Subgrid Modelling for Large-Scale Atmospheric Flow

by

Luke Fitzpatrick

A masters research paper
presented to the University of Waterloo
in fulfilment of the
requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2019

Abstract

Understanding turbulent flow is vastly important for our everyday lives. For instance accurate weather prediction, something most of us rely on everyday, would not be possible without the understanding and modelling of turbulence. It is unfortunate that even in the 21st century we are far from having the computational power necessary to implement direct numerical simulation on the majority of turbulent simulations. Thus, the use of a turbulence model is a necessity. The Smagorinsky model, originally developed by Joseph Smagorinsky in 1963, has continued to be a useful model over the decades; however, it is not without its flaws. In particular it is known to be overly dissipative - a fault thought to be linked to a violation of scale invariance. To remedy this the Dynamic Smagorinsky Model (DSM) can be used. We implement the DSM into the Weather Research and Forecasting (WRF) model and use WRF's real data capabilities to simulate Hurricane Igor. First we present a base case featuring results strictly from the Smagorinsky model. We compare with recorded data to validate our simulation. We then compare against DSM for several physical quantities using two methods to compute the Smagorinsky constant c_s and find little difference. Energy spectra are presented for all cases using two domains and resolutions. Surprisingly, we find almost no differences with the classic Smagorinsky model; however, a time analysis shows that within the confines of WRF, the additional cost is negligible. Thus there are no major downsides of using it. A few differences we do note are more energy at small length scales when coarse resolution is used and more vigorous gravity waves propagating away from regions of precipitation. We conclude that under these specific circumstances, using the DSM for large scale atmospheric flows presents little benefit; however, there may be altered approaches to improve results.

Acknowledgements

I would like to thank all the little people who made this research paper possible. In particular, my supervisor Dr. Michael Waite for all his help in just about every part of the project. I'd also like to thank Dr. Marek Statsna for being my secondary reader and providing secondary feedback. Finally I'd like to thank a fellow masters student Kwan Lai for his initial help setting up the filtering for my implementation of the Dynamic Smagorinsky Model.

Table of Contents

List of Tables vi

List of Figures vii

Abbreviations ix

1 Introduction 1

1.1 Motivation . . . 1

1.2 Hurricane Igor . . . 4

1.3 The Governing Equations . . . 6

1.3.1 Equation of State . . . 6

1.3.2 Thermodynamic Relations . . . 7

1.3.3 Thermodynamic Equations for Fluids . . . 8

1.3.4 The Set of Equations . . . 11

1.4 Objectives . . . 12

2 Large-Eddy Simulation 14

2.1 Direct Numerical Simulation . . . 14

2.2 Filtering . . . 15

2.3 LES Equations 18
2.3.1 LES in WRF 21
2.4 Scale Invariance 22
2.5 Dynamic Smagorinsky Model 24

3 Weather Research and Forecasting Model **29**
3.1 Overview 29
3.2 Simulation Setup 30
3.3 Implementing Dynamic Smagorinsky 33

4 Results **36**
4.1 Base Case 36
4.2 Comparison with Dynamic Smagorinsky 44
4.3 Energy Spectra 47

5 Conclusion **53**

References **55**

List of Tables

4.1 Minimum pressure of Hurricane Igor for various times and locations. Data provided by Weather Underground. 38

List of Figures

1.1 Satellite image showing Igor on September 20th, shortly before landing in Newfoundland. Provided by [44]. 5

3.1 Simulation domain contoured by terrain height. 31

4.1 Sea Level Pressure contours at times: **(a)** 00 UTC, **(b)** 06 UTC, **(c)** 12 UTC, and **(d)** 18 UTC on September 21st. Additionally u velocity vectors at 10 m and temperature at 2 m are shown. 37

4.2 Simulated track (black) versus data obtained from Weather Underground (white) of Igor via the minimum pressure. Δt black: 3 hours; Δt white: 6 hours. Units are hPa. 39

4.3 Total rainfall over Newfoundland. Obtained from [38]. 40

4.4 Rain rate per 3 hours for times: **(a)** 18 UTC on September 20th and **(b)** 03 UTC, **(c)** 12 UTC, and **(d)** 21 UTC on September 21st. Coloured by terrain. 41

4.5 Spatial averaged rain rate per 3 hours over the entire domain. 42

4.6 Vorticity at times: **(a)** 18 UTC on September 20th and **(b)** 03 UTC on September 21st. 43

4.7 Sea level pressure contours at time 12 UTC for: **(a)** Smagorinsky model, **(b)** DSM with filtering, and **(c)** DSM on September 21st. 44

4.8 Rain rate per 3 hours. **Top:** September 20th, 18 UTC, **Bottom:** September 21st, 03 UTC. **Left:** DSM with filtering, **Right:** DSM. Coloured by terrain. 46

4.9 Vertical velocity contours at time 09 UTC for: **(a)** Smagorinsky model, **(b)** DSM with filtering, and **(c)** DSM on September 21st. 48

4.10 Vertical velocity contours at time 09 UTC for: **(a)** Smagorinsky model, **(b)** DSM with filtering, and **(c)** DSM on September 21st. Small, coarse domain. 49

4.11 Energy spectrum of horizontal velocity u on a loglog scale. **Top:** small, coarse domain, **Bottom:** large, fine domain. **Left:** DSM with filtering, **Right:** DSM. 51

4.12 Energy spectrum of vertical velocity w on a loglog scale. **Top:** small, coarse domain, **Bottom:** large, fine domain. **Left:** DSM with filtering, **Right:** DSM. 52

Abbreviations

CFD Computational Fluid Dynamics 3, 32, 54

CFL Courant-Friedrichs-Lewy 15

DNS Direct Numerical Simulation 3, 14–16

DSM Dynamic Smagorinsky Model ii, vii, viii, 3, 13, 14, 17, 21, 22, 24, 26–30, 32, 33, 36, 44–54

GRIB General Regularly-distributed Information in Binary form 30, 31

LES Large-Eddy Simulation 3, 14–16, 18, 19, 21, 23, 24, 26, 50

NARR North American Regional Reanalysis 30–32, 42

NCAR National Center for Atmospheric Research 29, 30

NCEP National Centers for Environmental Prediction 29, 30, 54

NSE Navier-Stokes Equations 14, 16, 19, 22

RANS Reynolds-Averaged Navier-Stokes 2, 15

SFS sub-filter-scale 20

SGS subgrid-scale 14, 16, 18–20, 26

SSHWS Saffir-Simpson Hurricane Wind Scale 4

WPS WRF Preprocessing System 31, 32

WRF Weather Research and Forecasting ii, 3, 4, 6, 11, 13, 20, 21, 29, 30, 32–34, 53, 54

Chapter 1

Introduction

1.1 Motivation

What is turbulence? Why do we study it? There is a lot to be said about the nature of turbulent flows. Humanity has been making strides to understand turbulence for hundreds of years. Indeed, perhaps the oldest account of someone truly understanding the concept was that of Leonardo da Vinci circa in the early 1500s. Leonardo was ahead of his time; illustrating what was supposed to be flow entering a tank and the formation of various turbulent structures not too unlike what we know true today [19]. Leonardo's idea of turbulence has been investigated further by Monaghan and Kajtar (2014) [27]. One of the most important discoveries came much later in the 1800s at the hands of Sir Osborne Reynolds (1883) [35]. In his famous pipe-flow experiment, Reynolds was able to demonstrate the onset of turbulent flow. Through this he could determine a criteria for the transition of flow from laminar to turbulent. This coined the famous *Reynolds number*: $\mathrm{Re} = uL/\nu$, where u and ν are the fluids velocity and kinematic viscosity, and L defines a length scale. It is known that low Reynolds numbers characterize laminar flow where viscosity is dominant and flow is smooth. Then high Reynolds numbers characterize turbulent flow dominated by inertial forces. Since then the study of turbulent flow has only grown. A quick search of a few keywords on Google Scholar is proof of that. If not turbulent flows directly then turbulence modelling or test cases in which turbulence plays

a major factor. We still have yet to answer - what is turbulence anyway?

The ideas in the remainder of this section can be referenced from [15]. Anyone that has been on an airplane before surely has some concept of turbulence. The vigorous shaking of an aircraft as it travels through a turbulent regime is certainly one example. However, one need not book a flight just to experience turbulence themselves. For turbulence is around us in everyday life. Smoke rising from a chimney or fire, water rapidly flowing down a rapids, wind sharply changing direction and speed are all examples of flow which can demonstrate turbulent qualities. More specifically, turbulence is characterized by unsteady, irregular, seemingly random and chaotic flow. It is linked with the unpredictable creation of chaotic 3D vorticity, or eddies - a rotational movement of fluid creating a vortex. The idea that large eddies can excite small eddies through vortex stretching is essential in turbulence. An example of turbulent flow can be observed simply by turning the tap on a kitchen sink. While the initial flow of water will be laminar, the spraying of water droplets around the basin demonstrates turbulent flow - each droplet of water is highly unpredictable. Increase the pressure and let the water strike a rough surface for further exaggeration. With its highly chaotic nature, one might think turbulence would fit well under the relatively new chaos theory; however, it is not a perfect fit. Unlike chaos in the dynamical systems sense, turbulence has a space dependence in addition to time dependence. Still many of its features closely resemble chaos such as high sensitivity to initial conditions in turbulent jets. Because of this randomness, turbulence is often studied in a statistical sense. Computational methods such as Reynolds-Averaged Navier-Stokes (RANS) rely completely on an averaging technique.

Why we study turbulence should be clear. It is present in everyday life; thus, an interest among physicists, mathematicians, and engineers alike is undeniable. A pessimist might argue that turbulence is too complex and we know too little to be considering engineering problems. There are always fewer equations than unknowns to predict anything other than instantaneous motions. Hence the problem with turbulence closure. Yet real engineering problems must be tackled: airplanes must fly, weather must be predicted, water plants must be build and so on. Then the question is not why, but *how*. Of course experiments

were always the first step as demonstrated by Reynolds (1883) [35]. Experiments are helpful in that they provide concrete evidence in how turbulent flows in the real world are supposed to behave. The difficulty in experiments is often the logistics. It is easy for the average person to obtain a small tank to fill with water and do basic experiments, but it very quickly becomes out of reach. Access to large (say a few cubic meters) tanks will likely only be accessible to top research facilities. The same can be said for a large wind tunnel. Consider wanting to study large scale flows in the atmospheric boundary layer. Do we have the means to set probes atop of large obstacles (i.e. buildings) to record data? Because of these challenges the development of Computational Fluid Dynamics (CFD) techniques has been a necessity.

If we will not use an experiment to study turbulence then we must model it and simulate it on a computer. Hence the concept of turbulence modelling, which this report has focus on. In particular we are interested in what is known as the 2D horizontal *Smagorinsky* model for the eddy viscosity, and how it can be improved via the DSM for numerical models of the atmosphere. The term model in this context truthfully is a model in that we are not representing the equations exactly. To do so would require a technique known as Direct Numerical Simulation (DNS), which presents a slue of challenges. This is why we opt for Large-Eddy Simulation (LES). Turbulence modelling techniques were being studied well before we had the computational power to simulate anything substantial. This is evident by the Smagorinsky model's inception by Joseph Smagorinsky in 1963 [43] and its first implementation by Deardorff in [13]. In the present work we make use of the WRF model and its real data capabilities to simulate Hurricane Igor. We note that how we are using WRF does not make use of LES in the traditional sense, but the approach is very similar. This is discussed more in Chapter 2. The choice of a large scale weather system like a hurricane has no direct relation to the turbulence models. One could study the DSM using many different test cases, but this allows us to determine if the expected deficits and improvements will actually be observed using WRF for such large scale flow. Hurricanes are also exciting, which hopefully gives this report a more interesting premise.

To close this section we note that our understanding of turbulence is still quite limited.

Look no further than the often inaccurate prediction of weather forecasts leading to frustrations of people daily. There is still much to learn about turbulent flows and some of what we know now may end up untrue. George (2013) [15] puts it quite nicely. The idea that the sun revolved around the Earth was a fine idea at the time - one which almost anyone would believe. The only problem was when someone looked up closely and realized it wasn't true. While a revelation on such a scale may be a bit extreme, there is certainly much to discover.

1.2 Hurricane Igor

As mentioned we will be using WRF's real data capabilities to simulate Hurricane Igor using various turbulence models. This section will provide the reader with basic background on said hurricane. Much of the information in this section was obtained from [32]. Otherwise it is cited accordingly. Hurricane Igor was the most destructive tropical cyclone to ever strike the island of Newfoundland - the home province of the author. Because of this fact, it was of particular interest to simulate. On top of what was just mentioned, it was the strongest hurricane during the 2010 Atlantic hurricane season. Igor originated off the west coast of Africa due to a broad area of low pressure moving off the Cape Verde islands on September 6th, 2010. It slowly crept across the Atlantic reaching tropical storm levels by the 10th. Explosive intensification took place on the 12th, where Igor reached Category 4 status on the Saffir-Simpson Hurricane Wind Scale (SSHWS). With a 27 km wide eye, deep convection, and spiral banding, Igor retained this status for 4 days. It reached peak strength on September 15th around 0000 UTC, with 250 km/h 1-minute sustained wind speeds and a minimum pressure of 924 hPa. Around this time Igor began its north-westward track up the Atlantic Ocean. Thankfully, Igor was quite far from land mass during its most destructive phase, and damages were relatively small outside of Newfoundland. Igor began to weaken on the 17th due to increased wind shear and dry air intrusion as it sat roughly 555 km northeast of the Leeward Islands. Over the next few days Igor fell to a Category 1 status; however, it grew very large in size with tropical storm-force winds covering an area about 1390 km [7]. On September 20th Igor's centre reached Bermuda with wind speeds now as low as 120 km/h. Igor began transitioning

to an extratropical cyclone as it approached Newfoundland. Figure 1.1 pictures exactly this. Deep convection became less consistent through its centre; however, the storm re-intensified as it passed a baroclinic zone while nearing Newfoundland. This allowed it to strengthen despite the lowering sea surface temperatures [10]. Igor made landfall near Cape Race of Newfoundland around 1500 UTC on September 21st. Wind speeds reach 140 km/h. Shortly after Igor completed its transition to an extratropical storm. Two days later the remnants of Igor were absorbed by another extratropical cyclone within the Labrador sea. Newfoundland is a province which is seldom on the receiving end of powerful tropical

Figure 1.1: Satellite image showing Igor on September 20th, shortly before landing in Newfoundland. Provided by [44].

storms, so even a Category 1 hurricane was quite devastating. Unprecedented amounts of rainfall were recorded across parts of eastern Newfoundland, leading to widespread flooding [10]. In Bonavista, more than 250 mm was estimated to have fallen between the 20th and 21st. A confirmed 238 mm fell in St. Lawrence, ranking Igor as the third-wettest tropical cyclone in Canadian history [11]. Sustained wind speeds of 130 km/h were maintained with gusts reaching 172 km/h [32]. Igor produced storm tides off the coast up to 1.1 m. Additionally, massive offshore waves reaching heights up to 25.5 m were recorded [32]. Significant damage was sustained throughout Newfoundland due to the torrential rains,

leading to excessive runoff and flash flooding. In many rural areas entire bridges, roads and homes were destroyed. In some regions flood waters were higher than entire homes [31][34]. Roughly 150 communities were temporarily isolated as all roads leading to them had been destroyed or taken substantial damage [31]. Two severe cases were: a 30 m section of the Trans-Canada Highway being significantly eroded, disconnecting the main population of the island from the rest of Newfoundland, and a bridge washout on the Burin Peninsula leaving 20,000 people stranded [31][34]. High winds downing trees and power lines caused more damage to homes and left many without power. An estimated 50,000 residents were left without electricity [5]. Overall, the total losses were substantial reaching $200 million, ranking it as the costliest cyclone in Newfoundland history. A single Newfoundlander lost his life due to severe flooding causing a collapsed driveway and being dragged out to sea.

1.3 The Governing Equations

In this section will we discuss some thermodynamic background leading up to the governing equations. Section 1.4, 1.5, and 1.6 of [46] will be used for reference. We also give a subset of the equations used by WRF using [42] for reference.

1.3.1 Equation of State

Consider the dry, compressible equations of motion in 3D. Five equations are present (three momentum equations, the continuity equation, and the equation for energy), but there are 6 unknowns - three velocity components, pressure (p), density (ρ), and temperature (T). Clearly another equation is needed. The *equation of state* is an expression that diagnostically relates the various thermodynamic variables to each other. For example, the thermal equation of state is an expression that relates temperature, pressure, composition (the mass fraction of various components), and density. It can be written generally as

$$p = p(\rho, T, \phi_n), \tag{1.3.1}$$

where ϕ_n is the mass fraction of the nth constituent. This is not the most fundamental equation of state from a thermodynamic perspective, but it connects readily measurable

quantities. For an ideal gas (air in the atmosphere is essentially ideal) the thermal equation of state is

$$p = \rho R T, \tag{1.3.2}$$

where R is the ideal gas constant. R is a specific constant, and is related to the universal gas constant R^* by $R = R^*/\bar{\mu}$, where $\bar{\mu}$ is the mean molar mass of the constituents of the gas. We can also relate R to the number of molecules per unit mass n_m and Boltzmann constant k_B via $R = n_m k_B$, hence R is proportional to the number of molecules contained in a unit mass. Air has essentially constant composition except for variations in water vapour. This makes the gas constant in the equation of state have a weak dependence on the water vapour content, but we can often regard R as constant.

