LLNL-TH-431627

Immersed Boundary Methods for High-Resolution Simulation of Atmospheric Boundary-Layer Flow Over Complex Terrain

K. A. Lundquist

May 13, 2010

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Immersed Boundary Methods for High-Resolution Simulation of Atmospheric Boundary-Layer Flow Over Complex Terrain

by

Katherine Ann Lundquist

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in

Engineering - Mechanical Engineering

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Fotini Katopodes Chow, Co-Chair
Professor Philip S. Marcus, Co-Chair
Professor Ronald W. Yeung
Professor Mark T. Stacey
Professor Julie K. Lundquist

Spring 2010

Immersed Boundary Methods for High-Resolution Simulation of Atmospheric
Boundary-Layer Flow Over Complex Terrain

Abstract

Immersed Boundary Methods for High-Resolution Simulation of Atmospheric Boundary-Layer Flow Over Complex Terrain

by

Katherine Ann Lundquist

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Fotini Katopodes Chow, Co-Chair

Professor Philip S. Marcus, Co-Chair

Mesoscale models, such as the Weather Research and Forecasting (WRF) model, are increasingly used for high resolution simulations, particularly in complex terrain, but errors associated with terrain-following coordinates degrade the accuracy of the solution. Use of an alternative Cartesian gridding technique, known as an immersed boundary method (IBM), alleviates coordinate transformation errors and eliminates restrictions on terrain slope which currently limit mesoscale models to slowly varying terrain. In this dissertation, an immersed boundary method is developed for use in numerical weather prediction. Use of the method facilitates explicit resolution of complex terrain, even urban terrain, in the WRF mesoscale model.

First, the errors that arise in the WRF model when complex terrain is present are presented. This is accomplished using a scalar advection test case, and comparing the numerical solution to the analytical solution. Results are presented for different orders of advection schemes, grid resolutions and aspect ratios, as well as various degrees of terrain slope. For comparison, results from the same simulation are presented using the IBM.

Both two-dimensional and three-dimensional immersed boundary methods are then described, along with details that are specific to the implementation of IBM in the WRF code. Our IBM is capable of imposing both Dirichlet and Neumann boundary conditions. Additionally, a method for coupling atmospheric physics parameterizations at the immersed boundary is presented, making IB methods much more functional in the context of numerical weather prediction models. The two-dimensional IB method is verified through comparisons of solutions for gentle terrain slopes when using IBM and terrain-following grids. The canonical case of flow over a Witch of Agnesi hill provides validation of the basic no-slip and zero gradient boundary conditions. Specified diurnal heating in a valley, producing anabatic winds, is used to validate the use of flux (non-zero) boundary conditions. This anabatic flow set-up is further coupled to atmospheric physics parameterizations, which calculate surface fluxes, demonstrating that the IBM can be coupled to various land-surface parameterizations in atmospheric models.

Additionally, the IB method is extended to three dimensions, using both trilinear and inverse distance weighted interpolations. Results are presented for geostrophic flow over a three-dimensional hill. It is found that while the IB method using trilinear interpolation works well for simple three-dimensional geometries, a more flexible and robust method is needed for extremely complex geometries, as found in three-dimensional urban environments. A second, more flexible, immersed boundary method is devised using inverse distance weighting, and results are compared to the first IBM approach. Additionally, the functionality to nest a domain with resolved complex geometry inside of a parent domain without resolved complex geometry is described. The new IBM approach is used to model urban terrain from Oklahoma City in a one-way nested configuration, where lateral boundary conditions are provided by the parent domain.

Finally, the IB method is extended to include wall model parameterizations for rough surfaces. Two possible implementations are presented, one which uses the log law to reconstruct velocities exterior to the solid domain, and one which reconstructs shear stress at the immersed boundary, rather than velocity. These methods are tested on the three-dimensional canonical case of neutral atmospheric boundary layer flow over flat terrain.

To my parents.

Contents

List of Figures

List of Tables

Acknowledgements

My sincerest thank you goes to my advisor at UC Berkeley, Tina Katopodes Chow, who possessed a seemingly unwavering believe in my abilities as a researcher. She believed in me, even when I doubted myself, happily cheering me on at intimidating events such as my first conference presentation and my qualifying examination. Her support has been greatly appreciated. At the beginning of this journey, she patiently answered my questions, assuaging my insecurities by exuding a non-judgemental demeanor. Later in graduate school, I came to appreciate our lively research discussions and the knowledge that she passed to me, gained from her own endeavors in the field of computational fluid mechanics. It is not often that we are blessed with relationships that fundamentally change the core of our beings for the better, but I can honestly say that I would not be the person who I am today without her influence.

Additionally, thanks go to Julie Lundquist, who mentored me at Lawrence Livermore National Laboratory. She patiently helped me to bridge the gap between traditional fluid mechanics and atmospheric science. It has also been through my interactions with her that I have learned about important processes such as scientific collaborations and the art of writing a research proposal.

I also thank my dissertation committee Mark Stacy, Philip Marcus, and Ronald Yeung for their guidance and input. Mark Stacey always showed enthusiasm for my research, and asked insightful questions at the many presentations he saw me give in environmental fluid mechanics group meetings. Additional thanks go to Stanley Berger for serving as the chair of my qualifying exam, and to Jon Wilkening and Stephen Morris who were also committee members.

Thanks go to the entire environmental fluid mechanics group at UC Berkeley, who have made graduate school a fun endeavor. I have been very lucky to find great friends in my lab mates. Special thanks go to Megan Daniels, Mary Cousins, Lissa MacVean, and Rebecca Leonardson who have been a source of unconditional support and camaraderie. I feel very lucky to have shared this experience with them. My office mates Bowen Zhou, Maureen Downing-Kunz, Nikola Marjanović, and Maureen Martin have been a pleasure to work with.

During my last few years in graduate school, I spent much of my time at LLNL. The post-docs and scientists there have greatly influenced my work. Thank you to Branko Kosović and Jeff Mirocha who's door was always open to discuss technical issues. Branko will be collaborating with me on continued efforts to expand the work in chapter 8 modeling surface momentum fluxes to complex terrain. Jeff Mirocha was a new post-doc at the lab, while I was a new doctoral student. Additionally, we work with the same numerical code (WRF). It has been fun to watch his journey learning about large-eddy simulation, while I learned about the immersed boundary method. He has also proved invaluable in discussing the nuts and bolts of the model, as well as providing advice on debugging and the set-up of simulations. Sonia Wharton, Gokhan Kirkil, Steve Chin, Peter Caldwell, Matthew Simpson, Celine Bonfils, Peter Goldstein, Bill Henshaw, Mike Singer, Kyle Chand, Lee Glascoe, Jeff Roberts, Dave Bader, and Gavin Bell have all made LLNL an enjoyable place to work, and convinced me to stay at LLNL after graduation. I look forward to continued collaborations and friendships in the near future. I am also grateful to Lourdes Placeres who helped me to successfully navigate the bureaucracy at the lab, and always does so with a terrific smile and laugh .

I thank many of the scientists at the National Center for Atmospheric Research for their interest in my work. Bill Skamarock, Joe Klemp, and Jimy Dudhia have provided insightful discussions on the numerical code. Tom Warner and Andrzej Wyszogrodzki invited me to visit NCAR for collaborations. Additionally, I thank Peter Sullivan and Bernard Geurts for their inspirational and thoughtful lectures at the atmospheric boundary layer summer school that I attended in France. Finally, I would also like to thank Juerg Schmidli, Luca Delle Monache, Reed Maxwell, Oliver Fuhrer, Wayne Angevine, Xylar Asay-Davis, Todd Ringler, Steven Krueger, and many others who have helped me gain insight along the way.

I am grateful to my family and friends for their continued support and encouragement. I am especially grateful for my parents, John and Judy Lundquist, who provide unconditional love, and instilled a curiosity about the world in me at a young age that surely lead me study the sciences. They always pushed me to do my best work, and always desired the best for me. So much so, that my father once declared that he 'would give his right arm for me to attend Stanford' (Dad, I am sorry if Berkeley is a disappointment). This dissertation is dedicated to them. I am also thankful to have wonderful parents-in-law, Tom and Jody Case, and sisters-in-law Martha and Michelle Case. Finally, my husband Justin Case, has been at my side providing love and encouragement throughout graduate school. I am blessed to have him as my partner in life.

Financial support for this work was provided by the Lawrence Scholars Program at Lawrence Livermore National Laboratory. The fellowship program is gratefully acknowledged, as is the Physical and Life Sciences Directorate, which provided access to the LLNL supercomputing facilities as well as other resources needed for this work.

Chapter 1

Introduction and overview

1.1 Motivation

The atmospheric boundary layer (ABL) is defined as the bottom layer of the atmosphere that is directly in contact with the earth's surface [Stull, 1988]. It is characterized as a region whose structure is strongly dependent on surface fluxes of momentum, heat, and moisture. It is also the portion of the atmosphere that we inhabit, and as such, there is great interest in modeling the physical processes of the ABL. As the availability of large computing platforms increases, simulations of the ABL are becoming more widely used for a variety of applications including weather and storm forecasting, predictions of air quality and contaminant transport, and wildfire modeling. Additionally, the output from meteorological models is used to analyze wind loads on structures such as buildings and wind turbines, and even in snow dynamics models which predict avalanches on snow covered mountain peaks.

For each of these applications accurate prediction of velocity fields and scalars are required in the presence of complex terrain. Terrain significantly affects the flow direction and structure of the ABL. One effect of topography is to divert or channel mean flow horizontally around topographic obstacles. Perhaps more significant is the effect of topographically induced vertical velocity. Because our atmosphere is stably stratified and laden with water vapor, vertical motion results in phenomena such as mountain lee waves and orographic precipitation. Yet, despite the impact that flow predictions have on our lives, the task of including topography in today's numerical weather prediction models is still challenging and a large source of uncertainty and error.

Complex terrain exists at every scale, and larger terrain features (relative to the grid spacing) are resolved in the numerical model, while unresolved features are parameterized. At the microscale (2 mm to 2 km), where it is possible to resolve urban scale terrain, accurate flow fields can be used to predict dispersion of hazardous substances such as those released in an industrial accident or malicious attack. Accurate plume predictions are vital in densely populated urban areas where they are used to aid in emergency response planning. Current mesoscale models cannot explicitly resolve the terrain in urban regions, thereby missing important three-dimensional heterogeneous processes such as the generation of turbulent kinetic energy and mechanical dispersion of the contaminant by the buildings. Urban parameterizations (where the buildings are treated as roughness elements) can over-predict wind velocity and shear. These over-predicted quantities lead to faster transport and increased dispersion, and consequently an underestimation of the contaminant concentration and associated risks.

At the mesoscale (2 to 2000 km), where mountainous terrain is resolved, flow predictions are used in the planning of wind farms. Currently, average power production is systematically over-predicted by 11% to 13% [Jones and Randall, 2006; Jones, 2008]. The main causes of this over-prediction are biases in the assessment of wind availability, wind resource prediction, and energy loss prediction [Johnson et al., 2008]. Examples of wind resource prediction biases are errors in wind flow modeling (especially the inclusion of topographic effects), errors associated with vertical extrapolation to the turbine hub height, and inaccurate assessments of velocity shear and turbulence generated in the boundary layer (possibly by complex terrain). Walter et al. [2009] demonstrated that up to 3% of the over-prediction could be accounted for by including the effects of velocity shear over the span of the rotor. Improved simulations with complex terrain address the wind resource prediction errors, and can lead to better turbine design, wind farm layouts, and predictions of operational power output. Clearly an accurate and robust method for modeling complex terrain would enhance simulation capabilities and improve our ability to forecast for a variety of applications.

1.2 Background

1.2.1 Differences between computational fluid dynamics and numerical weather prediction

Both computational fluid dynamics (CFD) models and numerical weather prediction (NWP) models (also known as mesoscale, limited area, or regional models) solve the Navier-Stokes equations. Yet, despite solving the same basic governing equations, these two types of models were designed for use at different length scales. As discussed in Wyngaard [2004], in mesoscale modeling, turbulence is not resolved because the length scale of the energy containing turbulence is much smaller than the grid scale, which is used as an implicit filter length for the turbulence closure. In computational fluid dynamics, and specifically large-eddy simulations (LES), turbulence is resolved because the filter width is much larger than the turbulent length scales. In the past, the ratio of turbulent length scale to filter width was either very large or small, however, advances in computing have led to simulations where the ratio is on the order of one. Simulations at this resolution have been coined as the “Terra Incognita” by Wyngaard, and many open research questions exist regarding numerical modeling at this scale. The method developed in this dissertation is an example of how this modeling gap can be bridged, and therefore is also the scale at which many of the simulations described here take place.

Microscale simulations of flow in complex geometries have traditionally been preformed with CFD codes that solve the incompressible Navier-Stokes equations with an elliptical Poisson equation for pressure gradients. Buoyancy effects are included with the Boussinesq approximation, and other atmospheric physics such as differential surface heating, moisture, and unresolved surface roughness are neglected. These models often use boundary-fitted grids that are either curvilinear or unstructured, and handle complex terrain well. Lateral boundaries conditions are idealized, so that the boundaries are forced with idealized inlet profiles which neglect regional weather effects. Furthermore, rigid lids are often used at the top boundary so that while the domain top is much less than the height of the ABL, vertical fluxes are not permitted. These models most commonly use an eddy viscosity type of turbulence closure, either based on a Reynolds Averaged Navier-Stokes (RANS) or LES approach.

On the other hand, non-hydrostatic mesoscale simulations solve the fully compressible Navier-Stokes equations, because the domain height is generally larger than the scale height of the atmosphere (7 to 8 km), so that compressibility effects must be considered. Fast (acoustic and Lamb) modes are commonly dealt with through a combination of time-split integration schemes and filtering. A conservation equation for potential temperature is solved, and pressure values are diagnosed from an equation of state. Additional conservation equations are solved for moisture (vapor, cloud, rain, ice, etc.) as well as any other scalars to be considered, such as chemical species. Surface fluxes of heat, moisture, and momentum are calculated by atmospheric physics parameterizations that model solar radiation, soil physics, cumulus formation, etc. Mesoscale models use equations which are written for spherical coordinates on the globe, and then they are projected onto planar surfaces using a map projection (conical, cylindrical, etc.). Vertical coordinates can be based on pressure, height, or potential temperature, and are commonly mapped to follow the terrain (called ‘sigma’ coordinates for the variable used to represent the mapping function). Sigma coordinates are inherently limited in their ability to handle complex terrain, as steep slopes cause large skewness in the computational cells. Lateral boundary conditions are provided by actual meteorological data that has been assimilated and reanalyzed by a global model. Planetary boundary layer schemes parameterize vertical mixing, while high-order scale selective computational mixing is added in the horizontal dimensions.

Nesting higher-resolution ‘child’ domains within coarser ‘parent’ domains is common practice in mesoscale modeling. This functionality is the main feature which is enabling mesoscale models to be used at increasingly finer scales, and blurring the distinction between mesoscale and large-eddy simulations. For this reason, research-focused mesoscale models include LES turbulence closures. Chow et al. [2006] used a mesoscale model at LES scales to model flow in a valley. Moeng et al. [2007] examined the effect of two-way nesting on LES within LES simulations. Other open research questions include the effects of nesting LES simulations within simulations using a PBL scheme [Lundquist et al., 2008a, 2009a], modifications for using atmospheric parameterization at multiple scales [Talbot et al., 2010], and the use of icosahedral grids with gradually decreasing effective diameters to eliminate the need for nesting [Ringler et al., 2010]. In this work, we focus on methods for seamlessly representing complex terrain as more small scale terrain features are captured at increasingly higher resolutions. This behavior is illustrated in figure 1.1, where a slice of mountainous terrain, along with the resulting slope, is plotted at three different horizontal resolutions.

1.2.2 Complex terrain in numerical weather prediction

Terrain-induced mesoscale systems include features such as sea and land breezes, mountain-valley winds, urban circulations, and forced advection over topographic obstacles [Pielke, 1984]. Terrain is frequently represented in mesoscale models by mapping the vertical coordinate to the terrain. The most frequently used mapping function, $\bar{z} = \sigma = \frac{z_{top}(z - z_{ht}(x,y))}{z_{top} - z_{ht}(x,y)}$, was first proposed in Gal-Chen and Somerville [1975]. This function maps a domain with an irregular lower boundary onto a Cartesian grid, thereby simplifying the application of lower boundary conditions. Due to this advantage, this mapping function (or similar variants based on pressure or potential temperature and those using a non-dimensional formulation) is utilized in most modern mesoscale models. The disadvantage of this coordinate transformation is that it introduces additional terms into the governing equations. For example, the covariant velocity vectors which are perpendicular to coordinate surfaces (figure 1.2) are given by equation 1.1.

$$\begin{pmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \frac{z_{top} - \bar{z}}{z_{top}} \frac{\partial z_{ht}}{\partial x} \\ 0 & 1 & \frac{z_{top} - \bar{z}}{z_{top}} \frac{\partial z_{ht}}{\partial y} \\ 0 & 0 & \frac{z_{top} - z_{ht}}{z_{top}} \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{1.1}$$

In the above equation, the overbar denotes the quantity in the transformed coordinate, z_{top} is the top of the domain, and $z_{ht}(x,y)$ is the terrain height. The extra terms are not difficult to handle numerically, as long as the transformation is well-behaved. As noted by Gal-Chen and Somerville this transformation is only well-behaved for terrain with continuous second derivatives, and when the determinant of the Jacobian $(\frac{z_{top} - z_{ht}}{z_{top}})$ is close to unity. Therefore, there is a wide variety of terrain, for which this transformation should not be used.

Inaccuracies from the coordinate transformation are present in each spatially discretized term of the Navier-Stokes equations. Errors in the computation of horizontal pressure gradients [Janjić, 1977, 1989], diffusion [Zängl, 2002, 2003], and horizontal advection [Schär et al., 2002] terms have been noted in the presence of sloping coordinate surfaces and steep topog-

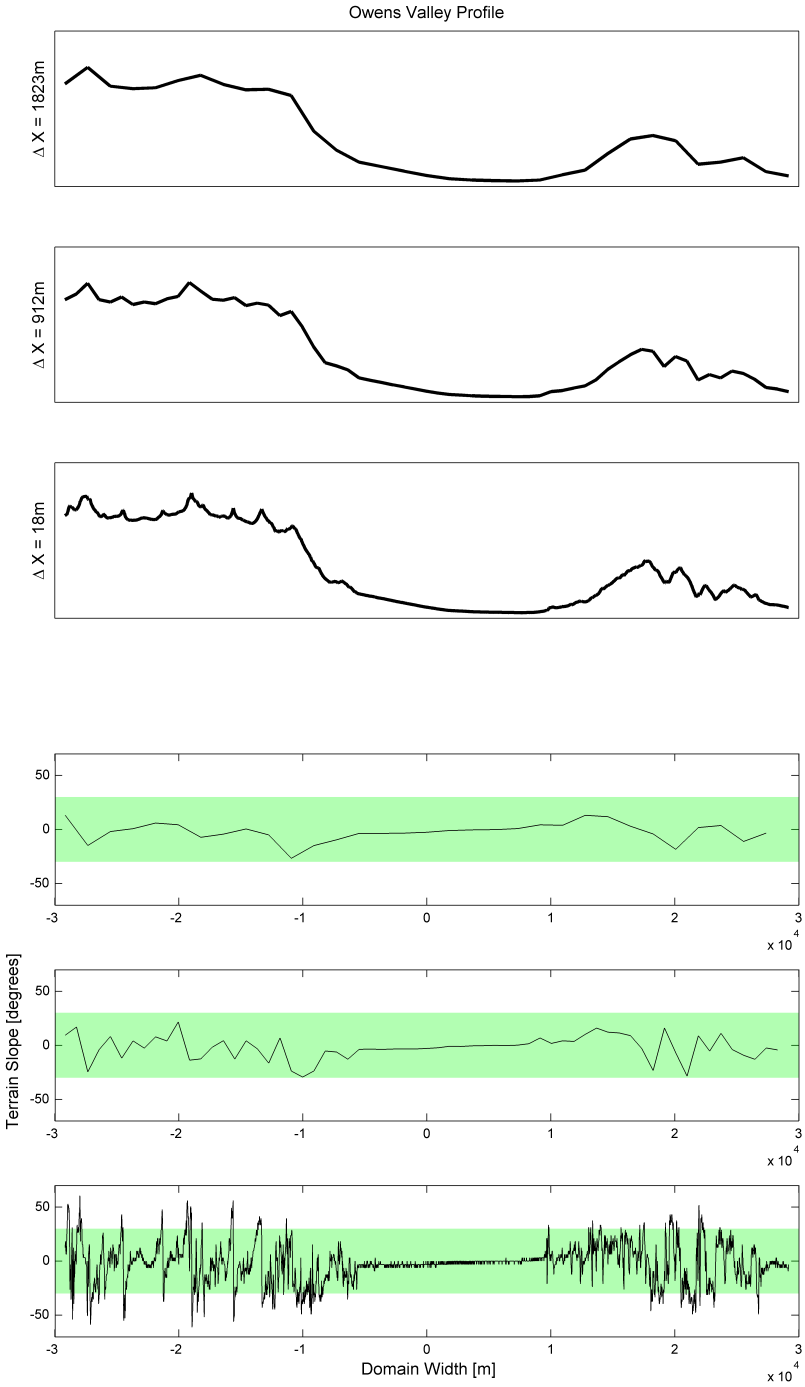


Figure 1.1. Terrain is sampled at three different resolutions (1823 m, 912 m, and 18 m) for a slice of terrain in Owens Valley, CA. As resolution increases, more fine scale terrain features are captured. This leads to increased terrain slopes (shown on the bottom for each of the three resolutions) which need to be represented in simulations of the ABL.

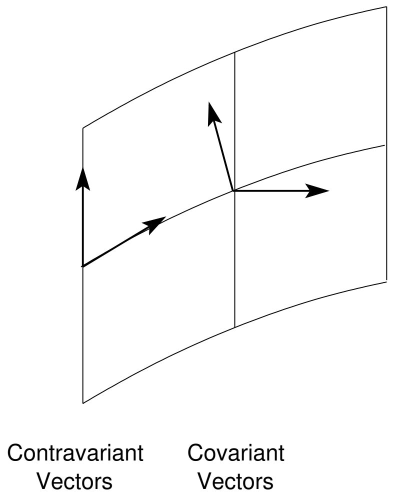


Figure 1.2. In a non-orthogonal coordinate system, the contravariant basis vectors are defined as tangent to the coordinate surfaces, and the covariant basis vectors are defined as perpendicular to coordinate surfaces.

raphy. Furthermore, Mahrer [1984] notes that the common practice of stretching the grid towards the bottom boundary to achieve fine vertical resolution near the surface, exacerbates these errors. Numerically inconsistent horizontal derivatives arise when the distance between two vertical grid points (Δz) is smaller than the elevation difference between two horizontally adjacent points, even when the metric term is included in the calculation (computing horizontal gradients along sloping surfaces without consideration for the metric terms is also commonplace). In the transformed coordinate, a horizontal derivative is calculated as:

$$\left.\frac{\partial}{\partial x}\right|_z = \left.\frac{\partial}{\partial x}\right|_{\bar{z}} + \left.\frac{\partial \bar{z}}{\partial x}\right|_z \frac{\partial}{\partial \bar{z}}.$$

In a forward finite differencing scheme, this derivative is approximated as:

$$\left.\frac{\partial F}{\partial x}\right|_z = \frac{F(i+1,j)-F(i,j)}{\Delta x} + \left.\frac{\partial \bar{z}}{\partial x}\right|_{i,j} \frac{F(i+1,j+1)-F(i+1,j)}{\Delta \bar{z}}.$$

The stencil used in this finite difference scheme is shown in figure 1.3. It can clearly be seen that for a true horizontal derivative in physical space, it would be more appropriate to use the computational points located at $F(i+1,j+3)$ and $F(i+1,j+4)$. Mahrer [1984] demonstrated a reduction in errors when using this modified stencil.

Geometrically, these errors are more likely to occur at large aspect ratios, when the horizontal grid spacing is much larger than the vertical grid spacing, making it more common for the vertical change in height over the horizontal grid spacing to be larger than the vertical grid spacing of the computational cell. This point is illustrated in figure 1.4, which depicts the skewness of computational cells as a function of terrain slope and aspect ratio. The gray area delineates the parameter space where numerical inconsistencies occur in horizontal derivatives. In practice, mesoscale modelers often stretch the grid towards the surface,

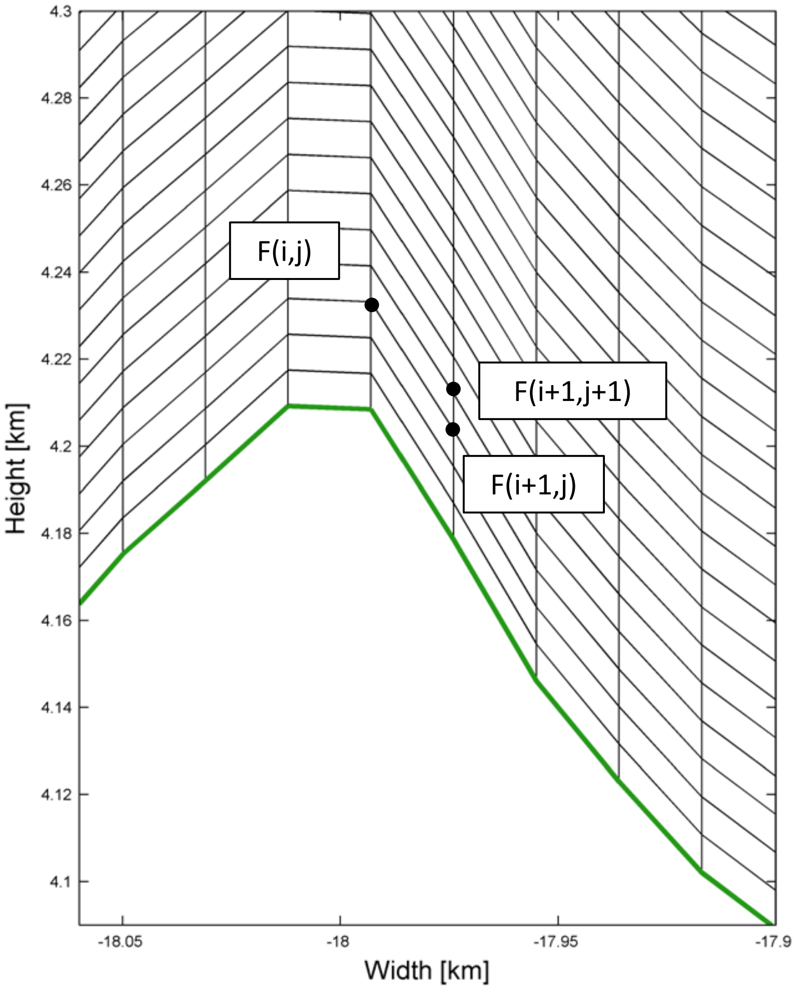


Figure 1.3. A schematic of the stencil used in a forward finite difference approximation of a horizontal derivative.

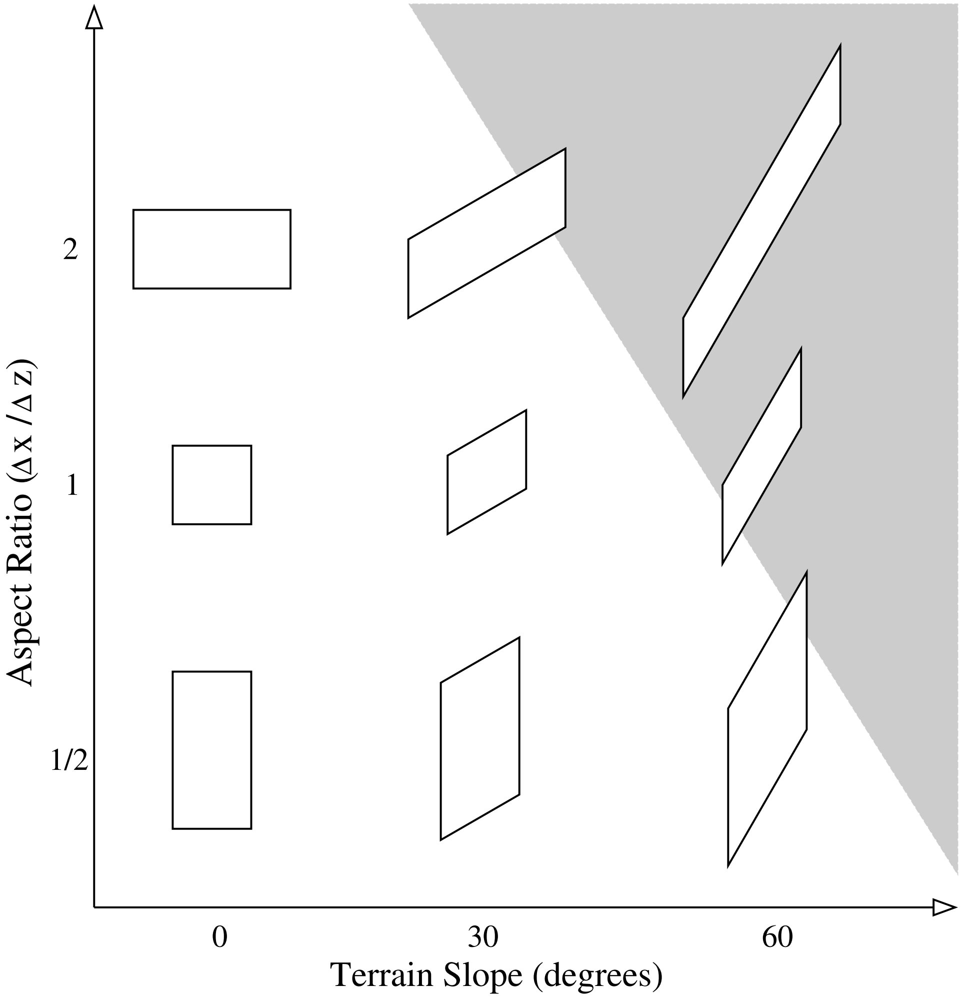


Figure 1.4. The magnitude of errors arising from the use of terrain-following coordinates is a function of terrain slope and aspect ratio. Example computational cells are shown at a variety of terrain slopes and aspect ratios. The shaded region denotes the parameter space where the use of terrain-following coordinates is questionable because the change in height over two horizontally adjacent points is larger than the vertical grid spacing Δz.

using very large grid aspect ratios. For example, a grid with 1 km horizontal spacing may have the first vertical grid point located at 50 m, leading to an aspect ratio of 20. From experience most modelers believe that the use of terrain-following coordinates is limited to approximately 30 degree slopes (shown as green in figure 1.1), however, at an aspect ratio of 20 the guidelines of Mahrer are violated with just a 2.9 degree terrain slope. It is clear from this plot that decreasing the aspect ratio or horizontal grid spacing (Δx) can reduce these errors.

1.2.3 Improvements for numerical simulations of flow over complex terrain

This research seeks to improve the ability of a numerical weather prediction model to predict flow over complex terrain (such as mountainous and urban terrain). Improvements to the mesoscale model are accomplished by modifying the approach used for meshing the computational domain by representing the bottom boundary conditions with an immersed boundary method (IBM). IBM allows the boundaries of complex surfaces to be represented on a Cartesian grid, eliminating difficulties arising from the terrain-following vertical coordinate system in the mesoscale model. Rather than creating or transforming a grid which conforms to the terrain, the effects of the boundaries are represented by an additional forcing term in the momentum equation. This is achieved by applying boundary conditions to nodes neighboring the immersed terrain boundary that enforce the desired boundary condition on the surface.

It will be possible to model flow over complex terrain, including an urban environment, in detail using a mesoscale model combined with the immersed boundary method. This will be the first time that a mesoscale model will include the functionality needed to model a domain at the urban scale by accurately representing complex building surfaces. Improved results from urban simulations will lead to more accurate numerical representations of physical processes such as contaminant dispersion. Once heterogeneous urban environments can be explicitly modeled, they can be nested into much larger mesoscale domains. This means that the larger mesoscale model will provide lateral boundary conditions with the correct forcing to the nested urban scale model. Regional mesoscale weather features will then be included in the urban-scale model instead of just simplified boundary conditions. Such seamless integration between the larger mesoscale model and the urban scale does not currently exist, so the IBM approach will have large impacts in the atmospheric modeling community.

1.3 Overview

The goal of this research is to develop improved surface representations that are suitable for accurately dealing with complex terrain. This will be accomplished by adding immersed boundary method capabilities to the Weather Research and Forecasting (WRF) model, and eliminating use of the terrain-following coordinate in complex terrain.

Several different implementations of IBM have been posed by researchers, and these are outlined in chapter 2. Common formulations of the forcing term and boundary reconstruction techniques are discussed. This chapter also includes a description of boundary conditions for large-eddy simulations, including a review of approximate boundary conditions (wall models). The different boundary conditions that can be achieved with IBM are described. IB methods are primarily used in direct numerical simulations (DNS) to impose no-slip boundary conditions for velocity, however, some previous literature exists in which the method is used in LES and extends the method for flux boundary conditions or allows the use of approximate boundary conditions to represent rough surfaces.

The mesoscale model used in this research, WRF, is described in chapter 3. This chapter includes a derivation of the governing equations used in WRF, which have been transformed into a terrain-following pressure-based coordinate system. Information is also included on temporal and spatial discretization schemes, and available native bottom boundary conditions.

Chapter 4 illustrates and analyzes the errors that arise in the WRF model when complex terrain is present. This is accomplished using the scalar advection test case of Schär et al. [2002], and comparing to the analytical solution. Results are presented for different orders of advection schemes, grid resolutions and aspect ratios, as well as various degrees of terrain slope. For comparison, results from the same simulation are presented using IBM. This allows us to quantify the error reduction from using our new gridding technique, providing justification for further development of the IBM method.

A two-dimensional immersed boundary method is described in chapter 5, along with details that are specific to the implementation of IBM in the WRF code. A new boundary condition option (no-slip) for the native terrain-following version of WRF is described, and is added to enable direct comparisons between solutions computed on the IBM grid and those computed using the terrain-following coordinate. Finally, as noted in Zängl [2003], previous Cartesian grid methods were not coupled to boundary layer parameterizations, and it was unknown how to accomplish this. A method for coupling atmospheric parameterization at the immersed boundary is presented in this chapter, making IB methods much more functional in the context of numerical weather prediction models.

The two-dimensional IB method is verified in chapter 6. This is accomplished through comparisons of solutions for gentle terrain slopes when using IBM and terrain-following grids. The canonical case of flow over a Witch of Agnesi hill provides validation of the basic no-slip and zero gradient boundary conditions. Specified diurnal heating in a valley, producing anabatic winds, is used to validate the use of flux (non-zero) boundary conditions.

This anabatic flow set-up is further coupled to atmospheric physics parameterizations, which calculate surface fluxes, demonstrating that the IBM can be coupled to various land-surface parameterizations in atmospheric models. Finally, results are presented for flow over a two-dimensional urban skyline.

The method described and validated in chapters 5 and 6 is extended to three dimensions in chapter 7. Results are presented for flow over a three-dimensional hill. While, the IB method used in chapters 5 and 6 works well for simple three-dimensional geometries, a more flexible and robust method is needed for extremely complex geometries, as found in three-dimensional urban environments. A second, more flexible, immersed boundary method is devised, and results are compared to the first IBM approach. Additionally, the functionality to nest a domain with resolved complex geometry inside of a parent domain without resolved complex geometry is described. The new IBM approach is used to model urban terrain from Oklahoma City in a one-way nested configuration where lateral boundary conditions are provided by the parent domain.

In chapter 8 the IB method is extended to include wall model parameterizations for rough surfaces. Two possible implementations are presented, one which uses the log law to reconstruct velocities exterior to the solid domain, and one which reconstructs shear stress at the immersed boundary, rather than velocity. These methods are tested on the three-dimensional canonical case of neutral atmospheric boundary layer flow over flat terrain.

Chapter 9 summarizes the work presented here, and includes a detailed discussion of future work needed to fully integrate IBM into numerical weather prediction models. Additionally, this chapter includes information about plans to further validate our method. Specifically, the results for flow over Oklahoma City presented in chapter 7 will be validated against observations. Field data exists for this urban environment from the Joint URBAN 2003 field campaign. The field data is extensive and includes measurements of velocity profiles, temperature, turbulent kinetic energy, and concentration from releases of a passive tracer. Using input data from the intensive observation periods in Oklahoma City, comparisons are planned between predicted and real measurements of velocity fields, temperature, turbulent kinetic energy, and scalar concentrations.

1.4 Summary of contributions

In summary the main contributions of this research are:

1. An investigation of error arising from the terrain-following coordinate transformation in numerical weather prediction models caused by the presence of steep topography.
2. Improvement or elimination of these errors using the immersed boundary method.
3. Two different original and unique three-dimensional ghost-cell immersed boundary

methods, which function in WRF with the native time dependent pressure-based coordinate system.

4. Formulations of each immersed boundary method for two-dimension terrain, reducing computational effort for idealized terrain.

5. A method for coupling the immersed boundary method to atmospheric physics parameterizations.

6. Validation of the immersed boundary method with Dirichlet and Neumann boundary conditions, as well as validation of the representation of unresolved physical processes.

7. Application of the method in real three-dimensional urban terrain in a nested configuration.

8. Representation of surface fluxes of momentum, including the original option of reconstructing shear stresses at the immersed surface.

Chapter 2

Background of LES boundary conditions and the immersed boundary method

Air is a viscous fluid, and therefore, must come to rest at the point where it contacts the earth's surface. This requirement is known as the no-slip boundary condition. The structure of the atmospheric boundary layer is then dominated by the viscous effects of friction at the surface. The computational cost of simulating fluid flows is dependent on the need to capture the effects of these near-wall turbulent structures on the outer flow, but resolving these structures is extremely challenging. This chapter details the structure of the turbulent boundary layer, and the computational challenges associated with resolving this region. Alternative methods for modeling the near-wall region are presented, which circumvent the need to directly resolve this layer. Additionally, the atmospheric boundary layer is complicated by the presence of complex terrain. The immersed boundary method is deployed as a method for dealing with the effects of terrain, and the background of IBM is included here. Finally these two techniques, wall modeling and the immersed boundary method, must be used in combination for large-eddy simulations of atmospheric flows. Attempts at combining these techniques are reviewed.

2.1 Large-eddy simulation of wall bounded turbulent flows

Most engineering and geophysical flows have Reynolds numbers on the order of 10^6 to 10^9, while direct numerical simulation is presently limited to Reynolds numbers on the order of 10^4. Therefore, there is a great disparity between the flows that we can directly calculate and the flows that we wish to calculate. In order to close this gap, several techniques have been invented which take advantage of the statistical nature of turbulence. These methods include Reynolds-averaged Navier-Stokes (RANS) and large-eddy simulation (LES). Results from RANS are often criticized as too inaccurate because the turbulence models require parameters which are not universal to all flows. Improved results can be achieved with LES, but at a very high (and often prohibitive) computational cost.

As well resolved LES is often too expensive, techniques have been used to reduce the cost. Zonal grids are one technique that is often used, in which the domain is decomposed into grids of various resolutions, with the finest grids placed near boundaries. Often researchers choose to resolve grids only in the wall normal direction. Fine resolution in only the wall normal direction ignores the need for increased tangential resolution near boundaries. Many researchers preform LES on grids that are coarse, letting the turbulence model account for a large portion of the Reynolds stresses in the near wall regions. While this is not the optimal way to use the large eddy simulation technique, it enables engineers and meteorologists to model flows that could not otherwise be modeled. The no-slip boundary condition cannot be used in the case of coarse LES because it would under predict the surface stresses. Therefore coarse LES almost always requires some sort of averaging treatment of the boundary layer in order to calculate the correct mean surface stresses. Such parameterization of the boundary layer is known as a wall model or approximate boundary condition.

2.1.1 Structure of the turbulent boundary layer

Much of what is known of turbulent boundary layers has been gained through computation or experimentation. Reynolds numbers for our application, the atmosphere, are much higher than what can currently be achieved through direct simulation or novel experimental design. In this section the structure of the turbulent boundary layer is described at lower Reynolds numbers than are found in the atmosphere, due to the availability of data and documentation. The extrapolation of low or moderate Reynolds number results to very high Reynolds numbers can be problematic, and is the subject of ongoing study. A series of field experiments on atmospheric boundary layers have been conducted at the Surface Layer Turbulence and Environmental Science test facility located in the Utah Great Salt Lake Desert, with the goal of understanding Reynolds number dependence of structure and statistics of the boundary layer [Metzger and Klewicki, 2001; Priyadarshana and Klewicki, 2004; Kunkel and Marusic, 2006; Morris et al., 2007; Klewicki et al., 2008]. In this series of studies it was found that the mean streamwise velocity profile in the surface layer, normalized by the inner viscous scales, exhibited Reynolds number similarity. Morris et al. [2007] show a near perfect fit of the neutral atmospheric surface layer mean velocity with the log law, while the location of peak turbulent momentum flux was shown to exhibit Reynolds number dependence. Visualization of the ABL by Hommema and Adrian [2003] indicates that the near-wall coherent structures (hairpin vortex packets) seen at lower Reynolds numbers are also present in the ABL with scale similarity. Thus, knowledge about turbulence boundary layers gained at lower Reynolds numbers is still useful at much higher Reynolds numbers.

In any turbulent flow there is a balance of production, dissipation, pressure work, and transport in the turbulent kinetic energy (TKE) budget. Away from walls or boundaries production generally balances dissipation. This assumption is so common that many turbulence models have been based on this idea of a local equilibrium. In contrast, direct numerical simulations of channel flow show that the TKE budget is more complicated in the near wall region. Distance from wall can be defined in term of viscous wall units $y^+ = u_* y/\nu$, where ν is the kinematic viscosity, $u_* = \sqrt{\tau_w/\rho}$ is the friction velocity, τ_w is the wall stress, and ρ is the density. Furthermore, it should be noted that y is the coordinate in the wall normal direction in this chapter, while in the other chapters, the z coordinate represents the wall normal direction. The largest production of TKE occurs at $y^+ \approx 12$, while the maximum dissipation occurs at the wall. The scaling for the location of peak production is relatively independent of the Reynolds number [Bernard and Wallace, 2002]. While this structure is accurate for geometrically simple cases, like channel flow or a zero pressure gradient boundary layer, it is not known how these results extend to more complex flows.

The near wall region can be divided and classified according to the importance of processes (terms in the TKE equation) for a given region, which are shown in figure 2.1. The region closest to the wall ($y^+ < 5$) is called the viscous sublayer. This is the region where molecular diffusion transports TKE toward the wall where it is dissipated. In the buffer region ($5 < y^+ < 30$) production peaks. Turbulent transport is important in this region, and much of the production of TKE in the buffer layer is transported towards the wall. The region outside of the buffer layer is known as the overlap, logarithmic, or intermediate layer. This region begins at $y^+ = 30$, and encompasses 10% to 30% of the boundary layer. In the intermediate region, the scaling laws used for both the inner and outer layer are valid. In the inner layer the scaling used is the law of the wall, and in the outer layer it is the velocity defect law. The logarithmic law that is valid in this overlap layer can be derived by matching the velocity gradients of these two scaling laws (or by dimensional considerations alone). Finally, the outer layer extends from the logarithmic layer to the full depth of the boundary layer.

One of the earliest insights into the complexity of the turbulent boundary layer came from flow visualization work performed by Kline et al. [1967], which showed the presence of coherent structures called low speed streaks. These structures were visualized by stretching a wire spanwise across the flow, and electrifying it to form hydrogen bubbles which advect with the flow. Coherent structures are flow elements or eddies which show a considerable degree of organization and repetitiveness [Bernard and Wallace, 2002]. Turbulent statistics

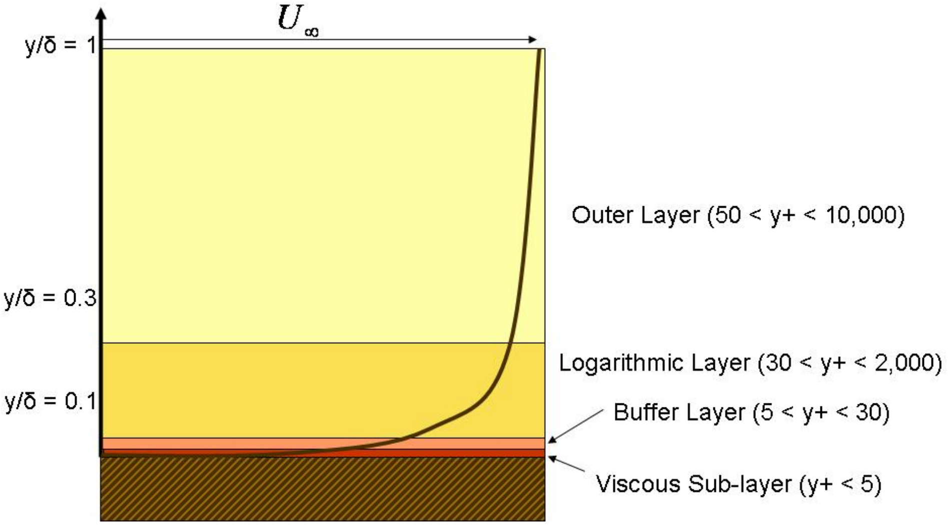

U_∞
y/δ = 1
y/δ = 0.3
y/δ = 0.1
Outer Layer (50 < y+ < 10,000)
Logarithmic Layer (30 < y+ < 2,000)
Buffer Layer (5 < y+ < 30)
Viscous Sub-layer (y+ < 5)

Figure 2.1. The turbulent boundary layer is divided into several regions based on the terms that dominate the TKE budget.

are highly correlated over the length scale of these structures. This correlation suggests that while turbulence was previously thought of as random and only suitable for statistical analysis, there may instead be order within the chaos.

Low speed streaks have been shown to have lengths of up to 1000 viscous wall units (y^+), and are spaced at 100 y^+ apart. These streaks are known to lift up into the buffer layer causing a localized shear layer in the velocity profile. This instability causes the streak to undergo oscillations. The oscillations eventually cause the structures to break up, which is called turbulent bursting, and a parcel of slow moving fluid is ejected into the faster moving fluid above it. Out of phase with ejections, parcels of high speed fluid are observed to rush into the region of low speed fluid. These events are called sweeps. Furthermore, it has been discovered that nearly all production of turbulent kinetic energy in the near wall region occurs during these bursting and sweeping events [Bernard and Wallace, 2002]. It should be noted that a burst is consistent with negative Reynolds stress ($\overline{u'v'} < 0$), because low speed fluid ($u' < 0$) is moving away from the wall ($v' > 0$). Likewise a sweep, which is high speed fluid approaching the wall, causes negative Reynolds stresses. Here, a prime denotes a fluctuation from the mean, and an overbar is an averaging operator.

For perspective, in neutral atmospheric flows, the typical boundary layer depth is 1000 m, the log layer extends for 100 m, and the viscous sublayer is 0.5 mm [Garratt, 1992]. A typical viscous length scale y^+ is on the order of 0.1 mm. With the logarithmic layer spanning a 100 m depth, it would extend to 2500 y^+ units at a Reynolds number of 10^6, and to more than a million y^+ at higher Reynolds numbers. This non-dimensional distance is greater than indicated in figure 2.1, which is typical of lower Reynolds numbers. The nominal streak spacing of 100 y^+ has been shown to hold for the smallest structures, even for atmospheric boundary layers at very high Reynolds numbers [Klewicki et al., 1995]. A constant streak

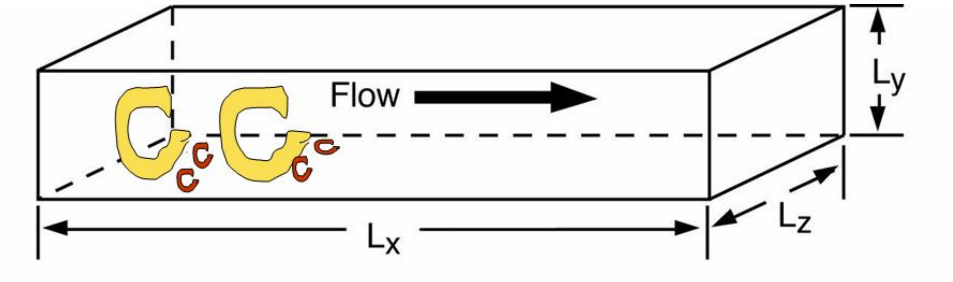


Figure 2.2. This diagram shows channel flow with streamwise follow in the x direction. The flow is bounded in the vertical extent. These boundaries cause the largest eddies (yellow) to scale with the physical dimensions of the domain, while the smallest eddies (red) occur at the viscous scale.

spacing indicates that these features become increasingly small in comparison to large scale motions in the logarithmic layer at high Reynolds numbers. This is supported by the model of Landahl [1990], who hypothesizes that streak dynamics are initiated within the viscous sublayer (as opposed to the log layer). Hommema and Adrian [2003] document near-wall coherent structures in the lowest 3 m of the atmosphere which lift with an angle of 3° to 35° with respect to the wall; consistent with the lift angle found at much lower Reynolds numbers.

In summary, the bottom 1% of the boundary layer is comprised of extremely complex dynamical processes. Coherent structures are evidence of this, and the mechanisms for producing turbulence in laboratory scale flows remains intact for atmospheric flows. Turbulent flow cannot be modeled without accounting for the effects of these structures and processes on the mean flow. These structures exist on the viscous scale, making them difficult to observe experimentally or resolve numerically. Turbulent flow in complex geometries is further complicated by the presence of separated flow. Coherent structures that are present in attached flows do not exist in regions of separation, which behave essentially like low Reynolds number flows Cabot and Moin [1999].

2.1.2 Resolution requirements for DNS and LES

Turbulent channel flow is an often used example of a flow that has a wide range of length scales present. Figure 2.2 depicts a channel that has walls at the top and the bottom of the domain, but is considered infinite in the horizontal extents. The presence of the wall has a profound effect on the turbulent flow. The length scale of the largest eddies in the flow is bounded by the physical presence of the boundaries. The smallest length scale in the flow is based on viscosity. In numerical simulations, the domain must be large enough to contain several vortical structures, while the grid spacing must be small enough to resolve individual structures.

Shih [2003] completed LES and DNS runs of channel flow and found that vortices have a streamwise length of 200 y^+, spanwise spacing of 100 y^+, and a vertical height of 50 y^+. Although the streamwise length of an individual vortical structure was found to be 200 y^+, it was also noted that the streamwise streaks created by their downstream advection may exceed 1000 y^+. Therefore, 1100 y^+ is commonly used as a streamwise length scale. Shih found that an overall domain size of 4000 y^+ in the streamwise direction, and 1000 y^+ in the spanwise direction was sufficient. This total domain size would include at least 4 streamwise structures, and 5 vortex pairs in the spanwise direction. Interestingly, it was found that the coherent structures near the boundary are so essential for maintaining turbulence that if the domain is too narrow the flow collapses to a laminar state.

Piomelli and Balaras [2002] state that to accurately represent structures in a direct numerical simulation of the near wall region, the first off wall grid point should be located at $y^+ < 1$. Resolution in the horizontal dimensions must be $\Delta x^+ \approx 15$ and $\Delta z^+ \approx 5$. This means that an absolute minimum of 20 grid points is needed in each direction to resolve a structure. Furthermore, as the Reynolds number approaches infinity, an increasing number of grid points must be used to resolve a layer of decreasing thickness.

Resolution requirements are slightly more relaxed in large-eddy simulations where the larger eddies are resolved, and the smaller scales are modeled. Piomelli and Balaras suggest that for LES, a horizontal grid spacing of $\Delta x^+ \approx 50 - 150$ and $\Delta z^+ \approx 15 - 40$ can be used. At this resolution, there would be 3 to 20 points needed in each dimension to resolve a coherent structure. LES work presented in Shih [2003] used a grid spacing of $\Delta y^+ \approx 3$ in the vertical, and $\Delta x^+ \approx 40$ and $\Delta z^+ \approx 4$ in the horizontal. This created a well resolved LES with about 20 grid points resolving a streak in each direction. The total domain size of this large-eddy simulation was 128 x 256 x 129 grid points, for a total of 4.2 million grid points. The highest Reynolds number used was $Re_\tau = 720$.

If the no-slip boundary condition is used at the wall, then the convective term $u_i u_j$ is known, and differentiation of the velocity profile is used to determine the viscous stresses. This is only accurate when the wall layer is well resolved. When the grid is not fine enough to resolve the near wall gradients, then the wall layer must be modeled by specifying a correlation between the velocity of the outer flow, and the stress at the wall. This can decrease the resolution needed to $\Delta y^+ \approx 30 - 150$, $\Delta x^+ \approx 100 - 600$, and $\Delta z^+ \approx 100 - 300$, or even more.

2.1.3 Classes of wall models

Deardorff [1970] performed the first coarse LES of turbulent channel flow with approximate boundary conditions. Deardorff worked at the National Center for Atmospheric Science, and therefore was motivated to model high Reynolds number flows. Computational resources limited him to 6720 grid points, and due to this coarse resolution his first grid point was well outside the viscous sublayer. Realizing that the no-slip boundary condition at the wall was inappropriate, Deardorff implemented boundary conditions based on the law of the wall.

Even with the increases in computing power that have occurred since 1970, resolution of the near wall region is still the largest obstacle to performing well-resolved simulations of turbulent flows. Since this time, many researchers have proposed wall models for use in coarse LES. Nearly all of these models work by estimating the stress at the wall, and supplying this stress to the outer flow as a boundary condition. Current wall models generally fall into three classes. The first of these is equilibrium stress models. In this type of model a logarithmic profile is imposed on the mean or instantaneous horizontal velocities. The second class of models are zonal approaches, where different equations are applied to the outer and inner flows. Finally, the third class of methods attempts to incorporate our knowledge of coherent structures by using a linear stochastic correlation to supply shear stress at the wall based on off wall velocity statistics. Each of these methods has had limited success. Some researchers such as Baggett [1997] question whether it is even sufficient to supply the shear stress to the outer flow, or if velocities are needed instead.

