sarma, M. D. Turner, K. T. Waight, S. H. Young, and J. W. Zack. A Dynamically Adapting Weather and Dispersion model: The Operational Multiscale Environment Model with Grid Adaptivity (OMEGA). *Monthly Weather Review*, 128:2044–2067, 2000.

[5] T. Benacchio and L. Bonaventura. Absorbing boundary conditions: a spectral collocation approach. *Journal of Computational Physics*, 72(9):913–936, 2013.

[6] T. Benacchio and L. Bonaventura. An extension of dg methods for hyperbolic problems to one-dimensional semi-infinite domains. *Applied Mathematics and Computation*, 350:266–282, 2019.

[7] M. J. Berger, D. A. Calhoun, C. Helzel, and R. J. Leveque. Logically rectangular finite volume methods with adaptive refinement on the sphere. *Philosiphical Transactions of the Royal Society*, 367, 2009.

[8] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512, 1984.

[9] J.P. Boyd. The erfc-log filter and the asymptotics of the Euler and Vandeven sequence accelerations. In *Proceedings of the Third Internatioal Conference on Spectral and High Order Methods*, pages 267–276. Houston Journal of Mathematics, 1996.

[10] S. A. Braun and W. K. Tao. Sensitivity of high-resolution simulations of hurricane bob (1991) to planetary boundary layer parameterizations. *Monthly Weather Review*, 128, 2000.

[11] G. Bryan, J. C. Wyngaard, and J. M. Fritsch. Resolution requirement for the simulation of deep moist convection. *Monthly Weather Review*, 131:2394–2416, 2003.

[12] G. H. Bryan. Effects of surface exchange coefficients and turbulence length scales on the intensity and structure of numerically simulated hurricanes. *Monthly Weather Review*, 140:1125–1143, 2012.

[13] G. H. Bryan, J. C. Knievel, and M. D. Parker. A multimodel assessment of RKW theory's relevance to squall-line charac- teristics. *Monthly Weather Review*, 134:2772–2792, 2006.

[14] C. Burstedde, L. C. Wilcox, and O. Ghattas. `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011.

[15] C. Chen, F. Xiao, and X. Li. An adaptive multimoment global model on a cubed sphere. *Monthly Weather Review*, 139:523–548, 2011.

[16] T. A. Cram, J. Persing, M. T. Montgomery, and S. A. Braun. A lagrangian trajectory view on transport and mixing processes between the eye, eyewall, and evironment using a high resolution simulation of hurricane bonnie (1998). *Journal of Atmospheric Sciences*, 64:1835–1856, 2007.

[17] M. DeMaria, J. L. Franklin, M. J. Onderlinde, and J. Kaplan. Operational forecasting of tropical cyclone rapid intensification at the national hurricane center. *Atmosphere*, 12(6):683, 2021.

[18] M. DeMaria, C. Sampson, J. Knaff, and K. Musgrave. Is tropical cyclone intensity guidance improving. *Bulletin of the American Meteorological Society*, 95(3):387–398, 2014.

[19] D. Durran and J. Klemp. A compressible model for the simulation of moist mountain waves. *Monthly Weather Review*, 111(12):2341–2361, 1983.

[20] K. Emanuel. An air-sea interaction theory for tropical cyclones. part i:steady-state maintenance. *Journal of Atmospheric Sciences*, 43:585–604, 1986.

[21] Kerry Emanuel. Some aspects of hurricane inner-core dynamics and energetics. *Journal of Atmospheric Sciences*, 54(8):1014–1026, 1997.

[22] R. C. Foster. Why rolls are prevalent in the hurricane boundary layer. *Journal of Atmospheric Sciences*, 62:2647–2661, 2005.

[23] S. Gaberšek, F. X. Giraldo, and J. Doyle. Dry and moist idealized experiments with a two-dimensional spectral element model. *Monthly Weather Review*, 140:3163–3182, 2012.

[24] T. Gal-Chen and R. Somerville. On the use of a coordinate transformation for the solution of the navier-stokes equations. *Journal of Computational Physics*, 17:209–228, 1975.

[25] C. Geuzaine and J. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Internation Journal for Numerical Methods in Engineering*, 0:1–24, 2009.

[26] F. X. Giraldo. *An Introduction to Element-based Galerkin Methods on Tensor-Product Bases - Analysis, Algorithms, and Applications*, volume 24. Springer, Gewerbestrasse, Switzerland, 2020.

[27] F. X. Giraldo and T. E. Rosmond. A scalable spectral element Eulerian atmospheric model (SEE-AM) for NWP: Dynamical core tests. *Monthly Weather Review*, 132(1):133–153, 2004.