1.3.2 Thermodynamic Relations

A fundamental postulate of thermodynamics is that the internal energy of a system at equilibrium is a function of volume, entropy, and the mass of various constituents - the extensive properties. Extensive meaning the value is proportional to the amount of material present. This is in contrast to something intensive such as temperature. We consider each of these quantities as divided by the mass of the fluid present, so expressing the internal energy per unit mass I as a function of the specific volume $\alpha = \rho^{-1}$, the specific entropy η, and the mass of fractions of its various components. We are interested in two-component fluids (i.e. dry air and water vapour) so we may parametrize the composition by a single parameter, q, representing the water vapour mixing ratio. Standard thermodynamic notation is used, except that I is internal energy over u, since u is typically fluid velocity, and η instead of S for entropy, to be consistent with [46]. Then we can write the entropy in terms of internal energy, density, and q as

$$I = I(\alpha, \eta, q) \qquad \text{or} \qquad \eta = \eta(I, \alpha, q). \tag{1.3.3}$$

Either of these expressions gives a complete description of the macroscopic state of a system in equilibrium, and we call either of them the *fundamental equation of state.* We can derive the thermal equation of state from 1.3.3, but not vice versa. The first differential is (we

consider the leftmost equation)

$$dI = \left(\frac{\partial I}{\partial \eta}\right)_{\alpha,q} d\eta + \left(\frac{\partial I}{\partial \alpha}\right)_{\eta,q} d\alpha + \left(\frac{\partial I}{\partial q}\right)_{\alpha,\eta} dq. \tag{1.3.4}$$

The subscript notation indicates a parameter being held constant. We can give a physical interpretation for each of these differentials. Conservation of energy tells us internal energy of a body may change due to three things: work down by or on it, heat input, or a change in its chemical composition (e.g. salinity or water vapour content). We write this as

$$dI = đQ + đW + đC, \tag{1.3.5}$$

where $đW$ is the work done by the body, $đQ$ is the heat input to the body, and $đC$ is the change in internal energy caused by a change in chemical composition. We call this 'chemical work.' The infinitesimal quantities denoted by $đ$ are so-called inexact differentials. The use of inexact differentials is that Q, W, and C are not functions of the state of body, and the internal energy cannot be regarded as the sum of a 'heat' and 'work'. Thus we think of them only as fluxes of energy, or rates of energy input - not as amounts of energy. Their sum changes the internal energy of a body, which *is* a function of its state. It is possible to rewrite 1.3.5 in a form known as the *fundamental thermodynamic relation*:

$$dI = Td\eta - pd\alpha + \mu dq, \tag{1.3.6}$$

where μ represents the chemical potential. The fundamental equations of state (1.3.3) describe the properties of a particular fluid, while the fundamental thermodynamic relation (1.3.6) is associated with conservation of energy. Much classical thermodynamics follows from these two expression.

1.3.3 Thermodynamic Equations for Fluids

The fundamental thermodynamic relation can be applied to identifiable bodies or system, hence heat input affects the fluid it is applied to, and we can apply the material derivative to the thermodynamic relations to obtain the equations of motion. We make two assumptions:

i The fluid is in thermodynamic equilibrium locally.

ii Macroscopic fluid motions are reversible and so not entropy producing.

The first point implies that quantities like temperature, pressure, and density can be locally related by thermodynamic relationships. Using equation 1.3.5, conservation of energy for an infinitesimal fluid parcel can be written as

$$dI = -pd\alpha + đQ_E, \tag{1.3.7}$$

where $đQ_E$ is the total energy input from heating and change in composition. Based on our first assumption, we may take a material derivative to obtain

$$\frac{DI}{Dt} + p\frac{D\alpha}{Dt} = \dot{Q_E}, \tag{1.3.8}$$

where $\dot{Q_E}$ is the rate of total energy input, per unit mass, with possible contributions from thermal fluxes and fluxes of composition. We can use the mass continuity equation in the form $D\alpha/Dt = \alpha\nabla \cdot \mathbf{u}$, to write

$$\frac{DI}{Dt} + p\alpha\nabla \cdot \mathbf{u} = \dot{Q_E}. \tag{1.3.9}$$

This is the *internal energy equation* for a fluid. $\dot{Q_E}$ in general contains energy fluxes due to changes in composition, so we need to know said composition. The composition of a fluid parcel is carried with it as it moves, and changes only if there are non-conservative sources and sinks. Thus its evolution can be determined by

$$\frac{Dq}{Dt} = \dot{q}, \tag{1.3.10}$$

where $\dot{q}$ represents all non-conservative terms such as condensation or evaporation for vapour. We obtain the *entropy equation* by taking the material derivative of 1.3.5, and applying equations 1.3.9 and 1.3.10 arriving at

$$\frac{D\eta}{Dt} = \frac{1}{T}\dot{Q_E} - \frac{\mu}{T}\dot{q} \equiv \frac{1}{T}\dot{Q}, \tag{1.3.11}$$

where $\dot{Q}$ gives the heating rate per unit mass. The entropy equation is not independent of internal energy, rather it is related by the thermodynamic relations and the equations of state. By using the internal energy equation we can indeed calculate the entropy via the equation of state $\eta = \eta(I, \alpha, q)$ and vice versa. However, the heating term $\dot{Q}$ is not always easy to accurate determine in practice as it is affected by gradients of composition, viscosity, and latent heating; thus, the internal energy equation may be more straightforward to use.

Potential Temperature

It is common in meteorology to express the entropy in terms of the potential temperature θ, which is defined as the temperature that a fluid would have if it moved adiabatically and at constant composition to some reference pressure (usually 1000 hPa - approximately Earth's surface pressure). Thus, adiabatic flow in the potential temperature obeys $D\theta/Dt = 0$. To derive an expression for θ we start with the first law of thermodynamics for an ideal gas, using the fact that internal energy becomes a function of temperature only $dI = c_v dT$ (c_v is the heat capacity at constant volume). Using this with 1.3.5 and 1.3.6 gives

$$đQ = c_p dT - \alpha dp, \tag{1.3.12}$$

where c_p is the heat capacity at constant pressure. Now when a fluid parcel changes pressure adiabatically, it expands or contracts and, using 1.3.12 with $đQ = 0$, we see the temperature change is determined via $c_p dT = \alpha dp$. We use equation 1.3.12 and the equation of state for an ideal gas to relate θ to other thermodynamic variables

$$d\eta = c_p d\ln T - R d\ln p. \tag{1.3.13}$$

If we move adiabatically ($d\eta = 0$) from p to p_R the temperature changes, by definition from T to θ. Taking the integral of 1.3.13 under these bounds can be solved with constant c_p and R to give

$$\theta = T\left(\frac{p_R}{p}\right)^{\kappa}, \tag{1.3.14}$$

where p_R is the reference pressure and $\kappa \equiv R/c_p$. Thus, it follows from the previous two equations that potential temperature is related to entropy via

$$d\eta = c_p d\ln\theta, \tag{1.3.15}$$

and if c_p is constant (which it nearly is for Earth's atmosphere),

$$\eta = c_p \ln\theta. \tag{1.3.16}$$

We finally arrive at our last governing equation by taking the material derivative of 1.3.16 and using 1.3.11 to get

$$c_p \frac{D\theta}{Dt} = \frac{\theta}{T}\dot{Q}. \tag{1.3.17}$$

1.3.4 The Set of Equations

Of course the first two equations in our set of governing equations for dry air are the evolution equations for the density

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1.3.18}$$

and velocity

$$\frac{D\mathbf{u}}{Dt} + 2\boldsymbol{\Omega} \times \mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}. \tag{1.3.19}$$

Equation 1.3.18 is known as the *mass continuity equation*. It states that the rate at which mass enters a system is equal to the rate at which mass leaves plus the accumulation of mass within. The time derivative can be interpreted as the accumulation (or loss) of mass in the system, while the divergence term represents the difference of flow in versus flow out. Equation 1.3.19 is known as the *momentum equation*, which describes how the velocity of a fluid responds to internal or imposed forces. $\mathbf{F}$ represents external body forces such as gravity, ∇p is the pressure gradient force, while the Laplacian term represents the viscous force. The second term on the left hand side represents the Coriolis acceleration due to Earth's rotation. To complete our set of equations we add several of the previously discussed thermodynamic equations:

$$p = \rho R T, \qquad \theta = T\left(\frac{p_R}{p}\right)^{\kappa}, \qquad c_p \frac{D\theta}{Dt} = \frac{\theta}{T}\dot{Q}.$$

Note that we now have a closed set of equations. There are 7 equations present (three momentum equations), and 7 unknowns: u, v, w, ρ, p, T, and θ.

WRF Equations with Moisture

For reference we give the Euler equations as used by WRF which include moist thermodynamic effects. Note that WRF does not use height as a vertical coordinate. Rather they implement a terrain-following hydrostatic pressure vertical coordinate that was first proposed by Laprise (1992) [22]. The idea is that using this coordinate, WRF can more

effectively handle rough terrain on the bottom boundary. The vertical coordinate is given by

$$\zeta = (p_{dh} - p_{dht})/\mu_d, \tag{1.3.20}$$

where μ_d represents the mass of dry air in a column and p_{dh} and p_{dht} represent the hydrostatic pressure of the dry atmosphere and the hydrostatic pressure at the top of the dry atmosphere. The full set of equations are

$$\partial_t U + (\nabla \cdot \boldsymbol{V} u) + \mu_d \alpha \partial_x p + (\alpha/\alpha_d)\partial_\zeta p \partial_x \phi = F_U \tag{1.3.21}$$

$$\partial_t V + (\nabla \cdot \boldsymbol{V} v) + \mu_d \alpha \partial_y p + (\alpha/\alpha_d)\partial_\zeta p \partial_y \phi = F_V \tag{1.3.22}$$

$$\partial_t W + (\nabla \cdot \boldsymbol{V} w) - g[(\alpha/\alpha_d)\partial_\zeta p - \mu_d = F_W \tag{1.3.23}$$

$$\partial_t \Theta + (\nabla \cdot \boldsymbol{V} \theta) = F_\Theta \tag{1.3.24}$$

$$\partial_t \mu_d + (\nabla \cdot \boldsymbol{V}) = 0 \tag{1.3.25}$$

$$\partial_t \phi + \mu_d^{-1}[(\boldsymbol{V} \cdot \nabla \phi) - gW] = 0 \tag{1.3.26}$$

$$\partial_t Q_m + (\nabla \cdot \boldsymbol{V} q_m) = F_{Q_m}, \tag{1.3.27}$$

with the diagnostic equation for dry inverse density

$$\partial_\zeta \phi = -\alpha_d \mu_d, \tag{1.3.28}$$

and the equation of state for the full pressure (water vapour plus dry air)

$$p = p_0 (R_d \theta_m / p_0 \alpha_d)^\gamma. \tag{1.3.29}$$

There are a slue of additional terms due to the vertical coordinate. We give a breakdown of any new symbols. A subscript d refers to dry air specificity. There are several coupled variables defined as $\boldsymbol{V} = \mu_d \boldsymbol{u} = (U, V, W)$ and $\Theta = \mu_d \theta$. The geopotential is $\phi = gz$. F_* represents all the forces on the right hand side of the momentum equation. $\gamma = 1.4$ is a constant and $Q_m = \mu_d q_m$, where $q_m = q_v, q_c, q_r, \ldots$ represents any mixing ratio (mass per mass of dry air) for water vapour, clouds, rain, etc. Finally, $\theta_m \approx \theta(1 + 1.61 q_v)$.

1.4 Objectives

The main objectives of this work are:

- Successfully simulate Hurricane Igor using WRFs real data capabilities with the 2D horizontal Smagorinsky model.
- Implement the DSM and compare results - specifically energy spectra - to determine if the DSM's has made any changes such as decreasing dissipation. Additionally, is it worth the extra computational cost?

Chapter 2

Large-Eddy Simulation

This chapter aims to cover all necessary background information on the computational approach for turbulence: LES. It will also provide insight on the DSM and why we are using it. We start by discussing a major hurdle in turbulence modelling (cost) and why DNS is usually not feasible. Then we discuss the mechanics: the filtering operation, and modelling the subgrid-scale (SGS) stress tensor. Finally we discuss the concept of scale invariance and why it motivates the use of the DSM.

2.1 Direct Numerical Simulation

A major challenge in turbulence modelling is cost. Technology and computers have continued to improve, but we are still far from a time where turbulence models will be unnecessary to save cost. While using DNS would be ideal, it is often far from realistic. Using DNS, the Navier-Stokes Equations (NSE) are numerically solved without the use of any turbulence model. This means all ranges of spatial and temporal scales of the turbulence are resolved completely. In particular the spatial scales of the turbulence must be resolved down to the smallest energy dissipating eddies, the Kolmogorov scale given by $\eta = (\nu^3/\epsilon)^{1/4}$ where ν is the kinematics viscosity and ϵ is the kinetic energy dissipation. In the atmosphere the Kolmogorov is as small as $\eta \approx 1$ mm. A simple analysis using the Kolmogorov and integral

scale shows that a three-dimensional DNS requires a massive amount of grid points. If L is the length scale then the number of grid points in each cardinal direction is

$$N_x \sim N_y \sim N_z \sim \frac{L}{\eta} = \mathrm{Re}^{3/4}, \tag{2.1.1}$$

where $\mathrm{Re} = uL/\nu$ is the Reynolds number (u is the velocity scale). Thus, the total number of grid points N^3 is of order $\mathrm{Re}^{9/4}$ (chapter 9 of Pope [33]). This easily becomes overwhelming in large scale atmospheric flows. For example the kinematic viscosity of air is approximately $\nu = 1.5 \times 10^{-5}$ m^2/s. Using a relatively small velocity scale of $u = 2.5$ m/s and $L = 50$ m we get $\mathrm{Re} = 8.3 \times 10^6$. On top of this, the integration for the solution in time must be done using an explicit method. Thus for accuracy, the integration often needs to be done using a time step, Δt, small enough such that a fluid particle only moves a fraction of the grid spacing h each step. In other words, the Courant-Friedrichs-Lewy (CFL) condition [8] must be met:

$$C = \frac{u\Delta t}{h} < 1.$$

The number of time steps also grows at a rate $N_t \sim \mathrm{Re}^{3/4}$ [33]. Thus, we can get an estimate on the number of floating point operations required to complete the simulation as $\mathrm{Re}^{9/4} * \mathrm{Re}^{3/4} = \mathrm{Re}^3$.

DNS is very expensive, even at low Re. Hence the need for turbulence models such as RANS and the aforementioned LES. If DNS is one extreme where all scales are resolved, RANS is at the opposite spectrum where a large portion of the flow is modelled. This saves on computational cost greatly at the downfall of accuracy. LES provides a powerful compromise between the two. This is illustrated quite eloquently in a figure given on slide 4 of [6] which depicts a range of scales showing what is resolved versus modelled for each scheme.

2.2 Filtering

We use Pope [33] as our reference for this subsection. LES is a powerful tool able to accurately simulate turbulent flows spanning vast length scales. As stated, LES provides

an effective compromise between computational costs and accuracy of simulations. To achieve this, LES separates the largest scale eddies from the smallest energy containing eddies to be resolved directly. Then the smallest (or SGS) eddies are filtered, and the effect of the sub-filter scale is modelled via a mathematical formulation called a turbulence model. In the present section we discuss the procedure of filtering. There are four conceptual steps involved in LES:

1. A filtering operation which decomposes the velocity $\boldsymbol{U}(\boldsymbol{x}, t)$ into the sum of the resolved (or filtered) component $\boldsymbol{u}(\boldsymbol{x}, t)$ and the SGS (or residual) component $\boldsymbol{u'}(\boldsymbol{x}, t)$.

2. The equations of motion for the filtered velocity field are derived from the NSE, where the addition of subgrid velocities give rise to the SGS stress tensor.

3. The modelling of the SGS stress tensor via a turbulence model, obtaining closure.

4. The model filtered equations are solved numerically for $\boldsymbol{u}(\boldsymbol{x}, t)$, providing an approximate solution to the large scales of flow.

Note that we are using (bold) $\boldsymbol{x}$ to indicate a vector. The most important step in this process is the filtering. Recall that DNS must resolve the velocity field $\boldsymbol{U}(\boldsymbol{x}, t)$ down to the Kolmogorov scale leading to a cost in memory of $\mathrm{Re}^{9/4}$. However, with LES the filtered velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ can be adequately resolved on a relatively coarse grid, with a reduced cost of log(Re) in the right conditions (for instance exercise 13.29 of Pope [33]). This is a substantial improvement. More specifically, the required grid spacing should be proportional to the specified filter width. The filter width should be slightly smaller than the size of the smallest energy containing motions to ensure everything is resolved effectively. Filtering in this way gives us a large grid spacing to reduce cost, while still being small enough to resolve the energy containing motions.