Equilibrium models

The simplest way to relate stresses at the wall to velocities in the outer flow is to neglect all of the terms in the streamwise momentum equation except for the gradient of the Reynolds stress [Piomelli and Balaras, 2002]. It is assumed that the first grid point is far enough from the wall that viscous effects are negligible, and additionally that the acceleration and pressure gradient terms can be neglected at the first off wall point. If the first off wall grid point is located in the constant flux region then the velocity profile will follow the log law. In terms of viscous wall units, this is given by equation 2.1a for an aerodynamically smooth wall or 2.1b for a rough wall, where κ is the von Kármań constant and y_o is the roughness length scale.

$$u^+ = \frac{u}{u_*} = \frac{1}{\kappa} \ln y^+ + C \tag{2.1a}$$

$$u^+ = \frac{1}{\kappa} \ln \frac{y}{y_o} \tag{2.1b}$$

Deardorff [1970] first modeled turbulent channel flow neglecting molecular viscosity, and using a sub-grid scale (SGS) model to evaluate the turbulent stresses. Deardorff used the Smagorinsky model which subsequently became well-known and often used Smagorinsky [1963]; Smagorinsky et al. [1965]. Recognizing the need for approximate boundary conditions, and noting that the first off wall grid point was located in the logarithmic region, Deardorff set the second derivative of the velocities at the first off wall grid point to be:

$$\frac{\partial^2 \bar{u}}{\partial y^2} = -\frac{1}{\kappa(\Delta y)^2} + \frac{\partial^2 \bar{u}}{\partial z^2} \tag{2.2a}$$

$$\frac{\partial^2 \bar{w}}{\partial y^2} = \frac{\partial^2 \bar{w}}{\partial x^2}. \tag{2.2b}$$

Additionally, there must be no wall normal flow so that $\bar{v} = 0$ on the wall. Here overbars denote filtered quantities, and velocities and coordinates have been non-dimensionalized by u_* and the channel height, respectively. The above boundary conditions require that the horizontally plane averaged velocities satisfy the log law at Δy, and that the turbulent fluctuations be isotropic. Deardorff preferred equation 2.2 because it is independent of the roughness length scale. His results did not match experimental results, however, it is believed that the primary sources of error were the coarse resolution and SGS model, not the approximate boundary conditions. Thus, this work inspired many subsequent researchers to pursue the development of approximate boundary conditions or wall models.

Schumann [1975] used an alternative method of relating shear stress at the wall to the velocities in the outer flow. Schumann calculated the plane averaged streamwise velocity at the first off wall grid point to be:

$$\langle \bar{u}(x, \Delta y, z) \rangle = \frac{\langle \tau_w \rangle^{1/2}}{\kappa} [\ln(\Delta y E) - 1], \tag{2.3}$$

where E is a constant that accounts for a roughness length scale, the over bars denote filtered quantities, and the angled brackets represent horizontal plane averaging. The shear stress at the wall is then calculated iteratively with the velocity according to the relationship:

$$\tau_{xy}(x, z) = \frac{\langle \tau_w \rangle}{\langle \bar{u}(x, \Delta y, z) \rangle} \bar{u}(x, \Delta y, z) \tag{2.4a}$$

$$\tau_{yz}(x, z) = \frac{1}{Re_\tau} \frac{\bar{u}(x, \Delta y, z)}{\Delta y}. \tag{2.4b}$$

This model assumes that deviations from the mean velocity profile correlate linearly with deviations from the mean wall stress. Schumann's simulations had good agreement with experimental results. With this success, several subsequent researchers have used similar iterative methods to calculate mean wall stresses.

Alternatively, the log law can be enforced instantaneously by applying equation 2.5 at the first off wall grid point, where $\bar{u}$ is the filtered non-dimensional velocity, essentially the derivative of equation 2.1. This assumes local equilibrium in the first grid cell.

$$\frac{\partial \bar{u}}{\partial y} = \frac{\bar{u}}{\Delta y \ln(\Delta y / y_o)} \tag{2.5}$$

Deardorff [1970], Cabot and Moin [1999], and other researchers have used this approach, and found the results just as accurate as when the log law is applied in an averaged sense.

The power law can be used in the same manner. These wall models (and others of the equilibrium type) work best when there is very coarse resolution or very high Reynolds numbers. In this case, the first computational cell is large in comparison to the eddies, so that many eddies are contained within the cell. The time step is also large compared to the lifespan of the viscous eddies. This makes an averaging approach to modeling the bulk effects of the boundary layer more appropriate than when the near wall structures are partially resolved. Piomelli and Balaras [2002] found success with grid spacing of 1500 viscous units in the streamwise direction, and 700 viscous units in the spanwise direction. It is noted that in this case, the root-mean-square of the difference between the log law and the instantaneous velocity profile is less than 10%.

Zonal models

Zonal models are based on the idea that different governing equations should be used for the inner and outer layers. In an approach proposed in Balaras et al. [1996] the filtered Navier-Stokes equations are solved for the outer flow, while the boundary layer equations are solved for the inner flow. This is referred to as the two-layer model (TLM). In the two-layer model, the boundary layer equations are used to solve for the flow from the wall to the first off wall grid point. The equation is as follows, where $i = 1, 3$:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial (\bar{u}_2 \bar{u}_i)}{\partial x_i} = -\frac{\partial \bar{P}}{\partial x_i} + \frac{\partial}{\partial x_2}[(\nu + \nu_t)\frac{\partial \bar{u}_i}{\partial x_2}]. \tag{2.6}$$

For the boundary layer equation, the boundary conditions used are no-slip at the wall, and the velocity from the first grid point, called the interface velocity. The shear stress is then calculated at the wall, and this is used as the boundary condition for the outer flow. Results from the TLM are in good agreement with results from DNS and resolved LES. The drawback of this method is that the cost is higher than implementing an equilibrium model. Additionally, Cabot and Moin [1999] compared the TLM with the instantaneous log law wall model and found little difference.

Both equilibrium models and the TLM have been used on separated flows, but the validity of these wall models is rightly in question for this use. Equilibrium models are based on a local equilibrium of production and dissipation, and therefore do not take the localized pressure gradients causing the separation into account. The boundary layer equation solved in the TLM is valid only for attached flows, and assumes a weak interaction between the inner and outer layers which breaks down in separated flows. Detached Eddy Simulation (DES) is another type of zonal model, which was developed to address separated flows [Spalart et al., 1997; Spalart, 2009]. DES uses a single grid, but uses two different turbulence models by combining RANS and LES. The RANS equations are used in the attached boundary layer regions, while LES is used in the separated regions. This approach has had mixed results which seem to be dependent on geometry and Reynolds number. It works well, if it is known in advance where the RANS assumption can be successfully used.

Other types of wall models

Several researchers have used DNS data as off wall boundary conditions for coarse LES with success. The drawback of this method is obvious, it requires that DNS data exist for the specific flow geometry. Baggett [1997] tests the application of velocity boundary conditions, rather than a wall stress, by providing a time series of velocity data from a coarse DNS to the plane consisting of the first off wall grid points in a coarse LES. Perfect agreement was achieved between the DNS, and the DNS-driven LES.

The most recent wall models that have had some success are based on control theory. Control theory is used to force the velocity profile to take a logarithmic shape. Piomelli and Balaras [2002] notes that these types of wall models yield improved agreement over simple models, but at excessive cost (20 times the cost of the standard LES).

2.1.4 Wall models for atmospheric applications

Cabot and Moin [1999] conclude at the end of their review of approximate boundary conditions that '... with regard to numerical resolution, it appears in many cases that the old adage, "you get what you pay for", still applies.' However, keeping in mind that resolution of the near-wall region is not an option for atmospheric flows, the use of a wall models is a necessity. Equilibrium models, and in particular the instantaneous log law, are the most common wall models used in meteorology (including the WRF model).

For meteorological applications the logarithmic law is often modified to account for stratification. Modifications generally consist of corrections for the wall heat flux or temperature gradient. For example, Moeng [1984] used equation 2.7 as a wall model in LES simulations of the ABL, where Φ_M is the Monin-Obukhov stability function and L is the Obukhov length scale.

$$\frac{\partial \bar{u}}{\partial y} = \frac{u_*}{\kappa y}\Phi_M \tag{2.7a}$$

$$\Phi_M = 1 + \beta \frac{y}{L} \tag{2.7b}$$

The parameter β is empirically determined, based on stability regimes. For a neutral atmosphere, the function $\Phi_M(0) = 1$, and the log law is not modified. For a stable atmosphere $\Phi_M > 1$, and for unstable conditions $\Phi_M < 1$.

Cabot and Moin [1999] demonstrated that simple equilibrium models work fairly well, especially when their low cost is taken into consideration. This is especially true for high Reynolds number flows, such as atmospheric flows, where several eddies are contained within the grid cell nearest to the wall, making an averaging approach especially appropriate. The use of equilibrium models is questionable with complex geometries (i.e. with regions of flow separation). In these more complex flows, better results are often achieved with a more complex wall model, although these models become increasingly difficult to implement with complex terrain. For example, methods that enforce the log law in a mean sense by including horizontal plane averaging, such as in equations 2.2 and 2.3, are particularly difficult to apply to flows over complex terrain.

Currently, WRF has five implementations of surface layer schemes. Each is based on the instantaneous log law, but differ in the treatment of momentum, heat, and moisture fluxes at the surface. For instance, one scheme might use the same roughness length scale for both momentum and heat, while another scheme allows these two length scales to differ. Generally, each surface layer scheme is intended for use with a specific PBL scheme that characterizes vertical mixing. When a LES closure is used instead of a PBL scheme, WRF recommends using the MM5 surface layer scheme. For our purposes, we have implemented an instantaneous log law that does not include a Monin-Obukhov stability function. A description is provided in section 3.4. In section 2.2.3 the use of wall models with the immersed boundary method is discussed. In chapter 8 our basic wall model is used to simulate neutral boundary layer flow with terrain-following coordinates, and the results are compared to results from simulations using an IBM in combination with a wall model.

2.2 Background of the immersed boundary method

The numerical simulation of atmospheric flow over complex geometries, especially those geometries found in mountainous and urban environments, requires effective and efficient computational techniques. Terrain following grids have found common use in atmospheric codes as a method for dealing with complex geometries. Although this method is effective, it cannot be successfully used to represent geometries with extreme slopes, such as buildings. An alternative in this case is the immersed boundary method (IBM). When using IBM, numerics are solved on a Cartesian grid, and boundaries that are immersed within the grid are represented with the addition of a forcing term in the Navier-Stokes equations (see figure 2.3). This section gives a brief history of the development of IBM.

The immersed boundary method was first proposed by Peskin [1972] and [1977], who developed the method to simulate blood flow through the mitral valve of the heart. In the original formulation of IBM, the incompressible Navier-Stokes equations as given by (2.8) are solved on a two-dimensional Cartesian grid. The effects of the external forcing of the fluid by the boundaries are represented by the addition of a body force term $\boldsymbol{F}$ in the equation for the conservation of momentum.

$$\rho(\boldsymbol{u}_t + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \boldsymbol{F} \tag{2.8a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.8b}$$

The forcing term takes a non-zero value at grid points that are in the vicinity of the boundary, but has no effect on the computation of grid points away from the boundary. The magnitude of the force term is calculated by modeling the boundary (the elastic heart walls) as a set of interconnected springs. It is then assumed that the solid boundary has no mass so that these

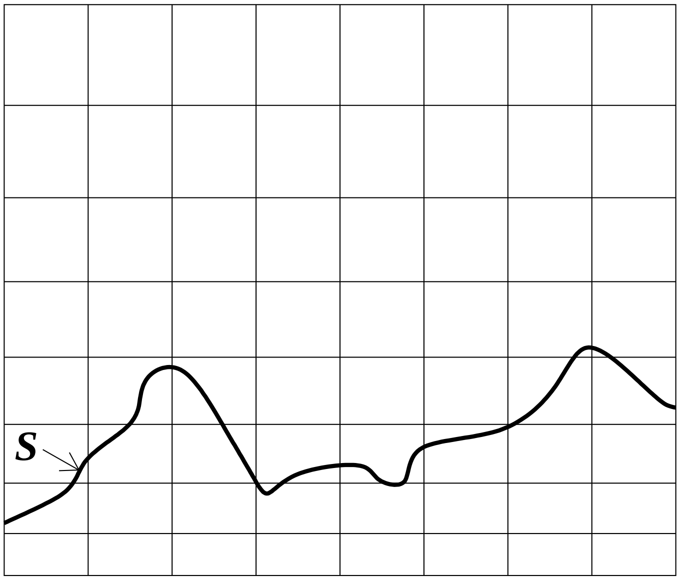


Figure 2.3. Cartoon sketch of an immersed boundary.

forces are transferred to the fluid flow. The main difficulty with this method, as noted by Peskin, is the determination of the forcing field because it requires modeling of the stresses and strains internal to the solid boundary. In addition, the computational cost of resolving flow in the region of the immersed boundary is high, and limited Peskin to low Reynolds number flows.

Peskin's application is extremely complex because the heart valve and walls are elastic and move with the fluid flow. Beginning in the late 1980's several researchers studied the use of the immersed boundary method to represent rigid boundaries. The rigid boundary case is much simpler because the location of the boundary is known. In theory, Peskin's method could be used to represent rigid walls in the limit of zero elastic deformation. In practice, this can lead to a numerically stiff problem that requires small time steps [Iaccarino and Verzicco, 2003].

2.2.1 Formulation of the forcing term

Feedback forcing methods

IBM was first employed for rigid boundary applications by Briscolini and Santangelo [1989] and Goldstein et al. [1993]. Briscolini and Santangelo modeled two-dimensional flow in a square cavity with one moving wall and flow around circular and square cylinders. Goldstein et al. modeled two-dimensional flow around circular cylinders and three-dimensional channel flow with smooth and ribbed surfaces. In both studies a spectral method was used and spurious oscillations appeared at the boundary unless the forcing was smoothed over several grid points, diffusing the location of the boundary and reducing the accuracy. Saiki and Biringen [1996] eliminated the spurious oscillations by using feedback forcing in a finite difference scheme to simulate flow over stationary, rotating, and oscillating cylinders. The feedback forcing term as appears in Saiki and Biringen [1996] is given by equation (2.9). In this equation $\mathbf{x_s}$ represents the location of nodes coincident with the boundary, and $\mathbf{U}$ is the fluid velocity at those points. The desired surface velocity used to set a Dirichlet boundary condition is $\mathbf{v}$.

$$\mathbf{F}(\mathbf{x_s}, t) = \alpha \int_0^t (\mathbf{U}(\mathbf{x_s}, t) - \mathbf{v}(\mathbf{x_s}, t))dt + \beta(\mathbf{U}(\mathbf{x_s}, t) - \mathbf{v}(\mathbf{x_s}, t)) \tag{2.9}$$

If the forcing term is balanced by the unsteady term in the Navier-Stokes equation, as shown in (2.10), it can more clearly be seen that the forcing term acts as a damped oscillator where α acts as the spring constant and β is the damping coefficient. In this case, $\mathbf{u} = \mathbf{U} - \mathbf{v}$, and the forcing term works to set the fluid velocity equal to the Dirichlet boundary value so that $\mathbf{u}$ tends to zero.

$$\frac{\partial \mathbf{u}}{\partial t} \approx \alpha \int_0^t \mathbf{u} dt + \beta \mathbf{u} \tag{2.10}$$

The disadvantage of feedback forcing is that the formulation contains parameters α and β which the user must tune empirically according to the fluid flow. Additionally, highly unsteady flows cause these coefficients to become large, so that the problem is numerically stiff. Goldstein et al. [1993] found that the time step must decrease by one to two orders of magnitude to ensure stability.

Direct forcing methods

An alternative forcing formulation developed by Mohd-Yusof [1997] does not affect numerical stability or require a smoothing function. In the IB approach used by Peskin [1972], Briscolini and Santangelo [1989], and Goldstein et al. [1993] the forcing function represents the action of the boundary on the flow, while the forcing in Mohd-Yusof [1997] is equivalent to setting a velocity boundary condition, and often referred to as direct forcing. The forcing term can be represented by (2.11), but need not be calculated in the IB implementation. Instead, the desired Dirichlet boundary value $\mathbf{v}$ is directly imposed on the boundary. Mohd-Yusof used direct forcing in a spectral method scheme to simulate laminar flow over a ribbed channel.

$$\mathbf{F} = \frac{\mathbf{v} - \mathbf{U^n}}{\Delta t} + \mathbf{U} \cdot \nabla \mathbf{U} + \frac{1}{\rho} \nabla p - \nu \nabla^2 \mathbf{U} \tag{2.11}$$

Fadlun et al. [2000] used both feedback forcing and direct forcing in a finite difference method to represent three-dimensional complex flows including formation of a vortex ring from a nozzle, flow around a sphere, and the flow in the cylinder of an internal combustion

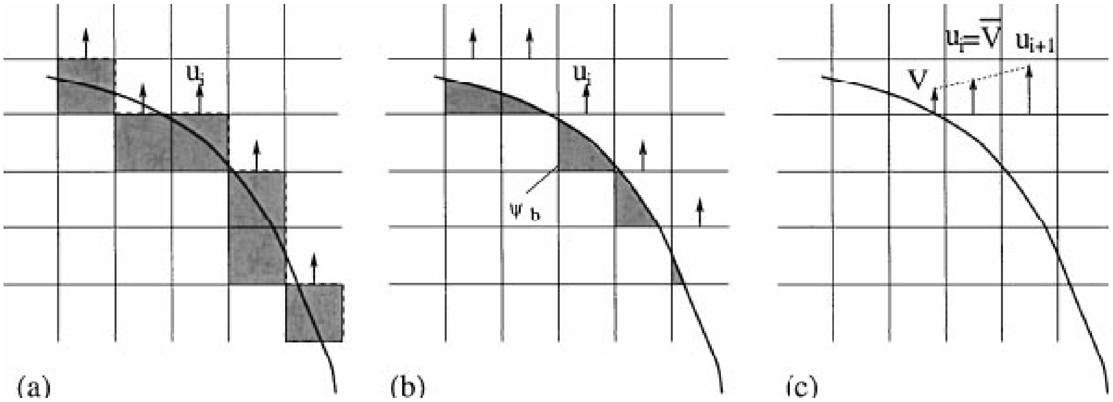


Figure 2.4. Figure from Fadlun et al. [2000] depicting velocity reconstruction interpolation methods: (a) stepwise geometry, (b) volume fraction weighting, (c) linear velocity interpolation. Reprinted from Fadlun et al. [2000], with permission from Elsevier.

(IC) engine. The IC piston simulation is at a high enough Reynolds number to require the use of a subgrid-scale turbulence model, and is the first instance of IBM being used with LES. Fadlun et al. [2000] found that the two forcing methods produced similar results, however, the direct forcing method is advantageous because there are no flow dependent parameters and the additional forcing term does not limit stability. Also, unlike with the feedback forcing term which oscillates, the boundary condition can be satisfied exactly at each time step with direct forcing. For these reasons, the IB methods presented in this dissertation use direct forcing. Additionally, the remainder of this discussion focuses on direct forcing IBM implementations.

2.2.2 Interpolation methods for boundary reconstruction

The forcing formulations found in section 2.2.1 are derived in the case that the grid locations coincide with the boundary. The discrete grid points are not generally coincident with the boundary nodes $\mathbf{x_s}$, and in particular this alignment is impossible on a staggered grid. Therefore, an interpolation method must be used to determine the forcing needed at actual computational nodes.

Fadlun et al. [2000] present results for three methods: stepwise geometry, volume fraction weighting, and linear velocity interpolation, as illustrated in Figure 2.4. In the stepwise geometry method, forcing is imposed at the closest grid point to the boundary, regardless of the node being interior or exterior of the boundary. No interpolation is needed for this procedure. In volume fraction weighting the force is scaled by ψ_b/ψ, the ratio of the volume of the cell occupied by the boundary ψ_b to the total cell volume ψ. The forcing is then applied to the closest fluid node to the boundary. In the velocity interpolation method, a velocity is imposed for the first fluid point using linear interpolation between the bound-

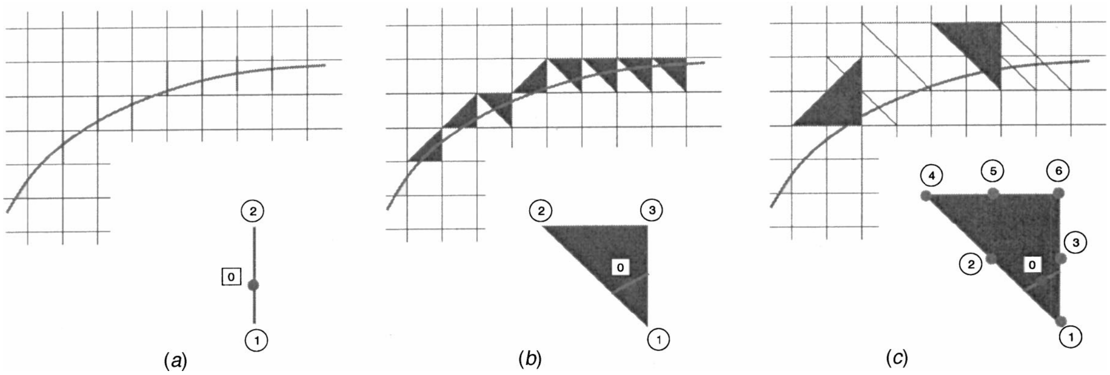


Figure 2.5. Figure from Iaccarino and Verzicco [2003] proposing several ghost point extrapolation methods: (a) linear one-dimensional, (b) linear multi-dimensional, (c) quadratic multi-dimensional. Reprinted from Iaccarino and Verzicco [2003], with permission from the American Society of Mechanical Engineers.

ary condition and the velocity at the second fluid point. Methods that satisfy the boundary conditions by setting the velocity at fluid nodes are often called velocity reconstruction methods, because the desired velocity field is being reconstructed in the vicinity of the boundary. Using Richardson extrapolation, Fadlun et al. determined that the stepwise geometry reconstruction converges slower than first order. The error decreases slightly better than first order for volume fraction weighting, and slightly worse than second order for linear velocity interpolation.

Iaccarino and Verzicco [2003] present three additional interpolation schemes: linear one-dimensional, linear multi-dimensional, and quadratic multi-dimensional. These are illustrated in Figure 2.5. All of these schemes belong to a class of IB approaches known as ghost cell methods, where the velocity is set on a node that is inside of the boundary. In all of the cases, the velocity at the node labeled (1) is found by extrapolation using the boundary value (0) and the fluid velocity at nodes that are marked as (2) or higher. In general this means that the fluid velocity at node (1) inside of the boundary is in the reverse direction of the exterior flow in order to enforce a no-slip boundary condition on the surface. The main difference between the three schemes is the increasing stencil size. Iaccarino and Verzicco [2003] found that one-dimensional interpolation was accurate for boundary geometries that were largely aligned with grid lines, however, for curvilinear geometries the multi-dimensional methods showed improved performance. Using the ghost cell approach, Iaccarino and Verzicco [2003] modeled flow around a cylinder, flow in a wavy channel, flow in a piston/cylinder assembly, and flow in a stirred tank.

Tseng and Ferziger [2003] also use the ghost cell approach to model flow past a cylinder, flow in a wavy channel, and geophysical flow over a three-dimensional Gaussian bump. In the ghost cell method, a common problem is large negative variable values when the immersed boundary is very near the fluid nodes in the interpolation stencil, and far from the ghost point. Tseng and Ferziger [2003] present two methods (illustrated in Figure 2.6) for dealing with this difficulty. In the image method, the interpolation scheme is used to solve for the

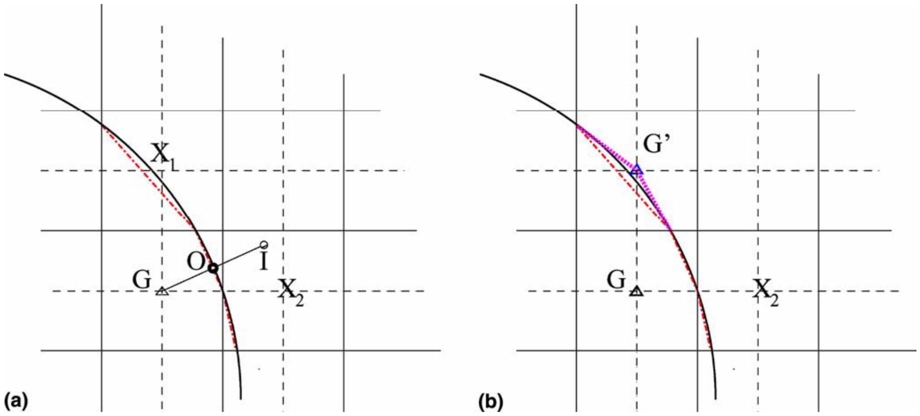


Figure 2.6. Figure from Tseng and Ferziger [2003] proposing two treatments to minimize numerical instability: (a) image method, (b) piecewise approximation. Reprinted from Tseng and Ferziger [2003], with permission from Elsevier.

fluid properties at the image point (I), which is the reflection of the ghost point across the boundary in the surface normal direction. Linear interpolation is then used to assign a value to the ghost point using $\phi_G = 2\phi_O - \phi_I$. In the piecewise linear approximation, the boundary is simply moved to the fluid node. The fluid node coinciding with the boundary is now the ghost node In this case the boundary condition is assigned to the ghost node, and no interpolation is necessary.

Balaras [2004] develop a multi-dimensional velocity reconstruction method, applying forcing at the first fluid point. This method is appropriate for two-dimensional convex geometries, and is extended to three dimensions in Gilmanov et al. [2003]. In this method, the solid geometry is described using an unstructured triangular mesh, as shown in figure 2.7. Nodes defining the cut-cell are numbered, with nodes 2, 3, and 6 residing in the solid domain. A surface normal vector is projected from the first fluid node, which in this case is the node labeled both 8 and b, to the immersed surface. Additionally, the location of a 'virtual' second fluid node is found by extending the surface normal until it intersects a computational cell face where all nodes making up the face are fluid nodes. The virtual second fluid node is marked c in figure 2.7, where it can be seen that the concept of a virtual point is similar to that of an image point. Linear interpolation between nodes α, β, δ, and γ is used to find the value at point c, and an additional linear interpolation is used between points a and c to reconstruct the velocity at the first computational node b. Gilmanov and Sotiropoulos [2005] extend this method to moving boundaries, and also utilize quadratic interpolation in the surface normal direction.

While the simulations in Gilmanov et al. [2003] are laminar, Balaras [2004] simulate turbulent flows using a LES closure with a test filter width of 2Δ to determine turbulent eddy viscosities. Near the immersed boundary the stencil of this filter requires use of solid nodes, therefore the turbulent eddy viscosity at the first fluid node is reconstructed rather than calculated with the subgrid stress model. In this case, the eddy viscosity is reconstructed in the same manner as used for velocity, and a linear relationship is assumed between the

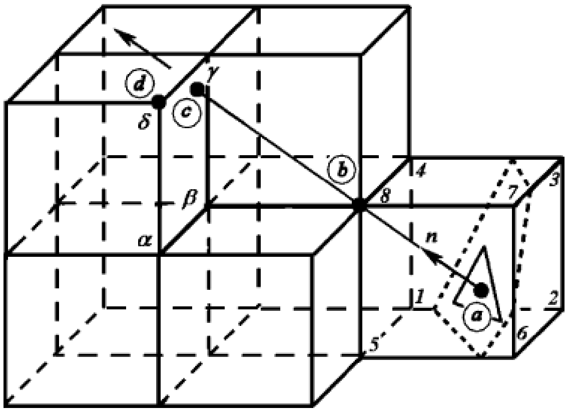


Figure 2.7. Figure from Gilmanov et al. [2003] proposing a multi-dimensional velocity reconstruction method for the first fluid node, to be used with three-dimensional geometries. Reprinted from Gilmanov et al. [2003], with permission from Elsevier.

eddy viscosity at the second virtual node, and the immersed boundary where it is set to zero. Balaras [2004] note that the use of a linear eddy viscosity reconstruction is most likely only acceptable because very fine grids are used near the immersed boundary.

Peller et al. [2006] reconstruct the velocity in the fluid domain using higher-order Lagrange and least squares interpolations. In this method, a higher-order polynomial based on either of the two interpolation methods is used along the surface normal, instead of a linear relationship. Several fluid points are required to calculate the coefficients of the polynomial. With curvilinear boundaries in two or three dimensions, polynomials are developed independently for each direction on the Cartesian grid, using the locations of computational nodes. The coefficients for the Cartesian polynomials are then geometrically weighting for an interpolation in the surface normal direction. This method eliminates the use of virtual fluid points and the associated interpolations, however, a large stencil of fluid nodes is required in each dimension for this method. This requirement limits the complexity of the geometry which can be handled with this method.

Gao et al. [2007] use inverse distance weighted interpolation with a ghost cell IBM to simulate flow past cylinders and spheres. In this method, a Taylor series expansion about the boundary point, given by equation 2.12 for two dimensions, relates the ghost node value φ_G to the boundary value φ_B.

$$\varphi_G = \varphi_B + \frac{\partial \varphi_B}{\partial x}\Delta x + \frac{\partial \varphi_B}{\partial y}\Delta y + \frac{1}{2}\left(\frac{\partial^2 \varphi_B}{\partial x^2}\Delta x^2 + 2\frac{\partial^2 \varphi_B}{\partial x \partial y}\Delta x \Delta y + \frac{\partial^2 \varphi_B}{\partial y^2}\Delta y^2\right) + O(\Delta x^3, \Delta y^3) \quad (2.12)$$

The value φ_B is known from the boundary condition, and the inverse distance weighted interpolation method is used to evaluate each derivative in the Taylor series on the boundary. Further details of this method are included in chapter 7. Additionally, an IBM method using inverse distance weighted interpolation is presented in chapter 7, however, our method uses

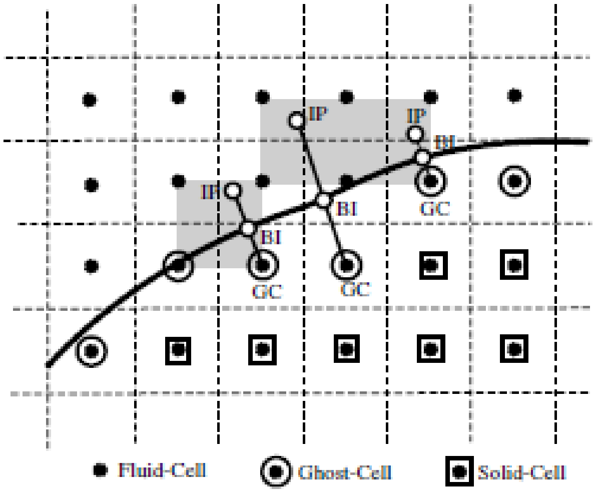


Figure 2.8. Figure from Mittal et al. [2008] proposing a ghost cell method using bilinear interpolation. Reprinted from Mittal et al. [2008], with permission from Elsevier.

the interpolation to find the velocity at an image point, rather than reconstructing derivatives on the immersed boundary.

A ghost cell IBM is combined with bilinear interpolation in Ghias et al. [2007] for compressible flow solvers using either Cartesian or curvilinear grids. The method is used to simulate flow around two-dimensional cylinder and airfoil geometries. The method is extended to incompressible flows around moving three-dimensional geometries in Mittal et al. [2008], and used to simulate flow over a fish and dragonfly, as well as canonical geometries. The interpolation method used in Mittal et al. [2008], shown in figure 2.8, makes use of an image point reflected from the ghost point in the surface normal direction. The neighbors chosen for the bilinear or trilinear interpolation are the nearest computational nodes, and for some cases the boundary point. If the neighbors for the interpolation are also ghost points, as in the left most case depicted in the figure, an iterative procedure is used to solve for the weighting coefficients and ghost point values.

The interpolation method used to represent a rigid boundary is obviously an integral part of the overall immersed boundary method, as demonstrated by the many options presented above. Interpolations for boundary reconstruction have been as much the focus of active research as the formulation of the forcing term. The above discussion on interpolation methods is by no means exhaustive, however, the ideas presented are the basis for the majority of the IBM literature. In our experience, the interpolations methods presented in this section did not work for our model and/or application, and new interpolation methods were required. The development and validation of the new IBM methods for use in WRF and with highly complex geometries, such as urban terrain, are presented in chapters 5, 6, 7, and 8.

2.2.3 Combining the immersed boundary method with wall models

When the immersed boundary method is used, it is nearly impossible to control the resolution of the near-wall region. As the boundary passes through the grid in a arbitrary manner, there may or may not be grid points within the viscous sub-layer or buffer layer. Furthermore, the grid cannot be refined in the wall normal direction, as it is not generally aligned with the immersed boundaries. A no-slip boundary condition is the most commonly used boundary condition at an immersed surface, effectively limiting its use to low Reynolds number flows. To extend use of the immersed boundary method to high Reynolds number flows, the method must be combined with a wall model. Previous attempts at combining wall models with the immersed boundary method are presented in this section.

An early attempt at combining wall models with IBM in an LES is presented in Tessicini et al. [2002], where a two-layer wall model is used to solve a simplified version of the boundary layer equations near the immersed boundary. This method was used to model flow over the trailing edge of an airfoil. In the method, flow is reconstructed at the first fluid point by solving a version of the boundary layer equations in which all terms are neglected, except for the viscous terms. By neglecting the unsteady, advective, and pressure gradient terms, this model reverts to an equilibrium stress model. The turbulent eddy viscosity is obtained through a damped mixing length model that is a function of y^+. An iterative procedure is used to solve for y^+, as it is dependent on the wall shear stress. A virtual second fluid point (as in Balaras [2004], which is not coincident with a computational node) provides boundary conditions to the boundary layer equations, as does the no-slip boundary condition at the surface. The virtual fluid point is defined by drawing a wall normal vector that intersects the first fluid point, and doubling the distance between the immersed boundary and the first fluid point. The velocity, tangential to the immersed boundary, at the virtual point is determined through interpolation. Tessicini et al. [2002] compared the results to a resolved LES with boundary-fitted coordinates, and found better agreement with the IBM simulations when the wall model was used than when it was not.

Senocak et al. [2004] propose a wall model with an immersed boundary method and use it to model neutral atmospheric boundary layer flow. We present a similar implementation in chapter 8 to model the same flow using the WRF code. The method used in Senocak et al. [2004] assumes that the first two fluid nodes reside in the logarithmic layer, and enforces a logarithmic velocity profile at the first fluid node above the terrain surface through the relationship given in equation 2.13.

$$U_1 = U_2 \frac{\ln(z_1/z_o)}{\ln(z_2/z_o)} \tag{2.13}$$

Here, U_2 is the horizontal velocity of the second fluid node, which is known, and z_o, z_1, and z_2 are the roughness length scale and the heights of the first and second nodes above the immersed boundary. Further details of this method are given in chapter 8.

Choi et al. [2007] use a method similar to that of Gilmanov et al. [2003] and Gilmanov and Sotiropoulos [2005], but additionally introduce a concept which they term ‘tangency correction’. In this method, the velocity is decomposed along the surface normal into tangential and normal components. The tangential component is reconstructed at the first fluid node using a power law function in the wall normal direction. The normal component is reconstructed using a cubic function, which is chosen so that the second derivative of the normal velocity vanishes at the immersed surface.

Additionally, our own methods for using the immersed boundary method with high Reynolds number flows are developed. Two methods for combining the immersed boundary method with equilibrium wall models are presented in chapter 8. The first method is a velocity reconstruction approach based on the method of Senocak et al. [2004]. The second method reconstructs the wall shear stress at a ghost point. To the author’s knowledge this is the first instance of a combined wall model and IBM method using a ghost cell approach. Both models enforce an instantaneous logarithmic velocity profile. As we are modeling complex terrain, this mean velocity profile is not valid at every point along the boundary. However, we still feel that this type of model is the best choice for coupling to an immersed boundary method because it is already used at the bottom boundary in meteorological models (including WRF) with terrain-following coordinates, and it is the most cost effective implementation of a wall model.

Chapter 3

Details of the Weather Research and Forecasting model

The WRF model is used for the simulations presented in this work. There are two main reasons for choosing this code. First, while we are extending the use of WRF to completely new applications and scales, the WRF dynamical core and its surrounding software framework and physics interfaces are well suited to our application of atmospheric boundary layer flows. WRF has been validated for numerical weather prediction [Klemp et al., 2007; Skamarock and Klemp, 2008], and the software framework for parallelization and grid nesting has been extensively developed [Michalakes et al., 2005]. The WRF model includes the important ability to nest from the mesoscale to LES scales, and includes basic eddy viscosity type LES turbulence closures. Secondly WRF is widely distributed, with well over 5,000 subscribed users, and 10,000 visitors to the code repository. Contributions to the WRF code have a large and far reaching effect on the atmospheric modeling community.

In this chapter, further details are given on the WRF model. The immersed boundary forcing is added to the governing equations; therefore, it is useful to understand these equations in detail. The governing equations are derived in section 3.2. The derivation begins with the compressible Euler equations, and ends with the perturbation form of the equations that have been transformed into the terrain-following pressure coordinate used in WRF. Section 3.3 discusses the time advancement scheme used in WRF, and section 3.4 details the treatment of lateral and vertical boundary conditions in WRF. It is especially important to examine the surface boundary condition, as this condition is the focus of the discussion in subsequent chapters on the implementation of the immersed boundary method.

3.1 General description of WRF

WRF is an open source community model that is designed to be used for a variety of purposes ranging from operational weather prediction to idealized geophysical flow simulations. The software is designed to be flexible and modular, which facilitates development of the code by the broad academic community. Currently, there are two dynamic solvers, called cores, that operate within the WRF software framework. These are the Non-hydrostatic Mesoscale Model (NMM) core and the Eulerian Mass (EM) core. The NMM core is used operationally (i.e. in weather forecasting by the National Weather Service), and the EM core is generally used for research purposes. The EM core is also known as Advanced Research WRF (ARW), and is the focus of the following discussion. ARW has principally been developed by the National Center for Atmospheric Research (NCAR). As of 2010, NCAR continues to develop the model, as well as provide user support. The EM core solves the non-hydrostatic compressible Euler equations, although a hydrostatic option is included. The code is fully portable, designed to operate in a massively parallel environment, and includes directives for both MPI and OpenMP.

ARW is a conservative finite difference model that is spatially discretized using an Arakawa-C staggered grid. Uniform grid spacing is used in the horizontal directions, and a terrain-following pressure coordinate is used in the vertical direction. Additionally, a stretching function may be applied to the vertical coordinate. Second to sixth order advection schemes are available in the WRF model. Even orders are centered, while odd orders are upwind biased. A time-split integration scheme is used to deal with the full range of frequencies admitted by the Euler equations. In this scheme a third order explicit Runge-Kutta method is used for time advancement of meteorologically significant low frequency physical modes, while a smaller time step is used to account for higher frequency modes such as acoustic and Lamb waves. Horizontally propagating acoustic modes are integrated using a forward-backward scheme, and vertically propagating acoustic modes and buoyancy oscillations are treated implicitly. One and two way grid nesting, as well as moving nests are supported.

Periodic, open, symmetric, and specified boundary conditions are allowed in the lateral directions. The bottom boundary condition sets the contravariant or normal velocity to zero, and allows free slip of the covariant velocities. The top boundary is a surface of constant pressure, and enforces the Cartesian vertical velocity w to be zero. Damping functions may be applied at the top boundary to control the reflection of waves.

3.2 The governing equations of WRF

WRF uses a terrain-following pressure coordinate, shown in Figure 3.1. Historically, it was convenient in many meteorological applications to use hydrostatic pressure as an independent variable instead of height. This notion of pressure as an independent coordinate in

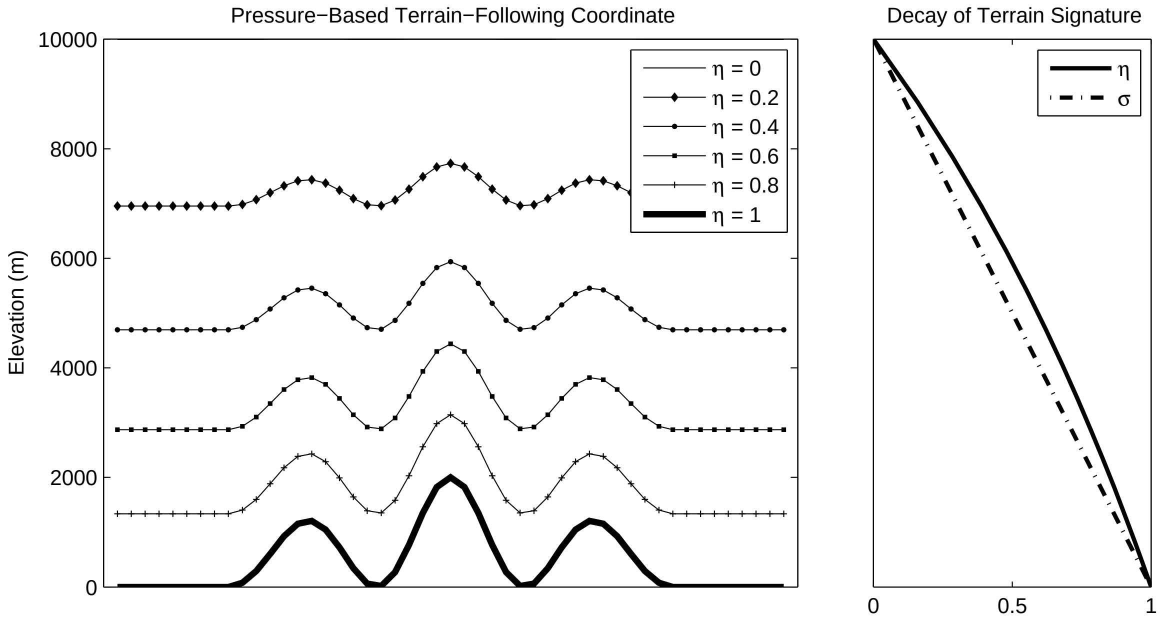


Figure 3.1. Plotted on the left is an example of the η coordinate used by WRF, which is a pressure-based terrain-following coordinate. On the right, the decay of the amplitude of the terrain signature is shown with respect to elevation for η (pressure-based) and σ (height-based) coordinates. It can be seen that the amplitude of the disturbance of the coordinate is larger for the η coordinate, and does not decay as rapidly with elevation when compared to the σ coordinate, which decays linearly.

meteorology has been around since at least 1910, when it was published by Bjerknes and Coll. in *Dynamic Meteorology and Hydrography.* Later, Eliassen [1949] framed the hydrostatic primitive equations using an isobaric vertical coordinate. Eliassen and others explained that in this framework, the vertical velocity can be diagnosed from the continuity equation using the instantaneous horizontal velocities and the thermodynamic fluid properties. This unique property would eventually lead to the development of many atmospheric numerical models with isobaric coordinate systems, because of the ease of implementation and significant computational savings arising from diagnosing the vertical velocity. Phillips further advanced the use of pressure coordinates when in 1957 he proposed the σ coordinate system. In the previous pressure system, the terrain surface did not coincide with a coordinate surface. In the σ system, $\sigma = p/\pi$, where π is the pressure at the Earth's surface. The non-dimensional σ coordinate ranges from unity at the surface to zero at the top of the atmosphere, and provides the important property of aligning the lower coordinate surface with the terrain. Kasahara [1974] synthesized all of these developments, and succinctly posed the equations of motion in terms of height, isobaric, and isentropic σ coordinates. Additionally, Kasahara transformed the equation for conservation of energy into each of the three coordinate systems.

As computational power increased and meshes became finer, it became clear that the hydrostatic assumption that worked so well at synoptic scales (100's of kilometers) did not work well at the mesoscale (10's of kilometers). In fact, Laprise [1992] notes that non-hydrostatic effects become perceptible when the spatial scale of interest falls below 100km, and must be incorporated at scales of 10km. For this reason, Laprise developed a transformation of the fully compressible non-hydrostatic Euler equations into a terrain-following hydrostatic pressure coordinate. In this transformation the advantage of a diagnostic continuity equation is lost, and it once again becomes prognostic. However, the equations of Laprise [1992] revert to the form given by Kasahara [1974] in the hydrostatic limit. The work of Laprise became the foundation for the framework of WRF, which solves the perturbation form of these equations for a moist atmosphere.

3.2.1 The compressible Euler equations

The prognostic equations governing the WRF model are given in Skamarock et al. [2007], however, it is worthwhile to investigate their derivation. They differ from the compressible Euler equations by the transformation to the terrain-following pressure coordinate, inclusion of moisture, map projections, and transformation to a perturbation form. The WRF form of the governing equations can be derived beginning from the inviscid and compressible Navier-Stokes equations given by (3.1).

$$\begin{gathered}\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V} + \alpha \nabla p + \vec{g} = F \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = 0\end{gathered} \tag{3.1}$$

Here α is the specific volume, and F includes Coriolis effects and any additional forcing terms such as turbulent mixing or model physics. In order for the Navier-Stokes equations to be used in WRF, they must first be transformed into a vertical terrain-following pressure coordinate. This is accomplished in two parts. First the equations are transformed to a pressure based coordinate system, and the hydrostatic assumption is imposed as the coordinate definition. Then the transformation to the terrain-following coordinate system is made. Finally, the perturbation form of the governing equations is discussed. For the purposes of this discussion the addition of moisture and mapping factors, which project the computational domain onto the Earth's surface, are not included.

3.2.2 Transformation to pressure coordinates

Following the transformation given by Kasahara [1974], and taking the pressure coordinate to be π, derivatives take the following form for any scalar a:

$$\begin{aligned}\left(\frac{\partial a}{\partial t}\right)_\pi &= \left(\frac{\partial a}{\partial t}\right)_z + \frac{\partial a}{\partial z}\left(\frac{\partial z}{\partial t}\right)_\pi \\ \left(\frac{\partial a}{\partial x}\right)_\pi &= \left(\frac{\partial a}{\partial x}\right)_z + \frac{\partial a}{\partial z}\left(\frac{\partial z}{\partial x}\right)_\pi .\end{aligned}$$

The subscript denotes the vertical coordinate that is being held constant for the partial differentiation. After substitution of the relationship $\frac{\partial a}{\partial z} = \frac{\partial a}{\partial \pi}\frac{\partial \pi}{\partial z}$ the temporal and spatial derivatives for the π coordinate take the form of equations (3.2a) and (3.2b) respectively.

$$\left(\frac{\partial a}{\partial t}\right)_\pi = \left(\frac{\partial a}{\partial t}\right)_z + \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial t}\right)_\pi \frac{\partial a}{\partial \pi} \tag{3.2a}$$

$$\left(\frac{\partial a}{\partial x}\right)_\pi = \left(\frac{\partial a}{\partial x}\right)_z + \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial x}\right)_\pi \frac{\partial a}{\partial \pi} \tag{3.2b}$$

Using the above transformations, the material derivative can be determined to take the form:

$$\begin{aligned}\left(\frac{Da}{Dt}\right)_\pi = &\left(\frac{\partial a}{\partial t}\right)_\pi - \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial t}\right)_\pi \frac{\partial a}{\partial \pi} \\ &+ u\left[\left(\frac{\partial a}{\partial x}\right)_\pi - \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial x}\right)_\pi \frac{\partial a}{\partial \pi}\right] \\ &+ v\left[\left(\frac{\partial a}{\partial y}\right)_\pi - \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial y}\right)_\pi \frac{\partial a}{\partial \pi}\right] \\ &+ w\frac{\partial \pi}{\partial z}\frac{\partial a}{\partial \pi}.\end{aligned} \tag{3.3}$$

Rearranging the terms of the material derivative by distributing the u and v velocities to the terms inside of the square brackets forms equation (3.4).

$$\left(\frac{Da}{Dt}\right)_\pi = \left(\frac{\partial a}{\partial t}\right)_\pi + u\left(\frac{\partial a}{\partial x}\right)_\pi + v\left(\frac{\partial a}{\partial y}\right)_\pi + \left[w - \left(\frac{\partial z}{\partial t} + u\frac{\partial z}{\partial x} + v\frac{\partial z}{\partial y}\right)_\pi\right]\frac{\partial \pi}{\partial z}\frac{\partial a}{\partial \pi} \tag{3.4}$$

The material derivative can then be arranged in the familiar form of (3.5a), where the expression for $\dot{\pi}$ or $\frac{\partial \pi}{\partial t}$ is defined to be of the form (3.5b).

$$\left(\frac{Da}{Dt}\right)_\pi = \left(\frac{\partial a}{\partial t}\right)_\pi + u\left(\frac{\partial a}{\partial x}\right)_\pi + v\left(\frac{\partial a}{\partial y}\right)_\pi + \dot{\pi}\frac{\partial a}{\partial \pi} \tag{3.5a}$$

$$\dot{\pi} = \frac{\partial \pi}{\partial t} = \frac{\partial \pi}{\partial z}\left[w - \left(\frac{\partial z}{\partial t} + u\frac{\partial z}{\partial x} + v\frac{\partial z}{\partial y}\right)_\pi\right] \tag{3.5b}$$

Equation (3.5b) can be solved for the vertical velocity w, which yields equation (3.6).

$$w = \left(\frac{\partial z}{\partial t} + u\frac{\partial z}{\partial x} + v\frac{\partial z}{\partial y}\right)_\pi + \dot{\pi}\frac{\partial z}{\partial \pi} \tag{3.6}$$

Additionally, after many instances of the chain rule, the vertical derivative of w is also found to be equation (3.7).

$$\frac{\partial w}{\partial z} = \frac{\partial w}{\partial \pi}\frac{\partial \pi}{\partial z} = \\ \frac{\partial \pi}{\partial z}\left[\left[\frac{\partial}{\partial t}\left(\frac{\partial z}{\partial \pi}\right) + u\frac{\partial}{\partial x}\left(\frac{\partial z}{\partial \pi}\right) + \frac{\partial u}{\partial \pi}\frac{\partial z}{\partial x} + v\frac{\partial}{\partial y}\left(\frac{\partial z}{\partial \pi}\right) + \frac{\partial v}{\partial \pi}\frac{\partial z}{\partial y}\right]_\pi + \dot{\pi}\frac{\partial}{\partial \pi}\left(\frac{\partial z}{\partial \pi}\right) + \frac{\partial \dot{\pi}}{\partial \pi}\frac{\partial z}{\partial \pi}\right] \tag{3.7}$$

Using the definition of the material derivative in the π coordinate system given by the set of equations in (3.5), the vertical derivative of w can be significantly simplified from the above form. Once simplified, the vertical derivative of w is expressed as (3.8).

$$\frac{\partial w}{\partial z} = \frac{\partial w}{\partial \pi}\frac{\partial \pi}{\partial z} = \frac{\partial \pi}{\partial z}\left[\frac{D}{Dt}\left(\frac{\partial z}{\partial \pi}\right) + \frac{\partial u}{\partial \pi}\frac{\partial z}{\partial x} + \frac{\partial v}{\partial \pi}\frac{\partial z}{\partial y}\right]_\pi + \frac{\partial \dot{\pi}}{\partial \pi} \tag{3.8}$$

With the determination of temporal and spatial derivatives complete, the continuity equation may now be transformed into a pressure coordinate system. This is accomplished by substituting the transformations for the partial derivatives in x, y, z and time into the continuity equation. The temporal and horizontal spatial partial derivatives were given above by equation (3.2), and the partial derivative with respect to z is given by equation (3.8). After these substitutions, the continuity equation in the pressure coordinate system is (3.9).

$$\left(\frac{\partial \rho}{\partial t}\right)_\pi - \frac{\partial \rho}{\partial \pi}\frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial t}\right)_\pi + \left(\frac{\partial(\rho u)}{\partial x}\right)_\pi - \frac{\partial(\rho u)}{\partial \pi}\frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial x}\right)_\pi + \left(\frac{\partial(\rho v)}{\partial y}\right)_\pi \\ - \frac{\partial(\rho v)}{\partial \pi}\frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial y}\right)_\pi + \frac{\partial \rho\dot{\pi}}{\partial \pi} + \frac{\partial \pi}{\partial z}\left[\frac{D}{Dt}\left(\rho\frac{\partial z}{\partial \pi}\right) + \frac{\partial(\rho u)}{\partial \pi}\frac{\partial z}{\partial x} + \frac{\partial(\rho v)}{\partial \pi}\frac{\partial z}{\partial y}\right]_\pi = 0 \tag{3.9}$$

The equation above then simplifies (after many more instances of the chain rule) to the much more familiar form of the continuity equation given by Kasahara [1974].

$$\left[\frac{\partial}{\partial t}(\rho\frac{\partial z}{\partial \pi}) + \frac{\partial}{\partial x}(\rho u\frac{\partial z}{\partial \pi}) + \frac{\partial}{\partial y}(\rho v\frac{\partial z}{\partial \pi})\right]_\pi + \frac{\partial}{\partial \pi}\left(\rho\dot{\pi}\frac{\partial z}{\partial \pi}\right) = 0 \tag{3.10}$$

Next, the momentum equation can be transformed into the pressure coordinate system using the same substitutions for the partial derivatives. As is standard in the literature on coordinate transformations for atmospheric applications, the horizontal and vertical momentum equations are given separately. In this case the vector $\vec{V}$ includes only the horizontal

u and v velocities. The horizontal momentum equation in the pressure coordinate system is found to be

$$\left(\frac{D\vec{V}}{Dt}\right)_\pi + \alpha\nabla_\pi p - \alpha\frac{\partial\pi}{\partial z}(\nabla_\pi z)\frac{\partial p}{\partial\pi} = F, \tag{3.11}$$

and the vertical momentum equation is

$$\left(\frac{Dw}{Dt}\right)_\pi + \alpha\frac{\partial p}{\partial\pi}\frac{\partial\pi}{\partial z} + g = F. \tag{3.12}$$

The effects of the transformation on the unsteady and advective terms in the momentum equations are accounted for by the definition of the transformed material derivative for the new π pressure coordinate. The changes to the pressure term obviously follow the spatial derivative transformation. It should be noted that the material derivative in the vertical momentum equation is operating on w, while the definition of the material derivative given by (3.5a) includes the vertical velocity $\dot{\pi}$ in the last term. Assuming the pressure coordinate π to be hydrostatic, the vertical gradient of the coordinate can be determined as:

$$\frac{\partial\pi}{\partial z} = -\rho g.$$

The use of the hydrostatic assumption above yields the following definitions shown in (3.13) for temporal and spatial derivatives in the vertical pressure coordinate system, where the operator ∇_π represents the horizontal gradients and D/Dt is the three-dimensional material derivative. As before $\vec{V}$ includes only the horizontal velocity components. In equations (3.13) geopotential has been introduced, and is defined as $\phi = gz$.

$$\left(\frac{\partial}{\partial t}\right)_\pi = \left(\frac{\partial}{\partial t}\right)_z - \rho\left(\frac{\partial\phi}{\partial t}\right)_\pi\frac{\partial}{\partial\pi} \tag{3.13a}$$

$$\nabla_\pi a = \nabla_z a - \rho(\nabla_\pi\phi)\frac{\partial a}{\partial\pi} \tag{3.13b}$$

$$\nabla_\pi\cdot\vec{V} = \nabla_z\cdot\vec{V} - (\rho\nabla_\pi\phi)\cdot\frac{\partial\vec{V}}{\partial\pi} \tag{3.13c}$$

$$\left(\frac{D}{Dt}\right)_\pi = \left(\frac{\partial}{\partial t}\right)_\pi + \vec{V}\cdot\nabla_\pi + \dot{\pi}\frac{\partial}{\partial\pi} \tag{3.13d}$$

When the hydrostatic assumption is imposed for the pressure coordinate, the unsteady term in the continuity equation vanishes. The continuity equation simplifies from the form given in (3.10) to be

$$\nabla_\pi\cdot\vec{V} + \frac{\partial\dot{\pi}}{\partial\pi} = 0, \tag{3.14}$$

and the horizontal and vertical momentum equations are

$$\left(\frac{D\vec{V}}{Dt}\right)_\pi + \alpha\nabla_\pi p + \frac{\partial p}{\partial\pi}\nabla_\pi\phi = F \tag{3.15}$$

$$\left(\frac{Dw}{Dt}\right)_\pi + g\left(1 - \frac{\partial p}{\partial\pi}\right) = F. \tag{3.16}$$

The set of equations above consisting of (3.14), (3.15), and (3.16) are the fully compressible non-hydrostatic Euler equations given by Laprise [1992].

