Numerical Simulation of the Atmospheric Boundary Layer over Complex Topography: A Modern Approach to a Classical Problem

A THESIS
SUBMITTED TO THE FACULTY OF THE
UNIVERSITY OF MINNESOTA
BY:

Noah Lial Andersen

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE
IN MECHANICAL ENGINEERING

Lian Shen

Advisor

May 2020

Acknowledgements

I would like to extend my greatest gratitude to my advisor, Professor Lian Shen, who has inspired my study of computational fluid dynamics. He has provided excellent guidance throughout my studies and has afforded me the opportunity to work on many exciting projects. His passion for mentoring, scientific exploration, and hard work will have a lasting impact on me far beyond my education at the University of Minnesota.

Furthermore, I would like to thank my committee members, Professor Michele Guala and Professor Jiarong Hong, for taking the time out of their busy schedules to assist me.

I would also like to thank the entire Fluid Mechanics Lab for assisting me in my research and course work. Being able to work alongside all of these bright minds encouraged me to continue striving to learn as much as possible and challenge the status quo.

Lastly, I would like to thank my family and friends who have been gracious enough to support me through this entire process. Their words of encouragement, support, and genuine interest in my research helped me nearly every day of my graduate studies.

Abstract

Numerical methods were developed and validated to simulate the atmospheric boundary layer (ABL) using large eddy simulation (LES). This framework captures the topography of the Earth's surface rather than modeling it. To robustly simulate the ABL, four unique capabilities (temperature transport, topographic data, immersed boundary method with wall modeling, and turbulent inflow generation) were added to a traditional finite difference computational fluid dynamics code. The accuracy of each capability was analyzed individually using validation tests. Then, a full scale simulation of the ABL over a tidal inlet was conducted. It was found that the resolved topography of the Earth's surface had a significant effect on the flow field. Furthermore, it was found that the results from LES are more accurate than mesoscale simulations. Lastly, it was found that the errors in the present simulation are a result of the roughness model used over the sea surface.

Contents

List of Figures

List of Abbreviations

ABL	**A**tmospheric **B**oundary **L**ayer
CBL	**C**onvective **B**oundary **L**ayer
CFD	**C**omputational **F**luid **D**ynamics
CFL	**C**ourant-**F**riedrichs-**L**ewy
COAMPS	**C**oupled **O**cean/**A**tmosphere **M**esoscale **P**rediction **S**ystems
DNS	**D**irect **N**umerical **S**imulation
IB	**I**mmersed **B**oundary
LES	**L**arge **E**ddy **S**imulation
M-O	**M**onin-**O**bukhoff
NOAA	**N**ational **O**ceanic and **A**tmospheric **A**dministration
QUICK	**Q**uadratic **U**pstream **I**nterpolation for **C**onvective **K**inematics
RK2	**2**nd order **R**unge-**K**utta
SBL	**S**tably Stratified **B**oundary **L**ayer
SGS	**S**ub-**G**rid **S**cale
STL	**ST**ereo**L**ithography
STS	**S**ub-**T**est **S**cale
TVD	**T**otal **V**ariation **D**iminishing
UTM	**U**niversal **T**ransverse **M**ercator
WM-LES	**W**all-**M**odeled **L**arge **E**ddy **S**imulation
WR-LES	**W**all-**R**esolved **L**arge **E**ddy **S**imulation

Chapter 1

Introduction

1.1 Motivation

In 1687, Isaac Newton revolutionized the study of the natural world with his publication of *Philosophiæ Naturalis Principia Mathematica* in which he established the basis for classical mechanics. The laws outlined by Newton have been studied, adapted, and utilized by countless mathematicians, scientists, and engineers for over 300 years. These principles have been used to predict the motion of galaxies, design skyscrapers, understand the molecular structures of materials, and even send man to the moon. However, despite the centuries spent studying and utilizing classical mechanics, there still remains one essential unsolved problem: turbulence.

Characterized by the chaotic motion of fluid particles, turbulence is a phenomenon that's been experienced by every individual living in the natural world. From Leonardo da Vinci sketching the random chaos in turbulent water (Figure 1.1), to airline passengers steadying their drinks as a plane passes through "unexpected rough air", everyone has an understanding of what turbulence is. Yet despite this shared experience, no one has determined a method to completely quantify and predict the behavior of turbulence. For this reason, countless studies have been conducted

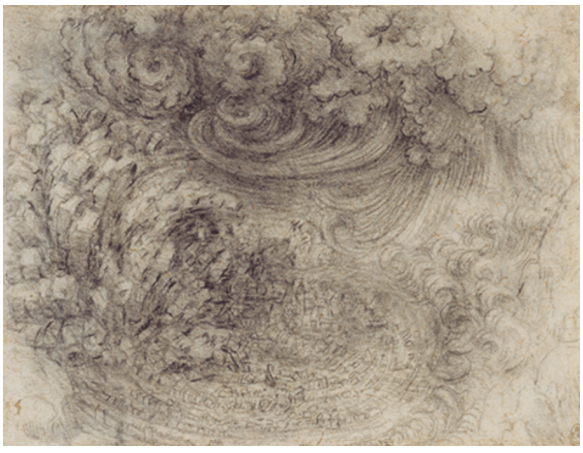


FIGURE 1.1: Da Vinci, Leonardo. *Sketch*. Windsor, Royal Library, 12378

to locate the elusive, all encompassing theory of turbulence. The hunt for this theory is best described by Werner Heisenberg, a pioneer in the field of quantum mechanics:

When I meet God, I am going to ask him two questions:

Why relativity? And why turbulence?

One of the greatest challenges caused by turbulence is that it is prevalent in countless fields of research and design and thus can not be neglected. One such field can be best seen in Vincent van Gogh's famous 19th century painting, *The Starry Night*. From his characteristic swirl in the center of the landscape to his chaotic brushstrokes, Van Gogh is able to highlight the prevalence of turbulence in the wind of the lower atmosphere. Furthermore, this composition highlights how the wind flow and Earth's surface form a coupled system. The hills in the background cause a change in the wind and likewise, the wind causes a change in the structure of the cypress tree in the foreground. The qualitative observations of Van Gogh's turbulent system provide a basis for numerous studies of the lowest layer of the atmosphere. This layer, called the atmospheric boundary layer (ABL), is still being extensively studied more than a century after Van Gogh first crafted his famous composition.

FIGURE 1.2: van Gogh, Vincent. *The Starry Night*. 1889, Museum of Modern Art, New York.

The turbulence in the ABL is a result of the friction caused by the Earth's surface combined with the low viscosity of air and large length scales. Additionally, the effect of energy transfer in the ABL due to the heating and cooling of the Earth's surface further complicates these processes. Due to the complexity of the ABL, this system has been extensively researched throughout the 20th and 21st centuries. However, since there is no unified theory of turbulence, researchers must rely on field studies and simulations in lieu of purely

analytical based studies. These studies are important because the ABL is the layer of the atmosphere in which humans reside. These studies can be used to improve local engineering systems (e.g. wind farms, wind reduction, plant fertilization) as well as global systems (e.g. global weather prediction, large-scale pollution transport).

Simulation of fluid systems using computational fluid dynamics (CFD) provides an extremely useful tool to study the ABL. Simulation based studies allow for the ABL to be extensively analyzed because they generate three dimensional snapshots of the velocity, pressure, and temperature of the flow field at any moment in time. This is an improvement over field based studies which may only provide measurements at limited locations. The difficulty of conducting a simulation based studies is that extreme care must be taken to ensure the results generated by the simulation accurately represent the physical system that they are trying to recreate. Using simulations to study the ABL requires various modifications to standard CFD packages in order to ensure the generated results are accurate and robust for various conditions. Simulation based studies of the ABL must be capable of capturing the effect of buoyancy caused by temperature differences, complicated surface topography, and highly turbulent flow fields.

1.2 Thesis Outline

In this thesis, the development, testing, and validation of a CFD code capable of simulating the ABL will be documented.

Chapter 2 will discuss the background information for the thesis. Section 2.1 will present relevant background information about the ABL. This includes a basic description of the physics and mechanisms of importance within the ABL. Then a historical overview of the numerical based studies used to analyze the ABL will be presented. This includes a discussion of studies using large eddy simulations (LES) and mesoscale simulations. Finally, the Monin-Obukhoff (M-O) similarity theory which is used to model the ABL will be presented. Section 2.2 will discuss the equations of motion for incompressible, Newtonian

fluids. The equations will then be non-dimensionalized based off characteristic values. Lastly, the governing equations will be filtered in order to obtain the governing equations for LES. In Section 2.3, an in depth exploration of an incompressible CFD code will be presented. This will include discussion of meshing, spatial and temporal discretizations, sub-grid scale models, immersed boundary (IB) methods, level-set functions, wall models, and turbulent boundary conditions. This section will consider all of the modules of the CFD code that were developed prior to the investigation documented in this thesis.

Chapter 3 will discuss the numerical methods that were implemented in this investigation to modify the CFD framework to accompany simulation of the ABL. These methods include the temperature transport equation, implementation of topographic data, wall models for IB method, and inflow boundary condition. Each section will conform to the following structure. First, the numerical methods will be described, then a test case will be presented, and finally the results will be analyzed.

Chapter 4 will discuss the full-scale test of turbulent flow over a tidal inlet. First, Section 4.1 will present the case setup. This includes a discussion of the computational domain, parameters of the governing equations, and boundary conditions. This will also include the methodology of how the simulation was connected to real-world measurement data. Section 4.2 will then present the results of the simulation to identify the effect of the complex topography, assess the accuracy of the simulation results, and identify areas for improvement.

Chapter 5 will provide conclusions for the present study and recommendations for future studies.

Chapter 2

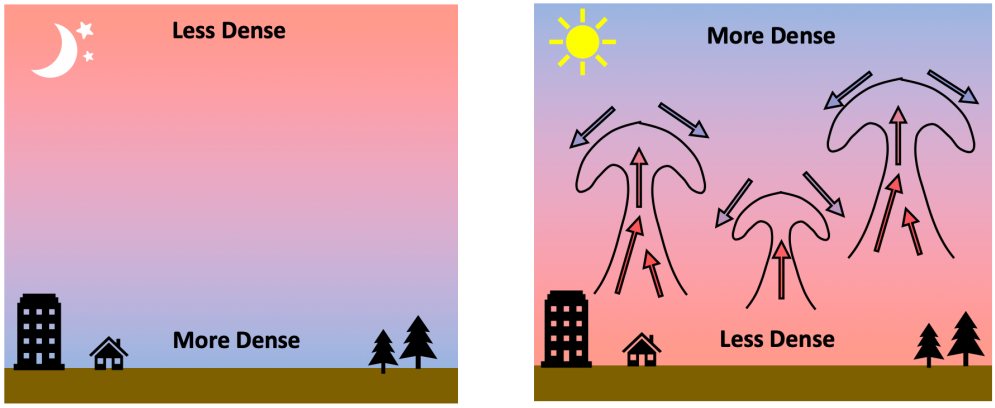
Background

2.1 Atmospheric Boundary Layer

The atmospheric boundary layer (ABL) is defined as the lowest layer of the atmosphere. This layer can vary in height depending on the time of day and location on the globe, but it typically ranges in height from a few hundred meters to a few kilometers above the Earth's surface. Studying the ABL is essential for many different applications of science and engineering. Understanding the ABL is critical for predicting the transport of emissions which can lead to better methods of mitigating air pollution. An improved understanding of the ABL can lead to better design of wind farms since designers will be able to understand where to position wind turbines to maximize energy extraction. Additionally, many weather prediction tools use models that have been developed and continue to be modified by studying the ABL.

The ABL is a dynamic system which contains numerous complicated phenomena such as diurnal cycles caused by the day-night cycle, strong turbulence caused by the aerodynamic drag of the Earth's surface, and complicated terrain caused by cities, vegetation, mountains, and valleys. All of these factors need to be considered when studying the ABL.

The time of day, which significantly affects the temperature gradient in the ABL, can affect the strength of the turbulence induced mixing. During the day, when the surface of the Earth is warmer than the air, the ABL will be unstably stratified (dense air on top of lighter air). This will result in chaotic convective plumes rising through the atmosphere that enhance the turbulence. For this reason, this type of boundary layer is called a convective boundary layer (CBL) (See Figure 2.1b). By contrast, at night the Earth will be cooler (thus



(a) (b)

FIGURE 2.1: Comparison of two separate ABL. (a) Stably stratified boundary layer with no buoyancy induced motions present. (b) Convective boundary layer with plumes caused by buoyancy.

more dense) than the air above, resulting in a more stable system. This is called a stably stratified boundary layer (SBL) (See Figure 2.1a). In a SBL, the density differences will not induce any turbulent mixing. This is why it is typically more windy during the day and early evening, and calmer during the evening and early morning (Stull, 1988).

The characteristics of the ABL also depend greatly on the topography (mountains, valleys, cliffs, etc.) and roughness (vegetation, buildings, water waves, etc.) of the surface. One parameter commonly used in ABL studies is the roughness length y_o. The roughness length is a parameter used when fitting a mean profile to the flow in the ABL. Larger roughness lengths indicate that the effects of the surface travel higher into ABL. Through experimental studies it has been found that the roughness layer is affected by both the roughness and the topography of the surface (Markfort, 2013); (Maeda, Homma, and Ito, 2004).

2.1.1 Numerical Based Studies

During the late 1960s, researchers began using LES, a powerful numerical tool capable of resolving many of the complicated dynamics of the ABL. The specific methodologies of LES will be discussed further in Section 2.2.3 and Section 2.3.3. These simulations had grid resolutions that were fine enough to resolve some of the larger eddies prevalent in the

ABL. Additionally, LES allowed for researchers to obtain data which was high resolution, three-dimensional, and temporal. This improved understandings of the dynamics of the ABL. The pioneers of LES based studies of the ABL primarily studied the CBL over a flat surface with periodic boundary conditions (Deardorff, 1974); (Sommeria, 1976). As the computational power increased over the next decades, researchers were able to perform more complicated simulations to expand their studies to the SBL (Mason and Derbyshire, 1990) and simulations over heterogeneous land (Bou-Zeid, 2005).

LES is an important tool used to study local effects on the ABL, but the size of many LES studies is currently limited by computational cost. Even with massive computational power, LES are incapable of simulating domain sizes on a global scale. Since many of the phenomena of interest in the ABL, such as pollution transport and weather prediction, are on scales much larger than is possible by LES, it is unlikely that LES will be the sole tool used to study these phenomena in the foreseeable future. However, LES can be extremely helpful in improving the accuracy and robustness of the larger scale simulations.

Mesoscale Simulations

Mesoscale meteorology includes scales which span the domain sizes from 10 to 1000 kilometers. Being able to accurately simulate weather patterns on the mesoscale level allows for weather prediction of smaller scale systems such as thunderstorms, weather fronts, and ocean breezes. Typically mesoscale simulations will have grid sizes on the order of 1 kilometer, which is much greater than the scales of the turbulence in the ABL. For this reason, the mesoscale simulations use models to capture the effects of the turbulence. The M-O similarity theory is used to address this problem. The M-O similarity theory was developed by using non-dimensional variables and the Buckingham-Pi theorem to describe the non-dimensional mean flow and temperature in the ABL as a function of a non-dimensional Obukhov length scale (Monin and Obukhov, 1954). The Obukhov length

scale is defined as

$$L_{mo} = -\frac{v_*^3\ \theta\ \rho\ c_p}{\kappa\ g\ q} \tag{2.1}$$

where v_* is the characteristic velocity, κ is the von Karman constant, θ is the temperature, q is the turbulent heat flux, ρ is the density, and c_p is the isobaric heat capacity. The largest problem with using the M-O similarity theory for mesoscale simulations is that the results are only accurate for homogeneous, flat topography. When the topography is not homogeneous (which is very common on the Earth's surface), the M-O assumption breaks down. For instance, one study of a coastal, heterogeneous environment found that M-O similarity theory overpredicted the surface drag coefficient by 2.6 times (Ortiz-Suslow et al., 2015). In order to improve the capabilities of these mesoscale simulations, new parameterizations to capture the effects of complicated topography must be developed. Using LES to study the ABL, particularly in regions of complicated topography, will lead to a better understanding of the ABL through improved mesoscale simulation capabilities.

2.2 Governing Equations

2.2.1 Incompressible Navier-Stokes

In order to simulate the flow field of the ABL, the conservation laws for mass (2.2), momentum (2.3), and energy (2.4) must be numerically solved. Einstein's notation is used in the following equations. The subscript 'i' in (2.3) indicates the component of the velocity being solved for, where $i = 2$ is the direction aligning with gravity.

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{2.2}$$

$$\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu\frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{g}{\theta_o}(\theta - \theta_o)\delta_{2i} \tag{2.3}$$

$$\frac{\partial \theta}{\partial t} + u_j\frac{\partial \theta}{\partial x_j} = \alpha\frac{\partial^2 \theta}{\partial x_j \partial x_j} \tag{2.4}$$

In these equations, u_i is the velocity, ν is the kinematic viscosity, g is the acceleration due to gravity, ρ is the density, θ_o is a reference temperature, and α is the thermal diffusivity. These equations have been simplified based on the assumptions that the fluid (air) is Newtonian and incompressible. The incompressible assumption holds for air because the flow is at a low Mach number (Pope, 2000). However, the density of air does change with fluctuations in temperature. In order to account for the differences in air density as it relates to temperature difference, a Boussinesq approximation is implemented. This approximation assumes that the density differences in the flow have a significant effect on the specific gravity of the fluid, but minimal influence on the inertia of the fluid. For this reason, the effect of density difference only appears in the final term of Equation 2.3.

2.2.2 Non-dimensional Equations

In order to reduce the number of parameters in the governing equations, they are non-dimensionalized. The following non-dimensional values are used

$$u_i^* = \frac{u_i}{U}, \quad x_i^* = \frac{x_i}{L}, \quad t^* = \frac{tU}{L} \quad p^* = \frac{p}{\rho U^2}, \quad \theta^* = \frac{\theta}{\theta_o} \tag{2.5}$$

where values with * are non-dimensional, and U and L_y are the characteristic velocity and length scale, respectively. Using the reference values defined above, the following non-dimensional numbers are defined

$$Re = \frac{UL}{\nu} \quad \text{(Reynolds Number)} \tag{2.6}$$

$$Fr = \frac{U}{\sqrt{gL}} \quad \text{(Froude Number)} \tag{2.7}$$

$$Pr = \frac{\nu}{\alpha} \quad \text{(Prandtl Number)} \tag{2.8}$$

By substituting these non-dimensional values into (2.2-2.4), the non-dimensional forms of the governing equations can be obtained.

$$\frac{\partial u_j^*}{\partial x_j^*} = 0 \tag{2.9}$$

$$\frac{\partial u_i^*}{\partial t^*} + u_j^* \frac{\partial u_i^*}{\partial x_j^*} = -\frac{\partial p^*}{\partial x_i^*} + \frac{1}{Re} \frac{\partial^2 u_i^*}{\partial x_j^* \partial x_j^*} + \frac{1}{(Fr)^2}(\theta^* - 1) \tag{2.10}$$

$$\frac{\partial \theta^*}{\partial t^*} + u_j^* \frac{\partial \theta^*}{\partial x_j^*} = \frac{1}{Re \cdot Pr} \frac{\partial^2 \theta^*}{\partial x_j^* \partial x_j^*} \tag{2.11}$$

For the remainder of this thesis, the non-dimensional governing equations will be exclusively used, so the $*$ (which indicates a non-dimensional value) will be dropped from equations (2.9-2.11).

2.2.3 Filtered Equations

In order for a numerical solution of the governing equations to be accurate, all scales of the flow field must be resolved. For this reason, when defining a mesh for a numerical simulation, it is important that the mesh is refined enough to resolve the smallest scales of motion. Simulations that resolve all scales of the flow are called direct numerical simulations (DNS). Flows in the ABL have a large Reynolds number and are therefore highly turbulent. In highly turbulent flow fields a characteristic length scale η was defined by Andrey Kolmogorov. This length scale, now termed the Kolmogorov length scale, represents the size of the smallest energy containing eddies. This length scale depends on the viscosity ν of the fluid and the dissipation rate ϵ (Kolmogorov, 1941). In a field study performed in the ABL, it was found that the Kolmogorov length scale is approximately 0.7 mm (Meneveau and Sreenivasan, 1991). Therefore, in order for a numerical solution of the ABL to be accurate, it must have a grid scale on the order of $O(1\ mm)$. The total size of the ABL, by contrast, is on the order of $O(1\ km)$. Due to the large difference in magnitude from the large scale to the small scale, a simulation would need to have a number of grid cells on

the order of $O((1,000,000)^3)$, which would require an unrealistically massive amount of memory and computational cost by current standards.

To reduce the computational cost of running simulations of the ABL, the smallest scales of motion are ignored by applying a low-pass filter to the governing equations (2.2-2.4). This results in

$$\frac{\partial \tilde{u}_j}{\partial x_j} = 0 \tag{2.12}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i} + \nu \frac{\partial^2 \tilde{u}_i}{\partial x_j \partial x_j} - \frac{g}{\theta_o}(\tilde{\theta} - \theta_o)\delta_{2i} - \frac{\partial \tau_{ij}^{sgs}}{\partial x_j} \tag{2.13}$$

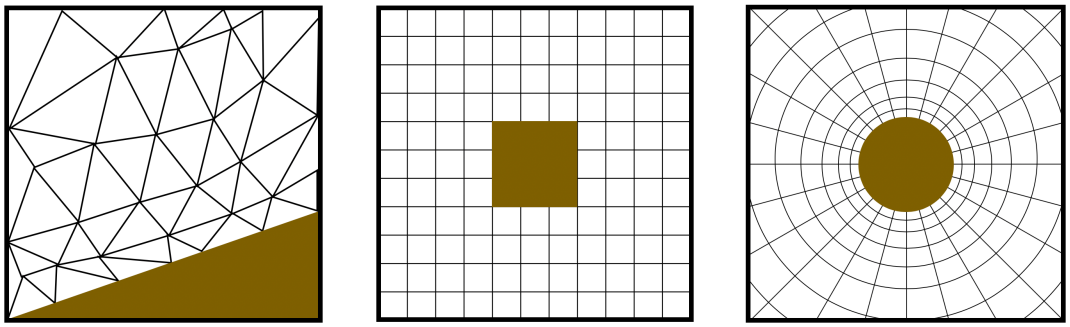
$$\frac{\partial \tilde{\theta}}{\partial t} + \tilde{u}_j \frac{\partial \tilde{\theta}}{\partial x_j} = \alpha \frac{\partial^2 \tilde{\theta}}{\partial x_j \partial x_j} - \frac{\partial \lambda_j^{sgs}}{\partial x_j} \tag{2.14}$$

where the tilde represents the filtered value of each variable, τ_{ij}^{sgs} is the sub-grid scale (SGS) stress tensor, and λ_j^{sgs} is the SGS heat flux. In order to close this model, the SGS stress tensor and heat flux must be calculated. Simulations that solve these filtered governing equations are called LES.