The general filtering operation is defined as

$$\boldsymbol{u}(\boldsymbol{x}, t) = \iiint_V G(\boldsymbol{r}, \boldsymbol{x}) \boldsymbol{U}(\boldsymbol{x} - \boldsymbol{r}, t)\, d\boldsymbol{r}, \tag{2.2.1}$$

where G is a specified filter function satisfying

$$\int G(\boldsymbol{r}, \boldsymbol{x})\, d\boldsymbol{r} = 1. \tag{2.2.2}$$

We will analyse the properties of this filter in 1D as it is easier and the extension to 3D is straightforward. Thus we let $U(x)$ be the velocity field and $G = G(r)$ be a homogeneous filter. Then the filtered velocity is given by the convolution

$$u(x) = \int_{-\infty}^{\infty} G(r)U(x-r)\, dr. \tag{2.2.3}$$

An example of some of the most common filters are the box filter or the sharp spectral filter, defined as $G(r) = \frac{1}{\Delta}H(0.5\Delta - |r|)$ and $G(r) = \frac{1}{\pi r}\sin(\pi r/\Delta)$ respectively. The transfer function to these filters, defined by

$$\hat{G}(k') = \int_{-\infty}^{\infty} e^{ik'r}G(r)\, dr, \tag{2.2.4}$$

are $\hat{G}(k') = \frac{1}{0.5k'\Delta}\sin(0.5k'\Delta)$ and $\hat{G}(k') = H(k_c - |k'|)$, where H is the Heaviside step function, Δ is the grid spacing, k' is the wave number, and $k_c = \pi/\Delta$. The simplest filter is the box filter, which is just an average of $U(x')$ in the interval $x - 0.5\Delta < x' < x + 0.5\Delta$. In a later section we discuss the use of a simple box filter with the two-dimensional trapezoid rule to implement the DSM.

We can examine the sharp spectral filter in wavenumber space to make its effects more clear. Suppose $U(x)$ has a Fourier transform

$$\hat{U}(k') = \mathcal{F}(U(x)).$$

From (2.2.3) and the properties of convolution we see the filtered velocity will have the Fourier transform

$$\begin{aligned}\hat{u}(k') &= \mathcal{F}(u(x)) \\ &= \hat{G}(k')\hat{u}(k'),\end{aligned}$$

where the transfer function is equal to 2π times the Fourier transform of the filter. I.e. $\hat{G}(k') = 2\pi\mathcal{F}(G(r))$. Recalling that $\hat{G}(k') = H(k_c - |k'|)$, we see the significance of the sharp spectral filter in that it eliminates all Fourier modes of wavenumber $|k'|$ greater than the cutoff wavenumber k_c, while having no effect on the lower wavenumber modes. The spectra of turbulent flows is central to LES SGS models, which reconstruct the effects of the high frequency wave numbers. A challenge in subgrid modelling is effectively being able to represent the cascade of kinetic energy from the high to low wave numbers. This makes the spectral properties of the implemented filter very important to the SGS modelling. While the sharp spectral filter is sharp in wavenumber space, it is decidedly non-local in physical scape. The opposite can be said of the box filter. A filter that can be used which is relatively compact in both spaces is the Gaussian filter, defined by

$$G(r) = \left(\frac{6}{\pi\Delta^2}\right)^{0.5} \exp\left(-\frac{6r^2}{\Delta^2}\right),$$

with

$$\hat{G}(k') = \exp\left(-\frac{k'^2\Delta^2}{24}\right).$$

Along with satisfying (2.2.2), LES filters which are to be applied to the NSE must also satisfy the property of linearity, and for homogeneous filters, commutation with derivatives [39]. If $\langle * \rangle$ is the filtering operation, then

$$\langle U + V \rangle = \langle U \rangle + \langle V \rangle,$$

and

$$\frac{\langle \partial U \rangle}{\partial s} = \frac{\partial \langle U \rangle}{\partial s},$$

for both $s = \boldsymbol{x}$ and $s = t$.

2.3 LES Equations

The Navier-Stokes Equations represent the motion of fluid flow from the largest to smallest scales,

$$\frac{\partial U_i}{\partial t} + \frac{\partial U_i U_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \nu\frac{\partial}{\partial x_j}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \tag{2.3.1}$$

$$\frac{\partial U_i}{\partial x_i} = 0, \tag{2.3.2}$$

where U_i and P represent the fluid velocity and pressure, while ρ and ν are the density and kinematic viscosity [21]. Note that we have both omitted the Coriolis term from equation 1.3.19 and are assuming incompressibility for simplicity. These equations are difficult to solve. There is no known analytical solution and as discussed in section 2.1, are challenging to simulate directly. Instead we use LES as a cost effective, accurate approach. LES solves the filtered NSE such that the velocities are decomposed as discussed in the previous section. To obtain the filtered equations we substitute the decomposition of the velocity field into equations 2.3.1 and 2.3.2 and filter, obtaining

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \overline{u_i u_j}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{P}}{\partial x_i} + 2\nu \frac{\partial \bar{S}_{ij}}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}, \tag{2.3.3}$$

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \tag{2.3.4}$$

and

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j, \tag{2.3.5}$$

where τ_{ij} is the SGS stress tensor grouping all unclosed terms. Note the over-bar indicates the filtering. We write the diffusion term in 2.3.1 in terms of the filtered strain rate tensor $S_{ij} = (\partial u_i/\partial x_j + \partial u_j/\partial x_i)/2$. The non-linear advection term in equation 2.3.1 is a major cause of difficulty in LES as it requires knowledge of the residual velocity field, which is unknown, and hence must be modelled. This term gives rise to the SGS stress tensor as written in 2.3.5. Leonard [23] decomposed the stress tensor such that $\tau_{ij} = L_{ij} + C_{ij} + R_{ij}$ and provided physical interpretations for each term. $L_{ij} \equiv \overline{\bar{u}_i \bar{u}_j} - \bar{\bar{u}}_i \bar{\bar{u}}_j$ is known as the Leonard tensor, and represents interaction between large scale flow. $R_{ij} \equiv \overline{u'_i u'_j} - \bar{u}'_i \bar{u}'_j$ is the Reynolds stress-like term, and represents interactions among the subgrid-scales. Finally $C_{ij} \equiv \overline{\bar{u}_i u'_j} + \overline{u'_i \bar{u}_j} - \bar{\bar{u}}_i \bar{u}'_j - \bar{u}'_i \bar{\bar{u}}_j$, the Clark tensor [12], represents cross interaction between small and large scales. The fact that τ_{ij} represents interactions between both resolved and SGS velocities is what makes effectively modelling it challenging, but it is the sole goal of SGS models.

The SGS stress tensor τ_{ij} represents the effect of the subgrid-scale eddies on the larger eddies and needs to be modelled via a turbulence model. To begin the discussion we note that there are two types of unresolved scales: resolved sub-filter-scale (SFS)s and SGSs. The distinction is that SFSs represent the wave numbers greater than the cutoff wave number k_c, but whose effects are dampened by the filter. As mentioned in the section on filtering, these only exist when filters non-local in wavenumber space are used (such as the box or Gaussian filter). We are concerned with the SGSs, that is, any scale smaller than the cutoff filter width Δ. The type of SGS model used can depend on the filter chosen. The type of model we are considering is known as a functional model, focusing only on dissipating energy at a rate that is physically correct. These assume what is known as an eddy viscosity assumption, where the effect of turbulence are lumped into a turbulent viscosity. Some limitations of this assumption are discussed in section 6.6.4 of Wyngaard (2010) [47]. Notably that the diagonal components of SGS deviatoric stress are largely maintained a tilting term (see equation (6.81) of Wyngaard), which is something our assumption lacks. The SGS stress tensor is related to the resolved strain tensor via

$$\tau_{ij} = -2\nu_t S_{ij}. \tag{2.3.6}$$

With this approximation the only unknown is ν_t, the turbulent viscosity. The first model for ν_t was given by Smagorinsky (1963)[43] and first implemented by Deardorff (1970)[13]. It models the turbulent viscosity as

$$\nu_t = (c_s\Delta)^2\sqrt{2S_{ij}S_{ij}} = (c_s\Delta)^2|S|, \tag{2.3.7}$$

where $\Delta = \sqrt[3]{\Delta x \Delta y \Delta z}$ is the filter width and c_s is the universal Smagorinsky constant. Recall that in the present case we are implementing 2D horizontal Smagorinsky using WRF, in which the magnitude of the strain is instead given by

$$|S| = ((S_{11} - S_{22})^2 + S_{12}^2)^{1/2} \tag{2.3.8}$$

[42]. This formulation in 2.3.8 is originally given by equation (4.23) of [43]. The Smagorinsky models' simplicity and consistency with conservation laws is perhaps why it is so widely used; however, it is not without its deficits. For instance, the Smagorinsky coefficient c_s being held constant may not be a good approximation as it's expected to vary throughout

the fluid. Pope [33] goes in some detail about these deficits. In particular a posteriori test concludes that Smagoirsnky is too dissipative - i.e. it transfers too much energy to the residule motions. Schaefer-Rolffs and Becker (2013) [40] argue that this deficit may be related to equation 2.3.7, i.e. the Smagorinsky model, is not scale invariant [29]. There have been studies done to improve the Smagorinsky model without concern for scale invariance e.g. Lévêque et al. [24]; however, we will focus on what scale invariance is and why it motivates the DSM.

2.3.1 LES in WRF

It is typically thought that LES is a turbulence model used on domains no larger than a few tens of kilometres. In the present case our horizontal length scale is already on the top end of that range. Thus, it may be correct to say there is no "real" LES within our simulations. However, the mechanisms WRF uses to compute the turbulent mixing and eddy viscosity are highly similar. The difference is largely that WRF computes the horizontal and vertical mixing length separately. The WRF manual [42] provides us with the equations for horizontal diffusion in physical space

$$\partial_t u = \cdots - m_x[\partial_x \tau_{11} + \partial_y \tau_{12} - \partial_z(z_x \tau_{11} + z_y \tau_{12})] - \partial_z \tau_{13}, \tag{2.3.9}$$

$$\partial_t v = \cdots - m_y[\partial_x \tau_{12} + \partial_y \tau_{22} - \partial_z(z_x \tau_{12} + z_y \tau_{22})] - \partial_z \tau_{23}, \tag{2.3.10}$$

$$\partial_t w = \cdots - m_z[\partial_x \tau_{13} + \partial_y \tau_{23} - \partial_z(z_x \tau_{13} + z_y \tau_{23})] - \partial_z \tau_{33}. \tag{2.3.11}$$

The additional terms of the momentum equation are excluded via the ... to save space. Note the additional variables m_x, z_x, etc. are a byproduct of the fact that WRF does not use a flat vertical coordinate. The punchline is that τ_{ij} no longer has a single ansatz that relates it to the strain (e.g. equation 2.3.6). Rather for components of the stress containing a subscript '3', the stress tensor is given by a vertical eddy viscosity (for example)

$$\tau_{13} = -2\mu_d K_v S_{13},$$

and in any other case $i, j \neq 3$, a horizontal eddy viscosity

$$\tau_{ij} = -2\mu_d K_h S_{ij}.$$

These two equations highly resemble that of 2.3.6 and K_h is identically equal to equation 2.3.7. K_v is then externally defined as a constant in our input.

2.4 Scale Invariance

Scale invariance is a feature of objects or laws that do not change if scales of length, energy, or other variables are multiplied by a common factor. Conservation laws are universal and independent from the effective length scale of the processes considered. For instance conservation of energy is connected to an invariance of the equations of motion with respect to a translation in time [41]. In contrast, scale invariance does not hold for the whole spectral range of scales and there is no conserved quantity that does not violate scale invariance for every scale [41]. Consider the turbulent kinetic energy cascade in the atmosphere, where there is a wide range of scales from planetary to micro-scales where viscosity becomes important. There is no single scale-invariant subrange that covers this. Instead the energy spectrum is divided into subranges where different terms of the governing equations are dominant. This is illustrated is figure 1 of Schaefer-Rolffs et al. (2014) [41] which depicts wavelengths in the range $[10^4, 10^{-4}]$ versus spectral density. There is a clear change in profile in spectral density between the non-inertial subranges. In the isotropic inertial subrange (which direclty precedes the viscous subrange), Kolmogorov's spectral theory (see chapter 8 of [45]) assumes a large inertial range between the scales of injection and dissipation of kinetic energy. In this range kinetic energy is transferred to smaller and smaller scales. This holds true as long as there is self-similarity within the inertial ranges. In other words, the NSE are only scale invariant if the external forcing and internal friction are negligible for scales in the inertial range. Additional evidence of the Smagorinsky model's shortcomings due to a lack of scale invariance are shown in figure 2 of the same paper. A kinetic energy spectrum is pictured for the Smagorinsky model at different mixing lengths $l_h = (c_s\Delta)$. Depending on the value of l_h, the model fails to accurately represent the spectrum at different wave number ranges. For instance at $l_h = 11.6$km the spectrum is only reasonably represented at the synoptic subrange (wavenumbers less than 100 m^{-1}). Conversely, the DSM, which obeys scale invariance, is reasonable for all range of scales pictured.

A simple way to see that the Smagorinsky model is not scale invariant is as follows. By equation 2.3.5 τ_{ij} is a difference of squared velocities and hence should be modelled as such. However using the Smagorinsky ansatz we have $\tau_{ij} = 2l_h^2|S|S_{ij}$ where the mixing length does not contribute to scaling. I.e. the modelled τ_{ij} scales with squared strain and therefore is not scale invariant. Schaefer-Rolffs et al. (2014) [41] have developed a more sophisticated approach. They define the scaling transformation of a variable as

$$a^* = e^{c_a} a \tag{2.4.1}$$

with c_a being the scaling factor. A relationship between temporal and spatial scaling factors under which scale invariance is held is obtained. To do so we consider length, speed and (advective) time scales: L, U, and T. Assume a constant spectral energy flux $\epsilon^* = \epsilon =$ constant (equivalently $c_\epsilon = 0$). Given that $\epsilon \sim L^2/T^3$ we arrive at the relation $0 = c_\epsilon = 2c_x - 3c_t$. Equivalently $c_t = \frac{2}{3}c_x$. This scaling transformation must not be violated by any other transformation of the momentum equations. An example showcasing how the NSE lead to a second relationship violating scale invariance is as follows. Consider a flow with characteristic scales as mentioned above. To ensure the same behaviour at different scales, one has to modify both L and T simultaneously to ensure $\mathrm{Re} = UL/\nu =$ constant [30]. An assumption of constant viscosity and a simple analysis of the equations $\mathrm{Re} = L^2/T\nu$ and $U = L/T$ lead to a second relationship $c_t = 2c_x$. This contradicts scale invariance as the only solution satisfying both relationships is $c_t = c_x = 0$. In short, in LES a scaling transformation of the equations of motion must not lead to a relationship between spacial and temporal parameters that contradicts $c_t = \frac{2}{3}c_x$. To indeed show that the Smagorinsky model violated this relationship, a condition to determine scale invariance was developed. Defining

$$\mathcal{G}_a(t, x_i, a, b_l) \equiv e^{\frac{2}{3}c_x - c_a} \mathcal{F}_a(e^{\frac{2}{3}c_x} t, e^{c_x} x_i, e^{c_a} a, e^{c_{b_l}} b_l), \tag{2.4.2}$$

where $\mathcal{F}$ is the right hand side of the momentum equation

$$\partial_t a + (\mathbf{v} \cdot \nabla) a = \mathcal{F}_a(t, x_i, a, b_1, b_2, ...),$$

Schaefer-Rolffs et al. (2014) [41] developed the condition

$$\mathcal{G}_a(t, x_i, a, b_l) = \mathcal{F}_a(t, x_i, a, b_l) \tag{2.4.3}$$

to ensure scale invariance is held. Then considering the eddy viscosity for the Smagorinsky model

$$\nu_t = l_h^2 |S|,$$

the scaling factor for the eddy viscosity can be written as

$$c_{\nu_t} = 2c_{l_h} - \frac{2}{3}c_x.$$

As we've previously mentioned the Smagorinsky coefficient c_s is constant in the classical model. A consequence is that the mixing length is constant such that $l_h^* = l_h$. Equivalently $c_{l_h} = 0$ (refer to equation 2.4.1). This gives a relationship based on equations 2.4.2 and 2.4.3

$$\mathcal{G}_{v,2} = e^{-2c_x} l_h^2 \nabla(|S|S) \neq \mathcal{F}_{v,2}, \tag{2.4.4}$$

where the equality cannot hold unless $c_x = 0$. However, doing so would imply no scaling transformation could be applied. This is a classical result of the constant mixing length and was first pointed out by Oberlack (1997) [28].

