UCRL-TH-215803

A Parallel Ocean Model With Adaptive Mesh Refinement Capability For Global Ocean Prediction

Aaron R. Herrnstein

December 2005

Disclaimer

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

A Parallel Ocean Model With Adaptive Mesh Refinement Capability for Global Ocean Prediction

by

AARON ROBERT HERRNSTEIN

B.A. (Wittenberg University) 1998
B.S. (Virginia Polytechnic Institute and State University) 1999
M.S. (University of California at Davis) 2001

DISSERTATION
Submitted in partial satisfaction of the requirements for the degree of
DOCTOR OF PHILOSOPHY

in

Engineering − Applied Science

in the

OFFICE OF GRADUATE STUDIES
of the
UNIVERSITY OF CALIFORNIA
DAVIS

Approved:

Professor Garry H. Rodrigue, Chair
Professor Thomas A. Cahill
Doctor Michael E. Wickett

Committee in Charge

2005

A Parallel Ocean Model With Adaptive Mesh Refinement Capability for Global Ocean Prediction

by

Aaron Robert Herrnstein

Aaron Robert Herrnstein

December 2005

Engineering − Applied Science

A Parallel Ocean Model With Adaptive Mesh Refinement Capability for Global Ocean Prediction

Abstract

An ocean model with adaptive mesh refinement (AMR) capability is presented for simulating ocean circulation on decade time scales. The model closely resembles the LLNL ocean general circulation model with some components incorporated from other well known ocean models when appropriate. Spatial components are discretized using finite differences on a staggered grid where tracer and pressure variables are defined at cell centers and velocities at cell vertices (B-grid). Horizontal motion is modeled explicitly with leapfrog and Euler forward-backward time integration, and vertical motion is modeled semi-implicitly. New AMR strategies are presented for horizontal refinement on a B-grid, leapfrog time integration, and time integration of coupled systems with unequal time steps. These AMR capabilities are added to the LLNL software package SAMRAI (Structured Adaptive Mesh Refinement Application Infrastructure) and validated with standard benchmark tests. The ocean model is built on top of the the amended SAMRAI library. The resulting model has the capability to dynamically increase resolution in localized areas of the domain. Limited basin tests are conducted using various refinement criteria and produce convergence trends in the model solution as refinement is increased. Carbon sequestration simulations are

performed on decade time scales in domains the size of the North Atlantic and the global ocean. A suggestion is given for refinement criteria in such simulations. AMR predicts maximum pH changes and increases in CO_2 concentration near the injection sites that are virtually unattainable with a uniform high resolution due to extremely long run times. Fine scale details near the injection sites are achieved by AMR with shorter run times than the finest uniform resolution tested despite the need for enhanced parallel performance. The North Atlantic simulations show a reduction in passive tracer errors when AMR is applied instead of a uniform coarse resolution. No dramatic or persistent signs of error growth in the passive tracer outgassing or the ocean circulation are observed to result from AMR.

This work is dedicated to my wonderful wife, Patty. This success is as much yours as it is mine. You are my angel, my all, my very self.

Contents

List of Figures

List of Tables

Acknowledgements

I foremost give thanks to Patty for her patience and understanding, and to my family in general for their never ending support in my endeavors. Thank you Mom for being my best teacher and instilling in me strong study habits. Thank you Dad for your sound advice whenever life has thrown a curve ball. Thank you Bart for being a role model to whom I always look up, and for marrying a fun wife, Janet. Thanks to their daughter Megan for being cute and photogenic. Thanks to Grandma Herrnstein for all of your cheer and happiness. Thanks to my late Grandpa Herrnstein, Grandma Jessie, and Grandpa Bob for many fond childhood memories. Thanks to Mom and Dad Jones for your hospitality, laughter, and for letting Patty move to California. Thanks to Grace for being my "favorite sister-in-law" and letting Patty move to California. Thanks to Bonnie for being my "favorite sister-in-law" and letting Patty move to California. Thanks to Dave for the Irish jigs and for teaching me how to play U2 on the air guitar. Thank you all from the bottom of my heart for the small things like coaching my soccer team or teaching me to spell (still working on that one), and for the big things like seeing me through college and enjoyable family vacations.

I acknowledge and extend my sincere gratitude to my dissertation committee for their instruction and guidance. I can not express my appreciation enough to my academic adviser Garry Rodrigue for keeping me focused on my dissertation goals and never leading me astray. In addition to academic and career advice, I thank you Garry for the knowledge you have imparted on me about numerical methods. I also appreciate that you help maintain the DAS affiliation with LLNL. I thank my technical adviser Mike Wickett for his guidance

in computational physics and for never refusing a research discussion over the past five years. Thank you also, Mike, for always pushing me to take things a step further. I acknowledge Tom Cahill for his honesty and constructive criticism.

I would also like to acknowledge the following individuals who have influenced my research and educational background: Andy Wissink for his knowledge of adaptive mesh refinement and his support pertaining to the SAMRAI software library; Greg Miller for an excellent class in CFD; John DeGroot for chairing my oral exam committee and for making physics fun; Laynam Chang for his endless enthusiasm and inspiration; Jerome Long for his sense of humor and for mentoring SPS; James Grigsby for his integrity and passion of physics; Jody Cutright and Mark Schiller for sparking my interest in science;

I thank my dog Zoey for waking me up at 5:30 am every morning (even on weekends) for an exciting game of fetch.

I also express my appreciation for the LLNL University Relations Program for their financial support through the SEGRF fellowship. I compliment Dee Kindelt and Joanna Allen for their exceptional administrative skills.

Finally, I must give thanks to my friends for all the good memories which make light of those often difficult and tedious time in both graduate and undergraduate school. I particularly thank Chad (a.k.a. Sunfish) for conquering two separate cases of cancer and for his renewed positive outlook on life. I also thank the following individuals: Rob, Jeff, and Dave for the fun times in Livermore and for countless nerdy conversations on Tuesday nights at the Ale House; Dave and Hyen-Kyung for all the tasty Korean cuisine and their introduction to a whole new world of board and card games; Jeff and Hsiao-Fen for the

crazy games of Taboo; Kris, Natasha, Pete, Kristi, and Ben for the adventures in Davis; Ben also for being a fellow and devoted Star Wars fan; my good buddies Brian, Adam (a.k.a. Phil, or is it Norm?), Dylan, and Mark for the never forgotten days in Blacksburg; Justin for demonstrating that flawless guitar performance is in fact possible and rewarding; and Sir Arthur Guinness for being an entrepreneur.

Chapter 1

Introduction

1.1 Overview

Extensive research has been conducted in recent years to reduce atmospheric carbon dioxide emissions from man made sources such as power plants. One proposal is to sequester emissions from coastal power plants to the largest reservoir on the planet - the ocean. Ocean general circulation models are used to simulate the propagation of injected CO_2 over time scales of hundreds of years. Recent studies show that such simulations are highly dependent on horizontal resolution when looking at details near the injection sites [76]. For simplicity, ocean models typically increase resolution globally creating long run times due to an increase in the number of data points and a shorter time step allowed by explicit numerics. Increasing global resolution by a factor of 2 in each horizontal direction increases run times by at least a factor of 16.

In this dissertation, CO_2 sequestration simulations are conducted using a new ocean general circulation model specifically designed for placing high resolution in localized

regions. The concepts of the numerical method known as structured Adaptive Mesh Refinement (AMR) are combined with the field of numerical ocean modeling. A typical AMR grid like that shown in Figure 1.1 is comprised of a logically Cartesian grid overlaid by finer grids called refinement levels. Each refinement level increases resolution by an integer multiple and is represented by a collection of refinement patches that change size and position as the solution evolves over time. When properly applied, AMR simulations require significantly less computation than uniform high resolution simulations and execute with shorter run times as a result.

AMR is one of various techniques practiced in computational physics today for increasing model accuracy. One technique common in computational physics today for increasing model accuracy uses unstructured grids which are irregularly shaped to reflect complicated physical boundaries or spatial variations in the model solution. High resolution in localized areas of interest is possible, but explicit numerics require extremely short time steps on the entire domain. This is inefficient for ocean models with inherently large time and space scales. The AMR method, however, maintains a structured grid, making it readily applicable to ocean and climate models with strictly longitude-latitude based grids. The time stepping procedure applied in structured AMR also has a computational advantage over that used in unstructured grids by allowing regions with coarser resolution to take larger time steps.

The use of AMR in climate and ocean modeling has not progressed far due to the use of numerical methods which do not fit into the traditional AMR methodology. Strategies were developed in this research which allow numerical methods common in ocean models to

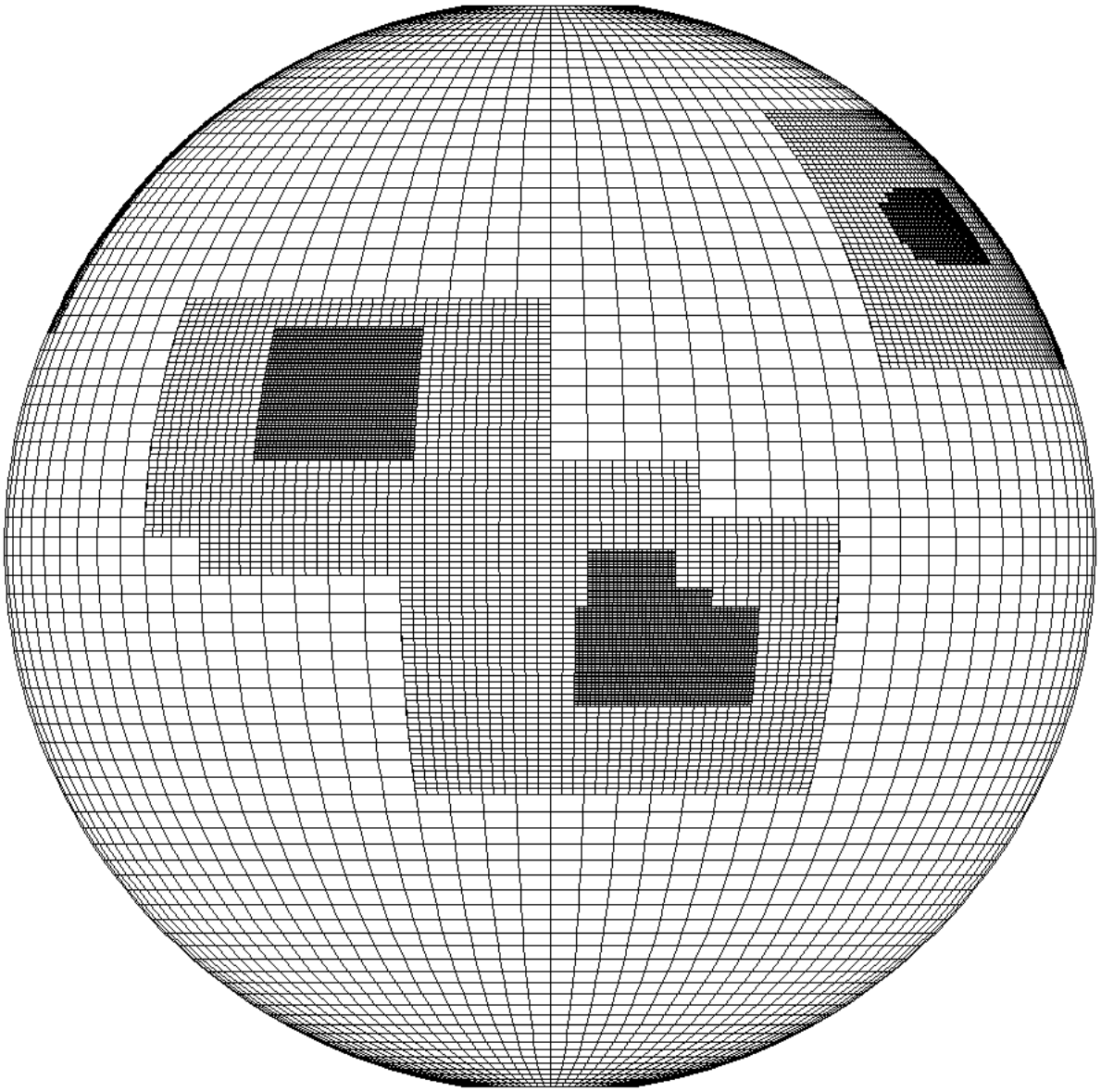


Figure 1.1: An example grid used by the AMR Ocean Model to apply localized resolution increases.

be applied in AMR simulations. These strategies were added to the AMR software library SAMRAI (Structured Adaptive Mesh Refinement Application Infrastructure) developed at Lawrence Livermore National Laboratory. The ocean model, henceforth referenced as the "AMR Ocean Model", is built on top of the amended SAMRAI library. The global ocean is simulated with realistic topography and surface forcing. Resolution is dynamically increased in the longitude and latitude directions using specified refinement criteria for determining the sizes and locations of refinement patches.