2.3 Numerical Simulations

2.3.1 Types of Mesh

There are two types of meshes which are commonly used in CFD simulations, unstructured and structured, each with inherit strengths and weaknesses. Unstructured meshes involve cells of varying geometry and sizes. Unstructured meshes are typically implemented when complicated geometries are involved, such as flow over a slope (Figure 2.2a). By contrast, structured meshes are characterized by a regular pattern of grid cells. Structured meshes can be in a variety of coordinate systems depending on the geometry of the simulation that is being performed. For instance, a Cartesian mesh can be used for rectangular geometries such as flow around a square (Figure 2.2b), while a polar mesh can be used for circular geometries such as flow around a circle (Figure 2.2c).



(a) (b) (c)

FIGURE 2.2: Various computational meshes. (a) Unstructured mesh for flow over slope. (b) Structured Cartesian mesh for flow over square. (c) Structured polar mesh for flow over circle.

Unstructured Meshes

There are several advantages and disadvantages to using an unstructured mesh for CFD simulations. The main benefit is that it is possible to create a mesh that aligns directly with the important boundaries in a simulation. This allows for the boundary conditions to be easily set. For instance, in Figure 2.2a the bottom set of cells all align with the fluid-solid interface, so it is simple to implement the no-slip boundary condition on this surface. There are also a number of disadvantages to using an unstructured mesh. In cases where a solid-fluid boundary is moving, the mesh will need to be regenerated every time step which drastically increases the computational cost. It is also more difficult to keep track of the locations of the grid cells because there is no structure to their layout. This can lead to difficulty when trying to parallelize simulations performed on an unstructured mesh.

Structured Meshes

Likewise, there are numerous advantages and disadvantages to using a structured mesh for CFD simulations. The first benefit to using a structured mesh is that the discretizations of gradients are significantly simpler due to the regular spacing of cells. Secondly, it is much simpler to develop and parallelize code for structured meshes, which results in more efficient, faster running simulations. The largest disadvantage to using a structured

mesh is that it is more difficult to represent complex geometries since the grid does not align with the structure. Since a majority of modern engineering problems involve complicated geometries, it is necessary to develop tools to allow for complicated geometries to be captured by structured meshes that do not align with the fluid-solid boundary (Section 2.3.4)

2.3.2 Numerical Discretizations

The discretizations used in these finite difference simulations are performed on a staggered, Cartesian mesh. With a staggered mesh, the velocity components are stored at the cell faces, while the scalars such as pressure and temperature are stored at the cell center. Figure 2.3 shows a two-dimensional representation of the staggered grid implemented in these simulations. Staggered grids are used in many incompressible flow solvers in order to avoid the checkerboarding instability that often arises when all variables (velocity components, pressure, temperature, etc.) are stored at the same location (Patankar, 1980).

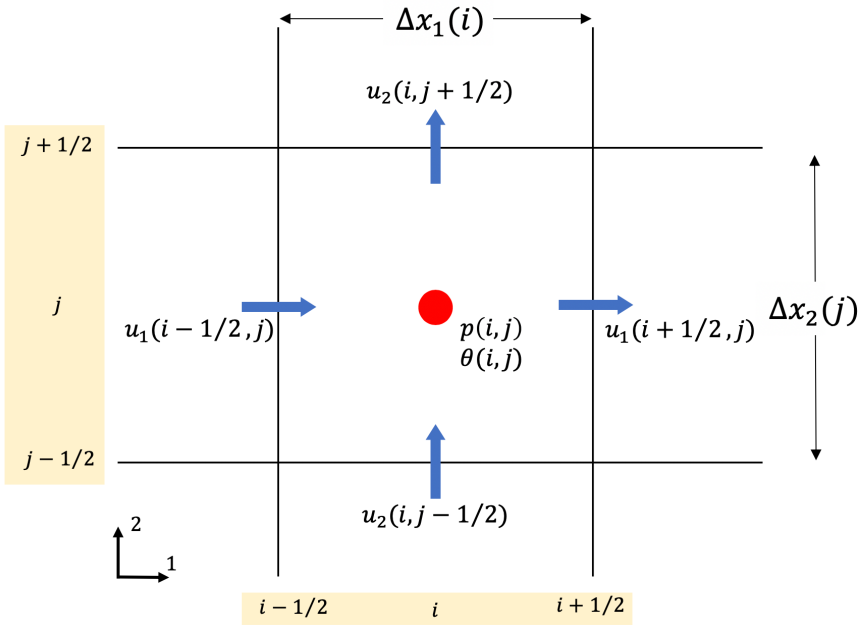


FIGURE 2.3: Two dimensional representation of staggered grid. Velocity components u_i are stored on faces of cells, while scalars such as pressure p and temperature θ are stored in cell center

Temporal Discretizations

The time advancement scheme implemented in this simulation is an explicit second order Runge-Kutta (RK2) method. Consider the following partial differential equation:

$$\frac{\partial u(x,t)}{\partial t} = G(u) \tag{2.15}$$

The first step of RK2 is to calculate a prediction value $u^{n+1/2}$ using a first order discretization:

$$u^{n+1/2}(x,t) = u^n + \Delta t \cdot G(u^n) + O(\Delta t) \tag{2.16}$$

Simply using this time advancement would result in a first order accurate time advancement. After the prediction step, the value at u^{n+1} is then calculated using both the initial value u^n and the prediction value $u^{n+1/2}$.

$$u^{n+1} = u^n + \frac{\Delta t}{2}\Big[G(u^n) + G(u^{n+1/2})\Big] + O(\Delta t^2) \tag{2.17}$$

The result of this methodology is a second order accurate time advancement scheme. For a more advanced description of how the RK2 method is used to solve the conservation of momentum equation, see Appendix A.

Spatial Discretizations

There are countless spatial discretizations that have been developed to calculate the gradients of the velocity, temperature, and pressure fields, each with inherit advantages and disadvantages. In this thesis, three separate discretizations were explored: central differencing, upwinding, and quadratic upstream interpolation for convective kinematics (QUICK).

Central differencing is one of the simplest discretizations. This method is implemented by using the points directly adjacent to the location where the gradient is desired. For

instance, in Figure 2.3, the velocity gradient at the cell center (i, j) would be calculated as

$$\left.\frac{\partial u_1}{\partial x_1}\right|_{i,j} = \frac{u_1(i+\frac{1}{2},j) - u_1(i-\frac{1}{2},j)}{\Delta x_1(i)} \tag{2.18}$$

The benefit of using central differencing is that there is no numerical dissipation. However, this can lead to instabilities particularly when the convective term is dominant (i.e. large Reynolds number) (Patankar, 1980).

Upwinding was developed as a means to resolve the stability issues with central differencing when calculating convective terms. The difference between upwinding and central differencing is that upwinding uses the characteristics of the flow field to determine the stencil points used. Using a first-order upwinding scheme to calculate the velocity gradient at the cell center (i, j) would be performed as

$$\left.\frac{\partial u_1}{\partial x_1}\right|_{i,j} = \begin{cases} \frac{u_1(i+\frac{1}{2},j)-u_1(i,j)}{\Delta x_1/2} & U < 0 \\ \frac{u_1(i,j)-u_1(i-\frac{1}{2},j)}{\Delta x_1/2} & U > 0 \end{cases} \tag{2.19}$$

The benefit of upwinding is that it removes the instabilities that arise when central differencing is used to calculate convective terms. However, this advantage is accompanied with an increase in numerical diffusion, which results in a smearing effect (Leonard, 1979).

The **QUICK Scheme** was developed to avoid the instabilities associated with central differencing, while also reducing the numerical dissipation caused by upwinding (Leonard, 1979). As indicated by the name of this method, a weighted quadratic interpolation is used from various, case dependent, cell faces in order to determine the value on the face of interest. The result of implementing the QUICK scheme is a solution that is both stable and less diffusive. Additionally, the QUICK scheme allows for a larger grid size to be implemented which reduces the computational cost.

2.3.3 Sub-Grid Scale Models

Original Smagorinsky Model

The first closure model for calculating the SGS stress tensor was developed in order to simulate the dynamic circulation of the atmosphere (Smagorinsky, 1963). The model was developed under the assumption of an "energy cascade", in which the smallest scale motions dissipate the energy received from larger scale motions. Algebraically, this is done by relating the SGS stress to the filtered rate of strain by means of a turbulent viscosity ν_t

$$\tau_{ij}^{sgs} = -2\nu_t \tilde{S}_{ij} \tag{2.20}$$

where $\tilde{S}_{ij}$ is the filtered rate of strain, defined as

$$\tilde{S}_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) \tag{2.21}$$

Finally, the turbulent viscosity is calculated by

$$\nu_t = (C_s \Delta)^2 |\tilde{S}| \tag{2.22}$$

where C_s is the Smagorinsky constant, Δ is the width of the filter (often taken to be two grid sizes), and $|\tilde{S}|$ is the magnitude of the filtered strain rate tensor. The Smagorinsky constant was determined to be $C_s = 0.18$ for homogeneous, isotropic turbulence (Lilly, 1966). For different types of flow (i.e. shear flows) the value of this constant varies. This is the reason for one of the largest pitfalls of the original Smagorinsky model. Since the value of the constant is case dependent, it needs to be known *a priori* which can lead to errors when solving flow fields that have not been extensively studied. Additionally, this model does not allow for backscatter of energy, a process by which the energy of the SGS motions is transferred up to the larger scale motions.

Dynamic Smagorinsky Model

In order to address the inability of the original Smagorinsky model to dynamically adjust depending on the type of flow field, a new model for calculating the constant was developed (Germano et al., 1991), (Lilly, 1992). In this method, termed the dynamic Smagorinsky model, the Smagorinsky constant is a function of both space and time. This model is calculated by using two filters: a "grid filter", which aligns with the grid of the simulation, and a "test filter", which is coarser than the grid filter. The dynamic Smagorinsky constant is calculated as

$$C_s = -\frac{1}{2}\left(\frac{L_{ij}M_{ij}}{M_{ij}M_{ij}}\right) \tag{2.23}$$

In this calculation, L_{ij} represents the subtraction of the test-averaged SGS stress tensor from the sub-test scale (STS) stress tensor and is calculated as

$$L_{ij} = -\widehat{u_i u_j} + \hat{\tilde{u}}_i \hat{\tilde{u}}_j \tag{2.24}$$

where the hat indicates test filtering, and the tilde indicates grid filtering. Likewise, M_{ij} is calculated as

$$M_{ij} = \hat{\Delta}^2 |\hat{\tilde{S}}| \hat{\tilde{S}}_{ij} - \Delta^2 \widehat{|\tilde{S}|\tilde{S}_{ij}} \tag{2.25}$$

For a more extensive derivation and explanation of these calculations, see (Lilly, 1992). The calculation of the dynamic Smagorinsky constant is superior to the original method because it varies spatially and temporally depending on the flow field. It can also have negative values which allows for the backscatter of energy.

The dynamic Smagorinsky model also allows for calculation of the SGS heat flux, using the same Smagorinsky constant. Given C_s, the SGS heat flux can be calculated as

$$\lambda_j^{sgs} = \frac{2C_s\Delta^2}{Pr_t}|\tilde{S}|\frac{\partial \tilde{T}}{\partial x_j} \tag{2.26}$$

where Pr_t is the turbulent Prandtl number and is calculated similarly to the turbulent

viscosity. See (Lilly, 1992) for an extensive derivation. All simulations presented in this thesis use the dynamic Smagorinsky model due to its ability to more accurately represent the SGS motions.

2.3.4 Immersed Boundary Method

As was discussed in Section 2.3.1, one of the disadvantages of using a structured mesh is that complicated geometries will not necessarily align with the grid. In order to allow for the simulation of complicated fluid-solid interfaces on a Cartesian mesh, the IB method was developed. The IB method was initially developed to perform simulations of blood flow through heart valves (Peskin, 1972). The main goal of the IB method is to capture a solid's effect on the fluid domain by applying a source (forcing) term to the fluid nodes directly adjacent to the solid. The location of the solid is determined using a Lagrangian framework, whereas the fluid flow is solved using an Eulerian framework. In Figure 2.4, the forcing terms would be applied to the diamond nodes. At all other nodes, the typical

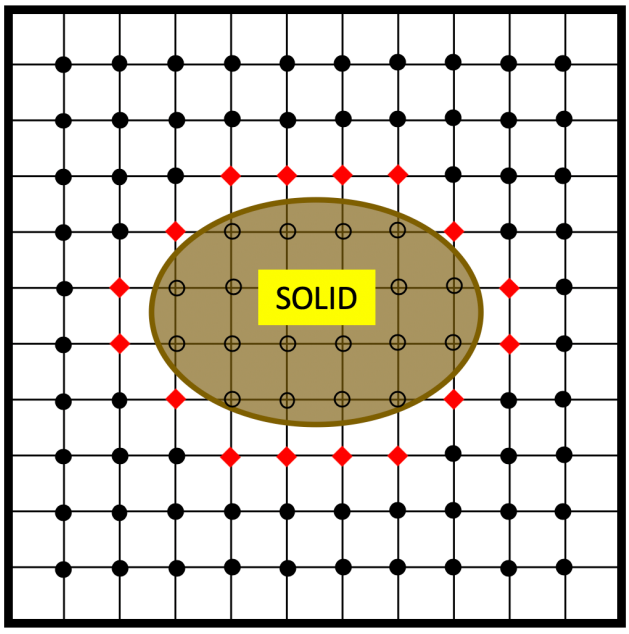


FIGURE 2.4: Two dimensional schematic of IB nodes for flow over ellipse. External forcing is applied to diamond nodes. Circle nodes are regular fluid nodes.

governing equations (2.12-2.14) would be solved. There are two categories by which this forcing term can be implemented into the governing equations, continuous forcing and discrete forcing. With continuous forcing, the forcing term is applied to the governing equations prior to discretization. Discrete forcing involves applying the forces after the discretizations have been made.

Continuous Forcing

The continuous forcing IB method was the first IB method developed by Peskin in 1972. Since this method involves applying a forcing function to the governing equation prior to the discretization, a continuous function representing the force from the solid onto the fluid is required. The function of the force applied from the solid to the fluid is

$$\boldsymbol{f}_m(\boldsymbol{x}, t) = \sum_k \boldsymbol{F}_k(t)\delta(||\boldsymbol{x} - \boldsymbol{X}_k||_2) \tag{2.27}$$

where $\boldsymbol{f}_m$ is the external force applied to the Eulerian fluid grid, $\boldsymbol{F}_k$ is the stress on the k^{th} Lagrangian solid point, $\boldsymbol{X}_k$ is the Lagrangian solid point location, δ is the Dirac delta function, and $||\cdot||_2$ is the Euclidean norm. In simulations where a discrete grid is implemented, the Dirac delta function must be replaced by smoother distribution functions. Various smooth distribution functions have been documented for different simulation methods (Mittal and Iaccarino, 2004).

The continuous forcing IB method is particularly effective for simulating flows with elastic immersed boundaries because it is simple to implement. It is also not necessary to capture the exact interface between fluid and solid. This method has been effectively used to simulate many biological flows such as blood flow (Peskin, 1982), air flow in the inner ear (Beyer, 1992), and aquatic locomotion (Fauci and McDonald, 1995).

Discrete Forcing

The discrete forcing method applies the forcing function to the governing equations after they have been discretized. When implementing the discrete IB method, the governing equations are first solved at the fluid nodes, and then the pressure Poisson equation is solved along with the external forces. This results in a model that is able to more accurately capture the fluid-solid boundary. Since this method results in an accurate representation of the fluid-solid interface, and does not require a solution for solid nodes, this is an effective method for solving high Reynolds number flows. This method has been implemented to simulate flow in a combustion engine (Verzicco et al., 1998) and flow around various blunt objects (He, Yang, and Shen, 2017).

2.3.5 Level-Set Functions

In order to implement the IB method, the position of the solid-fluid interface with regard to each fluid point must be determined. One common method of finding this is to use a level-set function $\phi(\boldsymbol{x}, t)$. The level-set function is a signed function which indicates the distance between a particular Eulerian point and the solid-fluid interface. The sign of ϕ indicates whether the Eulerian point is located in the fluid or solid domain and is defined as

$$\phi(\boldsymbol{x}, t) = \begin{cases} d & \text{Fluid Point} \\ 0 & \text{On Interface} \\ -d & \text{Solid Point} \end{cases} \tag{2.28}$$

where d is the smallest distance between the Eulerian point and the solid-fluid interface. For simple geometries, it may be possible to easily define the level-set function. For instance, if the solid body is a sphere of radius R centered at $\boldsymbol{x}_o(t)$, then the level-set function would be defined as

$$\phi(\boldsymbol{x}, t) = ||\boldsymbol{x} - \boldsymbol{x}_o(t)||_2 - R \tag{2.29}$$

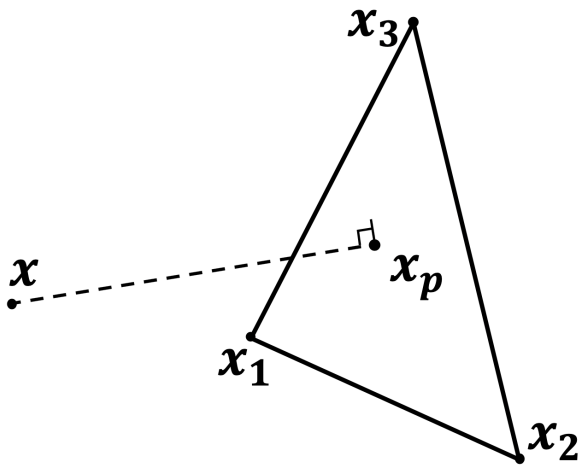


FIGURE 2.5: Schematic for calculation of distance from Eulerian fluid point (x) to solid mesh triangle.

where $||\cdot||_2$ is the Euclidean norm. However, for geometries that are much more complicated, it is very difficult to determine an analytical definition of the level-set function.

Complicated Geometries

In order to define the level-set function for complicated geometries, the solid needs to be discretized. This is done by breaking the three dimensional solid into an array of two dimensional triangles. The level-set function is then calculated by determining the distance from the Eulerian point to each triangle, and then selecting the minimum of all of these distances. To calculate the minimum distance from a Eulerian point $\boldsymbol{x}$ to a triangle $\Delta \boldsymbol{x_1 x_2 x_3}$ (Figure 2.5), $\boldsymbol{x}_p$ is calculated by projecting $\boldsymbol{x}$ onto the plane defined by $\Delta \boldsymbol{x_1 x_2 x_3}$. Next, the following linear system is solved to determine if the projected point is contained within the triangle.

$$\alpha \boldsymbol{x_1} + \beta \boldsymbol{x_2} + \gamma \boldsymbol{x_3} = \boldsymbol{x}_p \tag{2.30}$$

If it is found that α, β, and γ are all in $[0, 1]$, then the nearest point on the triangle is the projected point, and the magnitude of ϕ is defined as

$$d = ||\boldsymbol{x} - \boldsymbol{x}_p||_2 \tag{2.31}$$

If only one of α, β, and γ is positive, then the nearest point on the triangle is one of the vertices, and the the magnitude of ϕ is defined as

$$d = \begin{cases} ||\boldsymbol{x} - \boldsymbol{x_1}||_2, & \alpha > 0 \\ ||\boldsymbol{x} - \boldsymbol{x_2}||_2, & \beta > 0 \\ ||\boldsymbol{x} - \boldsymbol{x_3}||_2, & \gamma > 0 \end{cases} \tag{2.32}$$

Lastly, if only one of α, β, and γ is negative, then the nearest point on the triangle is on one of the lines and the the magnitude of ϕ is defined as

$$d = \begin{cases} distance(\boldsymbol{x}, \overline{\boldsymbol{x_2 x_3}}), & \alpha < 0 \\ distance(\boldsymbol{x}, \overline{\boldsymbol{x_1 x_3}}), & \beta < 0 \\ distance(\boldsymbol{x}, \overline{\boldsymbol{x_1 x_2}}), & \gamma < 0 \end{cases} \tag{2.33}$$

where $distance()$ is a function that calculates the shortest distance between a point and a line.

The final step of calculating the level-set function is to determine its sign (whether a point is in the solid or fluid domain). This is done by using the ray casting method, an algorithm initially used in the computer graphics community (Roth, 1982). Once a distance has been determined for an Eulerian point $\boldsymbol{x}$, a random ray is projected away from the point. The number of triangles N this ray intersects is calculated. Assuming the ray exits the domain in the fluid subdomain, the level-set function can be calculated as

$$\phi(\boldsymbol{x}, t) = \begin{cases} d & \text{N even or 0} \\ -d & \text{N odd} \end{cases} \tag{2.34}$$

2.3.6 Wall Models

One of the greatest difficulties in simulating the ABL is being able to accurately represent the turbulent flow in the region near the wall. Due to the high shear stress near the wall

surface, very large velocity gradients are present and the ability to accurately capture these effects is necessary to obtaining realistic simulation results.

Brief overview of wall bounded turbulence notation

One of the best ways to characterize wall-bounded turbulent flow (such as wind flow over the Earth's surface) is by analyzing the wall shear stress τ_w. The wall shear stress is defined as

$$\tau_w \equiv \mu \frac{\partial \langle U \rangle}{\partial y} \tag{2.35}$$

where $\langle U \rangle$ is the mean velocity parallel to the wall, and y is the normal direction to the wall. From this, it is possible to define a characteristic velocity, length scale, and Reynolds number. The friction velocity u_* is defined as

$$u_* \equiv \sqrt{\frac{\tau_w}{\rho}} \tag{2.36}$$

The viscous lengthscale δ_ν is defined as

$$\delta_\nu \equiv \nu \sqrt{\frac{\rho}{\tau_w}} = \frac{\nu}{u_*} \tag{2.37}$$

The friction Reynolds number Re_τ is thus defined as

$$Re_\tau \equiv \frac{u_* \delta}{\nu} = \frac{\delta}{\delta_\nu} \tag{2.38}$$

where δ is the length of the boundary layer. From these characteristic scales, a non-dimensional length and velocity can be defined as

$$y^+ \equiv \frac{y}{\delta_\nu} \tag{2.39}$$

$$u^+ \equiv \frac{\langle U \rangle}{u_*} \tag{2.40}$$

The law of the wall

One of the most important realizations of wall-bounded turbulent flows was discovered when Prandtl postulated that the mean velocity profile near a wall only depends on the viscous scales (Prandtl, 1925). The result of this postulation, which has since been validated by many studies (Klebanoff, 1954); (Kim, Moin, and Moser, 1987); (Spalart, 1988), is that u^+ is only a function of y^+ (e.g. $u^+ = f_w(y^+)$). This allows for a universal, non-dimensional function $f_w(y^+)$ to be developed to define the mean profile of flows near the wall $y/\delta < 0.1$. It has been determined that there are two important regions in the boundary layer: the viscous sublayer and the log-law region.