2.5 Dynamic Smagorinsky Model

It is clear that there are deficits with the classical Smagorinsky model - the violating of scale invariance being the highlight. As previous analysis has shown, it appears the assumption of a constant mixing length; that is, that the Smagorinsky constant c_s is unchanging throughout the flow is the culprit. Then one would like to develop an alternative to the Smagorinsky model which has variable mixing length and thus complies with scale invariance. That is exactly what the DSM entails. The DSM was originally developed in 1991 by Germano et al. [17]. They investigated several studies using the classic Smagorinsky model and found different values for c_s depending on the flow configurations. Thus the need to develop a model which computes c_s at run time.

To do this we introduce an additional filter (hereafter: test filter) with size greater than the LES filter: $\tilde{\Delta} > \bar{\Delta}$. A typical approach lets $\tilde{\Delta} = 2\bar{\Delta}$, but it should remain that $\tilde{\Delta}$ and

$\bar{\Delta}$ lie within the same turbulent inertial range. Germano et al. (1991) [17] initially derived a scheme for a sharp cutoff filter in a grid-point model. Shortly after, Germano (1992) [16] showed this approach is applicable to a variety of explicit filters such as spectral cutoff and Gaussian. In similarity to equation 2.3.5, we define the turbulent stress due to scales smaller than $\tilde{\Delta}$ as

$$T_{ij} = \widetilde{\overline{u_i u_j}} - \tilde{\bar{u}}_i \tilde{\bar{u}}_j. \tag{2.5.1}$$

Next we introduce the Germano identity [17]. This represents the turbulent stress due to scales between $\bar{\Delta}$ and $\tilde{\Delta}$ acting on scales larger than $\tilde{\Delta}$:

$$L_{ij} \equiv T_{ij} - \widetilde{\tau_{ij}} = \widetilde{\overline{u_i u_j}} - \tilde{\bar{u}}_i \tilde{\bar{u}}_j - \widetilde{\overline{u_i u_j}} + \widetilde{\bar{u}_i \bar{u}_j} = \widetilde{\bar{u}_i \bar{u}_j} - \tilde{\bar{u}}_i \tilde{\bar{u}}_j. \tag{2.5.2}$$

Equation 2.5.2 is expressed in terms of the resolved eddies, thus can be calculated directly from resolved scales. Then we make the same eddy viscosity assumption on T_{ij} and we did τ_{ij} and parametrize via the same ansatz. That is, we can express T_{ij} as

$$T_{ij} \approx -2(c_T \tilde{\Delta})^2 \widetilde{|\bar{S}|\bar{S}_{ij}}, \tag{2.5.3}$$

with Smagorinsky constant c_T for the test filter $\tilde{\Delta}$. Substituting eddy viscosity approximations for both T_{ij} and τ_{ij} into equation 2.5.2, we obtain the approximation relation:

$$\begin{aligned} L_{ij} &\approx -2[(c_T \tilde{\Delta})^2 \widetilde{|\bar{S}|\bar{S}_{ij}} - [(c_s \bar{\Delta})^2 |\bar{S}|\bar{S}_{ij}]^{\sim}] \\ &= -2c_s^2 \left[\left(\frac{c_T}{c_s} \right)^2 \tilde{\Delta}^2 \widetilde{|\bar{S}|\bar{S}_{ij}} - \bar{\Delta}^2 \widetilde{|\bar{S}|\bar{S}_{ij}} \right] \equiv c_s^2 M_{ij}, \end{aligned}$$

where $[...]^{\sim}$ indicates the test filter applied to everything within the square brackets. Note the second equality only holds if $\bar{\Delta}$ and c_s^2 are slowly varying at the scale of the test filter. This obviously holds since $\bar{\Delta}$ is numerically constant. We require the assumption $\widetilde{c_s^2} \approx c_s^2$ without mathematical justification (Ronchi et al. 1992) [36]. We also assume the classical Smagorinsky constant is comparable at the test filter and resolution scale, i.e. $c_T^2 = c_s^2$. Thus, we can compute the dynamic Smagorinsky parameter from

$$L_{ij} = c_s^2 M_{ij} \qquad \text{with} \qquad M_{ij} = -2[\tilde{\Delta}^2 \widetilde{|\bar{S}|\bar{S}_{ij}} - \bar{\Delta}^2 \widetilde{|\bar{S}|\bar{S}_{ij}}], \tag{2.5.4}$$

and the resulting turbulent stress tensor can be written as

$$(\tau_{ij})_{\text{DSM}} = -2c_s^2\bar{\Delta}^2|\bar{S}|\overline{S_{ij}}, \tag{2.5.5}$$

where c_s is now being computed from equation 2.5.4. A simple analysis will show scale invariance is no longer being violated. We have L_{ij} scaling with squared velocity from equation 2.5.2, while M_{ij} scales with squared shear, thus c_s scales with squared length. In short, $(\tau_{ij})_{\text{DSM}}$ scales analogously to τ_{ij} with squared velocity. We can apply the methodology from Schaefer-Rolffs et al. (2014) [41] and achieve the same result. It is no longer the case that $c_{l_h} = 0$ and consequently, $l_h^* \neq l_h$. Indeed, as c_S scales with squared length we have $l_h^2 \sim c_s^2$ giving the relationship

$$c_{l_h} = c_x.$$

This corresponds to the transformed turbulent diffusion term

$$\mathcal{G}_{v,2} = e^{-2c_x}\nabla[(e^{c_x}l_h)^2|S|S]$$

which reduces to

$$\mathcal{G}_{v,2} = \nabla(l_h^2|S|S) = \mathcal{F}_{v,2}. \tag{2.5.6}$$

Hence, by the criteria laid out in section 2.4 the DSM is scale invariant. Therefore, the DSM correctly provides scaling properties to allow for a turbulent energy cascade from the resolved to unresolved scales, as required by LES.

All that remains is to solve equation 2.5.4 for the dynamic Smagorinsky parameter c_s^2. First there are two things to made note of. As discussed in Schaefer-Rolffs and Becker (2013) [40], the shear production must be positive definite for any diagnostic SGS model. This is due to the second law of thermodynamics and the fact that entropy always increases. For the DSM the shear production is given by

$$\epsilon = -\tau_{ij}\overline{S_{ij}} = \nu_t|\bar{S}|^2 = c_s^2\bar{\Delta}^2|\bar{S}|^3, \tag{2.5.7}$$

so it is equivalent to say the second law is satisfied is $\nu_t \geq 0$. Equivalently, we require the dynamically computed Smagorinsky parameter c_s^2 to be positive definite. It is sufficient to

relax this condition by taking a spatial or temporal average (denoted $\langle\cdot\rangle$) over the resolved scale, instead requiring $\langle c_s^2\rangle \geq 0$. If it's the case that $\langle c_s^2\rangle < 0$, then some numerical instabilities will likely arise (Zang et al. 1993) [48]. Secondly we note that both L_{ij} and M_{ij} are symmetric and have zero trace, so each tensor has 2 independent elements. This means equation 2.5.4 is a system of two independent equations with only one unknown c_s^2, which is generally not mathematically exact. As this is essentially an approximation similar to the classical Smagorinsky ansatz to describe the subgrid processes, we shouldn't expect a perfect correlation between the stress and strain tensor elements. Hence, equation 2.5.4 is at most valid in a statistical sense and therefore any useful solution will require further approximation.

There are several proposed methods to solve equation 2.5.4 such as Moin's (1991) [26] suggestion to relax the assumption $c_T^2 = c_s^2$ and defining several new tensors, or the "solution applying the tensor norm" as discussed in Schaefer-Rolffs and Becker (2013) [40]. The only method we will be discussing is the least squared approach by Lilly (1992) [25] as it is the method we used in our implementation of the DSM. For this approach [25] solves $\partial(L_{ij} - c_s^2 M_{ij})^2/\partial c_s^2 = 0$ yielding

$$c_s^2 = \frac{L_{ij}M_{ij}}{M_{kl}M_{kl}}, \tag{2.5.8}$$

where we are using the Einstein summation convention. While this a very simple solution, an unfortunate side effect is the product $L_{ij}M_{ij}$ can become negative leading to a negative c_s^2. This may lead to numerical instabilities if not handled. One must either apply some spatial or temporal averaging, truncation to a value, or some combination of both.

We make note of a few circumstances in which the DSM has shown to give improved results. Schaefer-Rolffs and Becker (2013) [40] made use of the DSM within the Kühlunsborn mechanistic general circulation model. In Figure 3 of their work they present two kinetic energy spectra comparing the classical Smagorinsky model and the DSM. They found the DSM gave a more realistic spectrum for the higher wave numbers including a transition to the -5/3 inertial range. Khani and Waite (2015) [20] implemented the DSM for stratified turbulence and found the DSM to be less dissipative at high wavenumbers (Figure 4).

Specifically, they found that the DSM produced results similar to the Smagorinsky model using a grid spacing twice as large.

Chapter 3

Weather Research and Forecasting Model

The chapter will provide a general overview of what WRF is and what it is used for. Following that, a description of how our simulations were set up, and finally we discuss how the DSM was implemented for our tests.

3.1 Overview

We use [2] as a reference for the overview. The Weather Research and Forecasting model is a next-generation mesoscale numerical weather prediction system designed to serve both atmospheric research and operational forecasting needs. While having "weather" in its name may lead one to believe WRF can only be used to model the largest of scales, it actually serves meteorological scale from tens of meters to thousands of kilometres. WRF is the culmination of work by many organizations. Development began in the latter part of the 1990 and while the majority of work and support has been from the National Center for Atmospheric Research (NCAR), the National Oceanic and Atmospheric Administration (represented by the National Centers for Environmental Prediction (NCEP)) and the (then) Forecast Systems Laboratory, and the Air Force Weather Agency, there have been

contributions from the Naval Research Laboratory, the University of Oklahoma, and the Federal Aviation Administration as well. WRF offers two dynamical solvers: the WRF-ARW (Advanced Research WRF) core and the WRF-NMM (Non-hydrostatic Mesoscale Model) core. NCAR is the prime developer of the ARW model and is maintained by their Mesoscale and Microscale Meteorology Laboratory. On the other hand, the NMM core was developed by NCEP and is currently at use in their hurricane WRF system. For our simulations we use WRF-ARW.

WRF has the functionality to produce simulations based on real atmospheric conditions (i.e. data from observations or analysis) or true idealized conditions. At the beginning of this research project we knew we wanted to test the DSM versus classical Smagorinsky using WRF; however, we had not decided whether to use an idealized case or not. Ultimately the decision came down to the fact that using real data is more interesting (i.e. cooler). The decision was made to use Hurricane Igor as the real weather event of choice as discussed in section 1.2.

3.2 Simulation Setup

To perform a real data simulation you need not only atmospheric weather data (for initial and boundary conditions), but data from the static geography as well. WRF uses data given by [4]. There is an extensive list of data - some which one may or may not need for individual simulations. To obtain the actual weather data we employ the services of North American Regional Reanalysis (NARR). NARR is a regional reanalysis of North America containing data for dozens of variables, and is produced by NCEP. The NARR model takes in, or assimilates, a large amount of observational data to produce a long-term picture of weather over all of North America. The data is produced in General Regularly-distributed Information in Binary form (GRIB) format. To obtain the data required to simulate Hurricane Igor we simply put in a request for data over the time frame of the event. For our simulations we requested data from September 20th to September 22nd in the year 2019. This time frame leaves Hurricane Igor striking Newfoundland roughly halfway through the

simulation. Note NARR distributes data in 3 hour increments.

As NARR supplies data for all of North America, we need to specify the region we are interested in. To do so we make use of WRF Preprocessing System (WPS). The supported functions we use in WPS to be specify our domain are: *geogrid.exe*, *ungrid.exe*, and *metgrid.exe*. Geogrid takes static geographical data and fits it to the specified grid. Ungrid takes the static GRIB data and turns it into an intermediate file format. These can be run independently of each other, but metgrid, which takes output from geogrid and ungrid and interpolates the data to the domain for the specified times, must be run last. The NARR data is used as the boundary conditions in our domain. It's important that the domain is large enough that the boundaries do not interfere with the body of the hurricane as it moves along its path. The only limitations on domain size are run time and required memory, so it is encouraged to go large; however, with Newfoundland containing the most eastern point in *North America* and with our data being sourced from *North America*, we actually end up outside the domain for data NARR provides if we stretch our domain too far south east. We feature two domains of varying sizes and grid spacing for comparison.

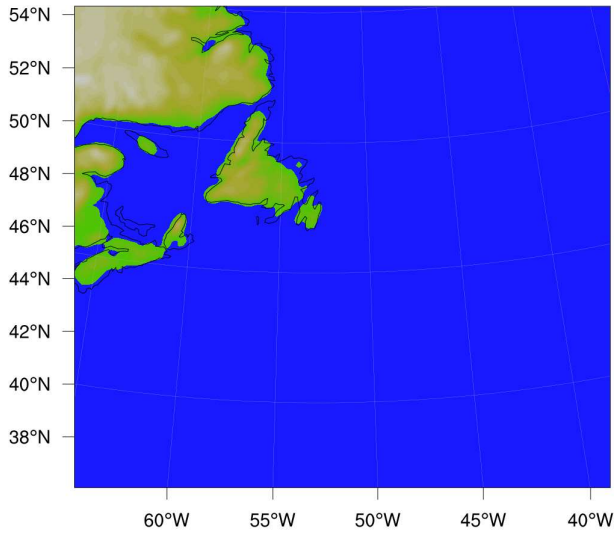

Figure 3.1: Simulation domain contoured by terrain height.

Figure 3.1 shows the size of our larger, finer domain with terrain height contours. You

can see there is still plenty of space between the boundary in the south and east. For the larger domain the reference latitude and longitude, corresponding to the centre point, are 46°N and 51.5°W, respectively. For the grid we use a spacing of $dx = dy = 15,000$ m, with west-east grid points $n_{we} = 150$ and south-north grid points $n_{sn} = 135$. This gives a domain size of $(L_{we}, L_{ns}) = (2250, 2025)$ km. The smaller domain uses a reference latitude and longitude of 44.5°N and 54.5°W, respectively. The grid spacing is $dx = dy = 20,000$ m with grid points $n_{we} = 105$ and $n_{ns} = 90$. Thus the domain size is $(L_{we}, L_{ns}) = (2100, 1800)$ km. There are 30 vertical levels ranging from a minimum height of approximately 40 m to a maximum of approximately $16,500$ m. The minimum layer separation is about 45 m at the lower boundary, while the maximum is about 800 m near the top. With this separation the troposphere lies below the 22nd layer. We use a timestep of $\Delta t = 90$ s. WRF uses a staggered grid such that a majority of scalar variables are taken at the centre of the cell; however, the velocities are taken at the midpoint on the corresponding boundary. Because of this we ensure to interpolate the velocities to the centre point in our implementation of the DSM.

Once we've successfully run WPS to build our domain we can finally run our simulation. The first step is to run *real.exe*. This will vertically interpolate data created during WPS and create the boundary and initial conditions. The final step is to run *wrf.exe* which generates the simulation. There is a *nameslist.input* text file containing a multitude of parameters we can use to alter the simulation. In general we follow the best practices as desribed in [3]. Our test case is relatively quick to run. Using 16 processors we are able to complete a simulation in approximately 15 minutes. In the greater world of CFD this is quite fast, so one might think we should decrease the resolution. However, the data we obtain from NARR has fairly coarse lower bound of 32 km and we cannot bring our grid size to be much smaller than this without receiving errors. We could address this by using nested domains, but we decided against this for simplicity. Using this procedure we can easily run a real data simulation using the 2D horizontal Smagorinsky model given by equations 2.3.7 and 2.3.8; however, WRF does not include an implementation of the DSM for real data simulations, so we must write it ourselves.