Simulations of ocean CO_2 sequestration by direct injection are ideal ocean applications for applying AMR given that the areas of interest (i.e., the injection sites) are highly localized. The simulations performed in this research place refinement patches around injection sites to reveal fine scale details that are unattainable with a uniform coarse resolution and require impractical execution times with a uniform fine resolution. Other applications in climate and ocean modeling where AMR can be advantageous include hurricane studies, regional ocean circulation studies, and regional weather modeling.

The research reported in this thesis is organized as follows. A synopsis of previous and related work is given in the section proceeding this overview. In Chapter 2, the primitive equations describing ocean dynamics are derived from first principles. Descriptions of the numerical methods applied in the ocean model are given in a manner requiring no knowledge of AMR to make the model more applicable to the ocean modeling community. The alterations made to the SAMRAI library for incorporating ocean model numerical methods are described in Chapter 3. Benchmark tests of these AMR strategies are given in Chapter 4 along with limited basin tests of the AMR Ocean Model. The model is applied in

CO_2 sequestration simulations of the North Atlantic and of the Global Ocean in Chapters 5 and 6, respectively.

1.2 Historical Background

1.2.1 Ocean Models

The first widely accepted three dimensional numerical model of the global ocean is credited to Kirk Bryan in 1969 [12]. Many aspects of Bryan's model such as the B-grid staggering of momentum and tracers are still seen in ocean models today. His original model is also widely cited for modeling ocean momentum as a set of external and internal modes coupled together in space and time. The two modes primarily represent motion due to gravity and non-linear fluid motion, respectively. Bryan modeled this coupled system by eliminating surface pressure variations with a "rigid lid" boundary condition at the surface. This allowed his model to use a large enough time step to simulate several 100 years of ocean circulation using the computer resources of the time. In the 1970's, Semtner [61] rewrote much of Bryan's model to take advantage of various computer architectures. His model was further enhanced by Cox [19] with capabilities such as variable horizontal resolution and multiple passive tracers. It became commonly referenced in the 1980's as "the Cox code". Today, ocean models based on the work of these three individuals are classified as Bryan-Cox-Semtner models.

The 1990's gave birth to faster computers with sufficiently larger memory banks. The Cox code was the basis for two of the more popular ocean models in use today known as the Modular Ocean Model (MOM) [53] and the Parallel Ocean Program (POP) [64] devel-

oped at the Geophysical Fluid Dynamics Laboratory and Los Alamos National Laboratory, respectively. Numerous features are present in MOM and POP which are additions or slight modification to the traditional Bryan-Cox-Semtner model. The most notable characteristic used in this thesis is an explicit representation of the free surface height [40]. MOM and POP are the primary resources used for the ocean model design given in Chapter 2.

1.2.2 Adaptive Mesh Refinement

The AMR concept is credited to the work of Marsha Berger, Joseph Oliger, and Phil Collela [9, 8] in the 1980's. It was originally designed for shock wave propagation simulations in which only the wave front was kept highly resolved. Finite differences were used to discretize hyperbolic systems of partial differential equations, and the solutions were advanced in time using a two-time level method. Error estimates and solution gradients were the original criteria used to determine the sections of the domain in need of refinement. Berger developed a clustering algorithm [10] in the early 1990's to more efficiently determine the locations of refinement patches given a logically Cartesian grid with cells flagged as 1 for refinement or 0 otherwise. This algorithm is still in use today.

The increased popularity of AMR at the turn of the century gave rise to AMR software libraries. This made many of the operations for typical AMR applications readily available and simplified the code development otherwise required to build an AMR model from scratch. The library used in this research, SAMRAI, was modified to when constructing the AMR Ocean Model.

1.2.3 Nested Grid Ocean Models

As previously stated, few AMR applications exist in climate or ocean modeling primarily because the fields traditionally use different numerical methods. For example, typical AMR fluid dynamics models use higher-order upwind-weighted numerical methods, while typical primitive equation ocean models use centered differences with leapfrog time integration. Some regional nested models have been tested which show that resolution can be added locally. Spall and Holland [65] developed a nested Bryan-Cox-Semtner ocean model in the early 1990's. Horizontal grid resolution was increased in a static refinement region placed in the center of the model domain. The same time step was used for both the fine grid and coarse grids, and two way grid communication was made possible but without regard to flux matching at refinement interfaces. Standard benchmark tests showed the nested grid applied to short time scale simulations ($\sim$months) reduced error when the area of interest was initialized inside the region of higher resolution.

The work of Spall and Holland was extended by Fox and Maskell [26] to increase the nested resolution in the vertical. A more elaborate time integration procedure was also implemented which allowed larger time steps on the coarser grid. A smoothing operation was applied near grid interfaces to reduce noise and flux matching at the interfaces was again neglected. Short time scales revealed similar results to Spall and Holland.

Laugier et al [44] also produced a Bryan-Cox-Semtner ocean model with static nesting based on the work of Spall and Holland. The nested region increased horizontal resolution and the model was advanced with uniform time steps. Several methods were tested for calculating the external mode of momentum and their test cases showed slight

improvements over the Spall and Holland results. The flux matching issue was once again neglected limiting the nested model to short time integrations with the assumption that the local area of interest will stay inside the nested region.

Nested grid models started to incorporate more of the AMR ideology toward the turn of the century with work by Ginis et al [30] and later by Rowley and Ginis [59]. The nesting capabilities allowed for multiple single patch refinement levels that could change position over time, but the ocean model itself did not fall under the Bryan-Cox-Semtner classification. Specifically, momentum was not separated into external and internal modes and a non-staggered grid was used. A flux matching scheme at refinement interfaces allowed for simulations on longer time scales (∼years) than Spall and Holland.

Blayo and Debreu [11] also developed an AMR strategy for ocean models. Their work was benchmarked using a barotropic modon and a quasi-geostrophhic box model. A Bryan-Cox-Semtner model was not produced and all reported simulations were performed in Cartesian coordinates on a beta plane.

1.2.4 Comparisons to the AMR Ocean Model

This research presents a Bryan-Cox-Semtner ocean model with the traditional AMR capabilities modified accordingly to produce the AMR Ocean Model. External and internal modes of momentum are modeled on a staggered grid with leap frog time integration. Multiple levels of refinement can contain multiple refinement patches which are created and destroyed periodically as the solution steps forward in time using larger time steps on coarser refinement levels. Data are communicated conservatively between adjacent refinement levels and fluxes are matched at interfaces. Benchmark tests of the AMR

algorithms are presented using simplified flow models. AMR convergence is tested for geophysical flows using a limited basin. Global ocean simulations are performed with AMR on time scales up to two decades.

Chapter 2

The Ocean Model

The physics governing the AMR Ocean Model is described in this chapter starting with elementary fluid dynamics principles that build up to a mathematical description of the world's ocean. The model designed for this research has components primarily taken from the Modular Ocean Model [53] (MOM). Many characteristics of MOM are compatible with standard AMR approaches although such uses were not anticipated in its conception. It is of no surprise that several MOM components perform poorly in the context of AMR as a result. In such cases, these minor features have been replaced with those found in other resources [64, 40]. All attention to AMR implementation is addressed in the next chapter.

2.1 The Continuity Equation

The model of a fluid's physical structure depends on the length scale at which it is studied. At microscopic scales it appears as a collection of molecules in constant motion. A fluid at macroscopic scales, however, appears as a continuous medium. The two views are

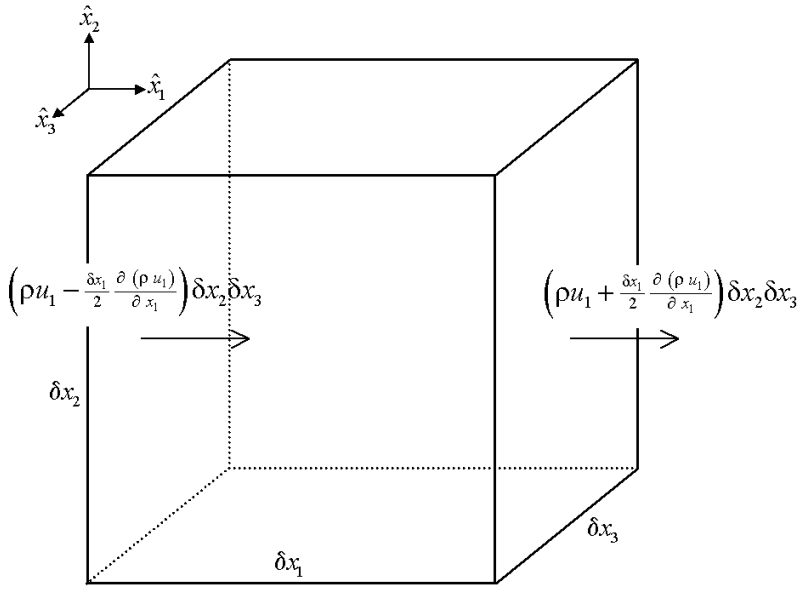


Figure 2.1: A control volume with flow through faces normal to $\hat{x}_1$.

drastically different in theory yet both describe the same material. It is therefore important in fluid dynamics to identify which theory is most relevant for the situation at hand. The length scale of the oceans as presented here ($\sim 10^6\ m$) describes macroscopic phenomena since it is far greater than that of neighboring molecules ($\sim 10^{-8}\ m$). All quantities are considered to be the average manifestation of microscopic processes. This is known as the continuum approach and is valid so long as the length scale is much greater than the mean free path of the molecules in the fluid [41].

A unit of a continuous fluid called a control volume is shown in Figure 2.1 for an arbitrary coordinate system ($\hat{x}_1,\ \hat{x}_2,\ \hat{x}_3$). It has volume, $d\Omega$, and mass, m, equal to

$$d\Omega \quad = \quad \delta x_1\ \delta x_2\ \delta x_3 \tag{2.1}$$

and

$$m \;=\; \rho \; d\Omega \tag{2.2}$$

respectively, where the density, ρ, is constant within $d\Omega$.

If the fluid has velocity $\vec{V} = (u_1, u_2, u_3)$, the flow of mass at the center of the the control volume in the $\hat{x}_1$ direction is $(\rho \; u_1 \; \delta x_2 \delta x_3)$. Taylor expansion gives the approximate flow through the left and right faces of the control volume.

$$F_L \;=\; (\rho \; u_1 \; \delta x_2 \delta x_3) \;-\; \frac{\delta x_1}{2} \frac{\partial \left(\rho \; u_1 \; \delta x_2 \delta x_3\right)}{\partial x_1} \tag{2.3}$$

$$F_R \;=\; (\rho \; u_1 \; \delta x_2 \delta x_3) \;+\; \frac{\delta x_1}{2} \frac{\partial \left(\rho \; u_1 \; \delta x_2 \delta x_3\right)}{\partial x_1} \tag{2.4}$$

The net inward flow due to the movement of mass through faces normal to the $\hat{x}_1$ direction is

$$-(F_R - F_L) \;=\; -\frac{\partial \left(\rho \; u_1\right)}{\partial x_1} \; d\Omega \tag{2.5}$$

for constant δx_i. The combination of (2.5) with similar analysis for the net flow in the $\hat{x}_2$ and $\hat{x}_3$ directions gives the total mass rate of change within $\partial\Omega$.

$$\frac{\partial \rho}{\partial t} \;=\; -\vec{\nabla} \cdot \left(\rho \; \vec{V}\right) \tag{2.6}$$

Equation (2.6) rewritten in terms of the total time derivative, $\frac{d}{dt} = \frac{\partial}{\partial t} + (\vec{V} \cdot \vec{\nabla})$, gives the general form of the continuity equation.

$$\frac{1}{\rho} \frac{d\rho}{dt} \;=\; -\vec{\nabla} \cdot \vec{V} \tag{2.7}$$

2.2 Equations of Motion

Fluid motion in the coordinate system $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$ is derived from Newton's second law of motion

$$\sum \vec{F} \;=\; m\,\vec{a} \tag{2.8}$$

which states the sum of the forces, $\vec{F}$, acting on a body of mass m is proportional to the product of the body's mass and acceleration, $\vec{a}$. The time derivative of the velocity, $\vec{V} = (u_1, u_2, u_3)$, is equal to the acceleration. The forces acting on the control volume are categorized as the net pressure forces, $\vec{F^p}$, the net shear forces, $\vec{F^s}$, and the net external forces, $\vec{F^\chi}$. Equation (2.8) is rewritten as

$$\rho\,\frac{d\vec{V}}{dt}\,d\Omega \;=\; \vec{F^p} + \vec{F^s} + \vec{F^\chi}\;. \tag{2.9}$$

