LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TH-226657

Implementation of the Immersed Boundary Method in the Weather Research and Forecasting model

K. A. Lundquist

December 8, 2006

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Implementation of the Immersed Boundary Method in the Weather Research and Forecasting model

by

Katherine Ann Lundquist

B.S. (University of Texas at Austin) 2000

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

Masters of Science, Plan II

in

Mechanical Engineering

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Fotini Katopodes Chow, Chair
Professor Stanley Berger
Julie Lundquist, Ph.D.

Fall 2006

The Plan II project of Katherine Ann Lundquist is approved.

Chair Date

Date

Date

University of California, Berkeley

Fall 2006

Abstract

Implementation of the Immersed Boundary Method in the Weather Research and Forecasting model

by

Katherine Ann Lundquist

Masters of Science, Plan II in Mechanical Engineering

University of California, Berkeley

Professor Fotini Katopodes Chow, Chair

Accurate simulations of atmospheric boundary layer flow are vital for predicting dispersion of contaminant releases, particularly in densely populated urban regions where first responders must react within minutes and the consequences of forecast errors are potentially disastrous. Current mesoscale models do not account for urban effects, and conversely urban scale models do not account for mesoscale weather features or atmospheric physics. The ultimate goal of this research is to develop and implement an immersed boundary method (IBM) along with a surface roughness parameterization into the mesoscale Weather Research and Forecasting (WRF) model. IBM will be used in WRF to represent the complex boundary conditions imposed by urban landscapes, while still including forcing from regional weather patterns and atmospheric physics. This document details preliminary results of this research, including the details of three distinct implementations of the immersed boundary method. Results for the three methods are presented for the case of a rotation influenced neutral atmospheric boundary layer over flat terrain.

Contents

List of Figures

List of Tables

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Chapter 1

Introduction

Large Eddy Simulation (LES) of the atmosphere has become the preferred method for computational research in mesoscale weather prediction. While LES is still uncommon in operational forecasting due the limits of current computing power, it will likely become the preferred method in the future. As atmospheric LES codes are widely used in the research community and promise to be the future standard for operational forecasting, there is a great deal of interest on the part of academic researchers in improving the forecasting ability of these models. Technology developed by researchers has a direct path to transfer to operational use through the updating and distribution of open source numerical weather prediction codes. Atmospheric simulations often include complex terrain and are coarsely resolved, leaving the accuracy of the prediction to be determined by the quality of the parameterizations. The prolific use of parameterizations results in many challenges and opportunities for improvement by researchers. One such opportunity is improving the treatment of surface boundary conditions for complex terrain through the use of the immersed boundary method (IBM). For this project, three different immersed boundary treatments were implemented into the Weather Research and Forecasting (WRF) model which is a mesoscale code with LES options suitable for both research and operational use. The effects of the IBM implementations were studied for neutral atmospheric boundary layer flow. While the neutral boundary layer case is greatly idealized, the study provides insight on

which implementation should be further pursued for use with three-dimensional complex terrain.

1.1 Motivation

Accurate prediction of velocity fields and turbulent kinetic energy from mesoscale weather forecasting models is necessary to predict dispersion of hazardous substances such as those released in an industrial accident or malicious attack. Accurate plume predictions become even more vital in densely populated urban areas where they are used to aid in emergency response planning. Governmental agencies such as the National Atmospheric Release Advisory Center (NARAC) at Lawrence Livermore National Laboratory (LLNL) are responsible for providing dispersion predictions in emergency situations. Dispersion models used by these agencies often rely on inputs from mesoscale numerical weather forecasting models; therefore it is imperative that the predictions are correct. Current mesoscale models can not accurately deal with heterogeneous urban regions, and have been shown to over predict wind velocity and shear in those cases. These over predicted quantities lead to increased dispersion and consequently an underestimation of the contaminant concentration and associated risks.

The National Center for Atmospheric Research (NCAR) has developed the numerics for a new, sophisticated mesoscale model known as the Weather and Research Forecasting (WRF) model. This model, like other mesoscale codes, is subject to several limitations and is currently not able to explicitly represent flow in urban areas. One such limitation is the use of terrain-following vertical coordinates which is standard in WRF and in other mesoscale models. Steep terrain gradients such as those formed by tall buildings cause extreme distortion of the near surface grid cells in terrain following coordinates, which makes it difficult if not impossible to obtain accurate forecasts in urban environments. Because of this meshing limitation in mesoscale models, classic computational fluid dynamics (CFD) codes, typically with unstructured grids, are currently used instead of atmospheric codes such as WRF when urban-scale resolution is desired. Urban-scale CFD models are forced

with simplified boundary conditions, neglecting the effects of regional weather patterns. Additionally, they do not normally include atmospheric processes such fluxes of surface heat and moisture. Idealized boundary conditions and the absence of atmospheric processes are presently major sources of error in CFD simulations of urban dispersion. Thus current mesoscale models, which simulate regional-scale atmospheric processes, are unable to represent urban areas, while current CFD models, which can simulate complex geometries, are unable to incorporate realistic atmospheric physics. This research seeks to improve the ability of WRF to predict dispersion of contaminants in urban environments by allowing simulations at urban-scale resolution. This requires improvements of numerical representations of velocity fields and turbulent kinetic energy in regions of complex urban terrain. Improvements to the mesoscale model will be accomplished by modifying the approach used for the boundary conditions of the urban terrain.

Once the project is completed it will be possible to model an urban environment in detail using the mesoscale model WRF. This will be the first time that a mesoscale model will include the functionality needed to model a domain of the urban scale by accurately representing complex building surfaces. Improved results from urban simulations will lead to more accurate numerical representations of contaminant dispersion. Once heterogeneous urban environments can be modeled in WRF, 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 and would be a large contribution to the atmospheric dispersion modeling community.

1.2 Overview

The goal of this research is to develop improved surface representations that are suitable for accurately dealing with the complex terrain found in an urban setting. This will be

accomplished by adding immersed boundary method (IBM) capabilities to WRF. WRF uses a terrain-following coordinate system; therefore topographical gradients skew computational cells near the surface. The implementation of IBM will enable the representation of steep surface gradients like those formed by buildings, which normally cause WRF to fail due to a combination of grid skewing and strong vertical velocity components. 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 WRF. Rather than explicitly adding structures to the terrain surface, 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.

Several different implementations of IBM have been posed by researchers. The first step in implementing IBM into WRF is to study the effects of various techniques used in IBM. The simplest canonical case for geophysical flows is that of a three-dimensional neutral atmospheric boundary layer. Other canonical cases often used for verifying numerical models are two-dimensional flow over an isolated hill and three-dimensional flow over a Gaussian bump. These test cases provide an excellent proof of concept, and verify accurate implementation of the new method for treating boundary conditions. Results are presented for three variants of the IB method for the neutral boundary layer case. The next step is to extend the IBM formulation to complex terrain, and reproduce the isolated hill and Gaussian bump cases.

Finally, IBM will be applied to a real urban case. Field data exists for an urban environment from the Joint URBAN 2003 field campaign in Oklahoma City, in which LLNL participated. The field data is extensive and includes measurements of velocity profiles, temperature, turbulent kinetic energy, and concentration from releases of a passive scalar. Once IBM is implemented in three dimensions the one meter terrain data for Oklahoma City can be accurately represented in WRF. Using input data from the intensive observation periods in Oklahoma City, comparisons can be made between predicted and real measurements of velocity fields, temperature, turbulent kinetic energy, and scalar concentrations.

Chapter 2

Details of the Weather Research and Forecasting Model

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 will 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, and the EM core is generally used for research purposes. The EM core is also known as Advanced Research WRF (ARW), and is the core that is the focus of the following discussion. ARW has principally been developed by the National Center for Atmospheric Research (NCAR). As of 2006 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 hydrostatic 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 needed to account for the 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.

In this chapter, further details are given on the WRF model. The governing equations are derived in Section 2.1. 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 hydrostatic pressure coordinate used in WRF. Section 2.2 discusses the time advancement scheme used in WRF, and Section 2.3 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 Chapter 3 on the implementation of the immersed boundary method.

2.1 The Governing Equations of WRF

WRF uses a terrain following hydrostatic pressure coordinate, shown in Figure 2.1. Historically, it was convenient in many meteorological applications to use pressure as an

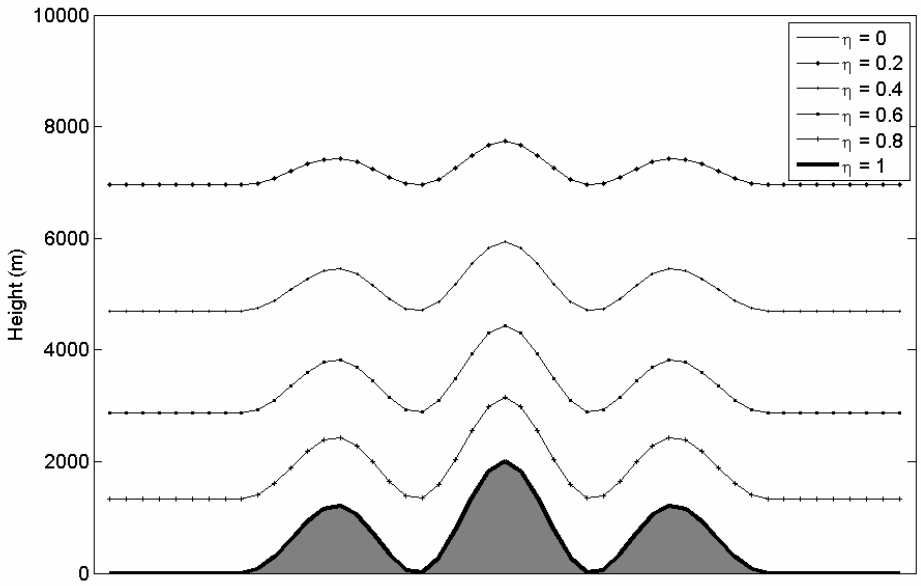

Figure 2.1. An example of the η coordinate used by WRF.

independent variable instead of height. The notion of pressure as an independent coordinate in 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 equations of motion 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 inspiration for the framework of WRF, which solves the perturbation form of these equations for a moist atmosphere.