3.3 Implementing Dynamic Smagorinsky

There are essentially two steps required to implementing the DSM. The first is filtering all the required variables at the test filter width, and the second is solving for the Smagorinsky constant c_s. The second step is much simpler than the first. We just have to apply equation 2.5.8 after computing L_{ij} and M_{ij}. Referencing equations 2.5.2 and 2.5.4, we'll need to filter the velocity field (and its product) to compute L_{ij} and filter the strain rate tensor, its magnitude, and their product to compute M_{ij}. Once we successfully filter one field, the others are the same. We can use the subroutine already present in the WRF code for horizontal Smagorinsky. This way all the proper framework is in place and we just need to add the extra steps to compute c_s. We apply a box filter

$$\overline{u(a,b)} = \frac{1}{\Delta^2}\int_{a-\frac{\Delta}{2}}^{a+\frac{\Delta}{2}}\int_{b-\frac{\Delta}{2}}^{b+\frac{\Delta}{2}} u(x,y)\,dxdy, \tag{3.3.1}$$

and use the 2D trapezoidal rule given by [1]. For a function $f(x,y)$ over a rectangle $\mathbf{R} = \{\{x,y\} : a \le x \le b, c \le y \le d\}$ subdivided into grid points ranging $x_0, ..., x_m$ and $y_0, ..., y_n$ with h and k the grid spacing we have

$$\iint_{\mathbf{R}} f(x,y)\,dA = \int_a^b\int_c^d f(x,y)\,dxdy \approx \mathrm{T2D}(f,h,k), \tag{3.3.2}$$

where

$$\begin{aligned}\mathrm{T2D}(f,h,k) = \frac{hk}{4}&(f(a,c)+f(b,c)+f(a,d)+f(b,d)\\ &+2\sum_{i=1}^{m-1} f(x_i,c)+2\sum_{i=1}^{m-1} f(x_i,d)+2\sum_{j=1}^{n-1} f(a,y_i)+2\sum_{j=1}^{n-1} f(b,y_i)\\ &+4\sum_{j=1}^{n-1}\sum_{i=1}^{m-1} f(x_i,y_i)).\end{aligned}$$

This equation for T2D gives us the formula we use to filter all of our fields. Initially this looks much more complex than simply applying a 1D trapezoid rule twice (once for x and y); however, doing so is less efficient. This is because to implement 1D trapezoid rule we have to embed a loop within the existing loops over our grid points as the filter has to be

applied at every node in the grid. This adds an additional order of cost to the complexity. It is possible to avoid this using a 2D method as we can write T2D as a matrix:

$$\text{T2D} = \frac{hk}{4}\begin{bmatrix} 1 & 2 & 2 & \dots & 2 & 2 & 1 \\ 2 & 4 & 4 & \dots & 4 & 4 & 2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 2 & 4 & 4 & \dots & 4 & 4 & 2 \\ 1 & 2 & 2 & \dots & 2 & 2 & 1 \end{bmatrix}$$

which is very easy to construct using Fortran. Using this method we can keep all our loops the same order. We just have to manually construct our trapezoid matrix and perform "matrix" multiplication with the quantity we want to filter. This works because naively multiplying matrices (which are truthfully 3D arrays) is done component-wise, so we get the desired expression. We then multiply by the appropriate constant.

To determine the size of the trapezoid matrix we must consider the size of our test filter. It is recommended to use a test filter twice the size of the default filter. This corresponds to a 3 grid point filter and thus our trapezoid matrix will be size 3×3. The constant is found via $hk/(4\tilde{\Delta}^2)$. As we have the same grid spacing in both directions this reduces completely. Using $h = k = dx$ and $\tilde{\Delta} = 2dx$ gives us a constant $1/16$. In pseudo-code this will look something like

```
do,do,do (j,k,i)
U(i,k,j) = sum(0.5*(u(i-1:i+1,k,j-1:j+1) + u(i:i+2, k, j-1:j+1))*T2D)
end,end,end
U = U/16
```

There is no concern for points near the boundary as WRF calculates up to 10 ghost points for each of its fields. This means we can freely start and end the loop indexes without fear of going out of range via the $i \pm 1$. Tests were done on constant and noisy vectors to ensure the correctness of the filtering. Once we have a properly working filter code we can calculate the tensors L_{ij} and M_{ij}. From there the only concern is negative values of c_s.

We found approximately half the values of c_s to be negative as found in other studies; e.g. Khani and Waite (2015) [20]. We attempted filtering c_s through a variety of filter sizes, but found that only a smaller percentage of the negative values were being removed each time. This meant we had to apply a truncation method, with which the remaining negative values were set to zero. Compared with a pure truncation method, we generally found little difference. We will be presenting results using a two methods: where c_s is filtered twice on a $8dx$ width filter and where there is no filtering of c_s, i.e. a pure truncation method.

Chapter 4

Results

We first present our "base case" using the standard Smagorinsky model and, using several physical quantities, show our simulation captures basic features of Hurricane Igor such as its track and regions of precipitation. We present a comparison of these quantities with the DSM to identify which features depend on the subgrid model. We then give several energy spectra of u and w to definitively deduce whether our implementation of the DSM has a decreased dissipation at the small scales.

4.1 Base Case

As stated in the objectives, we would like to "successfully simulate" Hurricane Igor; however, this statement is bit vague. More specifically we would like to see that what we are simulating does indeed reproduce some basic features of Igor, such as pressure and precipitation. Perhaps one of the easiest ways to verify we have recreated a hurricane is to follow its path as determined by point of lowest pressure. To do this we plot sea level pressure contours over the time frame of our simulation. From this we can easily identify the storm as it moves and get a good estimate on its centre i.e. the point of lowest pressure. Table 4.1 features the minimum pressure at various times along with latitude and longitude values. This data is obtained from Weather Underground and is used for comparison. Figure 4.1

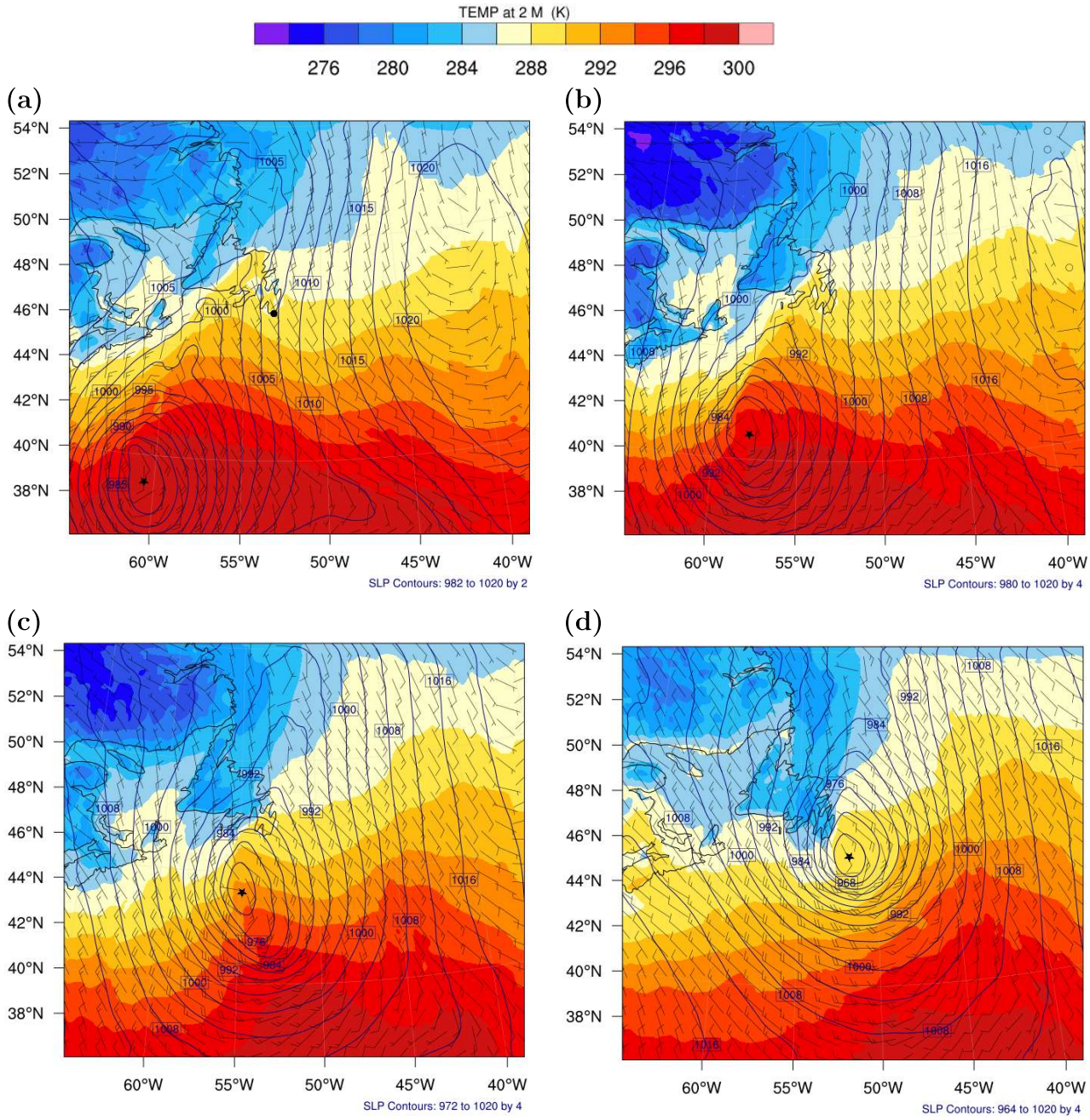

Figure 4.1: Sea Level Pressure contours at times: **(a)** 00 UTC, **(b)** 06 UTC, **(c)** 12 UTC, and **(d)** 18 UTC on September 21st. Additionally u velocity vectors at 10 m and temperature at 2 m are shown.

Date	Time (UTC)	Lat (°N)	Lon (°W)	Pressure (hPa)
09/21/2010	00 UTC	39.0	60.4	960
09/21/2010	06 UTC	41.5	57.0	960
09/21/2010	12 UTC	44.8	54.4	955
09/21/2010	18 UTC	48.5	52.1	950

Table 4.1: Minimum pressure of Hurricane Igor for various times and locations. Data provided by Weather Underground.

displays pressure contours on September 21st for the times listed in table 4.1. Additionally velocity vectors of u at 10 m and contours of temperature at 2 m are given. Results are only presented for our larger, finer, domain as the differences between the pressure, velocity, and temperature are minute; the same trends are observed. The first thing we note is the simulation is consistent with the timing given in section 1.2. Hurricane Igor landed at Cape Race (marked by the black dot in Figure 4.1 **(a)**) at approximately 15 UTC. Figure 4.1 **(c)** and **(d)** show this quite well. The centre of the storm clearly passes over the Avalon Peninsula between these times. Comparing with the given data, the earliest time we see the centre of the inner contour is in very good agreement in terms of location. Using the fine, blue tinted text beneath each figure, we see the minimum contour is 982 hPa. We can estimate the minimum value of pressure by taking P at the lowest contour value and subtracting ΔP. This gives us 980 hPa for the earliest time. This is quite a bit larger than 960 hPa; however, this improves with time. At the latest time the estimated minimum pressure is 960 hPa, which is only 10 hPa higher than expected.

Casson and Coles (1999) [9] observed the opposite trend. In Figure 8 of their paper they feature quantile-quantile plots of the minimum pressure for landfalling and non-landfalling hurricanes, where landfalling implies part of the hurricane's track intercepts land. The model they used is based off of the HURDAT database which, at the time, contained information on hurricanes and tropical storms in the North Atlantic from 1886 to 1994. Their approach involved simulating an n-year epoch of the hurricane history. The general trend was that for non-landfalling hurricanes, the observed pressure was slightly larger than the simulated pressure. The opposite was true for landfalling hurricanes. By their definition

Igor is a non-landfalling hurricane for the majority of its duration, meaning our simulated results disagree with their findings. Aside from this, the order of error in recorded minimum pressure values are similar to what we found. A more accurate depiction of the

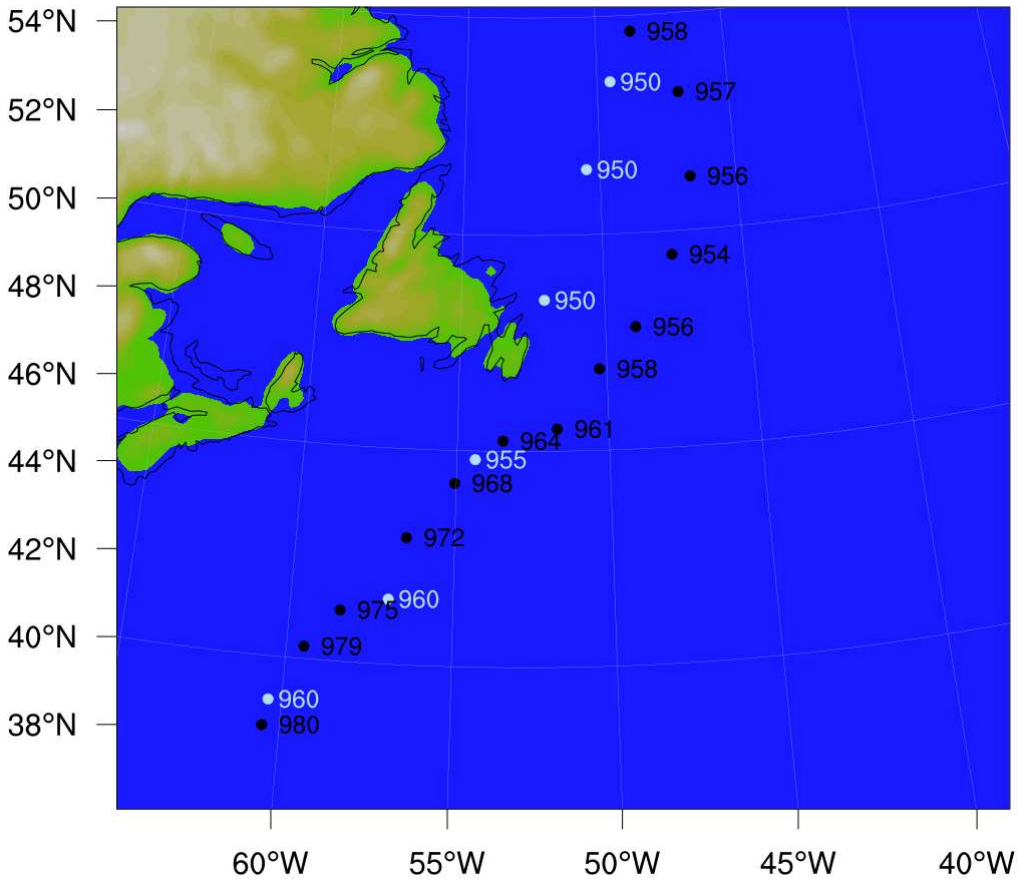

Figure 4.2: Simulated track (black) versus data obtained from Weather Underground (white) of Igor via the minimum pressure. Δt black: 3 hours; Δt white: 6 hours. Units are hPa.

hurricanes track and minimum pressure is given in Figure 4.2. The black dots indicate an exact location of the minimum pressure for different times as extracted directly from our output files. The white dots indicate the same but data was taken from Weather Underground. Note for every white dot there is two black dots. The earliest pressure corresponds to Figure 4.1 **(a).** You can see our estimation of 980 hPa is spot on. The same can be

said for the other plotted times. The locations of minimum pressure are in good agreement for all times displayed with the exception of 18 UTC and later. Here both pressure and longitude show good agreement, but the expected latitude falls short by several degrees. Notice the earliest white 950 hPa dot is directly north of the black 961 hPa dot. This is seen again with the second white 950 hPa dot and first black 956 hPa dot. We can assume the simulation moves more slowly than the actual storm. The velocity vectors are shown to see that they do follow the pressure contours as expected. From the temperature we can infer a shift from night to day. Notice the temperature of Nova Scotia increase by up to 10 Kelvin between 06 UTC and 18 UTC. Locally this is 2:00 a.m. to 2:00 p.m. In any case, this shows our simulation of Igor follows an appropriate path with reasonable pressure.