The pressure and shear forces are defined by analyzing the forces exerted on the faces of $d\Omega$ as shown in Figure 2.2 for the $\hat{x}_1$ direction. A control volume with pressure p at its center will have pressure forces at its right and left faces expressed in Taylor series as $-\left(p + \frac{\delta x_i}{2}\frac{\partial p}{\partial x_i}\right)\delta x_j \delta x_k$ and $\left(p - \frac{\delta x_i}{2}\frac{\partial p}{\partial x_i}\right)\delta x_j \delta x_k$ respectively. The net pressure force in the $\hat{x}_i$ direction, F_i^p, is the difference in the pressure forces at the faces,

$$F_i^p \;=\; -\frac{\partial p}{\partial x_i} d\Omega\;. \tag{2.10}$$

The shear force is expressed in terms of the tensor τ_{ji}. The $\hat{x}_i$ direction will have shear stress components τ_{1i}, τ_{2i}, and τ_{3i} defined at the control volume center. Taylor expansion to the faces of $d\Omega$ gives a net shear force of

$$F_i^s \;=\; \left(\frac{\partial \tau_{1i}}{\partial x_1} + \frac{\partial \tau_{2i}}{\partial x_2} + \frac{\partial \tau_{3i}}{\partial x_3}\right) d\Omega\;. \tag{2.11}$$

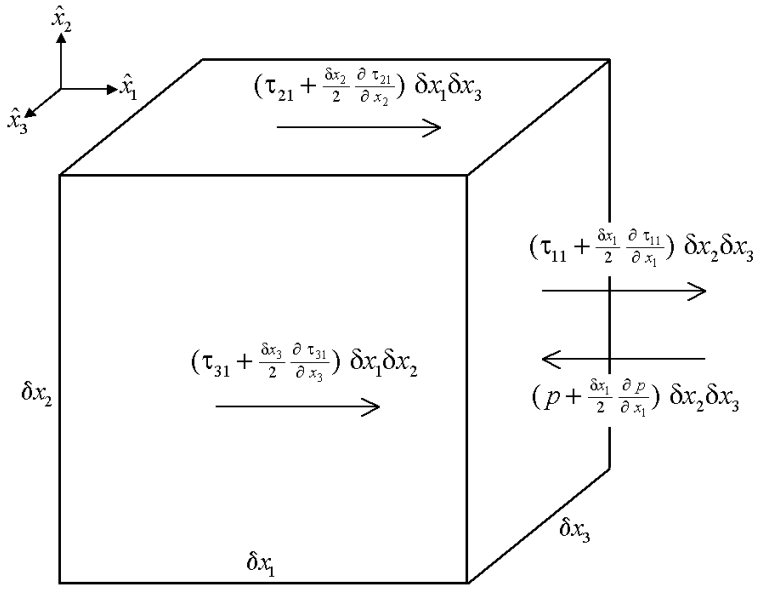


Figure 2.2: Pressure and shear forces in the $\hat{x}_1$ direction on the faces of a control volume.

Numerous texts [4, 6, 75, 69, 53] are available for reference on the definition and simplification of the stress tensor. All follow Stoke's law for linear stress and strain which reduces the shear force to

$$F_i^s = \left(\vec{\nabla} \cdot \mu_i \vec{\nabla} u_i\right) d\Omega \tag{2.12}$$

where μ_i is the viscosity coefficient in the $\hat{x}_i$ direction. The shear force expression is also written in the form

$$F_i^s = \rho\ \mathtt{F}_i\ d\Omega \tag{2.13}$$

$$\mathtt{F}_i = \left(\vec{\nabla} \cdot \nu_i \vec{\nabla} u_i\right) \ , \quad \nu_i = \frac{\mu_i}{\rho} \tag{2.14}$$

which is more convenient for simplification.

The external forces acting on the ocean,

$$\vec{F^{\chi}} \;=\; \left(\rho\ \vec{g}\ +\ \vec{\Pi}\right)\ d\Omega\ , \tag{2.15}$$

are primarily due to gravity, $\rho\vec{g}$, and wind stress, $\vec{\Pi}$. The gravitational acceleration, $\vec{g}$, points toward the center of the Earth. The wind stress definition is similar to the shear stress component in equation (2.11) acting on the control volume face normal to the ocean surface. For example, a wind vector with components equal to $\vec{W} = (\theta_i, \theta_j, \theta_k)$, and a surface normal unit vector $\hat{x}_k$, makes the wind stress equal to

$$\Pi_i \;=\; \frac{\partial \theta_i}{\partial x_k}\ . \tag{2.16}$$

Substituting equations (2.10), (2.12), and (2.15), into (2.9) gives the ocean's equations of motion,

$$\frac{d\vec{V}}{dt} \;=\; -\frac{1}{\rho}\ \vec{\nabla} p\ +\ \vec{\mathtt{F}}\ +\ \vec{g}\ +\ \frac{1}{\rho}\vec{\Pi}\ , \tag{2.17}$$

in a static and arbitrary geometric coordinate system.

2.2.1 Rotating Coordinate System

The coordinates of the equations of motion (2.17) are $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$. The earth is in constant rotation about its north-south axis with rotational velocity, Ω. The coordinate system with which we relate everyday occurrences is in this "rotating" frame of reference and will be represented by $(\hat{x}'_1, \hat{x}'_2, \hat{x}'_3)$. Any vector, $\vec{A}$, can be defined in the static and the rotating frames.

$$\vec{A} \;=\; A_1\ \hat{x}_1\ +\ A_2\ \hat{x}_2\ +\ A_3\ \hat{x}_3 \tag{2.18}$$

$$\vec{A} \;=\; A'_1\ \hat{x}'_1\ +\ A'_2\ \hat{x}'_2\ +\ A'_3\ \hat{x}'_3 \tag{2.19}$$

The rate of change of $\vec{A}$ in the two frames is given by

$$\frac{d_s\ \vec{A}}{dt} = \frac{dA_1}{dt}\ \hat{x}_1 + \frac{dA_2}{dt}\ \hat{x}_2 + \frac{dA_3}{dt}\ \hat{x}_3 \tag{2.20}$$

$$\frac{d_r\ \vec{A}}{dt} = \frac{dA'_1}{dt}\ \hat{x}'_1 + \frac{dA'_2}{dt}\ \hat{x}'_2 + \frac{dA'_3}{dt}\ \hat{x}'_3 \tag{2.21}$$

and are related by the vector property

$$\frac{d_s\ \vec{A}}{dt} = \frac{d_r\ \vec{A}}{dt} + \left(\vec{\Omega}\ \times\ \vec{A}\right) \tag{2.22}$$

where $\vec{\Omega}$ points along the axis of rotation.

Equations (2.18)-(2.22) are now used to express $\frac{d\vec{V}}{dt}$ in the equations of motion, (2.17), in the rotating frame of reference. The vector $\vec{A}$ is defined as an arbitrary position vector, $\vec{r}$. The velocity in both frames is given by (2.20) and (2.21).

$$\vec{V}_s = \frac{d_s\ \vec{r}}{dt} \tag{2.23}$$

$$\vec{V}_r = \frac{d_r\ \vec{r}}{dt} \tag{2.24}$$

Substituting the above into (2.22) gives the relation

$$\vec{V}_s = \vec{V}_r + \left(\vec{\Omega}\ \times\ \vec{r}\right)\ . \tag{2.25}$$

A similar definition of $\vec{A} = \vec{V}_s$ relates the acceleration in the two frames,

$$\frac{d_s\ \vec{V}_s}{dt} = \frac{d_r\ \vec{V}_s}{dt} + \left(\vec{\Omega}\ \times\ \vec{V}_s\right)\ . \tag{2.26}$$

Equation (2.25) is substituted into the right hand side of (2.26) and simplified resulting in

$$\frac{d_s\ \vec{V}_s}{dt} = \frac{d_r\ \vec{V}_r}{dt} + \left(2\ \vec{\Omega}\ \times\ \vec{V}_r\right) + \left(\vec{\Omega}\ \times\ \vec{\Omega}\ \times\ \vec{r}\right)\ . \tag{2.27}$$

The equations of motion, (2.17), can now be expressed in the rotating coordinate system.

$$\frac{d_r\vec{V}_r}{dt} = -\frac{1}{\rho}\ \vec{\nabla}p + \vec{\mathtt{F}} + \vec{g} + \frac{1}{\rho}\vec{\Pi} - \left(2\ \vec{\Omega}\ \times\ \vec{V}_r\right) - \left(\vec{\Omega}\ \times\ \vec{\Omega}\ \times\ \vec{r}\right)\ . \tag{2.28}$$

The last two terms in (2.28) are commonly known as the Coriolis and centrifugal forces respectively. The rotating coordinate system is used for the duration of the text and the $_r$ subscripts are dropped from here on.

2.2.2 Spherical Coordinates

The individual terms in the equations of motion (2.28) will now be described in a spherical coordinate system. The unit vectors are

$$\left(\hat{x}_1',\ \hat{x}_2',\ \hat{x}_3'\right) \quad = \quad \left(\hat{\lambda},\ \hat{\phi},\ \hat{r}\right) \tag{2.29}$$

where $\hat{\lambda}$ points to the east, $\hat{\phi}$ points to the north, and $\hat{r}$ points radially away from the center of the sphere. The respective longitude, latitude, and radial position coordinates are represented by λ, ϕ, and r. The gradient and Laplacian for a scalar, $q(\lambda, \phi, r)$, are

$$\vec{\nabla}\ q \quad = \quad \frac{\hat{\lambda}}{r\cos\phi}\frac{\partial q}{\partial \lambda} \quad + \quad \frac{\hat{\phi}}{r}\frac{\partial q}{\partial \phi} \quad + \quad \hat{r}\ \frac{\partial q}{\partial r} \tag{2.30}$$

$$\nabla^2\ q \quad = \quad \frac{1}{r^2\cos^2\phi}\frac{\partial^2 q}{\partial \lambda^2} \quad + \quad \frac{1}{r^2\cos\phi}\frac{\partial}{\partial \phi}\left(\cos\phi\ \frac{\partial q}{\partial \phi}\right) \quad + \quad \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\ \frac{\partial q}{\partial \phi}\right)\ . \tag{2.31}$$

The velocity vector, $\vec{V} = (u,\ v,\ w)$, in spherical coordinates is

$$\vec{V} \quad = \quad \hat{\lambda}\ r\cos\phi\frac{\partial \lambda}{\partial t} \quad + \quad \hat{\phi}\ r\frac{\partial \phi}{\partial t} \quad + \quad \hat{r}\ \frac{\partial r}{\partial t}\ , \tag{2.32}$$

and the divergence and curl of $\vec{V}$ are

$$\vec{\nabla}\cdot\vec{V} = \frac{1}{r\cos\phi}\left[\frac{\partial u}{\partial\lambda} + \frac{\partial\left(\cos\phi\ v\right)}{\partial\phi}\right] + \frac{1}{r^2}\frac{\partial\left(r^2 w\right)}{\partial r} \tag{2.33}$$

$$\begin{aligned}\vec{\nabla}\times\vec{V} &= \frac{\hat{\lambda}}{r}\left[\frac{\partial}{\partial\phi}\left(w\right) - \frac{\partial}{\partial r}\left(rv\right)\right] \\ &+ \frac{\hat{\phi}}{r\cos\phi}\left[\frac{\partial}{\partial r}\left(r\cos\phi\ u\right) - \frac{\partial}{\partial\lambda}\left(w\right)\right] \\ &+ \frac{\hat{r}}{r^2\cos\phi}\left[\frac{\partial}{\partial\lambda}\left(rv\right) - \frac{\partial}{\partial\phi}\left(r\cos\phi\ u\right)\right] .\end{aligned} \tag{2.34}$$

Simplifying the velocity total time derivative, $\frac{d\vec{V}}{dt} = \frac{\partial\vec{V}}{\partial t} + (\vec{V}\cdot\vec{\nabla})\vec{V}$, in spherical coordinates is a tedious task if done directly. An easier approach found in [31] is to use the following vector identity.

$$\left(\vec{V}\cdot\vec{\nabla}\right)\vec{V} = \frac{1}{2}\vec{\nabla}\left(\vec{V}\cdot\vec{V}\right) + \left(\vec{\nabla}\times\vec{V}\right)\times\vec{V} \tag{2.35}$$