[28] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969.

[29] S. Guimond, J. M. Reisner, S. Marras, and F. X. Giraldo. The impacts of dry dynamic cores on asymmetric hurricane intensification. *Journal of Atmospheric Sciences*, 73:4661–4684, 2016.

[30] S. R. Guimond, M. Bourassa, and P. Reasor. A latent heat retrieval and its effects on the intensity and structure change of hurricane guillermo (1997). part i: The algorithm and observations. *Journal of the Atmospheric Sciences*, 68(8):1549–1567, 2011.

[31] S. R. Guimond, P. Reasor, and A. Didalake. The rapid intensification of hurricane karl (2010): New remote sensing observations of convective bursts from the global hawk platform. *Journal of Atmospheric Sciences*, 73:3617–3639, 2016.

[32] S. R. Guimond, P. Reasor, G. M. Heymsfield, and M. McLinden. Multiscale observations of hurricane dennis(2005): The effects of hot towers on rapid intensification. *Journal of Atmospheric Sciences*, 67, 2010.

[33] S. R. Guimond, J. M. Reisner, S. Marras, and F . X. Giraldo. The impacts on dry dynamic cores on asymmetric hurricane intensification. *Journal of Atmospheric Sciences*, 73:4661, 2016.

[34] S. R. Guimond, S. Sroke, and D. Proztko. A large eddy simulation of hurricane intensification. *33rd Conference on Hurricanes and Tropical Meterology, American Meteorological Society*, 17, 2008.

[35] S. R. Guimond, J. A. Zhang, J. W. Sapp, and S. J. Frasier. Coherent turbulence in the boundary layer of Hurricane Rita (2005) during an eyewall replacement cycle. *Journal of Atmospheric Sciences*, 75:3071–3093, 2018.

[36] T. A. Guinn and W. H. Schubert. Hurricane spiral bands. *Journal of Atmospheric Sciences*, 50:3380–3403, 1993.

[37] M. B. Hasan, S. R. Guimond, M. L. Yu, S. Reddy, and F. X. Giraldo. The effects of numerical dissipation on hurricane rapid intensification with observational heating. *Journal of Advances in Modeling Earth Systems*, 14(8), 2022.

[38] E. A. Hendricks, M. T. Montgomery, and C. A. Davis. The role of "vortical" hot towers in the formation of tropical cyclone diana (1984). *Journal of Atmospheric Sciences*, 61, 2004.

[39] E. A. Hendricks, W. H. Schubert, R. K. Taft, H. Wang, and J. P. Kossin. Lifecycle of hurricane-like vorticity rings. *Journal of Atmospheric Sciences*, 66:705–722, 2009.

[40] E.A. Hendricks, M. Kopera, F.X. Giraldo, S.M. Peng, J.D. Doyle, and Q. Jiang. Evaluation of the utility of static and adaptive mesh refinement for idealized tropical cyclone problems in a spectral element shallow-water model. *Monthly Weather Review*, 144:3697–3724, 2016.

[41] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, volume 54. Springer-Verlag New York Inc, 2008.

[42] K. A. Hill and G. M. Lackmann. Analysis of idealized tropical cyclone simulations using the weather research and forecasting model: Sensitivity to turbulence parametrization and grid spacing. *Monthly Weather Review*, 137:745–765, 2009.

[43] S. Hong. A new stable boundary-layer mixing scheme and its impact on the simulated east asian summer monsoon. *Quarterly Journal of the Royal Meteorological Society*, 146:1481–1496, 2010.

[44] S. Hong, Y. Noh, and J. Dudhia. A new vertical diffusion package with an explicit treatment of entrainment processes. *Monthly Weather Review*, 134(9):2318–2341, 2006.

[45] R A. Houze. *Cloud Dynamics*. Academic Press, Berkeley, California, U.S.A, 1993.

[46] R. A. Houze, S. Chen, B. Smull, W. Lee, and M. Bell. Hurricane intensity and eyewall replacement. *Science*, 315:1235–1239, 2007.

[47] J. Ito, T. Oizumi, and H. Niino. Near-surface coherent structures explored by large-eddy simulation of entire tropical cyclones. *Nature Scientific Reports*, 7:3798, 2017.

[48] P. A. Jiménez, J. Dudhia', J. F. González-Rouco, J. Navarro, and J. P. Montávez. A revised scheme for the wrf surface layer formulation. *Monthly Weather Review*, 140(3):898–918, 2012.

[49] C. L. Jordan. Mean soundings for the west indies area. *Journal of Meteorology*, 15:91–97, 1958.