2.1.1 The Compressible Euler Equations

The prognostic equations governing the WRF model are given in the NCAR technical note NCAR/TM-468+STR, 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 (2.1).

$$\begin{aligned} \frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V} + \alpha \nabla p + \vec{g} &= F \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) &= 0 \end{aligned} \tag{2.1}$$

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 hydrostatic pressure coordinate. This will be 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.

2.1.2 Transformation to Hydrostatic 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:

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

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 (2.2a) and (2.2b) respectively.

$$\left(\frac{\partial a}{\partial t}\right)_\pi = \left(\frac{\partial a}{\partial t}\right)_z + \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial t}\right)_\pi \frac{\partial a}{\partial \pi} \tag{2.2a}$$
$$\left(\frac{\partial a}{\partial x}\right)_\pi = \left(\frac{\partial a}{\partial x}\right)_z + \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial x}\right)_\pi \frac{\partial a}{\partial \pi} \tag{2.2b}$$

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

$$\begin{aligned}\left(\frac{Da}{Dt}\right)_\pi = &\left(\frac{\partial a}{\partial t}\right)_\pi - \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial t}\right)_\pi \frac{\partial a}{\partial \pi} \\ &+ u\left[\left(\frac{\partial a}{\partial x}\right)_\pi - \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial x}\right)_\pi \frac{\partial a}{\partial \pi}\right] \\ &+ v\left[\left(\frac{\partial a}{\partial y}\right)_\pi - \frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial y}\right)_\pi \frac{\partial a}{\partial \pi}\right] \\ &+ w\frac{\partial \pi}{\partial z}\frac{\partial a}{\partial \pi}.\end{aligned} \tag{2.3}$$

Rearranging the terms of the material derivative by distributing the u and v velocities to the terms inside of the square brackets forms equation (2.4).

$$\left(\frac{Da}{Dt}\right)_\pi = \left(\frac{\partial a}{\partial t}\right)_\pi + u\left(\frac{\partial a}{\partial x}\right)_\pi + v\left(\frac{\partial a}{\partial y}\right)_\pi + \left[w - \left(\frac{\partial z}{\partial t} + u\frac{\partial z}{\partial x} + v\frac{\partial z}{\partial y}\right)_\pi\right]\frac{\partial \pi}{\partial z}\frac{\partial a}{\partial \pi} \tag{2.4}$$

The material derivative can then be arranged in the familiar form of (2.5a), where the expression for $\dot{\pi}$ or $\frac{\partial \pi}{\partial t}$ is defined to be of the form (2.5b).

$$\left(\frac{Da}{Dt}\right)_\pi = \left(\frac{\partial a}{\partial t}\right)_\pi + u\left(\frac{\partial a}{\partial x}\right)_\pi + v\left(\frac{\partial a}{\partial y}\right)_\pi + \dot{\pi}\frac{\partial a}{\partial \pi} \tag{2.5a}$$

$$\dot{\pi} = \frac{\partial \pi}{\partial t} = \frac{\partial \pi}{\partial z}\left[w - \left(\frac{\partial z}{\partial t} + u\frac{\partial z}{\partial x} + v\frac{\partial z}{\partial y}\right)_\pi\right] \tag{2.5b}$$

Equation (2.5b) can be solved for the vertical velocity w, which yields equation (2.6).

$$w = \left(\frac{\partial z}{\partial t} + u\frac{\partial z}{\partial x} + v\frac{\partial z}{\partial y}\right)_\pi + \dot{\pi}\frac{\partial z}{\partial \pi} \tag{2.6}$$

Additionally, after many instances of the chain rule, the vertical derivative of w is also found to be equation (2.7).

$$\frac{\partial w}{\partial z} = \frac{\partial w}{\partial \pi}\frac{\partial \pi}{\partial z} = \frac{\partial \pi}{\partial z}\left[\left[\frac{\partial}{\partial t}\left(\frac{\partial z}{\partial \pi}\right) + u\frac{\partial}{\partial x}\left(\frac{\partial z}{\partial \pi}\right) + \frac{\partial u}{\partial \pi}\frac{\partial z}{\partial x} + v\frac{\partial}{\partial y}\left(\frac{\partial z}{\partial \pi}\right) + \frac{\partial v}{\partial \pi}\frac{\partial z}{\partial y}\right]_\pi + \dot{\pi}\frac{\partial}{\partial \pi}\left(\frac{\partial z}{\partial \pi}\right) + \frac{\partial \dot{\pi}}{\partial \pi}\frac{\partial z}{\partial \pi}\right] \tag{2.7}$$

Using the definition of the material derivative in the π coordinate system given by the set of equations in (2.5), the vertical derivative of w can be significantly simplified from the above form. Once simplified, the vertical derivative of w is expressed as (2.8).

$$\frac{\partial w}{\partial z} = \frac{\partial w}{\partial \pi}\frac{\partial \pi}{\partial z} = \frac{\partial \pi}{\partial z}\left[\frac{D}{Dt}\left(\frac{\partial z}{\partial \pi}\right) + \frac{\partial u}{\partial \pi}\frac{\partial z}{\partial x} + \frac{\partial v}{\partial \pi}\frac{\partial z}{\partial y}\right]_\pi + \frac{\partial \dot{\pi}}{\partial \pi} \tag{2.8}$$

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 (2.2), and the partial derivative with respect to z is given by equation (2.8). After these substitutions, the continuity equation in the pressure coordinate system is (2.9).

$$\left(\frac{\partial \rho}{\partial t}\right)_\pi - \frac{\partial \rho}{\partial \pi}\frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial t}\right)_\pi + \left(\frac{\partial(\rho u)}{\partial x}\right)_\pi - \frac{\partial(\rho u)}{\partial \pi}\frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial x}\right)_\pi + \left(\frac{\partial(\rho v)}{\partial y}\right)_\pi - \frac{\partial(\rho v)}{\partial \pi}\frac{\partial \pi}{\partial z}\left(\frac{\partial z}{\partial y}\right)_\pi + \frac{\partial \rho \dot{\pi}}{\partial \pi} + \frac{\partial \pi}{\partial z}\left[\frac{D}{Dt}\left(\rho\frac{\partial z}{\partial \pi}\right) + \frac{\partial(\rho u)}{\partial \pi}\frac{\partial z}{\partial x} + \frac{\partial(\rho v)}{\partial \pi}\frac{\partial z}{\partial y}\right]_\pi = 0 \tag{2.9}$$

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].

$$\left[\frac{\partial}{\partial t}(\rho\frac{\partial z}{\partial \pi}) + \frac{\partial}{\partial x}(\rho u\frac{\partial z}{\partial \pi}) + \frac{\partial}{\partial y}(\rho v\frac{\partial z}{\partial \pi})\right]_\pi + \frac{\partial}{\partial \pi}\left(\rho\dot{\pi}\frac{\partial z}{\partial \pi}\right) = 0 \tag{2.10}$$

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

$$\left(\frac{D\vec{V}}{Dt}\right)_\pi + \alpha\nabla_\pi p - \alpha\frac{\partial \pi}{\partial z}(\nabla_\pi z)\frac{\partial p}{\partial \pi} = F, \tag{2.11}$$

and the vertical momentum equation is

$$\left(\frac{Dw}{Dt}\right)_\pi + \alpha\frac{\partial p}{\partial \pi}\frac{\partial \pi}{\partial z} + g = F. \tag{2.12}$$

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 (2.5a) 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 (2.13) 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

(2.13) geopotential has been introduced, and is defined as $\phi = gz$.

$$\left(\frac{\partial}{\partial t}\right)_\pi = \left(\frac{\partial}{\partial t}\right)_z - \rho\left(\frac{\partial \phi}{\partial t}\right)_\pi \frac{\partial}{\partial \pi} \tag{2.13a}$$

$$\nabla_\pi a = \nabla_z a - \rho(\nabla_\pi \phi)\frac{\partial a}{\partial \pi} \tag{2.13b}$$

$$\nabla_\pi \cdot \vec{V} = \nabla_z \cdot \vec{V} - (\rho \nabla_\pi \phi) \cdot \frac{\partial \vec{V}}{\partial \pi} \tag{2.13c}$$

$$\left(\frac{D}{Dt}\right)_\pi = \left(\frac{\partial}{\partial t}\right)_\pi + \vec{V} \cdot \nabla_\pi + \dot{\pi}\frac{\partial}{\partial \pi} \tag{2.13d}$$

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 (2.10) to be

$$\nabla_\pi \cdot \vec{V} + \frac{\partial \dot{\pi}}{\partial \pi} = 0, \tag{2.14}$$

and the horizontal and vertical momentum equations are

$$\left(\frac{D\vec{V}}{Dt}\right)_\pi + \alpha \nabla_\pi p + \frac{\partial p}{\partial \pi}\nabla_\pi \phi = F \tag{2.15}$$

$$\left(\frac{Dw}{Dt}\right)_\pi + g\left(1 - \frac{\partial p}{\partial \pi}\right) = F. \tag{2.16}$$

The set of equations above consisting of (2.14), (2.15), and (2.16) are the fully compressible non-hydrostatic Euler equations given by Laprise [1992].