The right hand side terms of (2.35) reduce to

$$\begin{aligned}\frac{1}{2}\vec{\nabla}\left(\vec{V}\cdot\vec{V}\right) &= \frac{\hat{\lambda}}{r\cos\phi}\left[u\frac{\partial u}{\partial\lambda} + v\frac{\partial v}{\partial\lambda} + w\frac{\partial w}{\partial\lambda}\right] \\ &+ \frac{\hat{\phi}}{r}\left[u\frac{\partial u}{\partial\phi} + v\frac{\partial v}{\partial\phi} + w\frac{\partial w}{\partial\phi}\right] \\ &+ \hat{r}\left[u\frac{\partial u}{\partial r} + v\frac{\partial v}{\partial r} + w\frac{\partial w}{\partial r}\right] ,\end{aligned} \tag{2.36}$$

and

$$\begin{aligned}\left(\vec{\nabla}\times\vec{V}\right)\times\vec{V} &= \hat{\lambda}\left[\frac{w}{r}\frac{\partial\left(ru\right)}{\partial r} - \frac{w}{r\cos\phi}\frac{\partial w}{\partial\lambda} - \frac{v}{r\cos\phi}\frac{\partial v}{\partial\lambda} + \frac{v}{r\cos\phi}\frac{\partial\left(u\cos\phi\right)}{\partial\phi}\right] \\ &- \hat{\phi}\left[\frac{w}{r}\frac{\partial w}{\partial\phi} - \frac{w}{r}\frac{\partial\left(rv\right)}{\partial r} - \frac{u}{r\cos\phi}\frac{\partial v}{\partial\lambda} + \frac{u}{r\cos\phi}\frac{\partial\left(u\cos\phi\right)}{\partial\phi}\right] \\ &+ \hat{r}\left[\frac{v}{r}\frac{\partial w}{\partial\phi} - \frac{v}{r}\frac{\partial\left(rv\right)}{\partial r} - \frac{u}{r}\frac{\partial\left(ru\right)}{\partial r} + \frac{u}{r\cos\phi}\frac{\partial w}{\partial\lambda}\right] .\end{aligned} \tag{2.37}$$

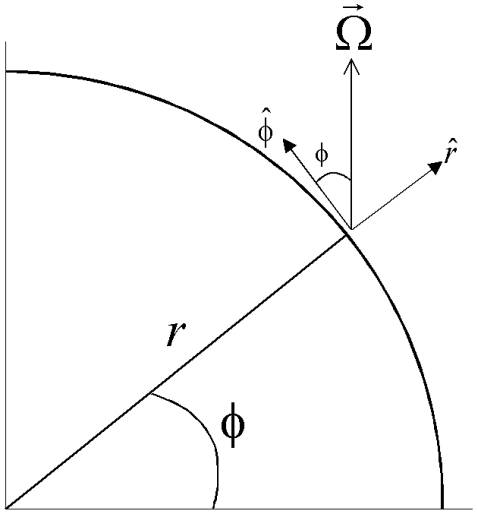


Figure 2.3: Rotation vector orientation ($\vec{\Omega}$).

The vector identity is reduced to

$$\begin{aligned}\left(\vec{V}\cdot\vec{\nabla}\right)\vec{V} &= \hat{\lambda}\left[\frac{wu}{r} - \frac{uv\tan\phi}{r} + \left(\vec{V}\cdot\vec{\nabla}\right)u\right] \\ &+ \hat{\phi}\left[\frac{wv}{r} + \frac{u^2\tan\phi}{r} + \left(\vec{V}\cdot\vec{\nabla}\right)v\right] \\ &+ \hat{r}\left[-\frac{u^2+v^2}{r} + \left(\vec{V}\cdot\vec{\nabla}\right)w\right]\end{aligned} \tag{2.38}$$

where $(\vec{V}\cdot\vec{\nabla})q = \frac{u}{r\cos\phi}\frac{\partial q}{\partial\lambda} + \frac{v}{r}\frac{\partial q}{\partial\phi} + w\frac{\partial q}{\partial r}$. The total time derivative of the velocity in spherical coordinates becomes

$$\begin{aligned}\frac{d\vec{V}}{dt} &= \hat{\lambda}\left[\frac{\partial u}{\partial t} + \left(\vec{V}\cdot\vec{\nabla}\right)u + \frac{wu}{r} - \frac{uv\tan\phi}{r}\right] \\ &+ \hat{\phi}\left[\frac{\partial v}{\partial t} + \left(\vec{V}\cdot\vec{\nabla}\right)v + \frac{wv}{r} + \frac{u^2\tan\phi}{r}\right] \\ &+ \hat{r}\left[\frac{\partial w}{\partial t} + \left(\vec{V}\cdot\vec{\nabla}\right)w - \frac{u^2+v^2}{r}\right].\end{aligned} \tag{2.39}$$

The Coriolis and centrifugal terms of the equations of motion, (2.28), are written in spherical coordinates by first defining the vector components of $\vec{\Omega}$. Figure 2.3 shows a

cross section of the sphere with $\hat{\lambda}$ pointing into the paper. The rotation vector with respect to the coordinate axis at position $(\lambda,\ \phi,\ r)$ has vector components

$$\vec{\Omega} \ = \ \Omega \ \Big(\cos\phi \ \hat{\phi} \ - \ \sin\phi \ \hat{r}\Big) \ . \tag{2.40}$$

The Coriolis and centrifugal terms become

$$-\Big(2\vec{\Omega}\times\vec{V}\Big) \ = \ \big(-f'w \ + \ fv\big)\,\hat{\lambda} \ - \ (fu)\,\hat{\phi} \ + \ \big(f'u\big)\,\hat{r} \tag{2.41}$$

$$-\Big(\vec{\Omega}\times\vec{\Omega}\times\vec{V}\Big) \ = \ \frac{rf'}{4}\Big(-f\hat{\phi} \ + \ f'\hat{r}\Big) \tag{2.42}$$

where $f = 2\Omega\sin\phi$ and $f' = 2\Omega\cos\phi$. Velocity components are coupled by the Coriolis force vector. The centrifugal force vector points perpendicular to the rotation axis and its magnitude, $r\Omega^2\cos\phi$, goes to zero at the poles giving the Earth a slight bulge near the equator.

Substituting (2.30), (2.39), (2.41), and (2.42) into (2.28) gives the following form of the equations of motion in spherical coordinates

$$\frac{du}{dt} \ + \ \frac{wu}{r} \ - \ \frac{vu\tan\phi}{r} \ = \ -\frac{1}{\rho\ r\cos\phi}\frac{\partial p}{\partial\lambda} \ + \ \mathtt{F}_\lambda \ + \ \frac{1}{\rho}\frac{\partial\theta_\lambda}{\partial r} \ + \ fv \ - \ f'w \tag{2.43}$$

$$\frac{dv}{dt} \ + \ \frac{wv}{r} \ + \ \frac{u^2\tan\phi}{r} \ = \ -\frac{1}{\rho\ r}\frac{\partial p}{\partial\phi} \ + \ \mathtt{F}_\phi \ + \ \frac{1}{\rho}\frac{\partial\theta_\phi}{\partial r} \ - \ fu - \ \frac{f'f}{4r} \tag{2.44}$$

$$\frac{dw}{dt} \ - \ \frac{u^2+v^2}{r} \ = \ -\frac{1}{\rho}\frac{\partial p}{\partial r} \ + \ \mathtt{F}_r \ + \ f'u \ + \ \frac{f'^{\,2}}{4r} - \ g \ . \tag{2.45}$$

2.3 Scaling and Approximations

Simplifications are now made to simplify the continuity equation, (2.7), and the equations of motion, (2.43)-(2.45). The Boussinesq approximation is made which says changes in density of fluids such as ocean water are small enough to be considered constant,

except when giving rise to buoyancy forces [29]. All instances of ρ in the continuity equation and the horizontal equations of motion are replaced with the average density, ρ_0. As a result, the horizontal equations of motion become equal to the equations for change in horizontal momentum $\left(\rho_0 \frac{du}{dt} = \frac{d(\rho_0 u)}{dt}\right)$. Another consequence is that the continuity equation simplifies to

$$\vec{\nabla} \cdot \vec{V} \;\; = \;\; 0 \tag{2.46}$$

since the time derivative of a constant density field is zero. Fluids with this property are said to be "incompressible", and the movement of mass is solely due to spatial variations in the velocity field. The continuity equation for incompressible flow in spherical coordinates is expressed as

$$\frac{1}{r \cos\phi} \left[\frac{\partial u}{\partial \lambda} \; + \; \frac{\partial \cos\phi \; v}{\partial \phi} \right] \; + \; \frac{\partial w}{\partial r} \; + \; \frac{2w}{r} \;\; = \;\; 0 \; . \tag{2.47}$$

An approximation is made to the radial coordinate, r, given that the maximum ocean depth is relatively small compared to the average radius of the Earth. All instances of r are approximated as $r \approx R$. The terms r and $\hat{r}$ are interchanged with the more intuitive notation z and $\hat{z}$.

The values used for constants in the momentum and continuity equations are:

$$\begin{aligned}
\text{average density} \quad (\rho_0) \;\; &= \;\; 1.035 \times 10^3 \; kg/m^3 \\
\text{radius of the Earth} \quad (R) \;\; &= \;\; 6.37122 \times 10^6 \; m \\
\text{rotational velocity} \quad (\Omega) \;\; &= \;\; 7.29212 \times 10^{-5} \; s^{-1} \\
\text{gravity} \quad (g) \;\; &= \;\; 9.80661 \; m/s^2
\end{aligned}$$

These constants are combined with typical scale values [78] for each variable to estimate the magnitude of each term in the momentum and continuity equations.

$$\begin{aligned}
\text{horizontal length scale} \quad & (R\Delta\lambda, R\Delta\phi) \sim 10^6\ m \\
\text{vertical length scale} \quad & (\Delta z) \sim 4\times 10^4\ m \\
\text{horizontal velocity scale} \quad & (\Delta u, \Delta v) \sim 10^{-1}\ m/s \\
\text{vertical velocity scale} \quad & (\Delta w) \sim 10^{-4}\ m/s \\
\text{horizontal pressure scale} \quad & (\Delta P_H) \sim 10^3\ Pa \\
\text{vertical pressure scale} \quad & (\Delta P_V) \sim 4\times 10^4\ Pa \\
\text{time scale} \quad & (\Delta t) \sim 10^7\ s
\end{aligned}$$

The last term in (2.47) is ignored since it is roughly four orders of magnitude smaller than the other terms. The continuity equation then becomes

$$\frac{1}{R\cos\phi}\left[\frac{\partial u}{\partial\lambda} + \frac{\partial(\cos\phi\ v)}{\partial\phi}\right] + \frac{\partial w}{\partial z} = 0\ . \tag{2.48}$$

The terms containing w in the horizontal momentum equations, (2.43) and (2.44), are ignored since they are the smallest terms by roughly three orders of magnitude. The centrifugal terms are ignored since they are small compared to g. The vertical equation of motion, (2.45), is dominated by the pressure and gravity terms by roughly twelve orders of magnitude. Ignoring all other terms in this equation results in a balance between the gravitational force and the vertical pressure gradient and is referred to as the hydrostatic approximation.

$$\frac{\partial p}{\partial z} = -\rho g \tag{2.49}$$

The density is determined from experimental data which was reduced to a polynomial expression, $\rho = \rho(T, S, z)$, by the Joint Panel on Oceanographic Tables and Standards [71, 29, 64]. This necessitates equations to describe the ocean's temperature, T, and salinity, S. These quantities are treated as tracers and modeled with advection and diffusion properties.