The viscous sublayer is defined as the region nearest the wall where the gradient of the mean velocity is approximately equal to unity. It has been found that this approximation is valid for $y^+ < 5$. The log-law region is the region further away from the wall and hence the effect of viscosity is significantly reduced. This results in a velocity gradient of

$$\frac{\partial u^+}{\partial y^+} = \frac{1}{\kappa y^+} \tag{2.41}$$

where κ is the von Karman constant. By integrating 2.41, an analytic form of the mean velocity in the log-layer region is defined as

$$u^+ = \frac{1}{\kappa} \ln y^+ + B \tag{2.42}$$

where B is the integration constant. The values of these constants were found to be $\kappa \approx 0.41$ and $B \approx 5.2$ (Karman, 1930). The log-law region is valid for $y^+ > 30$. The region between the viscous sublayer and the log-law region, called the buffer layer, contains both viscous dominated and turbulence dominated flows. These three layers can be seen in Figure 2.6, with data taken from a DNS of channel flow with $Re_\tau = 1020$ (Abe, Kawamura, and Matsuo, 2004).

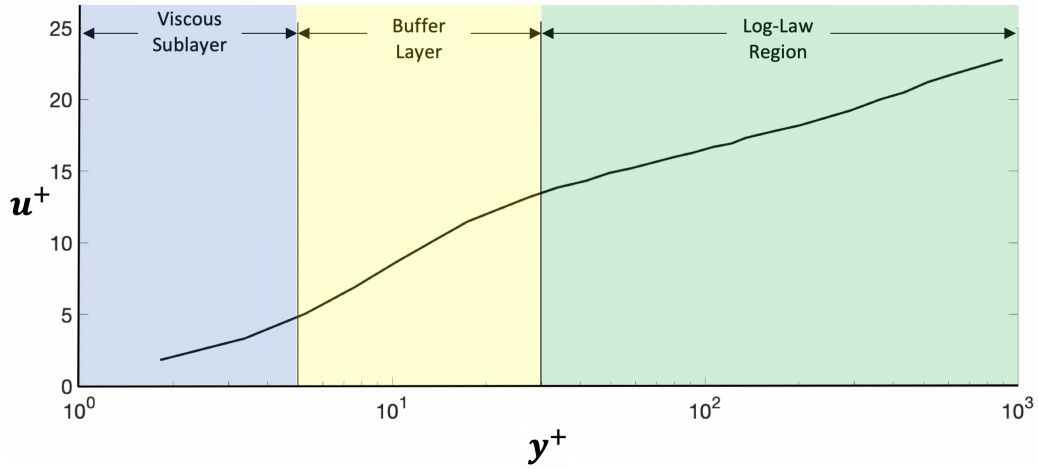


FIGURE 2.6: Mean streamwise velocity from channel flow DNS at $Re_\tau = 1020$. (Abe, Kawamura, and Matsuo, 2004).

Wall-resolve v Wall-model

In order to accurately simulate wall-bounded turbulent flows, it is essential that all of the phenomena of the near-wall region are captured by the simulation. When performing LES of wall bounded flows, there are two options that can be used: wall-resolved LES (WR-LES) and wall-modeled LES (WM-LES). Implementing a WR-LES requires that all of the small scale motions near the wall be captured directly by the mesh. For this reason, a requirement of WR-LES is that the grid point nearest the wall be in the viscous sublayer ($y^+ < 5$). This is because simulations typically use a linear interpolation to determine the velocity at the nearest point, and the velocity profile is only linear in the viscous sublayer. Using WR-LES results in solutions that are particularly accurate even near the wall. However, performing WR-LES can become expensive, particularly if the viscous sublayer is small (Re_τ is very large). In the ABL, the friction velocity u_* is typically on the order of $O(0.1\frac{\mathrm{m}}{\mathrm{s}})$ (Weber, 1999). This corresponds to a viscous sublayer that has a length scale on the order of $O(1mm)$. Therefore, to perform a WR-LES of the ABL, a spatial resolution on the order of $O(1mm)$ would be required near the surface, which is unrealistic with current computational power.

To accurately simulate flows with high Reynolds numbers, such as the ABL, WM-LES

must be used. The goal of WM-LES is to model the near-wall region in order to decrease the computational cost required. There have been many methods developed to model the near-wall region, but one of the simplest methods is to use the law of the wall. It is known from Equation 2.42 that in the log-law region of the boundary layer, there is a relationship between the non-dimensional, mean velocity u^+ and the non-dimensional distance from the wall y^+. Using this knowledge, one simple model would be to assume the grid point nearest to the wall is in the log-law region, and interpolate the velocity using Equation 2.42, rather than using a linear interpolation. However, various studies have shown that this simple method leads to an error deemed the 'log-layer mismatch' by which the wall shear stress is either underpredicted (Kawai and Larsson, 2012); (Roman, Armenio, and Fröhlich, 2009) or overpredicted (Lee, Cho, and Choi, 2013).

There have been a variety of proposed methods of addressing the 'log-layer mismatch' associated with WM-LES. Numerous studies initially considered modifying the calculation of the SGS stress models near the wall in an attempt to more accurately calculate the wall shear stress (Chow et al., 2005); (Ding, Arya, and Lin, 2001). However, it was found that these methods were not robust enough to handle many different cases. Therefore, a new method was implemented by which the first point away from the wall was no longer used, and instead LES data from a point further away from the wall was used (Kawai and Larsson, 2012). This resulted in a much more robust method of reproducing the effects of the near wall region. One drawback of this method is it becomes difficult to classify which grid point to use, particularly when a complicated geometry is involved.

2.3.7 Inflow Condition

When simulating highly turbulent flows such as the ABL, it is important that the inflow condition has the chaotic velocity fluctuations associated with turbulence. If, by contrast, a laminar flow is used as an inlet boundary condition, the flow may remain laminar for a significant time in the simulation, which is unrealistic when attempting to simulate the ABL.

To ensure accurate inflow boundary conditions, many different approaches have been implemented.

Periodic Domains

For cases with no spatial variability in the domain, it may be possible to use a periodic domain rather than an inflow/outflow condition. In these idealized cases the flow is driven by constant pressure gradient or constant shear stress, and the flow at the outlet is 'recycled' to the inlet of the domain. The benefits of using a periodic domain is that the initial condition is not an important consideration when setting up the simulation. As long as the initial condition has an approximately accurate profile with velocity fluctuations, the inlet flow field will develop into physical turbulence. This assumes that the simulation has run sufficiently long enough for the flow field to develop. Using a periodic domain with a constant pressure gradient was used by some of the first numerical studies that examined the ABL (Deardorff, 1972). This method is still being used today when periodic physical boundaries are considered (Chen and Chamecki, 2019). One of the greatest drawbacks of using a periodic domain is that it is difficult to study the effect of wakes caused by blunt objects such as buildings or topography.

Auxiliary Simulation

For simulations that do not have homogeneity in the mean direction of the flow, it is possible to run an auxiliary, periodic simulation prior to the main simulation. The results of this auxiliary simulation can then be used as a boundary condition for the main simulation (See Figure 2.7). Since the main simulation is not periodic, care must be taken to ensure that the outflow condition is correct. The outlet boundary condition is chosen to ensure that the mass and momentum of the entire simulation is conserved. The benefit of using this method is that it allows for the study of wakes and heterogeneous boundary conditions, while also ensuring that the inflow boundary condition is physical. The obvious downside is that this requires a second simulation be run, which will significantly increase the

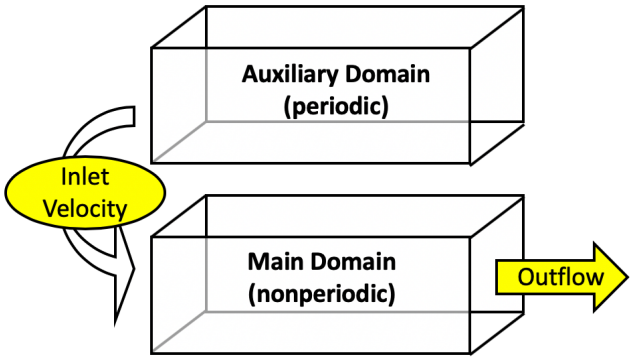


FIGURE 2.7: Schematic of auxiliary simulation method. Flow field from auxiliary domain is used as inlet velocity for main domain.

computational cost. Additionally, it can be difficult to specify a particular mean velocity profile for the inlet based off of field or lab measurements.

Synthetic Turbulence Generation

Another method of generating an inflow boundary condition is to generate synthetic, turbulent flow fields. The velocity field is then used as the inflow boundary condition for the simulation (See Figure 2.8). The synthetic field is generated by numerically integrating an energy spectrum of homogeneous turbulence using a stochastic Fourier-Stieltjes integral

$$u_i(\boldsymbol{x}) = \sum_k e^{i\boldsymbol{k}\cdot\boldsymbol{x}} C_{ij}(\boldsymbol{k}) n_j(\boldsymbol{k}) \tag{2.43}$$

where C_{ij} is a tensor based on the energy spectrum of homogeneous turbulence, $\boldsymbol{k}$ is the wavenumber, and n_j are stochastic, Gaussian, complex variables (Mann, 1998). The result of evaluating 2.43 is an instantaneous, three dimensional flow field with homogeneous turbulence. This three dimensional flow field can then be converted into a two dimensional time series by using Taylor's frozen turbulence hypothesis (Taylor, 1938). Finally, this two dimensional time series can be used as the inlet velocity condition for the main simulation.

One benefit of using synthetic turbulence to generate an inflow condition is that it requires

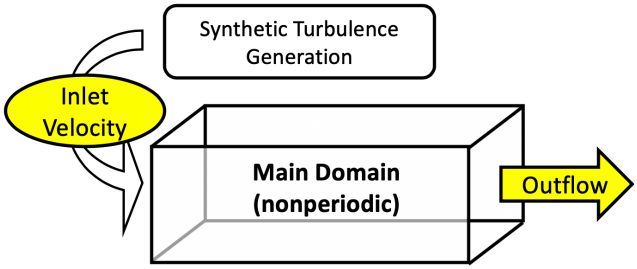


FIGURE 2.8: Schematic of synthetic turbulence generation simulation method. Flow field from synthetic turbulence generation is used as inlet velocity for main domain.

less computational work than running a completely separate auxiliary simulation. Additionally, it is easy to adjust the inflow boundary condition to match a particular mean profile. This is particularly helpful when simulation results are being compared to laboratory and field measurements. The downside of this method is that the turbulence generated is not generated directly by the governing equations, so it is not physical. For this reason, simulations using synthetic turbulence generation may require an entrance region in order for the synthetic, turbulent field to develop into physical turbulence (Yang et al., 2019).

Chapter 3

Numerical Methods and Validation

3.1 Temperature Transport

In order to accurately simulate the ABL, conservation of energy needs to be implemented. This takes the form of a convection-diffusion equation. Recall from Section 2.2.3 that the filtered temperature equation solved by LES is

$$\frac{\partial \tilde{\theta}}{\partial t} + \tilde{u}_j \frac{\partial \tilde{\theta}}{\partial x_j} = \alpha \frac{\partial^2 \tilde{\theta}}{\partial x_j \partial x_j} - \frac{\partial \lambda_j^{sgs}}{\partial x_j} \tag{3.1}$$

In this section, the implementation and verification of the convection and diffusion terms will be described. Finally, the complete equation will be validated with two channel flow tests.

3.1.1 Diffusion

The diffusion term is discretized using second order, central differencing. The second derivatives of temperature are calculated at the cell center. On the staggered grid, the resultant second derivative in the x direction is discretized as

$$\frac{\partial^2 \theta}{\partial x^2} = \frac{2}{\Delta x_i} \left[\frac{\theta_{i+1} - \theta_i}{\Delta x_{i+1} + \Delta x_i} - \frac{\theta_i - \theta_{i-1}}{\Delta x_i + \Delta x_{i-1}} \right] \tag{3.2}$$

The derivatives in the other directions are calculated similarly.

Verification

In order to verify the calculation of the diffusion term in 3.1, the convection and SGS heat flux terms are ignored. This results in the three dimensional heat equation. Unbounded, three dimensional diffusion is used to test the convergence of the diffusion term. The governing equation and initial condition are defined as

$$\begin{aligned} \frac{\partial \theta}{\partial t} &= D\frac{\partial^2 \theta}{\partial x_j \partial x_j}, \quad x_1, x_2, x_3 \in \mathbb{R}^3, \quad t > 0 \\ \theta\Big|_{t=0} &= M\delta(x_1, x_2, x_3) \end{aligned} \tag{3.3}$$

where $\delta(x_1, x_2, x_3)$ is the Dirac delta function, M represents the 'total temperature' in the domain, and D represents the diffusion rate. The analytic solution to Equation 3.3 can be calculated using the fundamental solution of the heat equation

$$\theta^* = \frac{M}{(4\pi t D)^{3/2}} \exp\left(\frac{-(x_1^2 + x_2^2 + x_3^2)}{4tD}\right) \tag{3.4}$$

There are two main problems when attempting to replicate this analytic solution using a numerical solution. The first problem is that it is impossible to accurately replicate the initial condition. This is because it is not possible to represent a Dirac delta function using a finite grid. To mitigate this error, the initial conditions to the numerical solution can be the analytic solution at a time t^* just after $t = 0$ when the analytic solution is a continuous function. The other problem is that it is impossible to numerically simulate an infinite domain. In order to address this problem, the domain is simply chosen to be large enough that the edge of the boundary has a minimal effect on the diffusion near the center. The homogeneous Dirichlet boundary condition is used on the edge of the domain. The problem with this method is that if the simulation runs for long enough, the boundary conditions will eventually have an effect on the solution. To mitigate this problem the total run time of the solution is bounded. By taking these considerations into account, the new governing

equation becomes

$$\begin{gathered}\frac{\partial\theta}{\partial t} = D\frac{\partial^2\theta}{\partial x_j \partial x_j}, \quad x_1, x_2, x_3 \in \Omega, \quad t > t^* \\ \theta = 0 \in \partial\Omega \\ \theta\Big|_{t=0} = \theta^*(t^*)\end{gathered} \tag{3.5}$$

where Ω is the box $[-L, L] \times [-L, L] \times [-L, L]$, and $\partial\Omega$ is the boundary of Ω. The convergence of the solution was tested by calculating the shape error E_{shape} at the end of the simulation $t = t^F$. The shape error is defined as

$$E_{shape} = \sum_n \left|\theta_n - \theta_n^*\right| \cdot \Delta x_1 \Delta x_2 \Delta x_3 \tag{3.6}$$

where the summation is over all the grid cells in the simulation. The simulation was performed on three separate grid sizes, $(Nx)^3 = (32)^3, (64)^3, (128)^3$, with diffusion rate $D = 0.1$, initial temperature $M = 2$, initial time $t^* = 0.05$, and final time $t^F = 2.05$. For the time advancement, a RK2 method was used. As can be seen in Figure 3.1, the shape error converges with second order convergence, as expected.

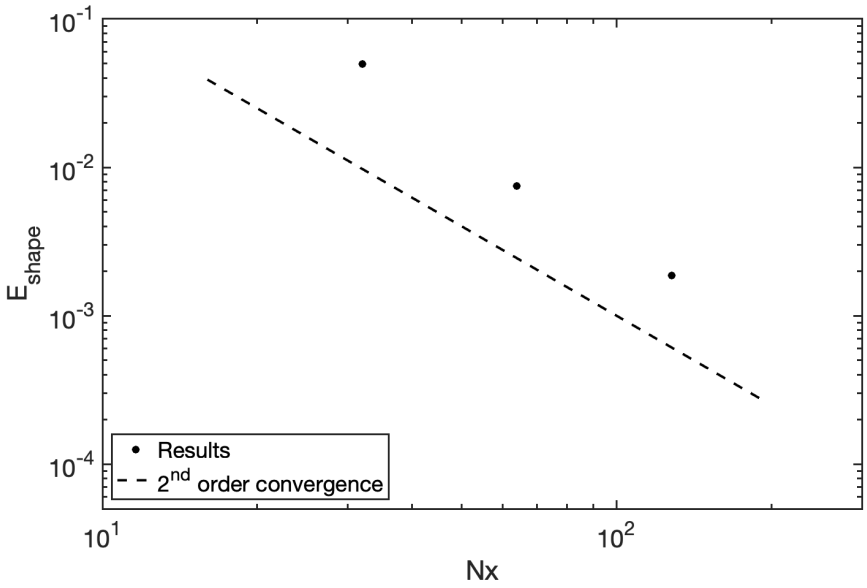


FIGURE 3.1: Shape error convergence of three dimensional diffusion equation.

3.1.2 Convection

The convection term is discretized with two different methods. At first, the derivative was discretized using a central differencing method. However, it was found that this led to instabilities, so an upwinding total variation diminishing (TVD) scheme was implemented (Harten, 1983). The idea behind the TVD scheme is to preserve monotonicity. With regards to a TVD scheme, preserving monotonicity means that the maximums of a function must not be increasing and the minimums of a function must not be decreasing. This prevents the creation of new local extrema that are associated with instabilities. The result of using a TVD scheme is a stable solution that maintains second order accuracy while also reducing the amount of numerical dissipation associated with a strictly upwinding scheme.

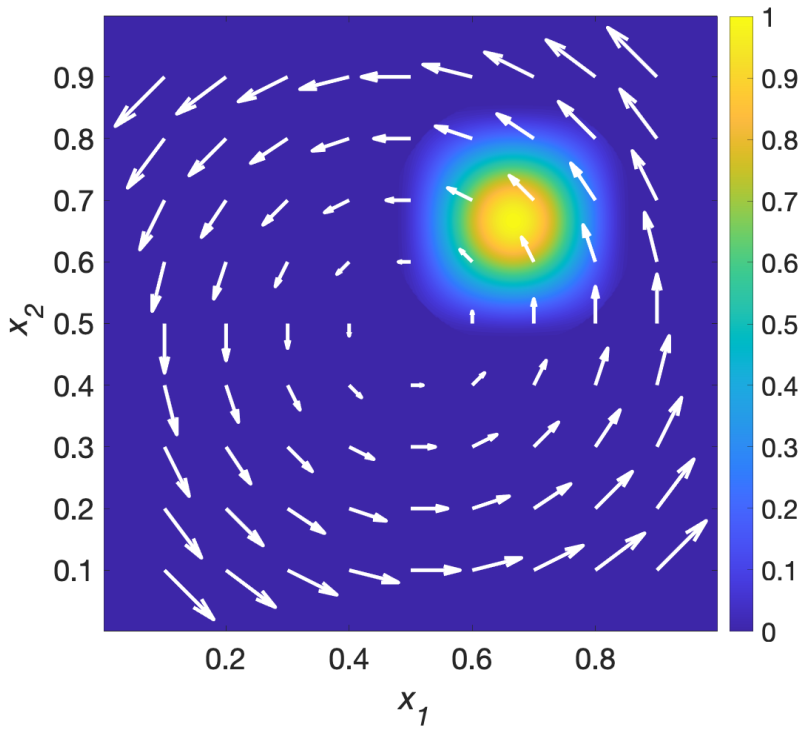


FIGURE 3.2: Initial temperature distribution (contours) and velocity field (vectors).

Verification

To verify the calculation of the convection term in 3.1, the diffusion and SGS heat flux terms are ignored. The two dimensional rotating cosine hill verification case is used to study the convergence of the convection schemes (Donea and Huerta, 2013). The governing equation, initial condition, and boundary conditions are defined as

$$\frac{\partial \theta}{\partial t} = u_j \frac{\partial \theta}{\partial x_j}, \quad x_1, x_2 \in \Omega, \quad t > 0$$

$$\theta_0 = \begin{cases} \frac{1}{4}\left(1+\cos\left(\pi\frac{x_1-2/3}{0.2}\right)\right)\left(1+\cos\left(\pi\frac{x_2-2/3}{0.2}\right)\right), & \left(\frac{x_1-2/3}{0.2}\right)^2+\left(\frac{x_2-2/3}{0.2}\right)^2 < 1 \\ 0, & \left(\frac{x_1-2/3}{0.2}\right)^2+\left(\frac{x_2-2/3}{0.2}\right)^2 \geq 1 \end{cases}$$

$$\theta = 0, \quad (x_1, x_2) \in \partial\Omega \tag{3.7}$$

with the domain $\Omega \in [0,1] \times [0,1]$ contained by $\partial\Omega$. The temperature is transported by a constant rotational velocity field

$$\boldsymbol{u} = u_1\boldsymbol{e}_1 + u_2\boldsymbol{e}_2 = (-x_2 + 0.5)\boldsymbol{e}_1 + (x_1 - 0.5)\boldsymbol{e}_2 \tag{3.8}$$

The initial temperature distribution and velocity field are shown in Figure 3.2. This velocity field is purely rotational, so if the solution were analytically solved, the 'temperature hill' would rotate about $(0.5, 0.5)$ and return to the exact same position. To verify the numerical solution, the simulation is run until the 'temperature hill' completes one full rotation, and then the final temperature distribution is compared to the initial condition. Simulations at grid sizes of $(Nx)^2 = (25)^2, (50)^2, (100)^2, (200)^2, (400)^2$ were performed. The time step Δt was chosen so that the Courant-Friedrichs-Lewy (CFL) condition (a condition necessary for convergence of hyperbolic PDEs) was held.

$$C = \max(\boldsymbol{u})\frac{\Delta t}{\Delta x} < 0.5 \tag{3.9}$$

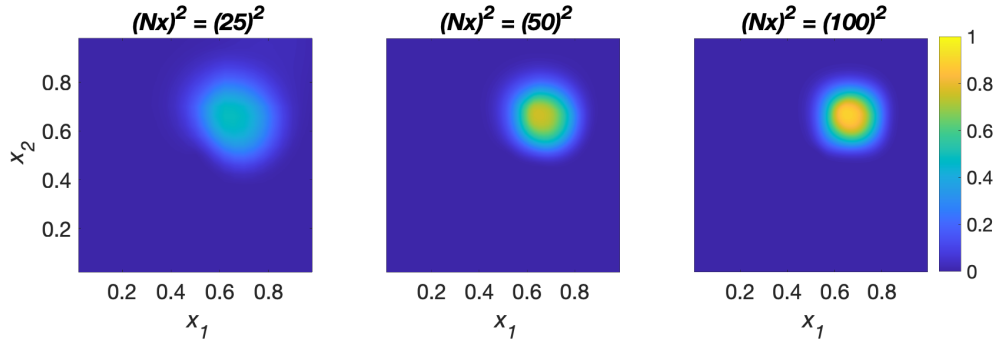


FIGURE 3.3: Final temperature distribution for three grid sizes.