2.1.3 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 (2.17) for the

η coordinate are analogous to those in (2.2) for the π coordinate.

$$\left(\frac{\partial a}{\partial t}\right)_\eta = \left(\frac{\partial a}{\partial t}\right)_\pi + \frac{\partial \eta}{\partial \pi}\left(\frac{\partial \pi}{\partial t}\right)_\eta \frac{\partial a}{\partial \eta} \tag{2.17a}$$

$$\left(\frac{\partial a}{\partial x}\right)_\eta = \left(\frac{\partial a}{\partial x}\right)_\pi + \frac{\partial \eta}{\partial \pi}\left(\frac{\partial \pi}{\partial x}\right)_\eta \frac{\partial a}{\partial \eta} \tag{2.17b}$$

The material derivative in the η coordinate is

$$\left(\frac{D}{Dt}\right)_\eta = \left(\frac{\partial}{\partial t}\right)_\eta + \vec{V}\cdot\nabla_\eta + \dot{\eta}\frac{\partial}{\partial \eta}, \tag{2.18}$$

where

$$\dot{\eta} = \frac{\partial \eta}{\partial t} = \frac{\partial \eta}{\partial \pi}\left[\dot{\pi} - \left(\frac{\partial \pi}{\partial t} + u\frac{\partial \pi}{\partial x} + v\frac{\partial \pi}{\partial y}\right)_\eta\right]. \tag{2.19}$$

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 (2.20).

$$\left[\frac{\partial}{\partial t}\left(\frac{\partial \pi}{\partial \eta}\right) + \frac{\partial}{\partial x}\left(u\frac{\partial \pi}{\partial \eta}\right) + \frac{\partial}{\partial y}\left(v\frac{\partial \pi}{\partial \eta}\right)\right]_\eta + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial \pi}{\partial \eta}\right) = 0 \tag{2.20}$$

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

$$\left(\frac{D\vec{V}}{Dt}\right)_\eta + \alpha\left(\nabla_\eta p - (\nabla_\eta \pi)\frac{\partial \eta}{\partial \pi}\frac{\partial p}{\partial \eta}\right) + \frac{\partial p}{\partial \eta}\frac{\partial \eta}{\partial \pi}\left(\nabla_\eta \phi - (\nabla_\eta \pi)\frac{\partial \eta}{\partial \pi}\frac{\partial p}{\partial \eta}\right) = F \tag{2.21a}$$

$$\left(\frac{Dw}{Dt}\right)_\eta + g\left(1 - \frac{\partial p}{\partial \eta}\frac{\partial \eta}{\partial \pi}\right) = F. \tag{2.21b}$$

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 (2.20), its form simplifies substantially. The simplified continuity equation below is equivalent to the equation given in the WRF description by Skamarock et al. [2005], 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.

$$\left(\frac{\partial \mu}{\partial t} + \frac{\partial(\mu u)}{\partial x} + \frac{\partial(\mu v)}{\partial y}\right)_\eta + \frac{\partial(\mu\dot{\eta})}{\partial \eta} = 0 \tag{2.22}$$

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:

$$\mu \left(\frac{D\vec{V}}{Dt} \right)_\eta - \frac{\partial \phi}{\partial \eta} \nabla_\eta p + \frac{\partial p}{\partial \eta} \nabla_\eta \phi = F. \tag{2.23}$$

Some manipulation is required to get the momentum equation into the strong conservation form, as it appears in [Skamarock et al., 2005, 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.

$$\left(\frac{\partial (\mu \vec{V})}{\partial t} \right)_\eta + \nabla_\eta \cdot (\mu \vec{V}; \vec{V}) + \frac{\partial}{\partial \eta}(\mu \dot{\eta} \vec{V}) - \frac{\partial}{\partial x} \left(p \frac{\partial \phi}{\partial \eta} \right) + \frac{\partial}{\partial \eta} \left(p \frac{\partial \phi}{\partial x} \right) = F \tag{2.24}$$

$$\left(\frac{\partial (\mu w)}{\partial t} \right)_\eta + \nabla_\eta \cdot (\mu \vec{V} w) + \frac{\partial}{\partial \eta}(\mu \dot{\eta} w) - g \left(\frac{\partial p}{\partial \eta} - \mu \right) = F \tag{2.25}$$

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.

$$\left(\frac{\partial (\mu \theta)}{\partial t} \right)_\eta + \nabla_\eta \cdot (\mu \vec{V} \theta) + \frac{\partial}{\partial \eta}(\mu \dot{\eta} \theta) = F_\theta \tag{2.26}$$

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.

$$p = p_o \left(\frac{R_{dry} \theta}{p_o \alpha_{dry}} \right)^{\gamma_{dry}} \tag{2.27}$$

2.1.4 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 $p = \bar{p}(z) + p\prime$, $\alpha = \bar{\alpha}(z) + \alpha\prime$, $\phi = \bar{\phi}(z) + \phi\prime$, and $\mu_d = \bar{\mu}_d(x, y) + \mu_d\prime$ 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 $\partial\phi\prime/\partial\eta = -\bar{\mu}\alpha\prime - \mu\prime\alpha$. Now the time invariance of the base state is used, and the equation for mass conservation becomes

$$\left(\frac{\partial\mu\prime}{\partial t} + \frac{\partial(\mu u)}{\partial x} + \frac{\partial(\mu v)}{\partial y}\right)_\eta + \frac{\partial(\mu\dot{\eta})}{\partial\eta} = 0. \tag{2.28}$$

In order to develop the perturbation form for the horizontal momentum equations, the form given in (2.29) is used as a starting point.

$$\left(\frac{\partial(\mu\vec{V})}{\partial t}\right)_\eta + \nabla_\eta \cdot (\mu\vec{V}; \vec{V}) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}\vec{V}) + \mu\alpha\nabla_\eta p + \frac{\partial p}{\partial\eta}\nabla_\eta\phi = F \tag{2.29}$$

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 $\mu\bar{\alpha}\nabla_\eta\bar{p} + \mu\alpha\nabla_\eta p\prime + \mu\alpha\prime\nabla_\eta\bar{p} + \mu\nabla_\eta\bar{\phi} + \mu\nabla_\eta\phi\prime + \frac{\partial p\prime}{\partial\eta}(\nabla_\eta\phi) - \mu\prime(\nabla_\eta\phi)$. When the base state is subtracted, the perturbation form of the horizontal momentum equation is

$$\left(\frac{\partial(\mu\vec{V})}{\partial t}\right)_\eta + \nabla_\eta \cdot (\mu\vec{V}; \vec{V}) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}\vec{V}) + \mu\alpha\nabla_\eta p\prime + \mu\alpha\prime\nabla_\eta\bar{p} + \mu\nabla_\eta\phi\prime + (\nabla_\eta\phi)\left(\frac{\partial p\prime}{\partial\eta} - \mu\prime\right) = F. \tag{2.30}$$

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

$$\left(\frac{\partial(\mu w)}{\partial t}\right)_\eta + \nabla_\eta \cdot (\mu\vec{V}w) + \frac{\partial}{\partial\eta}(\mu\dot{\eta}w) - g\left(\frac{\partial\bar{p}}{\partial\eta} + \frac{\partial p\prime}{\partial\eta} - \bar{\mu} - \mu\prime\right) = F \tag{2.31}$$

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

$$\left(\frac{\partial(\mu w)}{\partial t}\right)_\eta + \nabla_\eta \cdot (\mu \vec{V} w) + \frac{\partial}{\partial \eta}(\mu \dot{\eta} w) - g\left(\frac{\partial p\prime}{\partial \eta} - \mu\prime\right) = F. \tag{2.32}$$

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 (2.28), (2.30), and (2.32).

2.2 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. The large time step is defined by the user, and the acoustic time step can be user defined or calculated by WRF. If n acoustic time steps are specified by the user then 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. It is common to use six to twelve acoustic time steps per large time step. The time advancement scheme proceeds

Table 2.1. The time advancement sequence used in WRF.

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

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 2.1.

2.3 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. [2005] and include periodic, open or radiative, symmetric, and specified. 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. Nesting is not currently implemented in the vertical direction. This means that 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 from the coarse domain. In a two way nest, the fine domain still receives boundary conditions from the coarse domain, but at each time step the solution from the fine domain replaces the solution on the coarse domain.