2.4 The Primitive Equations

The full set of primitive equations after scaling approximations are

$$\frac{du}{dt} - u\left(\frac{f + v\tan\phi}{R}\right) = -\frac{1}{\rho_0 R\cos\phi}\frac{\partial p}{\partial\lambda} + \mathtt{F}_\lambda + \frac{1}{\rho_0}\frac{\partial\theta_\lambda}{\partial z} \tag{2.50}$$

$$\frac{dv}{dt} + u\left(\frac{f + u\tan\phi}{R}\right) = -\frac{1}{\rho_0 R}\frac{\partial p}{\partial\phi} + \mathtt{F}_\phi + \frac{1}{\rho_0}\frac{\partial\theta_\phi}{\partial z} \tag{2.51}$$

$$\frac{\partial w}{\partial z} = -\frac{1}{R\cos\phi}\left[\frac{\partial u}{\partial\lambda} + \frac{\partial\left(\cos\phi\ v\right)}{\partial\phi}\right] \tag{2.52}$$

$$\frac{\partial p}{\partial z} = -\rho g \tag{2.53}$$

$$\rho = \rho\left(T, S, p\right) \tag{2.54}$$

$$\frac{dT}{dt} = \vec{\nabla}_h\cdot\left(\mu^T\vec{\nabla}_h T\right) + \frac{\partial}{\partial z}\left(\kappa^T\frac{\partial T}{\partial z}\right) \tag{2.55}$$

$$\frac{dS}{dt} = \vec{\nabla}_h\cdot\left(\mu^S\vec{\nabla}_h S\right) + \frac{\partial}{\partial z}\left(\kappa^S\frac{\partial S}{\partial z}\right) \tag{2.56}$$

where

$$\vec{\nabla}_h\cdot\left(a\vec{\nabla}_h b\right) = \frac{1}{R^2\cos^2\phi}\frac{\partial}{\partial\lambda}\left(a\frac{\partial b}{\partial\lambda}\right) + \frac{1}{R^2\cos\phi}\frac{\partial}{\partial\phi}\left(a\cos\phi\frac{\partial b}{\partial\phi}\right) \tag{2.57}$$

is the horizontal components of the gradient operator. The diffusion coefficients for the temperature and salinity equations are given by μ^T, μ^S, κ^T, and κ^S. The horizontal viscosity

terms in spherical coordinates are

$$\mathtt{F}_\lambda = \vec{\nabla}_h \cdot \left(\nu \vec{\nabla}_h u\right) + \nu \left(\frac{1-\tan^2\phi}{R^2} u - \frac{2\sin\phi}{R^2\cos^2\phi}\frac{\partial v}{\partial \lambda}\right) + \frac{\partial}{\partial z}\left(\kappa \frac{\partial u}{\partial z}\right) \quad (2.58)$$

$$\mathtt{F}_\phi = \vec{\nabla}_h \cdot \left(\nu \vec{\nabla}_h v\right) + \nu \left(\frac{1-\tan^2\phi}{R^2} v + \frac{2\sin\phi}{R^2\cos^2\phi}\frac{\partial u}{\partial \lambda}\right) + \frac{\partial}{\partial z}\left(\kappa \frac{\partial v}{\partial z}\right) . \quad (2.59)$$

See [53] for more details.

A set of boundary conditions is imposed on the above system of equations. Horizontal velocity and tracer fluxes are zero at lateral boundaries. Tracer and momentum fluxes are prescribed from data at the surface ($z = 0$) and are zero at the ocean floor ($z = -H$). The vertical velocity at the surface is equal to the change in surface height, η, and attributed to changes in topography at the ocean floor.

$$w(0) = \frac{d\eta}{dt} \quad (2.60)$$

$$w(-H) = -\left(\vec{V} \cdot \vec{\nabla}\right) H \quad (2.61)$$

2.5 The Bryan-Cox-Semtner Model

Most ocean models apply the methodology originally introduced by Bryan [12] and later carried on by Semtner [13] and Cox [19]. This involves discretizing the primitive equations (2.50) - (2.56) with finite differences on a structured longitude-latitude (long-lat) grid and accounting for complicated topography. The original Bryan-Cox-Semtner model was amended by Killworth et al [40] to include an explicitly representation of spatial variations in free surface height. Surface pressure accounts for barotropic motion dominated by gravity waves traveling at speeds near 200 m/s. Such speeds are much faster than baroclinic motion generated by processes such as convection, viscosity, and density variations.

The two motions represent different physical properties and are equally important for ocean circulation studies.

The barotropic flow is modeled as the depth integrated velocity

$$(U, V) \;=\; \int_{-H}^{\eta} (u, v)\; dz \;. \tag{2.62}$$

Baroclinic flow is defined as the perturbation (u', v') from the depth mean flow. The horizontal velocities are the sum of the barotropic and baroclinic contributions.

$$(u, v) \;=\; \frac{1}{H}\,(U, V) \;+\; \left(u', v'\right) \tag{2.63}$$

Pressure is obtained from the hydrostatic relation, (2.53), and is

$$p \;=\; p_s \;+\; \int_{z}^{0} \rho g\; dz \tag{2.64}$$

where the surface pressure

$$p_s \;=\; \rho_0 g \eta \tag{2.65}$$

is attributed to change in surface height. The surface pressure is temporarily set to zero when solving equations (2.50) and (2.51) for a temporary velocity field, (u^*, v^*). The baroclinic velocities are obtained by subtracting off the depth mean.

$$\left(u', v'\right) \;=\; (u^*, v^*) \;-\; \frac{1}{H} \int_{-H}^{0} (u^*, v^*)\, dz \tag{2.66}$$

Barotropic motion is accounted in the continuity equation by integrating (2.52) over $\hat{z}$ and applying the boundary conditions on w to give

$$\frac{d\eta}{dt} \;=\; -\frac{1}{R \cos\phi} \left[\; \frac{\partial U}{\partial \lambda} \;+\; \frac{\partial \left(\cos\phi\; V\right)}{\partial \phi} \;\right] \;. \tag{2.67}$$

The barotropic equations of motion are the primitive equations for horizontal velocity integrated over $\hat{z}$

$$\frac{\partial U}{\partial t} - fV = -\frac{gH}{R\cos\phi}\frac{\partial \eta}{\partial \lambda} + X \tag{2.68}$$

$$\frac{\partial V}{\partial t} + fU = -\frac{gH}{R}\frac{\partial \eta}{\partial \phi} + Y \tag{2.69}$$

where the forcing terms

$$\begin{aligned} X = & -\int_{-H}^{\eta} (\vec{V}\cdot\vec{\nabla})u \; dz + \int_{-H}^{\eta}\left(\frac{uv\tan\phi}{R}\right)dz \\ & - \frac{g}{\rho_0 R\cos\phi}\int_{-H}^{\eta}\int_{z}^{0}\frac{\partial \rho}{\partial \lambda}\; dz' \; dz + \int_{-H}^{\eta} \mathtt{F}_\lambda \; dz + \frac{1}{\rho_0}\theta_\lambda \end{aligned} \tag{2.70}$$

and

$$\begin{aligned} Y = & -\int_{-H}^{\eta} (\vec{V}\cdot\vec{\nabla})v \; dz - \int_{-H}^{\eta}\left(\frac{u^2\tan\phi}{R}\right)dz \\ & - \frac{g}{\rho_0 R}\int_{-H}^{\eta}\int_{z}^{0}\frac{\partial \rho}{\partial \phi}\; dz' \; dz + \int_{-H}^{\eta} \mathtt{F}_\phi \; dz + \frac{1}{\rho_0}\theta_\phi \end{aligned} \tag{2.71}$$

are the depth integral of the right hand side terms in (2.50) and (2.51). The barotropic system can also be expressed with viscosity

$$\frac{\partial U}{\partial t} - fV = -\frac{gH}{R\cos\phi}\frac{\partial \eta}{\partial \lambda} + \vec{\nabla}_h\cdot\left(\nu\vec{\nabla}_h U\right) + X \tag{2.72}$$

$$\frac{\partial V}{\partial t} + fU = -\frac{gH}{R}\frac{\partial \eta}{\partial \phi} + \vec{\nabla}_h\cdot\left(\nu\vec{\nabla}_h V\right) + Y \tag{2.73}$$

where the forcing terms are defined as

$$\begin{aligned} X = & -\int_{-H}^{\eta} (\vec{V}\cdot\vec{\nabla})u \; dz + \int_{-H}^{\eta}\left(\frac{uv\tan\phi}{R}\right)dz \\ & - \frac{g}{\rho_0 R\cos\phi}\int_{-H}^{\eta}\int_{z}^{0}\frac{\partial \rho}{\partial \lambda}\; dz' \; dz + \int_{-H}^{\eta}\left(\mathtt{F}_\lambda - \vec{\nabla}_h\cdot\left(\nu\vec{\nabla}_h u\right)\right)dz + \frac{1}{\rho_0}\theta_\lambda \end{aligned} \tag{2.74}$$

and

$$
\begin{aligned}
Y \;=\; & -\int_{-H}^{\eta} (\vec{V}\cdot\vec{\nabla})v \; dz \;-\; \int_{-H}^{\eta} \left(\frac{u^2 \tan\phi}{R}\right) dz \\
& -\frac{g}{\rho_0 R}\int_{-H}^{\eta}\int_{z}^{0} \frac{\partial \rho}{\partial \phi}\; dz' \; dz \;+\; \int_{-H}^{\eta} \left(\mathtt{F}_{\phi} - \vec{\nabla}_h \cdot \left(\nu \vec{\nabla}_h v\right)\right) dz \;+\; \frac{1}{\rho_0}\theta_{\phi} \; .
\end{aligned}
\tag{2.75}
$$

In summary, the barotropic system is represented by equations (2.67)-(2.71) or equations (2.67) and (2.72)-(2.75). The former models barotropic viscosity in the forcing terms, (2.70) and (2.71), and the latter models barotropic viscosity directly in barotropic velocity equations, (2.72) and (2.73). The latter form was suggested by Killworth [40] as a means of reducing numerical artifacts produced from discretization and is the form used by the AMR Ocean Model.

2.6 Spatial Discretization

The Earth is discretized on a longitude-latitude (long-lat) grid with uniform angular grid spacing $(\Delta\lambda, \Delta\phi)$. A structured grid such as this is important for AMR implementation as will be seen in the next chapter. Tracer and pressure variables (T) are defined at cell centers, and horizontal momentum variables (U) are arranged at cell vertices (see Figure 2.4). Arakawa and Lamb [3] showed that finite difference schemes with this type of variable staggering yield the best representation of geostrophic adjustments. The coordinates of $T_{i,j,k}$ and $U_{i,j,k}$ will be referred to as (λ_i^T, ϕ_j^T) and (λ_i^U, ϕ_j^U) respectively.

The ocean depth is divided into layers of various thicknesses as shown in Figure 2.5. Vertical velocities are positioned at the bottom and top of each layer. The depth

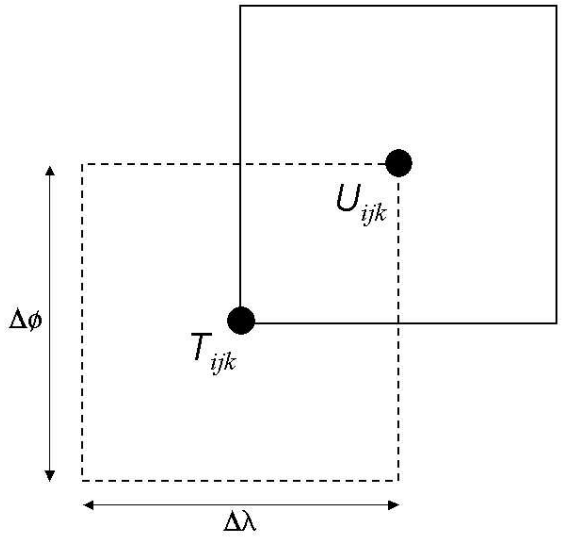


Figure 2.4: Arakawa B-Grid for T and U Variables.

locations of T and U variables are set according to the relation

$$z_k^{TU} = \begin{cases} \frac{1}{2} z_1^w & , \quad k = 1 \\ 2z_{k-1}^w - z_{k-1}^{TU} & , \quad k > 1 \end{cases} \tag{2.76}$$

which ensures w_k is equally spaced between TU_k and TU_{k+1}. This arrangement positions T and U variables off center when the vertical resolution in nonuniform, but does not lessen numerical accuracy provided the grid resolution still maps to a smooth function [70].