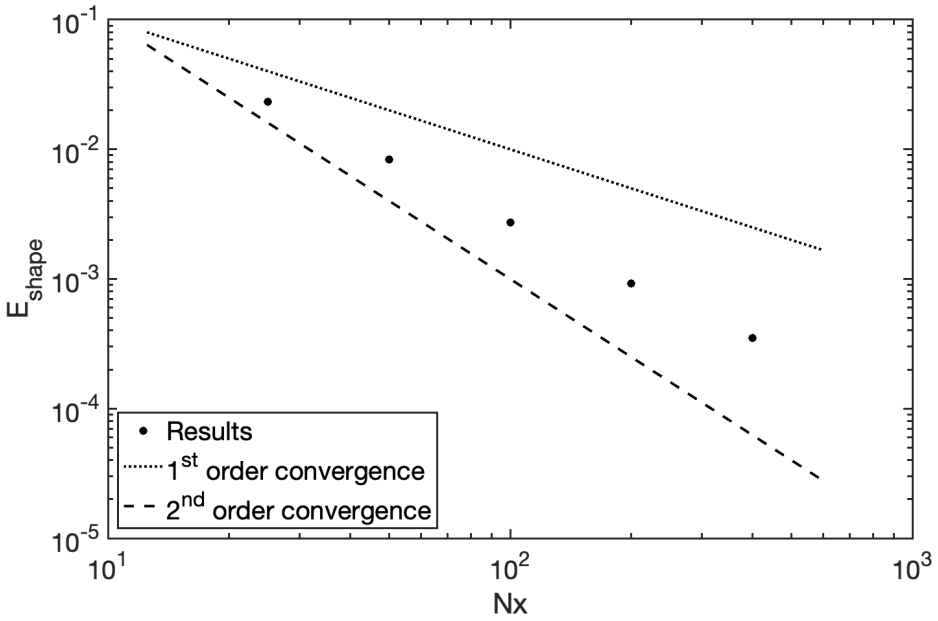


FIGURE 3.4: Convergence test for rotating two dimensional cosine hill test.

The shape error (Equation 3.6) was calculated to analyze the convergence of the discretizations. The final temperature distribution for three of the simulations can be seen in Figure 3.3. It can be seen that the final solution approaches the initial condition as the resolution of the simulation increases. The results of this convergence test can be seen in Figure 3.4. From this graph, it can be seen that the convergence of the shape error is between first and second order convergence. While second order convergence is expected, it is possible that this discrepancy could be from the implementation of upwinding.

3.1.3 Channel Flow Validation

The SGS heat flux term is calculated using the dynamic Smagorinsky model discussed in Section 2.3.3. Once the SGS heat flux term has been calculated at all of the fluid points, the gradient is then calculated using second order, central differencing.

In order to ensure the convection-diffusion equation (Equation 3.1) was being accurately solved, fully developed turbulent channel flow with temperature was simulated. The results were validated through comparison to DNS results (Jaszczur, 2014). See Figure 3.5 for the case setup. In this WR-LES, the flow parameters were $Re_\tau = 150$ and $Pr = 1.0$. The grid parameters were $(N_x, N_y, N_z) = (64, 48, 64)$, where the y-direction grid refines near the wall to resolve the near wall turbulence. The flow was driven by a constant pressure gradient, and the horizontal boundary conditions were periodic.

Before the temperature equations were solved, the mean velocity profile and turbulent fluctuations needed to be validated. Channel flow with $Re_\tau = 150$ was simulated by solving the filtered conservation of mass and momentum equations (2.12,2.13). Once this solution reached a statistically steady state, the mean streamwise velocity and velocity

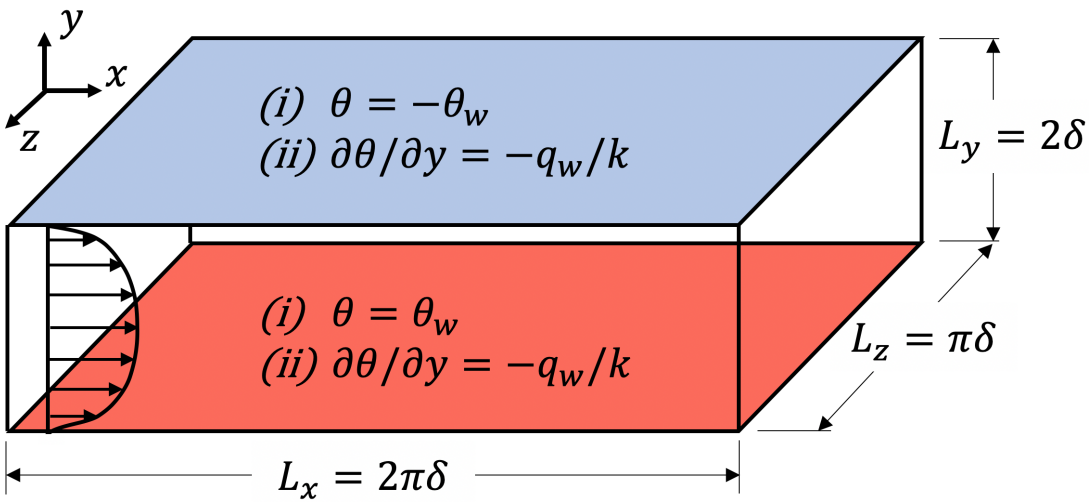
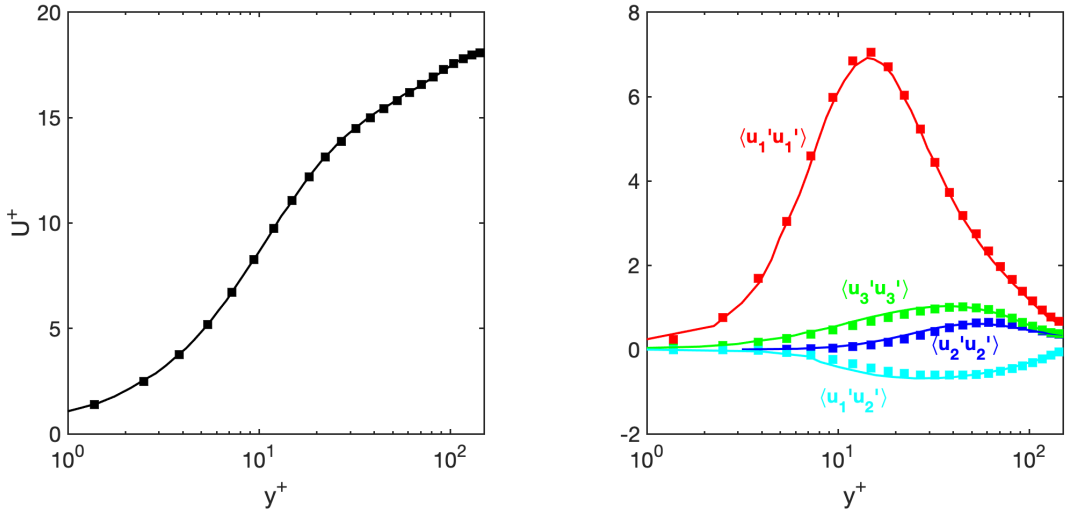


FIGURE 3.5: Channel flow setup with two distinct vertical boundary conditions, (i) constant wall temperature θ_w (Dirichlet) and (ii) constant heat flux q_w (Neumann).

fluctuations were analyzed to ensure the flow characteristics of channel flow were accurate. These results were compared to DNS results in Figure 3.6. As can be seen from these figures, the present LES was able to accurately replicate the results from DNS. Once a fully developed channel flow field was generated, the temperature solver was turned on and two separate boundary conditions were considered: (i) constant wall temperature and (ii) constant heat flux.



(a) Normalized mean streamwise velocity profile.

(b) Normalized plots of velocity fluctuations

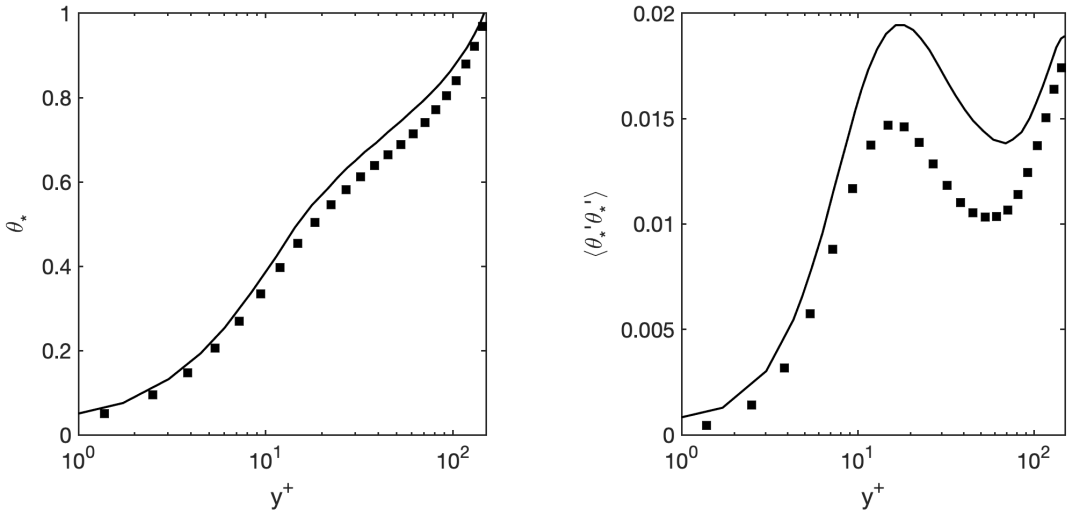
FIGURE 3.6: Velocity statistics for channel flow at $Re_\tau = 150$. Results from LES (■) are compared with DNS (—) (Jaszczur, 2014).

Validation I: Constant wall temperature

The channel flow with constant wall temperature validation test was run until a statistically steady solution was reached. The mean profiles and turbulence statistics were then calculated from forty time realizations, separated by approximately ten non-dimensional time units. In terms of the large eddy turnover time ($t_e = \frac{2\delta}{u_*}$), this corresponds to five times the large eddy turnover time between realizations. In order to compare present LES studies to DNS results, a non-dimensional temperature related to the wall temperature is defined as

$$\theta_* = \frac{\theta - \theta_w}{\theta_w} \tag{3.10}$$

The mean temperature distribution and second-order statistics are compared to DNS results in Figure 3.7. It can be seen that the mean temperature profile closely follows the DNS results. The temperature fluctuations are slightly underpredicted when compared to DNS results. Despite the discrepancy in magnitude of temperature fluctuations, it can be seen that the trend (i.e. local maxima and minima) are accurately captured. These results, while not perfect, indicate that the temperature equation is accurately being simulated. These results could likely be improved by implementing more advanced convection schemes and SGS models.



(a) Normalized mean temperature profile.

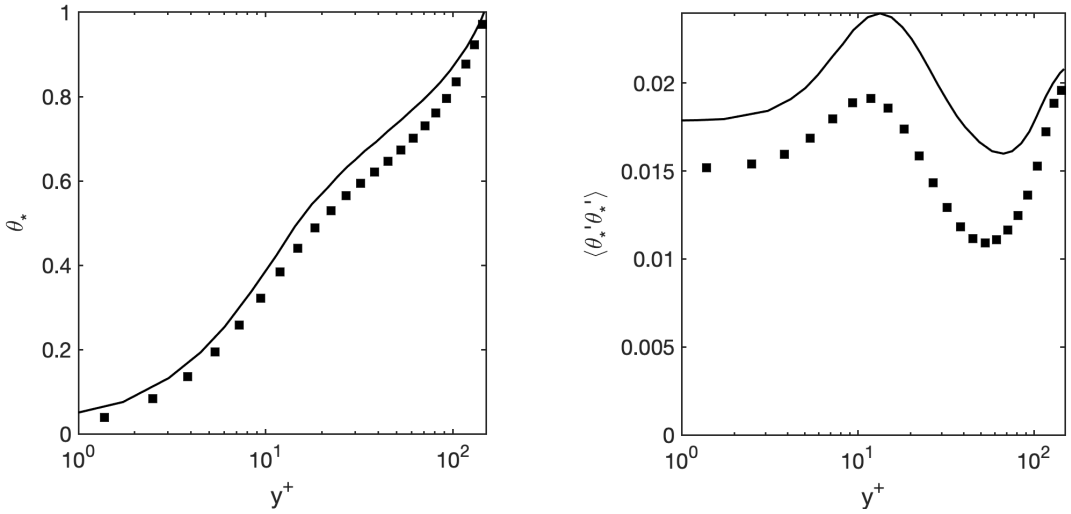
(b) Normalized temperature fluctuations.

FIGURE 3.7: Temperature statistics for channel flow at $Re_\tau = 150$ with the Dirichlet boundary condition. Results from LES (■) are compared with DNS (—) (Jaszczur, 2014).

Validation II: Constant heat flux

The channel flow with constant heat flux validation test was simulated using the same case setup as the constant temperature case. The only difference with this method was that the wall temperature θ_w was calculated using a spatial and temporal average. This is because the wall temperature fluctuates due to the Neumann boundary condition.

The mean temperature and second-order statistics are compared with DNS results in Figure 3.8. Similar to the constant wall temperature case, the mean temperature is accurately captured, while the temperature fluctuations are underpredicted. Despite these underpredictions, the trend is accurately captured again.



(a) Normalized mean temperature profile.

(b) Normalized temperature fluctuations.

FIGURE 3.8: Temperature statistics for channel flow at $Re_\tau = 150$ with the Neumann boundary condition. Results from LES (■) are compared with DNS (—) (Jaszczur, 2014).

3.2 Topographic Data

3.2.1 Obtaining Raw Data

In order to simulate flows over real, complex topography, a method of importing raw topographic data into the simulation needed to be developed. The first step of this process is to locate a massive database of high resolution data. One of the largest databases of high resolution topographic data is the National Oceanic and Atmospheric Administration (NOAA) which has 1 m resolution topographic data of most coastlines in the continuous United States, in addition to numerous inland sites.

The data can be downloaded using various coordinate systems, but the Universal Transverse Mercator (UTM) coordinate system is the optimal choice for the simulations performed in the present study. The UTM coordinate system divides the globe into 60 distinctive zones, each with 6 degrees of longitude. These sections are then broken into a rectangular grid with a standard Cartesian coordinate system. Since the UTM system is used in lieu of a traditional Mercator system, the distortion is minimal (Hager, Behensky, and Drew, 1989). The UTM is an ideal coordinate system for present studies because it projects the Earth's surface onto a Cartesian grid rather than using the traditional latitude/longitude coordinate systems. While projecting a sphere onto a flat surface results in some distortion, it does not have a large effect on the present study since the domain size is small compared to the global scale.

3.2.2 Calculating Level-Set Function

Once the raw topographic data is obtained, it must be imported into the simulation by means of the level-set function ϕ. As was discussed in Section 2.3.5, the level-set function is used to identify the location of the fluid-solid interface. Since the level-set function is calculated by reading from an array of two dimensional triangles, the raw UTM data files must be converted into a stereolithography (STL) file.

There are numerous software suites capable of generating a mesh that can be converted into an STL file. However, since the data is on a uniform Cartesian mesh, it is easy to generate an STL file by using a simple algorithm. Each data point in the UTM file is uniformly spaced in the horizontal directions. It is possible to loop over the horizontal directions and create two triangles for every square made between grid points. This repeating array of triangles can be seen in the right image in Figure 3.9. This simple algorithm results in a greater number of triangles than an efficient meshing algorithm used in state of the art software. However, the benefit of using this method is that it can be easily parallelized.

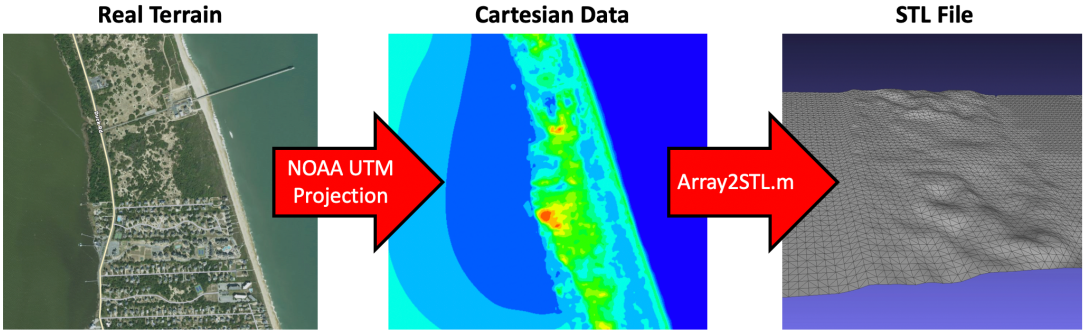


FIGURE 3.9: Generating STL file of topography from Duck, NC. Left image taken from Google Maps.

Improving efficiency of level-set function calculation

Even though the level-set function only needs to be calculated one time during the simulation, efforts can still be made to increase the computational efficiency of this step. Consider a simulation with a domain size of 5 kilometers in both horizontal directions and 1 kilometer in the vertical direction. To run an accurate LES on this domain, a mesh of 512 points in both horizontal directions, and 256 points in the vertical direction would be required. Assuming this simulation were run in parallel on 512 processors, each processor would contain over 10^5 points. Additionally, consider a complex topography with a resolution of 5 meters. This means that the STL file of the surface has 2×10^6 triangles. Since the level-set function is calculated by finding the minimum distance of each point to all of the triangles, the distance calculation would be conducted 2×10^{11} times on each processor.

The reason this process is so computationally expensive is that the fluid domain is parallelized, but each processor has to loop over the entire topographic grid. To vastly improve the efficiency of this step, the large STL file containing the topographic data can be broken into numerous smaller STL files that align with the division of the processors on the fluid domain. This means that each fluid point only searches for the triangles in the local vicinity when calculating the nearest distance. By doing this, the calculations required to calculate the level-set function reduces drastically as the number of processors used in the simulation increases.

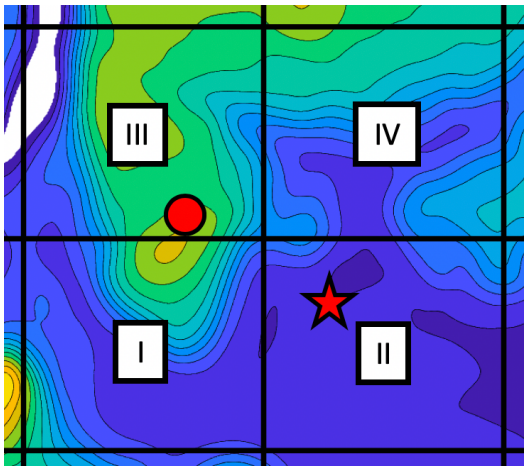


FIGURE 3.10: Elevation map with four smaller regions (I, II, III, IV) representing separate STL files.

There is one important source of error to consider when dividing the topographic STL file into smaller files. Consider the elevation map shown in Figure 3.10. In this figure, the large STL file has been divided into four separate files denoted by regions I-IV. Now consider a fluid point located above the star in region II. By the parallelized searching algorithm, the level-set function at this point will be calculated only by searching the smaller topographic file contained in region II. In this case, the nearest point of the surface will be in region II. However, when calculating the level-set function at a fluid point located above the circle in region III, a source of error may occur. This is because the nearest location of the topography may actually be in region I, since the fluid point may be closer to the top of the hill in region I, rather than the ground directly beneath it in region III. Since the fluid point over the circle is only searching in region III, the level-set function will be inaccurately calculated. To mitigate this source of error, each of the smaller STL files will span a region of the surface which expands into the adjacent regions. This ensures that the level-set function is always accurately calculated.

3.2.3 Verifying Level-Set Function

Verifying the calculation of the level-set function can be performed by analyzing the iso-surface where $\phi = 0$. This corresponds to the location of the air-surface interface in the

simulation. For this particular verification test, topographic data of Elkhorn Slough, a tidal inlet in Monterey Bay, California was imported into the simulation. The isosurface $\phi = 0$ is shown in Figure 3.11. This elevation data was divided into 128 separate STL files, each read by different processors. From qualitative observation of the isosurface, it can be seen that the topography is captured smoothly across all points. Note that in Figure 3.11, the plane representing the sea and tidal inlet surface is not a part of the isosurface $\phi = 0$ as the level-set function is only used to capture the interface between the land and air. The interface between the air and the sea aligns with the boundary of the simulation domain, and thus does not need to be calculated.