The staggered grid, shown in Figure 2.2, 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

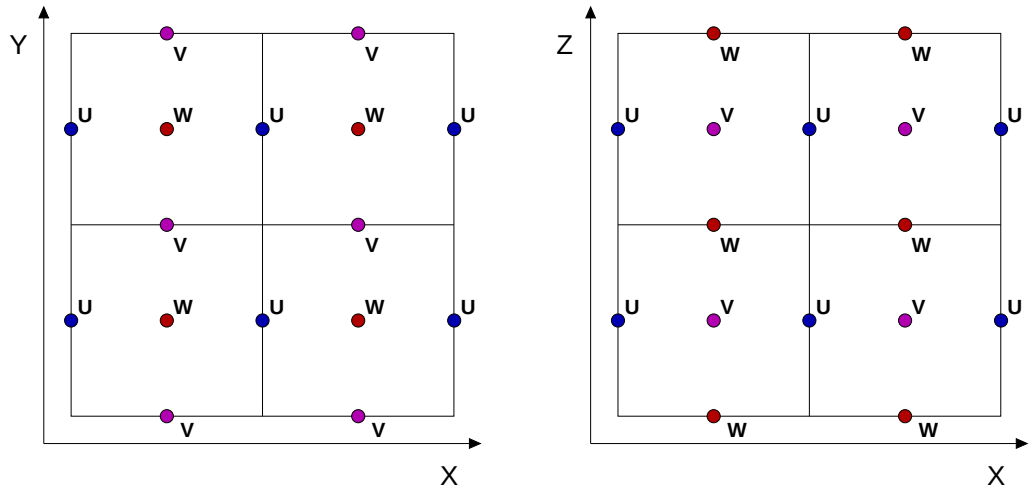

Figure 2.2. A plan view and elevation view of the WRF staggered grid.

the x direction. Conversely in the y direction there are (ny) v grid points, and $(ny-1)$ u and w grid points. In real cases, WRF includes six extra grid points outside of the domain in each of the horizontal extents. 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. 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 (2.33a) and (2.33b) create a free slip bottom boundary condition.

$$\dot{\eta}_{surf} = 0 \tag{2.33a}$$

$$w_{surf} = u_{surf}\frac{\partial h}{\partial x} + v_{surf}\frac{\partial h}{\partial y} \tag{2.33b}$$

In equation (2.33b) 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 by specifying a user determined coefficient of drag.

In the case of a rough terrain surface, the shear stress at the wall is calculated using (2.34), where C_d is the coefficient of drag and $|U|$ is the magnitude of the horizontal

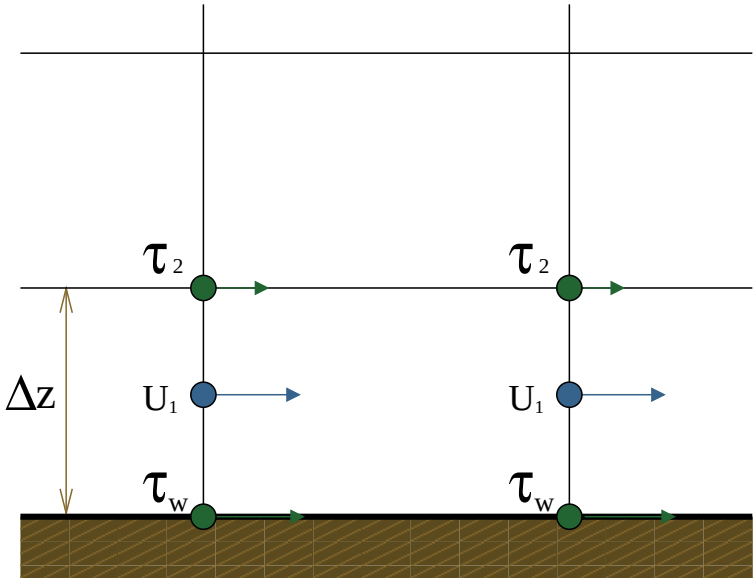

Figure 2.3. WRF sets τ_w to model a rough surface.

velocities.

$$\tau_{w_{xz}} = C_d|U|u, \qquad \tau_{w_{yz}} = C_d|U|v \tag{2.34}$$

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 (2.35).

$$\left.\frac{\partial(\mu u)}{\partial t}\right|_1 = \ldots + \mu\left[\frac{\tau_2}{\Delta z} - \frac{\tau_w}{\Delta z}\right] \tag{2.35}$$

Diffusion terms are calculated in physical space, so the use of the z coordinate (instead of η) is appropriate here. Equation (2.35) along with equation (2.33) 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 additional 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 3

Implementation of the Immersed Boundary Method

The numerical simulation of atmospheric flow over complex geometries, especially those geometries found in 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 can not 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 are represented with the addition of a forcing term in the Navier-Stokes equations. This chapter gives a brief history of the development of IBM in Section 3.1, and details three different IBM implementations in Section 3.2. Each of the three implementations has been used to model a neutral boundary layer, and the results of that study are given in Chapter 4.

3.1 Background of the Immersed Boundary Method

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

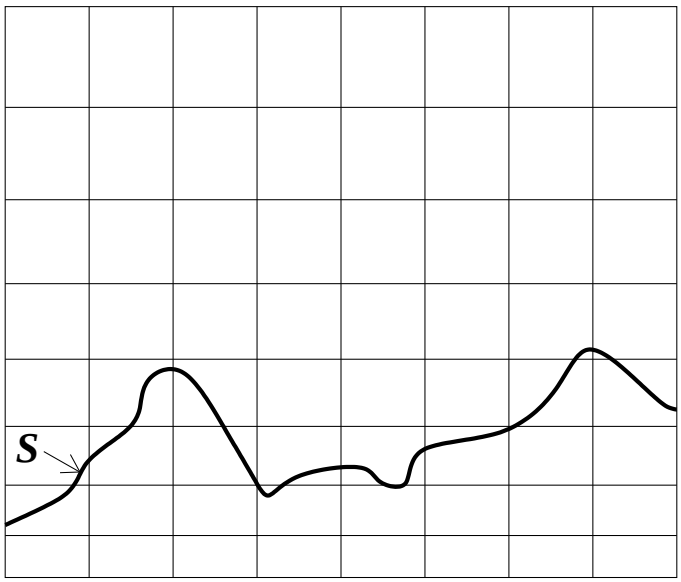

Figure 3.1. Cartoon sketch of an immersed boundary.

the original formulation of IBM, the incompressible Navier-Stokes equations as given by (3.1) 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 F in the equation for the conservation of momentum.

$$\rho(\vec{u}_t + \vec{u} \cdot \nabla\vec{u}) = -\nabla p + \mu\nabla^2\vec{u} + F \tag{3.1a}$$

$$\nabla \cdot \vec{u} = 0 \tag{3.1b}$$

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 forces are transfered 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].

3.1.1 Feedback Forcing Methods

IBM was first employed for rigid boundary applications by Briscolini and Santangelo [1989] and Goldstein, Handler, and Sirovich [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 (3.2). 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 $\mathbf{v}$.

$$\mathbf{F}(\mathbf{x_s}, t) = \alpha \int_0^t (\mathbf{U}(\mathbf{x_s}, t) - \mathbf{v}(\mathbf{x_s}, t))dt + \beta(\mathbf{U}(\mathbf{x_s}, t) - \mathbf{v}(\mathbf{x_s}, t)) \tag{3.2}$$

If the forcing term is balanced by the unsteady term in the Navier-Stokes equation, as shown in (3.3), 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.

$$\frac{\partial \mathbf{u}}{\partial t} \approx \alpha \int_0^t \mathbf{u}dt + \beta\mathbf{u} \tag{3.3}$$

The disadvantage of feedback forcing is that the formulation contains parameters α and β which the user must tune according to the frequency of 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.

3.1.2 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 (3.4), 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.

$$\mathbf{F} = \frac{\mathbf{v} - \mathbf{U^n}}{\Delta t} + \mathbf{U} \cdot \nabla \mathbf{U} + \frac{1}{\rho} \nabla p - \nu \nabla^2 \mathbf{U} \tag{3.4}$$

Fadlun, Verzicco, Orlandi, and Mohd-Yusof [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 (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.

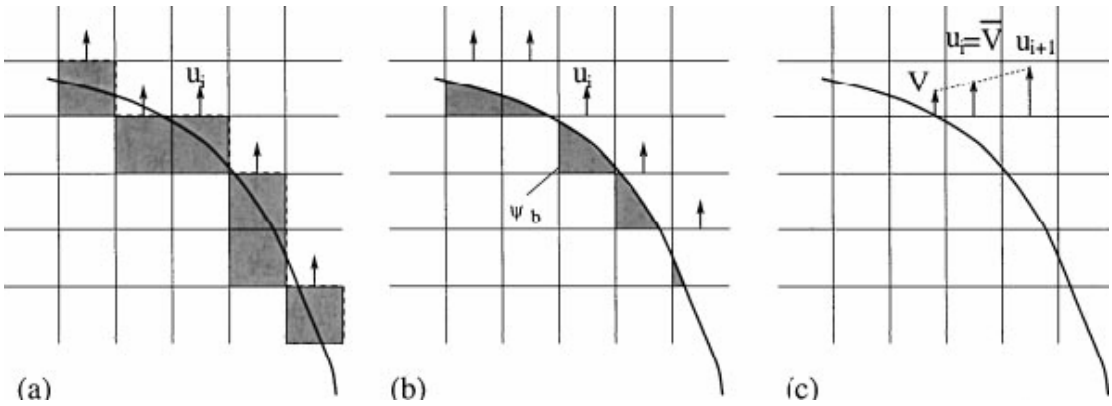

Figure 3.2. Figure from Fadlun et al. [2000] depicting velocity reconstruction interpolation methods: (a) stepwise geometry, (b) volume fraction weighting, (c) linear velocity interpolation.