2.6.1 Topography

Topography is set using data measured by NOAA [63] which is read in from a file and interpolated to the location (λ_i^T, ϕ_j^T) to define $H_{i,j}^T$. Depths less than z_2^w are set to zero. The depth variable is smoothed using a Gaussian weighted smoothing function, and then rounded to the closest z_k^w. A land mask variable, $M_{i,j,k}^T$, distinguishes between land and

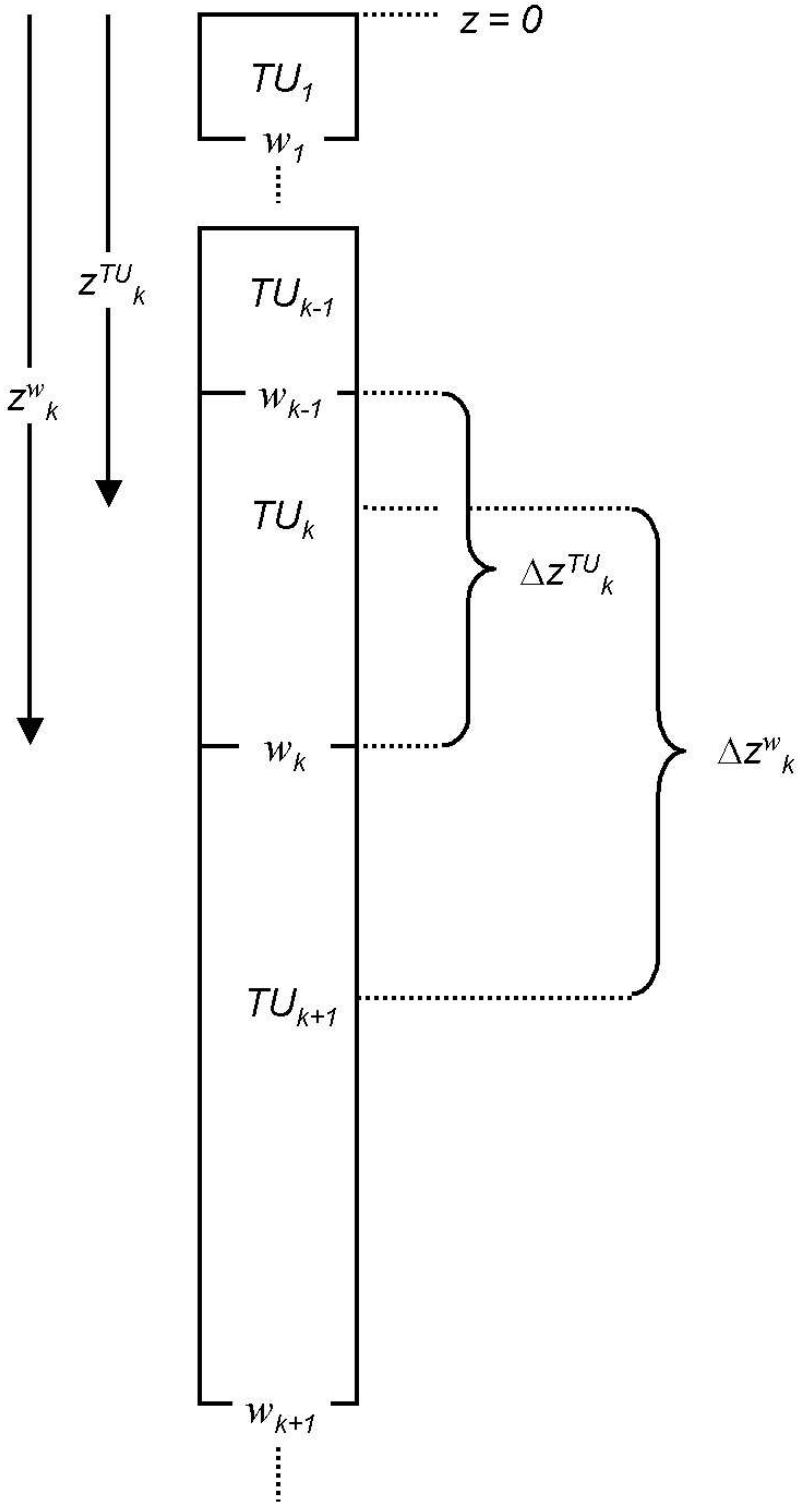


Figure 2.5: Vertical Grid for T, U, and w Variables.

water cells and is defined as

$$M^T_{i,j,k} = \begin{cases} 0 & , \quad z^w_{i,j,k} > H^T_{i,j} \\ & \\ 1 & , \quad z^w_{i,j,k} <= H^T_{i,j} \end{cases} . \tag{2.77}$$

The U cell depth and mask are defined as the minimum of those on neighboring T cells.

$$H^U_{i,j,k} = \min\left(H^T_{i,j,k},\ H^T_{i+1,j,k},\ H^T_{i,j+1,k},\ H^T_{i+1,j+1,k} \right) \tag{2.78}$$

$$M^U_{i,j,k} = \min\left(M^T_{i,j,k},\ M^T_{i+1,j,k},\ M^T_{i,j+1,k},\ M^T_{i+1,j+1,k} \right) \tag{2.79}$$

2.6.2 Finite Difference Operators

The incompressibility relation (2.46) is used to express all advective terms in "conservative flux form",

$$\left(\vec{V} \cdot \vec{\nabla}\right) q = \vec{\nabla} \cdot \left(\vec{V} q\right) , \tag{2.80}$$

which discretely conserves the quantity q when derivative terms are numerically cast as a difference of fluxes through the faces of the control volume. Variables are spatially averaged using the operators

$$\overline{q_{i,j,k}}^{\lambda} = \frac{1}{2}(q_{i,j,k} + q_{i-1,j,k}) \tag{2.81}$$

$$\overline{q_{i,j,k}}^{\phi} = \frac{1}{2}(q_{i,j,k} + q_{i,j-1,k}) \tag{2.82}$$

$$\overline{q_{i,j,k}}^{z} = \frac{1}{2}(q_{i,j,k} + q_{i,j,k+1}) \tag{2.83}$$

to define fluxes at the faces of T and U cells. A shorthand operator notation is also used for taking the difference of a variable.

$$\delta^{\lambda}(q_{i,j,k}) = \frac{q_{i+1,j,k} - q_{i,j,k}}{R\ \Delta\lambda} \tag{2.84}$$

$$\delta^{\phi}(q_{i,j,k}) = \frac{q_{i,j+1,k} - q_{i,j,k}}{R\ \Delta\phi} \tag{2.85}$$

$$\delta^{z}(q_{i,j,k}) = \frac{q_{i,j,k-1} - q_{i,j,k}}{\Delta z} \tag{2.86}$$

2.6.3 Tracer Equations

The equation for tracer transport is of the form

$$\left.\frac{\partial\varphi}{\partial t}\right|_{i,j,k} = -ADV(\varphi_{i,j,k}) + DIFF^{\lambda\phi}(\varphi_{i,j,k}) + DIFF^{z}(\varphi_{i,j,k}) \tag{2.87}$$

where $\varphi_{i,j,k}$ represents the T cell centered value for temperature, salinity, or some passive tracer. Advective velocities are first defined on the western and southern faces of T cells by averaging state velocities at cell vertices. The vertical velocity is then calculated using the continuity equation (2.52).

$$\text{Adv\_Vel\_Tw}_{i,j,k} = \overline{u_{i-1,j,k}}^{\phi} \tag{2.88}$$

$$\text{Adv\_Vel\_Ts}_{i,j,k} = \cos\phi^{U}_{j-1}\ \overline{v_{i,j-1,k}}^{\lambda} \tag{2.89}$$

$$\text{Adv\_Vel\_Tb}_{i,j,k-1} = \text{Adv\_Vel\_Tb}_{i,j,k} - \left(\frac{\Delta z^{T}_{k}}{\cos\phi^{T}_{j}}\right) \times \left[\delta^{\lambda}(\text{Adv\_Vel\_Tw}_{i,j,k}) + \delta^{\phi}(\text{Adv\_Vel\_Ts}_{i,j,k})\right] \tag{2.90}$$

This definition sets the tracer advective velocity equal to zero on any tracer cell face that touches land. The vertical advective velocity at the surface is explicitly set to zero to maintain tracer conservation. This allows a small divergence in the velocity field at the surface as the surface height changes.

The advective velocities are used to calculate the tracer advective flux on each face of the tracer cell.

$$\text{Adv\_Flux\_Tw}_{i,j,k} = \text{Adv\_Vel\_Tw}_{i,j,k}\ \overline{\varphi_{i,j,k}}^{\lambda} \tag{2.91}$$

$$\text{Adv\_Flux\_Ts}_{i,j,k} = \text{Adv\_Vel\_Ts}_{i,j,k}\ \overline{\varphi_{i,j,k}}^{\phi} \tag{2.92}$$

$$\text{Adv\_Flux\_Tb}_{i,j,k} = \text{Adv\_Vel\_Tb}_{i,j,k}\ \overline{\varphi_{i,j,k}}^{z} \tag{2.93}$$

The advective term of the tracer transport equation represents the net advection through the cell and is defined as the difference of the advective fluxes.

$$\begin{aligned} ADV(\varphi_{i,j,k}) = & \frac{1}{\cos\phi_j^T}\ \delta^{\lambda}(\text{Adv\_Flux\_Tw}_{i,j,k}) \\ & + \frac{1}{\cos\phi_j^T}\ \delta^{\phi}(\text{Adv\_Flux\_Ts}_{i,j,k}) \\ & + \delta^{z}(\text{Adv\_Flux\_Tb}_{i,j,k}) \end{aligned} \tag{2.94}$$

The diffusion terms are calculated in a similar manner by first defining fluxes at T cell faces,

$$\text{Diff\_Flux\_Tw}_{i,j,k} = \mu^{\varphi}\frac{1}{\cos\phi_j^T}\ \delta^{\lambda}(\varphi_{i-1,j,k})\ M_{i-1,j,k}^T\ M_{i,j,k}^T \tag{2.95}$$

$$\text{Diff\_Flux\_Ts}_{i,j,k} = \mu^{\varphi}\cos\phi_{j-1}^U\ \delta^{\phi}(\varphi_{i,j-1,k})\ M_{i,j-1,k}^T\ M_{i,j,k}^T \tag{2.96}$$

$$\text{Diff\_Flux\_Tb}_{i,j,k} = \begin{cases} \frac{\Delta z_1^{TU}}{\tau_{\varphi}}\ \left(\varphi_{i,j}^{surf} - \varphi_{i,j,1}\right)\ M_{i,j,1}^T & , \quad k = 0 \\ \kappa_k^{\varphi}\ \delta^{z}(\varphi_{i,j,k+1})\ M_{i,j,k+1}^T\ M_{i,j,k}^T & , \quad k > 0 \end{cases} \tag{2.97}$$

where μ^{φ} and κ_k^{φ} are the tracer's horizontal and vertical diffusion coefficients respectively. The surface flux at $k = 0$ is defined by Newtonian restoring where $\varphi_{i,j}^{surf}$ is the reference value of the tracer at the surface, and τ_{φ} is the restoring time scale. The net diffusion

through the cell is the difference in diffusive fluxes,

$$\begin{aligned} DIFF^{\lambda\phi}(\varphi_{i,j,k}) &= \frac{1}{\cos\phi_j^T}\,\delta^{\lambda}(\text{Diff\_Flux\_Tw}_{i,j,k}) \\ &\quad + \frac{1}{\cos\phi_j^T}\,\delta^{\phi}(\text{Diff\_Flux\_Ts}_{i,j,k}) \end{aligned} \tag{2.98}$$

$$DIFF^{z}(\varphi_{i,j,k}) = \delta^{z}(\text{Diff\_Flux\_Tb}_{i,j,k})\ . \tag{2.99}$$

Occasionally the temperature and salinity fields will take on values which create a hydrostatic instability where $(\rho_{i,j,k} > \rho_{i,j,k+1})$. Such situations require the tracers in T cells (i,j,k) and $(i,j,k+1)$ to undergo a vertical mixing procedure. This is accomplished by increasing the vertical diffusion coefficient to a very large number ($\sim 10^6 cm^2/s$) on cell bottom faces which center the instability. Doing so places a huge constraint on the time step unless vertical diffusion is solved semi-implicitly. Details of semi-implicit tracer vertical diffusion are discussed in Section 2.7.2. While not used here, other techniques explicitly mix unstable columns of water through iterative procedures. See [53, 64] for more details.

2.6.4 Baroclinic Equations

The discrete momentum equations for predicting the baroclinic velocity field as described in Section 2.5 are of the form

$$\begin{aligned} \left.\frac{\partial u}{\partial t}\right|_{i,j,k} &= [\ -ADV(u_{i,j,k}) \ +\ ADV\_MET(u_{i,j,k}) \\ &\quad +\ VISC^{\lambda\phi}(u_{i,j,k}) \ + VISC^{z}(u_{i,j,k}) \ +\ VISC\_MET(u_{i,j,k}) \\ &\quad +\ COR(v_{i,j,k}) \ -\ GRAD^{\lambda}(p_{i,j,k}) \ +\ WIND(u_{i,j,k})\ ]\ M^{U}_{i,j,k} \end{aligned} \tag{2.100}$$

and

$$\left.\frac{\partial v}{\partial t}\right|_{i,j,k} = [\, -ADV(v_{i,j,k}) \; + \; ADV\_MET(v_{i,j,k}) \tag{2.101}$$
$$+ \; VISC^{\lambda\phi}(v_{i,j,k}) \; + VISC^{z}(v_{i,j,k}) \; + \; VISC\_MET(v_{i,j,k})$$
$$+ \; COR(u_{i,j,k}) \; - \; GRAD^{\phi}(p_{i,j,k}) \; + \; WIND(v_{i,j,k}) \,]\; M^{U}_{i,j,k}$$

where each right hand side term is a U cell centered quantity. The horizontal and vertical advective velocities are positioned on the western-southern and bottom faces of cell $U_{i,j,k}$ respectively. They are defined as the average of the nearest neighbor tracer advective velocities given in equations (2.88), (2.89), and (2.90),

$$\text{Adv\_Vel\_Uw}_{i,j,k} = \overline{\text{Adv\_Vel\_Tw}_{i+1,j+1,k}}^{\lambda\phi} \tag{2.102}$$

$$\text{Adv\_Vel\_Us}_{i,j,k} = \overline{\text{Adv\_Vel\_Ts}_{i+1,j+1,k}}^{\lambda\phi} \tag{2.103}$$

$$\text{Adv\_Vel\_Ub}_{i,j,k} = \overline{\text{Adv\_Vel\_Tb}_{i+1,j+1,k}}^{\lambda\phi} \quad . \tag{2.104}$$

Defining the momentum advective velocities in this manner reduces noise in the vertical velocity field [64], and gives an approximation to the boundary condition at the ocean floor, $w = -(\vec{V} \cdot \vec{\nabla})H$.