Another quantity we can plot to observe the validity is the rain rate. In section 1.2 we mentioned the highest confirmed rain total was 238 mm $\approx$ 9.37 inches in St. Lawrence. Figure 4.3 illustrates exactly thus by giving the total rainfall over Newfoundland from

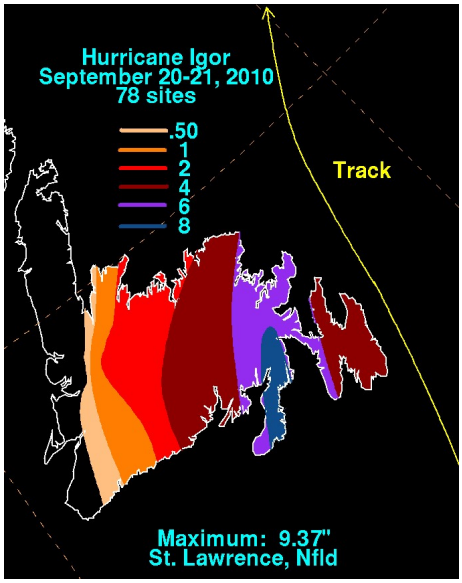

Figure 4.3: Total rainfall over Newfoundland. Obtained from [38].

the 20th to the 21st. Note that St. Lawrence is located in the southern most section of the dark blue colour. We could not find a radar image displaying rain rate of Igor over

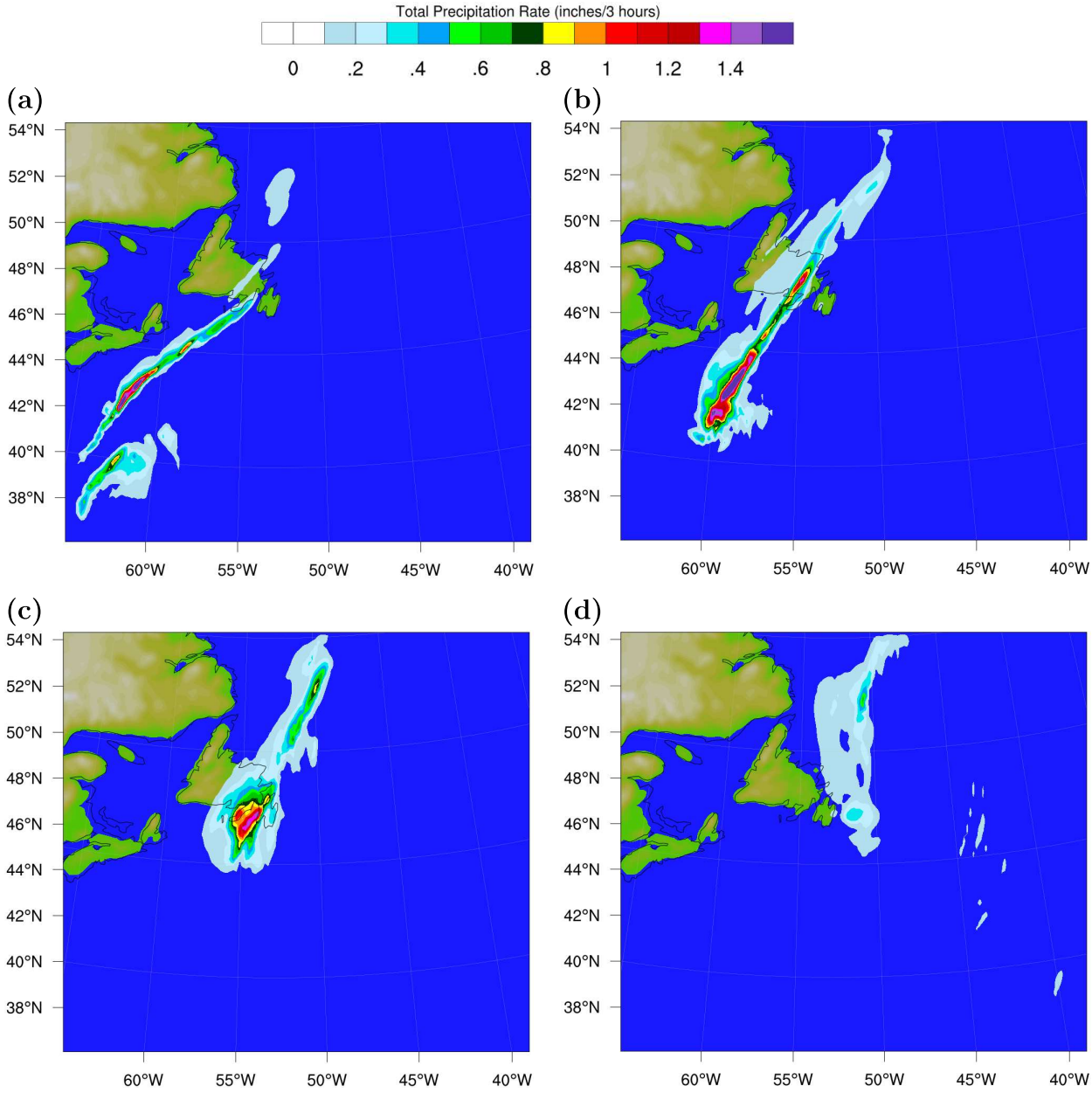

Figure 4.4: Rain rate per 3 hours for times: **(a)** 18 UTC on September 20th and **(b)** 03 UTC, **(c)** 12 UTC, and **(d)** 21 UTC on September 21st. Coloured by terrain.

Newfoundland, but this will still be a useful comparison. Rain rate in inches per 3 hours is given by Figure 4.4. We use 3 hour increments as our data from NARR is given as such. Only the larger domain is displayed as there is nothing new to gain from the smaller. The first thing to notice is the band-like structure that is present. This strong band of rainfall forms as early as 15 UTC on September 20th and continues as such until petering out shortly after 21 UTC on the 21st (see Figure 4.4 **(d)**). This banding can actually be seen in the satellite image given in Figure 1.1, which is timed shortly before Igor arrives at Newfoundland. Furthermore, Igor quickly devolved into a extratropical cyclone as it pasted over Newfoundland, where these large band structures are commonly found [18]. Geographically the bands of heavy precipitation are located over the St. Lawrence area from 18 UTC on the 20th until 12 UTC on the 21st. In the next immediate time frame

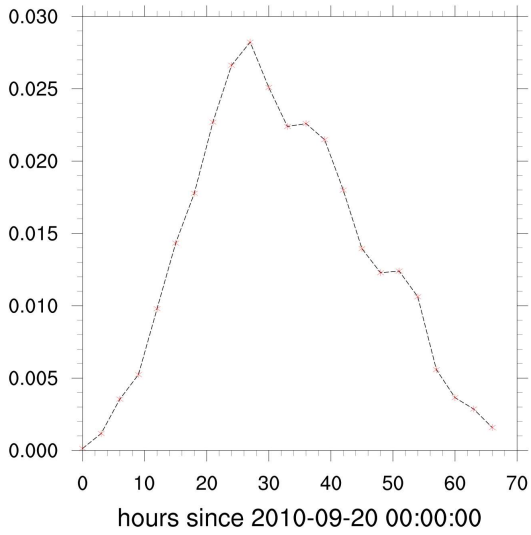

Figure 4.5: Spatial averaged rain rate per 3 hours over the entire domain.

rainfall over St. Lawrence has reduced to approximately 0.2 inches per 3 hours and has completed fallen off by 21 UTC on September 21st. Thus, it is clear that our simulation places the largest amount of rain in the expected location. Using the full set of our rain rate figures for all times we estimate total rainfall to be approximately the same as expected in the high intensity areas. In addition to plotting contours at various times, we can take a spatial average of rain rate over the entire domain and plot that over the simulation

time. Figure 4.5 features exactly this. Note that the values on the y-axis are essentially irrelevant as being averaged over the entire domain significantly reduces the values. What this does tell us is during which times was the rainfall the heaviest. We see this to be around the end of September 20th and the beginning of September 21st. This time frame was the heaviest rainfall displayed in Figure 4.4, and indeed was one of the heaviest rain rate contours we obtained.

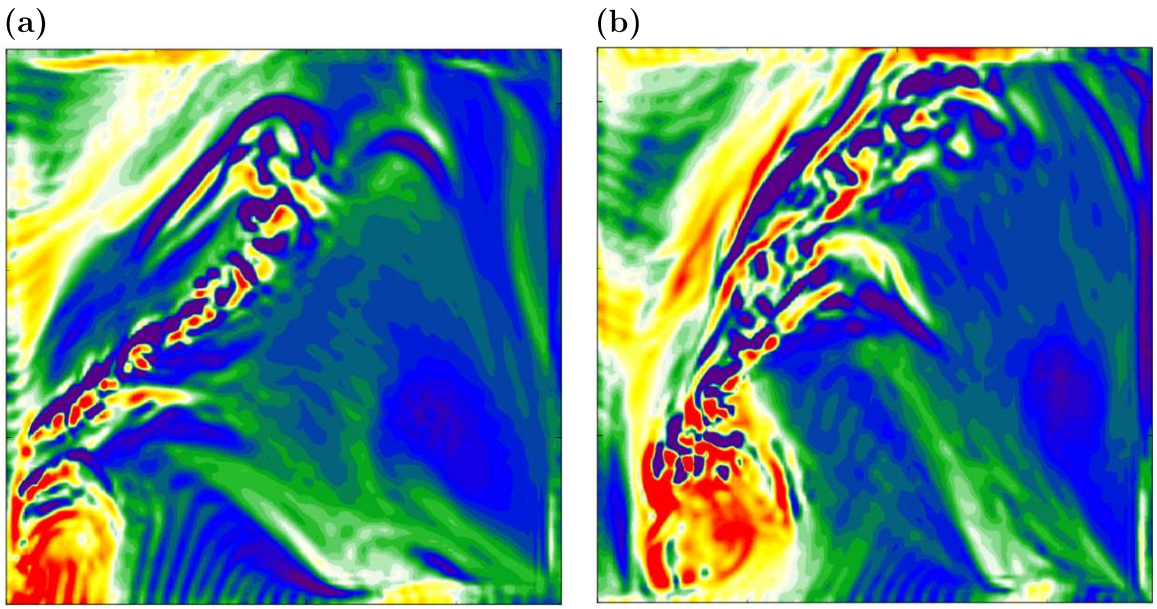

Figure 4.6: Vorticity at times: **(a)** 18 UTC on September 20th and **(b)** 03 UTC on September 21st.

Finally, we display the vorticity in Figure 4.6 to greatly emphasize the formation of the eye combined with the large band observed in both the rain rate and satellite imagery. It's clear that the centre of the storm shows positive (cyclonic) vorticity. At the small scale we see both positive and negative values in regions of strong precipitation (note these are the first two times pictured in Figure 4.4). The "slope" of the vorticity bands align very will with the bands of precipitation. Also the effects of the boundary can be observed; for instance, the ribbed structure we see around the bottom left corner.

4.2 Comparison with Dynamic Smagorinsky

In this section we provide a comparison for some of the quantities displayed in section 4.1 as well one other. We start by comparing the sea level pressure. Figure 4.7 features

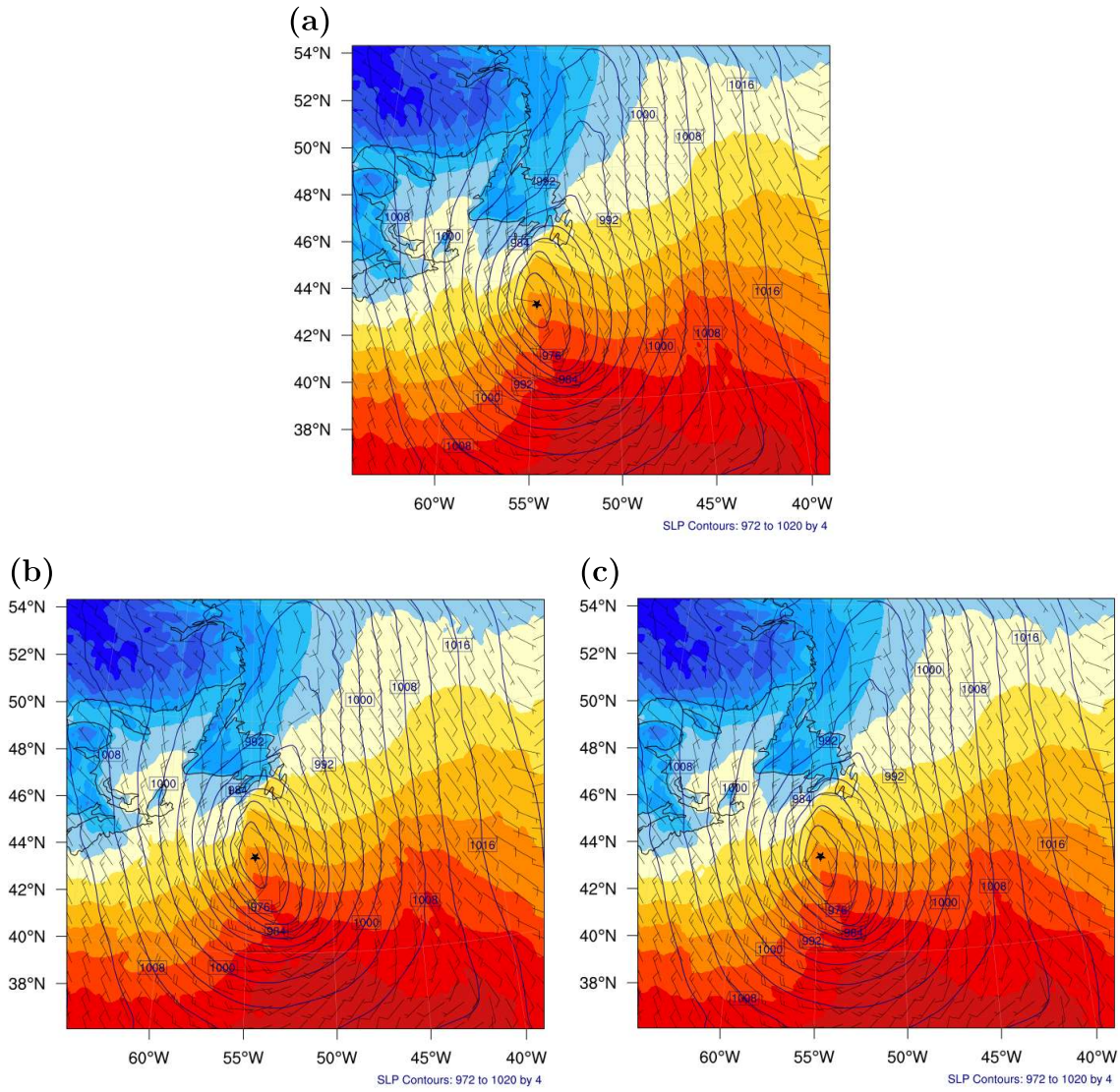

Figure 4.7: Sea level pressure contours at time 12 UTC for: **(a)** Smagorinsky model, **(b)** DSM with filtering, and **(c)** DSM on September 21st.

exactly this. Pictured is the sea level pressure on September 21st at 12 UTC for our base case and two versions of the DSM. The first implements a twice filtering technique with an $8dx$ filter on the Smagorinsky coefficients to try and remove many of the negatives before truncation, while the second is pure truncation method. The images are highly similar. This is observed for all times tested. The level of similarity may seem surprising; however, the pressure is such a large scale quantity that the effects of dissipation would be very small. Thus, the overly dissipative nature Smagorinsky is unlikely to have a large impact. Perhaps what is more interesting is that the difference between both versions of the DSM being so small. One would think the difference between simply chopping a negative to zero and averaging it to get something positive would be significant. As mentioned in Chapter 3, we found nearly half of the c_s values to be negative. Therefore, an averaging technique essentially just brings all values closer to the zero with a slight positive favouring. In fact, we chose to average twice on such a large filter because anything lesser did not remove a large enough portion of the negatives. Despite this, the pressure being so large scale trumps the minute differences. What this does tell us is that implementing the DSM did not break our code. There are far too many pressure contours to show in this paper, but we found a level of agreement similar to the Smagorinsky model with regards to Table 4.1.

Figure 4.8 gives the rain rate contours using the DSM for both methods to compute c_s. Notice these are quite a bit more different from one another than the pressure contours. We see what may be a consequence of over filtering c_s when comparing both methods in Figure 4.8 **(c)** and **(d)**. In particular, there is significantly less rain using the $8dx$ filter method, even when compared to the same time in Figure 4.4. This is most noticeable around 44°N and 55°W. As discussed, having so many c_s values less than zero means filtering brings the overall average of c_s a lot closer to zero. This is what we mean by over filtering - we would expect to observe more rain using a dynamic method. This same phenomena was extremely prominent in the immediate proceeding time frame as well. However, it seems the contrary is true when considering Figure 4.8 **(a)** and **(b)**. Though to a less extreme effect, we clearly see large values for the rain rate with the $8dx$ filter method, so this is rather inconclusive. Regardless, the aim of the DSM is to be less dissipative. Thus, for a small scale quantity like the rain rate, we expect to see larger quantities when compared

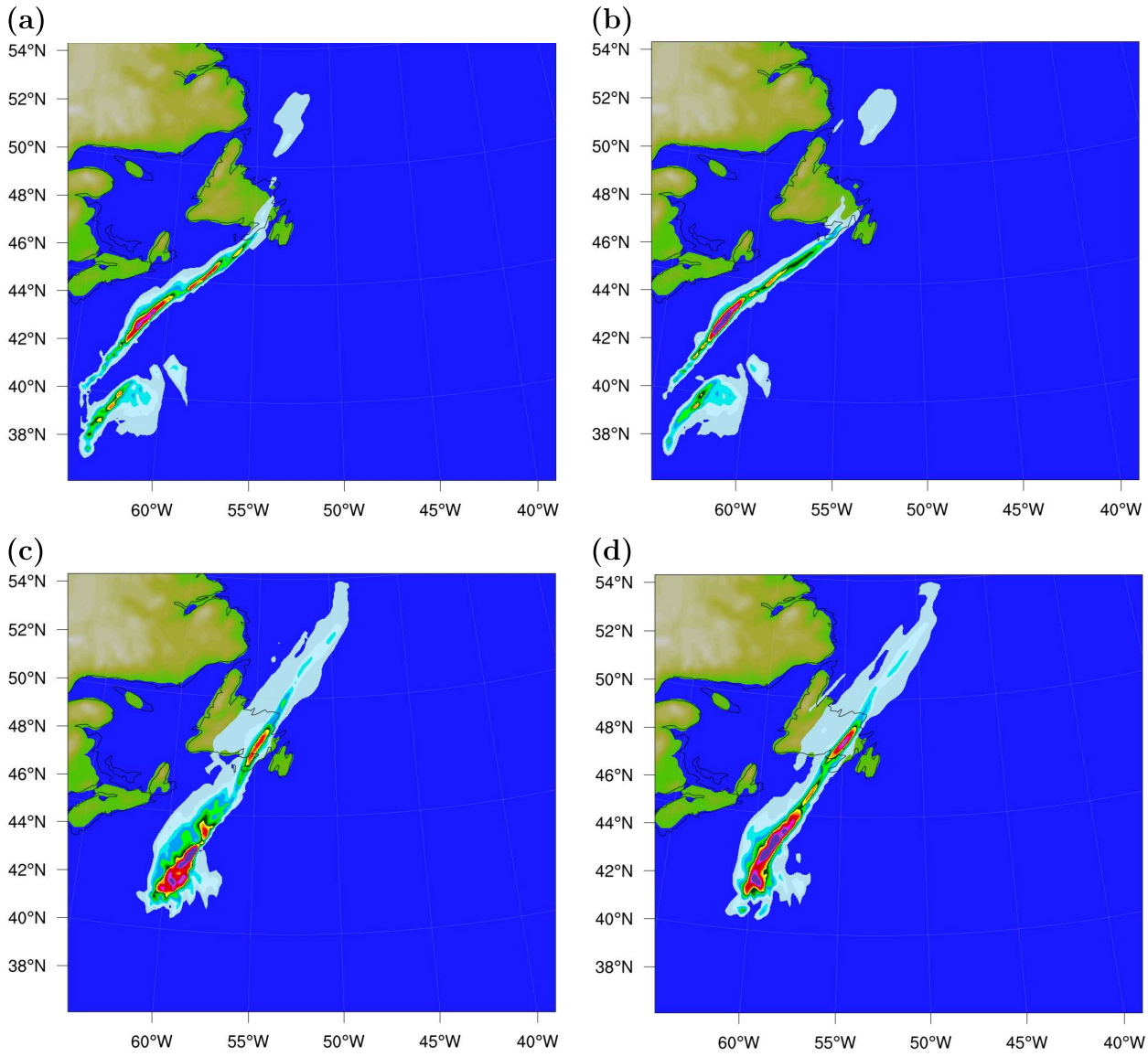

Figure 4.8: Rain rate per 3 hours. **Top:** September 20th, 18 UTC, **Bottom:** September 21st, 03 UTC. **Left:** DSM with filtering, **Right:** DSM. Coloured by terrain.

to the original Smagorinsky model. There is some truth to this if we take least dissipative figure from each method and compare with Figure 4.4; however, it's far more similar than expected. This is quite surprising, so we analyse another small scale quantity, the vertical

velocity w, to attempt to better understand the behaviour.