3.1.3 Interpolation Methods for Boundary Reconstruction

The forcing formulations found in Sections 3.1.1 and 3.1.2 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 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 3.2. 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 boundary 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

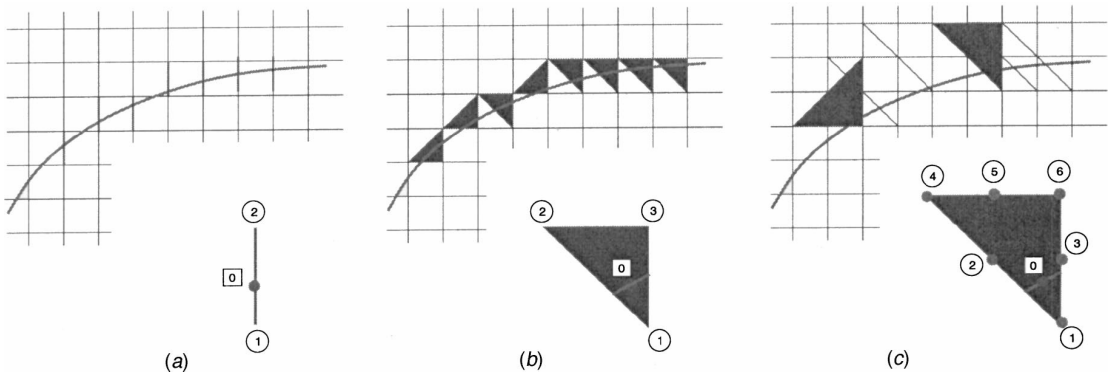

Figure 3.3. Figure from Iaccarino and Verzicco [2003] proposing several ghost point extrapolation methods: (a) linear one-dimensional, (b) linear multi-dimensional, (c) quadratic multi-dimensional.

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 3.3. 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 used 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 IB

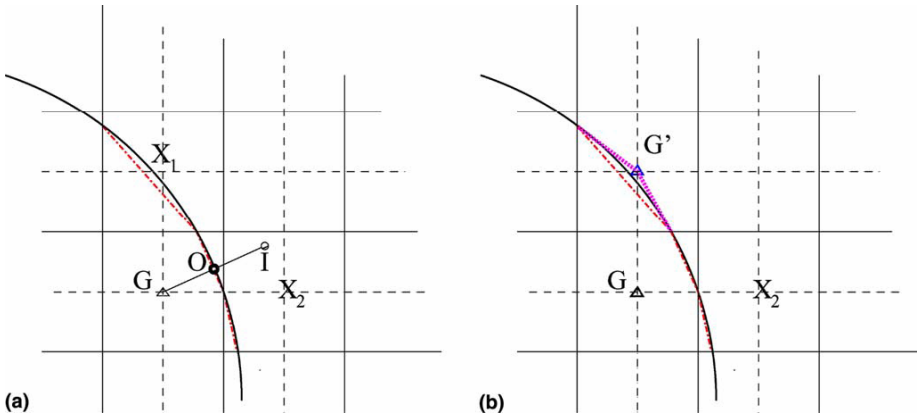

Figure 3.4. Figure from Tseng and Ferziger [2003] proposing two treatments to minimize numerical instability: (a) image method, (b) piecewise approximation

is very near the fluid nodes in the interpolation stencil, and far from the ghost point. Tseng and Ferziger [2003] presented two methods (illustrated in Figure 3.4) for dealing with this difficulty. In the image method, the interpolation scheme is used to solve for the fluid properties at the image point (I). Linear interpolation is then used to assign a value to the ghost point found by $\phi_G = 2\phi_O - \phi_I$. 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, however, in this case the boundary condition is assigned to the ghost node and no interpolation is necessary.

The interpolation method used to represent a rigid boundary is obviously an integral part of the overall immersed boundary method. It has 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 above ideas are the basis for more complicated methods. For instance, Kang, Iaccarino, and Moin [2004] present four additional interpolation methods using various stencils based on the ideas of Fadlun et al. [2000].

3.2 Implementations of IBM in WRF

IBM has successfully been used to model complex external flows like bluff body simulations of an automobile [Mittal and Iaccarino, 2005] and internal flows like flow in a stirred

tank [Iaccarino and Verzicco, 2003]. In these cases IBM is used in a classic CFD code, which solves the Navier-Stokes equations. The advantage here is that the flow is solved on a Cartesian mesh, and an unstructured boundary fitted grid does not need to be generated or solved. Finite difference methods can easily be implemented on a structured grid, and the algebraic system of equations to be solved has a diagonal structure. Unstructured grids are often better suited to finite element or finite volume methods. The nodes map to each other in a more complicated way, which leads to an irregular data structure that takes more computational effort to solve. Another undesirable aspect of unstructured grids is that they often require tedious pre-processing or cleaning of the geometry, and generation of a high quality grid can be extremely time consuming.

Currently in atmospheric modeling, urban scale simulations are performed using a CFD solver and mesoscale simulations are performed using a numerical weather prediction (NWP) code like WRF. In general, unstructured grids are used to model the extremely complex terrain found at the urban scale. The drawback here is that traditional CFD codes do not include atmospheric physics like surface fluxes, cloud formation, and radiation. Additionally, CFD codes often must be forced at the boundaries with idealized flows, neglecting the temporal variability of regional weather effects. Atmospheric physics parameterizations and regional weather effects are included in NWP codes, but the common use of terrain following coordinates makes it impossible to represent extremely complex geometries. So, unlike the above examples, the implementation of IBM into WRF is not to avoid tedious grid generation or the complications of solving irregular data structures. Instead IBM would negate the weaknesses of terrain following grids, and allow WRF to be used for urban simulations so that atmospheric physics and weather effects are included.

Three different IBM implementations have been added to WRF through the addition of a Fortran module. The implementations are a no slip boundary as in Iaccarino and Verzicco [2003], velocity reconstruction as in Fadlun et al. [2000] with the log-law modifications of Senocak et al. [2004], and shear stress reconstruction which is a blending of the original WRF boundary conditions with ghost cell IBM. Details of the three implementations are

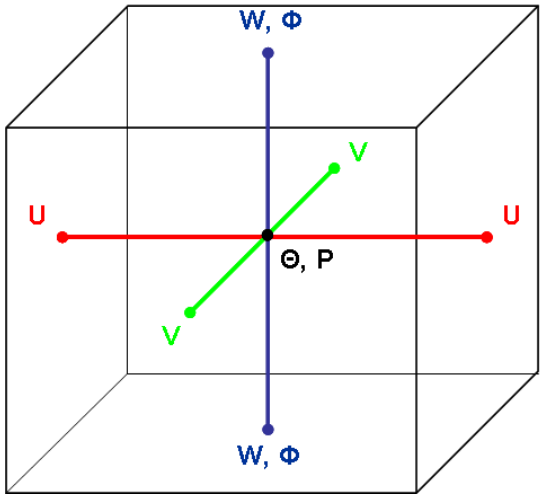

Figure 3.5. The Arakawa C staggered grid used in WRF.

given in the following sections, and in Chapter 4 they are applied to the case of a neutral boundary layer.

3.2.1 No Slip Case

The first step in all three of the methods is determination of the cells that are cut by the immersed boundary. WRF uses an Arakawa C staggered grid shown in Figure 3.5, which means that cut cells must be determined for each variable that will have a boundary condition imposed. Velocities are on the cell faces, and pressure and potential temperature are defined at the cell center. Geopotential ($\phi = gz$) is collocated with the w velocity, and vorticity points are centered on the cell edges. Currently in WRF, terrain height is a two-dimensional array that is defined at the locations where the cell centers would be vertically projected down to the surface. The first step is then to average the terrain height horizontally to the locations of the other variables (in the no slip case these are u and v). If the resolution of the terrain data is high enough, it is possible to assign a terrain height at every computational location in lieu of averaging. Before WRF begins iterating, each node for each variable (u, v, w) is marked as interior or exterior to the terrain, and the cut cells are determined. The nodes that are just interior to the terrain are assigned to be ghost

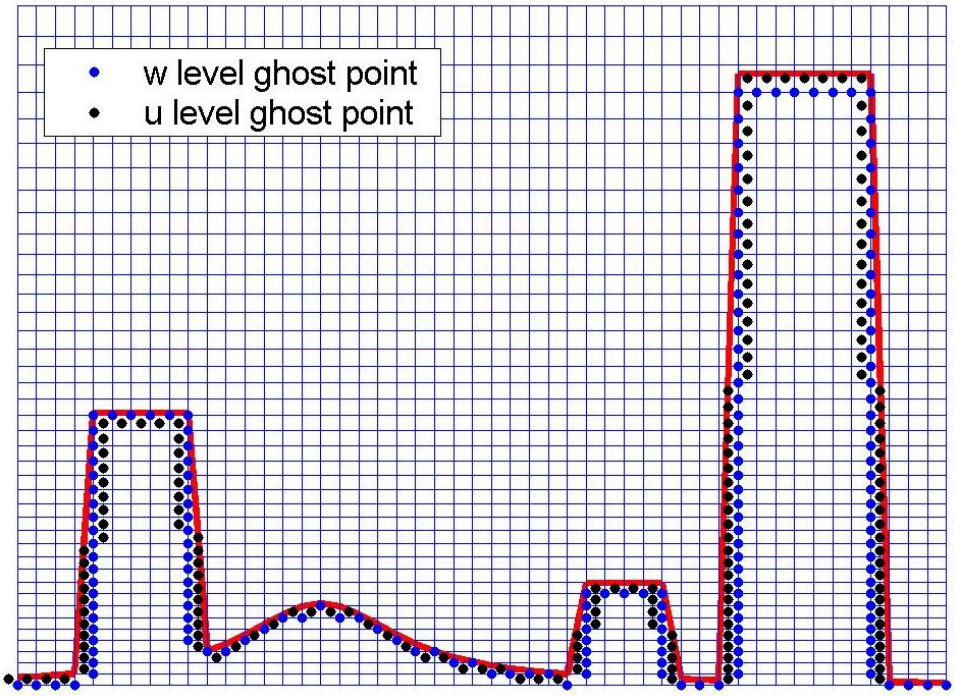

Figure 3.6. IBM is used to represent urban geometry with steep slopes. In this two-dimensional case, the u and w ghost points are shown.

points. Figure 3.6 was created using ghost point data from the WRF IBM module, and illustrates the location of u and w ghost points for a given complex terrain.