3.2.3 Transformation to terrain-following pressure coordinates

WRF uses a terrain-following hydrostatic pressure coordinate or mass coordinate η given in terms of the dry hydrostatic pressure P_{hs}. The coordinate η is defined such that it is zero at the top of the model, and unity at the surface of the terrain. The mass of the fluid in the column per unit area is then μ. This yields the coordinate definition $\eta = \frac{P_{hs}-P_{hstop}}{\mu}$, where $\mu(x,y) = P_{hs\,surface} - P_{hs\,top}$. The transformation to the terrain-following coordinate system uses the same formulation for the temporal and spatial derivatives as used previously. Following the derivative formulation, the temporal and spatial derivatives in (3.17) for the η coordinate are analogous to those in (3.2) for the π coordinate.

$$\left(\frac{\partial a}{\partial t}\right)_\eta = \left(\frac{\partial a}{\partial t}\right)_\pi + \frac{\partial \eta}{\partial \pi}\left(\frac{\partial \pi}{\partial t}\right)_\eta \frac{\partial a}{\partial \eta} \tag{3.17a}$$

$$\left(\frac{\partial a}{\partial x}\right)_\eta = \left(\frac{\partial a}{\partial x}\right)_\pi + \frac{\partial \eta}{\partial \pi}\left(\frac{\partial \pi}{\partial x}\right)_\eta \frac{\partial a}{\partial \eta} \tag{3.17b}$$

The material derivative in the η coordinate is

$$\left(\frac{D}{Dt}\right)_\eta = \left(\frac{\partial}{\partial t}\right)_\eta + \vec{V}\cdot\nabla_\eta + \dot{\eta}\frac{\partial}{\partial \eta}, \tag{3.18}$$

where

$$\dot{\eta} = \frac{\partial \eta}{\partial t} = \frac{\partial \eta}{\partial \pi}\left[\dot{\pi} - \left(\frac{\partial \pi}{\partial t} + u\frac{\partial \pi}{\partial x} + v\frac{\partial \pi}{\partial y}\right)_\eta\right]. \tag{3.19}$$

The physical meaning of $\dot{\eta}$ is the contravariant velocity of the vertical coordinate. Following the derivation of the continuity equation developed by Kasahara [1974] and explained in detail in the previous section, the continuity equation in terrain-following coordinates becomes equation (3.20).

$$\left[\frac{\partial}{\partial t}\left(\frac{\partial \pi}{\partial \eta}\right) + \frac{\partial}{\partial x}\left(u\frac{\partial \pi}{\partial \eta}\right) + \frac{\partial}{\partial y}\left(v\frac{\partial \pi}{\partial \eta}\right)\right]_\eta + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial \pi}{\partial \eta}\right) = 0 \tag{3.20}$$

This form of the continuity equation was given by both Kasahara [1974] and Laprise [1992]. The horizontal and vertical momentum equations are then found to be

$$\left(\frac{D\vec{V}}{Dt}\right)_\eta + \alpha\left(\nabla_\eta p - (\nabla_\eta \pi)\frac{\partial \eta}{\partial \pi}\frac{\partial p}{\partial \eta}\right) + \frac{\partial p}{\partial \eta}\frac{\partial \eta}{\partial \pi}\left(\nabla_\eta \phi - (\nabla_\eta \pi)\frac{\partial \eta}{\partial \pi}\frac{\partial p}{\partial \eta}\right) = F \tag{3.21a}$$

$$\left(\frac{Dw}{Dt}\right)_\eta + g\left(1 - \frac{\partial p}{\partial \eta}\frac{\partial \eta}{\partial \pi}\right) = F. \tag{3.21b}$$

In addition it is noted that the definition of the dry hydrostatic η coordinate used in WRF yields the relationships $\partial\eta/\partial\pi = 1/\mu$ and $\partial\phi/\partial\eta = -\alpha\mu$. When these relationships are substituted into the continuity equation (3.20), its form simplifies substantially. The simplified continuity equation below is equivalent to the equation given in the WRF description by Skamarock et al. [2007], where the notation differs slightly. Kasahara and Laprise use the del notation to operate only in the horizontal dimensions (as was used above), while Skamarock et al. uses a three-dimensional del operator.

$$\left(\frac{\partial\mu}{\partial t} + \frac{\partial(\mu u)}{\partial x} + \frac{\partial(\mu v)}{\partial y}\right)_\eta + \frac{\partial(\mu\dot{\eta})}{\partial\eta} = 0 \tag{3.22}$$

The above relationships for the η coordinate can also be utilized to simplify the momentum equations. After multiplying the horizontal momentum equations by μ, the following form is found:

$$\mu\left(\frac{D\vec{V}}{Dt}\right)_\eta - \frac{\partial\phi}{\partial\eta}\nabla_\eta p + \frac{\partial p}{\partial\eta}\nabla_\eta\phi = F. \tag{3.23}$$

Some manipulation is required to get the momentum equation into the strong conservation form, as it appears in [Skamarock et al., 2007, Section 2.2]. From the continuity equation, we know that $\frac{\partial\mu}{\partial t} = -\nabla_\eta\cdot(\mu\vec{V}) - \frac{\partial(\mu\dot{\eta})}{\partial\eta}$. The terms on the left and right hand side are added and subtracted respectively from the momentum equation. The term $p\frac{\partial}{\partial x}(\frac{\partial\phi}{\partial\eta})$ is also added and subtracted to the horizontal momentum equation, while the order of the partial differentiation is interchanged. These manipulations yield the strong conservation form of the momentum equations appearing in the WRF description. In this equation the semicolon notation represents the dyadic product.

$$\left(\frac{\partial(\mu\vec{V})}{\partial t}\right)_\eta + \nabla_\eta\cdot(\mu\vec{V};\vec{V}) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}\vec{V}) - \frac{\partial}{\partial x}\left(p\frac{\partial\phi}{\partial\eta}\right) + \frac{\partial}{\partial\eta}\left(p\frac{\partial\phi}{\partial x}\right) = F \tag{3.24}$$

$$\left(\frac{\partial(\mu w)}{\partial t}\right)_\eta + \nabla_\eta\cdot(\mu\vec{V}w) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}w) - g\left(\frac{\partial p}{\partial\eta} - \mu\right) = F \tag{3.25}$$

In addition to the conservation of mass and momentum, an equation for potential temperature is solved. Potential temperature θ is a conserved quantity when the atmosphere is assumed to be adiabatic, so the governing equation takes the form used for a conserved scalar.

$$\left(\frac{\partial(\mu\theta)}{\partial t}\right)_\eta + \nabla_\eta\cdot(\mu\vec{V}\theta) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}\theta) = F_\theta \tag{3.26}$$

Pressure is then diagnosed from the equation of state below, where γ is the ratio of heat capacities of dry air C_p/C_v and R_{dry} is the universal gas constant.

$$p = p_o\left(\frac{R_{dry}\theta}{p_o\alpha_{dry}}\right)^{\gamma_{dry}} \tag{3.27}$$

3.2.4 Perturbation form of the governing equations

It is advantageous to recast the governing equations into a perturbation form that is a departure from the hydrostatic state. This removes large canceling contributions from the horizontal pressure gradients, and reduces numerical error. It is assumed that pressure, specific volume, geopotential, and column mass per unit area take the form $p = \bar{p}(z) + p'$, $\alpha = \bar{\alpha}(z) + \alpha'$, $\phi = \bar{\phi}(z) + \phi'$, and $\mu_d = \bar{\mu}_d(x, y) + \mu_d{}'$ where the perturbation is the deviation from the hydrostatic and time invariant reference state given by $\nabla\bar{p} = g/\bar{\alpha}$. The hydrostatic reference state variables $\bar{p}$, $\bar{\alpha}$, and $\bar{\phi}$ are strictly functions of z in Cartesian coordinates, but are functions of (x, y, η) in the transformed terrain-following coordinate. After substituting in the mean and perturbation values, the reference state may be subtracted. That is, $\bar{\alpha}\nabla_\eta\bar{p} = \nabla_\eta\bar{\phi}$ may be subtracted. It is also noted that the following relationships exist within the framework of the perturbation formulation $\partial\bar{p}/\partial\eta = \bar{\mu}$, $\partial\bar{\phi}/\partial\eta = -\bar{\mu}\bar{\alpha}$, and $\partial\phi'/\partial\eta = -\bar{\mu}\alpha' - \mu'\alpha$. Now the time invariance of the base state is used, and the equation for mass conservation becomes

$$\left(\frac{\partial\mu'}{\partial t} + \frac{\partial(\mu u)}{\partial x} + \frac{\partial(\mu v)}{\partial y}\right)_\eta + \frac{\partial(\mu\dot{\eta})}{\partial\eta} = 0. \tag{3.28}$$

To develop the perturbation form for the horizontal momentum equations, the form given in (3.29) is used as a starting point.

$$\left(\frac{\partial(\mu\vec{V})}{\partial t}\right)_\eta + \nabla_\eta\cdot(\mu\vec{V};\vec{V}) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}\vec{V}) + \mu\alpha\nabla_\eta p + \frac{\partial p}{\partial\eta}\nabla_\eta\phi = F \tag{3.29}$$

Substitution of the base and perturbation variables changes only the pressure gradient in the horizontal conservation of momentum, and both the pressure and gravity terms in the vertical. In the horizontal momentum equation, the pressure term becomes $\mu\bar{\alpha}\nabla_\eta\bar{p} + \mu\alpha\nabla_\eta p' + \mu\alpha'\nabla_\eta\bar{p} + \mu\nabla_\eta\bar{\phi} + \mu\nabla_\eta\phi' + \frac{\partial p'}{\partial\eta}(\nabla_\eta\phi) - \mu'(\nabla_\eta\phi)$. When the base state is subtracted, the perturbation form of the horizontal momentum equation is

$$\begin{aligned}\left(\frac{\partial(\mu\vec{V})}{\partial t}\right)_\eta &+ \nabla_\eta\cdot(\mu\vec{V};\vec{V}) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}\vec{V})\\ &+ \mu\alpha\nabla_\eta p' + \mu\alpha'\nabla_\eta\bar{p} + \mu\nabla_\eta\phi' + (\nabla_\eta\phi)\left(\frac{\partial p'}{\partial\eta} - \mu'\right) = F.\end{aligned} \tag{3.30}$$

Substitution of the base state and perturbation quantities into the vertical momentum equation yields

$$\left(\frac{\partial(\mu w)}{\partial t}\right)_\eta + \nabla_\eta\cdot(\mu\vec{V}w) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}w) - g\left(\frac{\partial\bar{p}}{\partial\eta} + \frac{\partial p'}{\partial\eta} - \bar{\mu} - \mu'\right) = F \tag{3.31}$$

The contribution from the base state cancels out, and the perturbation form of the vertical momentum equation becomes

$$\left(\frac{\partial(\mu w)}{\partial t}\right)_\eta + \nabla_\eta\cdot(\mu\vec{V}w) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}w) - g\left(\frac{\partial p'}{\partial\eta} - \mu'\right) = F. \tag{3.32}$$

The equation for the conservation of potential temperature remains unchanged. The diagnostic relationship for pressure is non-linear, and therefore cannot be represented in perturbation form without approximation. Therefore, the final perturbation form of the transformed compressible Euler equations is the set of equations (3.28), (3.30), and (3.32).

3.3 Time integration

After the perturbation equations are temporally and spatially discretized, they are further divided into low frequency and high frequency acoustic terms for the time-split advancement scheme, which is documented in Wicker and Skamarock [2002]. The low frequency terms are meteorologically significant physical modes such as Rossby waves, gravity waves, and simple advection. High frequency modes such as acoustic and Lamb waves are integrated on a smaller time step to maintain numerical stability. This is a common strategy for achieving computational efficiency, because the most expensive terms to evaluate are calculated on the large time step.

A three step explicit Runge-Kutta (R-K) method is used to advance the large time step. Horizontally propagating acoustic modes are advanced using a forward-backward explicit scheme. In the forward-backward scheme the momentum equations are advanced using a forward scheme relative to the pressure term. Then the updated velocities are used with a backwards scheme to advance the pressure term. The vertical acoustic modes and buoyancy oscillations are advanced implicitly. This removes any restrictions on the time step for vertically propagating acoustic waves and the buoyancy frequency. Implicit treatment in the vertical direction is needed because the vertical grid spacing is generally much smaller than the horizontal grid spacing, and therefore supports higher frequency modes.

Both the large time step and the acoustic time step can be user defined or calculated by WRF. It is common to use six to twelve acoustic time steps per large time step. When n acoustic time steps are used, one is taken in the first R-K step, $n/2$ in the second R-K step, and n in the third R-K step. The time advancement scheme proceeds with the loop for the small acoustic time step nested into the larger Runge-Kutta loop. The time advancement sequence used in WRF is outlined in Table 3.1.

3.4 Boundary conditions

Several options for lateral boundary conditions are available to WRF users. These are detailed in the NCAR technical note by Skamarock et al. [2007] and include periodic, open or radiative, symmetric, and specified options. The first three boundary conditions are often used in idealized cases, where as specified boundary conditions are common in cases with real external data. Additionally, WRF supports one and two way horizontal nesting. Nested

Table 3.1. The time advancement sequence used in WRF. The Runge-Kutta loop is preformed three times.

```
Begin Runge-Kutta Loop
    1. For the first R-K step only: Compute F (turbulence and physics terms)
    2. Compute the tendencies for advection and pressure
    Begin Acoustic Loop
        3. Advance the horizontal momentum equations
        4. Advance the continuity equation
        5. Advance the conservation of potential temperature
        6. Advance the vertical momentum equation
        7. Diagnose acoustic step perturbations of pressure and density
    End Acoustic Loop
    8. Advance scalar equations
    9. Diagnose pressure and density perturbations on the R-K step
End Runge-Kutta Loop
```

domains can be run in a serial fashion, where the simulation of a parent domain must be complete before the simulation of the child domain. This type of nesting is equivalent to producing two simulations with an intermediate processing step. Alternatively, the domains can be run concurrently, where the child and parent domain run at the same time. Vertical nesting is implemented for nested domains which are run serially because vertical interpolation occurs during the intermediate processing step; however, vertical nesting cannot be used in concurrent runs. This means that for domains run concurrently, an interior nested domain may have finer resolution in the horizontal extents, but the resolution in the vertical dimension remains fixed. In a one-way nest, the fine domain receives boundary conditions interpolated from the coarse domain. In a two-way nest, which can only be used with concurrent simulations, the solution from the fine domain additionally replaces the solution on the coarse domain at each time step.

The Arakawa-C staggered grid, shown in Figure 3.2, begins with a u velocity point in the x dimension and a v velocity point in the y dimension. This results in (nx) u grid points in x, and $(nx-1)$ v and w grid points in x, where (nx) is the user specified number of grid points in the x direction. Conversely in the y direction there are (ny) v grid points, and $(ny-1)$ u and w grid points. WRF includes six extra grid points outside of the domain in each of the horizontal extents (for a total of 12 extra points in each direction), used to set the lateral boundary conditions. When MPI is used, WRF passes the information for up to six lateral grid points between processors in a so-called halo exchange.

In the vertical direction the top boundary condition is specified to be isobaric, and the Cartesian vertical velocity w is set to zero. Additionally, gravity waves can be absorbed with a diffusion or Rayleigh damping layer. The Rayleigh damping layer can be applied to all three velocity components, or just to the vertical velocity [Klemp et al., 2008]. At the bottom boundary the contravariant coordinate velocity is set to zero, and a kinematic boundary condition is used for the Cartesian vertical velocity. The set of equations given by

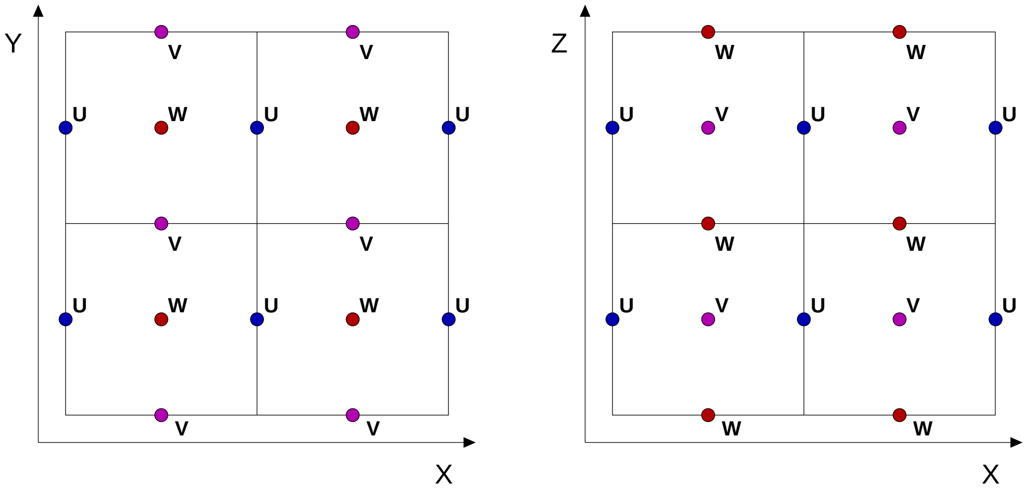


Figure 3.2. A plan view and elevation view of the WRF staggered grid.

(3.33a) and (3.33b) create a free slip bottom boundary condition.

$$\dot{\eta}_{surf} = 0 \tag{3.33a}$$

$$w_{surf} = u_{surf}\frac{\partial h}{\partial x} + v_{surf}\frac{\partial h}{\partial y} \tag{3.33b}$$

In equation (3.33b) h is a function specifying the terrain height. The u and v velocities are extrapolated to the surface using a quadratic Lagrange polynomial. The shear stress at the boundary is implicitly set to zero, unless the effects of friction are taken into account with an atmospheric surface layer scheme or a user specified coefficient of drag.

In the case of a rough terrain surface, a wall model is used as discussed in chapter 2. The wall model provides an estimate of the shear stress at the wall so that a Neumann boundary condition can be imposed on the velocity field, rather than the no-slip boundary condition. The shear stress at the wall is calculated using (3.34), where C_d is the coefficient of drag and $|U|$ is the magnitude of the horizontal velocities. No adjustments are made for the use of terrain-following coordinates, so it is assumed that the horizontal velocities are tangent to the land surface. This is a source of error in sloping terrain, that becomes larger with steeper slopes.

$$\tau_{w_{xz}} = C_d|U|u, \qquad \tau_{w_{yz}} = C_d|U|v \tag{3.34}$$

If a surface layer scheme is used, then it is usually an equilibrium stress model. The surface layer scheme returns a value for the friction velocity $u^\star$, which is based on a roughness length scale defined by the land use and vegetation type of the surface, and the atmospheric stability conditions. The coefficient of drag is then defined as $C_d = (u^\star/|U|)^2$. If a surface layer scheme is not used, then the coefficient of drag must be specified as a constant for the entire domain in the user input file. Drag coefficients are provided in references such as Arya [1988].

We have added two additional boundary condition options for use with terrain-following coordinates, described in chapters 5 and 8. The first is for a no-slip boundary condition.

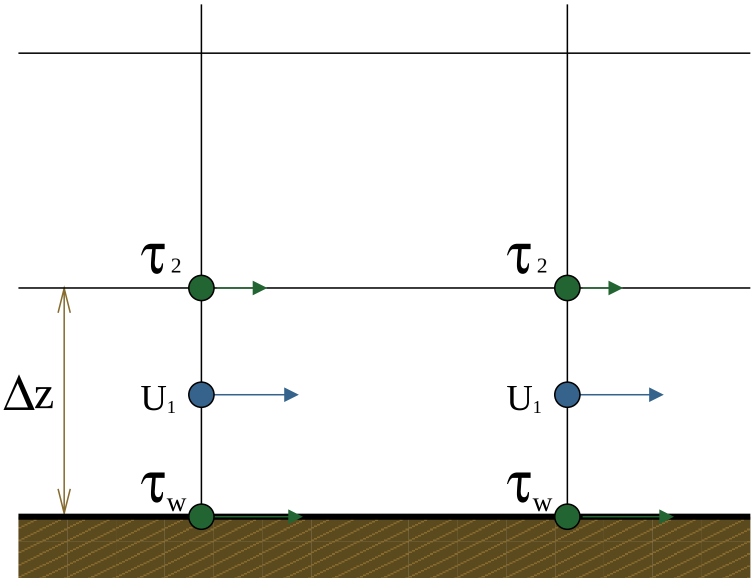


Figure 3.3. WRF sets τ_w to model a rough surface.

The second is a very simple equilibrium stress model that calculates the drag based on a user specified roughness parameter z_o. In this formulation, the coefficient of drag is given by $C_d = (\kappa / \ln \frac{z}{z_o})^2$.

Effects of the shear stress at the wall τ_w are reflected in the calculation of the diffusion terms in the horizontal momentum equation. For example, the discretized u momentum equation would be calculated as in (3.35).

$$\left.\frac{\partial(\mu u)}{\partial t}\right|_1 = \ldots + \mu \left[\frac{\tau_2}{\Delta z} - \frac{\tau_w}{\Delta z} \right] \tag{3.35}$$

Diffusion terms are calculated in physical space, so the use of the z coordinate (instead of η) is appropriate here. Grid points for u and v are located at one-half of the vertical grid spacing Δz above the surface, as shown in figure 3.3. Equation (3.35) along with equation (3.33) models a rough terrain surface.

In the vertical direction the staggered grid begins and ends at a w point, meaning that there are (nz) w points and $(nz - 1)$ u and v points. There are no halo points needed for boundary conditions. WRF never decomposes the domain in the vertical direction for parallel processing. This means that there is no need for boundary condition updates or to use MPI to exchange halos in the vertical dimension.

Chapter 4

Analysis of numerical errors arising from complex terrain

4.1 Introduction

Most mesoscale numerical models use terrain-following coordinates to accommodate complex terrain. Terrain-following or sigma coordinates conform to the bottom topography and the coordinate lines gradually become smoother and flatter with distance from the ground. Coordinate lines retain a signature of the underlying surface shape even when very far away from the ground. Coordinate transformations are introduced into the discretized equations and produce numerical truncation errors in addition to those associated with the chosen discretization scheme.

Several methods have been proposed to reduce the truncation error arising from terrain-following coordinates. Schär et al. [2002] proposed a modified sigma coordinate in terms of height in which grid distortion due to small scale terrain features decays with height more rapidly than distortion caused by large scale features. The modified coordinate flattens quickly with height and improves the accuracy of the solution. Zängl [2003] extended this method to pressure based coordinates. Klemp et al. [2003] investigated the errors that arise when numerical treatment of the metric terms is inconsistent with the discretization of other terms in the governing equations. Distortion seen in topographically induced gravity waves was reduced with consistent numerical treatment. Adcroft et al. [1997] used a shaved cell approach to represent topography on a Cartesian grid. This method eliminates grid distortion, but introduces complications in the numerical solution at the ground because the computational cells must be modified (shaved) where they intersect the topography.

Here we introduce an alternative gridding technique for flow over complex terrain using an immersed boundary method (IBM) in the Weather Research and Forecasting (WRF) model. With this method, the terrain surface intersects the grid, and variables are adjusted near the immersed boundary so that the flow is diverted by the boundary. Grid distortion and the associated truncation errors are thus avoided. Additionally, the method does not require modification of the computational stencil in the vicinity of the topography. Boundary conditions are imposed on the immersed surface for velocities and scalar quantities through interpolation. The implementation and validation of IBM in WRF in two dimensions is described in chapters 5 and 6. Here we focus on errors resulting from the coordinate transformation and the behavior of the flow far above steep topography.

The scalar transport test case of Schär et al. [2002] is presented in section 4.3. Schär et al. solved the advection-diffusion equation, but specified the underlying velocities. Zängl [2003] examined the case when the momentum equations are solved in addition to the advection-diffusion equation using the MM5 mesoscale model. In this chapter, comparisons are made between simulations using standard terrain-following coordinates and those using IBM. Large truncation errors are present in the native coordinate, and it is demonstrated that the immersed boundary method can be used within WRF to alleviate these errors. Truncation errors can be attributed to either the finite differencing scheme or the metric terms. Further analysis in section 4.4 apportions the error attributable to each cause.

4.2 Numerical method

Relevant details of the WRF coordinate, governing equations, and discretization are provided in this section, along with an introductory description of the immersed boundary method used here. Further details of the WRF governing equations and discretization techniques are found in Klemp et al. [2007] and Skamarock and Klemp [2008]. Skamarock et al. [2007] provides a complete description of the model, including default settings and model parameterizations.

4.2.1 Coordinate definitions

Terrain-following coordinates were first introduced by Phillips [1957] for numerical weather forecasting models using pressure as an independent variable representing the vertical coordinate. Gal-Chen and Somerville [1975] use a height-based sigma coordinate to map non-orthogonal coordinates onto a Cartesian grid. Most modern mesoscale models employ one of these two options to accommodate terrain, both of which introduce terrain-induced grid distortion.

A pressure or mass based vertical coordinate η is used in WRF, and is given in terms of the dry hydrostatic pressure P_{hs}. The coordinate is defined such that it is zero at the top

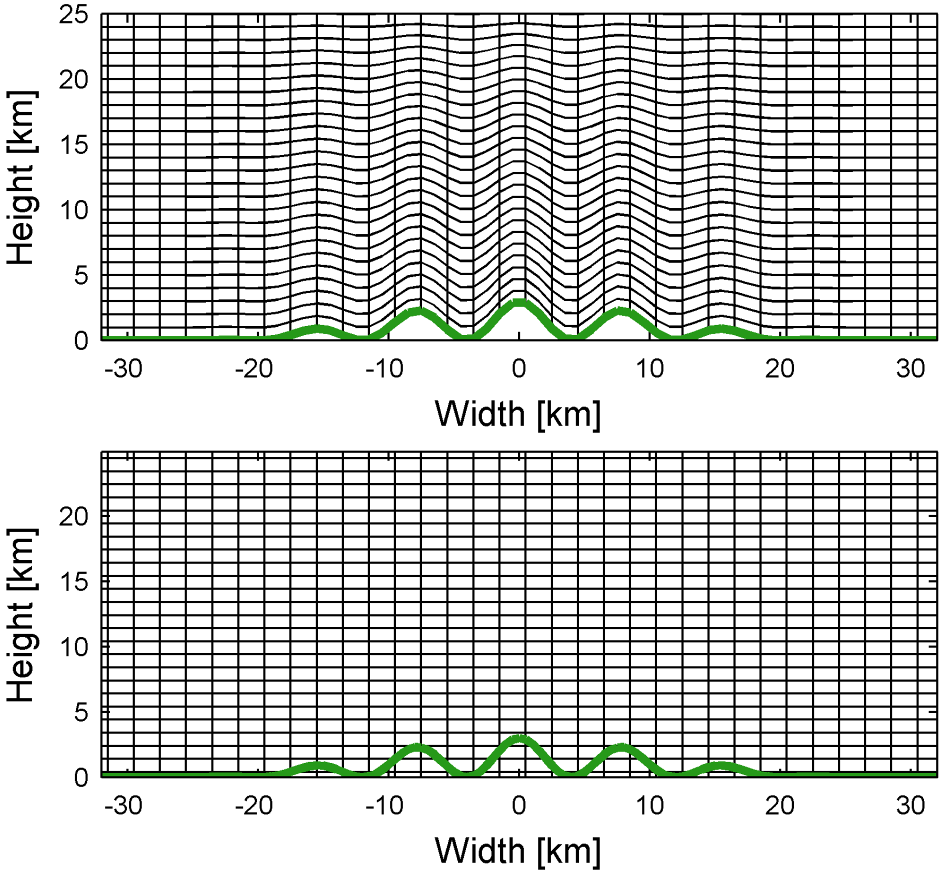


Figure 4.1. Terrain-following coordinates shown in the top figure, and coordinates where the immersed boundary method is used are show in the bottom figure. These grids are used for the idealized advection test in section 4.3. Every other coordinate line is shown.

of the computational domain and unity at the terrain surface. This yields the coordinate definition $\eta = \frac{P_{hs} - P_{hs\_top}}{\mu}$, where the column mass per unit area of the fluid is $\mu(x, y) = P_{hs\_surface} - P_{hs\_top}$. The grid used in the idealized advection test (section 4.3) is included in figure 4.1, where the η coordinate is shown in the top figure.

When the immersed boundary method is used, the grid is not transformed to align with the topography. Instead, the terrain is allowed to arbitrarily pass through the grid as shown in the bottom domain in figure 4.1. The effects of the solid boundaries on the fluid are represented by the addition of a body force term F_B in the conservation equations for momentum and scalars (4.1). Mathematically, the forcing term takes a non-zero value in the vicinity of the immersed boundary, but has no effect away from the boundaries.

$$\partial_t \vec{V} + \vec{V} \cdot \nabla \vec{V} = -\alpha \nabla p + \nu_t \nabla^2 \vec{V} + \vec{g} + \vec{F}_B \tag{4.1a}$$

$$\partial_t \varphi + \vec{V} \cdot \nabla \varphi = \nu_t \nabla^2 \varphi + F_B \tag{4.1b}$$

The forcing method used in this work is referred to as direct or discrete forcing, which was first introduced by Mohd-Yusof [1997]. With this method the velocity or scalar value is modified at forcing points near the terrain to enforce the boundary condition, eliminating the need for explicit calculation of the body force term. Terrain passes through the grid, and a bilinear interpolation method is used to determine the forcing needed at discrete grid points. The implementation of this method in the WRF model is documented in chapter 5.

4.2.2 Metric terms in the governing equations

The mesoscale model WRF solves the non-hydrostatic compressible Euler equations which have been transformed into a pressure-based terrain-following coordinate. Two coordinate transformations are required for the vertical coordinate. The first transforms the equations into the hydrostatic pressure coordinate, and the second transforms the equations into the terrain-following coordinate. An additional velocity is introduced in these transformations, and is defined as the contravariant velocity of the vertical coordinate $\dot{\eta}$. Therefore, WRF solves the transformed Navier-Stokes equations plus an additional equation representing $\dot{\eta}$. The coordinate velocity is relevant here because the definition contains terms from the Jacobian matrix for the coordinate transformation. The equation defining $\dot{\eta}$ is rearranged so that it appears in WRF as a prognostic equation for the geopotential ϕ. The transformed equations are given in 4.2.

$$\partial_t \mu + \nabla \cdot (\mu \vec{V}) + \partial_\eta (\mu \dot{\eta}) = 0 \tag{4.2a}$$

$$\begin{aligned} &\partial_t(\mu \vec{V}) + \nabla \cdot (\mu \vec{V}; \vec{V}) + \partial_\eta (\mu \dot{\eta} \vec{V}) \\ &\qquad - \nabla (p \partial_\eta \phi) + \partial_\eta (p \nabla \phi) = \vec{F} \end{aligned} \tag{4.2b}$$

$$\begin{aligned} &\partial_t(\mu w) + \nabla \cdot (\mu \vec{V} w) + \partial_\eta (\mu \dot{\eta} w) \\ &\qquad - g \left(\partial_\eta p - \mu \right) = F \end{aligned} \tag{4.2c}$$

$$\partial_t \phi + \vec{V} \cdot \nabla \phi + \dot{\eta} \partial_\eta \phi - g w = 0 \tag{4.2d}$$

In the above equations $\vec{V}$ only includes horizontal velocities, and ∇ operates on coordinate surfaces in the horizontal dimension. Geopotential is defined as $\phi = gz$, so that $\nabla \phi$ and $\partial_\eta \phi$ are surrogates for the Jacobian terms ∇z and $\partial_\eta z$.

Terms created by the change of coordinates may be evaluated analytically if the terrain function is differentiable and the Jacobian matrix is invertible. Instead it is often more practical to compute the metric terms numerically, as is the case in WRF. The Jacobian maps the physical topography onto a rectangular domain, but in WRF the terms are also affected by movement of the vertical coordinate during the time integration. Therefore, the Jacobian terms appearing in WRF must be evaluated numerically at each time step (making an analytical evaluation impractical). The terms are evaluated with an even-order finite difference scheme that is greater than or equal to the order of the advection scheme. Advection schemes range from 2nd to 6th order, therefore if a 3rd or 4th order advection scheme is used, the Jacobian terms are evaluated with a 4th order scheme. Turbulent diffusion and computational mixing are not used in the test case presented in this work.

When the immersed boundary method is used the coordinates are still changed into pressure coordinates, but the transformation to terrain-following coordinates is eliminated. Horizontal gradients of the coordinate are substantially reduced with IBM, but the magnitude of the gradients may not be exactly zero due to time variability of the grid arising from the transformation to pressure coordinates.

4.3 Idealized advection test

The idealized advection test presented in Schär et al. [2002] is used here to demonstrate the effects of truncation errors in the WRF model. In this section, truncation errors arising from both the finite-differencing scheme and the coordinate transformation are considered. It is shown that these errors are made to be negligible when the immersed boundary method is used.

For this test case, highly variable topography (with a maximum slope of 49 degrees) resides in a quiescent air mass, with a uniform horizontal flow aloft, as shown in figure 4.2. A shear layer in the velocity sounding persists without mixing due to the absence of

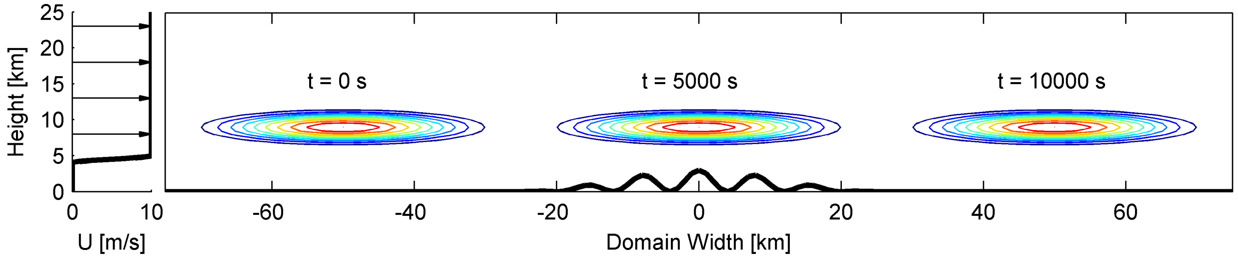


Figure 4.2. Set-up of the idealized advection test from Schär et al. [2002]. Topography is placed within a stagnant air mass, with uniform flow above. The analytic solution is shown for advection of a scalar cloud at three different times.

viscosity. The shear layer isolates the effects of the terrain from the flow aloft, so that when a scalar anomaly is introduced it advects over the terrain without distortion or diffusion. The analytical solution for the advection of a scalar cloud is presented in figure 4.2 at three different times. When terrain-following coordinates are used, the horizontal grid lines retain the signature of the topographic features. Discretization of the terrain-following coordinates leads to an additional truncation error which is a function of the Jacobian. These truncation errors cause distortion of the scalar as it advects through the domain as illustrated below.

4.3.1 Model set-up and initialization

In this test, the topography is specified as the product of two oscillatory functions. The first function has a large-scale wavelength of 50 km, and the second perturbation function has a wavelength of 8 km. The equation for the topography is given as 4.3, where $h_o = 3$ km, $a = 25$ km, and $\lambda = 8$ km.

$$h_x(x) = \begin{cases} h_o \cos^2(\frac{\pi x}{2a}) \cos^2(\frac{\pi x}{\lambda}) & \text{for } |x| \leq a \\ 0 & \text{for } |x| > a \end{cases} \tag{4.3}$$

Velocity, potential temperature, and water vapor mixing ratio are specified with a vertical sounding. Velocity is specified by equation 4.4, where $u_o = 10$ m s^{-1}, $z_1 = 4$ km, and $z_2 = 5$ km.

$$u(z) = \begin{cases} u_o & \text{for } z > z_2 \\ u_o \sin^2(\frac{\pi}{2}\frac{z-z_1}{z_2-z_1}) & \text{for } z_1 \leq z \leq z_2 \\ 0 & \text{for } z < z_1 \end{cases} \tag{4.4}$$

The grid, terrain height, and initial velocity profile are identical to those used in Schär et al. [2002]. Zängl modified the set-up by using a three-dimensional domain with different horizontal and vertical dimensions, shorter and wider topographic features that are a function of both x and y, and distributed the shear layer over 3.3 vertical kilometers.

The atmosphere is neutrally stable with a potential temperature of 288 K, although a stable atmosphere is considered in appendix A. Schär et al. did not include temperature effects, and Zängl considered a standard (stable) atmosphere. The analytical solution is independent of the background atmospheric stability, although in practice it makes a large difference in the numerical solution by modifying dynamic stability (as indicated by the bulk Richardson number). A dry atmosphere is considered here. Zängl uses moisture rather than a passive scalar in his tests.

The total domain size is (X, Y, Z) = (300 km, 2 km, 25 km) for the simulation with terrain-following coordinates. When the immersed boundary method is used, the domain is extended 1 km in the vertical dimension to (X, Y, Z) = (300 km, 2 km, 26 km). The vertical domain ranges from -1 km to 25 km, allowing for computational nodes below the zero terrain height. These extra nodes are used as forcing points in the immersed boundary method. The number of grid points in the terrain-following coordinate case is (n_x, n_y, n_z) = (301,3,51), and with the immersed boundary method it is (n_x, n_y, n_z) = (301,3,54). Horizontal resolution in the base case is $\Delta X = \Delta Y = 1$ km, and vertical resolution is $\Delta Z = 0.5$ km. Schär et al. [2002] use a 25 s time step; however, a smaller time step of 20 s is needed in WRF to achieve numerical stability.

The scalar cloud is defined by equation 4.5, where the maximum amplitude is $\varphi_o = 1$, the horizontal half width is $A_x = 25$ km, and the vertical half width is $A_z = 3$ km.

$$r = \left[\left(\frac{x - x_o}{A_x} \right)^2 + \left(\frac{z - z_o}{A_z} \right)^2 \right]^{1/2} \tag{4.5a}$$

$$\varphi(x, z) = \begin{cases} \varphi_o \cos^2(\frac{\pi r}{2}) & \text{for } r \leq 1 \\ 0 & \text{for } r > 1 \end{cases} \tag{4.5b}$$

The scalar is initialized at the location (X_o, Z_o) = (-50 km, 9 km). It is centered in the domain at $t = 5000$ s, and the center is located at (X, Z) = (50 km, 9 km) when the time integration ends at $t = 10000$ s.

4.3.2 Results using the default WRF settings

A comparison is made between two WRF simulations, the first with the native terrain-following coordinates, and the second with the newly implemented immersed boundary method. Default WRF options are used, and include a 3$^{\text{rd}}$ order Runge Kutta time stepping scheme, 5$^{\text{th}}$ order horizontal advection, and 3$^{\text{rd}}$ order vertical advection. The odd-order advection schemes are upwind-biased and diffusive. Default constants are used for filtering in time, and include a divergence damping coefficient $\gamma_d = 0.1$, external mode damping coefficient $\gamma_e = 0.01$, and acoustic time step off-centering of $\beta = 0.1$.

Figure 4.3 shows contours of u and w velocity at $t = 10000$ s, with results using terrain-following coordinates on top, and those using IBM below. In the analytical solution there

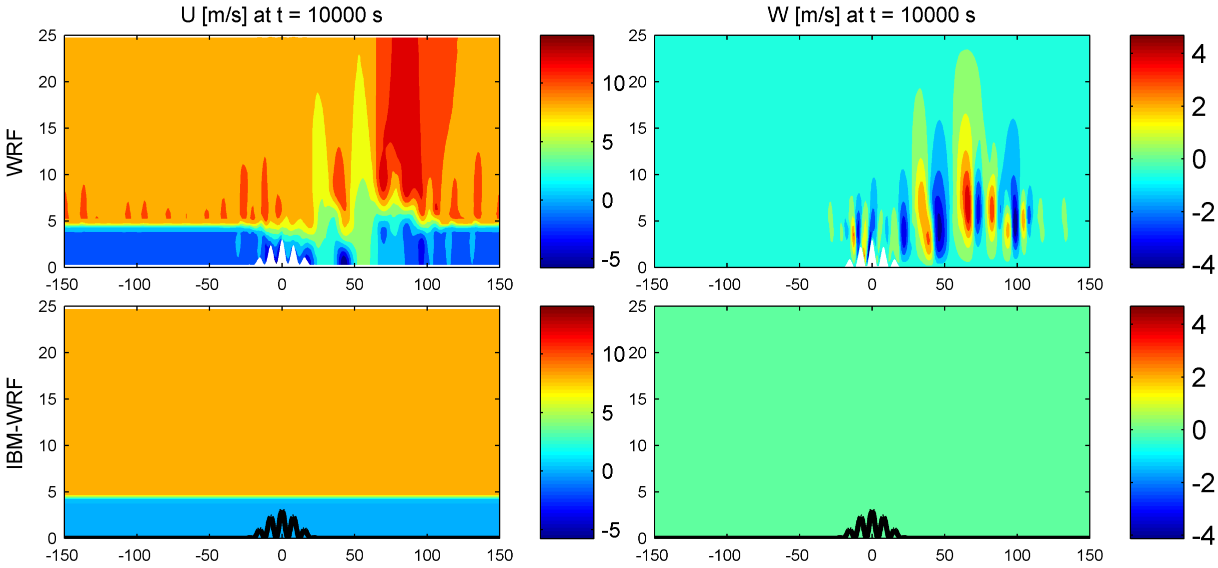


Figure 4.3. Contours of the u and w components of velocity in m s^{-1} for terrain-following coordinates (top) and the immersed boundary method (bottom) at t = 10000 s. Analytically, the velocity should equal the initial sounding throughout the duration of the simulation. Axes indicate domain size in km, and are not to scale.

is no interaction with the topography, and the velocity field is specified by equation 4.4 at all times. When terrain-following coordinates are used it is clear that the distortion of the grid makes it impossible to isolate the flow aloft from terrain effects. Waves, induced by errors in the coordinate transformation, form above the mountain range. Horizontal velocity should range from 0 to 10 m s^{-1}, and vertical velocity should remain zero. However, horizontal velocities of -5.8 to 14.1 m s^{-1} and vertical velocities greater than ±4 m s^{-1} are present. These errors are negligible in the IBM-WRF simulation. At the end of the simulation, horizontal velocity ranges between -0.04 and 10.08 m s^{-1}, and vertical velocity between -0.04 and 0.06 m s^{-1}.

While Schär et al. prescribed the velocity field, Zängl allowed the velocity to evolve. Zängl reported maximum vertical velocities of 0.19 m s^{-1} using the native MM5 coordinate. One reason why his results are much better than the WRF results using the native coordinate is his favorable set-up. As described in section 4.3.1 his topography was shorter and wider leading to less distortion of the grid, and his shear velocity layer spanned a larger distance, allowing additional resolution. The largest difference is that his atmosphere was stable (see appendix A for WRF simulations with a stable atmosphere), while ours is neutral, so that the growth of waves due to perturbations caused by numerical errors is effectively damped by the stable stratification. Zängl reported vertical velocities of 0.02 m s^{-1} using a modified coordinate that flattens quickly with height.

Snapshots of the scalar cloud are included in figure 4.4, along with the associated errors. Three different times (t = 0, 5000, and 10000 s) are depicted as the scalar advects from left

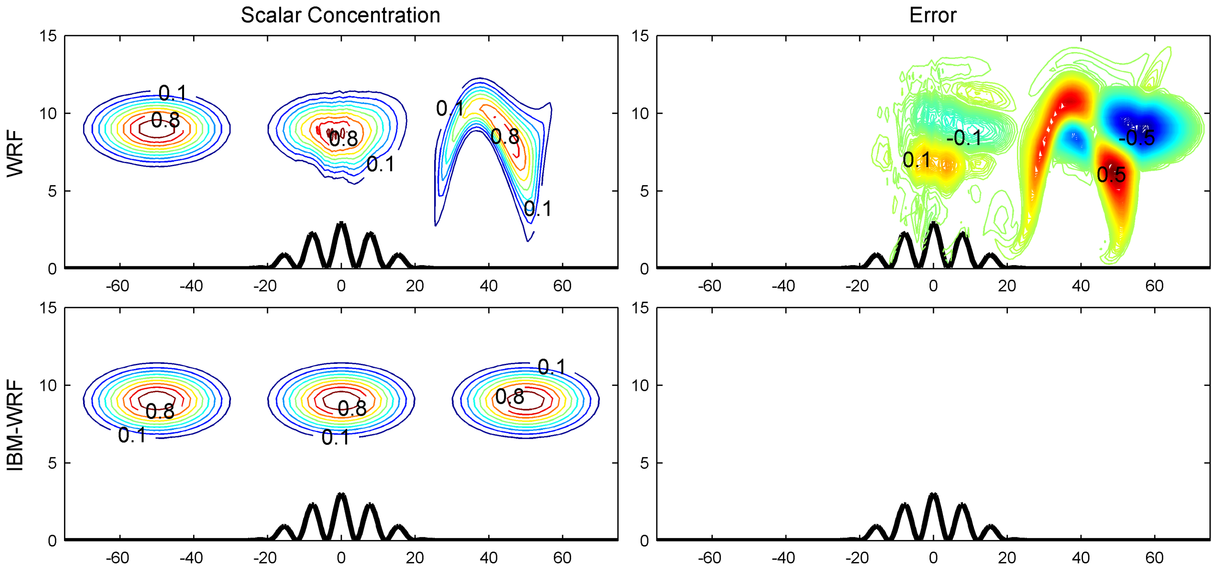


Figure 4.4. On the left, the scalar concentration is shown at $t = 0$, 5000, and 10000 s. Scalar units are non-dimensional with a range of 0 to 1. Contour intervals are in 0.1 increments. Error is shown on the right, and is calculated as the difference between the numerical and analytical solutions. Contour intervals are 0.01. The zero contour is suppressed. Axes indicate domain size in km, and are not to scale.

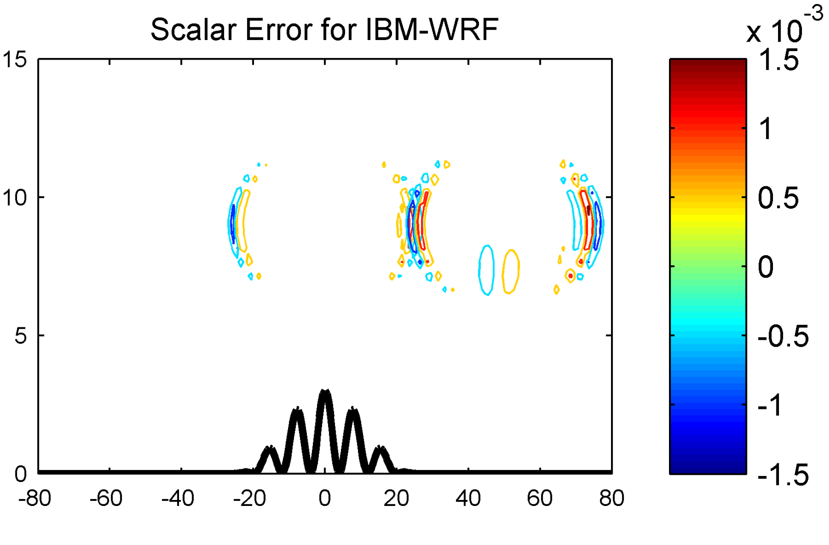


Figure 4.5. Error is shown for the immersed boundary method case at $t = 0$, 5000, and 10000 s. Contour intervals are 0.0005. The zero contour is suppressed. Axes indicate domain size in km, and are not to scale.

Table 4.1. Summary of errors at $t = 10000$ s for the WRF simulations with the default advection scheme and those presented in Schär et al. [2002] and Zängl [2003] (at t = 4 hours). Analytical values of φ_{min} and φ_{max} are 0 and 1. $\Delta\varphi$ is the difference between the numerical and analytical solutions.

Coordinate	Order of	φ		$\Delta\varphi$	
	Advection Scheme	min	max	min	max
WRF Sigma	h:5th, v:3rd	-0.039	0.856	-0.766	0.673
Schär et al. Sigma	1st	0.000	0.284	-0.700	0.213
	2nd	-0.168	0.953	-0.174	0.162
	4th	-0.058	1.001	-0.057	0.052
Zängl Sigma	2nd			-0.12	0.10
IBM-WRF	h:5th, v:3rd	-0.002	0.992	-0.002	0.002
Schär et al. No Topography	1st	0.000	0.762	-0.220	0.141
	2nd	-0.023	0.985	-0.023	0.021
	4th	-0.002	0.984	-0.002	0.002
Zängl No Topography	2nd			-0.02	0.02

to right in the domain. Significant distortion of the scalar anomaly occurs as it advects over the terrain features in the simulation with sigma coordinates. At the last time the shape of the cloud is not only distorted, but the center has advected 4.5 km less than in the analytical solution. Error is calculated as the difference between the numerical and analytical solution, and is shown with contour intervals of 0.01. At the last time shown, error ranges between -0.766 and 0.673. These errors are on the order of the analytical scalar concentration, which ranges from 0 to 1, indicating large errors arising from the use of sigma coordinates. In the simulation using the immersed boundary method, distortion of the cloud is eliminated. No contours appear in the IBM-WRF error plot, because the error is less than the threshold of the first contour (0.01). Errors in the IBM-WRF simulation are included in figure 4.5 with appropriate contour levels. In IBM-WRF the deviation from the analytical solution ranges from -0.002 to 0.002. Contour intervals are 0.0005. The results of these simulations indicate that the truncation error is dominated by the term arising from the transformation to terrain-following coordinates, and errors from the pressure coordinate transformation and the finite differencing scheme are negligible in comparison, as illustrated in the IBM-WRF solution.

A comparison of the WRF and IBM-WRF results and those presented in Schär et al. [2002] and Zängl [2003] is included in table 4.1. Schär et al. published results for simulations with 1st, 2nd, and 4th order advection schemes (among others which are not included here). For reference, both papers also included a set of simulations with no topography. Error in the reference simulations is exclusively caused by the finite differencing schemes. As expected, higher order advection schemes decrease error for both sigma coordinates and the reference cases of Schär et al.. The IBM-WRF simulation (with topography) performs as well or better than the 4th order reference case with no topography. This is a logical result,

Table 4.2. Summary of errors at $t = 10000$ s for the WRF and IBM-WRF simulations with advection schemes of increasing order. Analytical values of φ_{min} and φ_{max} are 0 and 1. $\Delta\varphi$ is the difference between the numerical and analytical solutions.

Coordinate	Order of Advection Scheme	φ		$\Delta\varphi$	
		min	max	min	max
WRF Sigma	3rd	-0.028	0.735	-0.746	0.480
	4th	-0.373	0.885	-1.073	0.867
	5th	-0.064	0.893	-0.857	0.852
	6th	-0.439	1.325	-1.050	0.820
IBM-WRF	3rd	-0.004	0.991	-0.004	0.005
	4th	-0.003	0.992	-0.003	0.003
	5th	-0.002	0.992	-0.002	0.002
	6th	-0.002	0.992	-0.002	0.001

as the immersed boundary method alleviates the need for a coordinate transformation and the simulation is of a similar order.

WRF with sigma coordinates produces more errors than originally expected. Despite the higher order scheme, WRF does not perform as well as the 2nd order sigma cases of Schär et al. and Zängl. The main difference between the simulations is that the wind is prescribed in Schär et al., but allowed to evolve in the WRF simulations. Prescribing the velocity field eliminates the explicit vertical advection of the scalar field that occurs in WRF due to errors in the vertical velocity component. In Zängl, instability from perturbations (non-physical errors due to the coordinate) are suppressed by intentionally prescribing a stable atmosphere. This is not the case in our simulations with WRF, where the atmosphere is neutral. The large differences in the models and set-ups probably account for the larger errors in the WRF simulations. It is important to note then that IBM-WRF correctly evaluates the flow field and transport of the scalar cloud. Additionally, both Schär et al. and Zängl demonstrate error reductions with their modified terrain-following coordinate systems. These results are not included here.