The advective velocities are used to calculate the advective flux on each face of the momentum cell.

$$\text{Adv\_Flux\_Uw}_{i,j,k} = \text{Adv\_Vel\_Uw}_{i,j,k}\; \overline{u_{i,j,k}}^{\lambda} \tag{2.105}$$

$$\text{Adv\_Flux\_Us}_{i,j,k} = \text{Adv\_Vel\_Us}_{i,j,k}\; \overline{u_{i,j,k}}^{\phi} \tag{2.106}$$

$$\text{Adv\_Flux\_Ub}_{i,j,k} = \text{Adv\_Vel\_Ub}_{i,j,k}\; \overline{u_{i,j,k}}^{z} \tag{2.107}$$

$$\text{Adv\_Flux\_Vw}_{i,j,k} = \text{Adv\_Vel\_Uw}_{i,j,k}\ \overline{v_{i,j,k}}^{\lambda} \tag{2.108}$$

$$\text{Adv\_Flux\_Vs}_{i,j,k} = \text{Adv\_Vel\_Us}_{i,j,k}\ \overline{v_{i,j,k}}^{\phi} \tag{2.109}$$

$$\text{Adv\_Flux\_Vb}_{i,j,k} = \text{Adv\_Vel\_Ub}_{i,j,k}\ \overline{v_{i,j,k}}^{z} \tag{2.110}$$

The net advection through the cell is defined as the difference of the advective fluxes,

$$\begin{aligned} ADV(u_{i,j,k}) &= \frac{1}{\cos\phi_j^U}\ \delta^{\lambda}(\text{Adv\_Flux\_Uw}_{i,j,k}) \\ &\quad + \frac{1}{\cos\phi_j^U}\ \delta^{\phi}(\text{Adv\_Flux\_Us}_{i,j,k}) \\ &\quad + \delta^{z}(\text{Adv\_Flux\_Ub}_{i,j,k}) \end{aligned} \tag{2.111}$$

$$\begin{aligned} ADV(v_{i,j,k}) &= \frac{1}{\cos\phi_j^U}\ \delta^{\lambda}(\text{Adv\_Flux\_Vw}_{i,j,k}) \\ &\quad + \frac{1}{\cos\phi_j^U}\ \delta^{\phi}(\text{Adv\_Flux\_Vs}_{i,j,k}) \\ &\quad + \delta^{z}(\text{Adv\_Flux\_Vb}_{i,j,k})\ . \end{aligned} \tag{2.112}$$

Conservation is maintained locally except for fluxes at land-water boundaries.

The viscous fluxes are proportional to the differences of u and v across the faces of the cell,

$$\text{Visc\_Flux\_Uw}_{i,j,k} = \nu\frac{1}{\cos\phi_j^U}\ \delta^{\lambda}(u_{i-1,j,k}) \tag{2.113}$$

$$\text{Visc\_Flux\_Us}_{i,j,k} = \nu\cos\phi_j^T\ \delta^{\phi}(u_{i,j-1,k}) \tag{2.114}$$

$$\text{Visc\_Flux\_Ub}_{i,j,k} = \zeta_k\ \delta^{z}(u_{i,j,k+1}) \tag{2.115}$$

$$\text{Visc\_Flux\_Vw}_{i,j,k} = \nu\frac{1}{\cos\phi_j^U}\ \delta^{\lambda}(v_{i-1,j,k}) \tag{2.116}$$

$$\text{Visc\_Flux\_Vs}_{i,j,k} = \nu\cos\phi_j^T\ \delta^{\phi}(v_{i,j-1,k}) \tag{2.117}$$

$$\text{Visc\_Flux\_Vb}_{i,j,k} = \zeta_k\ \delta^{z}(v_{i,j,k+1})\ , \tag{2.118}$$

where ν and ζ_k are the horizontal and vertical viscosity coefficients respectively. The viscosity terms,

$$\begin{aligned} VISC^{\lambda\phi}(u_{i,j,k}) &= \frac{1}{\cos\phi_j^U}\,\delta^{\lambda}(\text{Visc\_Flux\_Uw}_{i,j,k}) \\ &\quad + \frac{1}{\cos\phi_j^U}\,\delta^{\phi}(\text{Visc\_Flux\_Us}_{i,j,k}) \end{aligned} \tag{2.119}$$

$$VISC^{z}(u_{i,j,k}) = \delta^{z}(\text{Visc\_Flux\_Ub}_{i,j,k}) \tag{2.120}$$

$$\begin{aligned} VISC^{\lambda\phi}(v_{i,j,k}) &= \frac{1}{\cos\phi_j^U}\,\delta^{\lambda}(\text{Visc\_Flux\_Vw}_{i,j,k}) \\ &\quad + \frac{1}{\cos\phi_j^U}\,\delta^{\phi}(\text{Visc\_Flux\_Vs}_{i,j,k}) \end{aligned} \tag{2.121}$$

$$VISC^{z}(v_{i,j,k}) = \delta^{z}(\text{Visc\_Flux\_Vb}_{i,j,k})\ , \tag{2.122}$$

are defined as the difference of the viscous fluxes across the cell. Conservation is again maintained locally except for fluxes at land-water boundaries.

The metric terms due to advection are

$$ADV\_MET(u_{i,j,k}) = \frac{u_{i,j,k}v_{i,j,k}\ \tan\phi_j^U}{R} \tag{2.123}$$

$$ADV\_MET(v_{i,j,k}) = -\frac{u_{i,j,k}^2\ \tan\phi_j^U}{R}\ , \tag{2.124}$$

and those due to viscosity are given as

$$\begin{aligned} VISC\_MET(u_{i,j,k}) &= \nu\left(\frac{1-\tan^2\phi_j^U}{R^2}\right)u_{i,j,k} \\ &\quad - \nu\left(\frac{\sin\phi_j^U}{R^2\cos^2\phi_j^U}\right)\left(\frac{v_{i+1,j,k}-v_{i-1,j,k}}{\Delta\lambda}\right) \end{aligned} \tag{2.125}$$

$$\begin{aligned} VISC\_MET(v_{i,j,k}) &= \nu\left(\frac{1-\tan^2\phi_j^U}{R^2}\right)v_{i,j,k} \\ &\quad + \nu\left(\frac{\sin\phi_j^U}{R^2\cos^2\phi_j^U}\right)\left(\frac{u_{i+1,j,k}-u_{i-1,j,k}}{\Delta\lambda}\right)\ . \end{aligned} \tag{2.126}$$

Coriolis effects on momentum are accounted in the terms

$$COR(v_{i,j,k}) \quad = \quad f_j \; v_{i,j,k} \tag{2.127}$$

$$COR(u_{i,j,k}) \quad = \quad -f_j \; u_{i,j,k} \; , \tag{2.128}$$

where the Coriolis number, $f_j = 2 \; \Omega \; \sin \phi_j^U$, represents the effects of the Earth spinning about its axis with rotational speed Ω. The pressure gradient terms are due to spatial variation in density accumulated vertically. The state density, $\rho_{i,j,k}$, is first defined at T cell centers using the UNESCO international standard equation of state [29]. The pressure gradients are then calculated by differencing the density averaged to U cell faces.

$$GRAD^{\lambda}(p_{i,j,k}) \; = \begin{cases} \frac{g}{\rho_0 \cos \phi_j^U} \; \delta^{\lambda}(\overline{\rho_{i,j+1,1}}^{\phi}) \; \frac{\Delta z_1^{TU}}{2} & , \quad k = 1 \\ GRAD^{\lambda}(p_{i,j,k-1}) \; + \frac{g}{\rho_0 \cos \phi_j^U} \; \delta^{\lambda}(\overline{\rho_{i,j+1,k-1}}^{\phi z}) \; \Delta z_{k-1}^{w} & , \quad k > 1 \end{cases} \tag{2.129}$$

$$GRAD^{\phi}(p_{i,j,k}) \; = \begin{cases} \frac{g}{\rho_0} \; \delta^{\phi}(\overline{\rho_{i+1,j,1}}^{\lambda}) \; \frac{\Delta z_1^{TU}}{2} & , \quad k = 1 \\ GRAD^{\phi}(p_{i,j,k-1}) \; + \frac{g}{\rho_0} \; \delta^{\phi}(\overline{\rho_{i+1,j,k-1}}^{\lambda z}) \; \Delta z_{k-1}^{w} & , \quad k > 1 \end{cases} \tag{2.130}$$

The wind stress terms are non-zero for $k = 1$ only and are defined as

$$WIND(u_{i,j,k}) \; = \begin{cases} \frac{1}{\rho_0 \Delta z_1^{TU}} \; \theta_{i,j}^{u} & , \quad k = 1 \\ 0 & , \quad k > 1 \end{cases} \tag{2.131}$$

$$WIND(v_{i,j,k}) \; = \begin{cases} \frac{1}{\rho_0 \Delta z_1^{TU}} \; \theta_{i,j}^{v} & , \quad k = 1 \\ 0 & , \quad k > 1 \end{cases} \tag{2.132}$$

where $(\theta^u_{i,j}, \theta^v_{i,j})$ are the longitude and latitude components of wind stress either read in from a forcing file or defined analytically.

2.6.5 Barotropic Equations

The discrete barotropic equations as described in Section 2.5 are of the form

$$\left.\frac{\partial U}{\partial t}\right|_{i,j} = \Big[\ COR(V_{i,j}) \ - \ GRAD^{\lambda}(\eta_{i,j}) \ + \ VISC^{\lambda\phi}(U_{i,j}) \ + \ X_{i,j}\Big]\ M^U_{i,j,1} \quad (2.133)$$

$$\left.\frac{\partial V}{\partial t}\right|_{i,j} = \Big[\ COR(U_{i,j}) \ - \ GRAD^{\phi}(\eta_{i,j}) \ + \ VISC^{\lambda\phi}(V_{i,j}) \ + \ Y_{i,j}\Big]\ M^U_{i,j,1} \quad (2.134)$$

$$\left.\frac{\partial \eta}{\partial t}\right|_{i,j} = -CONT(U_{i,j}, V_{i,j})\ M^T_{i,j,1} \quad (2.135)$$

where the right hand sides are U cell centered in the momentum equations, and T cell centered in the surface height equation. The forcing terms,

$$X_{i,j} = \sum_{k=1}^{k_{max}} [\ -ADV(u_{i,j,k}) \ + \ ADV\_MET(u_{i,j,k}) \ - \ GRAD^{\lambda}(p_{i,j,k}) + \ VISC^{z}(u_{i,j,k}) \ + \ VISC\_MET(u_{i,j,k}) \ + \ WIND(u_{i,j,k})\ ]\ \Delta z^{TU}_k \quad (2.136)$$

$$Y_{i,j} = \sum_{k=1}^{k_{max}} [\ -ADV(v_{i,j,k}) \ + \ ADV\_MET(v_{i,j,k}) \ - \ GRAD^{\phi}(p_{i,j,k}) + \ VISC^{z}(v_{i,j,k}) \ + \ VISC\_MET(v_{i,j,k}) \ + \ WIND(v_{i,j,k})\ ]\ \Delta z^{TU}_k\ , \quad (2.137)$$

are the vertical sums of the advection, pressure, and metric terms discussed in the previous section (2.6.4) on baroclinic momentum discretization. Those terms used to calculate $X_{i,j}$ are defined in equations (2.111), (2.123), (2.129), (2.120), (2.125), and (2.131), and the $Y_{i,j}$ terms are defined in equations (2.112), (2.124), (2.130), (2.122), (2.126), and (2.132).