After the ghost points are determined, the velocities at all of the interior nodes are set to zero. Then the no slip boundary condition given by equation (3.5) is imposed by setting the velocities on the ghost points.

$$u_{surf} = v_{surf} = w_{surf} = 0 \tag{3.5}$$

For the neutral boundary layer case the surface is aligned with the coordinates, and one dimensional linear interpolation as proposed by Iaccarino and Verzicco [2003] and pictured in Figure 3.7 is used with accuracy. Of course, for curvilinear terrain a more sophisticated interpolation scheme should be implemented. During all of these pre-processing steps, the domain has been decomposed for parallel processing. Therefore, it is necessary to add additional halo exchanges and boundary condition updates just before the IBM pre-processing routines.

Next, WRF begins iterating over the large time step with a three step Runge-Kutta

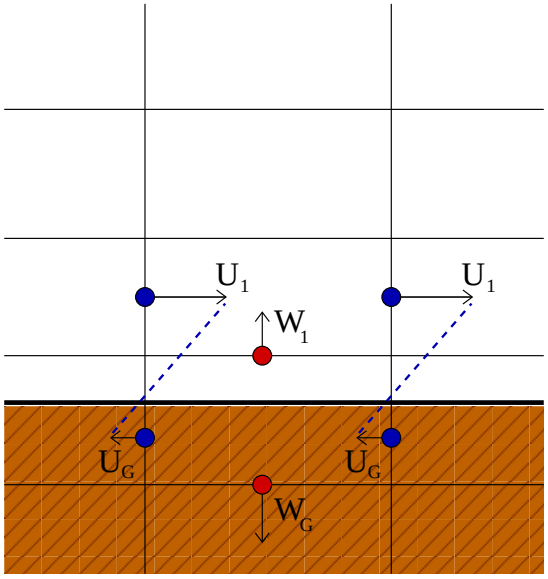

Figure 3.7. No Slip IBM.

method. An outline of the iteration process was previously presented in Table 2.1, and it may be useful to refer to the table for this discussion. During run time, the IBM boundary condition is set on every Runge-Kutta step just after the acoustic time step, and before the advancement of scalars. In Table 2.1 this is between steps seven and eight. At the end of the acoustic loop, the velocities as well as the geopotential are advanced. The first step is to complete halo exchanges and boundary condition updates. Next, the ghost points for u, v, and w are recalculated using the updated geopotential value. It should be noted that the vertical mass coordinate used in WRF is unsteady in time by definition. This means that unlike the other IB implementations referenced above, the ghost point locations must be updated periodically. Updating the ghost points on the Runge-Kutta loop provides a 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. Once the ghost point update is complete, linear interpolation is used to determine a value for the velocity at the ghost point which will enforce a no slip boundary condition, and the new velocity is imposed. Next, WRF advances the scalar equation and diagnoses the new pressure and density values. The same sequence of events repeats for the second and third step of the Runge-Kutta scheme.

3.2.2 Log-Law Velocity Reconstruction

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, this means the thin viscous sublayer should be resolved. This is of course an overwhelming computational requirement, and various techniques have been used to overcome this difficulty. If the boundary is stationary, the mesh is often locally refined near the surface. In the case of moving boundaries, adaptive mesh refinement can be coupled with direct forcing IBM. A third approach commonly used in atmospheric applications is wall or surface modeling, where a coarse grid is used and the stresses at the surface are modeled. Senocak et al. [2004] notes that applying a no slip boundary condition to a surface with coarse resolution will result in an overestimate of the implied stress, and suggests a surface model be used with IBM for topographic modeling in atmospheric flows.

In Fadlun et al. [2000] the immersed boundary is modeled by reconstructing the velocities on 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, 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 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 (3.6), 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.

$$\frac{U}{u_\star} = \frac{1}{\kappa} \ln\left(\frac{z}{z_o}\right) \tag{3.6}$$

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 (3.7) 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.

$$U_1 = U_2 \frac{\ln(z_1/z_o)}{\ln(z_2/z_o)} \tag{3.7}$$

Equation (3.7) 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 (3.8) at the second and third fluid nodes above the boundary.

$$\theta_2 = \arctan\left(\frac{v_2}{u_2}\right), \qquad \theta_3 = \arctan\left(\frac{v_3}{u_3}\right) \tag{3.8}$$

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 is used to determine the direction of fluid flow θ_1 on the cut cell. Direct forcing can then be imposed at the cut cell using the relationship given in (3.9).

$$u_1 = U_1 \cos\theta_1, \qquad v_1 = U_1 \sin\theta_1 \tag{3.9}$$

The vertical velocity at the surface is zero, and direct forcing is applied for w_1 at the cut cell using linear interpolation. 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. The velocity is set to zero on nodes that are interior to the terrain.

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 (3.7) requires that a roughness length be specified instead of a drag coefficient, so z_o was added to WRF for the IBM formulation. Next, the same pre-processor as previously described in Section 3.2.1 is used to find cut cells. The velocity nodes that are just exterior to the terrain are saved as cut cells. Equation (3.6) 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 (3.7), (3.8), and (3.9) the u

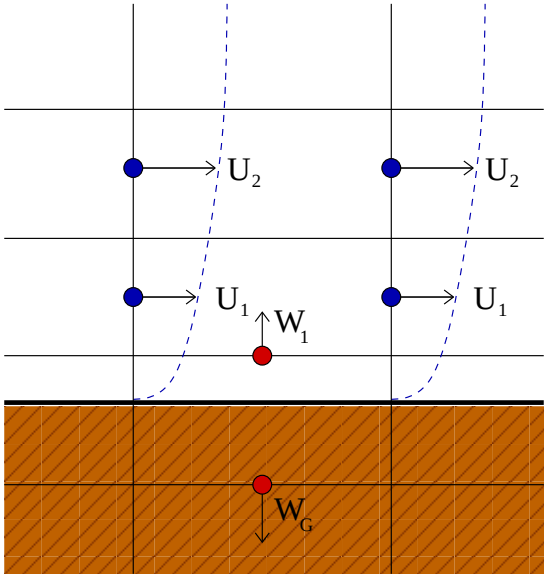

Figure 3.8. Velocity Reconstruction IBM.

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 surface using linear interpolation. A slight deviation from the Senocak et al. [2004] paper was 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.

WRF then begins iterating. 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 3.8 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.

3.2.3 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 (3.10a) and (3.10b), where (3.10a) is the kinematic boundary condition and (3.10b) sets the surface stress τ at the immersed boundary. In these equations μ is the column mass per unit area.

$$\mathbf{U} \cdot \hat{\mathbf{n}} = 0 \tag{3.10a}$$

$$\tau_{w_{xz}} = -\mu \left(\frac{\kappa}{\ln \frac{z_1 - h}{z_o}} \right)^2 |\mathbf{U}| u, \qquad \tau_{w_{yz}} = -\mu \left(\frac{\kappa}{\ln \frac{z_1 - h}{z_o}} \right)^2 |\mathbf{U}| v \tag{3.10b}$$

The shear stress reconstruction IB method is illustrated in Figure 3.9. 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 (3.10b), 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 are also needed for setting the shear stresses, but these are not found 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 2.1 this is step one. If the shear stress reconstruction method is being used, then additional IBM routines are called at this point.

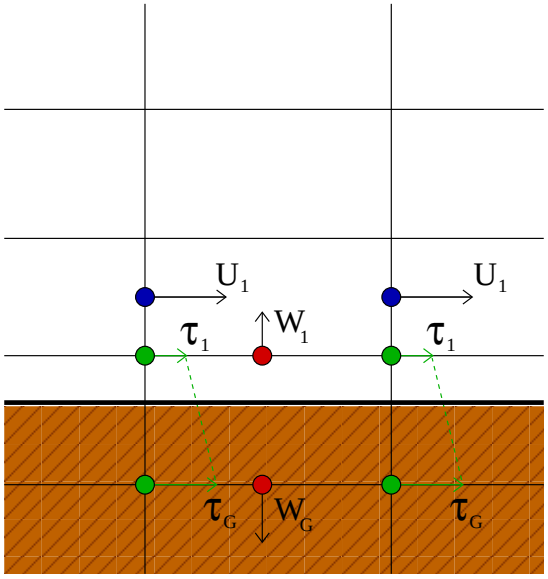

Figure 3.9. Shear Stress Reconstruction IBM.

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 (3.10b). 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.

Chapter 4

Neutral Atmospheric Boundary Layer Simulations

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 will 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. [2005, 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. Finally conclusions are drawn with suggestions for future work.

4.1 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, however, 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 $\theta = 288^{\circ}$ 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 $\sim$17.5 hours for the prescribed Coriolis force in this model. Figure 4.1 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 4.2 shows the same information on a hodograph. Results for the TKE turbulence closure are presented in Figures 4.3 and 4.4.