[50] G. Karniadakis and S. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2nd Ed., Oxford, United Kingdom, 2005.

[51] J. F. Kelly and F. X. Giraldo. Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: limited-area mode. *Journal of Computational Physics*, 231:7988–8008, 2012.

[52] E. Kessler. On the distribution and continuity of water substance in atmospheric circulation. *Meteorological Monographs*, 10:32, 1969.

[53] J. Klemp and R. Wilhelmson. The simulation of three-dimensional convective storm dynamics. *Journal of Atmospheric Sciences*, 35:1070–1096, 1978.

[54] J. B. Klemp, J. Dudhia, and A. D. Hassiotis. An upper gravity-wave absorbing layer for nwp applications. *Monthly Weather Review*, 136(10):3987–4004, 2008.

[55] J. B. Klemp and D. K. Lilly. Numerical simulation of hydrostatic mountain waves. *Journal of Atmospheric Sciences*, 35(1):78–107, 1978.

[56] J.B. Klemp, W.C. Skamarock, and S.-H. Park. Idealized global nonhydrostatic atmospheric test cases on a reduced-radius sphere. *Journal of Advances in Modeling Earth Systems*, 7(3):1155–1177, 2015.

[57] T. Knutson, S. Camargo, E. Kerry, H. Chang-Hoi, , K. James, and et al. Tropical cyclones and climate change assessment: Part ii. projected response to anthropogenic warming. *Bulletin of the American Meteorological Society*, 101(3):E303–E322, 2019.

[58] T. Knutson, J. Sirutis, M. Zhao, R. Tuleya, M. Bender, G. Vecchi, and et al. Global projections of intense tropical cyclone activity for the late twenty-first century from dynamical downscaling of downscaling of cmip5/rcp4.5 scenarios. *Journal of Climate*, 28(18):7203–7224, 2015.

[59] M. A. Kopera and F. X. Giraldo. Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations. *Journal of Computational Physics*, 275:92–117, 2014.

[60] Michal A Kopera and Francis X Giraldo. Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations. *Journal of Computational Physics*, 275:92–117, 2014.

[61] D. Kopriva. *Implementing Spectral Methods for Partial Differential Equations*. Springer, New York, New York, U.S.A., 2008.

[62] J.P Kossin and W. H. Schubert. Mesovortices, polygonal flow patternsa and rapid pressure falls in hurricane-like vortices. *Journal of Atmospheric Sciences*, 58:2196–2209, 2001.

[63] J. W. Lavelle and W. C. Thacker. A pretty good sponge: Dealing with open boundaries in limited-area ocean models. *Ocean Modelling*, 20(3):270–292, 2008.

[64] A. Leonard. Energy cascade in large eddy simulations of turbulent fluid flows. *Advances in Geophysics*, 18:237–248, 1974.

[65] R.J. LeVeque, D.L. George, and M.J. Berger. Tsunami modelling with adaptively refined finite volume methods. *Acta Numerica*, 20:211–289, 2011.

[66] J. Li, F. Fang, J. Steppeler, J. Zhu, Y. Cheng, and X. Wu. Demonstration of a three-dimensional dynamically adaptive atmospheric dynamic framework for the simulation of mountain waves. *Meteorology and Atmospheric Physics*, 2021.

[67] L. Li and P. Chakraborty. Slower decay of landfalling hurricanes in a warming world. *Nature*, 587:230–234, 2020.

[68] D. K. Lilly. On the numerical simulation of buoyant convection. *Tellus*, 14:148–172, 1962.

[69] F. Marks and L. Shay. Landfalling tropical cyclones: Forecast problems and associated research opportunities. *Bulletin of the American Meteorological Society*, 79:305–323, 1998.

[70] S. Marras and F X. Giraldo. A parameter-free dynamic alternative to hyper-viscosity for coupled transport equations: application to the simulation of 3D squall lines using spectral elements. *Journal of Computational Physics*, 283:360–373, 2015.

[71] S. Marras, J. F. Kelly, M. Moragues, A. M uller, A. M. Kopera, M. Vásquez, F. X. Giraldo, G. Houzeaux, and O. Jorba. A review of element-based galerkin methods for numerical weather prediction: Finite elements, spectral elements, and discontinuous galerkin. *Archives of Computational Methods in Engineering*, 23:673–722, 2015.

[72] S. Marras, M. A. Kopera, E. M. Constantinescu, J. Suckale, and F. X. Giraldo. A residual-based shock capturing scheme for the continuous/discontinuous spectral element solution of the 2d shallow water equations. *Advances in Water Resources*, 114:45–63, 2018.