Schär et al. cite several causes of error in the idealized advection tests, first reasoning that, "Schemes with implicit diffusion suffer particularly large coordinate transformation errors. Diffusion spreads out the solution in computational space, rapidly broadens the initial anomaly, and thereby makes the scheme more susceptible to coordinate transformations." In a grid refinement study, Schär et al. found that the solution was extremely sensitive to horizontal resolution, and large gains in accuracy could be achieved through increased horizontal resolution. The solution was shown to be insensitive to vertical resolution. Additionally, as shown in table 4.1, Schär et al. found that increasing the order of the advection scheme had a beneficial impact on the quality of the solution. Accuracy was largely gained from an increase in the order of the horizontal scheme, whereas the vertical scheme had little effect. These possible types of error reduction are investigated in the following section.

4.4 Analysis of truncation errors

Schär et al. [2002] carried out a theoretical analysis of truncation errors in a generalized transformed coordinate. The analysis considers the transformed one-dimensional advection equation (4.6).

$$\frac{\partial \rho}{\partial t} + J\frac{\partial(\rho u)}{\partial \bar{x}} = 0 \tag{4.6}$$

Here $J = \partial\bar{x}/\partial x$ is the Jacobian of the transformation and the overbar indicates the transformed coordinate.

After applying a Taylor series expansion, the error term is derived in computational space. Using the transformation back to physical space ($\Delta\bar{x} = J\Delta x$), the total truncation error is given by 4.7.

$$E = \frac{\Delta x}{2}J\frac{\partial}{\partial x}\left(uJ^{-1}\frac{\partial \rho}{\partial x}\right) + O(\Delta x^2) \tag{4.7}$$

Applying the chain rule, the truncation error may be split into two parts. E_{fd} is attributed to the finite differencing scheme, and E_t to the coordinate transformation. Errors are given in 4.8 for a first order upwind scheme.

$$E_{fd} = \frac{\Delta x}{2}\frac{\partial}{\partial x}\left(u\frac{\partial \rho}{\partial x}\right) + O(\Delta x^2) \tag{4.8a}$$

$$E_t = -\frac{\Delta x}{2}u\frac{\partial \rho}{\partial x}J^{-1}\frac{\partial J}{\partial x} + O(\Delta x^2) \tag{4.8b}$$

It is seen here that the leading term for the error due to each cause is of the same order of magnitude O(Δx). Large Jacobian terms as well as large gradients in the Jacobian lead to significant increases in the transformation truncation error. In the limit of the Jacobian approaching zero, the truncation error reduces to the theoretical form from the finite differencing scheme.

In this section the effects of the advection scheme, horizontal and vertical resolution, and terrain slope are studied. It is shown that the error produced by the finite differencing scheme is negligible in comparison to the error caused by the coordinate transformation used in WRF.

4.4.1 Effect of advection scheme

Schär et al. noted that a second order (even) advection scheme produced better results than a first order (odd) scheme which included numerical dissipation. As the default scheme in WRF is an odd order, we tested the next highest even order scheme for improved accuracy. Truncation errors for an odd order advection scheme include those from the next higher-order centered scheme, plus an upwind term that is dissipative. The dispersive terms in the truncation error of an odd order scheme and the next highest even order scheme are identical.

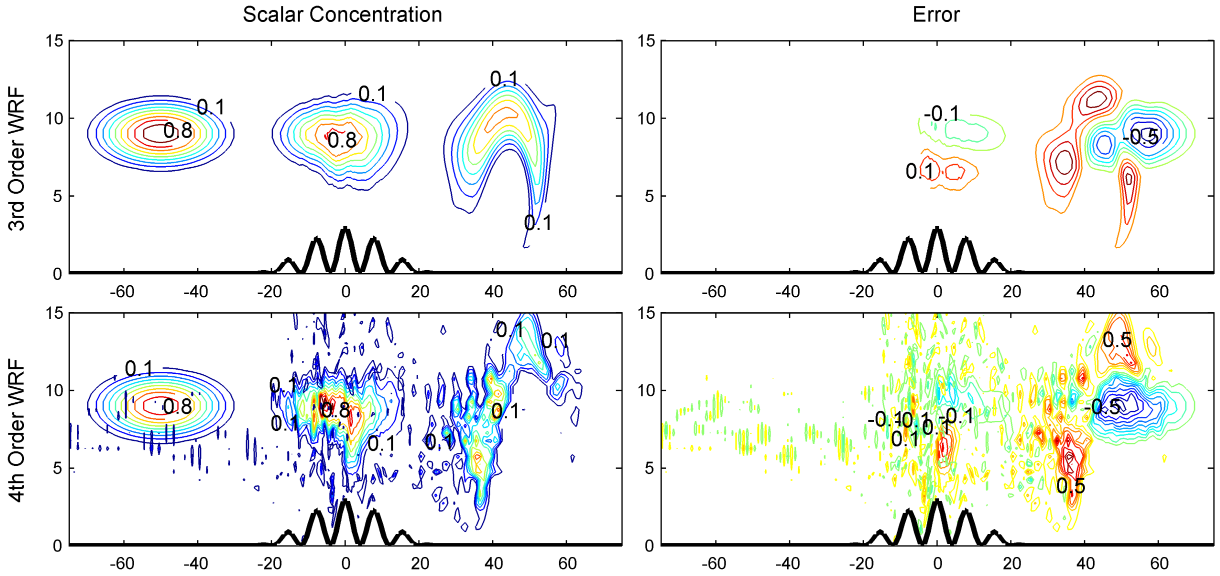


Figure 4.6. As in figure 4.4, but with error contour increments of 0.1. 3rd and 4th order advection schemes are used in a WRF simulation with terrain-following coordinates.

The even order scheme (in WRF fourth or sixth order) should advect a well-resolved smooth Gaussian shaped scalar plume without dissipation. The effects of the choice of advection scheme are presented here, where each available option is tested in WRF. For comparison, simulations are included with both terrain-following coordinates and the immersed boundary method.

Table 4.2 includes the error for each advection scheme with each type of coordinate (sigma and IBM). It can be seen here that in WRF with sigma coordinates, the odd order advection schemes produce less error than even order schemes, although the accuracy of both types is poor. Additionally, it is not clearly beneficial to increase the order of the advection scheme when the native WRF coordinate is used in this test case example. Both of these results are contrary to the findings of Schär et al..

In contrast to the large errors produced with sigma coordinates, errors in the IBM-WRF simulations are extremely small. With IBM it is clear that the solution benefits from increasing the order of the advection scheme. A preference is not shown for even or odd advection schemes in the simulations using IBM. For comparison, simulations in WRF with no topography yielded almost identical results to those with IBM. As found in the results with default settings, truncation error due to the finite differencing scheme is negligible in comparison to that of the coordinate transformation.

Figure 4.6 gives insight into the behavior of odd and even order advection schemes when used with the sigma coordinate in WRF. This figure depicts the scalar anomaly at three instances in time, along with error contours for each of the instances. The 3rd order advection scheme is shown in the top figures, with a 4th order scheme used in the figures below. The

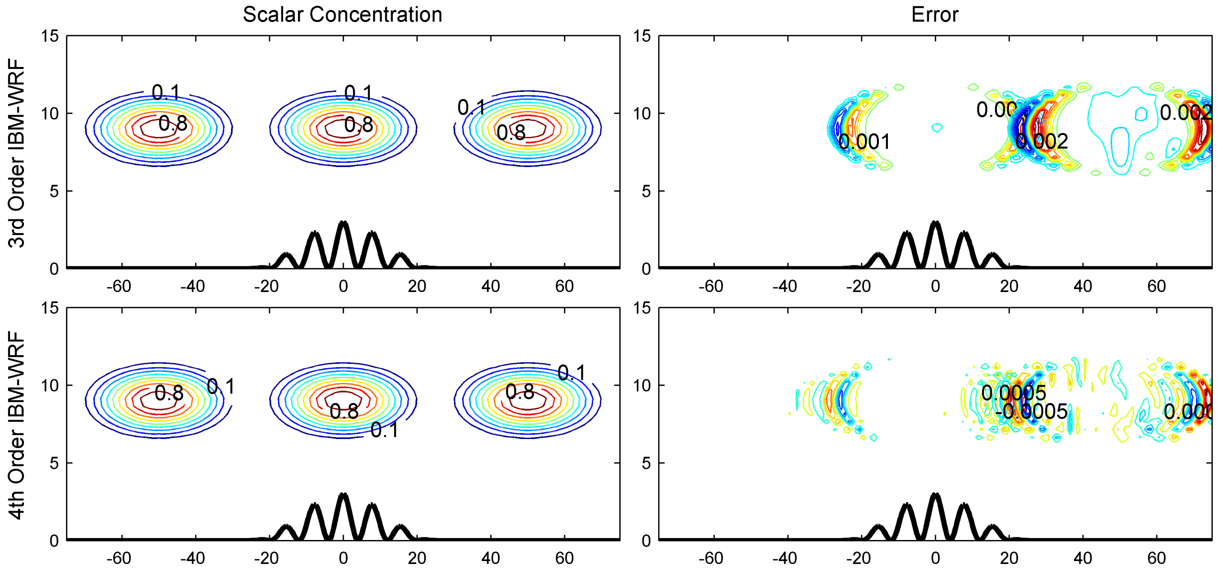


Figure 4.7. As in figure 4.4, but with error contour increments of 0.0005. 3rd and 4th order advection schemes are used in a WRF simulation with the immersed boundary method.

odd order scheme is upwind biased and thus dissipative. As the scalar advects over the peaks in the terrain, significant distortion of the cloud occurs. The numerical diffusion seems to be beneficial in the odd order scheme, in that it counteracts other errors (from the grid transformation and dispersive truncation error terms) and allows the cloud to remain a cohesive mass. In the even order scheme the scalar cloud disperses as it advects over the terrain, and variability is not smoothed out. This leads to large errors in the location of the cloud, as well as significant magnitudes of negative scalar concentration.

Figure 4.7 provides the same information for simulations where IBM is used. In this case the odd and even order advection schemes produce accurate results with similar error magnitudes. While one scheme is diffusive and the other is dispersive, the effect of these errors are not obvious in the solution. The results illustrate that a vast improvement in the solution can be achieved using IBM to eliminate the need for the terrain-following change of coordinate.

4.4.2 Effect of spatial resolution and grid aspect ratio

Schär et al. found that significant improvements in accuracy could be achieved through increasing horizontal resolution, while sensitivity to vertical resolution was limited. In fact, it was noted that increasing the resolution from $\Delta x = 1000$ m to $\Delta x = 500$ m reduced the error from 21% to 4%, while decreasing the resolution to $\Delta x = 1500$ m led to 78% error.

The effects of spatial resolution are examined in WRF using the default advection scheme,

Table 4.3. Summary of errors at $t = 10000$ s for WRF simulations with sigma coordinates at various spatial resolutions. The default advection scheme is used (5th order horizontal and 3rd order vertical).

Δx (m)	Δz (m)	φ		$\Delta\varphi$	
		min	max	min	max
1000	500	-0.039	0.856	-0.766	0.673
1000	250	-0.128	0.897	-1.040	0.897
1500	500	-0.071	0.699	-0.832	0.588
500	500	-0.030	0.300	-0.983	0.300

which is 5th order in the horizontal and 3rd order in the vertical. The results of the base case along with three additional simulations are presented in table 4.3. First, vertical resolution was increased from $\Delta z = 500$ m to $\Delta z = 250$ m. In the next two simulations, horizontal resolutions of $\Delta x = 1500$ m and $\Delta x = 1000$ m are used. Unlike in the Schär et al. results, it is not clearly beneficial to increase the resolution. In fact, the solution deteriorated more by increasing the horizontal resolution from $\Delta x = 1000$m to $\Delta x = 500$ m, than it deteriorated from decreasing the resolution to $\Delta x = 1500$ m. The solution also deteriorated with increased vertical resolution. In conclusion, while increased horizontal resolution should reduce errors from the coordinate transformation by improving the accuracy of horizontal derivatives, the solution is unstable to perturbations so that decreasing (or increasing) the magnitude of the numerical error does not significantly effect the accuracy of the overall solution. Furthermore, increased resolution supports higher frequency waves, and may further deteriorate accuracy. In contrast, IBM-WRF produces an accurate solution without the need for additional resolution.

4.4.3 Effect of coordinate transformation

The effect of the magnitude of the Jacobian on the accuracy of the solution is evaluated with a series of simulations with decreasing terrain height. In these simulations the terrain is still defined by equation 4.3, but the maximum amplitude of the terrain h_o ranges from 0 to 3000 m. The half width of the terrain is held constant. This results in a maximum terrain slope that ranges from 0 degrees to 49 degrees, when a 3000 m peak height is used. Analytically the solution for the scalar cloud should be independent of the terrain because it is isolated by the inviscid shear layer. Therefore, decreasing the terrain height affects the solution aloft by decreasing the grid distortion aloft caused by the terrain-following coordinate.

Results are presented in figure 4.8 for simulations with 2nd and 3rd order advection schemes. In this plot error is defined by equation 4.9.

$$E = \max |\varphi_{numerical} - \varphi_{analytical}| \tag{4.9}$$

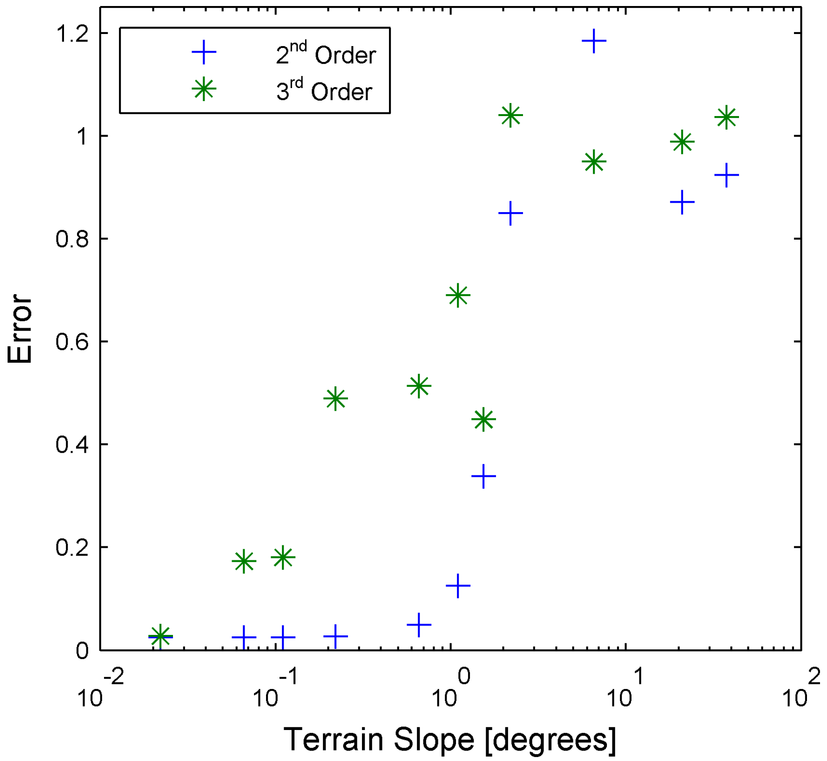


Figure 4.8. Error is plotted as a function of terrain slope for simulations with 2^{nd} and 3^{rd} order advection schemes.

It can be seen in this plot that error grows quickly with increasing terrain slope. In fact, a slope of just one degree will cause error of over 10% with the 2$^{\mathrm{nd}}$ order scheme, and almost 70% with the 3$^{\mathrm{rd}}$ order scheme. Beyond a one degree slope the error is very large regardless of the advection scheme.

Large errors are produced when the grid is not orthogonal in the region of the shear velocity layer. A very small coordinate slope in this region will lead to large truncation errors in the velocity field. This induces a wave in the flow, as was shown in figure 4.3, with a wavelength on the order of that specified in the equation for the terrain. The point that should be made here is that even with shallow terrain slopes the truncation error can have devastating effects on the accuracy of the solution when large gradients and discontinuities exist in the flow field.

4.5 Conclusions

The scalar advection test case is sensitive to perturbations, but nonetheless useful in illustrating sensitivity to coordinate errors. For example, orographic precipitation in mountainous terrain is sensitive to model dynamics. It is shown in Zängl [2004] that calculating diffusion along sloping coordinate surfaces, rather than truly horizontal, increases the average precipitation by 35% for a heavy-precipitation case over alpine terrain.

It has been demonstrated in this chapter that the immersed boundary method is an effective tool not only for representing the complex terrain boundary, but also for eliminating errors far from the boundary caused by the terrain-following coordinate transformation. An idealized advection test was used to illustrate the effectiveness of the immersed boundary method in eliminating the errors associated with terrain-following coordinates. Error in the simulations with sigma coordinates could not be reduced to the level of error in the IBM simulations. Attempts were made to decrease error in simulations with the sigma coordinate by increasing the order of the finite difference scheme, refining the grid resolution, and even reducing the terrain slope. This test case is sensitive to stratification, although the analytical solution is independent of stratification. Imposing a stable atmosphere can reduce error, but use of the immersed boundary method eliminated errors due to the terrain-following coordinate under all stability conditions. Ultimately, the IBM-WRF simulations reproduced the analytical solution with the highest accuracy.

Chapter 5

Implementation of the immersed boundary method[*]

This chapter describes a new implemetation of the immersed boundary method which facilitates explicit resolution of complex terrain within the Weather Research and Forecasting (WRF) model. Simulations with this approach may be performed in three dimensions; however, the method described in this chapter is limited to two-dimensional terrain. Validation cases are presented in chapter 6. These cases show near perfect agreement when flow over shallow terrain using the immersed boundary method is compared to results using a terrain-following grid. This method is extended to handle three-dimensional terrain in chapter 7.

5.1 Introduction

Most mesoscale numerical weather prediction (NWP) models use terrain-following coordinates, which accommodate complex terrain by transforming the physical domain onto a Cartesian grid. This formulation simplifies the application of lower boundary conditions by aligning the lowest coordinate with the topography. Coordinate lines gradually become

[*] This chapter is a reproduction (with minor modifications) of a portion of the paper "An Immersed Boundary Method for the Weather Research and Forecasting Model" by Katherine A. Lundquist (the principal author), Fotini Katopodes Chow, and Julie K. Lundquist, published in Monthly Weather Review, March 2010, Volume 138(3), pages 796-817 [Lundquist et al., 2010], ©Copyright 2010 American Meteorological Society. The full copyright notice is included as appendix C.

smoother and flatter with distance from the ground, but retain a signature of the underlying surface shape throughout the entire height of the domain. Metric terms from the terrain-following coordinate transformation are introduced into the governing equations, and when discretized these terms produce additional numerical errors [Janjić, 1977, 1989; Klemp et al., 2003; Schär et al., 2002; Zängl, 2002, 2003, 2004; Zängl et al., 2004]. As mesoscale models are increasingly being used for high resolution flows over complex terrain, these coordinate transformation errors can significantly degrade the quality of the numerical solution.

Several methods have been proposed to increase the fidelity of simulations with complex terrain, ranging from methods that reduce errors arising from coordinate transformations to methods which eliminate the need for coordinate transformations. Mahrer [1984] noted that the standard computational stencil used to calculate horizontal gradients is inappropriate when the vertical grid spacing is less than the elevation change over the horizontal dimension of the cell. In this case, the accuracy of horizontal gradients can be improved by modifying the stencil to include nodes which more closely follow a Cartesian grid. Schär et al. [2002] proposed a modified sigma coordinate in which grid distortion due to small scale terrain features decays with height more rapidly than distortion caused by large scale terrain features. The modified coordinate flattens quickly with height and improves the accuracy of the solution by reducing grid distortion aloft. As terrain slopes approach the vertical limit, coordinate transformation errors grow, making reductions of these errors increasingly difficult.

In environments with very steep slopes, such as urban or mountainous environments, it is desirable to completely eliminate the coordinate transformation. One approach is to represent topography by fitting it to a Cartesian grid, thereby creating a step approximation of the boundary. Large errors have been documented when a zeroth order terrain approximation is used to model topographically induced circulations [Fast, 2003], while higher order boundary representations alleviate these errors by eliminating the sharp corners [Adcroft et al., 1997]. Structured or unstructured body-fitted grids are often employed for complex geometries, as these methods are capable of higher order boundary representations. Hanna et al. [2006] present simulations of downtown Manhattan with five different urban computational fluid dynamics (CFD) models, each using conforming grids. While imposing boundary conditions on these grids is straightforward, there are drawbacks. Body-fitted coordinates require time consuming manual manipulation to ensure that the grid conforms to the boundaries while minimizing grid skewness. Unstructured grids produce an irregular data structure which increases computational cost. In addition to these gridding issues, CFD models are traditionally forced at lateral boundaries with idealized flow, neglecting dynamic forcing due to synoptic scale weather patterns. Furthermore, CFD codes do not generally include options for representing atmospheric processes such as surface fluxes of heat and moisture.

In this chapter, we introduce an immersed boundary method (IBM) which combines the favorable properties of a NWP code with the ability to handle complex terrain. The IBM enables flows with complex terrain to be simulated on non-conforming grids, while retaining the efficiency of a structured solver and the features of a NWP code. Terrain is included by modifying the treatment of nodes near the immersed boundary to impose the effects of the surface. Numerous methods employ this technique, and are known by a variety of names including immersed boundary, embedded boundary, fictitious domain, penalty, and Cartesian grid methods. The methods differ in the treatment of the nodes near the immersed or embedded boundary. The review articles by Iaccarino and Verzicco [2003] and Mittal and Iaccarino [2005] provide a comprehensive overview of existing methods for non-conforming grids, while focusing on variants of the immersed boundary method.

The IBM presented here represents a rigid interface at the boundary using direct forcing, as first suggested by Mohd-Yusof [1997]. This IBM is uniquely formulated to handle compressible viscous flows and work with the native isobaric terrain-following coordinate in the Weather Research and Forecasting (WRF) model. The newly developed algorithm accommodates movement in the vertical pressure coordinate while remaining well conditioned, and avoids numerical instabilities noted in other immersed and embedded boundary approaches [Saiki and Biringen, 1996; Tseng and Ferziger, 2003; Kirkpatrick et al., 2003]. Details of the WRF model and the IBM implementation are provided in sections 5.2 and 5.3. The IBM is capable of enforcing either Dirichlet or Neumann boundary conditions. A comparison of the native WRF and IBM-WRF boundary conditions is made in section 5.4, along with the details of a new no-slip option for terrain-following coordinates.

An additional feature distinguishing the IBM described here is that it includes coupling to a land-surface model which provides realistic surface forcing. IBM has been used for atmospheric flows in CFD codes [Tseng et al., 2006; Smolarkiewicz et al., 2007; Shi et al., 2008], however, these implementations do not handle generalized geometries and to the authors' knowledge this is the first IBM which interfaces with atmospheric parameterizations. Modifications required to include the atmospheric physics options are discussed in section 5.5. By retaining the native isobaric coordinate and coupling the IBM to the land-surface model, NWP features such as grid nesting and atmospheric physics function seamlessly with the IBM. Domains where IBM is used to explicitly resolve complex terrain can be nested into larger mesoscale domains which use the terrain-following coordinate.

5.2 Formulation of the numerical solver

This work uses the Advanced Research WRF (ARW) dynamics solver, which is described in detail by Skamarock et al. [2007]. Background information relevant to the IBM implementation is presented in this section, including descriptions of the sigma WRF and IBM-WRF coordinates, governing equations, and discretization schemes.

5.2.1 Coordinate definition

The WRF model is based on a pressure coordinate, which Laprise defines in terms of the dry hydrostatic pressure P_{hs}. The coordinate is defined such that it is zero at the top

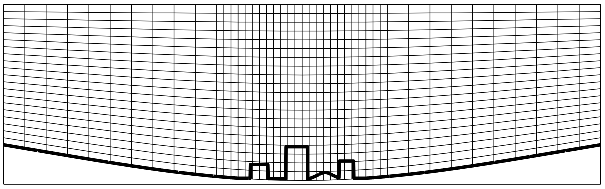


Figure 5.1. Nested domains can be used to resolve both coarse and fine scale terrain. Terrain-following coordinates conform to the gently sloping valley, while fine scale terrain (in this case urban features) is resolved with the immersed boundary method.

of the computational domain and unity at the terrain surface. This yields the coordinate definition $\eta = (P_{hs} - P_{hs\_top})/\mu$, where the column mass per unit area of the fluid is $\mu(x, y) = P_{hs\_surface} - P_{hs\_top}$. Note that we refer to the coordinate as 'eta' to be consistent with Laprise and the WRF documentation [Skamarock et al., 2007], though the coordinate is more commonly referred to in the literature with the variable sigma (σ).

With our implementation of the immersed boundary method, the sigma coordinate is retained. The grid is transformed to align with gently sloping background topography, but does not conform to terrain which is explicitly resolved by the IBM. Topography is defined by superposition of small and large scale features in domains using IBM. This concept is illustrated in figure 5.1, where urban scale terrain is centered in a shallow valley. Pressure-based terrain-following coordinates are used to resolve the valley features in an outer, coarser grid domain. In a finer, nested domain, the valley floor is defined by the sigma coordinate, while the buildings are represented with the IBM.

IBM is often used to eliminate coordinate transformations; however, in the case of flows within complex terrain it is beneficial to use a hybrid methodology that combines the use of IBM with a transformed background coordinate. As the background sigma coordinate conforms only to gently sloping terrain, grid skewness and the associated errors are largely eliminated by using this IBM, although metric terms appear in the discretization for both sigma WRF and IBM-WRF. Use of a hybrid coordinate facilitates grid nesting and increases flexibility of the model. Hybrid techniques for using IBM on a curvilinear grid have been pursued in aeronautical [Ghias et al., 2007] and cardiovascular [Ge and Sotiropoulos, 2007] applications as well. A difference here is that additional complexities arise in the treatment of nodes near the immersed boundary due to the time-dependent pressure coordinate.

5.2.2 Governing equations

This work uses the Advanced Research WRF (ARW) dynamics solver, which is a conservative finite-difference model that solves the non-hydrostatic compressible Navier-Stokes equations [Skamarock et al., 2007]. The moist Euler equations are transformed into the isobaric terrain-following coordinate η, while additional terms such as diffusion, Coriolis, and parameterized physics (represented by F) are computed in physical space. A velocity $\dot{\eta}$, defined as the contravariant velocity of the vertical coordinate, is introduced in the coordinate transformation, necessitating the solution of an additional equation. Perturbation variables are introduced to reduce numerical errors; the perturbations are defined as the deviation from a time invariant hydrostatically balanced reference state. Pressure p, specific volume (moist α_m and dry α_d), geopotential ϕ, and dry column mass μ_d are cast as mean and perturbation values as $\varphi = \bar{\varphi} + \varphi'$, where φ represents a generic variable and the overbar indicates the hydrostatic base state. After substitution into the momentum equations, the hydrostatically balanced terms cancel out. The transformed equations are given in equation (5.1).

$$\partial_t \mu_d' + \nabla_\eta \cdot (\mu_d \mathbf{V}) = 0 \tag{5.1a}$$

$$\begin{aligned} \partial_t(\mu_d \mathbf{V}_H) + \nabla_\eta \cdot (\mu_d \mathbf{V}_H \otimes \mathbf{V}) + \mu_d(\alpha_m \nabla_\eta p' + \alpha_m' \nabla_\eta \bar{p}) & \\ + \frac{\alpha_m}{\alpha_d}\left(\mu_d \nabla_\eta \phi' + (\nabla_\eta \phi)(\partial_\eta p' - \mu_d')\right) = \mathbf{F} & \end{aligned} \tag{5.1b}$$

$$\partial_t(\mu_d w) + \nabla_\eta \cdot (\mu_d \mathbf{V} w) - g\left(\frac{\alpha_m}{\alpha_d}\partial_\eta p' + \frac{\alpha_m - \alpha_d}{\alpha_d}\bar{\mu_d} - \mu_d'\right) = F \tag{5.1c}$$

$$\partial_t \phi' + \mathbf{V} \cdot \nabla_\eta \phi - gw = 0 \tag{5.1d}$$

In the above equations the velocity vector is $\mathbf{V} = (u, v, \dot{\eta})$, $\mathbf{V}_H$ includes the horizontal velocities, and $\nabla_\eta = (\partial_x, \partial_y, \partial_\eta)$ operates on coordinate surfaces. Geopotential is defined as $\phi = gz$, so that $\nabla_\eta \phi$ is a substitute for the Jacobian term $\nabla_\eta z$.

In addition to conservation of mass and momentum, a conservation equation (5.2) is solved for additional scalar quantities, such as potential temperature θ, water vapor q_v, ice q_i, and passive scalars.

$$\partial_t(\mu_d \varphi) + \nabla_\eta \cdot (\mu_d \mathbf{V} \varphi) = F_\varphi \tag{5.2}$$

Pressure is then diagnosed from the equation of state below, where γ_d is the ratio of heat capacities of dry air C_p/C_v, p_o is the surface pressure, and R_d is the universal gas constant.

$$p = p_o \left(\frac{R_d \theta}{p_o \alpha_d}\right)^{\gamma_d} \tag{5.3}$$

5.2.3 Discretization schemes

The governing equations are discretized on an Arakawa-C staggered grid. Uniform grid spacing is used in the horizontal directions, and the grid may be stretched in the vertical direction. Second through sixth order finite-difference schemes are available for discretization of the advective terms. For the simulations presented here the default finite-difference scheme for advection is used, which is fifth order for horizontal derivatives and third order for vertical derivatives. Diffusive terms are calculated with a second order scheme.

A conservative split-explicit time integration scheme handles the full range of frequencies admitted by the compressible Navier-Stokes equations [Wicker and Skamarock, 2002;

Klemp et al., 2007]. In this scheme a third order explicit Runge-Kutta method is used for time advancement of meterologically significant low frequency physical modes, while a smaller time step is needed to account for the higher frequency modes such as acoustic waves. Variables from the Navier-Stokes equations and the prognostic equation for potential temperature are split into the value from the most recent Runge-Kutta time step t and a perturbation on the acoustic time step τ, so that $\varphi = \varphi^t + \varphi^\tau$. Horizontally propagating acoustic modes are integrated using an explicit forward-backward scheme, and vertically propagating acoustic modes and buoyancy oscillations are treated implicitly.

5.3 Treatment at the immersed boundary

IBM is used to represent the effects of boundaries on a non-conforming structured grid. Boundary conditions are imposed with the addition of a body force term F_B in the conservation equations for momentum and scalars (5.4). The body force term takes a zero value away from the boundaries, but modifies the governing equations in the vicinity of the boundary.

$$\partial_t \mathbf{V} + \mathbf{V} \cdot \nabla \mathbf{V} = -\alpha \nabla p + \nu \nabla^2 \mathbf{V} + \mathbf{g} + \mathbf{F}_B \tag{5.4a}$$

$$\partial_t \varphi + \mathbf{V} \cdot \nabla \varphi = \nu_t \nabla^2 \varphi + F_\varphi + F_B \tag{5.4b}$$

The exact details of this modification have varied among researchers since IBM was introduced by Peskin in 1972. Generally IBM methods include a determination of the forcing term, and an interpolation scheme to reconstruct the boundary condition at a surface that is not coincident with computational nodes.

5.3.1 Flow reconstruction at the immersed boundary

The IBM used in this work falls into a category commonly referred to as discrete or direct forcing which first appeared in the work of Mohd-Yusof [1997]. With this method the velocity or scalar value is modified at forcing points near the boundary to enforce the boundary condition, eliminating the need for explicit calculation of the body force term in the numerical algorithm. The method is especially adept at handling rigid boundaries, and produces a sharp representation of the fluid-solid interface. Within the direct forcing class of methods, we have adopted a finite-difference approach where forcing is applied at ghost cells located in the solid domain. This method has several advantages, most notably the straightforward extension from two to three dimensions. One drawback of this method is that it is not strictly conservative at cut cells. Despite errors in local conservation, successful results using this method have been reported extensively in the literature [Mittal and Iaccarino, 2005; Ghias et al., 2007].

In the case that the boundary is coincident with computational nodes, the boundary condition can be imposed by assigning it at the coincident node. However when the boundary passes through the grid in an arbitrary manner, the grid points are not generally aligned with the boundary. This is always the case with staggered grids, such as the Arakawa-C grid used in WRF. An interpolation method must be used to determine the forcing needed at actual computational nodes. The first step in reconstructing the flow around a boundary is to specify the terrain independently of the grid. For implementation into the WRF model we have allowed specification of terrain height at twice the resolution of the horizontal grid. Terrain elevation is specified as a function of x and y, which is compatible with the typical format of raw lidar and digital elevation data.

The next step is to identify cells which are cut by the immersed boundary. With a staggered grid, cut cells must be determined for each flow variable that will have a boundary condition imposed. Each node, located at the cell center and on each face of the staggered grid, is marked as a solid or fluid node. Cells containing both types of nodes are defined as cut cells. Flow variables can be reconstructed at fluid nodes as in Fadlun et al. [2000], Gilmanov et al. [2003], Balaras [2004], Choi et al. [2007], or at solid nodes as in Gao et al. [2007], Ghias et al. [2007], Mittal et al. [2008]. In this work flow variables are reconstructed at solid nodes, a technique known as the ghost cell method. Ghost points are identified as the layer of nodes belonging to cut cells that are within the solid region of the domain, as shown in figure 5.2.

The value of the variable at the ghost cell which will enforce either a Dirichlet (5.5a) or Neumann (5.5b) boundary condition at the immersed surface Ω must be computed.

$$\varphi = \varphi_\Omega \tag{5.5a}$$

$$\mathbf{n} \cdot \nabla\varphi = \left.\frac{\partial\varphi}{\partial\mathbf{n}}\right|_\Omega \tag{5.5b}$$

Several different interpolation methods have been employed by researchers for the purpose of making this calculation, ranging from linear interpolation to inverse distance weighting schemes [Iaccarino and Verzicco, 2003]. For the pressure-based compressible flow equations in WRF and the associated scalar equations, we have developed a unique bilinear reconstruction scheme appropriate for two-dimensional terrain. Three-dimensional terrain is included by extending the method to use a trilinear interpolation scheme, as researchers have also done in CFD codes [Mittal et al., 2008].

The bilinear interpolation method used in this work is illustrated in figure 5.2. First, the location of the ghost point is reflected across the boundary in the surface normal direction, and this is labeled an image point. When a Dirichlet boundary condition is used, the ghost point value is related to the image point value with $\varphi_G = 2\varphi_\Omega - \varphi_I$. For a Neumann boundary condition the relationship is $\varphi_G = \varphi_I - \overline{GI}\, \frac{\partial\varphi}{\partial\mathbf{n}}\big|_\Omega$, where $\overline{GI}$ is the distance between the ghost and image points.

The value of the image point is calculated by using interpolation over the shaded region in figure 5.2, where the interpolant is:

$$\varphi = c_1 + c_2 x + c_3 z + c_4 xz. \tag{5.6}$$

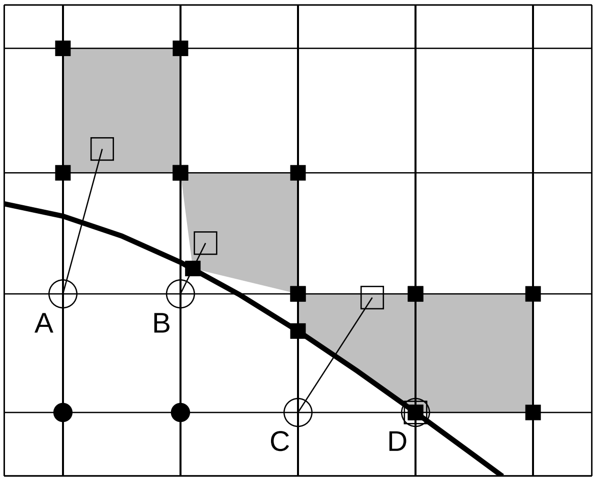


Figure 5.2. Four examples of the interpolation scheme used to reconstruct the immersed boundary. Ghost nodes are marked with open circles, and image points with open squares. The normal vector to the immersed surface connects the ghost and image points. Interior nodes are indicated with a solid circle, and the neighbors used in the interpolation scheme are marked with a solid square. The interpolation region is shaded.

Four neighboring points are used to define the interpolation region, and are chosen as either computational nodes or as points on the boundary. For Dirichlet boundary conditions, the neighbors nearest to the image point are chosen. For Neumann boundary conditions, neighbors nearest to the intersection of the surface normal with the boundary are chosen.

Because the boundary may intersect the computational cell in an arbitrary manner, the IBM algorithm must include logical directives to choose neighbors for a variety of geometric cases. Examples of the choices made for a Dirichlet boundary condition are shown in figure 5.2, but this illustration is not exhaustive. It can be seen in case A that if all four nodes surrounding the image point belong to the fluid domain, then these are chosen as the nearest neighbors. In case B, one of the surrounding computational nodes lies within the solid domain, so in this case the boundary point in the surface normal direction is chosen as a nearest neighbor. If two of the neighboring computational nodes are solid points, then two boundary points are chosen, as in case C. In case D the ghost point is in close proximity to the boundary, and with the time variant grid this point can move between the solid and fluid domains. In close proximity cases, flexibility is added to the reconstruction scheme so that it is applied at this node, regardless of being in the solid or fluid domain. If the ghost node lies in the fluid domain, then it is excluded from being used as a neighbor, as solid or ghost nodes are never used as neighbors in the interpolation scheme. Additionally, because the WRF grid remains curvilinear as it follows large scale terrain features, we found that special care must be exercised in determining the interpolation neighbors by accounting for horizontal gradients in the vertical coordinate. This is done by searching for neighbors in each vertical column of nodes independently, as adjacent nodes will not have identical vertical heights.

The constants $\mathbf{c}$ from the interpolant (5.6) are determined by solving a system of equations (5.7) for each ghost point, where the rank is equal to the number of neighbors.

$$\mathbf{c} = \mathbf{A}^{-1}\boldsymbol{\varphi} \tag{5.7}$$

The matrix $\mathbf{A}$ and the vector $\boldsymbol{\varphi}$ are dependent on the neighbors chosen for the interpolation and the type of boundary condition being imposed. For Dirichlet boundary conditions, equation (5.6) appears in the matrix equation. If the neighbor represented by equation (5.6) is a computational node, φ takes the calculated value at the node. If the neighbor is a boundary point, then the boundary condition (5.5a) is assigned to φ. For Neumann boundary conditions, the gradient of the interpolation function (5.6) is substituted into the boundary condition (5.5b), and equation (5.8) results.

$$\frac{\partial \varphi}{\partial \mathbf{n}} = c_2\mathbf{n}_x + c_3\mathbf{n}_z + c_4(\mathbf{n}_z x + \mathbf{n}_x z) \tag{5.8}$$

For neighbors on the boundary, this equation is used in (5.7) instead. Once the interpolation constants are determined, the value of the image point is found with equation (5.6). As a last step, the variable value at the ghost node is calculated and assigned.

The method proposed here is similar to the interpolation schemes used by Tseng and Ferziger [2003] and Mittal et al. [2008], but has additional favorable properties. Tseng and Ferziger used a linear interpolation scheme with three neighbors. The boundary point and the two nearest computational nodes to the ghost point were used for determining the weighting coefficients used in calculating the ghost point value. Large extrapolation coefficients result when the nearest neighbors are close to the boundary point and the ghost point is relatively far away. When large gradients exist in flow variables, such as in velocity when a no-slip boundary condition is used, large weighting coefficients cause the IBM to assign unphysical velocity values to the ghost points, often leading to numerical instabilities. Our method eliminates this problem by choosing neighbors which are closest to the image points when a Dirichlet boundary condition is used, and closest to the boundary point when a Neumann boundary condition is used. This ensures that the flow variable assigned at the ghost point has a physical value, and numerical stability is maintained. Mittal et al. [2008] allow neighboring solid nodes to be chosen as interpolation neighbors. These solid nodes are ghost nodes, leading to a coupling of the solution. An additional iterative procedure is used to deal with the coupling of ghost cell values. Our method avoids this additional step by only choosing neighbors in the fluid domain.

5.3.2 IBM in WRF

Initialization

Standard preprocessing routines are used to create domain input data for the IBM-WRF model, creating a terrain-following grid based on the large scale topography. Variables are initialized throughout the entire domain, even though some points will be reclassified as solid points during integration. Changes to WRF's preprocessing procedure include defining small scale terrain, which will be resolved with the IBM, separately from the large scale terrain, to which the terrain-following coordinates conform. For cases with atmospheric physics the land surface properties are modified to reflect the new location and properties of the small scale terrain.

An additional domain initialization routine exists in the run time portion of the WRF executable, and is called for all domains including the parent domain, newly created nested domains, and moving domains. This initialization routine is where the IBM initialization is added, and the immersed boundary condition is set for the prognostic variables. For the simulations presented in this paper, velocity fields are initialized with a no-slip condition and scalar fields are initialized with a zero flux condition.

During initialization, the IBM procedure begins with the determination of ghost points and ends by assigning values at ghost points. The full values (base state plus perturbations) of u, v, w, θ, and all scalars are used. First, nodes must be marked as interior or exterior to the domain. To make this determination, the elevation of each grid cell must be converted from geopotential by combining the base and perturbation states and dividing by gravity $z = (\bar{\phi} + \phi')/g$. Geopotential is defined on the horizontal faces of a computational cell, coincident with the w velocity. Height is averaged to the locations of the other variables which are located at the cell center and on vertical cell faces. Solid nodes, which are internal to the terrain, are set to zero for velocity variables but the values of temperature, moisture, and passive scalars are not modified at these points.

Several variables are not modified at the immersed boundary. Pressure and density are defined by equations of state, and do not require a boundary condition. The equation for perturbation geopotential (5.1d) is obtained by rearranging the definition of the coordinate velocity $\dot{\eta}$. As the coordinate is allowed to evolve independently of the immersed boundary, an immersed boundary condition is not applied.

Integration

As noted in section 5.2, a time-split scheme is used for numerical integration. Velocity and potential temperature are advanced on the small acoustic time step, while advection-diffusion equations for moisture and passive scalars are integrated exclusively on the large Runge-Kutta time step. IBM routines must be called on both time step frequencies to enforce boundary conditions for the separate integrations. Additionally, because the isobaric coordinate in WRF evolves in time, the IBM procedure of reconstructing the boundary with geometric interpolation must be completed for each iteration of the solver.

During integration u, v, w, and θ must be updated during the acoustic time step, on which the coupled perturbation variables are advanced $(\mu_d u)^{\tau}$, $(\mu_d v)^{\tau}$, $(\mu_d w)^{\tau}$, $(\mu_d \theta)^{\tau}$. IBM routines are called after advancement of all variables on the acoustic time step. In order to set the immersed boundary condition, the full uncoupled variable is calculated as $\varphi =$

$((\mu_d\varphi)^{\tau+\Delta\tau} + (\mu_d\varphi)^t)/\mu_d$, and the full grid height is calculated as $z = (\bar{\phi} + \phi'^t + \phi'^{\tau+\Delta\tau})/g$. Full variable values are used at nodes included in the interpolation which determine the ghost point value φ_G. The value of the coupled acoustic perturbation at the ghost point is then calculated as $(\mu_d\varphi)_G^{\tau+\Delta\tau} = \mu_d\varphi_G - (\mu_d\varphi)_G^t$, and this coupled value is used at the ghost point to enforce the boundary condition. Again, a boundary condition is not set at the immersed boundary for ϕ'^τ, and p'^τ and $\alpha_d'^\tau$ are diagnosed from state equations.

Scalar variables are advanced on the large Runge-Kutta time step. Potential temperature and scalars are colocated, so that if the same type of boundary condition is used for all of these variables, then the grid elevation, ghost point locations, and interpolation points from the last acoustic time step are still valid. This information must only be recalculated if the type of boundary condition varies. The IBM procedure as previously explained is used to set the boundary condition on each scalar, which for this work only includes water vapor. Lundquist et al. [2008b] performed simulations using the IBM with a passive tracer.

During integration the reconstructed solution is enforced at the first grid point within the solid domain, but several options exist for the treatment of additional interior nodes. When higher order finite-difference schemes are used, solid nodes are included in the stencil of fluid nodes, and can affect the accuracy of the derivatives at these nodes. Iaccarino and Verzicco [2003] used both a second order and a high order upwind finite-difference scheme, and found that with the direct forcing type of IBM the treatment of interior nodes has little or no effect on the external flow solution. A fourth order finite-difference scheme is used in von Terzi et al. [2001] along with four different interior treatments. This study found that in many cases the interior treatment did not affect the outer fluid flow, but when the accuracy of near-wall gradients is critical to obtaining the correct flow solution, greater accuracy can be achieved by using a one-sided finite-difference at the immersed boundary. We implemented various treatment options for solid nodes, and tested both second order and higher order advection schemes, and concluded that there was little effect for the test cases presented in this manuscript. Therefore, we allow prognostic variables to evolve in accordance with the governing equations within the solid domain, and use the default advection scheme (5$^{\text{th}}$ order horizontal, 3$^{\text{rd}}$ order vertical). Options for eliminating this issue in the future are to reduce the order of the finite-difference scheme at the immersed boundary, apply the reconstruction scheme to multiple layers of ghost nodes, or use one-sided differences at the boundary.

5.4 Boundary conditions

Our current IBM implementation is capable of Dirichlet boundary conditions, as well as Neumann conditions for scalar quantities in the surface normal direction. The sigma WRF model uses a free slip bottom boundary condition or a wall model to approximate surface fluxes of momentum. To verify the IBM implementation, a direct comparison of the sigma WRF and IBM-WRF results is needed, with a common bottom boundary condition. A discussion of the boundary conditions for both the sigma WRF and IBM-WRF models is included in this section, including a new no-slip boundary condition which we have added to the existing WRF options.

5.4.1 Comparison of native and immersed boundary conditions

The native bottom boundary conditions in the sigma WRF model set the contravariant coordinate velocity to zero and use a kinematic boundary condition for the Cartesian vertical velocity. The set of equations given by (5.9a) and (5.9b) create a free slip bottom boundary condition.

$$\dot{\eta}_{surf} = 0 \tag{5.9a}$$

$$w_{surf} = \mathbf{V} \cdot \nabla h \tag{5.9b}$$

In equation (5.9b) h is a function specifying the terrain height. Horizontal velocities are extrapolated to the surface using a quadratic Lagrange polynomial. The shear stress at the boundary is implicitly set to zero, unless the effects of friction are taken into account by using log-law similarity theory. Vertical gradients of scalars are zero unless forcing is provided by the atmospheric physics parameterizations.

Our IBM implementation uses a no-slip bottom boundary condition which is commonly used in CFD simulations over complex topography, such as urban simulations with explicitly resolved buildings. To enable comparisons between IBM-WRF and WRF, we have added a no-slip condition to WRF. This boundary condition is also useful for high resolution simulations performed with the WRF solver, and enables comparisons between WRF and other CFD codes that use a no-slip boundary condition. The no-slip implementation is described in the following section, and this boundary condition is used for the verification simulations presented in chapter 6.

Wall models which approximate surface fluxes of momentum are commonly used in NWP, and have also been applied at immersed boundaries. Cristallo and Verzicco [2006] use a two-layer model with the IBM, which reconstructs the velocity field at the first fluid point by solving a simplified boundary layer equation. Choi et al. [2007] reconstructs the velocity field by fitting the tangent component to a power law. These studies show promising results, and the development of a wall model for IBM-WRF is in progress, with preliminary findings reported in chapter 8.

When specifying surface fluxes, our IBM implementation can apply fluxes in either the surface normal or vertical direction. It is critical that flux boundary conditions be enforced in the surface normal direction over steep slopes, such as vertical building surfaces. WRF prescribes vertical fluxes at the surface; therefore, the vertical flux boundary condition can be used when simulations with IBM are compared directly to WRF, as is done in the idealized valley simulation. To further simplify comparisons between WRF and IBM-WRF, we have used a constant eddy viscosity. Turbulent viscosity is needed to compute surface fluxes, and variation in this quantity would lead to differences in the enforced boundary condition.

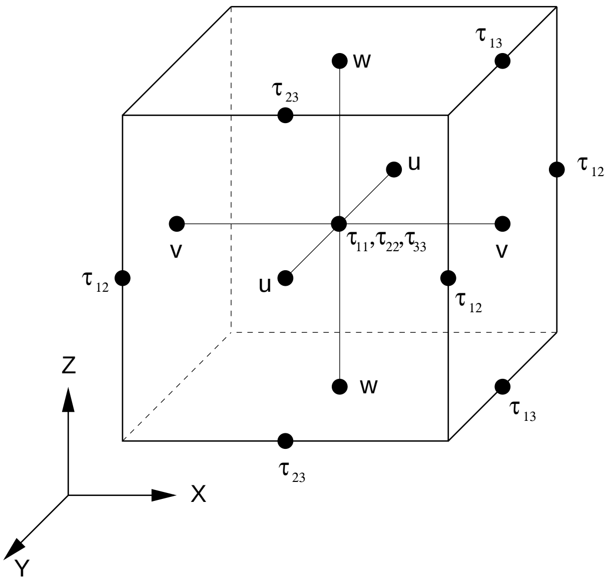


Figure 5.3. The Arakawa-C staggered grid used in WRF

5.4.2 Implementation of the no-slip boundary condition

As part of this work, a no-slip bottom boundary condition has been added to WRF as an additional option beyond the standard boundary conditions available in the code. To add the no-slip option, several modifications were made to the original WRF boundary conditions, which are given in the previous section as (5.9). The first equation (5.9a) is still appropriate, as it sets the contravariant velocity of the bottom coordinate to zero. This ensures that the bottom coordinate follows the terrain, and does not change position in time. The kinematic boundary condition given by (5.9b) is no longer valid, and is replaced with the requirement that the Cartesian vertical velocity $w = 0$ on the boundary. The need to extrapolate the horizontal velocities to the surface is eliminated, therefore no approximations are made in our formulation of the no-slip boundary condition. With these changes, the no-slip boundary condition is satisfied for the advective fluxes in the continuity and momentum equations. The use of an Arakawa-C staggered grid (illustrated in figure 5.3) eliminates the need to explicitly set the u and v velocities equal to zero on the surface. In the case of the vertical derivative in the advective term of the horizontal momentum equation (5.1b) it can be seen that because $\dot{\eta}$ and $\mathbf{V}_H$ are both zero at the surface, a one-sided vertical difference is sufficient for calculating this term at the bottom boundary of the domain.

Boundary conditions must be imposed on the diffusive flux terms in the momentum equation in order to achieve a no-slip condition. WRF employs eddy viscosity type turbulence models so that the turbulent mixing terms take the form given by (5.10), and the stress tensor is (5.11).

$$F_i = -\frac{\partial \tau_{ij}}{\partial x_j} + \frac{\partial}{\partial z}(\nabla_H z \cdot \tau_{ij}) \tag{5.10}$$

$$\tau_{ij} = -\mu\nu_t D_{ij} \tag{5.11}$$

Diffusive flux terms are calculated in physical space, so that the second term in equation (5.10) arises as the Jacobian of the coordinate transformation, where z is the physical height of the coordinate. The variable μ is the column mass per unit area, ν_t is the turbulent eddy viscosity, and D_{ij} is twice the strain rate or deformation tensor and is defined as

$$D_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - (\nabla_H z)_j \frac{\partial u_i}{\partial z} - (\nabla_H z)_i \frac{\partial u_j}{\partial z} \tag{5.12}$$

The locations of the components of the deformation and stress tensors are shown in figure 5.3. Diagonal elements of each tensor are located at the cell center, and off-diagonal elements are located at vorticity points which are centered on the cell edges.

The presence of the Jacobian terms complicates the implementation of the no-slip boundary condition by adding partial derivatives in the vertical direction to each term. The native WRF boundary condition (which is free slip) uses a quadratic Lagrange polynomial to estimate the u and v velocity at the surface for the purpose of calculating D_{11}, D_{22}, and D_{12}. This step is eliminated for the no-slip boundary condition, and a value of zero is used for u and v on the surface. In the native WRF boundary condition D_{13} and D_{23} are zero because they are located on the surface. For the no-slip boundary condition, the surface values are calculated using a one sided difference. These new values of the deformation tensor are used for calculating the turbulent stresses.

Finally, the equation for the diffusion terms includes additional terms arising from the Jacobian of the coordinate transformation, as seen in equation (5.10) above. With a free slip boundary condition, τ_{ij} is zero on the surface. To satisfy the no-slip boundary condition, certain elements of τ_{ij} must now be calculated at various locations on the surface, such as below the cell center, below u points, etc. The calculation of these additional surface stress terms is consistent with the procedure previously described for evaluating deformation at the surface, and the methods employed are one-sided differences, setting velocities to zero at the surface, and averaging surface values to intermediate locations.

5.5 Inclusion of atmospheric parameterizations

Atmospheric parameterizations often provide surface forcing in NWP models. Surface physics interact with the lowest model level when terrain-following coordinates are used, therefore, it was necessary to modify these routines to couple them to the immersed boundary. The set of schemes that have been coupled to the immersed boundary are the Rapid Radiative Transfer Model (RRTM) for longwave radiation, the MM5 (Dudhia) model for shortwave radiation, the MM5 surface layer scheme, and the NOAH land-surface model. Documentation of these schemes can be found in Skamarock et al. [2007] and the references within.

5.5.1 Radiation models

The radiation schemes currently used in WRF are column models, where each column is treated independently. The terrain is treated as if it is a horizontal plane at each column. To couple this model with the immersed boundary method, the vertical integration limits are modified to exclude any portion of the atmosphere below the terrain. Additionally, variables such as temperature, pressure, and grid spacing must be modified in the vicinity of the immersed boundary by extrapolating these values onto the surface using data from fluid points above the immersed terrain. The modified data for the atmospheric column is then used in the existing radiation routines.