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 this exception, the inertial oscillations appear to be sufficiently damped after two periods or $\sim$35 hours. For this reason, the results in Section 4.2 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 $20 < tf < 30$.

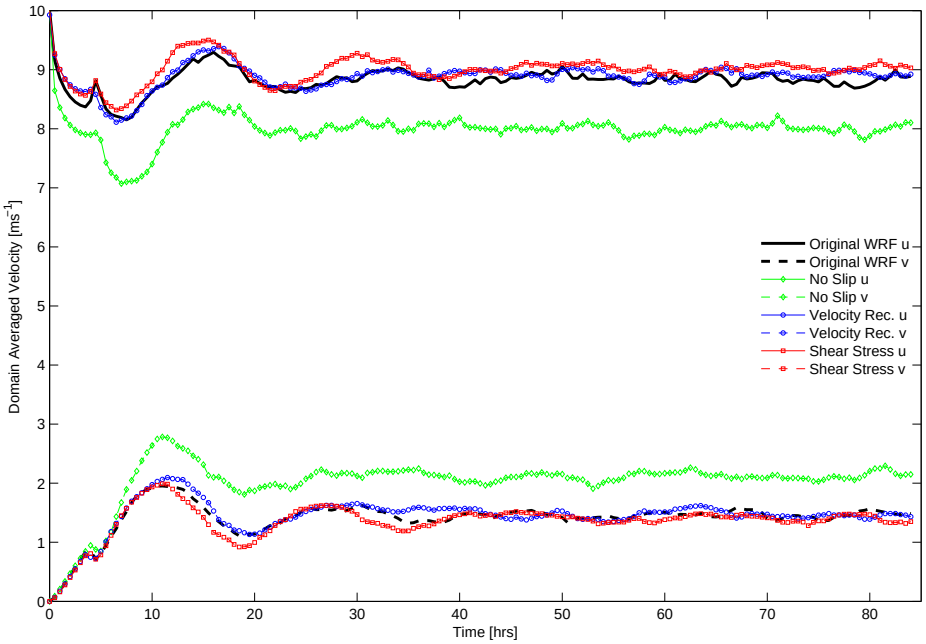

Figure 4.1. Time evolution of domain averaged u and v velocities showing the damping of inertial oscillations with the Smagorinsky closure.

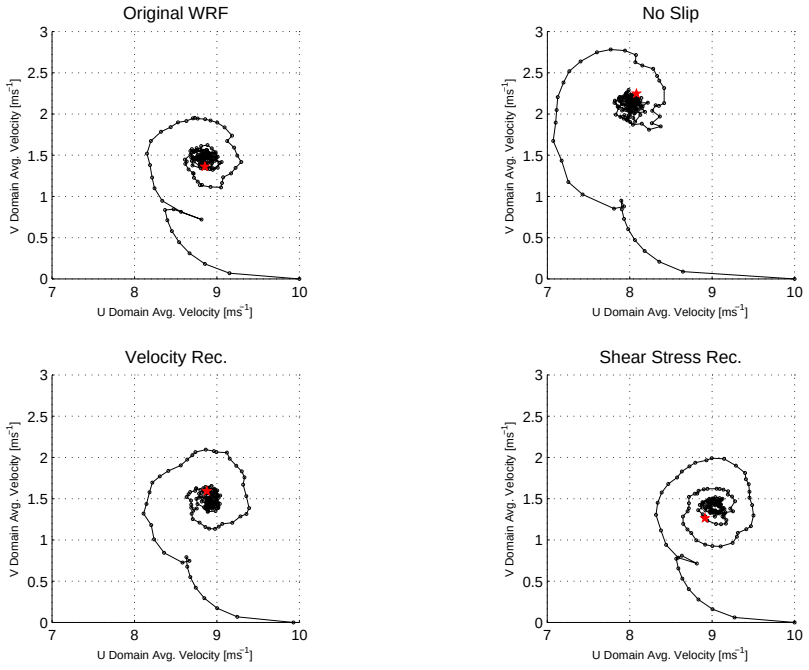

Figure 4.2. Domain averaged u and v velocity on a time series hodograph showing the damping of inertial oscillations with the Smagorinsky closure. The red star marks the time at 36 hours, which is after $\sim$2 periods.

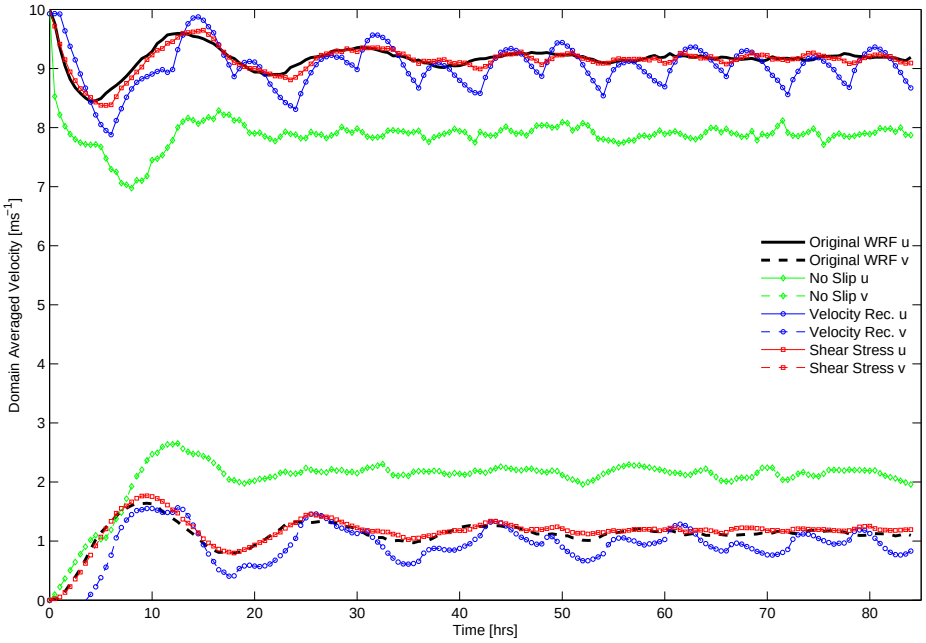

Figure 4.3. Time evolution of domain averaged u and v velocities showing the damping of inertial oscillations with the 1.5 order TKE closure.

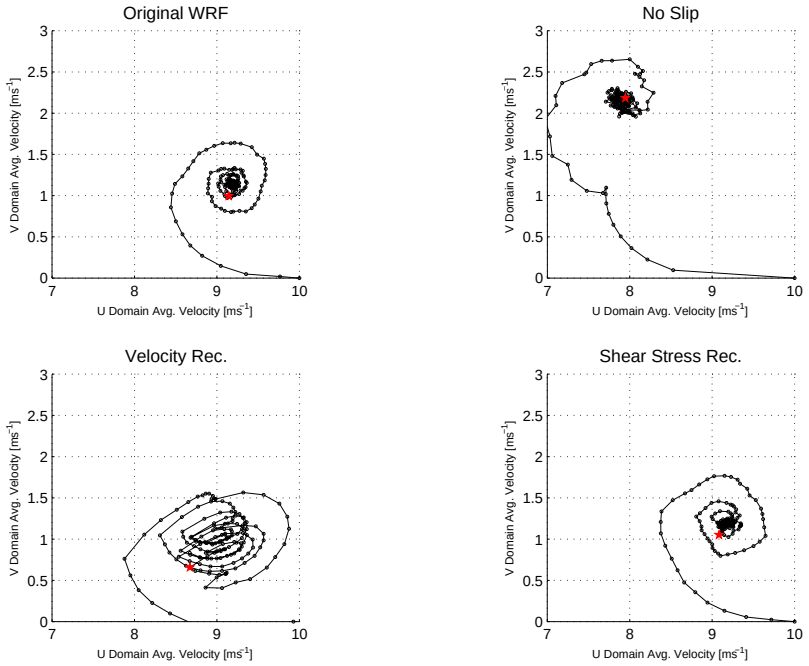

Figure 4.4. Domain averaged u and v velocity on a time series hodograph showing the damping of inertial oscillations with the 1.5 order TKE closure. The red star marks the time at 36 hours, which is after $\sim$2 periods.

4.2 Neutral Atmospheric Boundary Layer Results

Figures 4.5 and 4.7 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 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 4.6 with the Smagorinsky closure and Figure 4.6 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 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 (4.1a) and (4.1b):

$$u = U_g \left[1 - \exp(-a_o z)\cos(a_o z)\right] \tag{4.1a}$$

$$v = U_g \exp(-a_o z)\sin(a_o z) \tag{4.1b}$$

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 4.9 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 4.10 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 possible 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.