[73] S. Marras, M. Moragues, M. Vázquez, O. Jorba, and G. Houzeaux. A Variational Multiscale Stabilized finite element method for the solution of the Euler equations of nonhydrostatic stratified flows. *Journal of Computational Physics*, 236:380–407, 2013.

[74] S. Marras, M. Moragues, M R. Vázquez, O. Jorba, and G. Houzeaux. Simulations of moist convection by a variational multiscale stabilized finite element method. *Journal of Computational Physics*, 252:195–218, 2013.

[75] S. Marras and Y. Tissaoui. Jexpresso `https://github.com/smarras79/Jexpresso` 06/18/2024.

[76] P. McCorquodale, P. A. Ulrich, H. Johansen, and P. Colella. An adaptive multiblock high-order finite-volume method for solving the shallow-water equations on the sphere. *Communications in Applied Mathematics and Computational Science*, 10:121–162, 2015.

[77] J. Molinari and D. Vollaro. Rapid intensification of a sheared tropical storm. *Monthly Weather Review*, 138, 2010.

[78] M. Montgomery, G. Kilroy, R. Smith, and N. Crinivic. Contribution of mean and eddy momentum processes to tropical cyclone intensification. *Quarterly Journal of the Royal Meteorological Society*, 146:3101–3117, 2020.

[79] M. Montgomery, M. Nicholls, T. Cram, and A. Saunders. A vortical hot tower route to tropical cyclogenesis. *Journal of Atmospheric Sciences*, 63:355–386, 2006.

[80] M. T. Montgomery and R. J. Kallenbach. A theory for vortex rossby waves and its application to spiral bands and intensity changes in hurricanes. *Quarterly Journal of the Royal Meteorological Society Soc.*, 123:435–465, 1997.

[81] A. Mueller, J. Behrens, F. X. Giraldo, and V. Wirth. Comparison between adaptive and uniform deiscontinuous Galerkin simulations in 2D dry bubble experiments. *Journal of Computational Physics*, 235:371–393, 2013.

[82] A. Müller, M. Kopera, S. Marras, L. C. Wilcox, T. Isaac, and F. X. Giraldo. Strong scaling for numerical weather prediction at petascale with the atmospheric model NUMA. *The International Journal of High Performance Computing Applications*, 33:411–426, 2018.

[83] D. S. Nolan. Evaluating environmental favorableness for tropical cyclone development with the method of point-downscaling. *Journal of Advances in Modeling Earth Systems*, 3:1–28, 2011.

[84] D. S. Nolan and L. D. Grasso. Nonhydrostatic, three dimensional perturbations to balanced, hurricane-like vortices, part ii: Symmetric response and nonlinear simulations. *Journal of Atmospheric Sciences*, 60:2717–2745, 2003.

[85] D. S. Nolan and L. D. Grasso. Three-dimensional, nonhydrostatic perturbations to balanced, hurricane-like vortices. part ii: Symmetric response and nonlinear simulations. *Journal of the Atmospheric Sciences*, 60:2717–2745, 2003.

[86] D. S. Nolan, Y. Moon, and D. P. Stern. Tropical cyclone intensification from asymmetric convection: Energetics and efficiency. *Journal of the Atmospheric Sciences*, 61:3377–3405, 2007.

[87] Y. Ogura and T. Takahashi. Numerical simulation of the lifecycle of a thunderstorm cell. *Monthly Weather Review*, 99:895–911, 1971.

[88] J. Persing, M. Montgomery, J. C. McWilliams, and R. K. Smith. Asymmetric and axisymmetric dynamics of tropical cyclones. *Atmospheric Chemistry and Physics*, 13:12299–12341, 2013.

[89] E. Rappaport, J. Franklin, L. Avila, S. Baig, J. Beven, E. S. Blake, and et al. Advances and challenges at the national hurricane center. *Weather and Forecasting*, 24(2):395–419, 2009.

[90] R. Rogers, P. Reasor, and S. Lorsolo. Airborne doppler observations of the inner-core structural differences between intensifying and steady state tropcial cyclones. *Monthly Weather Review*, 141:2970–2991, 2013.

[91] R. Rotunno, C. Chen, W. Wang, C. Davis, J. Dudhia, and J. H. Holland. Large-eddy simulation of an idealized tropical cyclone. *Bulletin American Meteorological Society*, 90:1783–1788, 2009.

[92] R. Rotunno, J B. Klemp, and M L. Weisman. A theory for strong, long-lived squall lines. *Journal of Atmospheric Sciences*, 45:463–485, 1988.

[93] C. Schar, D. Leuenberger, O. Fuhrer, D. Luthic, and C. Girard. A new terrain-following vertical coordinate formulation for atmospheric prediction models. *Monthly Weather Review*, 130:2459–2480, 2002.