5.5.2 Surface physics

WRF calculates exchange coefficients of heat, moisture, and momentum using the MM5 surface layer model which is based on Monin-Obukhov similarity theory. Stability regimes are determined by the bulk Richardson number, defined by equation 5.13, where g is the gravitational constant, z is a reference height above the terrain, z_{surf} is the terrain level, θ_v is virtual potential temperature, and V is the horizontal wind speed which has been modified by a vertical convection velocity.

$$Ri_b = \frac{g(\theta_v(z) - \theta_v(z_{surf}))(z - z_{surf})}{\theta_v(z)V(z)^2} \tag{5.13}$$

WRF chooses the reference height z to be height of the first computational cell, so that the vertical grid spacing defines the offset z_o between the reference height and the surface. For the immersed boundary method, the reference height must be modified so that it is located above the terrain, rather than at the first grid point in the domain. With IBM-WRF, the offset z_o is added to the terrain height z_{surf}, as shown in figure 5.4. This modified reference height is above the immersed boundary, but unlike in the WRF case, it does not coincide with a computational node. Velocities and temperature, as well as those quantities needed for the conversion to virtual temperature must be interpolated to the modified reference height. Additionally the surface values must be replaced with values on the immersed boundary. The Monin-Obukhov length scale is then calculated based on the stability class. Next, the 2 m and 10 m reference height data are calculated. Finally, the exchange coefficients are calculated for heat and moisture. Again, the values used in these calculations are modified for the immersed boundary following the example given above for the bulk Richardson number. It is important to note that while scalar fluxes are calculated with this approach, surface momentum fluxes are not based on Monin-Obukhov theory when the no-slip boundary condition is used, as these conditions are incompatible. This is true of both the native WRF and IBM-WRF computations shown here.

WRF uses the NOAH land-surface model to provide heat and moisture fluxes at the land surface, using radiative and atmospheric forcing from the radiation and surface layer schemes,

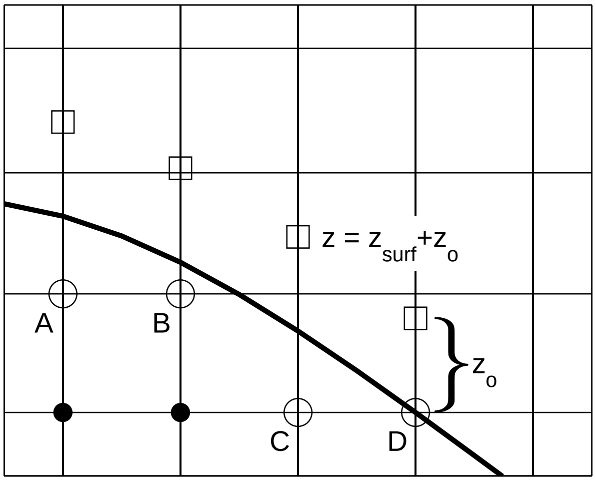


Figure 5.4. Examples of the modified reference height z used at the immersed boundary. Ghost nodes are marked with open circles, and modified reference heights with open squares.

as well as land-surface properties to determine the fluxes. These fluxes serve as the bottom boundary conditions for potential temperature and water vapor. This model also calculates a water budget over four model layers, including physical effects like evapotranspiration and runoff. Many of the quantities used by the land-surface model have already been modified to include the effects of the immersed boundary by the models discussed above. The NOAH model also makes use of similarity theory. As before, the reference height is added to the height of the immersed boundary, and the variables used in the calculation are modified accordingly.

5.6 Conclusions

We have developed an immersed boundary method for WRF which is sufficiently flexible to work with the time variant isobaric sigma coordinate which is native to WRF and accurately enforces Dirichlet and Neumann boundary conditions within the mode splitting time integration scheme. A unique interpolation scheme is used for the IBM, which avoids problematic numerical instabilities and the need for an iterative procedure to decouple the interior and exterior solutions as in previous methods [Tseng and Ferziger, 2003; Mittal et al., 2008].

Our IBM-WRF implementation is capable of handling highly complex terrain, while maintaining atmospheric parameterizations included in the WRF mesoscale model. We coupled a set of surface parameterizations to the immersed boundary, providing realistic land-surface forcing. These parameterizations may be appropriate for some scales where IBM is used, such as in mountainous terrain, but may need to be modified in urban terrain (where traditional similarity theory may not be applicable). Regardless, the capability exists to couple IBM-WRF to surface parameterizations when they are available. This means that IBM-WRF can be used in a grid-nesting configuration to accommodate synoptic scale lateral boundary conditions in an urban environment with grid resolution fine enough to explicitly resolve flow around buildings. We can therefore combine the best properties of CFD (capable of resolving buildings) and mesoscale codes (capable of handling atmospheric physics and boundary conditions). A standard terrain-following WRF simulation can be used at mesoscales, with a seamless transition to IBM-WRF at finer scales. Furthermore, IBM eliminates grid distortion which is present high above the land surface when using sigma coordinates over steep terrain.

Drawbacks to this approach are that it can be expensive to maintain fine near-surface grid resolution of the immersed boundary at elevated heights, such as at mountain peaks. The hybrid sigma and IBM coordinate may reduce this constraint by allowing gentle slopes in topography to be represented with sigma coordinates, while steep slopes are represented with the IBM. This could allow for increased resolution near the ground, while minimizing the number of nodes below the terrain. There are increased computational costs associated with the IBM algorithm, however, the IBM allows simulations of flow over terrain which could not otherwise be handled by WRF. Opportunities exist for optimization as the code is developed further.

The ability to integrate IBM simulations within a mesoscale model with full atmospheric physics is a significant milestone in the progress of atmospheric boundary layer simulations. The development of IBM-WRF enables a range of applications, including building-resolving simulations of urban dispersion for emergency response efforts or even urban air quality modeling. Similarly, simulations of flow in highly complex mountainous terrain with near-vertical slopes can now be accommodated without compromising the accuracy or stability of the solution. Extension to three dimensions is been relatively straightforward and is demonstrated in chapter 7. The extension simply requires a larger interpolation stencil for the IBM algorithm. Incorporation of similarity theory for surface momentum fluxes is also in progress (Chapter 8) and will allow even more realistic application to atmospheric boundary layer flows.

Chapter 6

Validation of the immersed boundary method[*]

The accuracy of the immersed boundary solver is examined in this chapter by comparing the sigma-coordinate WRF and IBM-WRF solutions for canonical flows. Validation cases include two-dimensional pressure driven flow over an isolated hill and idealized valley simulations with both specified and parameterized surface fluxes. Results are also presented for flow over two blocks of the New York City skyline, which demonstrate the method's ability to handle extremely complex terrain with sharp corners and vertical terrain gradients.

6.1 Introduction

Simulations are presented for canonical cases with shallow terrain slopes, and comparisons between simulations with the native terrain-following coordinates and those using the immersed boundary method show excellent agreement. Validation cases demonstrate the ability of the immersed boundary method to handle both Dirichlet and Neumann boundary conditions. Additionally, realistic surface forcing can be provided at the immersed boundary by atmospheric physics parameterizations, which are modified to include the effects of the

[*] **This chapter is a reproduction (with minor modifications) of a portion of the paper "An Immersed Boundary Method for the Weather Research and Forecasting Model" by Katherine A. Lundquist (the principal author), Fotini Katopodes Chow, and Julie K. Lundquist, published in Monthly Weather Review, March 2010, Volume 138(3), pages 796-817 [Lundquist et al., 2010], ©Copyright 2010 American Meteorological Society. The full copyright notice is included as appendix C.**

immersed terrain. Using the immersed boundary method, WRF is capable of simulating highly complex terrain, as demonstrated by a simulation of flow over an urban skyline.

6.2 Witch of Agnesi hill simulations

The performance of the IBM method is examined in this section by considering flow over a two-dimensional hill defined by the Witch of Agnesi curve. This case is often used to verify that topographic forcing is correct at the bottom boundary, and is chosen here because the terrain slopes are shallow enough so that a direct comparison can be made between WRF and IBM-WRF.

The test flow case is startup flow over a two-dimensional hill. The terrain height h_t is defined by the Witch of Agnesi curve given in equation 6.1 using a peak height h_p and mountain half-width a of 100 m each.

$$h_t(x) = \frac{h_p}{1 + (x/a)^2} \tag{6.1}$$

The flow is initialized with a neutral and quiescent sounding, and driven with a constant pressure gradient in the x direction. The number of grid points used in each direction is $(n_x, n_y, n_z) = (120{,}2{,}172)$ for a total domain size of $(X, Y, Z) = (595\text{ m}, 5\text{ m}, 600\text{ m})$. In the horizontal dimensions a constant 5 m grid spacing is used. In the vertical direction the grid is not stretched in η, however, the pressure coordinate naturally results in stretching in physical space. The WRF and IBM-WRF grids are shown in figure 6.1. At initialization the minimum vertical grid spacing in the WRF grid is $\Delta z_{min} \approx 2.85$ m, and occurs at the peak of the hill, while the maximum spacing is $\Delta z_{max} \approx 3.52$ m at the domain top. In the IBM domain the vertical grid spans the entire 600 m domain height, giving a more uniform (and slightly coarser) grid resolution, with $\Delta z_{min} \approx 3.42$ m and $\Delta z_{max} \approx 3.60$ m.

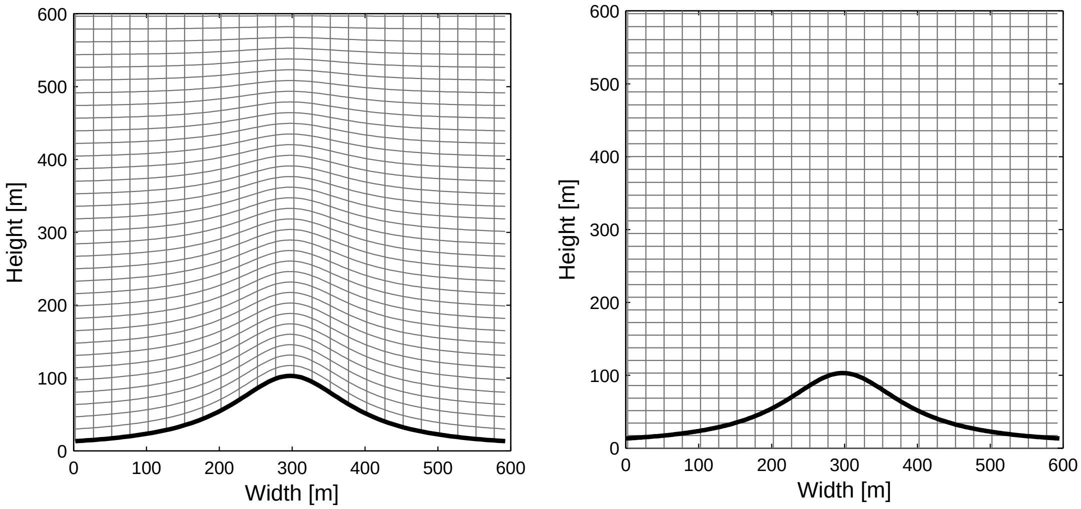


Figure 6.1. The WRF grid with terrain-following coordinates is shown (left) in contrast with the IBM-WRF grid (right). Terrain-following coordinates map to the bottom boundary, while with IBM the coordinates remain flat. Every fifth grid line is shown.

Table 6.1. Location of minimum and maximum velocities for steady state flow over a hill in the sigma WRF simulation. Velocity values at this location for both IBM-WRF and sigma WRF. Differences between the IBM and WRF solutions are given as both absolute and relative values.

	Location [m]		Velocity [m s^{-1}]		Difference	
	x	z	IBM	WRF	Absolute [m s^{-1}]	Relative
u_{min}	122.5	49.6	-0.203	-0.208	0.005	2.4%
u_{max}	402.5	500	10.736	10.713	0.023	0.2%
w_{min}	192.5	71.6	-0.077	-0.083	0.012	14.5%
w_{max}	272.5	107.6	0.192	0.192	-4.7e-5	0%

Periodic boundary conditions are used at the lateral boundaries. A no-slip boundary condition is set on velocity at the terrain surface, along with a zero flux condition on temperature. At the top of the domain, the native WRF boundary condition is used (isobaric and a material surface), with a diffusive damping layer 50 m in depth.

The solution is integrated until a steady state solution is obtained. Figure 6.2 shows the u velocity at the domain center for three different heights for 7 days of simulation time. Evolution of the velocity field is nearly identical between the WRF and IBM-WRF solutions, indicating that the rate of convergence toward steady state is not affected by the IBM.

Steady state results are shown for the hill case after 7 days (168 hours) of simulation time. Figure 6.3 shows u and w velocity profiles at several locations across the domain. It can be seen that the IBM solution matches the no-slip WRF solution almost exactly. To compare the results further, the IBM and WRF solutions are linearly interpolated onto a common terrain-following grid. Differences in the simulations are quantified in table 6.1, where maximum and minimum velocity values for the no-slip WRF solution are presented along with velocity values for the IBM-WRF solution at the same location. Given the different vertical grid spacing in the two simulations and the presence of errors associated with terrain-following coordinates in the sigma WRF simulation, the excellent agreement is remarkable.

Contours of the same data are shown in figure 6.4, where contours of velocity and the difference between the IBM and WRF solutions are included. The u velocity increases as it crests the hill, and areas of weak recirculation are observed in the troughs. In the IBM solution a reverse flow develops under the terrain. This flow can be seen in figure 6.3, however in figure 6.4 data below the hill is omitted because the IBM solution is interpolated onto a common terrain-following grid. The maximum differences present in figure 6.4 are 0.039 and -0.021 m s^{-1} for the u and w fields, respectively. The domain averaged differences are smaller, with $\Delta U_{ave} = 0.017$ and $\Delta W_{ave} =$ 6e-4 m s^{-1}.

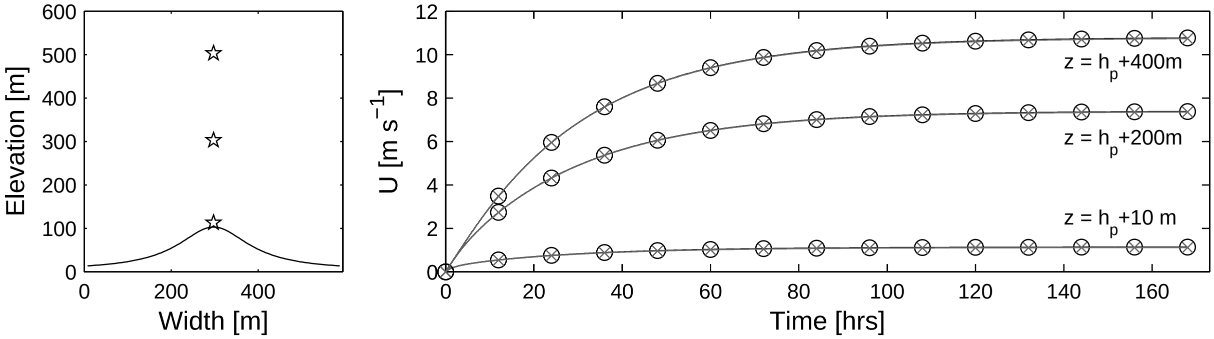


Figure 6.2. The u velocity component is plotted at the domain center for three different heights indicated by stars (left). The first point is 10 m above the terrain peak h_p, the second is 200 m, and the third is 400 m. IBM-WRF is plotted with a black line, while WRF is shown in grey (right). Markers are placed at 12 hour intervals. The WRF and IBM-WRF solutions are virtually indistinguishable .

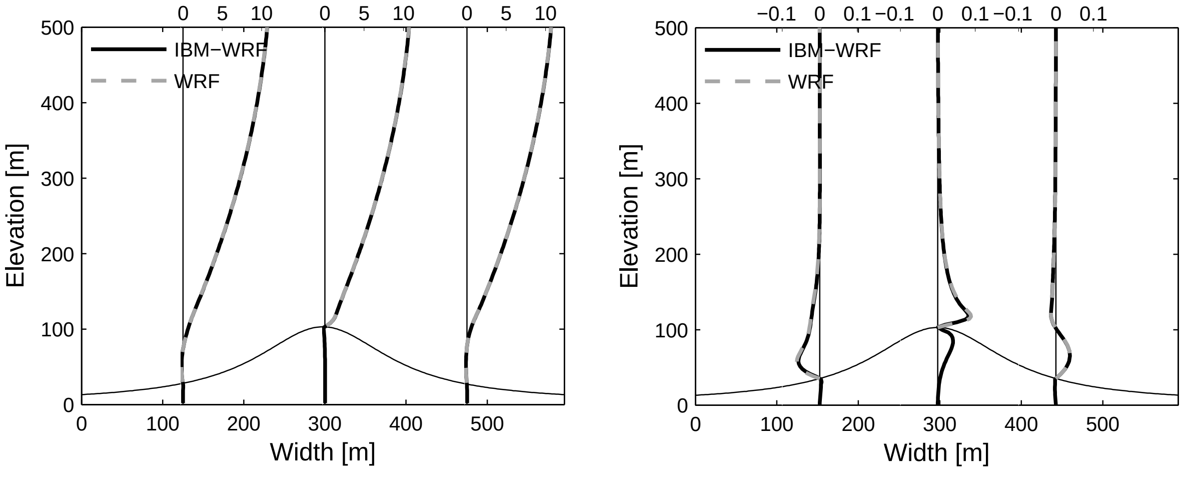


Figure 6.3. Profiles of u and w velocity [m s^{-1}] are shown for steady state flow over a hill at 168 hours.

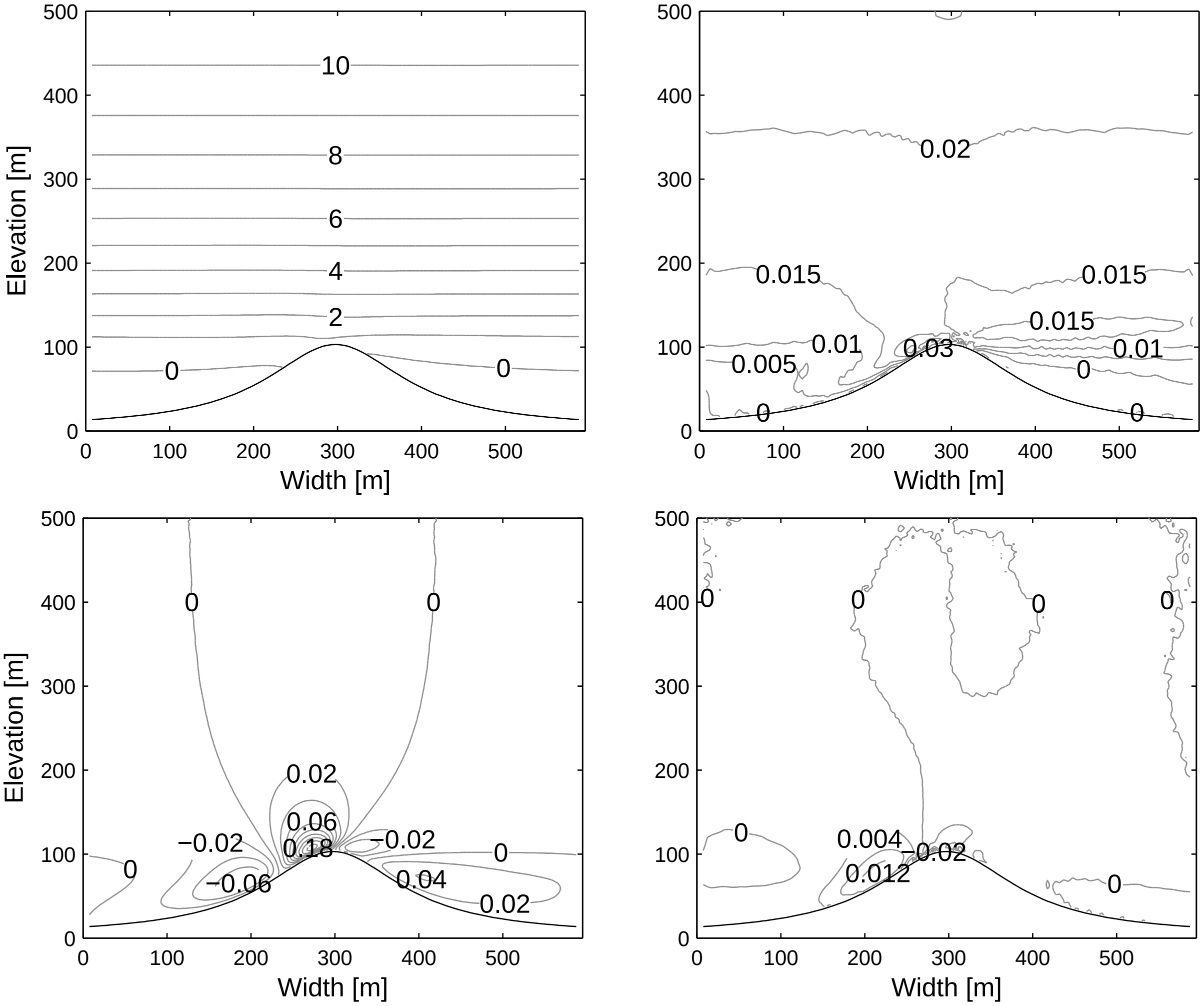


Figure 6.4. Contours of u (top left) and w (bottom left) velocity are shown for IBM-WRF for steady state flow over a hill at 168 hours. Contour intervals are 1 m s^{-1} for u and 0.02 m s^{-1} for w. Contours of the difference between the IBM-WRF and WRF solution are shown for u (top right) and w (bottom right) with intervals of 0.005 and 0.004 m s^{-1} respectively.

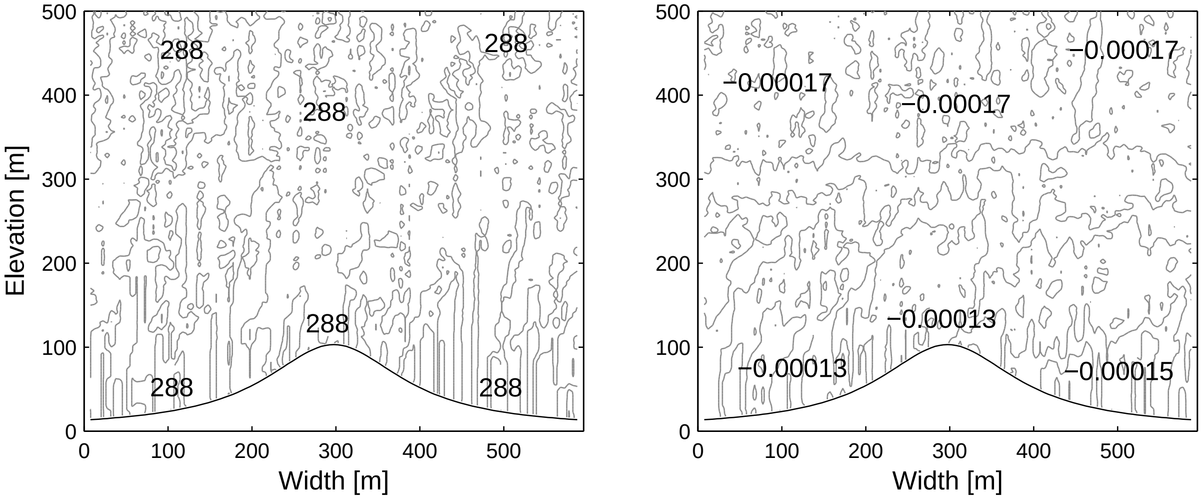


Figure 6.5. Contours of potential temperature are shown for IBM-WRF (left), along with the difference in the IBM and WRF solutions (right). All units are in Kelvin.

Contours of potential temperature for the IBM solution, and differences in the IBM and WRF solutions are shown in figure 6.5. It can be seen that with no surface heating, the IBM solution maintains the initialization temperature of 288 K, while there is a very slight increase in temperature in the WRF domain. It was found that small errors developed in the WRF solution due to the native bottom boundary condition, which uses a Lagrange polynomial to extrapolate temperature to the surface.

6.3 Idealized valley simulations

In the previous section it was shown that our IBM implementation is able to reproduce results from numerical simulations using the native terrain-following coordinate in WRF. A no-slip boundary condition was imposed on the velocity at the terrain surface and a zero gradient boundary condition was used for potential temperature. These two boundary conditions are adequate for studying a variety of flows, and are commonly the only boundary conditions available in a traditional CFD code. For realistic atmospheric flows it is necessary to represent non-zero fluxes at the surface, such as those of heat and moisture. Here we examine non-zero gradient boundary conditions on the immersed boundary in the context of thermally induced slope flows in an idealized valley.

The idealized valley terrain and initialization are similar to those used in the valley winds studies of Rampanelli et al. [2004] and Schmidli et al. [2008]. The valley terrain is defined by $h_t = h_p * h_x$, where the peak height is $h_p = 1.5$ km and the valley profile h_x is given by equation 6.2. In this equation, the valley floor half-width is $V_x = 0.5$ km, and the hill half-width is $S_x = 9$ km.

$$h_x(x) = \begin{cases} 0 & |x| \le V_x \\ 0.5 - 0.5\cos\left(\pi\frac{|x|-V_x}{S_x}\right) & V_x < |x| < V_x + S_x \\ 1 & V_x + S_x \le |x| \le V_x + S_x + P_x \\ 0.5 + 0.5\cos\left(\pi\frac{|x|-(V_x+S_x+P_x)}{S_x}\right) & V_x + S_x + P + x < |x| < V_x + 2S_x + P_x \\ 0 & |x| \ge V_x + 2S_x + P_x \end{cases} \tag{6.2}$$

Again, the terrain slopes are gentle enough that a direct comparison can be made between WRF using terrain-following coordinates and IBM-WRF. This case is more challenging for the immersed boundary method because the flow is induced by the approximate boundary conditions at the immersed surface itself, instead of being induced by a large scale pressure gradient or mean flow.

The domain is initialized with a quiescent sounding that is moist and stably stratified. The sounding prescribes a constant 40% relative humidity and the potential temperature is specified by $\theta(z) = \theta_s + \Gamma z + \Delta\theta\left[1 - \exp(-\beta z)\right]$, where $\theta_s = 280$ K, $\Gamma = 3.2$ K km^{-1}, $\Delta\theta = 5$ K, and $\beta = 0.002$ m^{-1}.

For the WRF simulation with terrain-following coordinates, the total domain size is $(X, Y, Z) = (60$ km, 0.6 km, 10 km$)$, where the number of grid points used in each direction is $(n_x, n_y, n_z) = (301{,}3{,}60)$. When IBM-WRF is used, the bottom of the domain is lowered to 200 m below the zero level, so that the domain size is $(X, Y, Z) = (60$ km, 0.6 km, 10.2 km$)$. This allows a minimum of two grid points below the immersed terrain, and the number of grid points becomes $(n_x, n_y, n_z) = (301{,}3{,}62)$. In each simulation the horizontal grid spacing is $\Delta X = \Delta Y = 200$ m. For the terrain-following coordinate, the minimum vertical grid spacing occurs at the mountain peaks ($\Delta Z_{min} = 95.6$ m), and the maximum at the domain top ($\Delta Z_{max} = 307.8$ m). For the IBM-WRF grid the minimum vertical grid spacing occurs at the lowest grid level which is below the terrain ($\Delta Z_{min} = 101.9$ m), and the maximum at the domain top ($\Delta Z_{max} = 307.4$ m).

Total integration time for the simulation is 12 hours, starting at 6:00 UTC and ending at 18:00 UTC. A constant eddy viscosity of 60 m^2 s^{-1} is used in the turbulent diffusion terms, which is consistent with the typical range of daytime eddy diffusivites of 10 to 100 m^2 s^{-1} [Yasuda, 1988]. When this eddy diffusivity is used the mixing is strong enough to maintain a neutral stability profile during surface heating, therefore avoiding the formation of thermal convection cells and making instantaneous comparisons between simulations possible.

Both uncoupled and coupled cases are considered. In the uncoupled cases, the surface heating is specified as a function of time, and there are no surface or land attributes such as vegetation or soil type. This allows verification of the surface flux condition at the immersed boundary without the added complexity of the land-surface model. In the coupled cases, the surface fluxes are prescribed by atmospheric parameterizations, which have been modified to recognize the immersed boundary as the terrain surface. Periodic boundary conditions are used at the lateral boundaries. A diffusive damping layer 3.8 km thick is used at the top of the domain, and a no-slip boundary condition is used at the terrain surface .

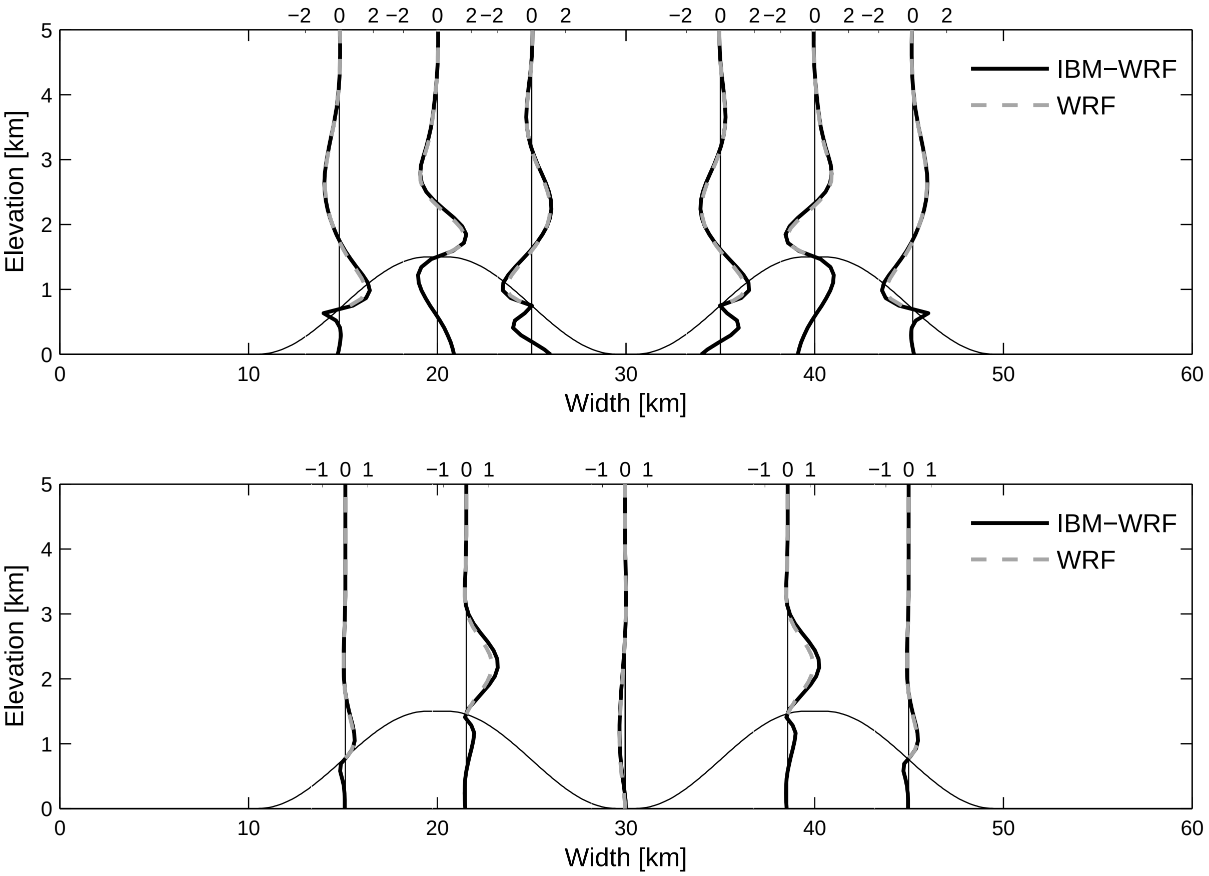


Figure 6.6. Profiles of u (top) and w (bottom) velocity [m s^{-1}] for IBM-WRF and WRF with terrain-following coordinates at 12:00 UTC.

6.3.1 Specified surface heating

In this case the land-surface model is turned off and a Neumann boundary condition is applied to potential temperature by defining the surface heat flux to be $Q(t) = Q_{max} \sin(\omega t)$, where the maximum sensible heat flux is $Q_{max} = 200$ W m^{-2} and a period of $\omega = 2\pi/24$ hrs is used to represent a daytime heating cycle.

Figure 6.6 depicts profiles of u and w velocity at 12:00 UTC, the time of maximum radiative forcing. A clear circulation pattern forms with anabatic winds converging at the mountain peaks, and subsidence at the valley center. The profiles match very well, and show that the boundary condition at the immersed boundary is actually achieved. Figure 6.7 shows instantaneous profiles of potential temperature at the valley center as it evolves in time. The evolution of the profiles for the IBM and WRF simulations compare well, and excellent agreement is achieved in the development of the mixed layer.

To further analyze the differences between the IBM-WRF and WRF solutions, both time and domain averaged differences are calculated on a common and time-independent terrain-

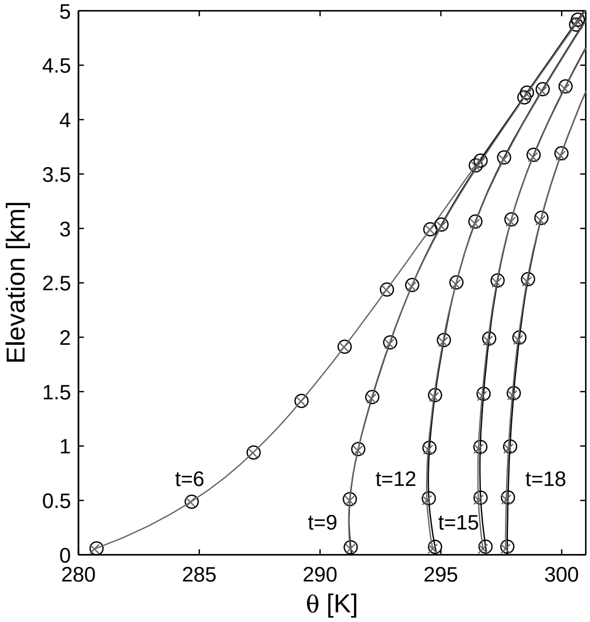


Figure 6.7. Profiles of potential temperature [K] at the valley center for times 6:00, 9:00, 12:00, 15:00 and 18:00 UTC. IBM-WRF is shown as a black line, and WRF with terrain-following coordinates is shown in grey. Markers are placed at every fourth data point.

following grid. Once the two solutions are on a common grid they may be compared directly, although the error introduced by the interpolation is unknown. Spatial variations in the two solutions are calculated for a given variable ϕ with equation (6.3a). Temporal variations are calculated with equation (6.3b).

$$\Delta\phi(x,\eta) = \overline{\phi_{IBM}(x,\eta,t) - \phi_{WRF}(x,\eta,t)}^{t} \tag{6.3a}$$

$$\Delta\phi(t) = \overline{|\phi_{IBM}(x,\eta,t) - \phi_{WRF}(x,\eta,t)|}^{x,\eta} \tag{6.3b}$$

The results of the time averaged spatial differences are presented in figures 6.8 and 6.9. Figure 6.8 shows time averaged differences in θ, where $\Delta\theta_{max}$ = 0.099 K and $\Delta\theta_{min}$ = -0.023 K. The IBM simulation predicts a higher temperature than WRF at the surface with the maximum difference occurring on the valley slopes, and a slightly cooler atmosphere aloft in the valley center. The higher surface temperature is attributed to differences in the vertical grid spacing. In the sigma WRF case, the average spacing between the first potential temperature node and the surface is Δz_{ave} = 51.0 m, where as the spacing is Δz_{ave} = 63.5 m for IBM-WRF. As the vertical grid spacing increases, a larger temperature increment is needed between the surface and the first vertical grid point in order to maintain the specified flux boundary condition. Differences in θ contribute to the differences in velocity shown in the next two figures. Velocity differences are $\Delta U_{max} = \Delta U_{min}$ = ±0.254 m s^{-1} and ΔW_{max} = 0.133 m s^{-1} and ΔW_{min} = -0.068 m s^{-1}. In u the largest differences are seen on the valley slopes, specifically inside of the valley. Additionally, the largest differences in w are seen at

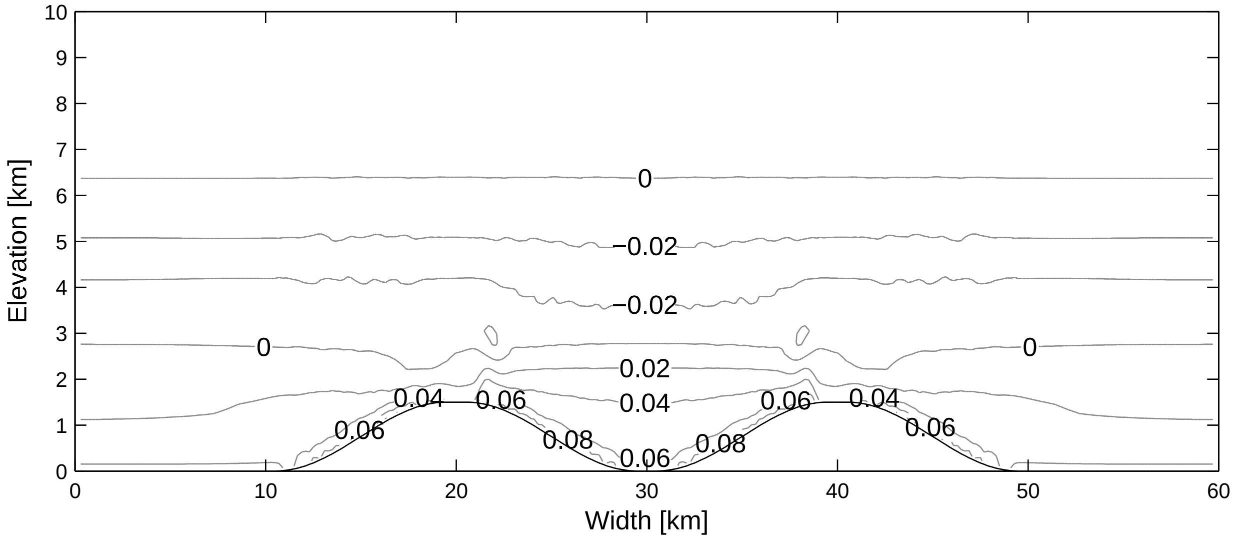


Figure 6.8. Time averaged differences between IBM-WRF and WRF for potential temperature. Differences are absolute with units of Kelvin.

the valley peaks, where there is a slight offset in the location of the buoyant plumes. These differences are small considering that the simulations have different grids, and use different methods for specifying boundary conditions.

Differences between the simulations can also be viewed as a function of time, as in figure 6.10. It can be seen here that the potential temperature fields are different at initialization. This difference is due to interpolation errors from both interpolating a non-linear initial sounding profile onto the WRF and IBM-WRF grids, and interpolating the solutions onto a common grid for comparison. Excellent agreement is seen in the potential temperature field, and the differences are on the same order of magnitude at the end of the simulation as at initialization. The velocity fields are initialized as quiescent, therefore they are identical at initialization. The differences in velocity rise during the first thirty minutes of the simulation, and then rapidly decrease. A stair step type appearance in the flow is observed in the first thirty minutes of the IBM simulation, as velocities at discrete points adjust to the smooth immersed boundary. As flow patterns develop, the stepped appearance quickly smoothes out and disappears. This peak indicates that differences could be reduced by adding spin-up time to the simulation. It is important to note that the differences do not grow in time, and in fact trend downwards as simulation time increases (even though the solution is time dependent).

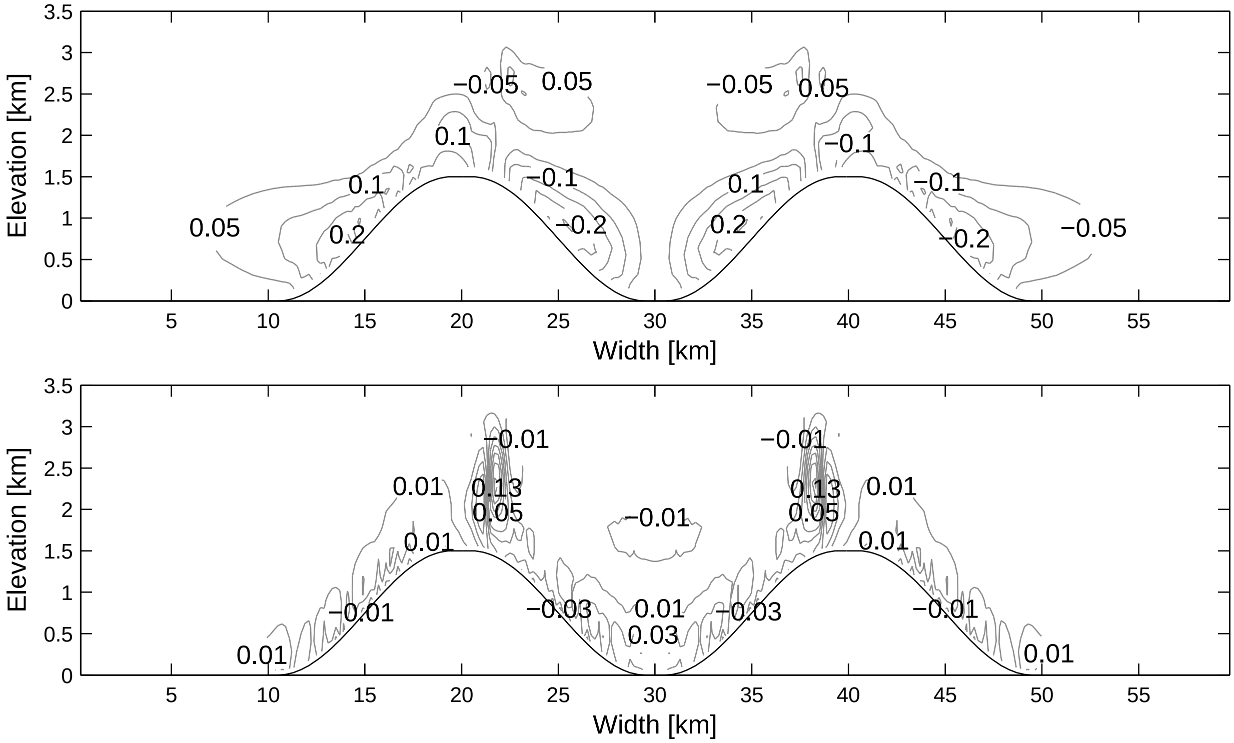


Figure 6.9. Time averaged differences between IBM-WRF and WRF for u (top) and w (bottom) velocity. Differences are absolute with units of m s^{-1}. Contour intervals of 0.05 m s^{-1} and 0.02 m s^{-1} are used for u and w respectively, with the zero contour suppressed.

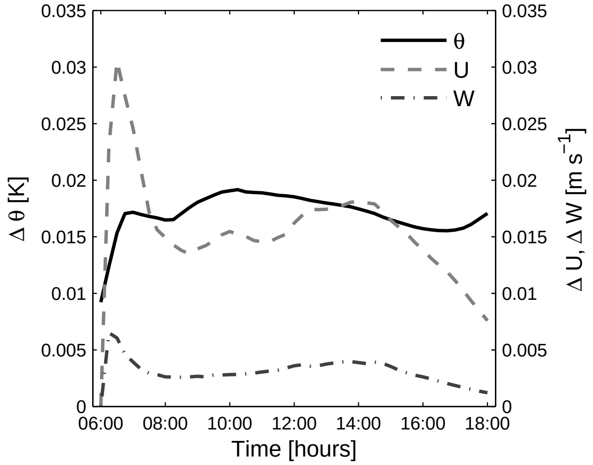


Figure 6.10. Domain averaged differences of θ, u, and w as a function of time.

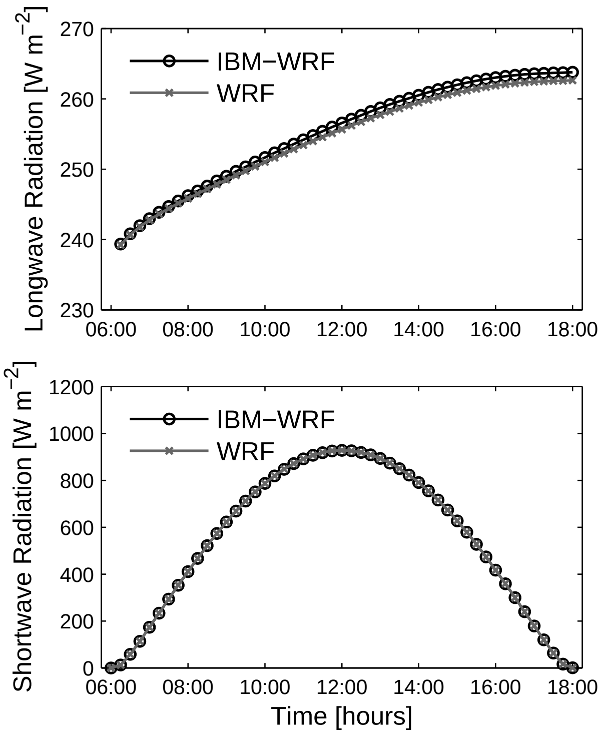


Figure 6.11. Domain-averaged downward longwave and shortwave radiation for the coupled simulations. Markers are placed at 15 minute intervals.

6.3.2 Coupled surface heating

In this section the simulations are fully coupled, meaning that the surface fluxes are calculated by the radiation and land-surface schemes. The domain set-up is as described for the cases with specified heating, however, in the coupled cases soil properties are also initialized. The valley is located at 36° North, 0° East. The simulation date is March 21, 2007, and the period is again from 6:00 to 18:00 UTC. The soil is initialized with uniform properties, with a soil type of sandy loam, vegetation type Savannah, and vegetation fraction 0.1. The soil temperature is initialized at equilibrium with the atmospheric temperature at the surface, and soil moisture is set at a 20% saturation rate. Four vertical levels are used in the soil model with depths of 0.1, 0.3, 0.6, and 1 m for a total depth of 2 m. The Rapid Radiative Transfer Model (RRTM) is used for longwave radiation, and the MM5 (Dudhia) scheme is used to model shortwave radiation. The MM5 surface layer model is used along with the NOAH land-surface model. As described in section 5.5 these schemes have been modified so that they are coupled to the immersed boundary.

Domain averaged longwave radiation varies slightly between the two simulations., as seen in figure 6.11. The WRF and IBM-WRF cases match almost exactly at the first data point (6:15), but diverge slightly during the simulation. At the end of the simulation (18:00) the difference in longwave radiation is 1.15 W m^{-2}, or 0.44%. Differences are even smaller for

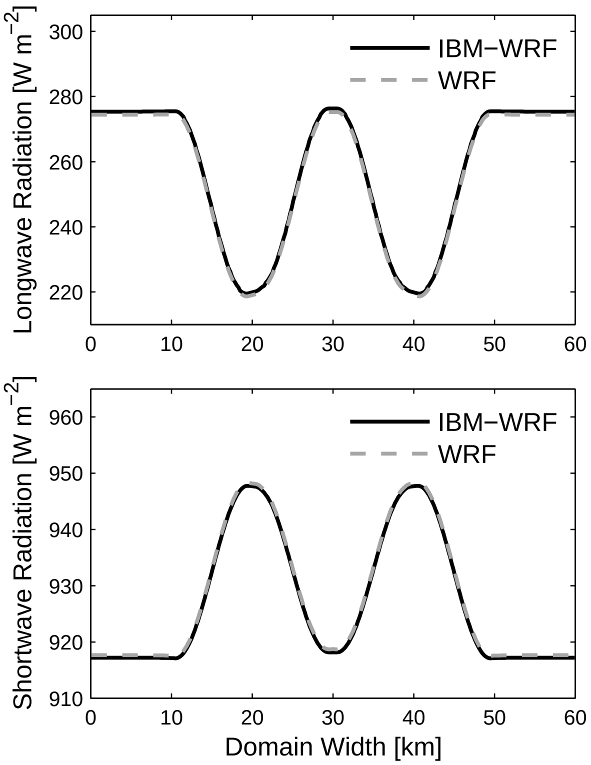


Figure 6.12. Instantaneous spatial variation in downward longwave and shortwave radiation at 12:00.

shortwave radiation, where nearly perfect results are achieved. The maximum difference between the two solutions is 0.50 W m^{-2} or 0.06%, and occurs at 14:00 UTC.

The spatial variations in longwave and shortwave radiation are documented in figure 6.12, where they are shown at 12:00 UTC. There is a difference of about 56 W m^{-2} in longwave radiation from the valley floor to peak. In comparison, at noon there is a maximum difference of 1.17 W m^{-2} in the two simulations. This shows that any error created by coupling the radiation scheme to the immersed boundary is negligible in comparison to variations due to terrain height. Shortwave radiation varies by about 31 W m^{-2} from valley peak to trough, with a maximum difference of 0.54 W m^{-2} between the simulations.

Domain averaged heat and moisture fluxes from the land-surface model are plotted in figure 6.13. The peak domain averaged heat flux occurs at 12:30. In the IBM-WRF simulation this peak is 193.80 W m^{-2}, and in the WRF simulation it is 193.31 W m^{-2}. The maximum difference between the two simulations occurs later in the day at 14:30, and is a difference of 1.10 W m^{-2} or 0.68%. Differences in moisture fluxes are also minimal. The maximum mean difference in the simulations occurs at 12:30, and is 5.39e-8 kg m^{-2} s^{-1} in magnitude, which is 0.82% of the WRF value.

Figure 6.14 shows the spatial variation in the heat and moisture fluxes at three different

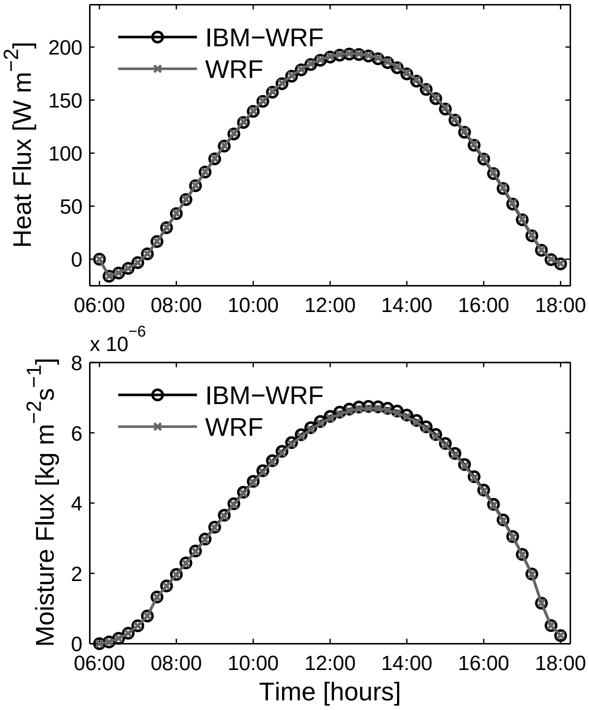


Figure 6.13. Domain averaged upward heat and moisture flux for the coupled simulations. Markers are placed at 15 minute intervals.

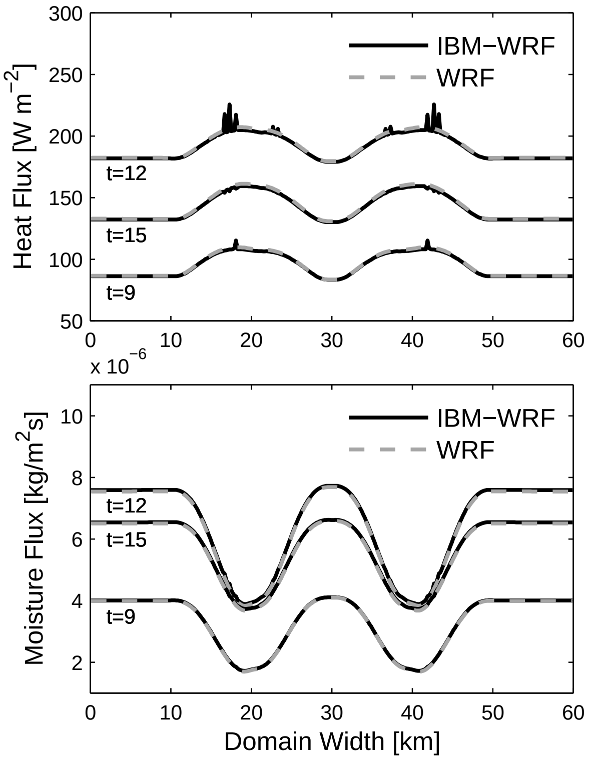


Figure 6.14. Spatial variation in upward heat and moisture flux at 9:00, 12:00, and 15:00 UTC.

Table 6.2. Maximum and minimum values of prognostic variables for the uncoupled and coupled idealized valley cases at noon. Differences are instantaneous, and calculated at the location of the indicated maximum or minimum value. Units are in m s^{-1} for velocity, Kelvin for potential temperature, and kg kg^{-1} for water vapor mixing ratio.

				Difference	
		IBM	WRF	Absolute	Relative
uncoupled	u_{min}	-2.029	-1.913	-0.116	6.1%
	u_{max}	2.029	1.913	0.116	6.1%
	w_{min}	-0.226	-0.243	0.018	7.4%
	w_{max}	1.400	1.369	0.032	2.3%
	θ_{min}	293.689	293.684	0.005	0%
	θ_{max}	316.337	316.328	0.009	0%
coupled	u_{min}	-2.011	-1.912	-0.098	5.1%
	u_{max}	2.011	1.912	0.098	5.1%
	w_{min}	-0.194	-0.215	0.021	9.7%
	w_{max}	1.198	1.253	-0.054	4.3%
	θ_{min}	293.149	293.135	0.015	0%
	θ_{max}	316.338	316.328	0.010	0%
	$q_{v\,min}$	1e-4	1e-4	2e-7	0.2%
	$q_{v\,max}$	0.002	0.002	7e-7	0%

times (9:00, 12:00, 15:00). At 12:00 UTC, spikes in the heat flux are seen near the valley peaks. The cause of this variation is the interpolation of variables to a modified reference height for use in the similarity parameters, as described in section 5.5. The modified reference height in the IBM case may lie between the immersed surface and the first grid point above the immersed surface or between the first two computational nodes above the surface. This interpolation difference occasionally leads to errors in the calculated heat flux in the IBM-WRF simulations. These spikes in heat flux do not affect the flow in this case, and plots of the velocity fields look nearly identical between the uncoupled and coupled cases. This is demonstrated in the data given in table 6.2, where the values and differences for the IBM and WRF simulations are given for the prognostic variables in both the uncoupled and coupled cases. The approximately 5% difference in u velocity is expected due to the differences in grids, application of the boundary conditions, and the added difficulty of forcing the flow at the immersed boundary rather than at a larger scale. The largest difference of 9.7% is seen in w when the values are small, and a small absolute difference leads to a large relative difference.