3.3 Immersed Boundary Method

3.3.1 Forcing Nodes and Stencil

The first step of the IB method is to determine the forcing nodes used to capture the effect of a solid wall on the fluid, as was discussed in Section 2.3.4. The velocity on these forcing nodes is calculated by using a linear (or logarithmic if wall model is being used, see 3.3.2) interpolation between the nonslip boundary condition on the wall and stencil points

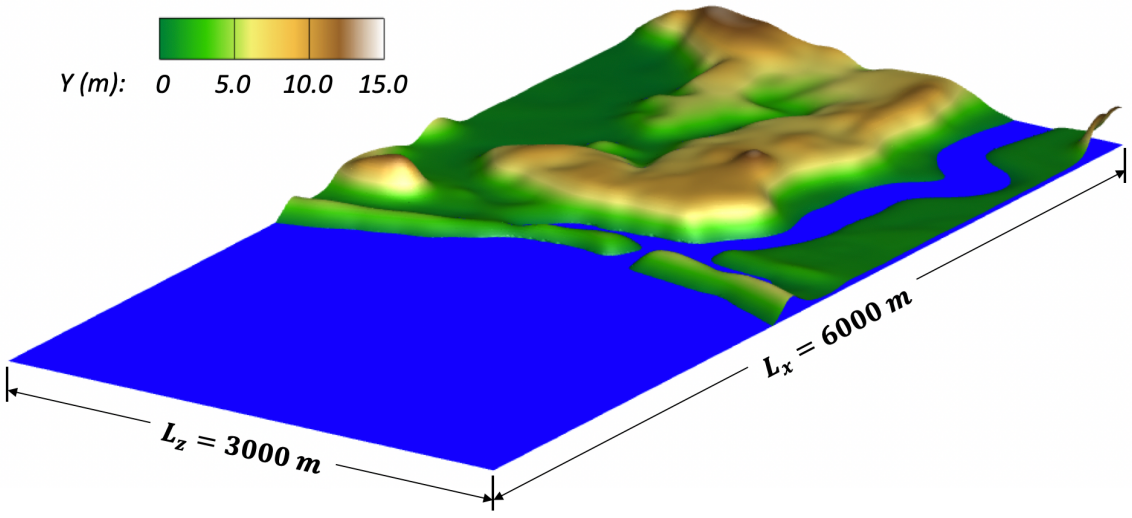


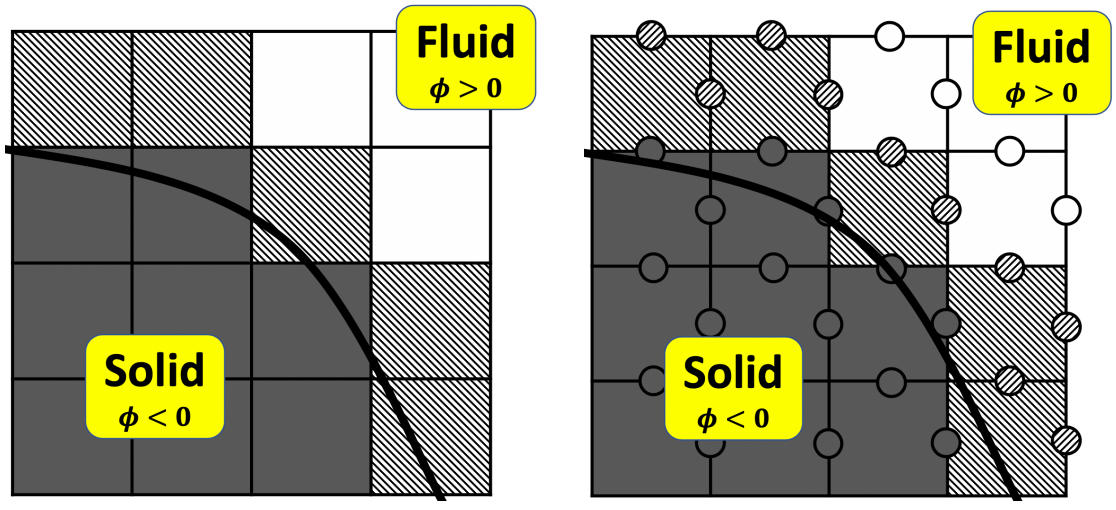
FIGURE 3.11: Isosurface for $\phi = 0$. Vertical direction is stretched by a factor of 15 to better illustrate topography.

which are located further in the fluid domain. Since a staggered grid is being used, these forcing nodes and stencil need to be determined for the grids located at the cell center (temperature, pressure, scalars) and the cell faces (velocity).

Algorithm

The cell-center (pressure) nodes are classified first (Figure 3.12a). If a node is in the solid domain ($\phi < 0$), then it is a solid node. If a node is in the fluid domain ($\phi > 0$), but is directly adjacent to a solid node, then it a forcing node. All other nodes are fluid nodes. Once the pressure nodes are classified, the cell-face (velocity) nodes are classified based off of the adjacent pressure nodes (Figure 3.12b). If a velocity node is directly adjacent to a solid pressure node, then it is a solid node. If a velocity node is between two fluid pressure nodes, it is a fluid node. All other cases are forcing nodes.

Once the forcing nodes have been identified, a stencil of three fluid points (two, if two dimensional simulation) are selected for each forcing node. These stencil points must be chosen such that the normal vector passing from the fluid-solid boundary through the forcing node must pass through the triangle (line segment) between the three (two) stencil



(a) Classification of pressure nodes.

(b) Classification of velocity nodes.

FIGURE 3.12: Solid (grey), forcing (crosshatch), and fluid (white) IB node classification

points. To determine which fluid nodes are to be used as stencils, a searching algorithm is implemented. This algorithm will search the adjacent nodes first and then expand the search bubble until the required number of stencils points are selected. While this method is robust for simple geometries on uniform grids, sources of error appear when more complicated geometries and grids are used.

Sources of error in stencil calculation

When the aspect ratio on a grid is large, an error in stencil point selection can occur. Consider the scenario presented in Figure 3.13. In this diagram, the velocity nodes in the x_2 direction are displayed in circles. If the stencil points for the forcing node ★ are being determined, the algorithm will first search the adjacent nodes; □ and △. Only one of these nodes is a fluid node (□) and selected as the first stencil point. The next nearest node (▽) is then checked and selected as the second and final stencil point. Note that in this context 'nearest' means closest on the grid, not necessarily the closest physically. The problem with using □ and ▽ as the stencil points is that the normal vector does not pass through this line segment. This results in a large error when interpolating the velocity at the forcing node ★. This problem would be mitigated if ◊ was used instead of ▽, however since ▽ is closer on the grid, it is chosen first.

To address this error, a new searching algorithm was implemented to ensure that the normal vector passes through the stencil points. Since this method will be implemented on topography representing the surface, it will be assumed that the vertical component of the normal vector is significantly larger than the other components. Given this assumption, it will be required that at least one of the stencil points be directly above the forcing node. The new algorithm will work in the same way as the previously described searching method, with the additional requirement that at least one stencil point must be directly above the forcing node. Applying this method to Figure 3.13, the searching algorithm will select □ as the first stencil point. It will then check, but reject ▽ since this would result in zero stencil points above ★. The nodes two grid steps away from ★ will then be

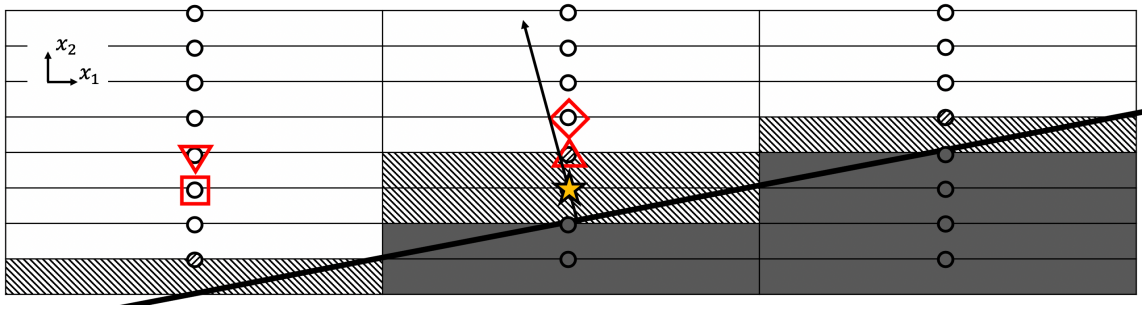


FIGURE 3.13: Potential stencil points for forcing node ★. Traditional searching algorithm will incorrectly identify □ and ▽ as the stencil points

checked and ◊ will ultimately be selected as the second forcing node. While this method has not been checked for an extensive variety of geometries, it has proven sufficient for all topographies being studied in this thesis.

The other error which may become present in stencil selection occurs when the solid geometry is convex, and there are no nearby fluid nodes that can be used as stencil points. However, this problem has not been found to be an issue in present studies, as the slopes of the topography are mild and no convex "pinch points" are present.

3.3.2 Wall Model with IB

In order to simulate the high *Re* flow in the ABL, wall layer models need to be implemented in conjunction with the IB method. The wall model described in this section is based off two similar methods (Roman, Armenio, and Fröhlich, 2009); (Kang, 2015) which use the law of the wall:

$$u^+ = \frac{1}{\kappa} \ln y^+ + B \tag{3.11}$$

Velocity interpolation

One of the simplest methods of implementing a wall model into the IB method is to assume the forcing node is in the log-region and use Equation 3.11 to interpolate the velocity at the forcing nodes. Consider the scenario outlined in Figure 3.14. The first step is to

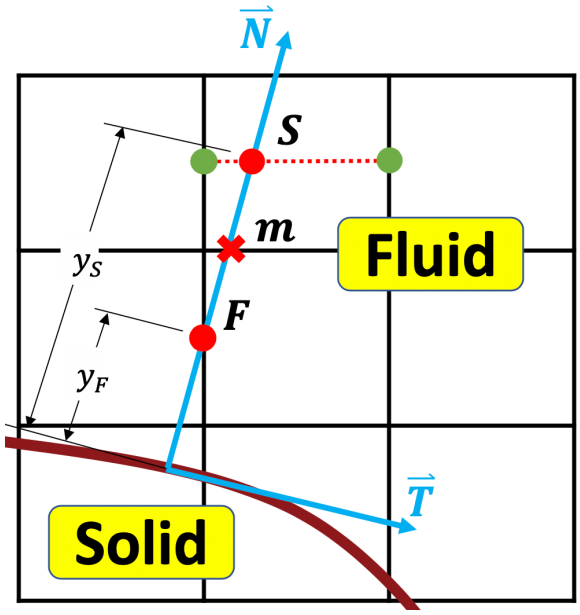


FIGURE 3.14: Schematic for velocity interpolation on IB forcing nodes.

calculate the stencil velocity $\boldsymbol{u}_S$ by interpolating the velocity from the stencil points onto the normal vector. The stencil velocity is then broken into tangential $\boldsymbol{u}_{S,T}$ and normal $\boldsymbol{u}_{S,N}$ components. Next, the stencil velocity $\boldsymbol{u}_S$ is used to calculate the friction velocity u_* by iteratively solving

$$\frac{u_{S,T}}{u_*} = \frac{1}{\kappa} \ln \frac{y u_*}{\nu} + 5.5 \tag{3.12}$$

evaluated at $y = y_S$. The tangential velocity is calculated by subtracting Equation 3.12 evaluated at $y = y_F$ from the same equation evaluated at $y = y_S$.

$$u_{F,T} = u_{S,T} - \frac{u_*}{\kappa} \ln \frac{y_S}{y_F} \tag{3.13}$$

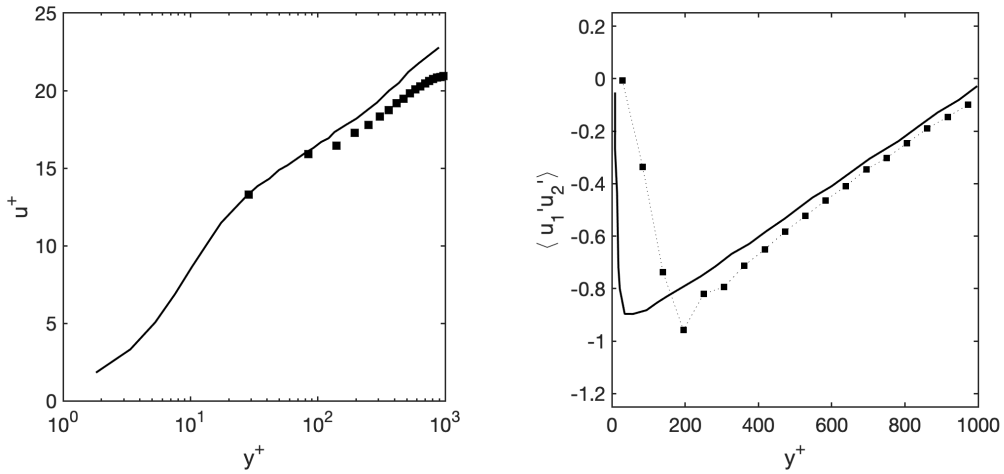
The normal velocity is calculated using a power law as follows

$$u_{F,N} = u_{S,N} \frac{y_F^2}{y_S^2} \tag{3.14}$$

The last step is to convert the velocity components back to the Cartesian mesh.

Validation

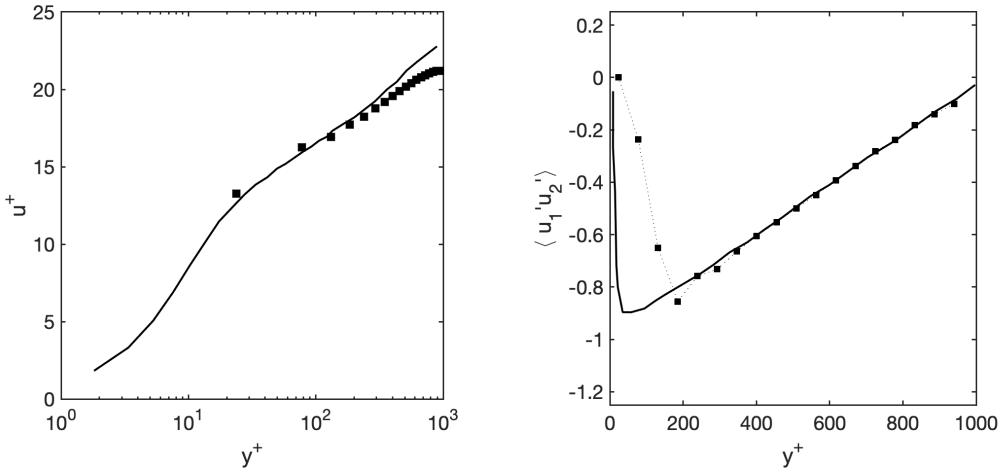
To validate this method, channel flow at $Re_\tau = 1000$ was simulated. Since the goal of this study was to test the effectiveness of the wall model and IB method, the channel was represented by an immersed boundary within the simulation. This is as opposed to simply using the simulation domain boundaries as the walls of the channel. Furthermore, two separate simulations were performed. The first test had the channel walls perfectly aligned with the simulation grid, while the second test had the channel walls offset between the grid spacing. The flow was driven by a constant pressure gradient with periodic boundary conditions in the horizontal directions. The size of the fluid domain was $(L_x, L_y, L_z) = (2\pi, 2, \pi)$ and the grid size was $(N_x, N_y, N_z) = (64, 40, 32)$. Note that while the y-direction has 40 grid points, only 36 are in the fluid domain due to the positioning of the channel walls within the domain. The simulation was run until a statistical steady state was reached. The results for the grid-aligned and grid-offset simulations were compared to DNS results (Alamo et al., 2004) in Figure 3.15 and Figure 3.16, respectively. The LES data was generated from forty time realizations separated by a time of approximately five times the large eddy turnover time.



(a) Normalized mean streamwise velocity.

(b) Normalized plotsof $\sqrt{\langle u_1', u_2' \rangle}$.

FIGURE 3.15: Velocity statistics for channel flow at $Re_\tau = 1000$ with IB **aligned** with grid. WM-LES data (■) are compared with DNS data (—) (Alamo et al., 2004)



(a) Normalized mean streamwise velocity.

(b) Normalized plot of $\sqrt{\langle u_1', u_2' \rangle}$.

FIGURE 3.16: Velocity statistics for channel flow at $Re_\tau = 1000$ with IB **offset** from grid. WM-LES data (■) are compared with DNS data (—) (Alamo et al., 2004).

From observing these figures, it can be seen that for both validation tests, the mean velocity and second order statistics are close to the expected results near the wall, but the error increases in the middle of the channel. This is a result of the 'log-layer mismatch' discussed in Section 2.3.6. The wall shear stress is being underpredicted, which is resulting in the underprediction of the mean streamwise velocity in the center of the channel.

Turbulent viscosity

In order to reduce the error caused by the 'log-layer mismatch', the viscosity at the forcing node is modified to ensure the wall shear stress is accurately calculated. This modification is based off a method developed to simulate channel and pipe flow (Kang, 2015). From the definition of the friction velocity, the wall shear stress is known to be $\tau_w = \rho u_*^2$. The shear stress at point m in Figure 3.14 is calculated using the velocity gradient as

$$\frac{\tau_m}{\rho} = (\nu + \nu_{t,m}) \frac{\partial u_T}{\partial n}\bigg|_m = (\nu + \nu_{t,m}) \frac{u_{S,T} - u_{F,T}}{y_S - y_F} \tag{3.15}$$

where $\nu_{t,m}$ is the SGS turbulent viscosity at point m. Note here that the lowercase subscript t represents a turbulent value, while an uppercase subscript T represents the tangential direction. Using Equation 3.15 and the definition of wall shear stress, the following is obtained

$$(\nu + \nu_{t,m})\frac{u_{S,T} - u_{F,T}}{y_S - y_F} = \beta u_*^2 \tag{3.16}$$

where β represents the distance into the boundary layer (since the forcing node is not on the wall) and is defined as

$$\beta = \frac{y_m}{\delta} \tag{3.17}$$

where δ is the boundary layer thickness. In ideal cases such as channel flow, the boundary layer thickness is known, but for cases where the boundary layer thickness is unknown it can be estimated from prior studies. The turbulent viscosity at m can then be calculated as

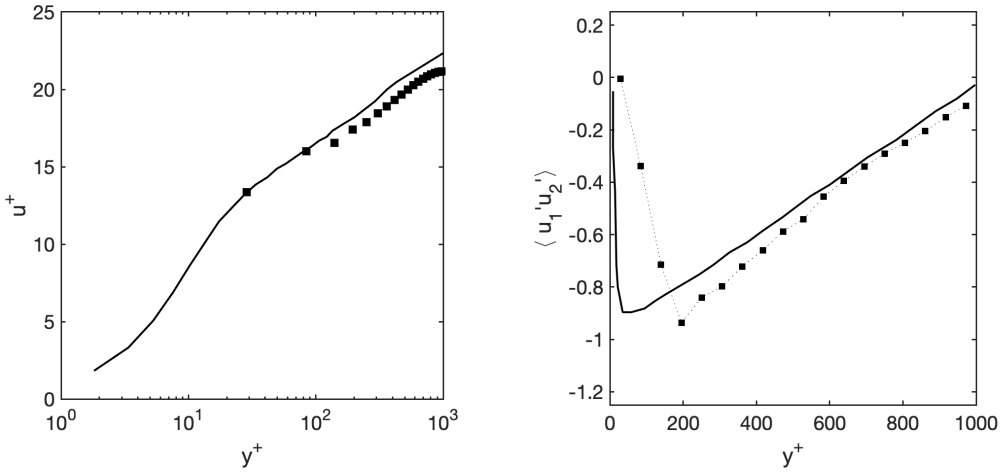
$$\nu_{t,m} = \beta u_*^2 \frac{y_S - y_F}{u_{S,T} - u_{F,T}} - \nu \tag{3.18}$$

Lastly, a linear interpolation is used to calculate the turbulent viscosity at the forcing node

$$\nu_{t,F} = 2\nu_{t,m} - \nu_{t,S} = 2\left(\beta u_*^2 \frac{y_S - y_F}{u_{S,T} - u_{F,T}} - \nu\right) - \nu_{t,S} \tag{3.19}$$

Validation

The validation simulations at $Re_\tau = 1000$ were repeated to ensure the turbulent viscosity implementation improves the capabilities. All of the same case parameters were used, the only difference being the modification of the turbulent viscosity at the forcing nodes $\nu_{t,F}$. The results of this test can be seen in Figure 3.17 and Figure 3.18. From analysis of these figures, it can be seen that the mean streamwise velocity profile is closer to the expected DNS results. In particular the turbulent viscosity improves the results of the mean profile in the center of the channel. This is because both the velocity profile and the



(a) Normalized mean streamwise velocity.

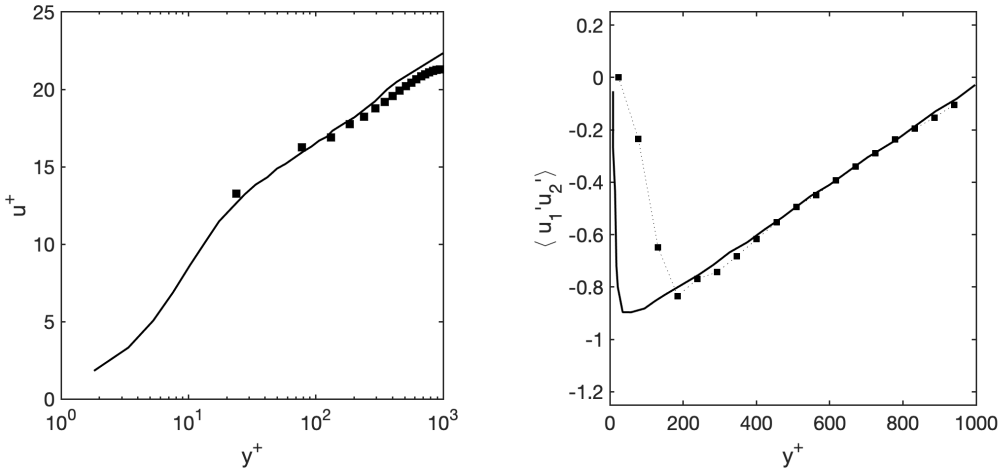
(b) Normalized plot of $\sqrt{\langle u_1', u_2' \rangle}$.

FIGURE 3.17: Velocity statistics for channel flow at $Re_\tau = 1000$ with IB **aligned** with grid and **turbulent viscosity** method implemented. WM-LES data (■) are compared with DNS data (—) (Alamo et al., 2004).

wall shear stress are being accurately represented using this method. From the second-order statistics, it can be seen that the results are still not completely aligned with the DNS results, particularly near the wall and in the center of the channel. This is likely due to the difficulty of fine tuning the parameters, which will be discussed in the next section.

Comment on parameter selection

Through the process of developing and implementing the wall model, it was found that the results are largely dependent on the method used to calculate the SGS turbulent viscosity. The results presented in this paper were generated using the original Smagorinsky model (Section 2.3.3) with a coefficient of $C_s = 0.07$. This is similar to the value of $C_s = 0.065$ used in the literature off of which this method is based (Roman, Armenio, and Fröhlich, 2009). It was found that the changes in mean profile caused by using different coefficients were even more significant when the wall model was implemented. Furthermore, the results determined when using the dynamic Smagorinsky model (Section 2.3.3) in the simulations were found to be more inaccurate and dependent on the simulation parameters. Further



(a) Normalized mean streamwise velocity.

(b) Normalized plot of $\sqrt{\langle u_1', u_2' \rangle}$.

FIGURE 3.18: Velocity statistics for channel flow at $Re_\tau = 1000$ with IB **offset** from grid and **turbulent viscosity** method implemented. WM-LES data (■) are compared with DNS data (—) (Alamo et al., 2004).

analysis on the selection of parameters in the SGS models would likely improve the results of these validation tests.

3.4 Inflow Boundary Condition

Since the ABL simulations are over a heterogeneous topography, conducting a periodic simulation is not possible. For this reason, a turbulent velocity and temperature field must be used as the inflow condition for the simulation. These fields must satisfy various requirements in order for accurate results to be simulated. These fields must: (1) be efficiently generated to prevent excessive computational cost, (2) represent a realistic shear flow field encountered in the ABL, and (3) be customizable in order to capture various conditions (i.e. mean profile and turbulence intensity). The synthetic turbulence generation method commonly used to study wind farms (Mann, 1998) will be implemented to ensure all three of these criteria are attained.