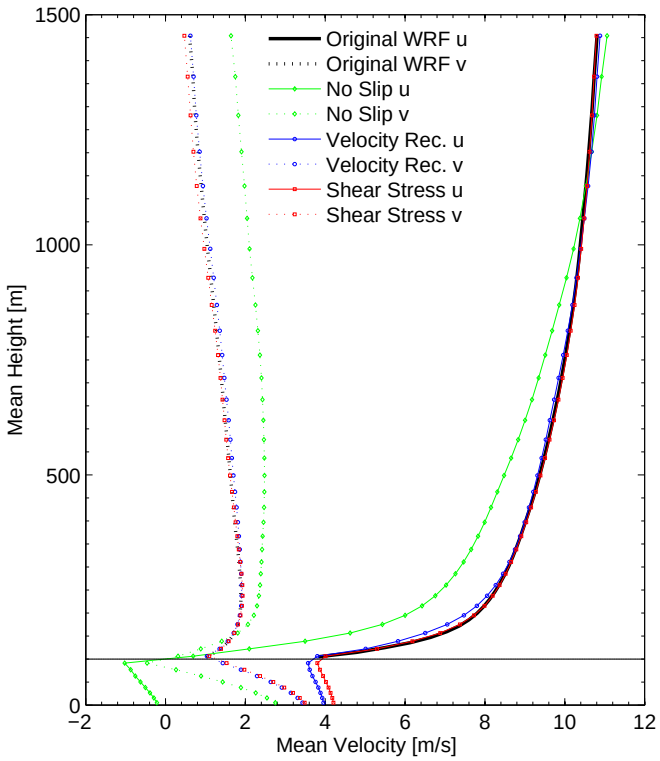

Figure 4.5. Mean U and V velocity with the three IBM implementations and the Smagorinsky closure. The flat plate is at 100 meters.

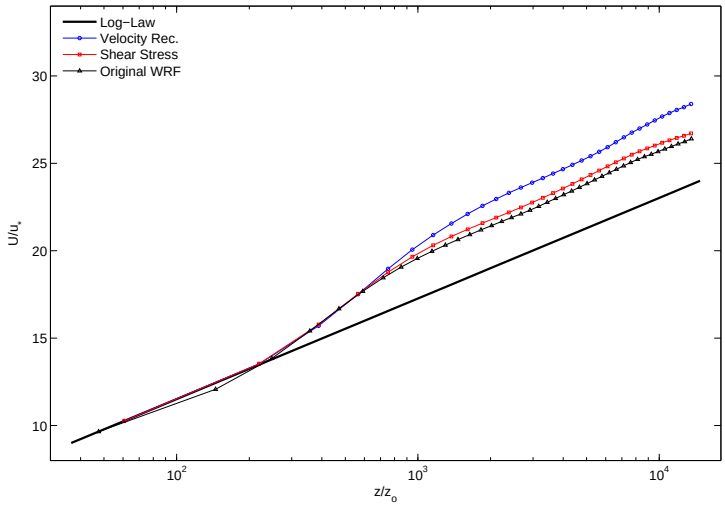

Figure 4.6. Non-dimensionalized mean velocity on a semi-log plot with the Smagorinsky closure.

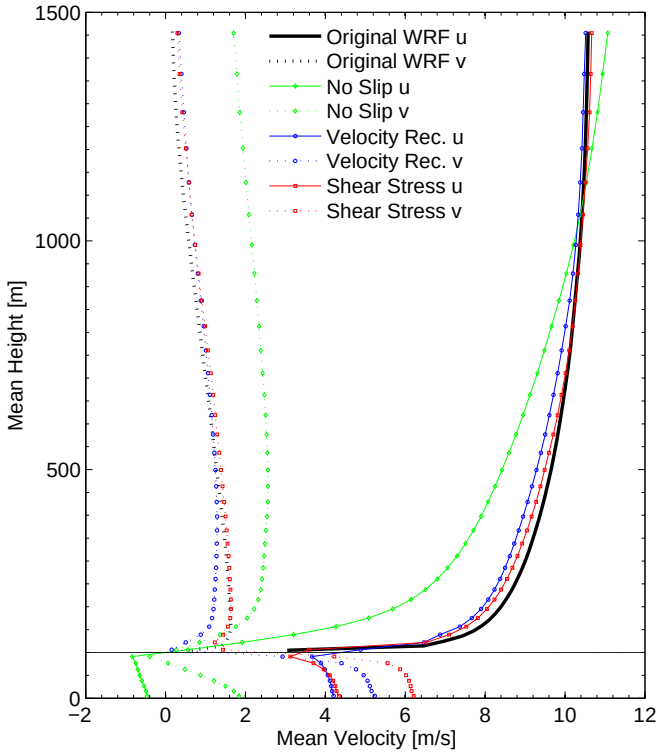

Figure 4.7. Mean U and V velocity with the three IBM implementations and the 1.5 order TKE closure. The flat plate is at 100 meters.

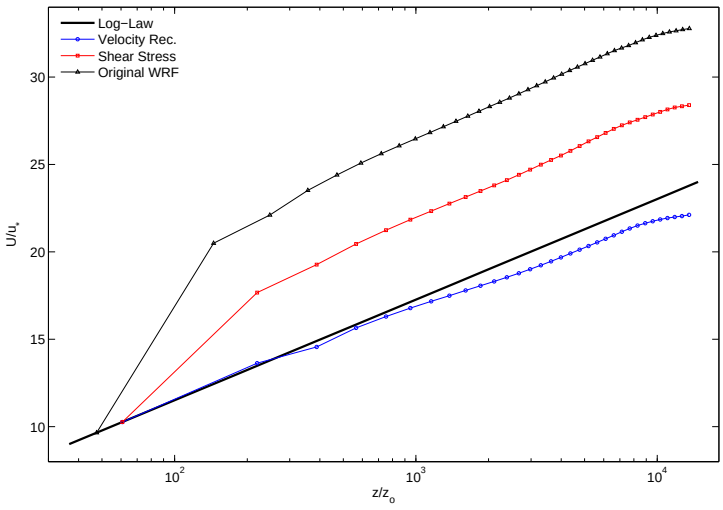

Figure 4.8. Non-dimensionalized mean velocity on a semi-log plot with the 1.5 order TKE closure.

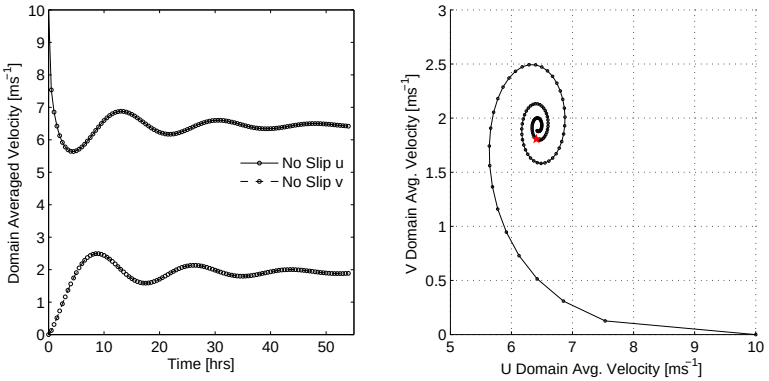

Figure 4.9. (a) Time evolution of domain averaged u and v velocities showing the damping of inertial oscillations for the no slip boundary condition with constant eddy viscosity. (b) Domain averaged u and v velocity on a hodograph showing the damping of inertial oscillations with a red star marking the time at 36 hours, which is after $\sim$2 periods.

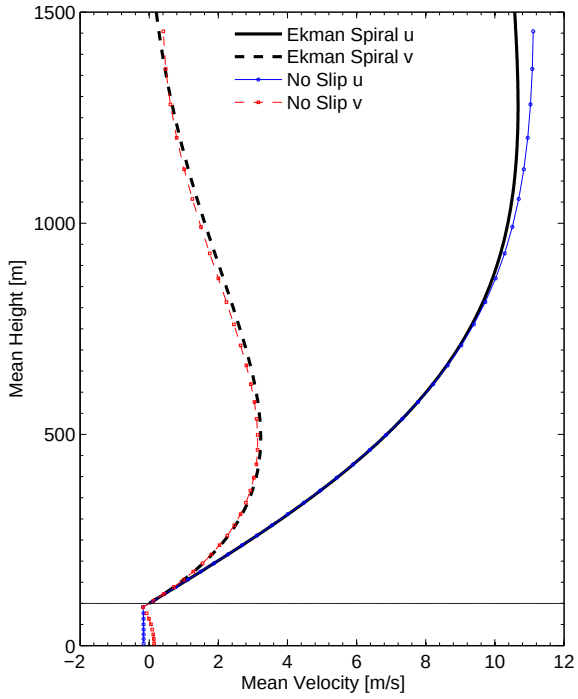

Figure 4.10. Ekman spiral and mean U and V velocity for the no slip IBM implementation with constant eddy viscosity. The flat plate is at 100 meters.

4.3 Conclusions

Most of the IBM research to date has simulated a no slip boundary condition. This is adequate because the applications have been at low or moderate Reynolds numbers where the flow domain was highly resolved. Atmospheric simulations model 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. It is possible that 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 next step in testing these methods would be to extend them to complex terrain and model the canonical cases of two-dimensional flow over an isolated hill and three-dimensional flow over a Gaussian hill.

It was seen that IBM is a viable option in NWP, and specifically WRF, for removing the barriers to modeling complex geometries created by the use of terrain following coordinates. Additionally, surface models combined with IBM show improved performance over the traditional no slip IBM boundary condition in their ability to recreate the solution given by the original WRF boundary conditions 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.

References

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.

Bjerknes, V., and Coll., *Dynamic Meterology 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.

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.

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.

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

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.

Iaccarino, G., and R. Verzicco, Immersed boundary technique for turbulent flow simulations, *Appl. Mech. Rev.*, *56*(3), 331–347, 2003.

Kang, S., G. Iaccarino, and P. Moin, Accurate and efficient immersed-boundary interpolations for viscous flows, , Center for Turbulence Research, NASA Ames/Stanford Univ., Palo Alto, CA, 2004.

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

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

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

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.

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.

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.

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.

Skamarock, W., J. Klemp, J. Dudia, 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, 2005.

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

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