The effects of grid resolution had a more significant impact on the vertical velocity, so we will feature contours from both resolutions. We took the velocity at the 15th vertical level which corresponds to about 5550 m. This puts us in the mid-troposphere. Figures 4.9 and 4.10 picture the vertical velocity for the large (with $dx = 15$ km) and small (with $dx = 20$ km) domains, respectively. We see using a finer resolution allows us to see much more of the velocity. This is especially evident in the light blue banding seen between 44°N and 48°N. The comparison between the Smagorinsky model and the DSM with no filtering is minimal. For both resolutions we see very little difference. It seems the model is doing little to improve the dissipation. On the other hand, the $8dx$ filter method shows some promise. Even at the low resolution we see light blue bands forming immediately north-west of the largest velocities, resembling the Smagorinsky model at a higher resolution. At the higher resolution the banding is strong not only above the eye of the storm (where the highest velocities are recorded), but well beneath it as well (say between 38°N and 42°N). These bands are actually prominent gravity waves that are not visible in any of the other cases. This result consistent with weather dissipation. This trend was observed for many of the other time frames, but it was of largest significance for the frame presented. Nevertheless, this is at least a result somewhat in favour of the DSM.

4.3 Energy Spectra

Overall, the results presented in the previous section seem to imply implementing the DSM does not seem to improve upon the over dissipation of the Smagorinsky model for large scale atmospheric simulations. We can further explain whether or not this is true by examining the energy spectrum of the velocities. To compute the the energy spectra of the velocity, say u, we calculate the Fourier transform in 1D of the velocity over the entire domain at a specific height. Just as with the vertical velocity contours, we use the 15th level. Then the energy in a particular wave number can be found via

$$E_u = \frac{1}{2}|\hat{u}|^2, \tag{4.3.1}$$

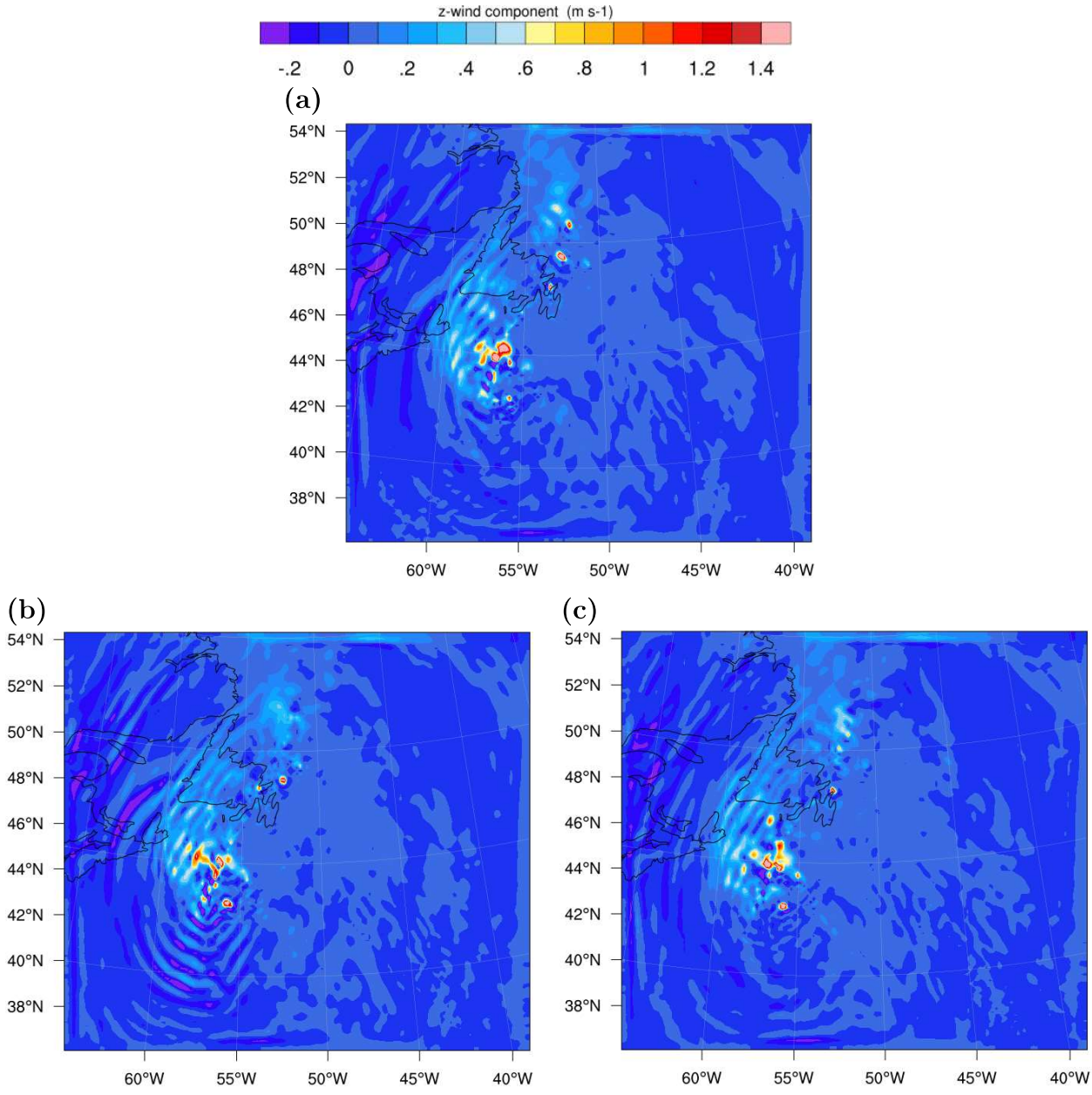

Figure 4.9: Vertical velocity contours at time 09 UTC for: **(a)** Smagorinsky model, **(b)** DSM with filtering, and **(c)** DSM on September 21st.

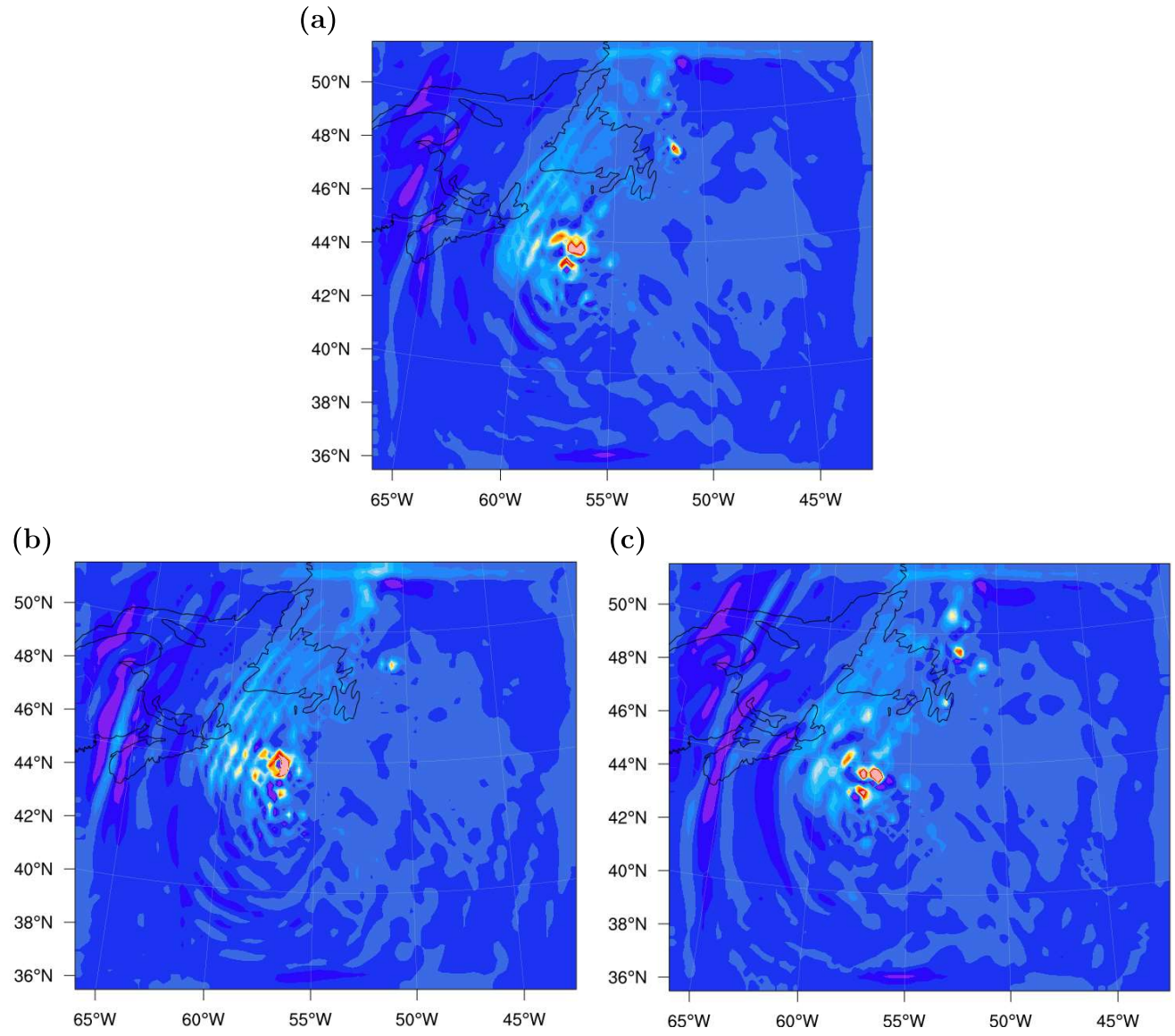

Figure 4.10: Vertical velocity contours at time 09 UTC for: **(a)** Smagorinsky model, **(b)** DSM with filtering, and **(c)** DSM on September 21st. Small, coarse domain.

where the carrot symbol indicated a transformed variable to the wavenumber space. To get the energy is just the x (or y) direction, we simply average over the number of rows (or columns). We then take a log-log plot to visualize the spectra. The only challenge is that

our domain is not periodic, which is required to take the Fourier transform. To solve this we can use a strategy discussed in [14], in which we construct a line using the first and last point in the velocity vector and then subtracting this linear trend from said vector. Thus, we have the spectra. There is a lot of theory about how the energy spectra should look like in a perfect LES and when using the Smagorinsky model. This was briefly discussed in the section 2.3 on scale invariance. So we can infer a lot based on how our spectra look.

Figure 4.11 gives the energy spectra of u for September 22nd, 00 UTC. A line with slope $-5/3$ is plotted for reference to the power law. The results are pretty definitive. In a perfect LES, the energy spectra would follow more or less what is pictured in Figure 4.11 **(c)** and **(d)** up to the medium wavenumbers (say $k \approx 50$ in the large domain case). For wavenumbers larger than this we see a steep drop in the energy. This is a known deficit of the Smagorinsky model - that it dissipates too much energy in the small resolved scales. One aim of the DSM was to try and correct this. In other words, we expect to see the Smagorinsky model and the DSM in high agreement up until the large wavenumbers. After this point the DSM should maintain roughly the same slope (barring some exceptions - say if the small scales have different physics), while the Smagorinsky model starts to drop. This would definitively show that the DSM has remedied the issue of over dissipation. This is decidedly *not* what we observe for domains of both sizes. There is slightly more promising result given in Figure 4.12 which depicts the energy spectrum of w for September 21st, 15 UTC. We again expect to see larger values of energy for the larger wavenumbers, and this is certainty not the case for the larger domain. There is a small sliver of hope for the DSM when considering a courser domain of $dx = 20$ km. For wavenumbers approximately $k \geq 20$ we do see a significant increase in the energy. This was observed for several of the time frames while the storm was active, but not all. On the other hand, the Figure 4.11 solidly represents what was observed for all relevant time frames. Overall, with a coarse resolution we observe more small-scale energy with DSM (albeit subtle); this is consistent with DSM being less dissipative.

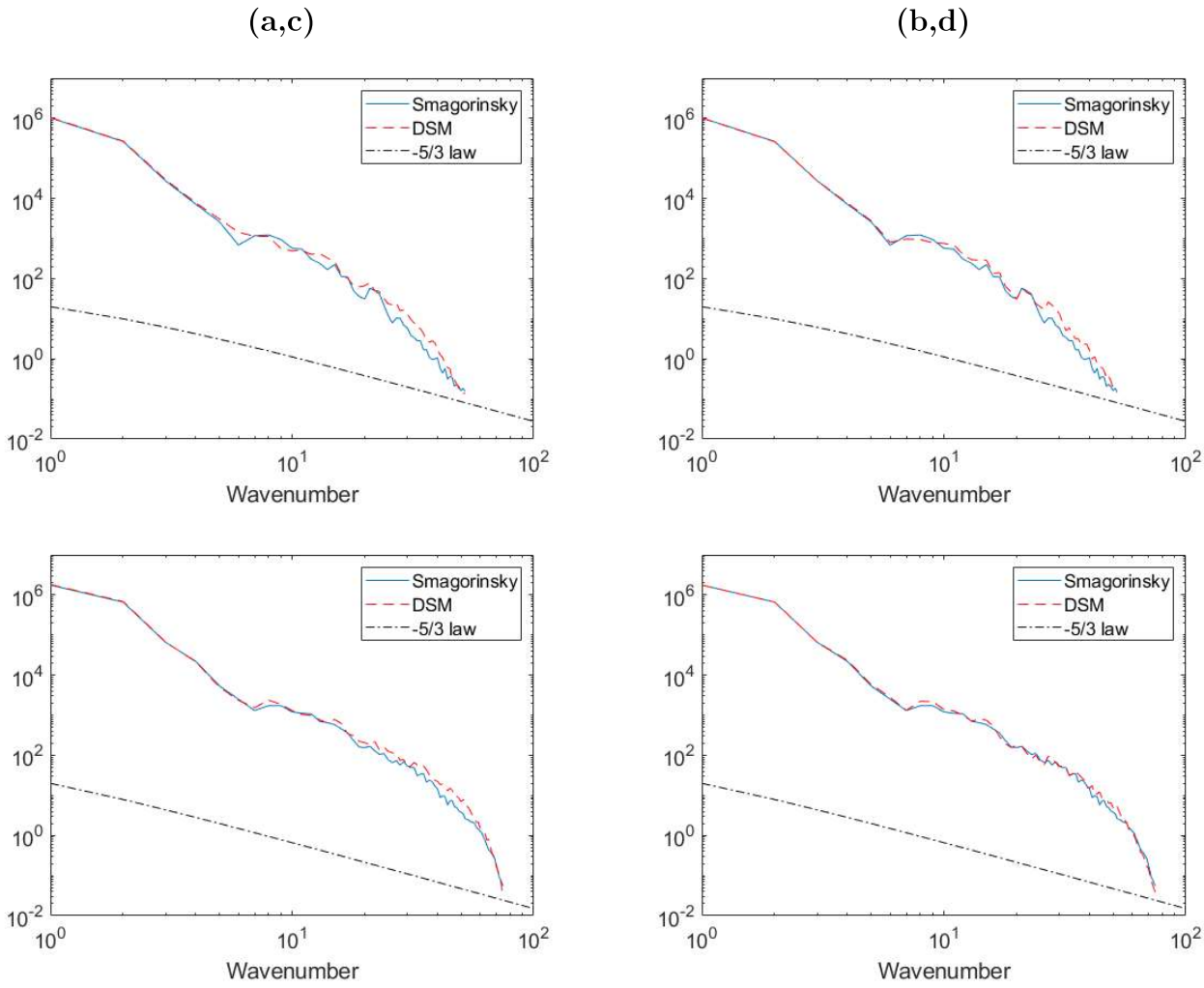

Figure 4.11: Energy spectrum of horizontal velocity u on a loglog scale. **Top:** small, coarse domain, **Bottom:** large, fine domain. **Left:** DSM with filtering, **Right:** DSM.