3.4.1 Generating Inflow Velocity

The method of synthetically generating a velocity field follows three steps: (1) generate three dimensional homogeneous 'turbulence box', (2) convert homogeneous flow field to shear flow field, and (3) convert three dimensional spatial field to two dimensional temporal field.

(1) Generate three dimensional homogeneous turbulence

The basis of generating a three dimensional turbulent flow field is to integrate over the spectral tensor. This is done by integrating a generalized stochastic Fourier-Stieltjes integral over the wavenumber space

$$\boldsymbol{u}(\boldsymbol{x}) = \int e^{i\boldsymbol{k}\cdot\boldsymbol{x}} d\boldsymbol{Z}(\boldsymbol{k}) \tag{3.20}$$

This is discretized into the form

$$u_i(\boldsymbol{x}) = \sum_k e^{i\boldsymbol{k}\cdot\boldsymbol{x}} C_{ij}(\boldsymbol{k}) n_j(\boldsymbol{k}) \tag{3.21}$$

where $\boldsymbol{k}$ are the wavenumbers, $n_j(\boldsymbol{k})$ are stochastic, Gaussian, complex variables which represent the 'randomness' in turbulence, and $C_{ij}(\boldsymbol{k})$ are coefficients based off the spectral tensor which need to be determined. Note that in Equation 3.21, the i in the exponent represents $\sqrt{-1}$, which is different from the i index that is being used in the other variables to represent direction. Through analysis of isotropic turbulence and the spectral tensor, the coefficients $C_{ij}(\boldsymbol{k})$ were determined to be (Batchelor, 1953)

$$C(\boldsymbol{k}) = \frac{\sqrt{2}\pi\Big(E\big(||\boldsymbol{k}||\big)\Big)^{1/2}}{V||\boldsymbol{k}||^2} \begin{bmatrix} 0 & k_3 & -k_2 \\ -k_3 & 0 & k_1 \\ k_2 & -k_1 & 0 \end{bmatrix} \tag{3.22}$$

where V represents the volume of the turbulence box $(L_1 \times L_2 \times L_3)$, $E\big(||\boldsymbol{k}||\big)$ is the energy spectrum, $||\cdot||$ represents the L_2 norm, and $k_i = \frac{2\pi}{L_i}$. The energy spectrum is taken to be (Karman, 1948)

$$E\big(||k||\big) = \alpha\epsilon^{2/3}L^{5/3}\frac{L^4||\boldsymbol{k}||^4}{\Big(1+L^2||\boldsymbol{k}||^2\Big)^{17/6}} \tag{3.23}$$

The coefficients in this equation are determined based off of the Kaimal spectra (Kaimal and Finnigan, 1994)

$$\begin{aligned} L &= 0.59y_o \\ \alpha\epsilon^{2/3} &= 3.2\frac{u_*^2}{y_o^{2/3}} \end{aligned} \tag{3.24}$$

where y_o and u_* are the roughness length and friction velocity. These parameters can be selected based off of measurement or simulation data (discussed in Section 3.4.3).

(2) Converting homogeneous flow field to shear flow field

Solving for the velocity field using Equation 3.21 does not provide a sufficient velocity field because this field assumes homogeneous turbulence, whereas flow in the ABL is characterized by large velocity gradients and shear stresses. In order to convert this homogeneous flow field into a flow field with shear, a linear transformation is applied to the coefficient matrix C_{ij}

$$C_{ij}^s = A_{ik}C_{kj} \tag{3.25}$$

where C_{ij}^s is the shear adjusted coefficient matrix and A_{ik} is the transformation matrix. In order to determine the value of A_{ik}, the rapid distortion theory of turbulence is implemented. This theory describes the distortion of turbulence when it is subjected to a rapid perturbation (Batchelor and Proudman, 1954). This theory was later applied to study the structure of turbulence when subjected to rapid shear (Pearson, 1959) and slower developing shear flows (Lee, Kim, and Moin, 1990). Solving the rapid distortion theory equations

for shear flow results in a transformation matrix defined by

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & \zeta_1 \\ 0 & 1 & \zeta_2 \\ 0 & 0 & ||\boldsymbol{k}_o||^2/||\boldsymbol{k}||^2 \end{bmatrix} \tag{3.26}$$

with $\boldsymbol{k}_o = (k_1, k_2, k_{3o})$, and $k_{3o} = k_3 + \beta k_1$. For these equations, β is a non-dimensional time scale defined as

$$\beta = \Gamma(kL)^{-2/3} \tag{3.27}$$

The coefficients are defined as

$$\begin{aligned} \zeta_1 &= C_1 - \frac{k_2}{k_1} C_2 \\ \zeta_2 &= \frac{k_2}{k_1} C_1 + C_2 \end{aligned} \tag{3.28}$$

with

$$\begin{aligned} C_1 &= \frac{\beta k_1^2 (||\boldsymbol{k}_o||^2 - 2k_{3o}^2 + \beta k_1 k_{3o})}{||\boldsymbol{k}||^2 (k_1^2 + k_2^2)} \\ C_2 &= \frac{k_2 ||\boldsymbol{k}_o||^2}{(k_1^2 + k_2^2)^{3/2}} \arctan\left[\frac{\beta k_1 (k_1^2 + k_2^2)^{1/2}}{||\boldsymbol{k}_o||^2 - k_{3o} k_1 \beta}\right] \end{aligned} \tag{3.29}$$

The only parameter required is $\Gamma = 3.9$. For a more in depth description of the derivations of these equations, see the original paper (Mann, 1998). The transformation matrix A_{ij} can be used to calculate C_{ij}^s. Then a three dimensional box of shear turbulence using can be generated by solving Equation 3.21 and replacing C_{ij} with C_{ij}^s.

(3) Convert three dimensional spatial field to two dimensional temporal field

Once a three dimensional turbulence box has been generated, it can be converted to a time series of two dimensional velocity slices which can be used as the inflow boundary condition. Assuming Taylor's frozen turbulence hypothesis (Taylor, 1938) holds, the two dimensional velocity slices can be calculated as

$$\boldsymbol{u}_{\text{inflow}}(x_2, x_3, t) = \boldsymbol{u}(U(x_2)t, x_2, x_3) \tag{3.30}$$

where x_2 is the vertical direction, x_3 is the spanwise direction, and $U(x_2)$ is the mean profile of the streamwise velocity.

3.4.2 Generating Temperature Inflow

The inflow boundary condition for the temperature equation is determined by decomposing the velocity and temperature fields into mean and fluctuating components

$$||\boldsymbol{u}_{\text{inflow}}(x_2, x_3, t)|| = U(x_2) + u'(x_2, x_3, t) \tag{3.31}$$

$$\theta_{\text{inflow}}(x_2, x_3, t) = \Theta(x_2) + \theta'(x_2, x_3, t) \tag{3.32}$$

where u' is the magnitude of the velocity fluctuations, $\Theta(x_2)$ is the mean temperature profile, and θ' is the temperature fluctuations. The mean temperature profile is determined from either field measurements or mesoscale simulations. The temperature fluctuations are calculated as

$$\theta'(x_2, x_3, t) = Pr \cdot u'(x_2, x_3, t) \tag{3.33}$$

based off of experimental findings that there is a strong correlation between velocity and temperature fluctuations in boundary layers (Antonia, Krishnamoorthy, and Fulachier, 1988).

3.4.3 Determining Inflow Condition from Measurements and Simulations

From the methodology described in Section 3.4.1 there are numerous parameters that must be obtained in order to generate the velocity and temperature fields used as the inflow condition. The mean profile of the streamwise velocity is assumed to be a logarithmic function of the form

$$U(x_2) = \frac{u_*}{\kappa} \log \frac{x_2}{y_o} \tag{3.34}$$

In order to prescribe a velocity profile $U(x_2)$, time averaged velocity data must be obtained and fit to a logarithmic function by determining the friction velocity u_* and aerodynamic

surface roughness y_o. Depending on the surface, the surface roughness can be determined in different ways. For instance, over the sea it has been shown that the friction velocity and aerodynamic surface roughness are related through the relationship (Charnock, 1955)

$$y_o = \beta \frac{u_*^2}{g} \tag{3.35}$$

where β is the Charnock constant and takes various values depending on the sea surface. In the open sea $\beta = 0.01$, while in the surf-zone (region of breaking waves) $\beta = 0.11$. By contrast, over land the aerodynamic surface roughness is often defined based on roughness characteristics such as vegetation and urban development (Nicholas and Lewis, 1980). Since a relationship for the roughness length is known, the only parameter used to fit the data to the curve is the friction velocity. Once the roughness length and friction velocity are known, the remaining parameters used in this method $(L, \alpha\epsilon^{2/3})$ can be solved using Equation 3.24.

Prescribing the inlet temperature profile $\Theta(x_2)$ is conducted similarly to the velocity profile. The difference is that the temperature profile is fit to the following logarithmic function

$$\Theta(x_2) = \theta_s + \frac{\theta_*}{\kappa} \log \frac{x_2}{y_T} \tag{3.36}$$

where θ_s is the surface temperature, y_T is the thermal roughness length, and θ_* is the turbulent temperature scale at the surface defined as $\theta_* = -\langle \theta' u_2' \rangle / u_*$. The thermal roughness length is taken to be $y_T = 0.1 y_o$ (Calaf, Parlange, and Meneveau, 2011). The turbulent temperature scale can be taken from measurement data, which leaves the surface temperature as the parameter being used to fit the vertical temperature data to the curve.

The data that is used for this fit can be obtained either through field measurements or mesoscale simulations. One of the difficulties in using data from field measurements is that it is difficult to measure the mean velocity and temperature of a flow field at positions far above the ground. The height of the ABL can potentially be on the order of kilometers

so it is difficult to get accurate measurements near the top of the ABL. One way to avoid this problem is to obtain data from mesoscale simulations. The benefit of using mesoscale simulations to fit the mean profile is that it is possible to obtain values at all heights of the simulation. This also provides a link to couple two simulations of varying scales, if a coupled mesoscale-LES simulation was ever implemented.

3.4.4 Verification

The method of generating a turbulent velocity and temperature field was verified by generating the fields with the following parameters

$$u_* = 0.13\frac{m}{s}, \quad \theta_s = 291.9\ K, \quad \theta_* = 0.046\ K \tag{3.37}$$

Instantaneous velocity and temperature figures can be seen in Figure 3.19. In order to ensure the accuracy of the velocity and temperature fields, the mean streamwise velocity (averaged in spanwise direction) and temperature from one time step are compared to the expected profile in Figure 3.20. It can be seen that mean inflow boundary condition accurately captures the expected profile.

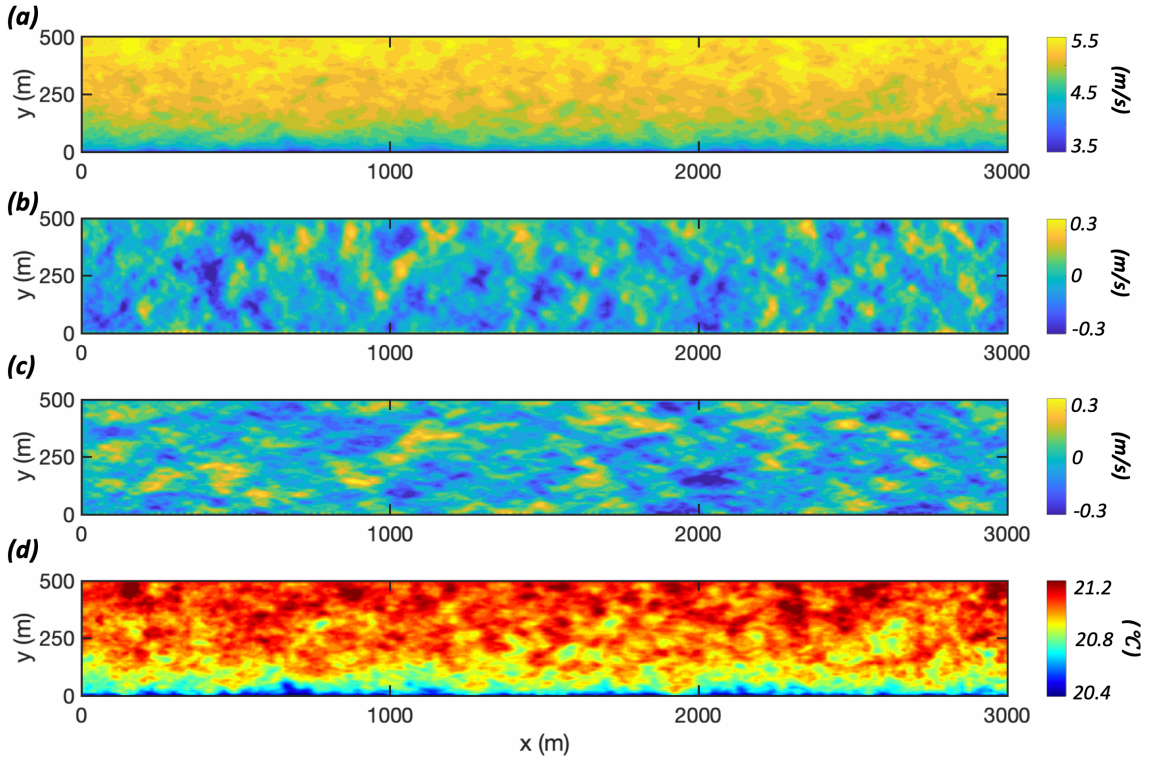
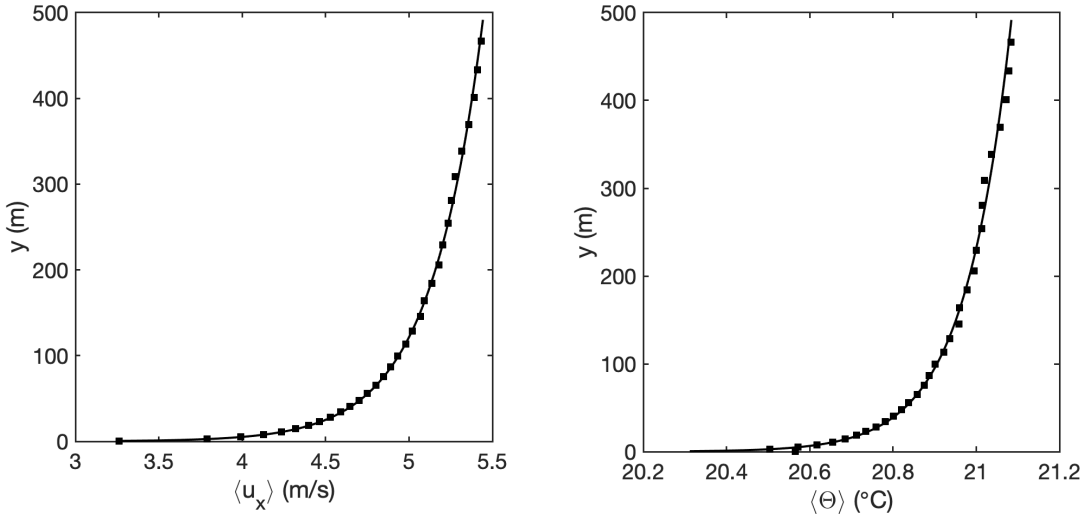


FIGURE 3.19: Slices of instantaneous (a) streamwise velocity, (b) vertical velocity, (c) spanwise velocity, and (d) temperature used as inflow condition.



(a) Mean streamwise velocity profile.

(b) Mean temperature profile.

FIGURE 3.20: Profiles of inflow boundary condition at a single time step, averaged in the spanwise direction (■), compared to expected values (—).

Chapter 4

Full Scale Simulation

After validating all the various capabilities necessary to simulate the ABL (Chapter 3), a full scale simulation was performed and compared to realistic field measurements and simulations.

4.1 Case Setup

The site selected for this study was Elkhorn Slough in Monterey Bay, California. This site was selected due to its complicated topography (tidal inlet and hills). Furthermore, field measurement studies (MacMahan, 2017) and previous numerical studies (Yang et al., 2018) provided results that the simulation could be compared to and validated with. Furthermore, this site frequently has an onshore mean wind direction. This makes it simpler to prescribe inlet boundary conditions since the inlet is over the relatively homogeneous sea surface.

Simulation Domain

The size of the simulation domain is $(L_x, L_y, L_z) = (6000\ m,\ 500\ m,\ 3000\ m)$ where x is the streamwise direction (perpendicular to the beach), y is the vertical direction, and z is the spanwise direction (parallel to the beach). The grid size is $(N_x, N_y, N_z) = (512, 128, 256)$ with a uniform grid in both horizontal directions and stretched grid in the vertical direction (more refined near surface). This corresponds to a resolution of approximately twelve meters in the horizontal directions and 1 meter in the vertical direction near the surface. The velocity vector is defined as $\boldsymbol{u} = u\boldsymbol{i} + v\boldsymbol{j} + w\boldsymbol{k}$. The simulation domain can be seen in Figure 4.1.

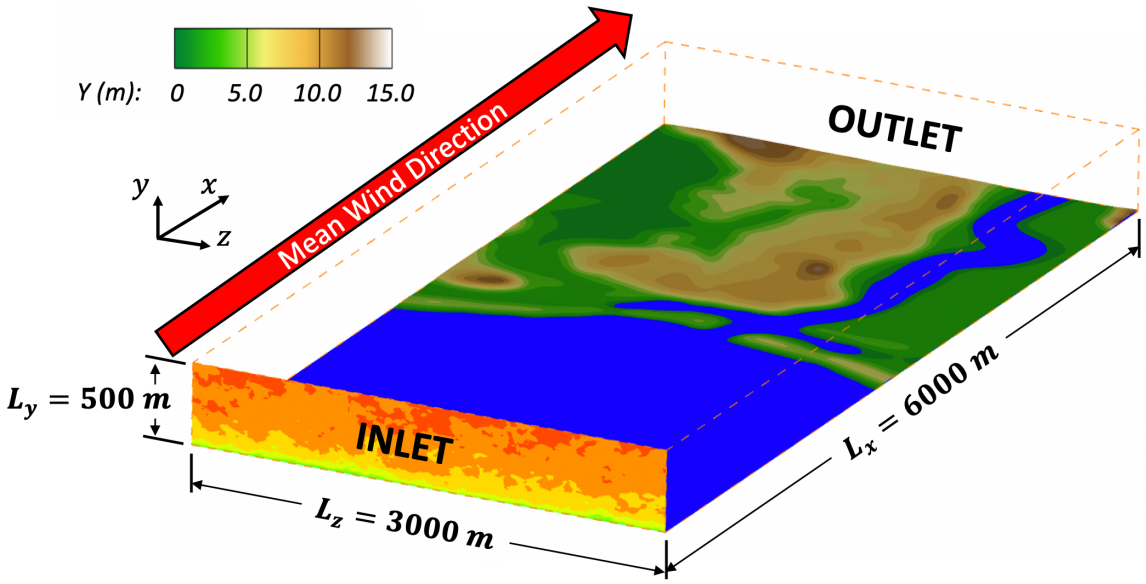


FIGURE 4.1: Simulation domain.

Governing Equations

The filtered governing equations (2.12-2.14) are numerically solved. The Reynolds number is defined as

$$Re = \frac{u_* L_y}{\nu_{air}} = 3.3 \times 10^6 \tag{4.1}$$

where u_* is the friction velocity at the inlet of the domain based off measurements (MacMahan, 2017). The Prandtl number is taken as $Pr = 0.71$, and the Froude number is taken as $Fr = 0.014$. The simulation is advanced using a RK2 method (Section 2.3.2 & Appendix A). For the momentum equation, the convective term is discretized using the QUICK Scheme (Section 2.3.2), and the diffusive term is discretized using central differencing. For the temperature equation, the convective term is discretized using an upwinding TVD scheme (Section 3.1.2), and the diffusion term is discretized using central differencing. The SGS heat flux and shear stress are calculated using the dynamic Smagorinsky model (Section 2.3.3) and discretized using central differencing.

Momentum Boundary Conditions

The inlet boundary condition is calculated by generating synthetic turbulence using the method described in Section 3.4.1. For this particular case, the friction velocity is selected to be $u_* = 0.11\ m/s$ based on field measurements (MacMahan, 2017). To calculate the outlet boundary condition, the convective condition is implemented as

$$\frac{\partial u_i}{\partial t} + U_{mean}\frac{\partial u_i}{\partial x} = 0 \tag{4.2}$$

where U_{mean} represents the average velocity at the outlet. The average outlet velocity U_{mean} is calculated and discretized as

$$\begin{aligned} U_{mean} &= \frac{1}{L_y L_z}\int_0^{L_y}\int_0^{L_z} u(L_x, y, z)dzdy \\ &= \frac{1}{L_y L_z}\sum_{j=1}^{N_y}\sum_{k=1}^{N_z}\left(u\Delta y\Delta z\right)\Big|_{(N_x,j,k)} \end{aligned} \tag{4.3}$$

This method ensures that both mass and momentum are conserved at the outlet of the domain.

To calculate the lower boundary conditions, two separate models are used depending on if a grid point is located over the sea or over the land. For points located over the sea (subscript 'ss'), the shear stress is prescribed as (Piomelli and Balaras, 2002); (Yang et al., 2014).

$$\tau_{xy} = -\rho u_{ss}\sqrt{(u_{ss}^2 + w_{ss}^2)}\left[\frac{\kappa}{\log(y_{ss}/y_o)}\right]^2 \tag{4.4}$$

$$\tau_{zy} = -\rho w_{ss}\sqrt{(u_{ss}^2 + w_{ss}^2)}\left[\frac{\kappa}{\log(y_{ss}/y_o)}\right]^2 \tag{4.5}$$

Recall from Equation 3.35 that y_o is the aerodynamic surface roughness and when located over the sea, it calculated from the friction velocity u_* and Charnock constant β. The impermeable condition is set for the vertical velocity ($v_{ss} = 0$). The boundary condition for points located above the land (i.e IB forcing nodes) is determined by using the wall

model discussed in Section 3.3.2. Lastly, the free slip boundary condition is used for the spanwise and top surfaces.

Temperature Boundary Conditions

The inlet boundary condition for the temperature equation is calculated using the synthetic turbulence generation method from Section 3.4.2. For this case, the parameters are set to $u_* = 0.11\ m/s$, $\theta_s = 291.0\ K$, and $\theta_* = 0.046\ K$. This corresponds to a stably stratified boundary layer. The outlet boundary condition was set using the convective condition

$$\frac{\partial \theta}{\partial t} + U_{mean}\frac{\partial \theta}{\partial x} \tag{4.6}$$

where U_{mean} is calculated using Equation 4.3.