Land-surface models for soil moisture can be used independently of atmospheric simulations, but when coupled the top boundary conditions of the land-surface model are set with inputs from the atmosphere. When IBM is used, it provides boundary conditions to

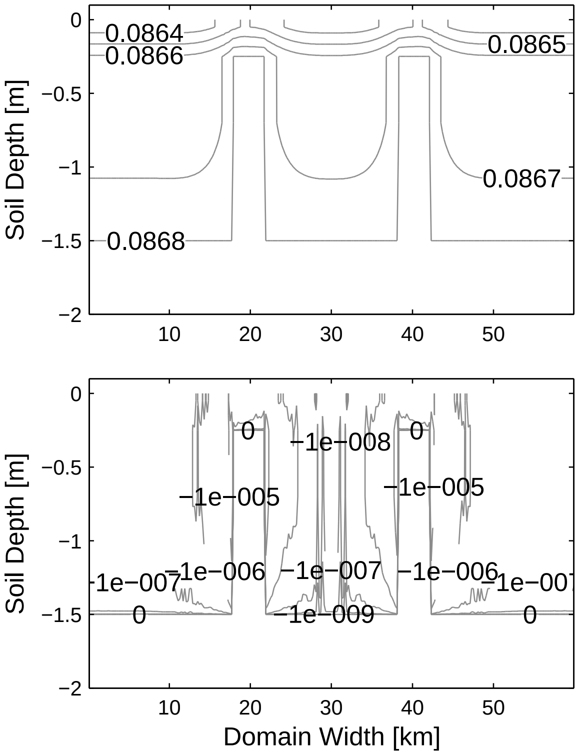


Figure 6.15. Volumetric soil moisture [m^3 m^{-3}] is plotted (top) at the end of the IBM-WRF simulation (18:00). This field was initialized with a constant saturation rate. Absolute difference between the IBM and WRF simulations for volumetric soil moisture is shown (bottom).

each model (WRF and NOAH) simultaneously. A comparison of soil moisture is shown in Figure 6.15 for the IBM-WRF and WRF simulations. At initialization the soil moisture has a constant volumetric water content of 0.0868 m^3 m^{-3}. During the simulation the soil begins to dry as moisture is transferred to the atmosphere. The depth to which this occurs is affected by the terrain height, and it can be seen that the soil remains slightly more moist at the terrain peaks in comparison to the valley floor and the terrain outside of the valley. Similar results are achieved in both of the simulations.

6.4 Two-dimensional urban terrain simulations

The IBM allows resolution of urban terrain, enabling WRF to simulate flows which cannot be computed using a standard terrain-following coordinate. To demonstrate this capability, we have modeled flow over a two-dimensional slice of downtown New York City. The tallest building that is included has a height of 248m. The domain size is $(X, Y, Z) = (698m, 4m, 600m)$ with $\Delta x = 2m$, $\Delta y = 2m$, and $\Delta z \approx 2.4m$. The total number of grid points in each direction is $(n_x, n_y, n_z) = (350, 3, 250)$. At initialization the atmosphere is at rest, and the flow is driven by a constant horizontal pressure gradient. Boundary conditions are identical to those previously used.

Figure 6.16 shows flow streamlines and the velocity magnitude at three different snapshots in time. After 30 minutes of flow spin up, recirculation regions form behind several of the buildings. By 2 hours there is a large area of recirculation between the buildings, and the flow velocities are fairly weak in these regions.

This case demonstrates the ability of our IBM algorithm to handle a wide variety of geometric cases while retaining numerical stability. Several configurations of ghost points and nearest neighbors are tested in this case, which were not present in the idealized hill and valley cases. As the isobaric coordinates evolve it can be challenging to maintain a well conditioned interpolation matrix in equation (5.7); however, this case demonstrates that we have developed a capable and robust IBM algorithm.

The focus of this work is the development and accuracy of boundary conditions at the immersed surface; however, for complex cases such as with urban terrain there are many factors which impact the accuracy of the simulation beyond the treatment of the lower boundary. Subfilter scale turbulence models and near-wall stress models play a large role in obtaining accurate solutions of flow in complex terrain. This is true for all types of grids regardless of being body or non-body conforming. Some of these additional challenges for high-resolution simulations of complex flows in the WRF model have been addressed in the work of Mirocha et al. [2007] and Lundquist et al. [2007], where the use of improved turbulence models is investigated.

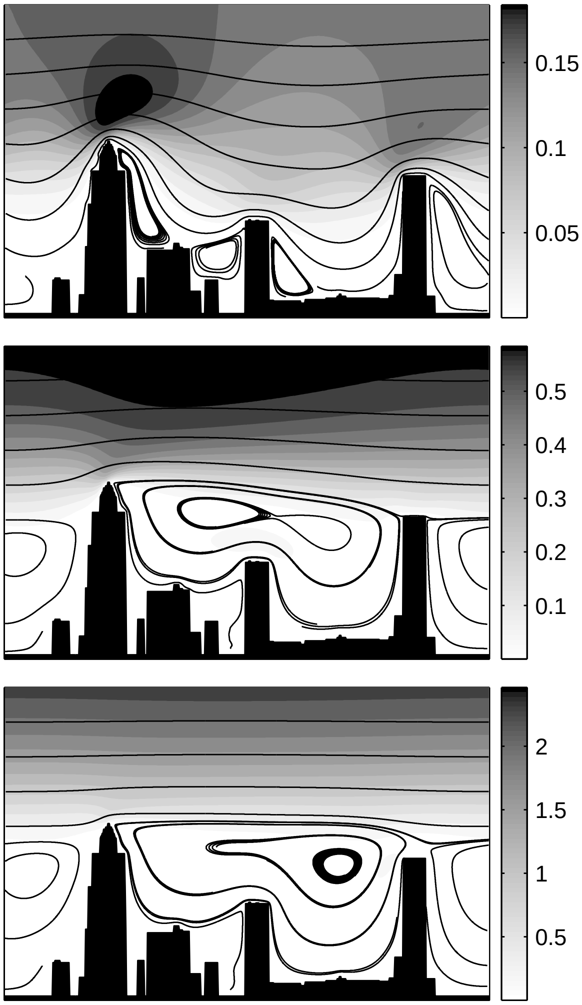


Figure 6.16. Velocity contours [m s $^{-1}$] and streamlines of startup flow over two-dimensional building data from New York City using IBM-WRF. Three different times are shown: 30 min (top), 2 hrs (middle), and 10 hrs (bottom).

6.5 Conclusions

The canonical cases of pressure driven flow over a hill and thermally driven flow in a valley were used to validate the implementation of the IBM by comparing the results to those achieved with the native WRF terrain-following coordinate. In the hill case, the domain averaged differences between the simulations at steady state were $\Delta U_{ave} = 0.017$ and $\Delta W_{ave} =$ 6e-4 m s^{-1}. In the valley case the domain averaged differences were $\Delta U_{ave} = 0.010$ and $\Delta W_{ave} = 0.002$ m s^{-1} (uncoupled), and $\Delta U_{ave} = 0.011$ and $\Delta W_{ave} = 0.002$ m s^{-1} (coupled) when averaged over the duration of the daytime diurnal cycle.

These errors are remarkably small when considering that there were unavoidable differences in the grids and application of boundary conditions in the WRF and IBM-WRF cases. This success was only achieved after careful development of our IBM within the WRF computational framework. Considerations were made to accommodate the fully compressible Navier-Stokes equations, the transformation of these equations into time variant pressure-based sigma coordinates, and the discretization of the equations with a time-split integration scheme. A simulation of flow over urban terrain demonstrated the ability of IBM-WRF to handle highly complex terrain, which could not be simulated using sigma coordinates.

Chapter 7

Extended implementations of the immersed boundary method

This chapter describes a three-dimensional immersed boundary method (IBM) that facilitates the explicit resolution of complex terrain within the Weather Research and Forecasting (WRF) model. Two different interpolation methods, trilinear and inverse distance weighting, are used at the core of the IBM algorithm. Functional aspects of the algorithm implementation and the accuracy of results are considered. Simulations of flow over a three-dimensional hill are preformed with both WRF's native terrain-following coordinate and with both IB methods. Comparisons of flow fields from the three simulations show excellent agreement, indicating that both IB methods produce accurate results. When ease of implementation is considered, however, inverse distance weighting (IDW) is superior. Furthermore, inverse distance weighting is shown to be more adept at handling highly complex urban terrain, where the trilinear interpolation algorithm breaks down. This capability is demonstrated by using the inverse distance weighting core of the IBM to model atmospheric flow in downtown Oklahoma City. Flow in the Oklahoma City domain is simulated concurrently with an outer domain with flat terrain using one-way nesting.

7.1 Introduction

A version of our IBM suitable for two-dimensional terrain is presented in chapter 5. In this chapter, the method has been extended to accommodate fully three-dimensional terrain and is now capable of running on highly parallelized machine architectures. A key component of the immersed boundary method is the formulation of the forcing term used to impose the correct boundary condition at the immersed surface. Because the immersed surface is not coincident with the grid, an integral part of the immersed boundary algorithm is the interpolation method used in the calculation of the forcing term. In this chapter, we examine the use of two different interpolation methods. In the first interpolation method, which is trilinear, weighting coefficients are determined by inverting a Vandermonde matrix. This method is chosen here because one or multi-dimensional linear interpolation methods were the first interpolation methods to appear in the IBM literature, and are still commonly used. In the second interpolation method, weighting coefficients are determined as a function of inverse radial distance. This method provides more flexibility in both the number and location of points influencing the interpolation because the Vandermonde matrix is eliminated, thereby eliminating the constraint that a fixed number of interpolation points be used and that the matrix be well-conditioned. Each method is used in simulations of flow over a three-dimensional hill, and the results are compared to those using terrain-following coordinates. Additionally, results are presented for flow in downtown Oklahoma City using one-way nested domains.

7.2 Interpolation in three dimensions

Several different interpolation methods have been employed by researchers for the purpose of interpolating the forcing needed on the immersed boundary to the nearby computational nodes where the forcing is actually applied. Interpolation methods used with IBM have included multi-dimensional linear and quadratic interpolation [Tseng and Ferziger, 2003], inverse distance weighting [Gao et al., 2007], and Lagrange and least squares interpolations [Peller et al., 2006]. The methods presented in this chapter are unique from those used by other researchers. Our three-dimensional trilinear interpolation method shares the benefits of the two-dimensional method, which are outlined in section 5.3.1. Mainly, our method eliminates the occurrence of numerical instabilities, does not require the use of an iterative solver, and works with moving pressure coordinates. Our inverse distance weighting method requires fewer applications of the interpolation method than the method described in Gao et al. [2007], even for the same order of accuracy at the boundary. Furthermore, Gao et al. [2007] only addressed the use of Dirichlet boundary conditions, while we additionally address the use of Neumann boundary conditions in this chapter.

7.2.1 Trilinear interpolation

In trilinear interpolation the interpolant is the product of three linear functions, one in each dimension. The value of the image point is calculated using the interpolant given by equation 7.1.

$$\varphi = c_1 + c_2x + c_3y + c_4z + c_5xy + c_6xz + c_7yz + c_8xyz \tag{7.1}$$

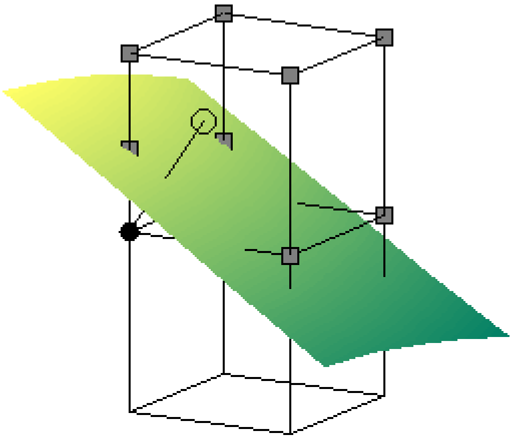


Figure 7.1. Ghost points are a layer of computational nodes just underneath the terrain. Here, a portion of terrain (the green surface) is shown with the computational cells that it cuts through. A ghost point is marked with a solid circle. An image point, marked with an open circle, is found by reflecting the ghost point across the terrain in the surface normal direction. A line connecting the ghost and image points is the surface normal. Eight neighbors marked by squares are chosen for use in determining the coefficients of the trilinear interpolant. In this case, six neighbors are computational nodes and two are located on the surface of the immersed boundary at the intersection of the boundary and one face of the cut cell.

Eight neighboring points are used to define the interpolation region, and are chosen as either computational nodes or boundary points. An example of this is shown in figure 7.1. The constants $\mathbf{c}$ in the interpolant are determined by solving a linear system of equations $\mathbf{c} = \mathbf{A}^{-1}\boldsymbol{\varphi}$ for each ghost point, where the rank is equal to the number of neighbors.

The matrix $\mathbf{A}$ and the vector $\boldsymbol{\varphi}$ are dependent on the neighbors chosen for the interpolation and the type of boundary condition being imposed. For Dirichlet boundary conditions, equation 7.1 appears in the matrix equation. If the neighbor is a computational node, then $\boldsymbol{\varphi}$ takes the value calculated at the node. If the neighbor is a boundary point, then the boundary condition is assigned to $\boldsymbol{\varphi}$. For Neumann boundary conditions, the gradient of the interpolation function is substituted into the boundary condition, and equation 7.2 results, where $\boldsymbol{n} = (n_x, n_y, n_z)$ is the unit vector in the surface normal direction.

$$\frac{\partial\varphi}{\partial\boldsymbol{n}} = c_2 n_x + c_3 n_y + c_4 n_x + c_5(n_y x + n_x y) + c_6(n_z x + n_x z) + c_7(n_z y + n_y z) \\ + c_8(n_z xy + n_y xz + n_z xy) \quad (7.2)$$

For neighbors on the boundary, equation 7.2 is used in the matrix equation, instead of equation 7.1. Once the interpolation constants are determined, the value of the image point is found with equation 7.1. As a last step, the variable value at the ghost node is calculated and assigned.

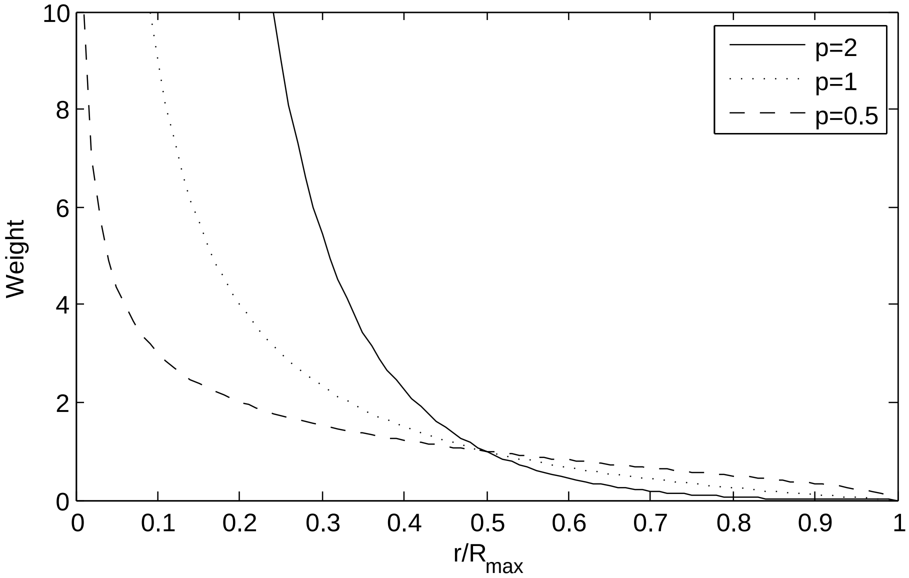


Figure 7.2. Weighting coefficients c_n for inverse distance weighted interpolation as a function of radius, normalized by the maximum radius R_{max}. Three different power parameters (p=2, 1, and 0.5) are shown here.

7.2.2 Inverse distance weighted interpolation

With the inverse distance weighted interpolation proposed in Franke [1982], the value of the image point is calculated with the interpolant given by equation 7.3, which is simply a weighted average of the neighboring points.

$$\varphi = \frac{\sum_n c_n \varphi_n}{\sum_n c_n} \tag{7.3}$$

This method was developed for geometrically scattered data sets, and any number of neighboring points can be used to define the interpolation region. Weighting coefficients c, given by equation 7.4, are a function of radial distance from the interpolation point, which in our method is the image point. In equation 7.4, R_{max} is the maximum radius from the interpolation point of the group of neighbors, and R_n and c_n are the radial distance and weighting coefficient for the n^{th} neighbor.

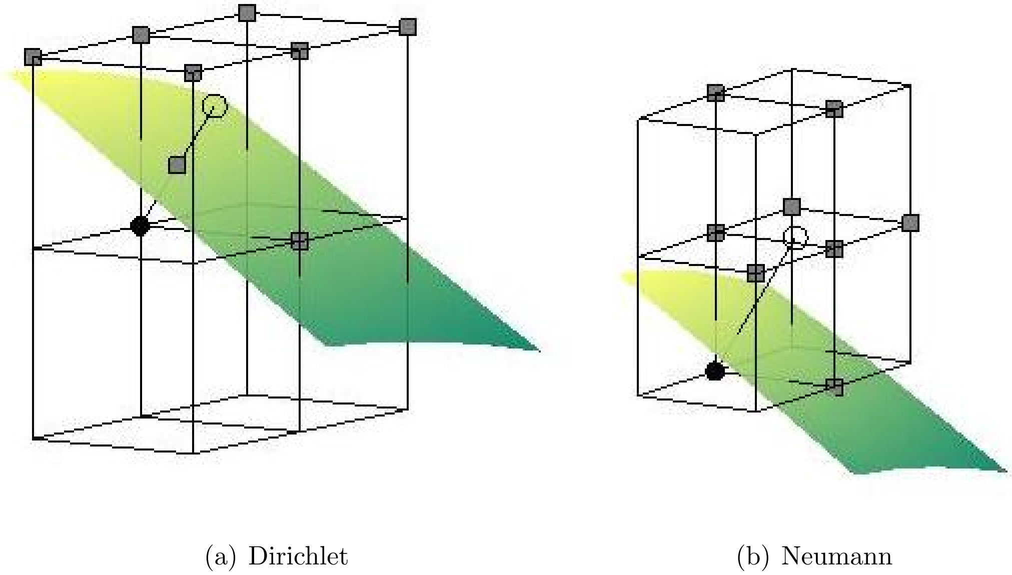
$$c_n = \left(\frac{R_{max} - R_n}{R_{max} R_n} \right)^p \tag{7.4}$$

This function, which is shown in figure 7.2, produces an infinite weight for a node that is coincident with the image point, while the node located furthest away at R_{max} is used to define the sphere of influence for the interpolation, but has no influence itself with a weighting factor of zero. The variable p is a power exponent that controls the rate of decay of the weighting coefficient with increasing radial distance. Gao et al. [2007] used a power parameter of 2. Numerical instabilities occurred for certain geometries of cut-cells in our simulations with the power parameter set to 2. These instabilities arose when the ghost cell was far from the boundary ($\approx$ 80-90% Δx), so that once reflected across the boundary the image point was much closer to the second fluid node from the boundary than the first. The image point was very near the second fluid node, so that its weighting coefficient was much larger than the weighting coefficient at the first fluid node. The second fluid node influenced the solution so disproportionately that the value at the second fluid node was nearly equal in magnitude (when using a no-slip boundary condition) to the value of the ghost point. The first fluid node essentially had no influence on the interpolation, so that its value could grow in magnitude and destabilize the entire solution. These instabilities were eliminated completely by using a parameter of either 1 or 1/2, so that each node in the interpolation contributed more equally to the interpolation. For the simulations presented here, p is set to 1/2.

In our algorithm, the first step in identifying neighbors to be used in the interpolation is searching the 64 computational nodes surrounding the image point (i.e. a distance of 2 nodes in each direction from the image point, which is not coincident with a computational node, which defines a 4^3 cloud of nodes surrounding the image point). Potential neighbors are identified as those residing in the fluid domain, eliminating those nodes in the solid domain. Potential neighbors are sorted by increasing radial distance, and eight neighbors are chosen by proximity for use in the interpolation. The particular choice of eight neighbors is arbitrary, and was chosen here because eight neighbors are used in the trilinear interpolation algorithm. Gao et al. [2007] note that they found that 3 to 4 nodes in two dimensions and 4 to 5 nodes in three dimensions provided sufficient accuracy.

When Dirichlet boundary conditions are used, the first neighbor is on the immersed boundary along the surface normal vector connecting the image and ghost points. The remaining seven neighbors are computational nodes, as illustrated in figure 7.3. Inverse distance weighting preserves maxima and minima, even during extrapolation. Therefore, it is guaranteed that the interpolated image point value will be bounded by the values of the boundary condition and neighboring computational nodes.

When Neumann boundary conditions are imposed, a boundary point is not used because the value on the immersed surface is unknown. In this case, all eight neighbors are computational nodes. Without a point on the boundary, the image point can lie outside of the interpolation region, usually when the ghost node is near the surface, as in figure 7.3. If extrapolation is used, the calculated value of the image point may not properly account for gradients in the variable field because the extrapolated value will be bounded by the values at the neighbors. In this case, the image point is modified by relocating it to be at the intersection of the surface normal with a face of the computational cut-cell, as shown in figure 7.3. Although the image point is no longer a true reflected image, the relationship $\varphi_G = \varphi_I - \overline{GI}\frac{\partial\varphi}{\partial_n}|_\Omega$ can still be used to achieve the Neumann boundary condition, where $\overline{GI}$ is the new distance between the ghost and image points and Ω represents the immersed surface.


(a) Dirichlet

(b) Neumann

Figure 7.3. Eight neighbors marked by squares are chosen for use in determining the coefficients of the inverse distance weighting interpolant. In (a) points are shown for a Dirichlet boundary condition, and in (b) points are shown for a Neumann boundary condition.

In either case (Dirichlet or Neumann), if eight or fewer potential neighbors exist, as may be the case for convex surfaces, then the sorting routine is skipped, and the algorithm proceeds using all available neighbors within the search area that also reside in the fluid domain. As with trilinear interpolation, once the value at the image point is calculated, the last step is to enforce the boundary condition by calculating and assigning the ghost point value.

Our inverse distance weighted IBM method differs significantly from the method proposed in Gao et al. [2007]. Our method retains the use of an image point, whose value is found through interpolation, and a relationship between the image point and the boundary condition is used to solve for the ghost point value. Gao et al. [2007] use the inverse distance weighting method along with a Taylor series expansion, where the origin of the Taylor series is the interpolation point, and is located on the immersed boundary (not at an image point). For Dirichlet boundary conditions the variable value on the boundary is known, so that inverse distance weighting is used to reconstruct both first and second derivatives on the boundary (origin of the series) for the first and second order terms in the Taylor series expansion. Therefore, the inverse distance weighting interpolation scheme must be applied once for each term in the Taylor series, which is five times for each ghost point for two-dimensional cases, and nine times for each ghost point for three-dimensional cases. If the calculated derivatives are colocated for a single variable, then the weighting coefficients can be calculated once, only applying equation 7.3 multiple times. Our method only applies the interpolant once, to the image point, for a linear approximation at the immersed boundary. As noted earlier, Gao et al. [2007] did not extend their method to the application of Neumann boundary conditions.

7.3 Verification case

In this section, we verify the implementation of our three-dimensional immersed boundary method in the WRF model, and evaluate the accuracy of the two interpolation methods. Verification is performed by simulating flow over a three-dimensional hill with the native terrain-following coordinate and with each of the immersed boundary methods, and comparing the results.

7.3.1 Model set-up and initialization

The test flow case is geostrophic flow over a three-dimensional hill. The terrain height h_t is defined by the Witch of Agnesi curve (equation 7.5), using a peak mountain height h_p of 691 m and a mountain half-width a of 800 m.

$$h_t(x,y) = \frac{h_p}{1 + (x/a)^2 + (y/a)^2} \tag{7.5}$$

The flow is initialized with a neutral sounding of 288 K, so that the perturbation temperature is -12 K off of the base state of 300 K. Additionally, the sounding specifies a constant velocity of 10 m s^{-1} for u, and 0 m s^{-1} for v. The flow is driven by a pressure gradient that would balance a geostrophic wind of 10 m s^{-1} in the x direction. The Coriolis parameter f is set to a constant value of 1 x 10^{-4} s^{-1}. The number of grid points in each direction is $(nx, ny, nz) =$ (60,60,91) for the terrain-following case, and $(nx, ny, nz) =$ (60,60,95) for the cases using the immersed boundary method. Four additional points are used in the vertical direction in the immersed boundary method to account for the fact that nodes are needed underneath the terrain, therefore the domain begins at $z =$ -200 m in the IBM cases (rather than at $z = 0$ m in the terrain-following case). In the horizontal dimensions, a constant 100 m grid spacing is used. In the vertical dimension, the grid points are equally spaced in physical space, so that $\Delta z = 50$ m. The top of the domain is located at a height of 4500 m. A constant eddy viscosity of 20 m^2 s^{-1} is used.

Periodic boundary conditions are used at the lateral boundaries. A no-slip boundary condition is set on velocity at the terrain surface, along with a zero flux condition on temperature. At the top of the domain, the native WRF boundary condition is used (isobaric and a material surface), with a Rayleigh damping layer that acts only on vertical velocity at the top 500 m of the domain.

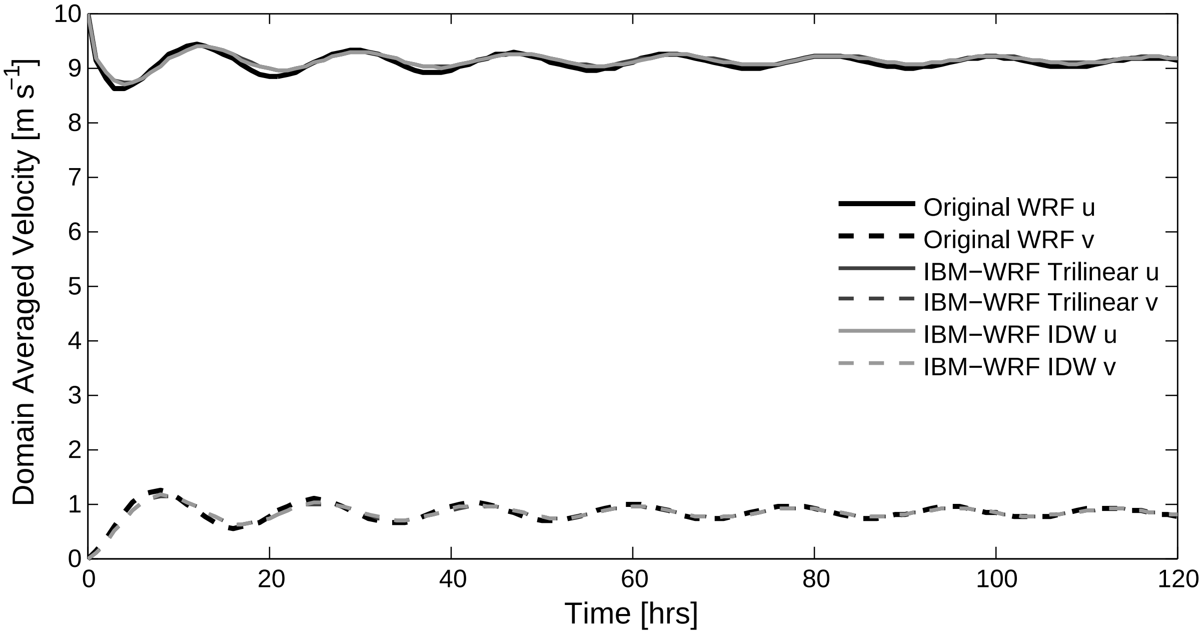


Figure 7.4. Domain averaged values of u and v plotted as a function of time for the original WRF coordinate and for IBM-WRF using trilinear and inverse distance weighted interpolations. Points underneath the terrain (for the IBM cases) are excluded from the domain average. The two IBM-WRF solutions are nearly identical, and also show good agreement with the terrain-following case.

7.3.2 Results

The flow is integrated for 120 hours, and inertial oscillations are present as the pressure gradient, Coriolis force, and surface friction terms come into balance. The damping of these oscillations are show as a function of time in figure 7.4, and on a hodograph in figure 7.5. It can be seen that the oscillations for the terrain-following and IBM cases are nearly equal in magnitude and are just slightly out of phase. The two IBM solutions using trilinear and inverse distance weighted interpolations are nearly identical, so much so that it is difficult to see both IBM lines. At 120 hours the oscillations are sufficiently damped that the solution is considered to be at a steady state. Results presented in the remainder of this section are instantaneous at a time of 120 hours.

Contours of velocity magnitude, along with quivers indicating the velocity direction are shown at a height of 400 m in figure 7.6 for each simulation. The region of separated flow in the lee of the hill is largest for the WRF simulation, and smallest for the IBM-WRF IDW simulation. Velocity contours using IBM-WRF with trilinear interpolation appear most similar to the terrain-following grid case. Profiles of each velocity component are shown in figure 7.7 for several locations along the x dimension. The locations of these slices are marked by two black lines in figure 7.6. From the perspective given in figure 7.7, the two IBM

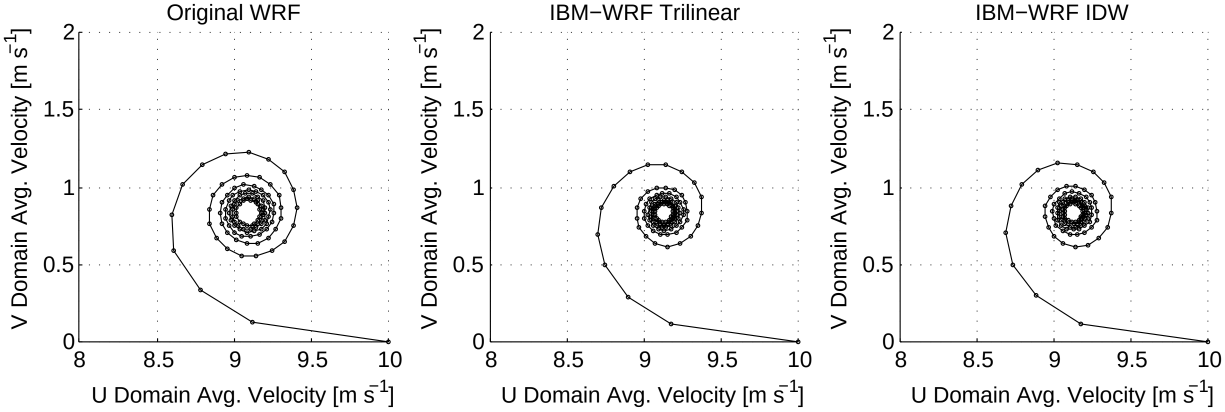
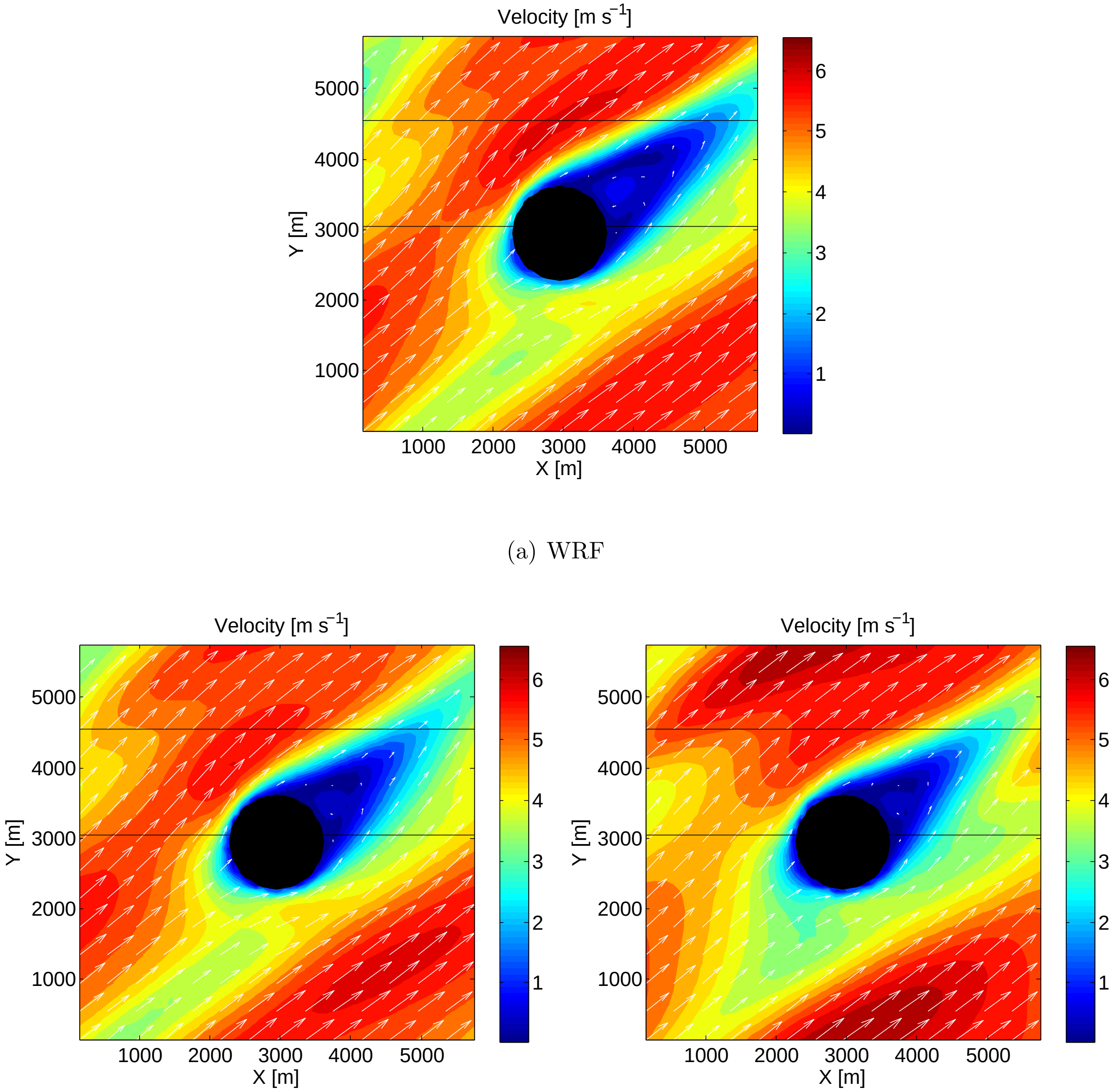


Figure 7.5. Domain averaged values of u and v plotted on a hodograph for the original WRF coordinate and for IBM-WRF using trilinear and inverse distance weighted interpolations. For our purposes the solution is considered steady state at a time of 120 hours.

solutions appear nearly indistinguishable, with the exception of the w velocity component located at $y \approx 4500$ m. In these plots the two IBM solutions tend to be identical, even when they differ from the WRF solution. This is most obvious in the profiles of v velocity, and the profile of w velocity located at the mountain peak. In the w profiles behind the hill ($y \approx 4500$ m), the peak w values seem to be captured more accurately by the IBM-WRF IDW solution than the IBM-WRF trilinear solution.

For a more quantitative comparison, the WRF and IBM solutions are interpolated onto a common time invariant terrain-following grid, so that they may be compared directly. This new grid uses the same horizontal spacing as the grid for the solution, however, the vertical grid spacing is independent of the computational grid. The IBM solution is subtracted from the WRF solution for each variable, and the minimum and maximum magnitudes of the differences in velocity are included in table 7.1. It can be seen that the minimum differences are very small, so that the solutions are nearly identical in those locations (as well as many other locations). Not surprisingly, the locations of the minimum differences are all above $z = 1100$ m, which is above the maximum terrain height. Additionally, the fact that the solutions are nearly identical in the the top three-fourths of the domain indicates that the phase difference in the inertial oscillation is negligible, and does not contribute significantly to the differences in the WRF and IBM-WRF solutions at $t = 120$ hours. The maximum differences are all located at 546 m $\leq z \leq$ 720 m, which is near the terrain peak of $h_p = 691$ m. Additionally, the maximum differences between the WRF and IBM solutions are slightly larger for the inverse distance weighted scheme, than the trilinear interpolation method.

The average height of the maximum difference included in table 7.1 is 642 m. To examine the solutions where the largest differences occur, contour plots are included in figure 7.8 at this height. Blue dots are placed at the locations in x and y of the maximum differences from table 7.1. Both sets of dots, from trilinear and inverse distance weighted interpolations,



(a) WRF

(b) IBM-WRF Trilinear

(c) IBM-WRF IDW

Figure 7.6. Contours of velocity (u, v, and w) magnitude (m s^{-1}) and quivers of velocity (u and v only) direction at a height of 400 m and a time of 120 hours for (a) WRF (b) WRF-IBM trilinear and (c) IBM-WRF IDW. The black lines indicate the locations of the velocity profiles shown in figure 7.7

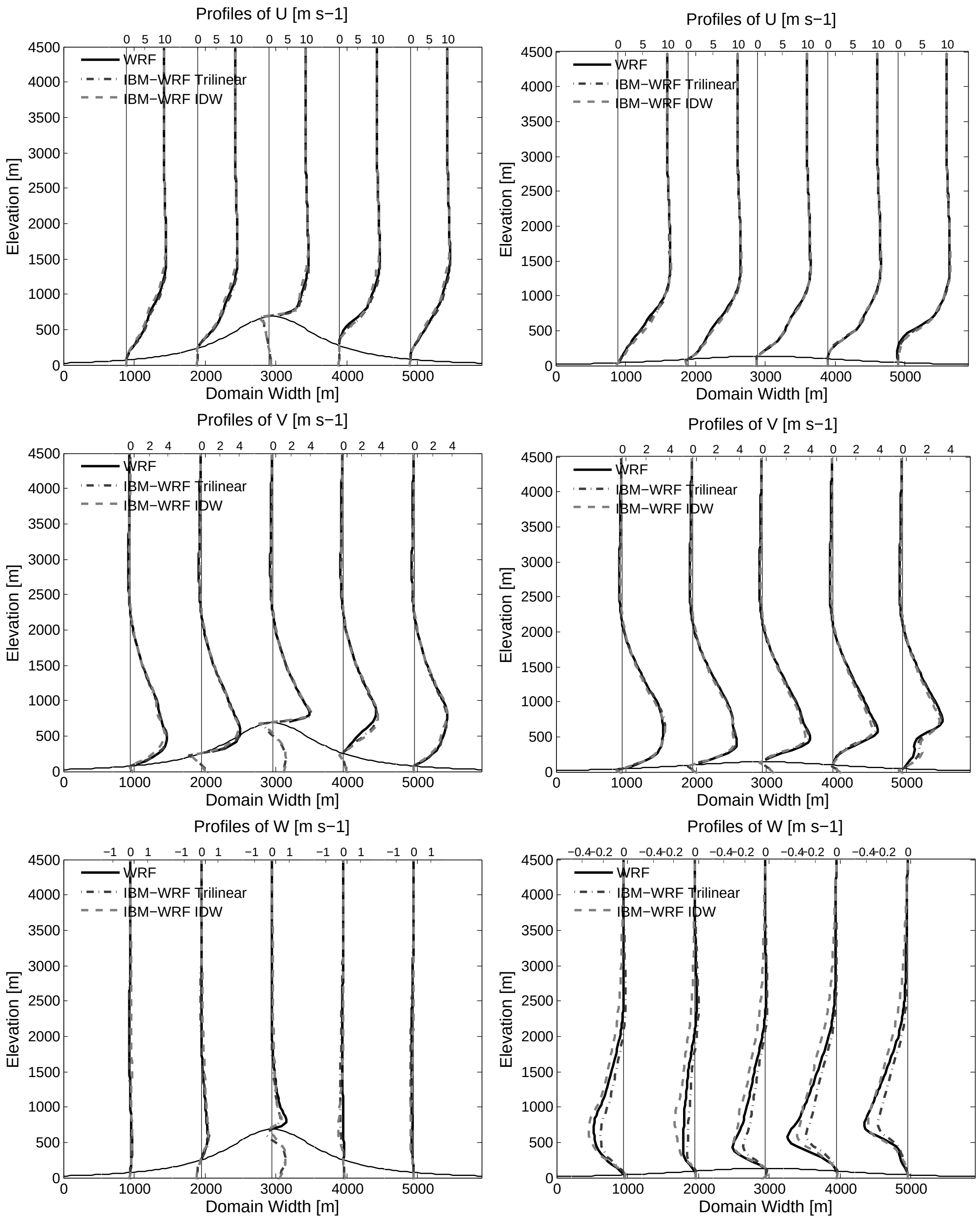


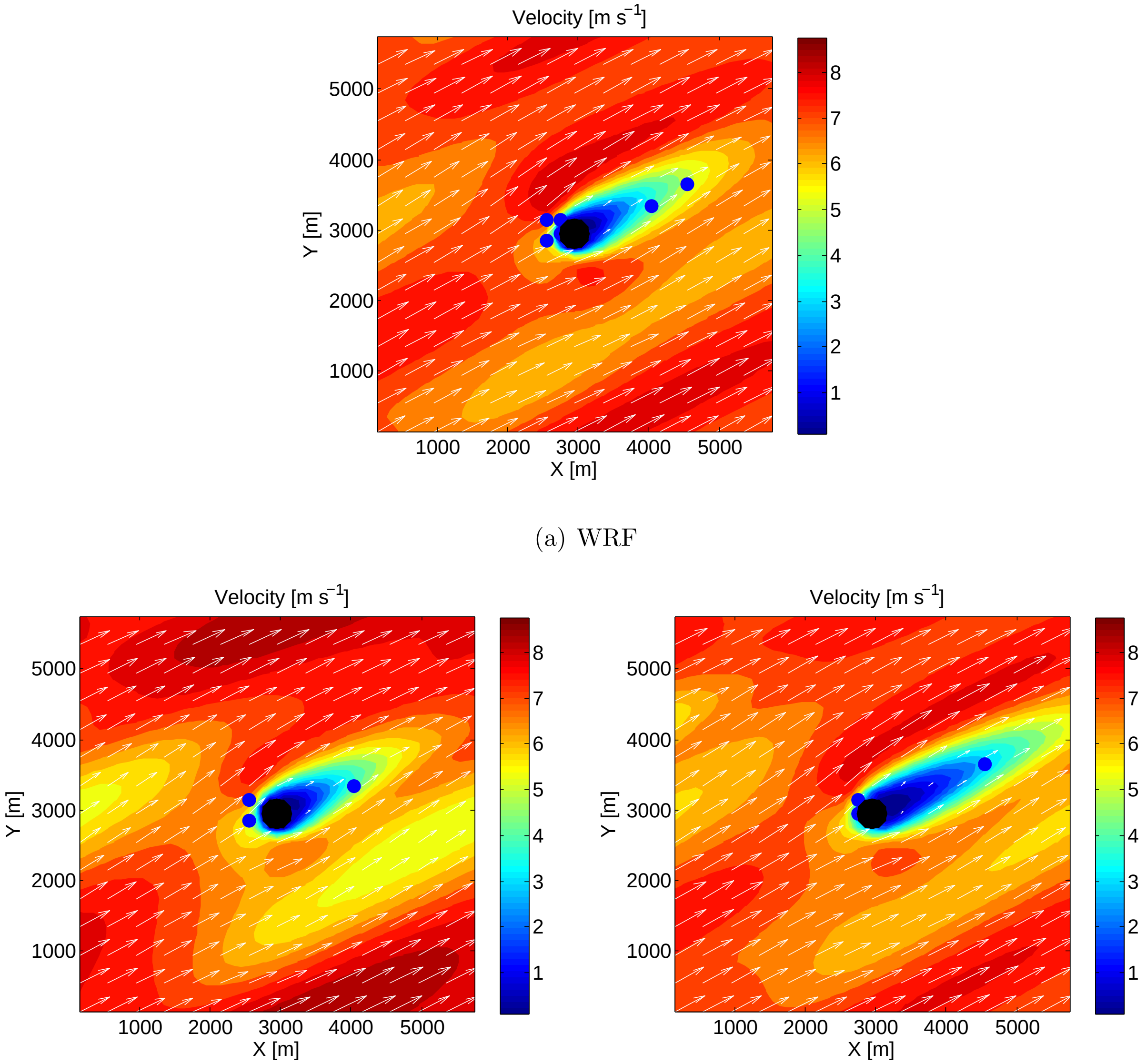
Figure 7.7. Profiles of u, v, and w velocity are shown for several horizontal locations located along a slice in the y dimension (slice locations are shown in figure 7.6). Profiles are located at $y \approx 3000$ m on the left and $y \approx 4500$ m on the right.

Table 7.1. The locations in the domain for the maximum and minimum differences between the WRF and IBM solutions for each velocity component. Velocity for each solution is given at that location, along with absolute and relative differences.

		Location [m]			Velocity [m s^{-1}]		Difference	
		x	y	z	WRF	IBM	Absolute	Relative
IBM Trilinear	Δu_{min}	750	4750	1182	10.34	10.34	7e-8	7e-7%
	Δu_{max}	4050	3350	692	2.85	4.83	1.98	69%
	Δv_{min}	5550	4350	2148	-0.04	-0.04	6e-7	2e-3%
	Δv_{max}	2550	3150	546	2.78	1.60	1.19	43%
	Δw_{min}	1750	750	3121	0.03	0.03	9e-8	3e-4%
	Δw_{max}	2550	2850	660	1.55	2.10	0.56	36%
IBM IDW	Δu_{min}	550	5650	4030	10.02	10.02	3e-8	3e-7%
	Δu_{max}	4550	3650	600	1.85	3.95	2.10	114%
	Δv_{min}	1150	2150	2280	-0.02	-0.02	3e-7	2e-3%
	Δv_{max}	2750	3150	636	2.52	0.76	1.76	69%
	Δw_{min}	5250	3950	1406	-0.27	-0.27	3e-7	1e-4%
	Δw_{max}	2750	2950	720	1.42	0.90	0.52	107%

are included on the figure for the solution using terrain-following coordinates. The dots in the wake of the hill are where the largest differences in u occur, the dots in front of the hill and offset in y are for v, and the dots almost directly in front of the hill are for w. The maximum differences between the WRF and IBM solutions occur in the same regions for both interpolation methods, indicating that the behavior of the two IB methods is similar. Furthermore, while the extent of the wake region behind the hill appears most similar to the WRF solution using IBM-WRF trilinear, the other velocity contours appear most similar using IBM-WRF IDW. Velocity profiles shown in figure 7.7 are representative of the regions which differ most. While the largest differences in table 7.1 seem large, the velocity profiles in figure 7.7 put these numbers in perspective and show that the solutions are nearly identical.

Contours of velocity differences are included in figure 7.9 for each velocity component. These slices are located at the average y value for the maximum velocity difference of the two methods presented in table 7.1, with each velocity component considered separately. It can be seen clearly here that the maximum differences for the two IBM methods occur in similar locations, and that the behavior of the two interpolation methods is similar. The main difference between the WRF and IBM solutions is the height of the shear layer in the velocity profile just behind the hill. The gradient in the velocity profile is very steep in the shear layer, so that a slight difference in the height of this flow feature causes a large difference in the velocity at a fixed location.

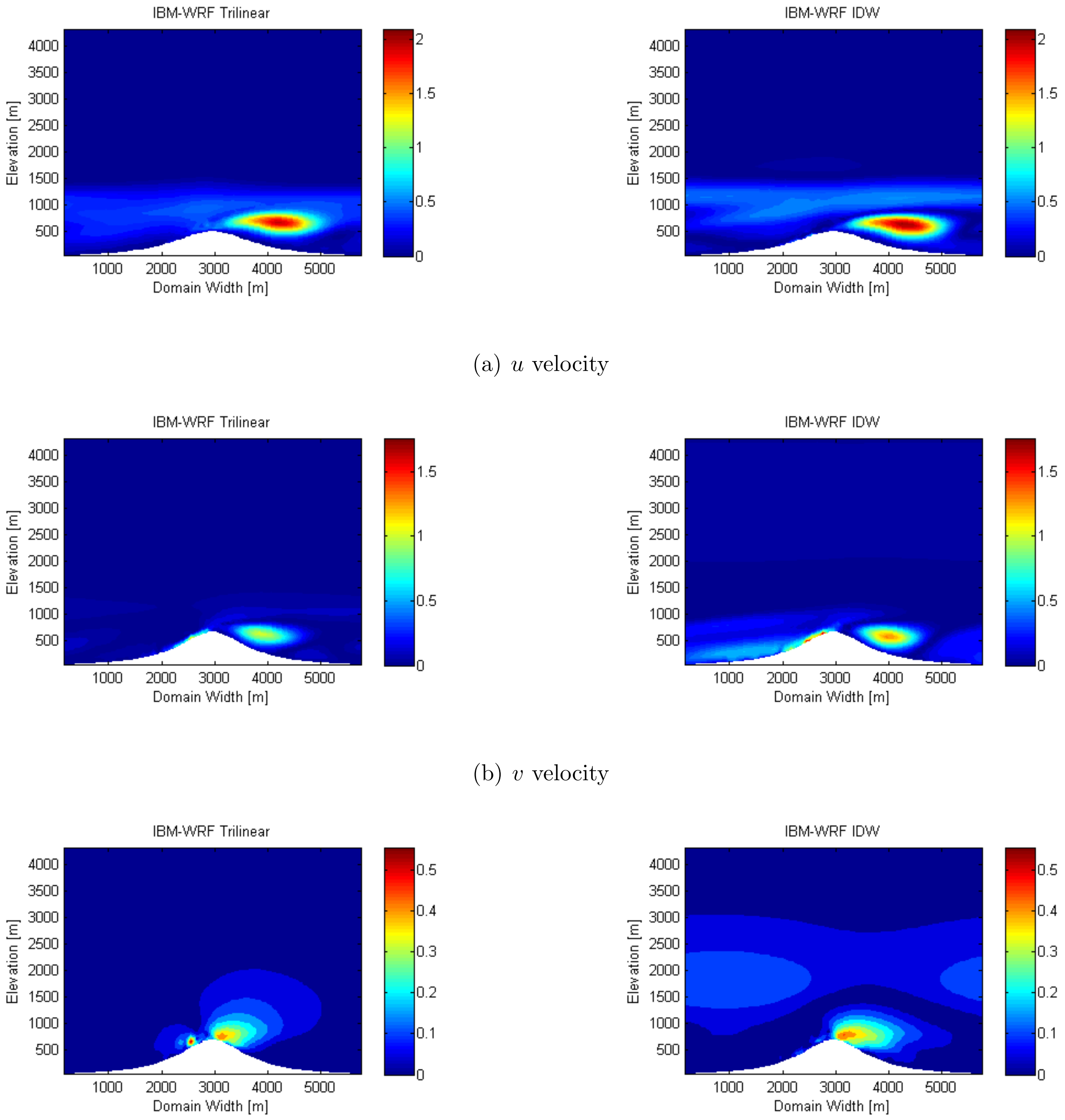


(a) WRF

(b) IBM-WRF Trilinear

(c) IBM-WRF IDW

Figure 7.8. Contours of velocity (u, v, and w) magnitude (m s^{-1}) and quivers of velocity (u and v only) direction at a height of 642 m, which is approximately where the largest differences in the solutions occur. Blue dots indicate the (x, y) location of the largest differences.



(a) u velocity

(b) v velocity

(c) w velocity

Figure 7.9. Contours of absolute velocity differences in units of m s^{-1} between the WRF and IBM solutions for each velocity component, located at the location in y where the differences are the largest. IBM-WRF Trilinear is shown in the left column, and IBM-WRF IDW is shown in the right column.

Table 7.2. Domain averaged differences and velocity values for each velocity component. All quantities have units of m s^{-1}.

		$\Delta\varphi_{ave}$	φ_{IBMave}	φ_{WRFave}
IBM Trilinear	u	0.07	9.14	9.09
	v	0.05	0.82	0.80
	w	0.02	3e-5	5e-4
IBM IDW	u	0.14	9.12	9.09
	v	0.08	0.82	0.80
	w	0.05	3e-3	5e-4

Domain averaged differences between the WRF and IBM solutions, as well as domain averaged velocity values are included in table 7.2 for each velocity component. When the data is viewed in an averaged sense, the differences are larger for IBM-WRF IDW than for IBM-WRF Trilinear. While the differences for IBM-WRF IDW are approximately twice that of the trilinear method, all of the differences are relatively small. It should be remembered that errors due to the use of terrain-following coordinates are present in the original WRF solution, and although we are comparing our IBM solution to the WRF solution, it cannot be considered the ‘exact’ solution. The terrain is relatively steep in this case, with a maximum slope of 30 degrees. Furthermore, it is possible that the IDW method can be improved through further refinement of the method. While IBM-WRF IDW appears to be the least favorable of the two IBM options for this flow case, the method has additional favorable properties, discussed earlier. For the urban geometry in the following section, inverse distance weighted interpolation worked, while trilinear interpolation did not. For that reason, the IDW core of the IBM method is used in section 7.4.

7.4 Flow in urban environments

The Joint Urban 2003 field campaign took place in Oklahoma City over a period of one month, and is detailed in Allwine and Flaherty [2006]. Over 20 institutions participated in the study, providing an extensive data set of urban atmospheric flow and dispersion. During the campaign, the city was instrumented with lidars, sodars, radars, sonic anemometers, aircraft-based meteorological sensors, fast-response tracer analyzers, and helicopter-based remote tracer detectors. This data provides an excellent test case for verifying the use of IBM in a numerical weather prediction model. In preparation for simulating intense observation periods (IOPs) from this field campaign, we have modeled flow over a portion of Oklahoma City using an idealized model set-up. Our set-up utilizes the one-way nesting capabilities in WRF, and additionally uses a large-eddy simulation (LES) turbulence closure. Both features are new to our IBM-WRF implementation.