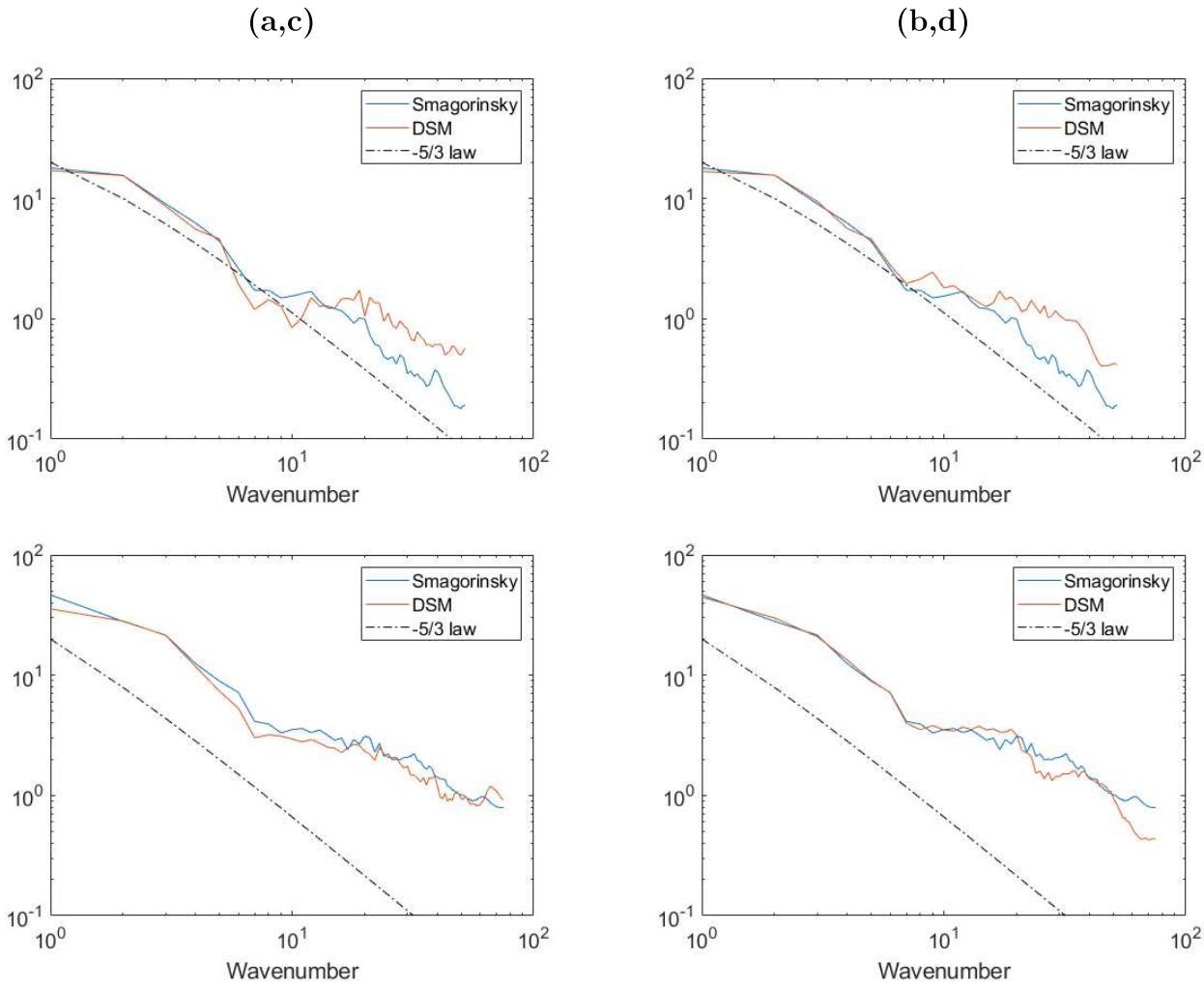

Figure 4.12: Energy spectrum of vertical velocity w on a loglog scale. **Top:** small, coarse domain, **Bottom:** large, fine domain. **Left:** DSM with filtering, **Right:** DSM.

Chapter 5

Conclusion

In this report we motivated the study of turbulence and turbulence modelling. We presented a simulation of Hurricane Igor using WRF's real data functionality to study two turbulence models - specifically the effects of using the DSM over the Smagorinsky model. We provided necessary background information to motivate the use of the DSM and argue why it should prove to be a useful improvement over the classic Smagorinsky model. A base case using the classical model is provided where we qualitatively show that our simulations are sufficiently correct in simulating Hurricane Igor. An implementation for the DSM using WRF is given. Using this we compare and contrast results with the Smagorinsky model. The findings are such that, despite the theory and previous evidence of the DSM success for other types of fluid dynamics simulations, we find it is likely not worth the additional cost to implement when using WRF for large scale atmospheric simulations.

In order to more definitively say whether or not the DSM is worth the additional cost, we ran our simulation on a single processor for both models and recorded the time. Implementing the DSM in WRF requires an additional 2 loops using a truncation method and 4 loops using a double filter method. The loops are over i, k, j, thus are of order $\mathcal{O}(N^3)$ if N is the total number of grid points. This is the same order as loops which are already present within the subroutine that we have implemented the DSM in. Hence, the total cost is only increasing linearly in its order. For a much simpler piece of software

or code, this could be very significant - easily doubling the cost [20]. However, WRF is a very complex piece of software that has to handle a lot of other processes during each step of the simulation. Because of this we wouldn't except anything close to a doubling of time, but there should still be a noticeable increase. After running on a single processor we found that the Smagorinsky model finished in 02:32:06, while the DSM finished in 02:15:42. This is clearly a due to a difference in processors, so we reran the simulations to account for this. A second run found the Smagorinsky model finishing in 02:32:33, and the DSM finishing in 02:25:29. From this we conclude it may or may not be worth the time to implement the DSM into WRF for large scale atmospheric simulations. The additional complexity is very small, but the improvements are also so small it may not be worth the effort. Truthfully the DSM *will* be slower eventually. Increasing computation time via increasing domain size or simulation length would eventually show this. With times this similar it may also be useful to either use a temporal average over several simulations. We could also ensure each simulation is run on the same processor. Regardless, this shows us the added time complexity is trivially small in comparison to everything else WRF is doing.

For future work we would look into getting our data from a different provider (for instance from the Global Forecast System provided by NCEP). One of the limitations we found was that the domain of the data was slightly too small when we're interested in regions south east of Newfoundland. Being able to increase the domain size further can have no negative effects. Furthermore, we'd like data that is taken over a larger resolution allowing us to decrease our simulations grid size further without nesting. Of course we could also implement nesting for future tests. It is possible this might let the DSM fare better. Additionally, there are several other methods used to calculate c_s. We could try using a more complicated method (for instance section 3 of [40]) that is will not produce negative results. Thus we can avoid any filtering and truncation. The DSM has proven its usefulness in other CFD simulations as mentioned earlier. Perhaps some of these changes would allow the DSM to continue to shine for large scale atmospheric simulations using WRF.

References

[1] 2D Integration Using the Trapezoidal and Simpson Rules. http://mathfaculty.fullerton.edu/mathews/n2003/SimpsonsRule2DMod.html. Accessed: 2019-10-19.

[2] The Weather Research and Forecasting Model. https://www.mmm.ucar.edu/weather-research-and-forecasting-model. Accessed: 2019-10-16.

[3] WRF nameslist.input File Description. https://esrl.noaa.gov/gsd/wrfportal/namelist_input_options.html. Accessed: 2019-10-16.

[4] WRF V3 Geographical Static Data Downloads Page. http://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog_V3.html. Accessed: 2019-10-16.

[5] QMI Agency. Igor cleanup will take months: Minister. https://www.webcitation.org/5xHPR3M3W?url=http://www.torontosun.com/news/canada/2010/09/27/15490226.html, 2010. Archived from original in 2011; Accessed 2019-10-29.

[6] A. Bakker. Lecture 10 - Turbulence Models. Applied Computational Fluid Dynamics. http://www.bakker.org/dartmouth06/engs150/10-rans.pdf, 2006. Accessed: 2019-10-12.

[7] R. Berg. Hurricane Igor Forecast Advisory Fifty-Four. https://www.nhc.noaa.gov/archive/2010/al11/al112010.discus.021.shtml?, 2011. Accessed: 2019-10-29.

[8] G. Caminha. The CFL Condition and How to Choose Your Timestep Size. https://www.simscale.com/blog/2017/08/cfl-condition/, 2019. Accessed: 2019-11-05.

[9] E. Casson and S. Coles. Simulation and Extremal Analysis of Hurricane Events. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(3):227–245, 1999.

[10] Canadian Hurricane Center. Special Weather Summary Message for Newfoundland and Labrador. https://web.archive.org/web/20120111051422/http://www.atl.ec.gc.ca/weather/bulletins/nf/20100923185938.txt.en, 2010. Archived from original in 2012; Accessed: 2019-10-29.

[11] D. Roth; Hydrometeorological Prediction Center. Hurricane Igor - September 20-21, 2010. https://www.wpc.ncep.noaa.gov/tropical/rain/igor2010.html, 2010. Accessed: 2019-10-29.

[12] R. A. Clark, J. H. Ferziger, and W. C. Reynolds. Evaluation of Subgrid-Scale Models Using an Accurately Simulated Turbulent Flow. *Journal of Fluid Mechanics*, 91(1):116, 1979.

[13] J. W. Deardorff. A Three-Dimensional Numerical Investigation of the Idealized Planetary Boundary Layer. *Geophys. Fluid Dyn.*, 1(3-4):377–410, 1970. cited By 76.

[14] R. M. Errico. Spectra Computed from a Limited Area Grid. *Monthly weather review*, 113(9):1554–1562, 1985.

[15] W. K. George. Lectures in Turbulence for the 21st Century. 2013.

[16] M. Germano. Turbulence: the Filtering Approach. *Journal of Fluid Mechanics*, 238:325–336, 1992.

[17] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A Dynamic Subgrid-Scale Eddy Viscosity Model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, 1991.

[18] R. A. Houze Jr. Clouds and Precipitation in Extratropical Cyclones. In *International Geophysics*, volume 104, pages 329–367. Elsevier, 2014.

[19] M. Kemp. https://www.nature.com/articles/d41586-019-02144-z, 2019. Accessed: 2019-11-19.

[20] S. Khani and M. L. Waite. Large Eddy Simulations of Stratified Turbulence: the Dynamic Smagorinsky Model. *Journal of Fluid Mechanics*, 773:327–344, 2015.

[21] P. K. Kundu, D. R. Dowling, G. Tryggvason, and I. M. Cohen. Fluid Mechanics. 2015.

[22] R. Laprise. The Euler Equations of Motion with Hydrostatic Pressure as an Independent Variable. *Monthly weather review*, 120(1):197–207, 1992.

[23] A. Leonard. Energy Cascade in Large-Eddy Simulations of Turbulent Fluid Flows. In F.N. Frenkiel and R.E. Munn, editors, *Turbulent Diffusion in Environmental Pollution*, volume 18 of *Advances in Geophysics*, pages 237 – 248. Elsevier, 1975.

[24] E. Lévêque, F. Toschi, L. Shao, and J-P. Bertoglio. Shear-Improved Smagorinsky Model for Large-Eddy Simulation of Wall-Bounded Turbulent Flows. *Journal of Fluid Mechanics*, 570:491–502, 2007.

[25] D. K. Lilly. A Proposed Modification of the Germano Subgrid-Scale Closure Method. *Physics of Fluids A: Fluid Dynamics*, 4(3):633–635, 1992.

[26] P. Moin. A New Approach for Large Eddy Simulation of Turbulence and Scalar Transport. In *New Approaches and Concepts in Turbulence*, pages 331–339. Springer, 1991.

[27] J. J. Monaghan and J. B. Kajtar. Leonardo da Vincis Turbulent Tank in Two Dimensions. *European Journal of Mechanics-B/Fluids*, 44:1–9, 2014.

[28] M. Oberlack. Invariant Modeling in Large-Eddy Simulation of Turbulence. *Annual Research Briefs*, pages 3–22, 1997.

[29] M. Oberlack. Symmetrie, Tnvarianz und Selbstähnlichkeit in der Turbulenz (Symmetry, Invariance, and Self-Similarity in Turbulence; habilitation). *RWTH Aachen, Aachen, 187pp*, 2000.

[30] M. Oberlack and A. Rosteck. New Statistical Symmetries of the Multi-Point Equations and its Importance for Turbulent Scaling Laws. *Discrete Contin. Dyn. Syst. Ser. S*, 3(3):451–471, 2010.

[31] Government of Canada. Canada: Reports of Hurricanes, Tropical Storms, Tropical Disturbances and Related Flooding During 2010. http://www.wmo.int/pages/prog/www/tcp/documents/Doc.4.2.2_Canada.doc, 2011. Doc file, Accessed: 2019-10-29.

[32] R. J. Pasch and T. B. Kimberlain. Tropical Cyclone Report Hurricane Igor (AL112010). *National Hurricane Center, Fla*, 2011.

[33] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

[34] Associated Press. Igor's rains flood parts of northeast Canada. https://www.webcitation.org/5xHNrkFyd?url=http://www.msnbc.msn.com/id/39286755/ns/weather/, 2010. Archived from original in 2011; Accessed: 2019-10-29.

[35] O. Reynolds. III. An Experimental Investigation of the Circumstances Which Determine Whether the Motion of Water Shall be Direct or Sinuous, and of the Law of Resistance in Parallel Channels. *Proceedings of the royal society of London*, 35(224-226):84–99, 1883.

[36] C. Ronchi, M. Ypma, and V. M. Canuto. On the Application of the Germano Identity to Subgrid-Scale Modeling. *Physics of Fluids A: Fluid Dynamics*, 4(12):2927–2929, 1992.

[37] F. Roquet, G. Madec, T. J. McDougall, and P. M. Barker. Accurate Polynomial Expressions for the Density and Specific Volume of Seawater Using the TEOS-10 Standard. *Ocean Modelling*, 90:29–43, 2015.

[38] D. Roth. WPC Tropical Cyclone Rainfall Data. https://en.wikipedia.org/wiki/Hurricane_Igor#/media/File:Igor_2010_rainfall.gif, 2011. Accessed: 2019-11-14.

[39] P. Sagaut and M. Germano. *Large Eddy Simulation for Incompressible Flows: an Introduction: Third Edition*. Shi jie tu shu chu ban gong si Beijing gong si, 3 edition, 2015.

[40] U. Schaefer-Rolffs and E. Becker. Horizontal Momentum Diffusion in GCMs Using the Dynamic Smagorinsky Model. *Monthly Weather Review*, 141(3):887–899, 2013.

[41] U. Schaefer-Rolffs, R. Knöpfel, and E. Becker. A Scale Invariance Criterion for LES Parametrizations. *Meteorol. Z*, 24(1):3–13, 2014.

[42] W. C. Skamarock, J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, W. Wang, and J. G. Powers. A Description of the Advanced Research WRF Version 3. NCAR Technical Note -475+STR, 2008.

[43] J. Smagorinsky. General Circulation Experiments with the Primitive Equations. *Monthly Weather Review*, 91(3):99–164, 1963.

[44] NASA/MODIS Rapid Response Team. Igor After it Passed Bermuda. https://www.nasa.gov/mission_pages/hurricanes/archives/2010/h2010_Igor.html, 2010. Accessed: 2019-11-14.

[45] G. K. Vallis. *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation*. Cambridge University Press, 2006.

[46] G. K. Vallis. *Atmospheric and Oceanic Fluid Dynamics*. Cambridge University Press, 2017.

[47] J. C. Wyngaard. *Turbulence in the Atmosphere*. Cambridge University Press, 2010.

[48] Y. Zang, R. L. Street, and J. R. Koseff. A Dynamic Mixed Subgrid-Scale Model and its Application to Turbulent Recirculating Flows. *Physics of Fluids A: Fluid Dynamics*, 5(12):3186–3196, 1993.