Similar to the momentum equation, the lower boundary condition is calculated in one of two ways depending on a grid point being over sea or land. For points located over the sea, the heat flux q_s'' is prescribed based off of the M-O similarity theory (Calaf, Parlange, and Meneveau, 2011)

$$q_s'' = \frac{-\kappa^2(\theta_{ss} - \theta_s)\sqrt{u_{ss}^2 + w_{ss}^2}}{\log(y_{ss}/y_o)\log(y_{ss}/y_T)} \tag{4.7}$$

where the subscript 'ss' represents the first grid point above the sea surface. For points located over the land (IB forcing nodes), a wall model interpolation is used

$$\theta_{IB} = \theta_s + \frac{\theta_*}{\kappa}\log(y_{IB}/y_T) \tag{4.8}$$

where subscript y_{IB} is the normal distance from the land to the IB forcing nodes. Lastly, an adiabatic boundary condition is used for the spanwise and top surfaces.

4.2 Results

The purpose of running a full-scale simulation based off measurement data was to answer the following three questions:

- How important is the influence of topography on the flow field?
- How do simulation results compare to other simulation and field measurements?
- What are the aspects of this simulation that still need improvement?

In order to address these questions, the simulation was run until a steady state was obtained. The mean velocity and turbulence statistics were then calculated over sixty realizations separated by approximately one minute. The mean streamwise velocity and temperature can be seen in Figure 4.2 and Figure 4.3, respectively.

How important is the influence of topography?

One of the most unique features of this simulation is that the topography is captured directly by the IB method rather than roughness modeling. However, one drawback of implementing this method is that it results in a more expensive computation. This increase

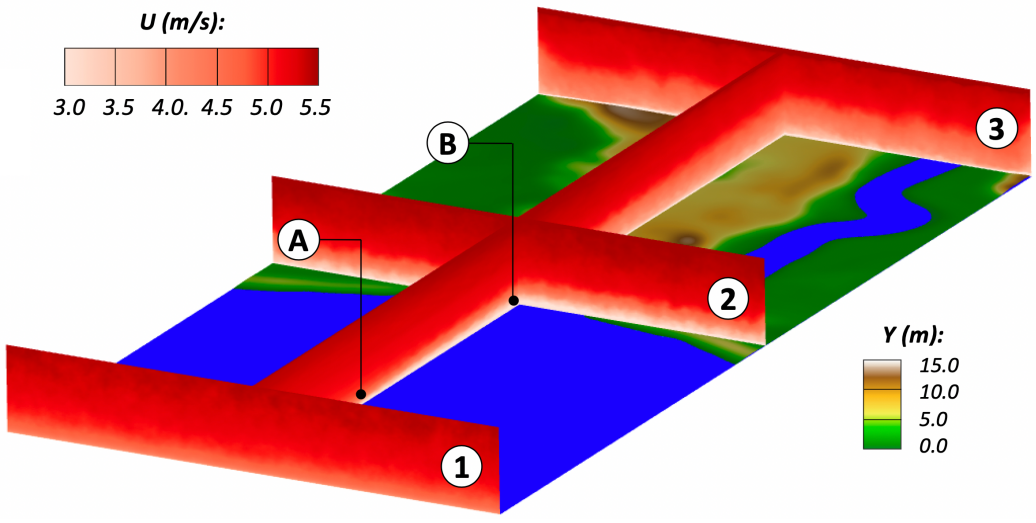


FIGURE 4.2: Time averaged streamwise velocity.

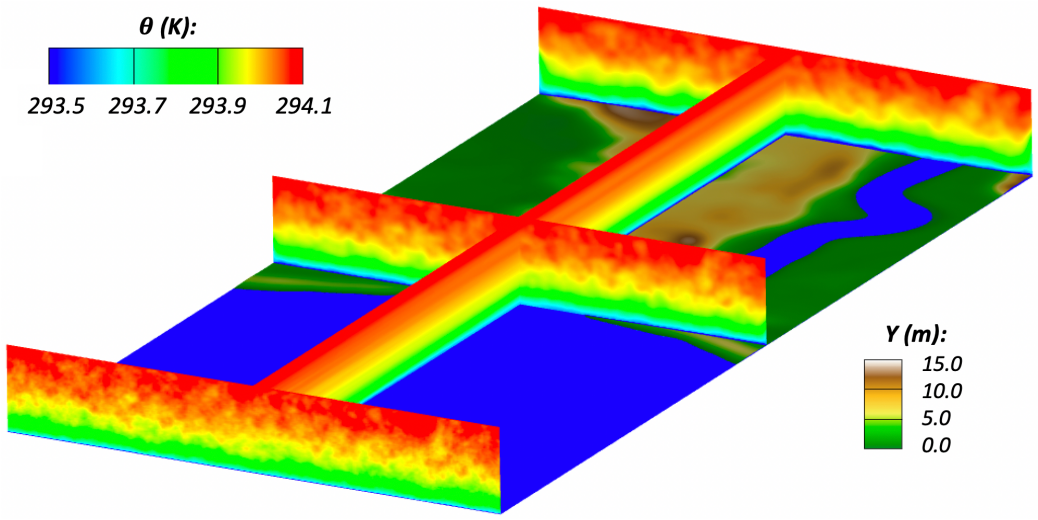


FIGURE 4.3: Time averaged temperature distribution.

in computational cost is acceptable provided the simulation generates a more accurate and detailed flow field. In order for this compromise to be accepted for the present LES study, the topography must be shown to have a tangible effect on the flow field.

The influence of topography on the simulation domain can be shown through analysis of the turbulence statistics of the flow field. First, the Reynolds stress is analyzed in Figure 4.4. The Reynolds stress is plotted as a function of the spanwise direction z at the inlet, coastline, and outlet (Planes 1, 2, and 3 in Figure 4.2, respectively). It can be seen that the magnitude of fluctuations at the inlet and coastline are similar. However, the Reynolds stress has a greater magnitude and variability at the outlet of the simulation. This indicates that the topography is enhancing the turbulence mixing. Similar phenomena have been observed in simulations of channel flow with ribs and surface roughness (Breuer et al., 2009); (Fang et al., 2017) as well as simulations of the ABL with real topography (Yang et al., 2018).

The temperature statistics can also be analyzed to further demonstrate the effect of topography on the results. One such statistic is the turbulent heat flux $\langle \theta' v' \rangle$. Figure 4.5 shows a contour plot of the turbulent heat flux as it passes over a dune. The magnitude of the turbulent heat flux is greatest in the region directly behind the dune. This phenomenon has

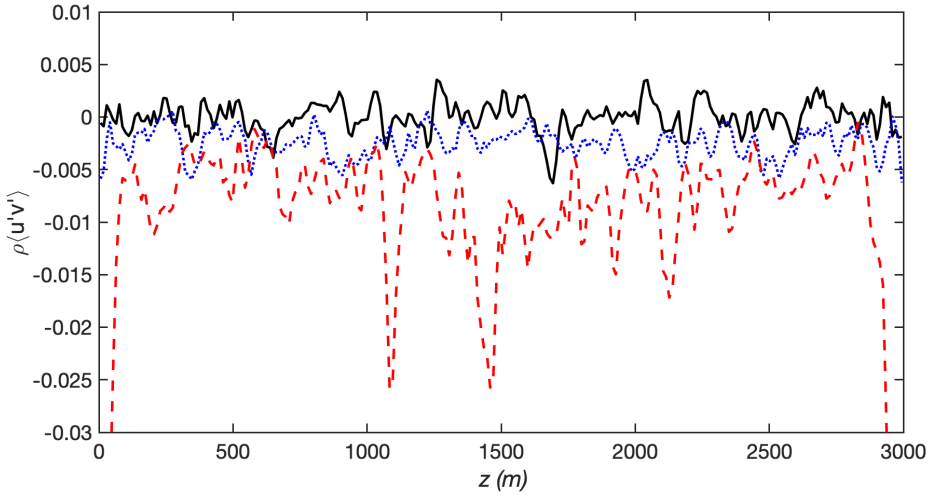


FIGURE 4.4: Reynolds stress of inlet A (—), coastline B (· · ·), and outlet C (- - -) at an elevation of 100 meters.

been observed in other simulations of the ABL with real topography (Yang et al., 2019). While these results are not conclusive indicators of the superiority of LES with complex topography, they do indicate that the topography has significant impacts on the flow field. Further studies are needed to quantify the accuracy gained by capturing the surface rather than modeling with roughness.

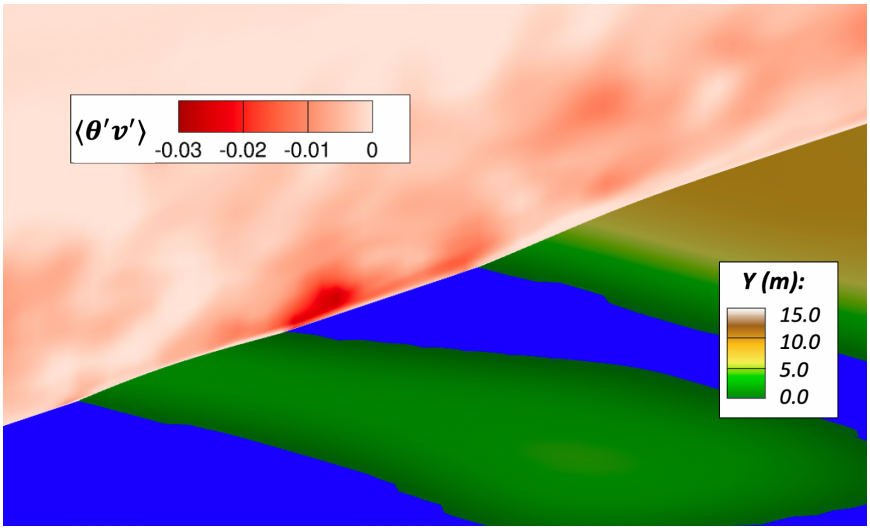


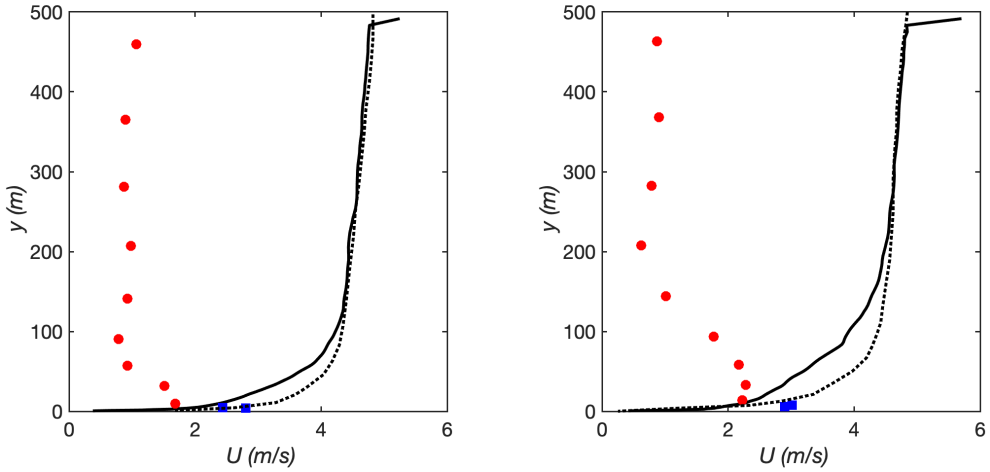
FIGURE 4.5: Turbulent heat flux in region behind dune on coastline. Wind direction is from left to right.

How do simulation results compare to other simulations and measurements?

Now that it has been shown that the introduction of complex topography has a tangible effect on the flow field, the next step is to determine the accuracy of these results. The site was chosen because both measurement and numerical results exist to make comparisons with (Yang et al., 2018). The results of the present LES are compared to three data sets: mesoscale simulations performed using the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS), measurement data taken from near the surface, and LES. Measurements were taken at two sites in the domain. The first site (A in Figure 4.2) was taken over the sea in the surf zone. The second site (B in Figure 4.2) was taken at the coast where the transition from surf zone to complex topography has a strong effect on the flow field.

In Figure 4.6, the vertical profiles of the mean streamwise velocities at the two measurement sites are plotted. It can immediately be seen that there is a discrepancy between the mesoscale simulation (COAMPS) and both LES results. The mesoscale simulation fails to capture the expected log-law region in the boundary because the M-O similarity theory fails in the heterogeneous transition from surf zone to complex topography. The velocity fields of both LES simulations are more similar to the measurement results. Since the velocity field predicted by COAMPS is inaccurate, for subsequent analysis the results of the prior LES will be taken as the truth to which the present simulation results can be compared. The COAMPS data will be used simply as a comparison to see the improvement of implementing LES over mesoscale simulation.

At both analysis locations in Figure 4.6, the mean profile is accurately captured far away from the wall and the top of the domain. However, there are deviations from the expected LES near the sea surface and top of the domain. Near the sea surface, the velocity is underpredicted. This deviation in velocity is larger over the coast (Figure 4.6b) than over the surf zone (Figure 4.6a). Additionally, there is an unrealistic jump in velocity at the top of the domain. Similar to the bottom surface, this discrepancy is worse at the coast than in



(a) Vertical profile at location A (surf zone).

(b) Vertical profile at location B (coast).

FIGURE 4.6: Mean streamwise velocity profiles of present LES (—) are compared to COAMPS (●), Field Measurements (■), and LES (· · ·) (Yang et al., 2018).

the surf zone. These two discrepancies appear to be the biggest difference between current LES and previous studies. Proposed methods of mitigating these errors will be discussed in the following section.

What are the aspects of this simulation that still need improvement?

The underprediction of velocity in the near wall region of both the surf zone and coast indicate errors with the roughness model used over the sea surface. Recall from the case setup that the sea surface is treated as a flat surface, and the roughness caused by surface waves is modeled using Equations 4.4 & 4.5. Consider the region of error to be the vertical distance where present LES deviates from the previously performed LES (shaded box in Figure 4.7). The region of error is larger for points that are further away from the inlet. This is similar to a boundary layer that may develop as a flow moves to a region of increased surface roughness. The velocity profile that would be expected for the current surface roughness model does not match the inlet profile. For this reason, a boundary layer develops in the simulation starting at the inlet. In order to improve the performance of the simulation, a more in depth study into the effect of the roughness model parameters

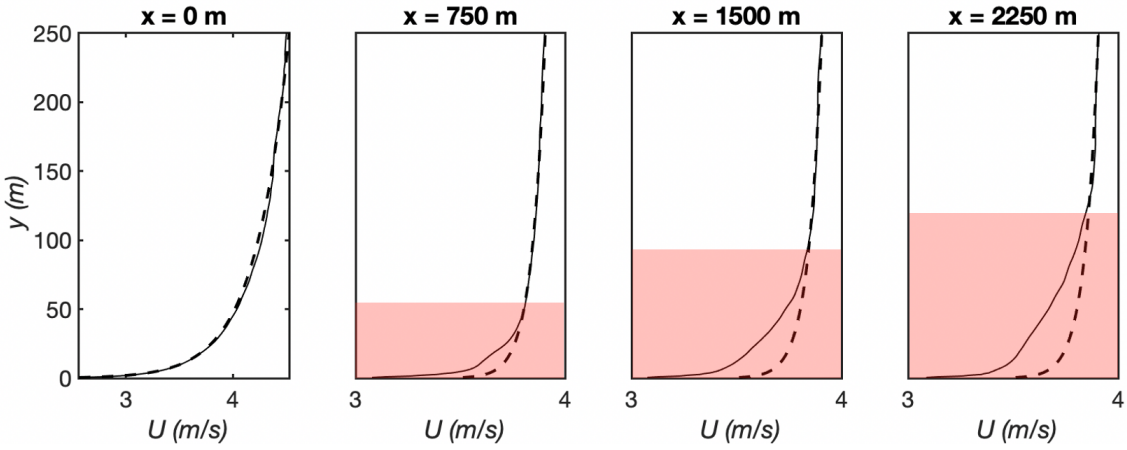


FIGURE 4.7: Comparison of present LES (—) with expected profile (- -) at various streamwise locations. Shaded box indicates region of error.

on the results will need to be conducted.

The velocity overprediction at the top of the domain may at first appear to be a result of an incorrect boundary condition imposed at the top of the domain. However, understanding this overprediction of the velocity requires an analysis of the complete domain. Consider the two dimensional control volume in Figure 4.8. For a control volume of an incompressible fluid (density changes are small compared to mass flux) conservation of mass can be written as

$$0 = \oint_{CS} \boldsymbol{u} \cdot \boldsymbol{n} dA \tag{4.9}$$

Since the boundary condition is free slip on the top surface (2) and no-slip on the bottom surface (3), the mass flux going through the inlet (1) and outlet (4) must be equal. Since the surface roughness model is resulting in the underprediction of velocity in the near wall region, this loss in mass flux at the outlet must be accompanied by an increase in velocity somewhere at the outlet. This is accounted for at the top of the domain where the velocity increases proportionally to the underprediction in the lower velocity. While the control volume in Figure 4.8 is in two dimensions, this same analysis holds for the three dimensions of present LES because the spanwise boundaries are also free slip.

Another area for improvement in present simulation is in the IB wall model. Due to the

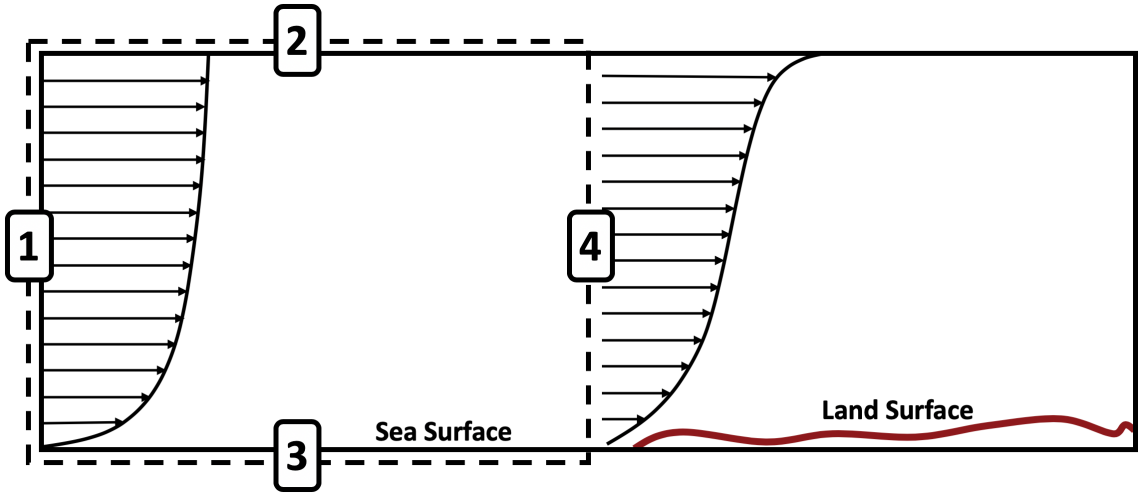


FIGURE 4.8: Control volume over sea surface. Surfaces 2 and 3 are free slip and no slip, respectively.

lack of field measurements on the land side of the domain, it was not possible to assess the performance of the LES beyond the measurement site at the coast. In order for a more in depth study of the accuracy of the LES to be conducted, more extensive measurement data over the land are required. This would allow for quanititative analysis of the effect that particular topographic features have on the performance of the LES.

Furthermore, the LES could be improved by conducting a study with temperature measurements and simulation. The measurement studies of Elkhorn Slough only obtained data on the velocity and turbulence statistics. Being able to validate the LES against temperature measurements and simulation would provide more tools to validate the fidelity of this simulation.

Chapter 5

Conclusions and Recommendations

5.1 Thesis Contributions

The primary contribution of this thesis is to explore the feasibility of using LES to study the effect of complex topography on the ABL. The end goal of these studies is to eventually be able to advance understandings of how complicated topography affects various statistics of the ABL such as mean velocity profiles, shear stresses, and turbulence statistics. Advancements in understandings of the ABL have the potential to improve numerous fields of engineering and science. For instance, novel parameterizations can eventually be developed to improve the fidelity and robustness of mesoscale simulations. These mesoscale simulations can be used to better predict weather patterns and pollution transport. Greater understanding of the ABL also provides an opportunity to improve the design of local systems such as wind farms or efficient fertilizer distribution. All of these benefits from a simulation based study are only possible given the development of robust, high fidelity LES capabilities. This thesis addressed some techniques, difficulties, and results of performing "topography-resolved" LES.

In Chapter 2, background information on the ABL, fluid mechanics, and CFD were presented. This included an overview of the important features of the ABL as well as the numerical methods that have been used to simulate the ABL. Then the governing equations for incompressible Newtonian fluids were presented. These equations were also nondimensionalized and filtered to generate the governing equations solved by LES. Lastly, an overview of the important aspects of CFD were presented. While far from universally encompassing, these sections provided the required background to understand the work conducted in Chapter 3 and Chapter 4.

In Chapter 3, the methodologies required to simulate ABL flows were presented, developed, and tested. In Section 3.1, the techniques for discretizing and solving the temperature transport equation were presented. The discretization of the time advancement, convective term, and diffusion term were tested using verification tests. Then the complete temperature transport equation was validated using channel flow. Next, Section 3.2 described the technique for importing real topographic data into the simulation through generation of an STL file. This method was parallelized to speed up the calculation and allow for topographic data with higher resolution. Furthermore, this methodology was verified by inspecting the level-set function used to capture the topography in the simulation. In Section 3.3, the IB method was improved to allow for simulation over topography. This included presenting a novel method of stencil point calculation that can be used on meshes with large aspect ratios. A wall model was presented to be used in conjunction with the IB method. This model was validated using a simple channel flow case. Finally, the method of generating a turbulent inflow boundary condition was presented in Section 3.4. This inflow boundary condition can be set to realistic conditions using either field measurements or mesoscale simulations. This method was verified by comparing the spatially averaged velocity field to an expected velocity profile.

In Chapter 4, a full scale simulation of a stably stratified flow at Elkhorn Slough in Monterey Bay California was conducted. The simulation used boundary conditions taken from real field measurements. The goal of the study was to answer three research questions: (1) how important is the influence of real topography on the flow field, (2) how do simulation results compare to other simulations and field measurements, and (3) what are the aspects that still need improvement. Through qualitative investigation it was found that the topography significantly influences the flow field. It was also found that the simulation accurately captures the mean profile in the region far above the surface. In the region near the surface, the velocity is under predicted. Unrealistically large velocities at the top of the domain were also observed. It was hypothesized that the cause of both these errors is a result of inaccuracies caused by the surface roughness model over the sea surface.