[94] W. H. Schubert, M. T. Montgomery, R. K. Taft, T. A. Guinn, S. R. Fulton, J.P Kossin, and J. P. Edwards. Polygonal eyewalls, asymmetric eye contraction, and potential vorticity mixing in hurricanes. *Journal of Atmospheric Sciences*, 56:1197–1223, 1999.

[95] J. Shen. Stable and efficient spectral methods in unbounded domains using laguerre functions. *SIAM Journal on Numerical Analysis*, 38(4):1113–1133, 2000.

[96] J. Shen and L.-L. Wang. Some recent advances on spectral methods for unbounded domains. *Communications in Computational Physics*, 5(2-4):195–241, 2009.

[97] S. Sherwin and G. Karniadakis. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, Oxford, United Kingdom, 2005.

[98] W. Skamarock and J. Klemp. Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow. *American Meteorological Society*, 121:788–804, 1993.

[99] W. Skamarock, J. Oliger, and RL L. Street. Adaptive grid refinement for numerical weather prediction. *Journal of Computational Physics*, 80(1):27–60, 1 1989.

[100] J. Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiement. *Monthly Weather Review*, 91:99–164, 1963.

[101] R. B. Smith. Linear theory of stratified hydrostatic flow past an isolated mountain. *Tellus*, 32(4):348–364, 1980.

[102] S. Soong and Y. Ogura. A comparison between axisymmetric and slab-symmetric cumulus cloud models. *Journal of Atmospheric Sciences*, 30:879–893, 1973.

[103] A. Sridhar, Y. Tissaoui, S. Marras, Z. Shen, C. Kawczynski, s. Byrne, K. Pamnany, M. Waruszewski, T.H. Gibson, J.E. Kozdon, V. Churavy, L. C. Wilcox, F. X. Giraldo, and T. Schneider. Large-eddy simulations with ClimateMachine v0.2.0: a new open-source code for atmospheric simulations on GPUs and CPUs. *Geoscientific Model Development*, gmd-2021-335, 2022.

[104] S. Sroka and S. R. Guimond. Organized kinetic energy backscatter in the hurricane boundary layer from radar measurements. *Journal of Fluid Mechanics*, 924, 2021.

[105] Y. Tissaoui, S. Marras, A. Quaini, F. Barranca, and F. X. Giraldo. A non-column based, fully unstructured implementation of Kessler microphysics with warm rain using continuous and discontinuous spectral elements. *Journal of Advances in Modeling Earth Systems*, 2022MS003283, 2023.

[106] F. Vismara and T. Benacchio. Efficient hyperbolic–parabolic models on multi-dimensional unbounded domains using an extended DG approach. *International Journal for Numerical Methods in Fluids*, 96(2):125–230, 2023.

[107] F. Vismara, T. Benacchio, and L. Bonaventura. A seamless, extended DG approach for advection–diffusion problems on unbounded domains. *Journal of Scientific Computing*, 90(5):1–27, 2022.

[108] Y. Wang. Vortex rossby waves in a numerically simulated tropical cyclone. part i: Overall structure, potential vorticity, and kinetic energy budgets. *Journal of Atmospheric Sciences*, 59:1213–1238, 2002.

[109] M. L. Weisman and J. B. Klemp. The dependence of numerically simulated convective storms on vertical wind shear and buoyancy. *Monthly Weather Review*, 110:504–520, 1982.

[110] M. L. Weisman, J. B. Klemp, and R. Rotunno. Structure and evolution of numerically simulated squall lines. *Journal of Atmospheric Sciences*, 45:1990–2013, 1988.

[111] M. L. Weisman and R. Rotunno. A theory for strong long- lived squall lines revisited. *Journal of the Atmospheric Sciences*, 61:361, 2004.

[112] M. L. Weisman, W. C. Skamarock, and J. B. Klemp. The resolution dependence of explicitly modeled convective systems. *Monthly Weather Review*, 125:527, 1997.

[113] H. E. Willoughby, J. A. Clos, and M. G. Shoreibah. Concentric eye walls, secondary wind maxima, and the evolution of the hurricane vortex. *Journal of Atmospheric Sciences*, 39:395–411, 1982.

[114] L. Wu, B. Wang, and S. A. Braun. Impact of air-sea interaction on tropical cyclone track and intensity. *Monthly Weather Review*, 133:3299–3314, 2005.

[115] H. Yamazaki, H. Weller, C. J. Cotter, and P. A. Browne. Conservation with moving meshes over orography. *Journal of Computational Physics*, 461:111217, 2022.