7.4.1 Model set-up and initialization

In this simulation, a domain with the Oklahoma City terrain is nested within a parent domain with flat terrain. Flow is driven in the parent domain with a pressure gradient in the y direction. Periodic lateral boundary conditions are used, so that the flow on the parent domain is fully developed turbulent channel flow over flat terrain. One-way nesting is used in a concurrent simulation run, so that boundary conditions are passed from the channel flow domain to the Oklahoma City domain at the frequency of the parent domain time step. This set-up is similar to that used in Golaz et al. [2009] to model flow over Askervein hill. Our decision to use nested domains has two main purposes. First, it provides fully developed turbulent inflow conditions to the Oklahoma City domain, and eliminates the need for periodic boundary conditions which are only appropriate for surface conditions that are homogeneous or periodic. Secondly, the use of one-way nesting allows us to examine how IBM works with nested domain configurations. Here terrain that is not resolved (or parameterized) on outer domains is explicitly resolved on the inner most domain using IBM.

One alternative for lateral boundary conditions is the perturbation recycling method of Mayor et al. [2002], where mean conditions are perturbed at the inlet with turbulent fluctuations that have been 'recycled' from fully-developed downstream flow. A second alternative, used in Chow and Street [2009], is to perform a separate LES of turbulent channel flow with periodic boundary conditions, and then impose this solution at the domain inlet at each time step. Our method is similar, in that solutions on two domains with separate topography are produced, but has additional benefits. First, our method is simple to implement in models with nesting capabilities. Second, our method is independent of the inflow direction. One possible drawback of our method is that the outlet flow (and the flow on all lateral edges) is forced to match the turbulent channel flow conditions. This condition is not imposed in the set-up of Chow and Street [2009], where zero-gradient boundary conditions are used at the outlet, allowing flow features such as wakes to exit the domain.

A two-dimensional array of terrain data for the Oklahoma City case is created by overlaying the horizontal WRF grid for the nested domain with an ERSI (Environmental Systems Research Institute) shapefile of the downtown region. A shapefile spatially describes geometries as sets of points, polylines, or polygons. For the Oklahoma City shapefile, the buildings are defined as sets of overlapping polygons, where each polygon is assigned a unique height. Once the WRF grid is overlaid on the shapefile, the building heights that are coincident with the nodes on the WRF grid are sampled to create a two-dimensional array of terrain heights. The shapefile data and resulting three-dimensional terrain are shown in figures 7.10 and 7.11.

The parent domain is initialized and spun-up for several hours before the inner domain is started. At initialization the atmosphere is neutral with a constant v velocity of 10 m s^{-1}. The flat plate, located at a height of 0 m, is represented with the inverse distance weighted immersed boundary method. Terrain-following coordinates can also be used on the parent domain, in lieu of the immersed boundary method. The ability to nest an IBM domain within terrain-following coordinates has been implemented, but is not used in this case, which is an

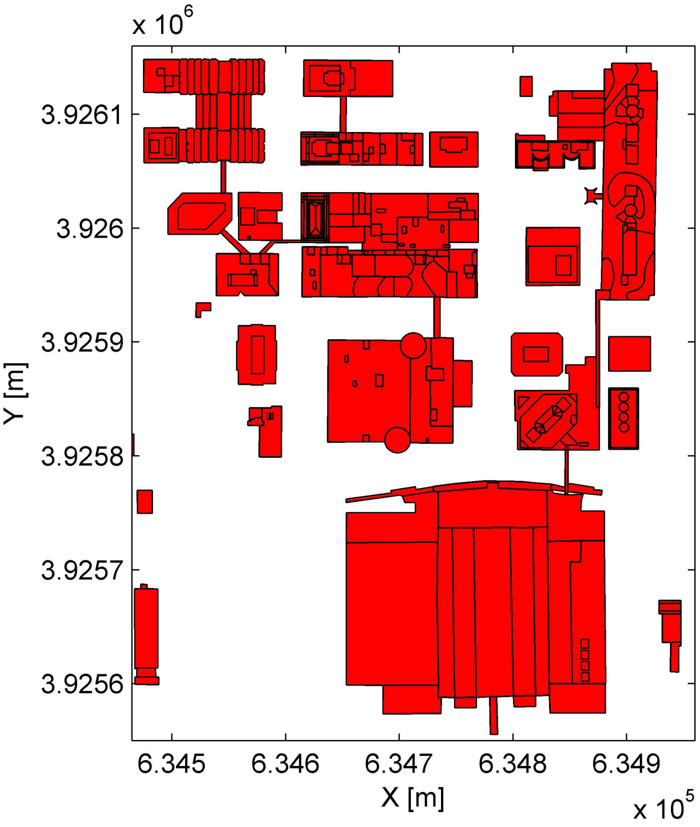


Figure 7.10. ERSI shapefile data for the buildings included in the Oklahoma City domain.

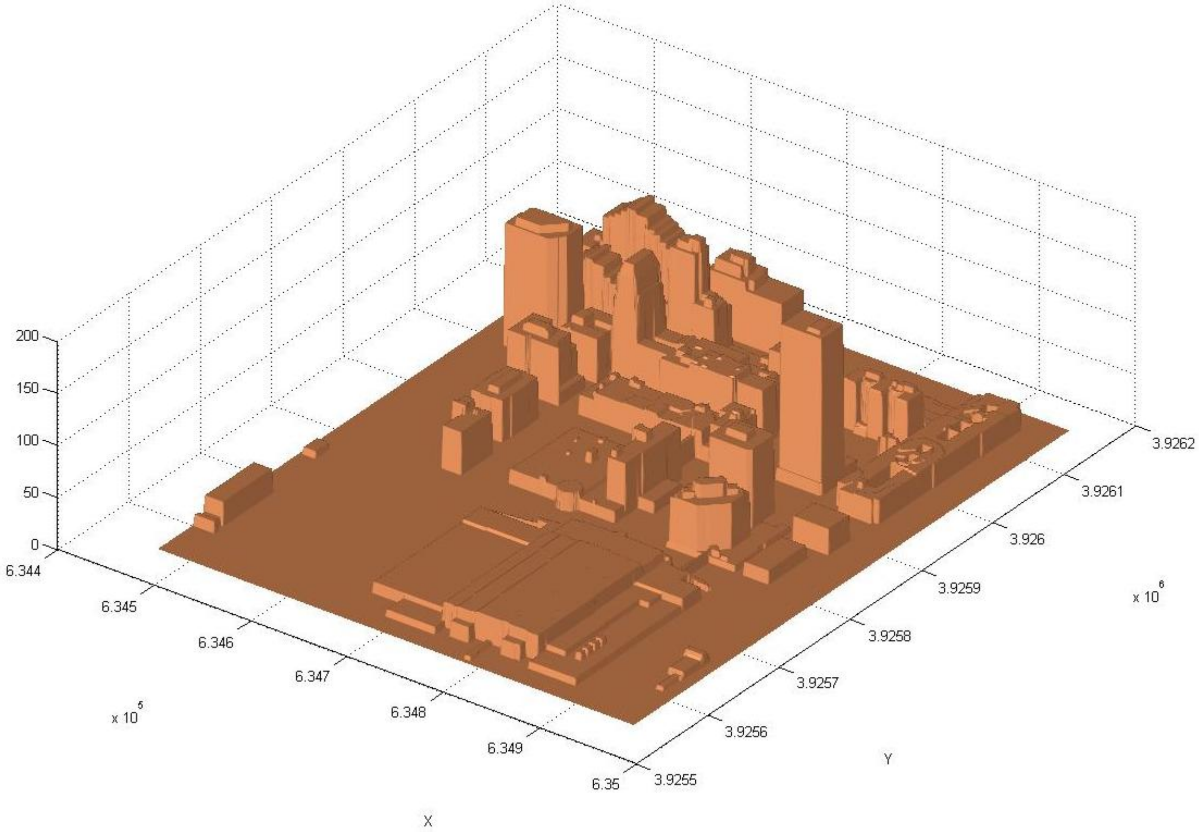


Figure 7.11. A two-dimensional array of terrain heights sampled from the ERSI shapefile is used to define the terrain used in the IBM-WRF simulation.

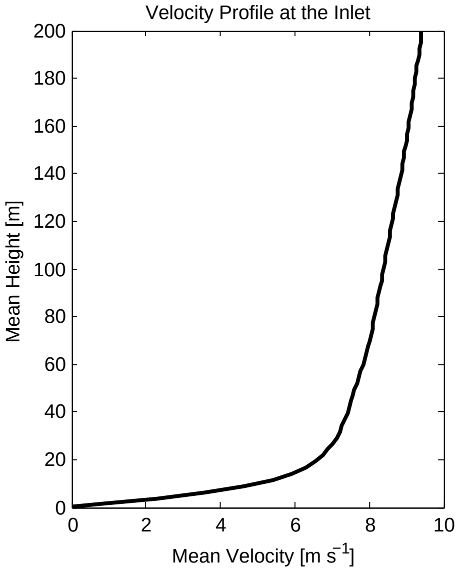


Figure 7.12. The domain averaged velocity profile for the parent domain at steady state.

IBM domain nested within another IBM domain. Flow is driven with a constant pressure gradient in the y direction, and periodic lateral boundary conditions are used. The number of grid points in each direction is (nx, ny, nz) = (96,123,170). In the horizontal directions, a constant 6 m grid spacing is used. In the vertical dimension, the grid points are equally spaced in the η pressure coordinate over the domain height of 435 m, which spans z = -10 to 425 m. The minimum vertical grid spacing is $\Delta z = 2.53$ m, and the maximum is $\Delta z = 2.62$ m. These dimensions result in a total domain size of 570 x 732 x 435 m^3. The Runge-Kutta time step is $\Delta t = 1/20$ s, with six acoustic time steps per Runge-Kutta step. A Rayleigh damping layer on w only is used at the top 40 m of the domain. The standard Smagorinsky turbulence model in the WRF distribution is used, and Coriolis forcing is neglected.

Small perturbations were added to the horizontal velocities at initialization, so that a turbulent flow would develop on the parent domain. Nonetheless, the flow was not turbulent after three hours of simulation. Therefore, a square ridge was added in the spanwise direction, to trip the flow. This ridge was subsequently removed after approximately 30 minutes, and the turbulent channel flow was spun-up until the mean velocity profile was steady (shown in figure 7.12). A larger pressure gradient is needed to drive the turbulent flow, once it transitions from being laminar. Therefore, several trial-and-error iterations were performed of modifying the pressure gradient, and checking the resulting mean profile, until the pressure gradient and frictional effects were in balance and the flow was steady in a mean sense.

A grid nesting ratio of 1:3 is used, so that the nested Oklahoma City domain has a horizontal resolution of $\Delta x = \Delta y = 2$ m, and a time step of $\Delta t = 1/60$ s. Nesting is not used in the vertical, and cannot currently be used in WRF for concurrent simulations. Nesting can be used for serial simulations; however, running in a serial mode severely limits the frequency at which lateral boundary conditions are updated on the nested domain. With our small time steps (fractions of a second), the simulations must be run concurrently to effectively pass information on the time scale of turbulent fluctuations between domains. The nested domain has (nx, ny, nz) = (259,340,170) grid points, for a total size of 516 x 678 x 435 m^3. The nested domain starts on the fifth grid point of the parent domain in both the x and y directions. Our parent domain is only slightly larger than our nested domain because this simulation is being done for demonstration purposes, and we wanted to minimize the required computational effort as much as possible. When the nested domain is started, the solution from the parent domain is interpolated onto the nested grid. The inverse distance weighted IBM is used to represent the Oklahoma City terrain, and the velocities within the solid domain are set to zero when the nest is initially spawned. Additionally, the IBM routines are used to impose no-slip boundary conditions on the buildings before integration of the nest begins.

7.4.2 Results for flow through urban terrain

Figure 7.13 shows contours of v velocity from a top view for the parent and nested domains, 18 minutes after initialization of the inner nest. This view is at a height of approximately 9 m, and the extent of the inner nest is depicted by black dashed lines on the outer domain. On the inner nest, the solution is masked by buildings with heights greater than the height of the plane shown. In this figure, the inlet flow is considered to be the southern edge of the domain. Here, it can be seen that the velocity features from the outer nest also appear at the inlet of the inner nest. At the outlet (the northern domain edge), the solution on the inner domain is forced to match the solution on the parent domain, even though there is clearly a wake region behind the northern most buildings that is being truncated. This mismatch of velocities and pressure gradient contaminates a portion of the solution near the outlet of the Oklahoma City domain. This may be acceptable because the scalar release in the IOP that we plan to simulate (IOP 3) is located in the southern portion of the domain, at approximately (x, y) = (150 m, 200 m) on the inner nest, but future investigation is required.

Figure 7.14 shows contours of velocity magnitude and quivers indicating flow direction in the plane shown for several locations within the domain. Many flow features are present in the simulation, including high speed jets at contractions of urban canyons and separation zones behind buildings.

This case was simulated using the inverse distance weighting core of the immersed boundary method. We were unable to successfully use trilinear interpolation with this terrain data. Common problems were the inability to find eight appropriate neighbors and ill-conditioned Vandermonde matrices. Additionally, in some cases (such as at corners) the direction of the flux boundary condition is ambiguous or prescribed in an unintended direction.

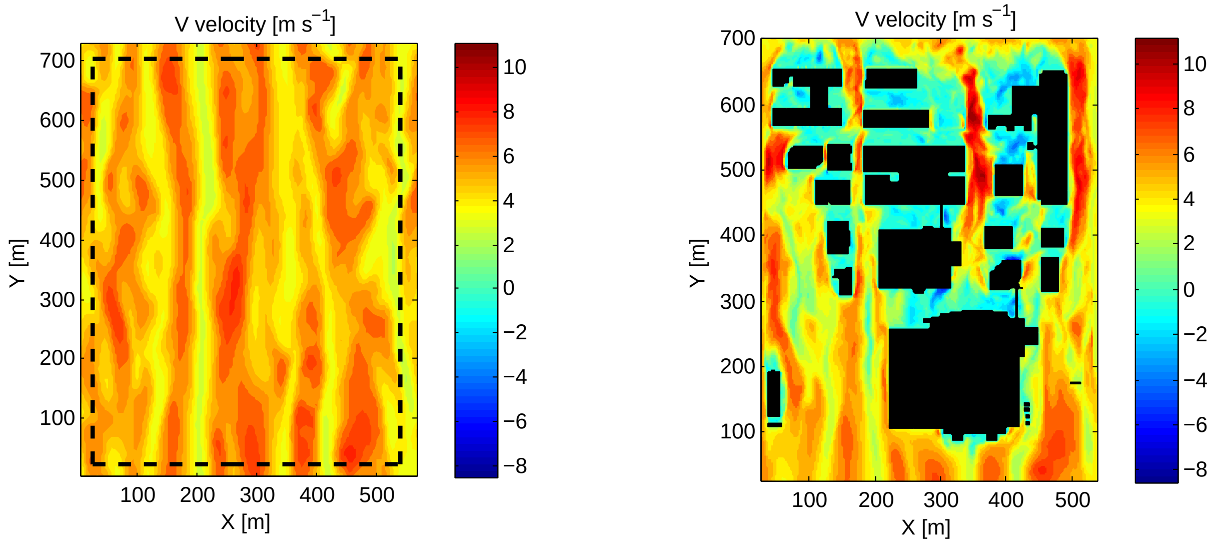


Figure 7.13. The v component of velocity for the outer nest (left) and the inner nest (right) at a height of $\approx$ 9 m. The dashed line in the outer domain indicates the horizontal extent of the inner nest.

7.5 Conclusions

We have developed an IB method for the WRF model which is capable of handling highly complex urban terrain, as demonstrated by our semi-idealized Oklahoma City test case. We first extended the two-dimensional IB method presented in Lundquist et al. [2010] into three dimensions, and validated the implementation by simulating flow over a hill and comparing the solution to results achieved using the native terrain-following coordinate. Additionally, a new IBM method was developed using an inverse distance weighted interpolation method. This second method was also validated with the canonical case of flow over a three-dimensional hill. We found that while the trilinear interpolation algorithm provided accurate results for flow over a smooth hill, the algorithm was not robust enough to be used with real urban terrain. The alternative method based on inverse distance weighted interpolation provided additional flexibility, and was also shown to produce accurate results for the hill test case. Additionally, the method proved to be robust enough to allow simulations of flow over real urban terrain data.

Additional work is in progress to enable comparisons with the Joint Urban 2003 field campaign. First, the pressure gradient used to drive the flow in the parent domain was chosen arbitrarily. Velocity data (magnitude and direction) is available for several locations in the Oklahoma City domain, including the scalar release site for IOP 3. Our plan is to modify the pressure gradient in the parent domain, to achieve the closet possible velocity match at the scalar release site in the inner domain. Additionally, some functionality must be added for using passive scalars in WRF. While we currently have the ability to add a passive scalar field at initialization, we must add the ability to simulate a continuous source release. Furthermore, while nested lateral boundary conditions are desired for the inner

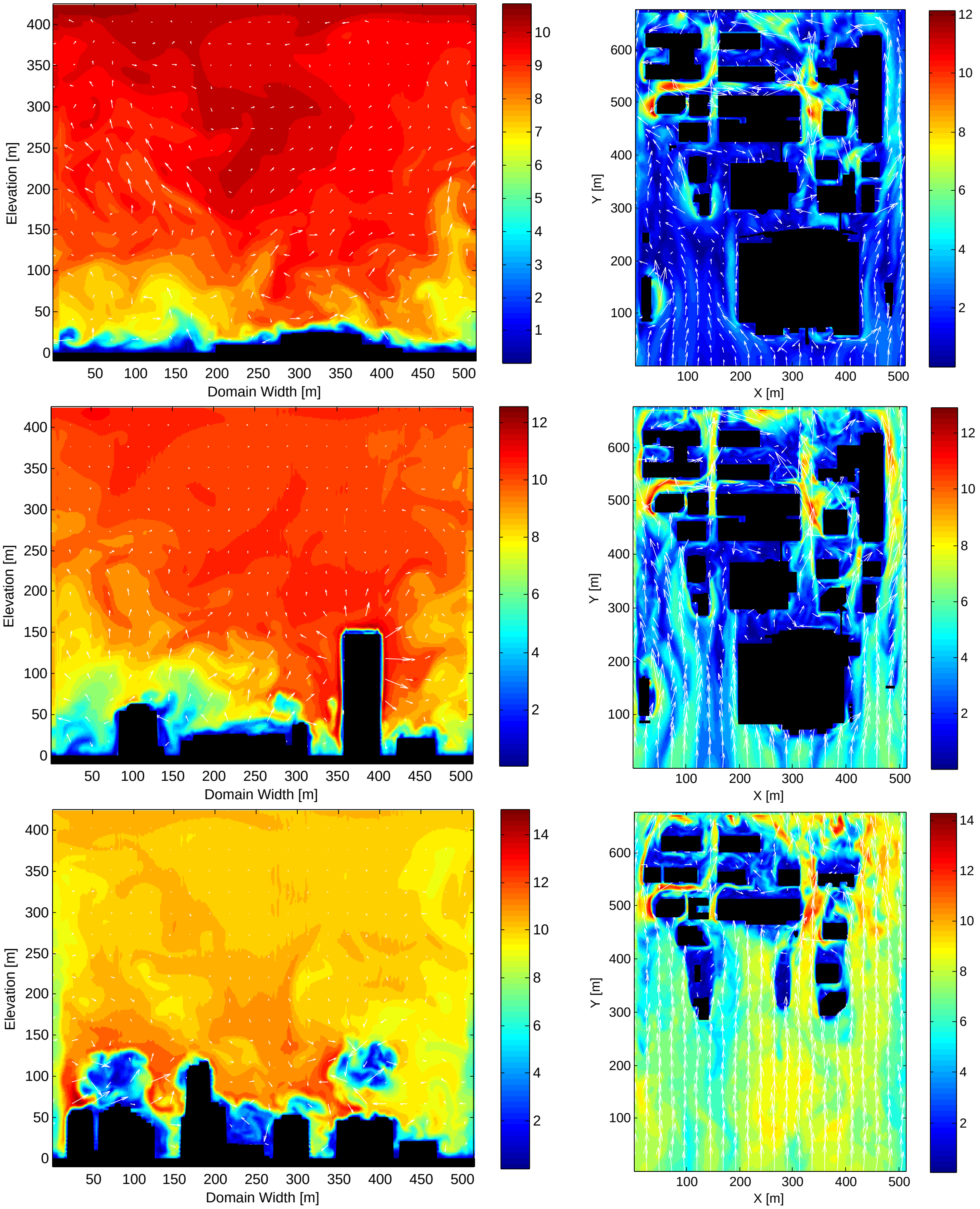


Figure 7.14. Contours are shown for velocity magnitude (m s^{-1}) and quivers indicate flow direction. Side profiles are shown on the left ($y \approx =$ 200, 440, and 560 m), and top views on the right ($z \approx$ 5, 10, and 50 m) for several locations in the domain.

nest on velocity and most other prognostic variables, they are not desired for the passive scalar. The ability to use open boundary conditions for the passive scalar, while using nested conditions for other variables must be added.

The simulation of flow over the Oklahoma City terrain in a one-way nested configuration demonstrates the ability to seamlessly integrate the IBM method into the current WRF framework, enabling simulation of a wide variety of cases with steep terrain. Resolution of the issues detailed above will allow comparisons with observations from the Joint Urban 2003 field campaign, providing further verification of the method.

Chapter 8

Wall modeling at the immersed boundary

8.1 Introduction

Three different surface treatments are examined at the immersed boundary, with the goal of developing an IBM method that enforces an approximate boundary condition (wall model) for use with coarse large-eddy simulations of high Reynolds number flows. As discussed in chapter 2, the vast majority of literature written on IB techniques for rigid bodies focuses on satisfying a no-slip boundary condition. For high Reynolds number flows, such as those in the atmospheric boundary layer, this means the thin viscous sublayer should be resolved. This is of course an impractical computational requirement, and various techniques have been used to overcome this difficulty. In CFD codes, the mesh is often locally refined near the surface where IBM is used if the boundary is stationary. In the case of moving boundaries, adaptive mesh refinement can be coupled with direct forcing IBM. Neither of these two approaches have been used successfully for atmospheric boundary layer flows. A third approach, which is commonly used in atmospheric applications, is wall or surface modeling. In this approach, a coarse grid is used and the effect of turbulent stresses on the outer flow is parameterized through the use of a wall model treatment at the boundary.

When the no-slip boundary condition appearing in chapters 5 through 7 is used, surface stresses will be under-predicted for high Reynolds number flows. In this chapter, two implementations of an IBM combined with an equilibrium wall model are developed and tested, and comparisons are made to results using the no-slip boundary condition. The first method is not a ghost-cell IBM, but is instead based on velocity reconstruction as in Fadlun et al.

[2000], where the IBM forcing is included in the Navier-Stokes equations at the first point in the fluid domain. This is combined with the log-law modifications of Senocak et al. [2004]. In the second method shear stress is reconstructed at a ghost cell, which is a blending of the original WRF boundary conditions with a ghost cell IBM. Details of the three implementations are given in the following sections, and in section 8.3 they are applied to the case of a neutral boundary layer.

8.2 Wall model implementations

8.2.1 No-slip boundary condition

For the neutral boundary layer case the surface is aligned with the coordinates, and the version of the no-slip IBM boundary condition has been simplified for use in this case. Here, one dimensional linear interpolation as proposed by Iaccarino and Verzicco [2003] and pictured in Figure 8.1 is used with the same accuracy that bilinear or trilinear interpolation would provide. Of course, for curvilinear terrain a more sophisticated interpolation scheme, such as those used in previous chapters, must be implemented. The linear interpolation in the vertical direction is applied to all three velocity components, and the velocity that enforces a no-slip boundary condition is set at a ghost cell. This implementation is, of course, not a wall model. This is the base case, and will a point of reference when examining the wall boundary conditions detailed below.

Additionally, in this chapter the IBM boundary conditions are enforced on the Runge-Kutta time step, rather than on the acoustic time step. In table 3.1, this is just after the acoustic loop, but before step eight where the scalar equations are advanced. This is true of all three implementations in this chapter. Updating the ghost points on the Runge-Kutta loop provides computational savings in comparison to updating them on the acoustic loop. The disadvantage is that the boundary condition is imposed on the less frequent large time step instead of the acoustic time step. This is acceptable for the work in this chapter because the implementations presented here are preliminary. Results from the neutral atmospheric boundary layer case will help us determine if each method warrants further development.

8.2.2 Log law velocity reconstruction

In Fadlun et al. [2000] the immersed boundary is modeled by reconstructing the velocities at fluid nodes. Senocak et al. [2004] suggests a scheme where the tangential flow is reconstructed at external fluid nodes using the log-law. For a neutrally stratified atmosphere over flat terrain, a log-law profile of the mean velocity is expected in the bottom ten percent of the boundary layer, which is usually 100 to 200 meters high [Garratt, 1992, Sec. 3.2]. The

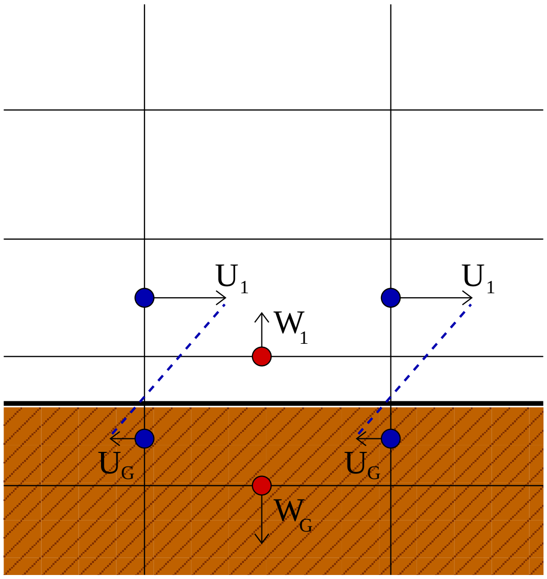


Figure 8.1. No-slip IBM.

log-law velocity reconstruction method proposed by Senocak et al. [2004] is outlined below for atmospheric boundary layer flow over a flat plate. It would need to be reformulated and cast in terms of normal and tangential components in order to extend its use to complex topography.

The tangential velocity reconstruction for the fluid nodes begins with the log-law given by (8.1), where U is the magnitude of the velocity at height z, $u_\star$ is the friction velocity, κ is the von Kármán constant, and z_o is the roughness length.

$$\frac{U}{u_\star} = \frac{1}{\kappa} \ln\left(\frac{z}{z_o}\right) \tag{8.1}$$

Within the logarithmic layer, the friction velocity $u_\star$ is constant in the surface normal direction [Senocak et al., 2004]. Using this property, the relationship given by (8.2) can be used to reconstruct the magnitude of the velocity at the first fluid point away from the boundary based on the velocity at the node above.

$$U_1 = U_2 \frac{\ln(z_1/z_o)}{\ln(z_2/z_o)} \tag{8.2}$$

Equation (8.2) is given in terms of the magnitude of the horizontal velocity, and it must be decomposed into u and v components. In order to do this, the direction of the flow is calculated using (8.3) at the second and third fluid nodes above the boundary.

$$\theta_2 = \arctan\left(\frac{v_2}{u_2}\right), \qquad \theta_3 = \arctan\left(\frac{v_3}{u_3}\right) \tag{8.3}$$

The direction of the fluid flow θ changes with respect to height. An example of this would the the Ekman spiral that forms in the atmosphere from Coriolis forcing. Linear extrapolation

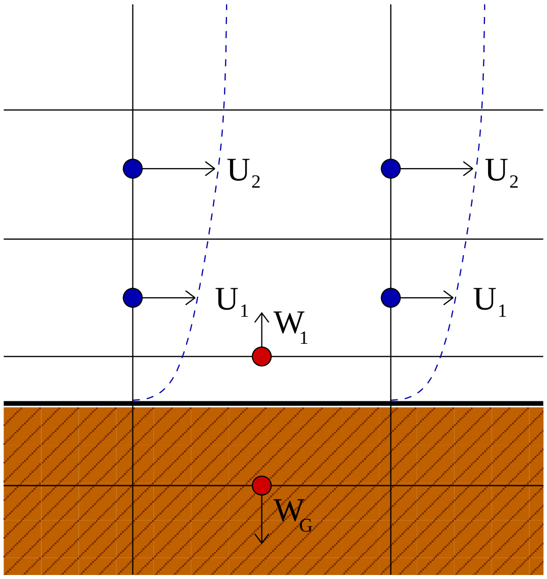


Figure 8.2. Log law velocity reconstruction IBM.

is used to determine the direction of fluid flow θ_1 at the ghost cell. Direct forcing can then be imposed at the cut cell using the relationship given in (8.4).

$$u_1 = U_1 \cos\theta_1, \qquad v_1 = U_1 \sin\theta_1 \tag{8.4}$$

The vertical velocity at the surface is zero, and direct forcing is applied for w_1 at the cut cell using linear interpolation (see figure 8.2). This satisfies the boundary condition $w_{surf} = 0$ for a flat plate. However, the complete kinematic boundary condition $\mathbf{U} \cdot \hat{\mathbf{n}} = 0$ would need to be satisfied for curvilinear boundaries.

Several steps were taken to implement this method into WRF, starting with the addition of a roughness length variable z_o. Originally in WRF the user defined a coefficient of drag. Equation (8.2) requires that a roughness length be specified instead of a drag coefficient, so z_o was added to WRF for the IBM formulation (this has also been implemented to work with the sigma coordinate using equation 8.1). Next, cells cut by the boundary are determined, and the velocity nodes that are just exterior to the terrain are saved as cut cells. Equation (8.1) is valid for $z > z_o$, therefore, the node for a cut cell must be at least a distance of z_o from the immersed boundary. This distance is checked for each node, and if the distance is less than z_o the next fluid node is assigned to be the cut cell. The velocities at nodes interior to the domain are initially set to zero, but not controlled further after the iteration begins. The velocity reconstruction scheme is then applied to the cut cells. In order to use equations (8.2), (8.3), and (8.4) the u and v velocities must be collocated; however, WRF uses a staggered grid. Therefore, for a u node a four point horizontal average of v is used, and conversely for a v node a four point horizontal average of u is used. Finally, the vertical velocity w is set to zero on the immersed surface using linear interpolation. A slight deviation from the Senocak et al. [2004] paper is made with respect to setting the w velocity. They set w at the first fluid node based on linear interpolation with the second fluid node. In the version implemented into WRF, linear interpolation is used to set the vertical velocity at a ghost node.

As in the no-slip implementation, the IBM routines are executed just after the acoustic time step loop on each of the three Runge-Kutta steps. First halo exchanges, boundary condition updates, and cut cell updates are performed. Then the velocity reconstruction is applied. Figure 8.2 illustrates the velocity reconstruction IBM method as implemented in WRF. To summarize, the velocity at U_1 is reconstructed using the magnitude at the second fluid node U_2 in the surface normal direction. No normal flow to the boundary is enforced by setting the vertical velocity at a ghost point w_G using the value at w_1.

8.2.3 Shear stress reconstruction

The shear stress reconstruction method blends the IBM ghost cell method with the original WRF boundary conditions, which set the shear stress at the surface using the log-law. This methods satisfies the boundary conditions given by equations (8.5a) and (8.5b), where (8.5a) is the kinematic boundary condition and (8.5b) sets the surface stress τ at the immersed boundary. In these equations μ is the column mass per unit area.

$$\mathbf{U} \cdot \hat{\mathbf{n}} = 0 \tag{8.5a}$$

$$\tau_{w_{xz}} = -\mu \left(\frac{\kappa}{\ln \frac{z_1 - h}{z_o}} \right)^2 |\mathbf{U}| u, \qquad \tau_{w_{yz}} = -\mu \left(\frac{\kappa}{\ln \frac{z_1 - h}{z_o}} \right)^2 |\mathbf{U}| v \tag{8.5b}$$

The shear stress reconstruction IB method is illustrated in Figure 8.3. The kinematic boundary condition is imposed by setting the velocity at ghost nodes using linear interpolation. In the case of a flat surface this requires $w_{surf} = 0$, and the velocity at w_G is set using linear extrapolation from the velocity at the first fluid point and the zero value at the surface. Surface stresses are also imposed at the ghost points. Once the desired surface stress is found from equation (8.5b), linear extrapolation is used to set the shear stress at τ_G. As with the velocity reconstruction method, the shear stress reconstruction method would need to be reformulated in terms of normal and tangential components for use with complex geometries.

Like all of the previous methods the first steps are halo exchanges, boundary condition updates, and determination of the ghost points. For the shear stress method, only the velocity ghost points are determined during the domain set up. Vorticity ghost points, located on the edges of computational cells, are also needed for setting the shear stresses. These are not located during the initial pre-processing IBM routines. Once the velocity ghost points are determined, the velocity is zeroed at all interior nodes. Then the kinematic boundary condition is satisfied by setting the velocities at the ghost points.

During the iteration process, the diffusive terms are only evaluated on the first step of the three step Runge-Kutta loop. In Table 3.1 this is step one. If the shear stress reconstruction method is being used, then additional IBM routines are called at this point.

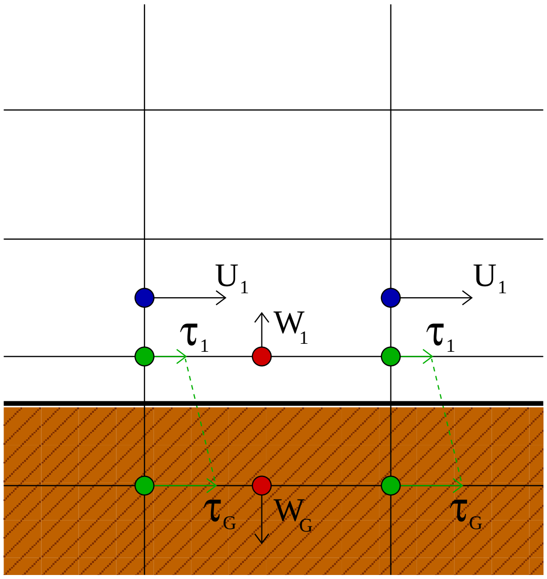


Figure 8.3. Shear stress reconstruction IBM.

First, geopotential is determined at the vorticity points using horizontal averaging. Then, the ghost points for $\tau_{w_{xz}}$ and $\tau_{w_{yz}}$ are found. Next, the stresses at the surface are determined from equation (8.5b). In this equation z_1 is the height of the first fluid node above the surface, h is the height of the immersed boundary, and z_o is the roughness length. It must be verified that $(z_1 - h)$ is greater than z_o. If it is not, then the next fluid node is used to calculate the drag coefficient. In addition to the coefficient of drag, the magnitude of the velocity must be known. Just like in the velocity reconstruction method, this requires averaging on a staggered grid. Four point horizontal averages of v are used to find v on a u node, and vise versa. Now, the desired shear stress at the surface is known. Linear extrapolation is used to determine the value of τ_G that would enforce the desired boundary condition, and the value of τ_G is set. During run time, the kinematic boundary condition is updated just after the acoustic loop on every large time step, as it is for the other two IBM implementations. This means that the kinematic boundary condition is set three times during a full Runge-Kutta loop, while the surface stresses are only set one time during the first R-K step.

8.3 Neutral atmospheric boundary layer

To test the performance of the IBM methods in WRF, the case of a neutral atmospheric boundary layer (ABL) is considered. The atmospheric boundary layer is the bottom region of the troposphere, which is in contact with the Earth's surface. Flow in the boundary layer region is greatly affected by surface friction and fluxes. A neutral atmosphere is one with a constant potential temperature. This means that the temperature decrease with height is prescribed exactly by the adiabatic lapse rate. The bottom ten percent of the atmospheric boundary layer is called the surface layer or constant flux region. If the atmosphere is neutral, then the mean velocity profile in the surface layer should follow the log-law.

WRF is an LES-capable numerical weather prediction code. It is distributed with three eddy viscosity turbulence closures including: constant eddy viscosity, Smagorinsky, and 1.5 order TKE models. Implicit filtering is used for the Smagorinsky and 1.5 TKE models, meaning that the length scale component of the eddy viscosity is provided by the grid spacing. Each of the closures is explained in detail in a WRF technical note (see Skamarock et al. [2007, Chap. 4]). For the case of constant eddy viscosity, the analytical solution of rotation influenced flow over flat terrain is an Ekman spiral. In the fully turbulent case, the log-law is expected from scale analysis [Garratt, 1992]. The following sections detail the domain and flow set-up, and the effects of the immersed boundary implementations. The goal is to capture the original WRF solution when using IBM, so comparisons are made between the simulation results when using the original WRF boundary conditions and each of the three IBM techniques. Results are also presented for both the Smagorinsky and 1.5 TKE closures. Additionally, the no-slip case is modeled with constant eddy viscosity and compared to the analytical Ekman spiral solution.

8.3.1 Domain and flow set-up

The neutral boundary layer simulations in WRF have a similar flow set-up to those presented by Andren et al. [1994] and Chow et al. [2005]. Flow is driven by a large scale constant pressure gradient which would balance a geostrophic wind of (U_g, V_g) = (10,0) ms^{-1}. The Coriolis parameter f is set to a constant value of 1 x 10^{-4} s^{-1}. The domain is horizontally periodic, and has 42 nodes with $\Delta x = \Delta y = 32$ m grid spacing in each of the horizontal dimensions for an overall domain size of 1312 m in each horizontal direction. The height of the domain is approximately 1500 m or in terms of the pressure coordinate 83.3 kPa, and the flat surface is placed at a height of 100 m. Forty-two grid points are used in the vertical dimension with an exponential grid stretching function used to bias the coordinate spacing towards the surface. The grid stretching function satisfies the rule of thumb that adjacent cells be stretched by no more than ten percent. In the terrain following coordinate, all 42 grid points are between 100 m and 1500 m. The minimum vertical grid spacing is $\Delta z_{min} \approx 9.5$ m and the maximum is $\Delta z_{max} \approx 85.2$ m. When IBM is used the 42 vertical grid points span the entire 1500 m, with $\Delta z_{min} \approx 10.1$ m and $\Delta z_{max} \approx 91.6$ m. A fifth order advection scheme is used in the horizontal dimensions, and third order is used in the vertical.

Flow is initialized with a sounding that includes constant velocities of (u, v) = (10,0) ms^{-1}, constant potential temperature θ = 288° K, and moisture mixing ratio q = 0 g/kg. The initial u velocity is seeded with small perturbations near the surface, so that the flow transitions to be fully turbulent. Drag at the surface is modeled using a roughness length z_o = 0.1 m for all cases except the no-slip case where it is not applicable.

A 0.5 s time step is used for the Runge-Kutta loop, and a 0.05 s time step is used for the acoustic time step. The simulations were run for 84 hours which is just over 30 non-dimensional time units tf normalized by the Coriolis parameter. Inertial oscillations are evident in the flow, and are due to imbalances between the pressure gradient and the Coriolis forcing while the flow is tending towards a steady state solution. The inertial oscillations have a period of $2\pi/f$, which is ~17.5 hours for the prescribed Coriolis force in this model. Figure 8.4 shows the time evolution of the domain averaged u and v velocities for the original WRF terrain following coordinate and the three IB methods using the Smagorinsky closure. Figure 8.5 shows the same information on a hodograph. Results for the TKE turbulence closure are presented in Figures 8.6 and 8.7.

It is immediately apparent that there is an interaction between the velocity reconstruction IB model and the 1.5 TKE closure that inhibits damping of the inertial oscillations. With the exception of this case, the inertial oscillations appear to be sufficiently damped after two periods or ~35 hours. For this reason, the results in section 8.3.2 are averaged over a time period spanning from 36 to 84 hours or approximately $13 < tf < 30$ non-dimensional time units. For reference, a star marks the 36th hour on the hodographs where the time averaging begins. In comparison Andren et al. [1994] averaged over $7 < tf < 10$, and Chow et al. [2005] averaged over $20 < tf < 30$.

8.3.2 Neutral atmospheric boundary layer results

Figures 8.8 and 8.10 show the horizontally and temporally averaged u and v velocity profiles with the Smagorinsky and 1.5 TKE closures respectively. The height quantity in the y-axis is also horizontally and temporally averaged over the horizontal coordinate surface for each case. This is necessary because the height of the pressure coordinate used in WRF is a function of space and time. The same averaged u and v quantities are used to calculate wind speed $U = \sqrt{\bar{u}^2 + \bar{v}^2}$, which is shown on a semi-log plot in Figure 8.9 with the Smagorinsky closure and Figure 8.9 with the 1.5 TKE closure. In these figures the wind speed is non-dimensionalized by the friction velocity defined as $u_\star = \sqrt{C_d}U$, and the average height is non-dimensionalized by the roughness length scale z_o. The theoretical log-law is also included and shown as a thick black line. Results from the no-slip IBM simulations are excluded from the logarithmic plot because this boundary condition does not include a wall model or roughness parameterization, so a logarithmic velocity profile is not expected due to inadequate near surface resolution.

It is clear from the figures that the velocity and shear stress reconstruction models do an excellent job of recreating the original WRF solution with the Smagorinsky turbulence closure. The no-slip condition would need to be modified to include a roughness parameterization in order to match the results of the original WRF boundary conditions. Looking at the results from the original WRF boundary condition, it can be seen that the Smagorinsky turbulence closure does a much better job of recreating the theoretical log-low results than the 1.5 order TKE model. It is a well known problem in neutral boundary layer simulations

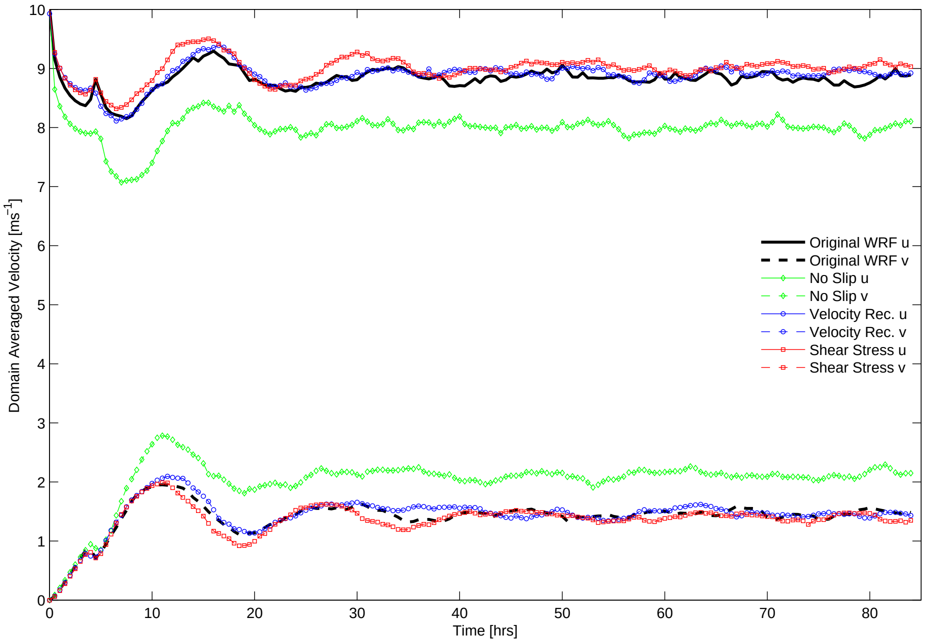


Figure 8.4. Time evolution of domain averaged u and v velocities showing the damping of inertial oscillations with the Smagorinsky closure.

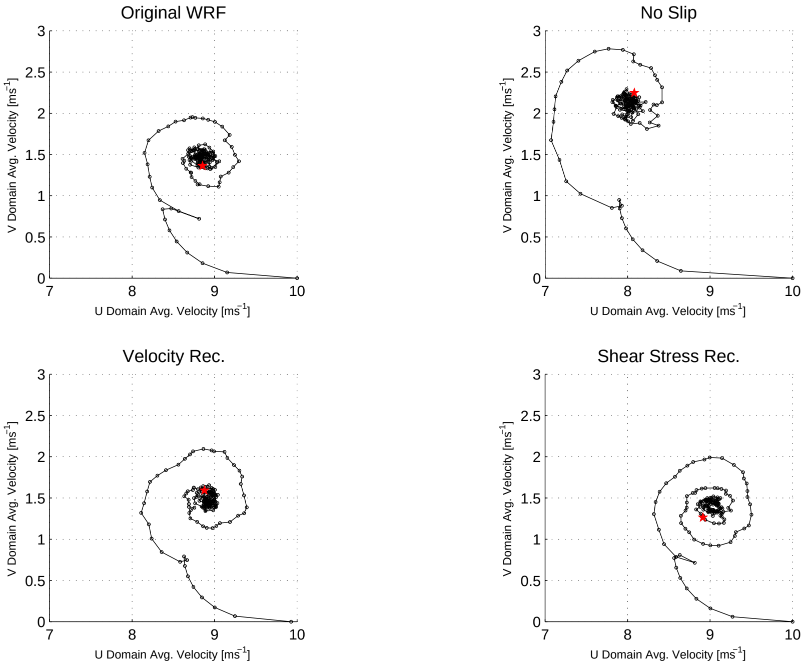


Figure 8.5. Domain averaged u and v velocity on a time series hodograph showing the damping of inertial oscillations with the Smagorinsky closure. The red star marks the time at 36 hours, which is after ~2 periods.

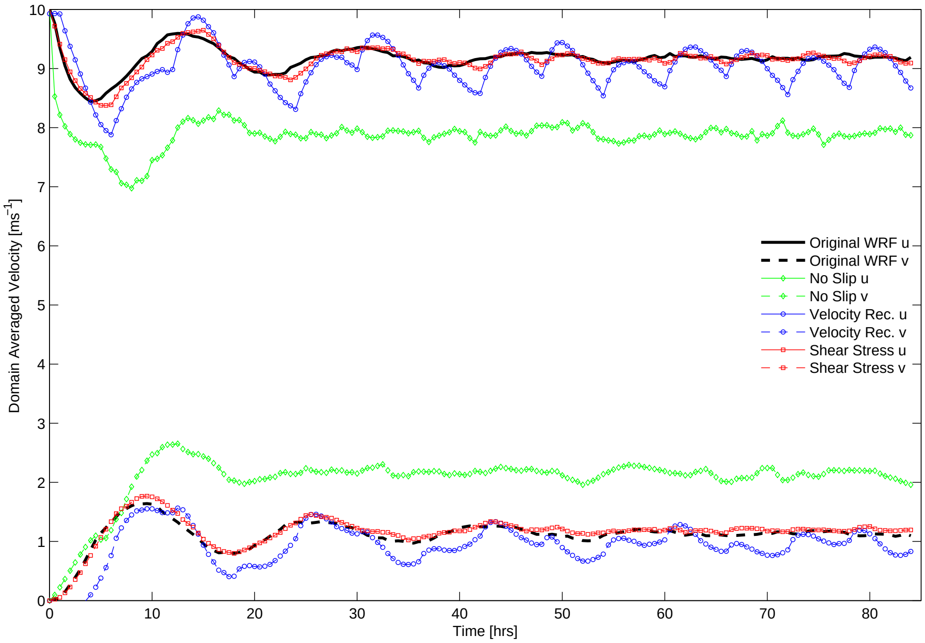


Figure 8.6. Time evolution of domain averaged u and v velocities showing the damping of inertial oscillations with the 1.5 order TKE closure.

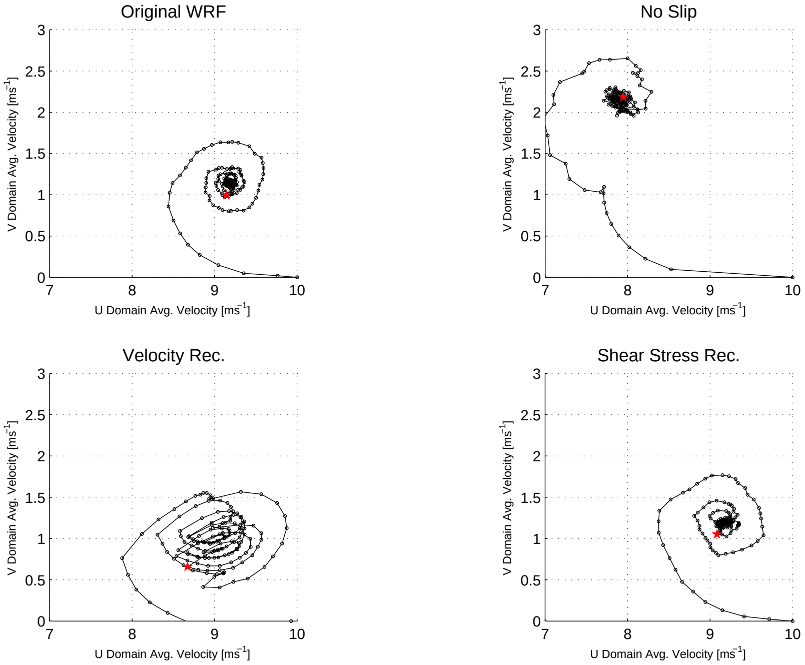


Figure 8.7. Domain averaged u and v velocity on a time series hodograph showing the damping of inertial oscillations with the 1.5 order TKE closure. The red star marks the time at 36 hours, which is after ~2 periods.

that eddy viscosity models often do not agree with similarity theory in the surface layer [Chow et al., 2005], so this discrepancy is not entirely unexpected. What is unexpected is that the velocity and shear stress reconstruction models do not recreate the original WRF solution with the TKE closure, although, in both the Smagorinsky and TKE models, the shear stress reconstruction IB method is closer to the original WRF solution than the velocity reconstruction method. TKE models use a bottom boundary condition for the TKE equation. It is possible that by imposing a boundary condition for TKE at the immersed boundary, the IBM results would better match the original WRF solution. Despite the fact that the inertial oscillations did not damp sufficiently when velocity reconstruction was combined with the TKE model, this solution best matches the log-law very near the surface when compared to the original WRF boundary conditions or the shear stress IB model with the TKE closure. This can be explained by the inherent property of the method which forces the velocities at the first two fluid nodes above the plate to match the log-law. The slope of the velocity and shear stress reconstruction models matches the log-law well, even though there is a significant departure from the log-law near the surface.

The implementation of the no-slip method can be further verified by comparing the simulation results with a constant eddy viscosity ν_t to the analytical solution of an Ekman spiral. In the northern hemisphere, where the Coriolis parameter is positive, the Ekman spiral due to geostrophic winds of $\vec{V_g} = (U_g, 0)$ is given by equations (8.6a) and (8.6b):

$$u = U_g \left[1 - \exp(-a_o z)\cos(a_o z)\right] \tag{8.6a}$$

$$v = U_g \exp(-a_o z)\sin(a_o z) \tag{8.6b}$$

where

$$a_o^2 = \frac{f}{2\nu_t}.$$

The boundary conditions for this solution require $\vec{V} = 0$ at the surface, and $\vec{V} \to \vec{V_g}$ as $z \to \infty$.

Figure 8.12 shows the damping of numerical oscillations for the no-slip case with a constant eddy viscosity of 12.5 $\mathrm{m^2 s^{-1}}$. This simulation ran for ∼20 non-dimensional time units, therefore temporal averaging is over the range of $13 < tf < 20$ or 36 to 54 hours. Figure 8.13 plots the horizontally and temporally averaged u and v velocity against average height. The theoretical Ekman spiral is also included in the plot. It can be seen that the two solutions agree well, especially within the boundary layer region. Some error is present in the velocity profiles near the top of the domain, and it is likely that this is due to different top boundary conditions. As previously stated, the analytical solution uses $\vec{V} \to \vec{V_g}$ as $z \to \infty$, and WRF uses $w_{surf} = 0$ and $p\prime = 0$ at the top vertical coordinate. Regardless, the agreement in the boundary layer provides proof that the no-slip IB method is implemented correctly.

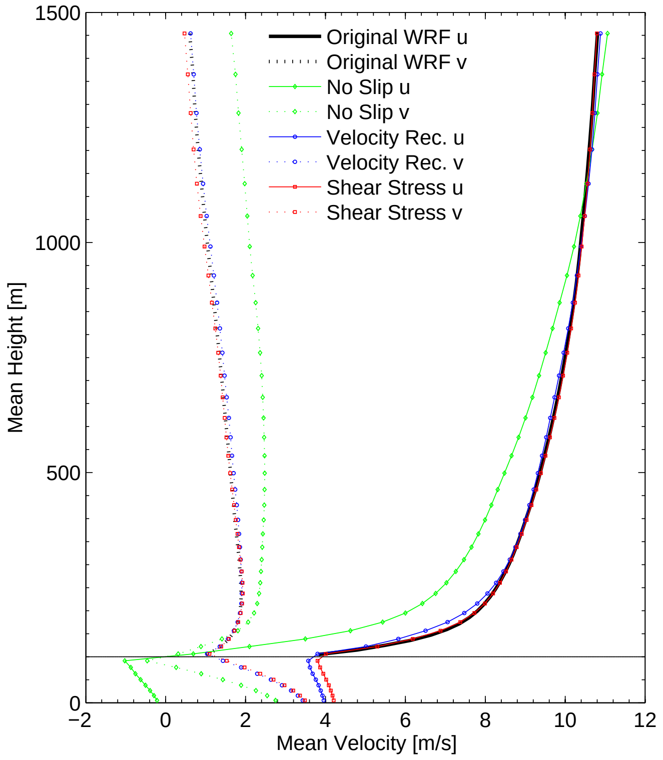


Figure 8.8. Mean U and V velocity with the three IBM implementations and the Smagorinsky closure. The flat plate is at 100 meters.

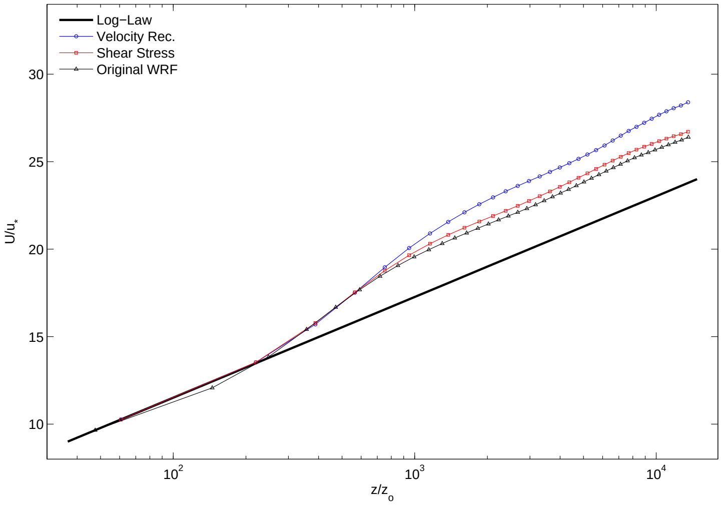


Figure 8.9. Non-dimensionalized mean velocity on a semi-log plot with the Smagorinsky closure.