5.2 Recommendations

The recommendations presented will be broken into two categories. The first section will discuss methods to improve the fidelity of the present LES. The second section will discuss further studies of the ABL that are possible because of these simulation capabilities.

Improving LES Fidelity

The fidelity of the present LES can be improved by addressing various deficiencies apparent from this thesis. One aspect of current capabilities that must be further tested and modified is the advancement of the surface roughness model. It was hypothesized that the errors in the full scale test are due to the inaccuracies caused by the surface roughness model. An in depth analysis of the effect that different surface roughness models have on the simulation would allow for more accurate results. Furthermore, the LES could be advanced by resolving the ocean waves rather than modeling them. This could lead to a fully coupled ocean/wind/land LES capable of resolving many fluid domains.

Another aspect of the present simulation that could be improved is the wall modeling. One critical assumption of the wall model was that it assumes the law of the wall. This is based off the analysis of flow fields that are not separated. In simulations of the ABL with complex topography, there are likely to be regions of separation in which these wall models may be less accurate. In this thesis, validation tests of the wall model were performed on an idealized channel flow. However, this validation case would not present any errors resultant of separated flow. Further development of wall model methods will require validation tests with regions of separated flow.

Another way of improving the LES fidelity is to conduct further validation studies based on field measurements. While the study presented in Chapter 4 did provide preliminary insight into accuracy of the present LES, there were significant gaps in the measurements collected. These gaps could be closed by conducting novel hybrid LES/field measurement studies. Having more measurement sites in the domain could result in a better analysis

of the effect of heterogeneous topography. Additionally, studies where temperature measurements are obtained would provide an accurate method of validating the transfer of temperature.

Advancing knowledge of ABL

The present LES could be used in future studies to advance the knowledge of the ABL. One study could be on the effect that mean wind direction has on the turbulent mixing. In Chapter 4, the topographic item of interest (coastline) was perpendicular to the mean flow direction. Simulating multiple cases with different mean flow directions could provide insight that could be used by mesoscale simulations to develop "coast aware" parameterizations based on the topography and mean flow direction. This would require developing novel methods of generating an inflow boundary condition for flows that are not over a homogeneous surface.

Another study that could advance understanding of the ABL is a coupled mesoscale/LES based study. Mesoscale and LES simulations could be run in conjunction over the same topography. The LES results would be taken as the true flow field that the mesoscale results could be compared with. Modifications to the M-O theory used in mesoscale models could then be made, and the impact of these modifications on the accuracy of the mesoscale simulation could be documented. These results could then be used to create new similarity models that depend on the characteristics of the topography.

5.3 Concluding Remarks

If I cease searching, then, woe is me, I am lost.

That is how I look at it - keep going, keep going come what may.

- Vincent van Gogh, 1880, Letter to Brother

While the staggering beauty depicted in *The Starry Night* has claimed global recognition and praise, the circumstances of the man sitting on the other end of the canvas are easy to

overlook. When he crafted this painting, Vincent van Gogh was amidst one of the most turbulent periods of his life. At the time, Van Gogh was a struggling artist who had just admitted himself to the *Saint Paul de Mausole Lunatic Asylum* after mutilating his own ear. *The Starry Night* depicts the view outside of Van Gogh's window. The chaotic whirls dominating this piece likely depict the turbulence within Van Gogh's mind in addition to the turbulent winds outside his window. This same turbulence depicted by Van Gogh still plagues atmospheric scientists to this day. In the same way that Van Gogh broke down the complexity with his small, deliberate brushstrokes, scientists must also break down the complexity of turbulence by investigating it piece by piece. Every small lesson learned about turbulence brings humankind one step closer to solving the most "essential unsolved problem of classical mechanics."

Bibliography

Abe, H., H. Kawamura, and Y. Matsuo (June 2004). "Surface heat-flux fluctuations in a turbulent channel flow up to $Re_\tau = 1020$ with Pr = 0.025 and 0.71". In: *International Journal of Heat and Fluid Flow* 25.3, pp. 404–419. URL: https://doi.org/10.1016/j.ijheatfluidflow.2004.02.010.

Alamo, J.C. del et al. (Apr. 2004). "Scaling of the energy spectra of turbulent channels". In: *Journal of Fluid Mechanics* 500, pp. 135–144. URL: https://torroja.dmt.upm.es/pubs/2004/delAlamoJimenezZandonadeMoser_jfm_2004.pdf.

Antonia, R.A., L.V. Krishnamoorthy, and L. Fulachier (Apr. 1988). "Correlation between the longitudinal velocity fluctuation and temperature fluctuation in the near-wall region of a turbulent boundary layer". In: *International Journal of Heat and Mass Transfer* 31.4, pp. 723–730. URL: https://doi.org/10.1016/0017-9310(88)90130-5.

Batchelor, G.K. (1953). *The Theory of Homogeneous Turbulence*. Cambridge University Press.

Batchelor, G.K. and I. Proudman (Jan. 1954). "The effect of rapid distortion of a fluid in a turbulent motion". In: *The Quarterly Journal of Mechanics and Applied Mathematics* 7.1, pp. 83–103. URL: https://doi.org/10.1093/qjmam/7.1.83.

Beyer, R.P. (Jan. 1992). "A computational model of the cochlea using the immersed boundary method". In: *Journal of Computational Physics* 98, pp. 145–162. URL: https://doi.org/10.1016/0021-9991(92)90180-7.

Bou-Zeid, E. (Jan. 2005). "A scale-dependent Lagrangian dynamic model for large eddy simulation of complex turbulent flows". In: *Physics of Fluids* 17, p. 025105. URL: https://doi.org/10.1063/1.1839152.

Breuer, M. et al. (Feb. 2009). "Flow over periodic hills — Numerical and experimental study in a wide range of Reynolds numbers". In: *Computers and Fluids* 38, pp. 433–457. URL: https://doi.org/10.1016/j.compfluid.2008.05.002.

Calaf, M., M.B. Parlange, and C. Meneveau (Dec. 2011). "Large eddy simulation study of scalar transport in fully developed wind turbine array boundary layers". In: *Phys. Fluids*, p. 126603. URL: https://doi.org/10.1063/1.3663376.

Charnock, H. (Oct. 1955). "Wind stress on a water surface". In: *Quarterly Journal of the Royal Meteorological Society* 81.350, pp. 639–640. URL: https://doi.org/10.1002/qj.49708135027.

Chen, B. and M. Chamecki (Apr. 2019). "Effects of topography on in-canopy transport of gases emitted within dense forests". In: *Quarterly Journal of the Royal Meteorological Society* 145. URL: https://doi.org/10.1002/qj.3546.

Chow, F.K. et al. (2005). "Explicit filtering and reconstruction turbulence modeling forlarge-eddy simulation of neutral boundary layer flow". In: *Journal of the Atmospheric Sciences* 62.7, pp. 2058–2077. URL: https://doi.org/10.1175/JAS3456.1.

Deardorff, J.W. (July 1972). "Numerical Investigation of Neutral and Unstable Planetary Boundary Layers". In: *National Center for Atmospheric Research*. URL: https://doi.org/10.1175/1520-0469(1972)029<0091:NIONAU>2.0.CO;2.

– (Oct. 1974). "Three-dimensional numerical study of turbulence in an entraining mixed layer". In: *Boundary Layer Meteorology* 7.2, pp. 199–226. URL: https://doi.org/10.1007/BF00227913.

Ding, F., S.P. Arya, and Y.L. Lin (Mar. 2001). "Large-eddy simulations of the atmospheric boundary layer using a new subgrid scale model – I. Slightly unstable and neutral cases". In: *Environmental Fluid Mechanics* 1.1, pp. 29–47. URL: https://doi.org/10.1023/A:1011547800570.

Donea, J. and A. Huerta (2013). *Finite Element Methods for Flow Patterns*. Wiley.

Fang, X. et al. (June 2017). "Large-eddy simulation of turbulent flow and structures in a square duct roughened with perpendicular and V-shaped ribs". In: *Physics of Fluids* 29, p. 065110. URL: https://doi.org/10.1063/1.4985715.

Fauci, L.J. and A. McDonald (Sept. 1995). "Sperm motility in the presence of boundaries". In: *Bulletin of Mathematical Biology* 57.5, pp. 679–699. URL: https://www.sciencedirect.com/science/article/abs/pii/009282409500022I.

Germano, M. et al. (1991). "A dynamic subgrid-scale eddy viscosity model". In: *Phys. Fluids* 1760. URL: https://doi.org/10.1063/1.857955.

Hager, J.W., J.F. Behensky, and B.W. Drew (1989). "The universal grids: Universal Transverse Mercator (UTM) and Universal Polar Stereographic (UPS)". In: *Defense Mapping Agency* TM8358.2, p. 065110. URL: https://earth-info.nga.mil/GandG/publications/tm8358.2/TM8358_2.pdf.

Harten, A. (Mar. 1983). "High resolution schemes for hyperbolic conservation laws". In: *Journal of Computational Physics* 49.3, pp. 357–393. URL: https://doi.org/10.1016/0021-9991(83)90136-5.

He, S., Z. Yang, and L. Shen (Nov. 2017). "Numerical Simulation of Interactions among Air, Water, and Rigid/Flexible Solid Bodies". In: *10th International Workshop on Ship and Marine Hydrodynamics*. URL: https://www.researchgate.net/publication/322953123_Numerical_Simulation_of_Interactions_among_Air_Water_and_RigidFlexible_Solid_Bodies.

Jaszczur, M. (Aug. 2014). "DNS benchmark solution of the fully developed turbulent channel flow with heat transfer". In: *Journal of Physics: Conference Series* 530.3, p. 012022. URL: https://doi.org/10.1088/1742-6596/530/1/012022.

Kaimal, J.C. and J.J. Finnigan (1994). *Atmospheric boundary layer flows: their structure and measurement*. Oxford University Press.

Kang, S. (Feb. 2015). "An improved near-wall modeling for large-eddy simulation using immersed boundary methods". In: *International Journal for Numerical Methods in Fluids* 78, pp. 76–88. URL: https://doi.org/10.1002/fld.4008.

Karman, T. von (1930). "Mechanische Ähnlichkeit und Turbulenz". In: *Proc. Third Int. Congr. Applied Mechanics*, pp. 85–105. URL: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930094805.pdf.

– (1948). "Progress in the Statistical Theory of Turbulence". In: *Proceedings of the National Academy of Sciences* 34.11, pp. 530–539. URL: https://doi.org/10.1073/pnas.34.11.530.

Kawai, S. and J. Larsson (Jan. 2012). "Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy". In: *Physics of Fluids* 24, p. 015105. URL: https://doi.org/10.1063/1.3678331.

Kim, J., P. Moin, and R. Moser (Apr. 1987). "Turbulence statistics in fully developed channel flow at low Reynolds number". In: *Journal of Fluid Mechanics* 177, pp. 133–166. URL: https://doi.org/10.1017/S0022112087000892.

Klebanoff, P.S. (1954). "Characteristics of turbulence in a boundary layer with zero pressure gradient". In: *NASA Technical Report* 1247. URL: https://archive.org/details/NASA_NTRS_Archive_19930092249/page/n2/mode/2up.

Kolmogorov, A.N. (1941). "The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers". In: *Dokl. Akad. Nauk. USSR* 30, pp. 299–303. URL: https://www.jstor.org/stable/51980.

Lee, J., M. Cho, and H. Choi (Aug. 2013). "Large eddy simulations of turbulent channel and boundary layer flows at high Reynolds number with mean wall shear stress boundary condition". In: *Physics of Fluids* 25.11, p. 110808. URL: https://doi.org/10.1063/1.4819342.

Lee, M.J., J. Kim, and P. Moin (July 1990). "Structure of turbulence at high shear rate". In: *Journal of Fluid Mechanics* 216, pp. 561–583. URL: https://doi.org/10.1017/S0022112090000532.

Leonard, B.P. (June 1979). "A stable and accurate convective modelling procedure based on quadratic upstream interpolation". In: *Computer Methods in Applied Mathematics and Engineering* 19.1, pp. 59–98. URL: https://doi.org/10.1016/0045-7825(79)90034-3.

Lilly, D.K. (Nov. 1966). "The representation of small-scale turbulence in numerical simulation experiments". In: *Proc. IBM Scientific Computing Symp. on Environmental Sciences*, pp. 195–210. URL: http://dx.doi.org/10.5065/D62R3PMM.

Lilly, D.K. (Feb. 1992). "A proposed modification of the Germano subgrid scale closure method". In: *Phys. Fluids* 633. URL: https://doi.org/10.1063/1.858280.

MacMahan, J. (June 2017). "Increased aerodynamic roughness owing to surfzone foam". In: *Journal of Physical Oceanography* 47, pp. 2115–2122. URL: https://doi.org/10.1175/JPO-D-17-0054.1.

Maeda, T., S. Homma, and Y. Ito (Dec. 2004). "Effect of complex terrain on vertical wind profile measured by SODAR technique". In: *Wind Engineering* 28.6, pp. 667–678. URL: https://doi.org/10.1260/0309524043729895.

Mann, J. (Oct. 1998). "Wind field simulation". In: *Probabilistic Engineering Mechanics* 13.4, pp. 269–282. URL: https://doi.org/10.1016/S0266-8920(97)00036-2.

Markfort, C.D. (2013). "Wake dynamics in the atmospheric boundary layer over complex terrain". PhD thesis. University of Minnesota. URL: https://conservancy.umn.edu/handle/11299/167268.

Mason, P.J. and S.H. Derbyshire (Oct. 1990). "Large-Eddy Simulation of the stably-stratified atmospheric boundary layer". In: *Boundary Layer Meteor.* 53, pp. 117–162. URL: https://doi.org/10.1007/BF00122467.

Meneveau, Charles and K.R. Sreenivasan (Mar. 1991). "The multifractal nature of turbulent energy dissipation". In: *Journal of Fluid Mechanics* 224, pp. 429–484. URL: https://doi.org/10.1017/S0022112091001830.

Mittal, R. and G. Iaccarino (2004). "Immersed boundary method". In: *Annual Review Fluid Mechanics* 37, pp. 239–261. URL: https://www.annualreviews.org/doi/10.1146/annurev.fluid.37.061903.175743.

Monin, A.S. and A.M. Obukhov (1954). "Basic laws of turbulent mixing in the surface layerof the atmosphere". In: *Tr. Akad. Nauk SSSR Geophiz. Inst.* 24.151, pp. 163–187. URL: https://mcnaughty.com/keith/papers/Monin_and_Obukhov_1954.pdf.

Nicholas, F.W. and J.E. Lewis (1980). "Relationships Between Aerodynamic Roughness and Land Use and Land Cover in Baltimore, Maryland". In: *Geological Survey Professional Paper*. URL: https://pubs.usgs.gov/pp/1099c/report.pdf.

Ortiz-Suslow, D.G. et al. (2015). "The spatial-temporal variability of air-sea momentum fluxes observed at a tidal inlet". In: *Journal of Geophysical Research: Oceans* 120, pp. 660–676. URL: https://doi.org/10.1002/2014JC010412.

Patankar, S. (1980). *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Co.

Pearson, R.A. (1959). "The effect of uniform distortion on weak homogeneous turbulence". In: *Journal of Fluid Mechanics* 5, pp. 274–288. URL: https://ui.adsabs.harvard.edu/abs/1959JFM.....5..274P/abstract.

Peskin, C.S. (Oct. 1972). "Flow patterns around heart valves: A numerical method". In: *Journal of Computational Physics* 10.2, pp. 252–271. URL: https://doi.org/10.1016/0021-9991(72)90065-4.

– (Jan. 1982). "The fluid dynamics of heart valves: experimental, theoretical and computational methods". In: *Annual Review of Fluid Mechanics* 14, pp. 235–259. URL: https://doi.org/10.1146/annurev.fl.14.010182.001315.

Piomelli, U. and E. Balaras (Jan. 2002). "Wall-layer models for large-eddy simulations". In: *Annual Review of Fluid Mechanics* 34, pp. 349–374. URL: https://doi.org/10.1146/annurev.fluid.34.082901.144919.

Pope, S.B. (2000). *Turbulent Flows*. Cambridge University Press.

Prandtl, L. (1925). "Bericht iiber die Entstehung der Turbulenz". In: *Z. Angew. Math. Mech.*, pp. 136–139. URL: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20050029454.pdf.

Roman, F., V. Armenio, and J. Fröhlich (Oct. 2009). "A simple wall-layer model for large eddy simulation with immersed boundary method". In: *Physics of Fluids* 21, p. 101701. URL: https://doi.org/10.1063/1.3245294.

Roth, S.D. (Feb. 1982). "Ray casting for modeling solids". In: *Computer Graphics and Image Processing* 18.2, pp. 109–144. URL: https://doi.org/10.1016/0146-664X(82)90169-1.

Smagorinsky, J. (Mar. 1963). "General circulation experiments with the primitive equations: I. The basic equation". In: *Monthly Weather Review* 91.3, pp. 99–164. URL: https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.

Sommeria, G. (1976). "Three-Dimensional Simulation of Turbulent Processes in an Undisturbed Trade Wind Boundary Layer". In: *Journal of the Atmospheric Sciences* 33, pp. 216–241. URL: https://doi.org/10.1175/1520-0469(1976)033<0216:TDSOTP>2.0.CO;2.

Spalart, P.R. (Feb. 1988). "Direct simulation of a turbulent boundary layer up to Rθ = 1410". In: *Journal of Fluid Mechanics* 187, pp. 61–98. URL: https://doi.org/10.1017/S0022112088000345.

Stull, R.B. (1988). *An Introduction to Boundary Layer Meteorology*. Springer, Dordrecht.

Taylor, G.I. (Feb. 1938). "The Spectrum of Turbulence". In: *Royal Society* 164, pp. 269–282. URL: https://doi.org/10.1098/rspa.1938.0032.

Verzicco, R. et al. (Jan. 1998). "LES in complex geometries using boundary body forces". In: *American Institute of Aeronautics and Astronautics Journal* 38, pp. 427–433. URL: https://www.researchgate.net/publication/251557520_LES_in_complex_geometries_using_boundary_body_forces.

Weber, R.O. (Nov. 1999). "Remarks on the Definition and Estimation of Friction Velocity". In: *Boundary-Layer Meteorology* 93.2, pp. 197–209. URL: https://doi.org/10.1023/A:1002043826623.

Yang, X. et al. (Aug. 2014). "Large eddy simulation of turbulent flow past wind turbines / farms: the Virtual Wind Simulator (VWiS)". In: *Wind Energy* 18, pp. 2025–2045. URL: https://doi.org/10.1002/we.1802.

Yang, Z. et al. (Feb. 2018). "Numerical Study on the Effect of Air–Sea–Land Interaction on the Atmospheric Boundary Layer in Coastal Area". In: *Atmosphere* 9(2). URL: https://doi.org/10.3390/atmos9020051.

Yang, Z. et al. (Jan. 2019). "Measurement-Based Numerical Study of the Effects of Realistic Land Topography and Stratification on the Coastal Marine Atmospheric Surface Layer". In: *Boundary Layer Meteorology* 171, pp. 289–314. URL: https://doi.org/10.1007/s10546-018-00423-2.

Appendix A

Incompressible Fluid Solver

The following describes the methodology of solving the incompressible Navier Stokes equations using the projection method. The equation being solved is the momentum equation in conservative form

$$\frac{\partial u_i}{\partial t} = \frac{\partial}{\partial x_j}\Big[H_{ij} - p\delta_{ij}\Big], \quad \tau_{ij} = \frac{2S_{ij}}{Re}, \quad H_{ij} = -u_i u_j + \tau_{ij} \tag{A.1}$$

In order to ensure mass is conserved ($\partial u_i / \partial x_i = 0$) the momentum equation needs to be solved using a projection method. The basis of the projection method is that Equation A.1 is broken into two equations

$$\begin{aligned} \frac{u^* - u^n}{\Delta t} &= \frac{\partial H_{ij}}{\partial x_j} \\ \frac{u^{n+1} - u^*}{\Delta t} &= -\frac{\partial p}{\partial x_i} \end{aligned} \tag{A.2}$$

The divergence of the second line in Equation A.2 is taken and combined with conservation of mass resulting in

$$\begin{aligned} \frac{\partial}{\partial x_i}\Big[u_i^{n+1} = u_i^* - \Delta t \frac{\partial p^n}{\partial x_i}\Big] \\ 0 = \frac{\partial u_i^*}{\partial x_i} - \Delta t \frac{\partial^2 p^n}{\partial x_i x_i} \\ \frac{1}{\Delta t}\frac{\partial u_i^*}{\partial x_i} = \frac{\partial^2 p^n}{\partial x_i x_i} \end{aligned} \tag{A.3}$$

In addition to the projection method, the RK2 time advancement is implemented which means this step needs to be performed twice. In the following steps, superscript '*' represents the projection step described above and superscript '$n + 1/2$' represents the intermediate velocity used in RK2.

First RK2 Step

The goal of the first RK2 step is to calculate the intermediate velocity $u_i^{n+1/2}$. First, the projection velocity at the intermediate time step $u_i^{n+1/2,*}$ is calculated as

$$u_i^{n+1/2,*} = u_i^n + \Delta t \frac{\partial H_{ij}^n}{\partial x_j} \tag{A.4}$$

Then the pressure p^n is calculated by solving the Poisson equation

$$\frac{\partial^2 p^n}{\partial x_i \partial x_i} = \frac{1}{\Delta t} \frac{\partial u_i^{n+1/2,*}}{\partial x_i} \tag{A.5}$$

Finally, the intermediate velocity is calculated as

$$u_i^{n+1/2} = u_i^{n+1/2,*} - \Delta t \frac{\partial p^n}{\partial x_i} \tag{A.6}$$

Second RK2 Step

The goal of the second RK2 step is to calculate the final velocity u_i^{n+1}. First, the projection velocity at the final time step $u_i^{n+1,*}$ is calculated

$$u_i^{n+1,*} = u_i^{n+1/2} + \Delta t \frac{1}{\partial x_j} \left[H_{ij}^{n+1/2} - H_{ij}^n + p^n \right] \tag{A.7}$$

Next, the pressure $p^{n+1/2}$ is calculated by solving the Poisson equation

$$\frac{\partial^2 p^{n+1/2}}{\partial x_i \partial x_i} = \frac{2}{\Delta t} \frac{\partial u_i^{n+1,*}}{\partial x_i} \tag{A.8}$$

The final velocity u^{n+1} is then be calculated as

$$u_i^{n+1} = u_i^{n+1,*} - \frac{\Delta t}{2} \frac{\partial p^{n+1/2}}{\partial x_i} \tag{A.9}$$