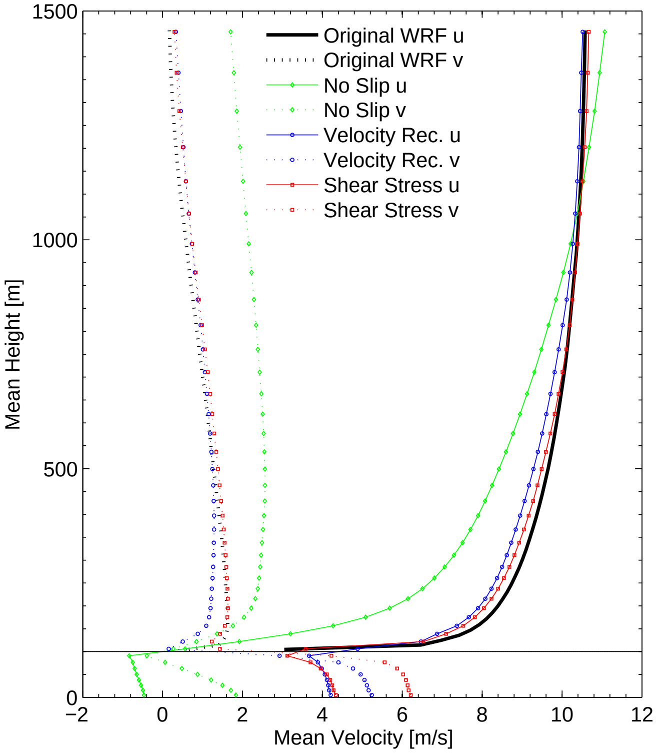


Figure 8.10. Mean U and V velocity with the three IBM implementations and the 1.5 order TKE closure. The flat plate is at 100 meters.

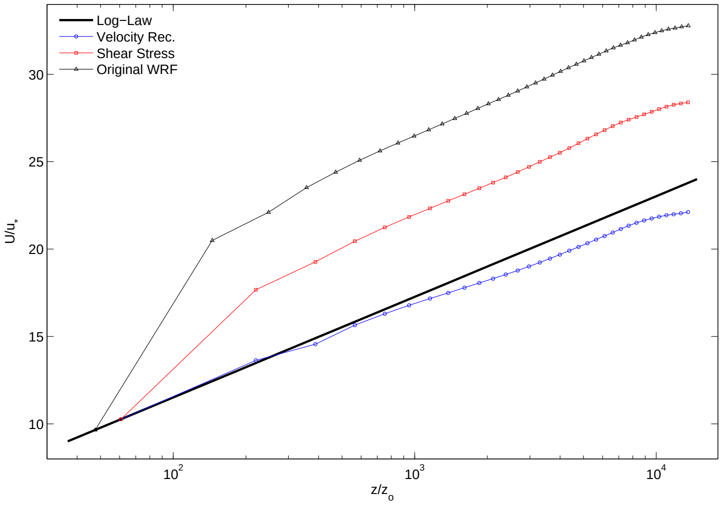


Figure 8.11. Non-dimensionalized mean velocity on a semi-log plot with the 1.5 order TKE closure.

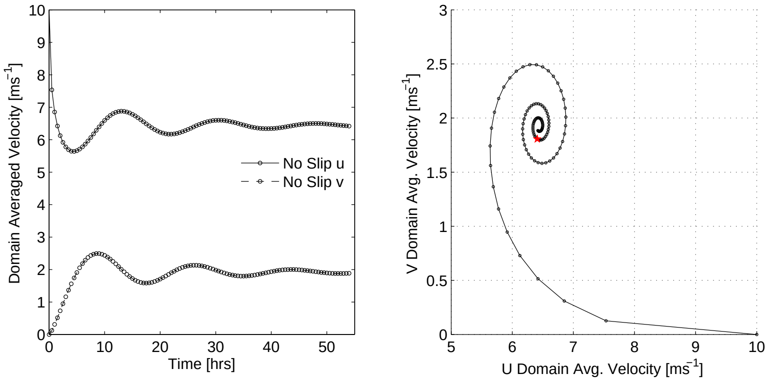


Figure 8.12. (a) Time evolution of domain averaged u and v velocities showing the damping of inertial oscillations for the no-slip boundary condition with constant eddy viscosity. (b) Domain averaged u and v velocity on a hodograph showing the damping of inertial oscillations with a red star marking the time at 36 hours, which is after ~2 periods.

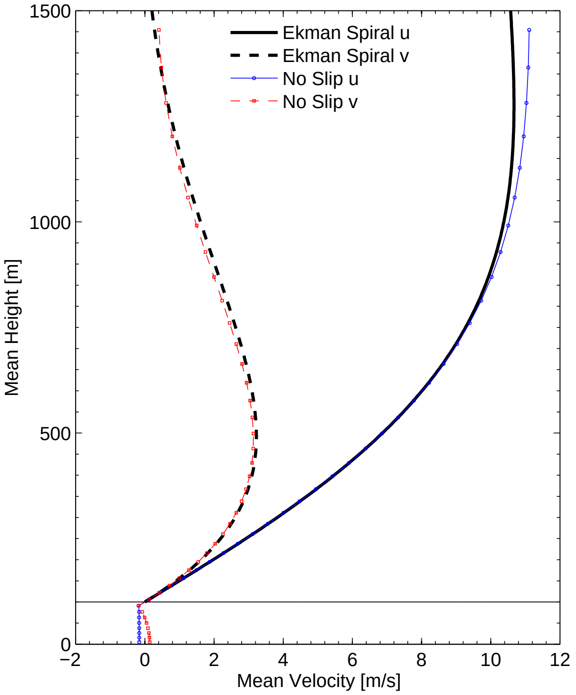


Figure 8.13. Ekman spiral and mean U and V velocity for the no-slip IBM implementation with constant eddy viscosity. The flat plate is at 100 meters.

8.4 Conclusions

Most of the IBM research to date has simulated flow with a no-slip boundary condition. This has been adequate because the applications have been at low or moderate Reynolds numbers where the flow domain was highly resolved. Atmospheric simulations involve flows with very high Reynolds numbers, therefore resolution of the viscous scales is not currently possible. To overcome this challenge, numerical weather prediction codes employ a surface roughness parameterization for the bottom boundary condition. This idea could be used in combination with IBM to successfully represent complex surfaces in atmospheric simulations. To test this theory, three IBM techniques were tested with the NWP code WRF for the case of a rotation influenced boundary layer.

The no-slip boundary condition has been tested by many researchers, and extended to complex boundaries in three-dimensional flows. Many interpolation methods for boundary reconstruction exist for the no-slip case. In contrast, the velocity and shear stress reconstruction methods would need to be reformulated in terms of normal and tangential components to be applied to complex terrain. The methods could be extended to complex terrain and tested for the canonical cases of two-dimensional flow over an isolated hill and three-dimensional flow over a Gaussian hill.

It was shown that surface stress models combined with IBM show improved performance over the traditional no-slip IBM boundary condition in their ability to recreate the expected logarithmic velocity profile for the case of a neutral boundary layer. Interactions that warrant further study were seen between the 1.5 TKE closure and the IBM implementations. While difficulties exist in reformulating these methods for complex terrain, the potential gain in performance makes further study a worthwhile endeavor.

Chapter 9

Summary, conclusions, and recommendations

9.1 Summary

Adding an immersed boundary capability to the Weather Research and Forecasting model greatly extends the functionality of this code, allowing for more accurate results in steep terrain and enabling simulations of urban terrain which could not previously be performed. Given the great need for accurate flow simulations in complex terrain, this added functionality will affect the modeling of a variety of diverse applications including urban transport and dispersion, wind energy siting, wildfire forecasting, among others.

Improvements to flow solutions using the IBM, rather than the native terrain-following coordinate, were illustrated and quantified in chapter 4. As demonstrated over chapters 5, 6, and 7, we successfully developed an immersed boundary method which can be used within the WRF framework while maintaining features of the code such as pressure-based coordinates, atmospheric physics, and nesting capabilities. Simulations with the immersed boundary method were shown to be able to reproduce the results achieved using terrain-following coordinates for flows over terrain with sufficiently shallow slopes. When inverse distance weighting is used for interpolation within the immersed boundary method, there are no limits on the complexity of the terrain that can be represented with our method.

Our IBM capability can be used effectively for studying flows over complex terrain, especially if it is combined with a wall model. Two possible implementations of wall models combined with the IBM were tested in chapter 8 for flow over flat terrain. Extension of these methods to complex terrain will greatly expand the applicability of our immersed boundary method. For example, flow in the Owens Valley is presented in appendix B using a no-slip boundary condition, while a wall model boundary condition would be more appropriate, especially for mountainous terrain.

9.2 Conclusions

Validation was carried out for flow over two and three-dimensional idealized terrain, however, additional validation is ongoing for real cases. In chapter 7 the urban geometry for Oklahoma City, OK was used for a simulation in a one-way nested configuration. The flow in the parent domain is periodic channel flow, which creates an idealized logarithmic velocity profile that is imposed as the lateral boundary condition for the nested urban domain. This type of inlet profile is sufficient for validation with the Joint Urban 2003 data set. Chan and Leach [2007] saw good agreement within the urban corridor for intensive observation periods 3 and 9 using a logarithmic inlet profile with the CFD code FEM3MP.

The obvious goal for the future is to nest domains using the immersed boundary method into domains at the mesoscale; however, there are still several barriers to running simulations like this. The primary limitation of running a microscale (i.e. urban) domain nested within a mesoscale domain that is large enough to capture synoptic weather patterns is computational cost. Lundquist and Mirocha [2008] note that "Because resolving the effects of individual buildings demands grid cells on the order of 3 m or lower, typical domains are on the order of 1 km x 1 km x 400 m." With uniform grid spacing, as currently required by WRF in the horizontal, at least 14.8 million grid points are required for a domain of this size and resolution. Grid stretching in the vertical could reduce the total number of grid points somewhat.

Nesting from the mesoscale down to the coarse LES scale ($\Delta x \approx 20$ m) is a computationally tractable problem because the overall size of the domain can decrease at the rate of the nesting ratio. For example, in a study of LES scale simulations nested within mesoscale simulations by Lundquist et al. [2009b] seven nests were used in a 1:3 ratio to nest from a 600 km x 600 km x 800 m domain with 12 km horizontal resolution to a domain 800 m x 800 m x 800 m in size with a horizontal grid spacing of 16.5 m. The same number of grid points can be used on each nest because the horizontal extents of the domain are one third of the parent domain's dimensions. In the Lundquist et al. [2009b] simulations, each domain had an estimated 50^3 grid points for a total of 875,000 nodes. Further nesting becomes much more computationally intensive because the smallest domain (800 m x 800 m x 800 m) cannot become any smaller, as the actual size of the urban area now defines the domain size. Therefore three nests, all 1 km x 1 km x 400 m in size at $\Delta x = 18$, 6, and 2 m, would require a total of 52 million grid points if isotropic grids are used on each domain (although isotropic grids are not possible without vertical nesting).

The Oklahoma City domain used in chapter 7 has a domain size of 516 m x 678 m x 435 m. With 2 m resolution, nearly 15 million grid points are needed for just the inner most domain. A second domain covering 570 m x 732 m x 435 m at 6 m horizontal resolution requires an additional 2 million grid points. Use of additional vertical grid stretching beyond what is naturally produced by the pressure-based coordinate should be investigated further, and can reduce the total number of grid points. However, even with additional vertical grid stretching this is still a large computational problem. Additionally, a smaller nesting ratio (i.e. 1:5 instead of 1:3) could further reduce computational expense. The Oklahoma City domain used here is much smaller than the domain used in the FEM3MP simulations of Chan and Leach [2007], where the domain size is 1030 m x 3010 m x 425 m. The FEM3MP model utilizes a graded mesh combined with a virtual building function, where some buildings are not explicitly resolved, but instead represented as regions of high drag. These features allow ∼1 m grid spacing at the boundaries of explicitly resolved buildings and at the ground, while only using 201 x 303 x 45 (∼2.8 million) grid points. So, while the simulations in Chan and Leach [2007] have about 600 m of fetch upwind of the IOP 3 scalar release, our simulation just covers the release area. This means that while their logarithmic input velocity profile is modified by the presence of buildings upwind of the scalar release, currently ours is not.

Additionally, even at 2 m resolution, there are many building features which are still not well resolved. Lidar data of the terrain for the Joint Urban 2003 field campaign exists at 1 m resolution. This data has been processed along with other data sources (such as satellite photographs) to develop a three-dimensional shapefile representation of the buildings. Within our Oklahoma City domain, the shortest terrain features have elevations of 3 m, and it can be seen in figure 7.11 that there are many building features on the length scale of 1 or 2 m in width. Additionally, there are several places where the space between buildings or building features is unresolved. Tseng et al. [2006] found that 6 to 8 grid points are needed across a bluff body for the immersed boundary method to produce accurate results. The effects of these under-resolved features should be studied further. Two possible strategies for dealing with this problem are to adopt a hybrid technique of using both explicitly resolved and virtual buildings or to develop an algorithm which processes the building data to filter it with respect to the resolution of the simulation.

Furthermore, guidelines must be developed for domains or terrain features which should be treated with the immersed boundary method, those which should be treated as virtual terrain with increased drag, and those which should be completely parameterized with a surface treatment such as an urban canopy parameterization. Using our previous example of three nested domains with horizontal resolutions of $\Delta x = 18$, 6, and 2 m, it may be as simple as treating each domain differently so that an urban canopy parameterization is used on the 18 m domain, a virtual building treatment is used on the 6 m domain, and the immersed boundary method is used on the 2 m domain. However, it is also possible that a combination of these methods would be needed within a single domain to deal with terrain features of various scales. Additionally, the effects of elements that are not included in the terrain file, such as vegetation, may need to be parameterized.

9.3 Recommendations for future study

This work has contributed to the development of immersed boundary methods, especially as they are applicable to mesoscale models, and the WRF code in general. Recommendations for future study fall into two broad categories: improvements which increase computational efficiency and those which add functionality.

9.3.1 Recommendations which improve computational efficiency

Tseng [2003] notes that the immersed boundary method can be implemented with no additional computational cost, however, we found this not to be the case in our model. In models with a fixed vertical coordinate, the locations of the ghost points and neighboring points that are included in the interpolation are fixed. The points can be located once during an initialization processes and stored for the remainder of the run. Additional work for the interpolation procedure such as inverting the Vandermonde matrix (for trilinear interpolation) or calculating the radial distance from the image point (for inverse distance weighting) can also take place before integration of the solution begins. In these cases the only additional work during integration is matrix multiplication. With a pressure-based coordinate, the index of the ghost node can vary during integration, although in practice we found this to be a rare event. Additionally, the height in physical space does vary at each time step for all points involved in the interpolation, meaning that the Vandermonde matrix or radial distance change at each time step. During development of the method we choose to call all of the IBM routines at each acoustic time step beginning with searching for ghost points and ending with the imposition of forcing. This was done to eliminate uncertainty associated with grid movement. We were positive during development that the boundary condition was enforced at each acoustic time step. Certainly, this process could be optimized further. At a minimum, the searching routine for locating ghost points could be replaced with a check to determine if the ghost point location is still valid. Additionally, it may be possible to obtain a solution that is sufficiently accurate, while updating the location of the interpolation points less frequently (for example only on the first Runge-Kutta time step). Another option would be to use a mesoscale model with a fixed time invariant vertical coordinate, such as the Advanced Regional Prediction System (ARPS), which shares many features with the WRF code.

Current grid nesting techniques also place severe constraints on the ability to nest down from the mesoscale to microscale. One-way nesting in WRF can be accomplished by running each domain in serial, in which case the history output file provides lateral boundary conditions to the child domain at the frequency of the times in the history file. Alternatively, the simulations can be run concurrently, so that boundary conditions on the child domain are updated at the frequency of the parent domain's time step. For LES, concurrent simulations are more effective for transferring high frequency turbulent fluctuations from the parent grid to the child grid. Serial simulations are limited by the history file size that would be required to save data from the entire domain at every time step. One possible solution for serial simulations is to write boundary files, containing only the information needed to fill the boundary condition region of the child domain, while integrating the parent domain, but not requiring history for every point in the grid.

Vertical nesting may be used while running serial simulations, but not while running concurrent simulations. This means that for concurrent simulations the vertical resolution desired on the inner most domain must be used on every domain. This is not only a computational burden, but it also produces highly anisotropic cells on the coarsest grids. This limitation could easily be removed by allowing for vertical nesting in concurrent simulations.

Finally, concurrent runs are not currently parallelized efficiently. Parent and child domains are decomposed onto the same processors, meaning that for the Oklahoma City simulation the parent domain with 2 million points is decomposed onto the same processors as the child domain with 15 million points. This is inefficient because the number of processors used is limited by the number of nodes on the coarsest domain. Additionally, multiple domains now have to fit onto the same processors that could be used to run just the outer most domain. This is a severe computational constraint for nested LES simulations, and different domain decomposition techniques should be investigated for concurrent simulations.

9.3.2 Recommendations which add functionality

The most important recommendation is that the rough surface parameterizations presented in chapter 8 be extended to complex terrain. This will allow for more accurate LES simulations of high Reynolds number flows without fully resolving the near-wall regions at immersed boundaries.

In the one-way nested simulation presented in chapter 7 the urban domain using IBM is nested into another domain using IBM to represent a flat plate. This allows for ghost points below all of the terrain, because the terrain elevation is always greater than the bottom of the domain. When an IBM domain is nested within a domain with terrain-following coordinates, as in figure 5.1, a ghost cell approach cannot be used at all locations. Seamless nesting requires that the vertical coordinates are aligned at the interface of the two nests. In the IBM domain, terrain without nodes below can be treated by applying forcing in the fluid domain. This capability has been added to our IBM, but has only received preliminary testing, and is therefore not used in the simulations presented here. The implementation of this feature was straight-forward, therefore, it should be possible to use seamless nesting between terrain-following and IBM coordinates in the future.

In our IBM, terrain is represented as a two-dimensional piecewise linear function. This representation was chosen because terrain in mesoscale models is defined as a two-dimensional array of heights. This representation is, however, somewhat limiting for urban terrain. Elevation changes must take place over the space of a horizontal grid, meaning that perfectly vertical surfaces cannot be represented. Furthermore, three-dimensional objects such as bridges or sky walks cannot be fully represented. Previous atmospheric urban IBM simulations [Tseng et al., 2006; Shi et al., 2008] align the buildings with the mesh, so that IBM resorts to a simple masking approach. Our IBM is thus capable of a higher order representation of the urban terrain than previous methods. Nonetheless, the boundary could be represented even more accurately if an unstructured triangulated mesh (such as a .stl file) is used to describe the urban terrain. Unfortunately, urban terrain is not meshed often because the descriptions of urban terrain create dirty geometries, creating difficulties with producing a watertight mesh. IBM methods (particularly our inverse distance weighting IBM) are well suited to interacting with this type of surface description and even with low-quality non-watertight grids. The accuracy of the flow solution should increase with the higher-order boundary representation.

Colette et al. [2003] demonstrated the importance of including the effects of topographic shading in simulations of valley flows, especially for fine-resolution small-scale simulations where steep topography is represented. Until version 3 of WRF, which was released in 2008, radiation was idealized as a completely vertical process. This is also how radiation is treated when coupled to our IBM as described in chapter 5. Current versions of the WRF code include the effects of the incident angle of radiation on sloped surfaces and topographic shading. Both of these new features should be coupled to the immersed boundary for increased accuracy.

For fully three-dimensional radiation effects at the urban scale, a three-dimensional urban energy balance model must be used instead of the mesoscale radiation parameterizations. WRF solves a conservation equation for potential temperature, and in the urban simulations presented in chapter 7, a zero-flux boundary condition is applied on the surfaces of the buildings. If a three-dimensional urban radiation model were used, such as presented in Krayenhoff and Voogt [2007], a temperature or flux boundary condition could be assigned to the building surfaces. However, three-dimensional radiation models still have limitations, often requiring idealized geometries and being extremely expensive computationally.

The suggested advances beyond the work presented here (surface roughness parameterizations, improvements to parallelizations, adding vertical nesting for concurrent runs, developing higher-order boundary representations, including three-dimensional radiative transfer, etc.) are substantial. Nonetheless, given the significant improvements afforded by IBM as demonstrated herein, and recognizing the demand for accurate simulations in complex terrain required for transport and dispersion, wind energy, and other applications, it is hoped that opportunities for extending this work will emerge.

Bibliography

Adcroft, A., C. Hill, and J. Marshall, Representation of Topography by Shaved Cells in a Height Coordinate Ocean Model, *Mon. Weather Rev.*, *125*, 2293–2315, 1997.

Allwine, K., and J. Flaherty, Joint Urban 2003: Study overview and instrument locations, , Tech. Report PNNL-15967, Pacific Northwest National Laboratory, Richland. WA, 2006.

Andren, A., A. Brown, J.Graf, P. Mason, C.-H. Moeng, F. Nieuwstadt, and U. Schumann, Large-eddy simulation of a neutrally stratified boundary layer: A comparison of four computer codes, *Q. J. R. Meteorol. Soc.*, *120*, 1457–1484, 1994.

Arya, S., *Introduction to Micrometeorology*, Academic Press, Inc., 1988.

Baggett, J., Some Modeling Requirements for Wall Models in Large Eddy Simulation, *Center for Turbulence Research, Annual Research Briefs*, p. 123, 1997.

Balaras, E., Modeling complex boundaries using an external force field on fixed Cartesian grid in large-eddy simulations, *Comput. Fluids*, *33*, 375–404, 2004.

Balaras, E., C. Benocci, and U. Piomeli, Two-Layer Approximate Boundary Conditions for Large-Eddy Simulations, *AAIA J.*, *34*(6), 1111, 1996.

Bernard, P., and J. Wallace, *Turbulent Flow Analysis, Measurement, and Prediction*, John Wiley and Sons, Inc., 2002.

Bjerknes, V., and Coll., *Dynamic Meteorology and Hydrography*, Washington, 1910-11.

Briscolini, M., and P. Santangelo, Development of the mask method for incompressible unsteady flows, *J. Comp. Phys.*, *84*(1), 57–75, 1989.

Cabot, W., and P. Moin, Approximate Wall Boundary Conditions in the Large-Eddy Simulation of High Reynolds Number Flow, *Flow, Turbulence, and Combustion*, *63*, 269–291, 1999.

Chan, S., and M. Leach, A Validation of FEM3MP with Joint Urban 2003 Data, *J. Appl. Meteor. Climatol.*, *46*, 2127–2146, 2007.

Choi, J., R. Oberoi, J. Edwards, and J. Rosati, An immersed boundary method for complex incompressible flows, *J. Comp. Phys.*, *224*, 757–784, 2007.

Chow, F., R. Street, M. Xue, and J. Ferziger, Explicit filtering and reconstruction turbulence modeling for large-eddy simulation of neutral boundary layer flow, *J. Atmos. Sci.*, *62*, 2058–2077, 2005.

Chow, F., A. Weigel, R. Street, M. Rotach, and M. Xue, High-Resolution Large-Eddy Simulations of Flow in a Steep Alpine Valley, Part I: Methodology, Verification, and Sensitivity Experiments, *J. Appl. Meteor. Climatol.*, *45*, 63–86, 2006.

Chow, F. K., and R. L. Street, Evaluation of Turbulence Closure Models for Large-Eddy Simulation over Complex Terrain: Flow over Askervein Hill, *J. Appl. Meteor. Climatol.*, *48*(5), 1050–1065, 2009.

Colette, A., F. Chow, and R. Street, A Numerical Study of Inversion-Layer Breakup and the Effects of Topographic Shading in Idealized Valleys, *J. Appl. Meteor.*, *42*, 1255–1272, 2003.

Cristallo, A., and R. Verzicco, Combined Immersed Boundary/Large-Eddy-Simulations of Incompressible Three Dimensional Complex Flows, *Flow Turbulence Combust.*, *77*, 3–26, 2006.

Deardorff, J., A numerical study of three-dimensional turbulent channel flow at large Reynolds Numbers, *J. Fluid Mech.*, *41*(2), 453–480, 1970.

Eliassen, A., The quasi-static equations of motion with pressure as an independent variable, *Geofys. Publ.*, *17*(3), 5–44, 1949.

Fadlun, E., R. Verzicco, P. Orlandi, and J. Mohd-Yusof, Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations, *J. Comp. Phys.*, *161*, 35–60, 2000.

Fast, J., Forecasts of Valley Circulations Using the Terrain-Following and Step-Mountain Vertical Coordinates in the Meso-Eta Model, *Wea. Forecasting*, *18*, 1192–1206, 2003.

Franke, R., Scattered data interpolation: tests of some methods, *Math. Comput.*, *38*, 181–200, 1982.

Gal-Chen, T., and R. Somerville, On the Use of a Coordinate Transformation for the Solution of the Navier-Stokes Equations, *J. Comp. Phys.*, *17*, 209–228, 1975.

Gao, T., Y. Tseng, and X. Lu, An improved hybrid Cartesian/immersed boundary method for fluid-solid flows, *Int. J. Numer. Meth. Fluids*, *55*, 1189–1211, 2007.

Garratt, J., *The atmospheric boundary layer*, Cambridge University Press, 1992.

Ge, L., and F. Sotiropoulos, A numerical method for solving the 3D unsteady incompressible Navier-Stokes equations in curvilinear domains with complex immersed boundaries, *J. Comp. Phys.*, *225*, 1782–1809, 2007.

Ghias, R., R. Mittal, and H. Dong, A sharp interface immersed boundary method for compressible viscous flows, *J. Comp. Phys.*, *225*, 528–553, 2007.

Gilmanov, A., and F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D geometrically complex, moving bodies, *J. Comp. Phys.*, *207*, 457–492, 2005.

Gilmanov, A., F. Sotiropoulos, and E. Balaras, A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids, *J. Comp. Phys.*, *191*, 660–669, 2003.

Golaz, J.-C., J. D. Doyle, and S. Wang, One-Way Nested Large-Eddy Simulation over the Askervein Hill, *J. Adv. Model. Earth Syst.*, *1*(6), 1–6, 2009.

Goldstein, D., R. Handler, and L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comp. Phys.*, *105*, 354–366, 1993.

Hanna, S., M. Brown, F. Camelli, S. Chan, W. Coirier, O. Hansen, A. Huber, S. Kin, and R. Reynolds, Detailed Simulations of Atmospheric Flow and Dispersion in Downtown Manhattan, An Application of Five Computational Fluid Dynamics Models, *Bull. Amer. Meteor. Soc.*, *87*, 1713–1726, 2006.

Hommema, S., and R. Adrian, Packet structure of surface eddies in the atmospheric boundary layer, *Boundary-Layer Meteorol.*, *106*, 147–170, 2003.

Iaccarino, G., and R. Verzicco, Immersed boundary technique for turbulent flow simulations, *Appl. Mech. Rev.*, *56*, 331–347, 2003.

Janjić, Z., Pressure gradient force and advection scheme used for forecasting with steep and small scale topography, *Beitr. Phys. Atmos.*, *50*, 186–189, 1977.

Janjić, Z., On the pressure gradient force error in σ coordinate spectral models, *Mon. Weather Rev.*, *117*, 2285–2292, 1989.

Johnson, C., A. Tindal, and K. Harman, Validation of GH Energy Predictions, in *Proceedings of the American Wind Energy Association Wind Resource and Project Energy Assessment Workshop*, American Wind Energy Association, Washington, DC, 2008.

Jones, S., Project Underperformance: 2008 Update, in *Proceedings of the American Wind Energy Association Windpower 2008 Conference*, American Wind Energy Association, Washington, DC, 2008.

Jones, S., and G. Randall, Examining Project Underperformance, in *Proceedings of the American Wind Energy Association Windpower 2006 Conference*, American Wind Energy Association, Washington, DC, 2006.

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

Kirkpatrick, M., S. Armfield, and J. Kent, A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid, *J. Comp. Phys.*, *184*, 1–36, 2003.

Klemp, J., W. Skamarock, and O. Fuhrer, Numerical Consistency of Metric Terms in Terrain-Following Coordinates, *Mon. Weather Rev.*, *131*, 1229–1239, 2003.

Klemp, J., W. Skamarock, and J. Dudhia, Conservative Split-Explicit Time Integration Methods for the Compressible Nonhydrostatic Equations, *Mon. Weather Rev.*, *135*, 2897–2913, 2007.

Klemp, J. B., J. Dudhia, and A. D. Hassiotis, An Upper Gravity-Wave Absorbing Layer for NWP Applications, *Mon. Weather Rev.*, *136*(10), 3987–4004, 2008.

Klewicki, J., M. Metzger, E. Kelner, and E. Thurlow, Viscous Sublayer Flow Visualizations at $R_\theta \sim$ 1,500,000, *Phys. Fluids*, *7*(4), 857–863, 1995.

Klewicki, J., P. Priyadarshana, and M. Metzger, Statistical structure of the fluctuating wall pressure and its in-plane gradients at high Reynolds number, *J. Fluid Mech.*, *609*, 195–220, 2008.

Kline, S., W. Reynolds, F. Schraub, and P. Runstadler, The structure of turbulent boundary layers, *J. Fluid Mech.*, *30*(4), 741–773, 1967.

Krayenhoff, E., and J. Voogt, A microscale three-dimensional urban energy balance model for studying surface temperatures, *Boundary-Layer Meteorol.*, *123*, 433–461, 2007.

Kunkel, G., and I. Marusic, Study of the near-wall-turbulent region of the high-Reynolds-number boundary layer using an atmospheric flow, *J. Fluid Mech.*, *548*, 375–402, 2006.

Landahl, M., On sublayer streaks, *J. Fluid Mech.*, *212*, 593–614, 1990.

Laprise, R., The Euler Equations of motion with hydrostatic pressure as an independent variable, *Mon. Weather Rev.*, *120*(7), 197–207, 1992.

Lundquist, J., and J. Mirocha, Interaction of nocturnal low-level jets with urban geometries as seen in Joint Urban 2003 data, *J. Appl. Meteor. Climatol.*, *47*, 44–58, 2008.

Lundquist, J., F. Chow, J. Mirocha, and K. Lundquist, An improved WRF for urban-scale and complex-terrain applications, in *7th Symposium on the Urban Environment*, p. 4, American Meteorological Society, San Diego, CA, 2007.

Lundquist, J., J. Mirocha, F. Chow, B. Kosović, and K. Lundquist, Nesting large-eddy simulations within mesoscale simulations for wind energy applications, in *Fall Meeting 2008*, American Geophysical Union, 2008a.

Lundquist, J., J. Mirocha, F. Chow, B. Kosović, and K. Lundquist, Nesting large-eddy simulations within mesoscale simulations for urban and complex terrain, in *8th Symposium on the Urban Environment*, American Meteorological Society, 2009a.

Lundquist, J., J. Mirocha, and B. Kosović, Nesting large-eddy simulations within mesoscale simulations for wind energy applications, in *2009 WRF User's Workshop*, National Center for Atmospheric Research, Boulder, CO, 2009b.

Lundquist, K., F. Chow, and J. Lundquist, An immersed boundary method for flow over complex terrain, in *13th Conference on Mountain Meteorology*, p. 10, American Meteorological Society, Whistler, BC, 2008b.

Lundquist, K., F. Chow, and J. Lundquist, An immersed boundary method for the Weather Research and Forecasting model, *Mon. Weather Rev.*, *138*(3), 796–817, 2010.

Mahrer, Y., An Improved Numerical Approximation of the Horizontal Gradients in a Terrain-Following Coordinate System, *Mon. Weather Rev.*, *112*, 918–922, 1984.

Mayor, S. D., P. R. Spalart, and G. J. Tripoli, Application of a Perturbation Recycling Method in the Large-Eddy Simulation of a Mesoscale Convective Internal Boundary Layer, *J. Atmos. Sci.*, *59*(15), 2385–2395, 2002.

Metzger, M., and J. Klewicki, A comparative study of near-wall turbulence in high and low Reynolds number boundary layers, *Phys. Fluids*, *13*(3), 692–701, 2001.

Michalakes, J., J. Dudhia, D. Gill, T. Henderson, J. Klemp, W. Skamarock, and W. Wang, The Weather Research and Forecast Model: Software Architecture and Performance, in *Proceedings of the Ninth ECMWF Workshop of the Use of High Performance Computing in Meteorology*, pp. 156–168, European Center for Medium-Range Weather Forecasts, Reading, U.K., 2005.

Mirocha, J., F. Chow, J. Lundquist, and K. Lundquist, Improved Subfilter Turbulence Modeling for Large Eddy Simulation Using WRF, in *7th Symposium on the Urban Environment*, p. 4, American Meteorological Society, San Diego, CA, 2007.

Mittal, R., and G. Iaccarino, Immersed Boundary Methods, *Annu. Rev. Fluid Mech.*, *37*, 239–261, 2005.

Mittal, R., H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, and A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comp. Phys.*, *227*, 4825–4852, 2008.

Moeng, C.-H., A Large-Eddy-Simulation Model for the Study of Planetary Boundary-Layer Turbulence, *J. Atmos. Sci.*, *41*(13), 2052, 1984.

Moeng, C.-H., J. Dudhia, J. Klemp, and P. Sullivan, Examining two-way grid nesting for large eddy simulation of the PBL using the WRF model, *Mon. Wea. Review*, *135*(6), 2295–2311, 2007.

Mohd-Yusof, J., Combined immersed boundary/B-spline methods for simulations of flow in complex geometry, , Center for Turbulence Research, NASA Ames/Stanford Univ., Palo Alto, CA, 1997.

Morris, S., S. Stolpa, P. Slaboch, and J. Klewicki, Near-surface particle image velocimetry measurements in a transitionally rough-wall atmospheric boundary layer, *J. Fluid Mech.*, *580*, 319–338, 2007.

Peller, N., A. L. Duc, F. Tremblay, and M. Manhart, High-order stable interpolations for immersed boundary methods, *Int. J. Numer. Meth. Fluids*, *52*, 1175–1193, 2006.

Peskin, C., Flow patterns around heart valves: A numerical method, *J. Comp. Phys.*, *10*(2), 252–271, 1972.

Peskin, C., Numerical analysis of blood flow in the heart, *J. Comp. Phys.*, *25*(3), 220–252, 1977.

Phillips, N., A coordinate system having some special advantages for numerical forecasting, *J. Meteor.*, *14*, 184–185, 1957.

Pielke, R., *Mesoscale Meteorological Modeling*, 612 pp., Academic Press, Inc., 1984.

Piomelli, U., and E. Balaras, Wall-Layer Models for Large-Eddy Simulations, *Ann. Rev. Fluid Mech.*, *34*, 349–74, 2002.

Priyadarshana, P., and J. Klewicki, Study of the motions contributing to the Reynolds stress in high and low Reynolds number turbulent boundary layers, *Phys. Fluids*, *16*(12), 4586–4600, 2004.

Rampanelli, G., D. Zardi, and R. Rotunno, Mechanisms of Up-Valley Winds, *J. Atmos. Sci.*, *61*, 3097–3111, 2004.

Ringler, T., T. J, J. Klemp, and W. Skamarock, A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids, *J. Comp. Phys.*, *229*, 3065–3090, 2010.

Saiki, E., and S. Biringen, Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method, *J. Comp. Phys.*, *123*(2), 450–465, 1996.

Schär, C., D. Leuenberger, O. Fuhrer, D. Lüthi, and C. Girard, A New Terrain-Following Vertical Coordinate Formulation for Atmospheric Prediction Models, *Mon. Weather Rev.*, *130*, 2459–2480, 2002.

Schmidli, J., et al., The T-REX valley wind model intercomparison project, in *13th Conference on Mountain Meteorology*, p. 11, American Meteorological Society, Whistler, BC, 2008.

Schumann, U., Subgrid Scale Model for Finite Difference Simulations of Turbulent Flows in Plane Channels and Annuli, *J. Comp. Phys.*, *18*, 376–404, 1975.

Senocak, I., A. Ackerman, D. Stevens, and N. Mansour, Topography modeling in atmospheric flows using the immersed boundary method, , Center for Turbulence Research, NASA Ames/Stanford Univ., Palo Alto, CA, 2004.

Shi, R., G. Cui, Z. Wang, C. Xu, and Z.S.Zhang, Large eddy simulation of wind field and plume dispersion in building array, *Atmos. Env.*, *42*, 1083–1097, 2008.

Shih, L., Numerical Simulations of Stably Stratified Turbulent Flow, Ph.D. thesis, Stanford University, 2003.

Skamarock, W., J. Klemp, J. Dudhia, D. Gill, D. Barker, W. Wang, and J. Powers, A Description of the Advanced Research WRF Version 2, *NCAR/TN-468+STR*, National Center for Atmospheric Research, Boulder, CO, 2007.

Skamarock, W. C., and J. B. Klemp, A time-split nonhydrostatic atmospheric model for weather research and forecasting applications, *J. Comp. Phys.*, *227*(7), 3465–3485, 2008.

Smagorinsky, J., General Circulation Experiments with the Primitive Equations, *Mon. Weather Rev.*, *91*(3), 99–164, 1963.

Smagorinsky, J., S. Manabe, and J. L. Hollaway, Numerical results from a nine-level general circulation model of the atmosphere, *Mon. Weather Rev.*, *93*(12), 727–768, 1965.

Smolarkiewicz, P., R. Sharman, J. Weil, S. Perry, D. Heist, and G. Bowker, Building resolving large-eddy simulations and comparison with wind tunnel experiments, *J. Comp. Phys.*, *227*, 633–653, 2007.

Spalart, P., Detached-Eddy Simulation, *Annu. Rev. Fluid Mech.*, *41*, 181–202, 2009.

Spalart, P., W.-H. Jou, M. Strelets, and S. Allmaras, Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach, in *Advances in DNS/LES*, edited by C. Liu and Z. Liu, pp. 137–147, Greyden Press, Columbus, OH, 1997.

Stull, R., *An Introduction to Boundary Layer Meteorology*, Kluwer Academic Publishers, 1988.

Talbot, C., E. Bou-Zeid, and J. Smith, Multiscale Atmospheric Simulations over a Complex Terrain: Surface Variability and Land-Atmosphere Exchange, in *22nd Conference on Climate Variability and Change and 24th Conference on Hydrology*, American Meteorological Society, 2010.

Tessicini, F., G. Iaccarino, M. Fatica, M. Wang, and R. Verzicco, Wall modeling for large-eddy simulation using an immersed boundary method, *Center for Turbulence Research, Annual Research Briefs*, pp. 181–187, 2002.

Tseng, Y., On the development of a ghost-cell immersed boundary method and its application to large eddy simulation and geophysical fluid dynamics, Ph.D. thesis, Stanford University, 2003.

Tseng, Y., and J. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comp. Phys.*, *192*, 593–623, 2003.

Tseng, Y., C. Meneveau, and M. Parlange, Modeling Flow around Bluff Bodies and Predicting Urban Dispersion Using Large Eddy Simulation, *Environ. Sci. Technol.*, *40*, 2653–2662, 2006.

von Terzi, D., M. Linnick, J. Seidel, and H. Fasel, Immersed Boundary Techniques for High-Order Finite-Difference Methods, in *31st AIAA Fluid Dynamics Conference and Exhibit*, p. 17, American Institute of Aeronautics and Astronautics, Anaheim, CA, 2001.

Walter, K., C. Weiss, A. Swift, J. Chapman, and N. Kelley, Speed and Direction Shear in the Stable Nocturnal Boundary Layer, *ASME J. Sol. Energy Eng.*, *131*, 1–7, 2009.

Wicker, L., and W. Skamarock, Time-Splitting Methods for Elastic Models Using Forward Time Schemes, *Mon. Weather Rev.*, *130*, 2088–2097, 2002.

Wyngaard, J., Toward numerical modeling in the “Terra Incognita”, *J. Atmos. Sci.*, *61*, 1816–1826, 2004.

Yasuda, N., Turbulent Diffusivity and Diurnal Variation in the Atmospheric Boundary Layer, *Boundary-Layer Meteorol.*, *43*, 209–221, 1988.

Zängl, G., An Improved Method for Computing Horizontal Diffusion in a Sigma-Coordinate Model and Its Application to Simulations over Mountainous Topography, *Mon. Weather Rev.*, *130*, 1423–1432, 2002.

Zängl, G., A Generalized Sigma-Coordinate System for the MM5, *Mon. Weather Rev.*, *131*, 2875–2884, 2003.

Zängl, G., The sensitivity of simulated orographic precipitation to model components other than cloud microphysics, *Q. J. R. Meteorol. Soc.*, *130*, 1857–1875, 2004.

Zängl, G., L. Gantner, G. Hartjenstein, and H. Noppel, Numerical errors above steep topography: A model intercomparison, *Meteorol. Z.*, *13*, 69–76, 2004.

Appendix A

Scalar advection test case under stable atmospheric conditions

Results are presented in this appendix for the scalar advection test case used in chapter 4 under stable atmospheric conditions. Results are only presented for the case of terrain-following coordinates, and the default WRF advection options, which are 5th order in the horizontal and 3rd order in the vertical. The set-up of the simulation is identical to the set-up used in chapter 4, with the exception of the potential temperature initialization. Potential temperature is initialized as given in a standard atmosphere, as was also done in Zängl [2003]. The standard atmosphere specifies a temperature of 288 K at sea level, a tropospheric vertical temperature gradient of -6.5 K km^{-1}, and an isothermal atmosphere above the Tropopause at a height of 11 km.

Figure A.1 shows contours of u and w velocity for the stable atmospheric case, as in figure 4.3 for the neutral case. At the end of the simulation (t = 10,000 s) horizontal velocities range from -0.53 to 11.72 m s^{-1}, and vertical velocities range from -0.59 to 0.65 m s^{-1}. This is much less error than seen in the neutral case, where u ranged from -5.8 to 14.1 m s^{-1} and w ranged from more than -4 to 4 m s^{-1}. Errors in the velocity field for the stable WRF case are still much larger than when IBM is used, where the error is negligible. Additionally, the errors for this simulation (using terrain-following coordinates and a stable atmosphere) are larger than seen by Zängl [2003] using the native MM5 terrain-following coordinate.

Figure A.2 shows contours of the scalar concentration, and error calculated as the difference from the analytical solution. This figure is as in figure 4.4 for the neutral case. At the end of the simulation, the minimum and maximum scalar concentrations are -0.04 and 0.83, which have non-dimensional units. The absolute error ranges from -0.18 to 0.19 at t = 10,000 s in the stable case, whereas it ranges from -0.77 to 0.67 in the neutral case. Again, the errors for this simulation are larger than in Zängl [2003], where errors in scalar concentration ranged from -0.12 to 0.10 with a stable atmosphere.

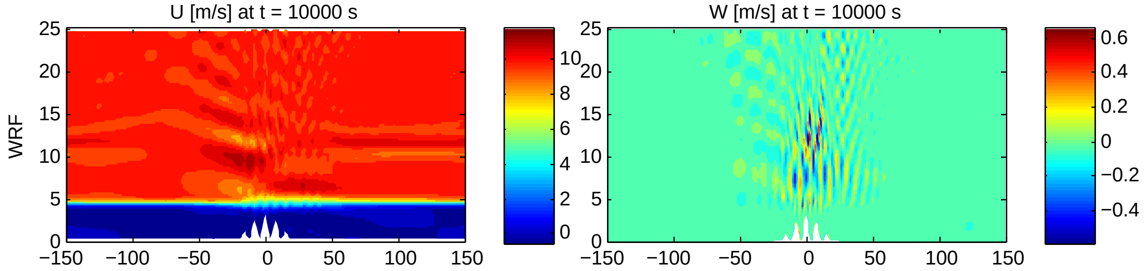


Figure A.1. This figure is as in figure 4.3, except with a stable atmosphere. Contours of the u and w components of velocity in m s^{-1} for terrain-following coordinates at t = 10000 s. Axes indicate domain size in km, and are not to scale.

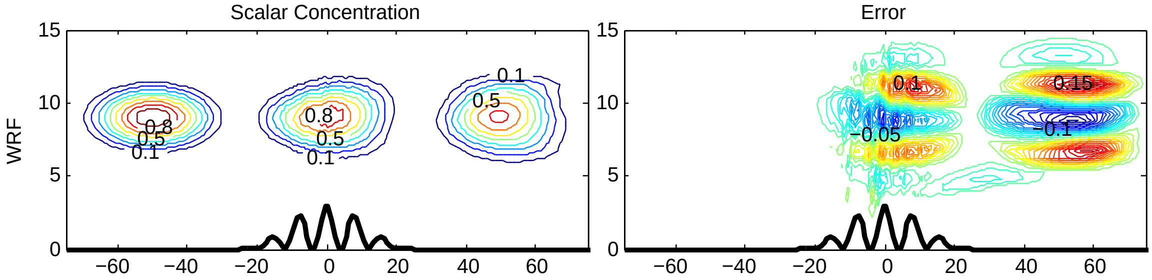


Figure A.2. This figure is as in figure 4.4, except with a stable atmosphere. On the left, the scalar concentration is shown at t = 0, 5000, and 10000 s. Scalar units are non-dimensional with a range of 0 to 1. Contour intervals are in 0.1 increments. Error is shown on the right, and is calculated as the difference between the numerical and analytical solutions. Contour intervals are 0.01. The zero contour is suppressed. Axes indicate domain size in km, and are not to scale.

Appendix B

Owens Valley simulations

The IBM allows explicit resolution of steep mountainous terrain, enabling WRF to simulate flows which cannot be computed using a standard terrain-following coordinate. To demonstrate this capability, we have modeled flow over a two-dimensional slice of the Owens Valley in California. The terrain slice, seen in figure B.2, is perpendicular to the valley axis. The terrain data is from the National Elevation Dataset (NED) at a resolution of 1/3 arc-second or approximately 10 m. As the terrain slice is not aligned with latitudinal coordinates, it was necessary to interpolate the data onto the grid. After the interpolation, the terrain resolution used in the simulation is 20 m. As a rule of thumb, terrain-following coordinates should not be used for slopes over 30 degrees. The slope of the 20 m terrain data is plotted in figure B.1, and it can be seen that there are several slopes of 60 degrees in this valley profile.

The domain size is (X, Y, Z) = (58.32 km, 0.04 km, 10 km) with (n_x, n_y, n_z) = (1459,2,120) grid points. The grid spacing is $\Delta X = \Delta Y = 40$ m in the horizontal. The minimum grid spacing in the vertical is $\Delta Z_{min} = 49.0$ m, and the maximum is $\Delta Z_{max} =$ 130.6 m. The initialization is the same as in the fully coupled cases in section 6.3.2, and as in those cases the soil properties are idealized.

Figures B.2 and B.3 are examples of flows in the Owens Valley. Figure B.2 shows a typical morning upslope flow due to diurnal heating. Figure B.3 shows westerly flow over the Sierra Nevada mountain range. Mountain waves are seen over the valley.

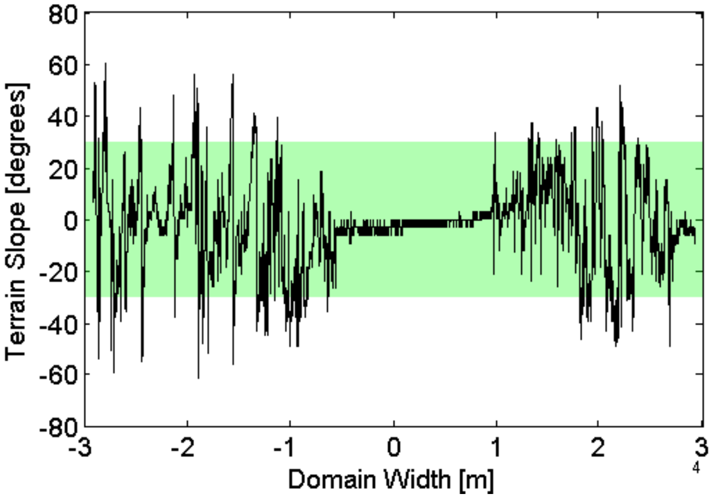


Figure B.1. Slope of the Owens Valley terrain data is plotted with the black line. The green background depicts the typical limits of terrain-following coordinates.

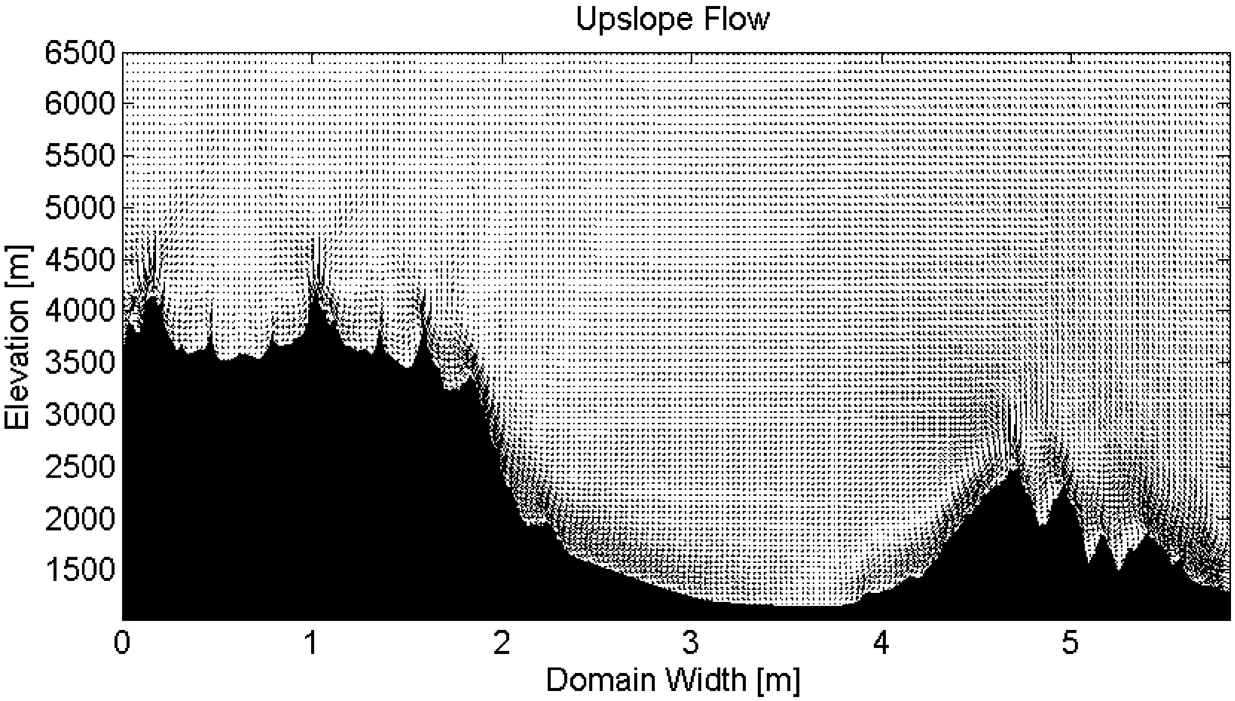


Figure B.2. Upslope flow induced by diurnal heating.

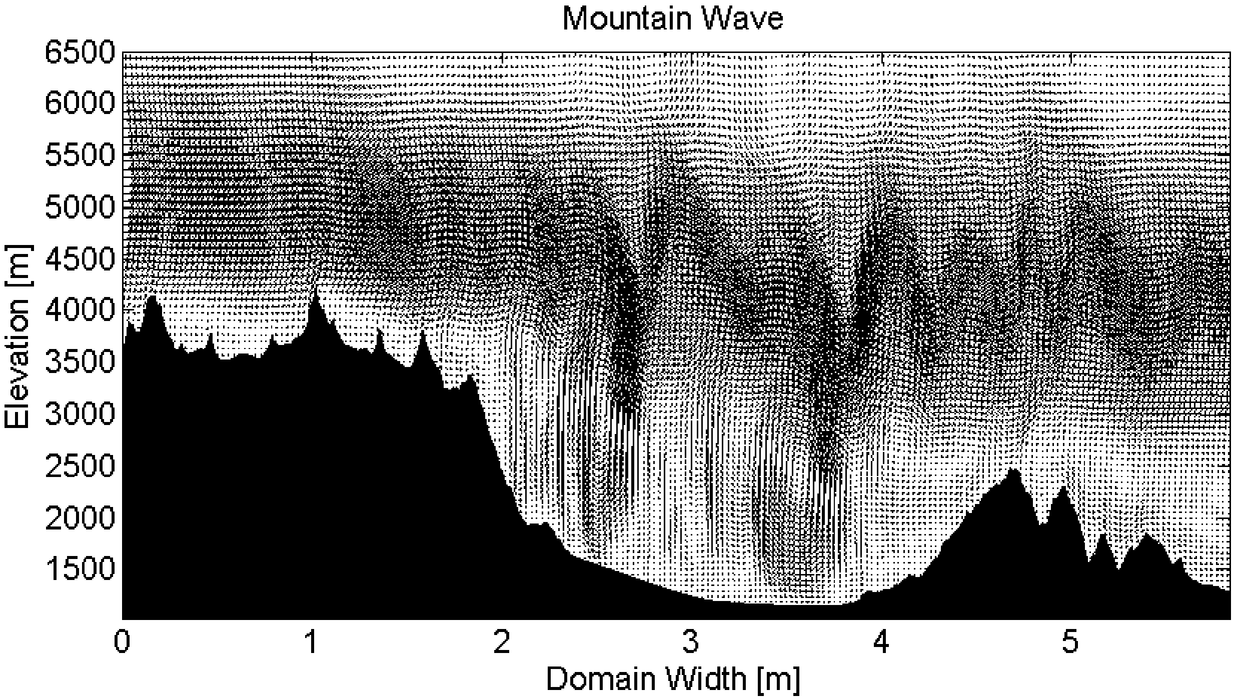


Figure B.3. Mountain wave over the Owens Valley from westerly flow over the Sierra Nevada mountain range.

Appendix C

American Meteorological Society full copyright notice