The Coriolis terms are defined as

$$COR(V_{i,j}) = f_j\ V_{i,j} \quad (2.138)$$

$$COR(U_{i,j}) = -f_j\ U_{i,j}\ , \quad (2.139)$$

where the Coriolis number is again defined as $f_j = 2\ \Omega\ \sin\phi_j^U$. Barotropic pressure effects are attributed to changes in surface height. The associated fluxes located at U cell faces are calculated by averaging surface height located at U cell vertices.

$$\text{Eta\_Flux\_w}_{i,j} = \overline{\eta_{i,j+1}}^{\phi} \tag{2.140}$$

$$\text{Eta\_Flux\_s}_{i,j} = \overline{\eta_{i+1,j}}^{\lambda} \tag{2.141}$$

The barotropic pressure gradient through the momentum cell is the difference in surface height fluxes.

$$GRAD^{\lambda}(\eta_{i,j}) = \frac{gH_{i,j}^T}{\cos\phi_j^T}\ \delta^{\lambda}(\text{Eta\_Flux\_w}_{i,j}) \tag{2.142}$$

$$GRAD^{\phi}(\eta_{i,j}) = gH_{i,j}^T\ \delta^{\phi}(\text{Eta\_Flux\_s}_{i,j}) \tag{2.143}$$

Horizontal viscosity is directly computed from the barotropic velocities. The viscous fluxes are proportional to the differences of U and V across the faces of the cell,

$$\text{ViscBT\_Flux\_Uw}_{i,j} = \nu\frac{1}{\cos\phi_j^U}\ \delta^{\lambda}(U_{i-1,j}) \tag{2.144}$$

$$\text{ViscBT\_Flux\_Us}_{i,j} = \nu\cos\phi_j^T\ \delta^{\phi}(U_{i,j-1}) \tag{2.145}$$

$$\text{ViscBT\_Flux\_Vw}_{i,j} = \nu\frac{1}{\cos\phi_j^U}\ \delta^{\lambda}(V_{i-1,j}) \tag{2.146}$$

$$\text{ViscBT\_Flux\_Vs}_{i,j} = \nu\cos\phi_j^T\ \delta^{\phi}(V_{i,j-1}) \tag{2.147}$$

where ν is the horizontal viscosity coefficient. The viscosity terms are defined as the difference of the viscous fluxes across the cell.

$$VISC^{\lambda\phi}(U_{i,j}) = \frac{1}{\cos\phi_j^U}\ \delta^{\lambda}(\text{ViscBT\_Flux\_Uw}_{i,j}) + \frac{1}{\cos\phi_j^U}\ \delta^{\phi}(\text{ViscBT\_Flux\_Us}_{i,j}) \tag{2.148}$$

$$VISC^{\lambda\phi}(V_{i,j}) = \frac{1}{\cos\phi_j^U}\,\delta^\lambda(\text{ViscBT\_Flux\_Vw}_{i,j}) + \frac{1}{\cos\phi_j^U}\,\delta^\phi(\text{ViscBT\_Flux\_Vs}_{i,j}) \quad (2.149)$$

The fluxes for the continuity term in the surface height equation are located at T cell faces. They are due to the divergence in the velocity field and are computed by averaging the barotropic velocities located T cell vertices.

$$\text{Cont\_Flux\_w}_{i,j} = \overline{U_{i,j}}^{\phi} \quad (2.150)$$

$$\text{Cont\_Flux\_s}_{i,j} = \cos\phi_{j-1}^U \overline{V_{i,j}}^{\lambda} \quad (2.151)$$

The continuity term is the net mass flux through the faces of T cells.

$$CONT(U_{i,j}, V_{i,j}) = \frac{1}{\cos\phi_j^T}[\delta^\lambda(\text{Cont\_Flux\_w}_{i,j}) + \delta^\phi(\text{Cont\_Flux\_s}_{i,j})] \quad (2.152)$$

2.7 Temporal Discretization

A variety of methods exist for time advancing partial differential equations of the form

$$\frac{\partial \mathbf{U}}{\partial t} = F(\mathbf{U}) \ , \quad (2.153)$$

where $F(\mathbf{U})$ represents all non-time derivative operations on the system of unknowns, $\mathbf{U}$. Schemes are categorized as explicit, implicit, or semi-implicit. Explicit schemes solve for $\mathbf{U}$ at $t = \tau$ using $\mathbf{U}$ at time $t < \tau$ to discretize $F(\mathbf{U})$. Fully implicit schemes discretize $F(\mathbf{U})$ using $\mathbf{U}$ at $t = \tau$ and often require a matrix inversion as a result. Semi-implicit methods such as those used for this work discretize $F(\mathbf{U})$ using a combination of $\mathbf{U}$ at $t = \tau$ and $t < \tau$. Coriolis, vertical diffusion, and vertical viscosity terms have the option for explicit or semi-implicit time advancement. All other terms are evaluated explicitly.

2.7.1 Explicit Time Advancement

Explicit time advancement in the AMR Ocean Model is primarily performed using the leapfrog time stepping scheme,

$$\mathbf{U}^{n+1} \;=\; \mathbf{U}^{n-1} \;+\; 2\Delta t\; F(\mathbf{U}^{n})\;, \tag{2.154}$$

which approximates the time derivatives using centered differences. The method is often called a three time-level method given that the advance to $\mathbf{U}^{n+1}$ involves the solution at steps $\mathbf{U}^{n}$ and $\mathbf{U}^{n-1}$. Storage is required for two previous time steps, but second order accuracy in time is achieved after a single evaluation of $F(\mathbf{U})$. This is ideal for ocean models since numerous time steps are required to reach equilibrium. Other explicit schemes such as Runge Kutta require multiple evaluations of $F(\mathbf{U})$ and are much more expensive computationally.

An undesirable characteristic of leapfrog is the generation of a computational mode which produces oscillations in the solution. The mode may be damped using a two time-level method referred to as a "mixing" step [53]. The leapfrog scheme in the AMR Ocean Model uses Euler forward-backward,

$$\begin{aligned} \mathbf{U}^{pred} &= \mathbf{U}^{n} \;+\; \Delta t F(\mathbf{U}^{n}) \\ \mathbf{U}^{n+1} &= \mathbf{U}^{n} \;+\; \Delta t F(\mathbf{U}^{pred})\;, \end{aligned} \tag{2.155}$$

as the mixing step numerical method, where $F(\mathbf{U}^{n})$ and $F(\mathbf{U}^{pred})$ evaluate all explicit terms of the PDE system with $\mathbf{U}^{n}$ and $\mathbf{U}^{pred}$ respectively. This first order accurate predictor-corrector scheme highly damps the solution's short wave Fourier components, but only slightly damps those components corresponding to large scales [40]. The result is a reduction

of noise in the solution, although a computational expense is added from two evaluations of $F(\mathbf{U})$. Mixing steps are therefore performed as infrequently as possible. A common modeling technique is to primarily advance with leapfrog and substitute Euler forward-backward on every 17^{th} step.

2.7.2 Tracer Semi-Implicit Vertical Diffusion

The vertical diffusion term in the discrete tracer equation (2.87) is separated into explicit and implicit terms in index k,

$$DIFF^z(\varphi_k) \quad = \quad EDIFF^z(\varphi_k^m) \quad + \quad IDIFF^z(\varphi_k^{n+1}) \quad , \tag{2.156}$$

where

$$EDIFF^z(\varphi_k^m) \quad = \quad \delta^z(\text{eDiff\_Flux\_Tb}_k) \tag{2.157}$$

$$IDIFF^z(\varphi_k^{n+1}) \quad = \quad \delta^z(\text{iDiff\_Flux\_Tb}_k) \tag{2.158}$$

and

$$\text{eDiff\_Flux\_Tb}_k = \begin{cases} \frac{\Delta z_1^{TU}}{\tau_\varphi} \ \left(\varphi^{surf} \ - \ (1-idiff)\varphi_1^m\right) \ M_{i,j,1}^T & , \quad k=0 \\ \kappa_k^\varphi \ (1-idiff)\delta^z(\varphi_{k+1}^m) \ M_{k+1}^T \ M_k^T & , \quad k>0 \end{cases} \tag{2.159}$$

$$\text{iDiff\_Flux\_Tb}_k = \begin{cases} \frac{\Delta z_1^{TU}}{\tau_\varphi} \ \left(-idiff\,\varphi_1^{n+1}\right) \ M_1^T & , \quad k=0 \\ \kappa_k^\varphi \ idiff\,\delta^z(\varphi_{k+1}^{n+1}) \ M_{k+1}^T \ M_k^T & , \quad k>0 \ . \end{cases} \tag{2.160}$$

The fraction of implicit vertical diffusion is specified as *idiff*. The explicit diffusion fluxes are evaluated using tracer data φ_k^m , where m is $(n-1)$ for leapfrog steps, n for predictor

Euler forward-backward steps, or *pred* for corrector Euler forward-backward steps. The advection terms are evaluated using φ_k^n for leapfrog and predictor Euler forward-backward steps, and φ_k^{pred} for corrector Euler forward-backward steps.

The time marching procedure of a leapfrog step or Euler forward-backward substep first advances the explicit terms in (2.87) to an explicit state, $E(\varphi_k)$. The step or substep is completed by solving the implicit diffusion tracer equation,

$$\varphi_k^M - \Delta\tau \; IDIFF^z(\varphi_k^M) \; = E(\varphi_k) \; , \tag{2.161}$$

where $\Delta\tau$ is $2\Delta t$ or Δt for leapfrog or Euler forward-backward respectively. The updated state, φ_k^M, is φ_k^{pred} for predictor Euler forward-backward steps, and φ_k^{n+1} for leapfrog and corrector Euler forward-backward steps. Expanding the vertical diffusion term results in a tridiagonal system in the index k,

$$A_k\varphi_{k-1}^M \; + \; B_k\varphi_k^M \; + \; C_k\varphi_{k+1}^M \;\; = \;\; E(\varphi_k) \; , \tag{2.162}$$

where

$$A_k \;\; = \;\; D_k \left(\frac{M_{k-1}^T}{\Delta z_{k-1}^w} \right) \; (1 \; - \; \delta_{k1}) \tag{2.163}$$

$$B_k \;\; = \;\; 1 \; - \; A_k \; - \; C_k \; + \; \left(\frac{idiff\, \Delta\tau}{\tau_\varphi} \right) M_k^T \delta_{k1} \tag{2.164}$$

$$C_k \;\; = \;\; D_k \left(\frac{M_{k+1}^T}{\Delta z_k^w} \right) \tag{2.165}$$

$$D_k \;\; = \;\; -\frac{idiff \;\; \Delta\tau \;\; \kappa_k^\varphi \;\; M_k^T}{\Delta z_k^{TU}} \; . \tag{2.166}$$

The surface restoring is accounted with the Kronecker delta, δ_{k1}, and the system is solved using the tridiagonal solver found in [57].

2.7.3 Barotropic Semi-Implicit Coriolis

The Coriolis terms in the discrete barotropic equations (2.133)-(2.135), are separated into explicit and implicit components,

$$COR(U) = (1 - icor)\ ECOR(U^m) + icor\ ICOR(U^{n+1}) \tag{2.167}$$

$$COR(V) = (1 - icor)\ ECOR(V^m) + icor\ ICOR(V^{n+1}) \ , \tag{2.168}$$

where

$$ECOR(U^m) = -f\ U^m \tag{2.169}$$

$$ECOR(V^m) = f\ V^m \tag{2.170}$$

$$ICOR(U^{n+1}) = -f\ U^{n+1} \tag{2.171}$$

$$ICOR(V^{n+1}) = f\ V^{n+1} \ , \tag{2.172}$$

and *icor* is the fraction of Coriolis modeled implicitly. Euler forward-backward is solely used to advance the barotropic equations in order to reduce noise in the surface height and barotropic velocities. This technique is common in Bryan-Cox-Semtner type ocean models [40, 53, 64].

The predictor barotropic advance first evaluates the explicit terms in (2.133)-(2.135) using $[U^n, V^n, \eta^n]$ giving the predicted surface height, η^{pred}, and the explicit predicted velocities $[E(U^n),\ E(V^n)]$. The predictor step is completed by solving for $\mathbf{V}^{pred} =$

$[U^{pred}, V^{pred}]$ in the implicit Coriolis equation,

$$[\ \mathbf{I}\ -\ icor\Delta t\mathbf{C}\ ]\ \mathbf{V}^{pred} = E(\mathbf{V}^n) \tag{2.173}$$

$$E(\mathbf{V}^n) = \begin{bmatrix} E(U^n) \\ E(V^n) \end{bmatrix} \tag{2.174}$$

$$\mathbf{C} = \begin{bmatrix} 0 & f \\ -f & 0 \end{bmatrix} . \tag{2.175}$$

The process is repeated using $[U^{pred}, V^{pred}, \eta^{pred}]$ in equations (2.133)-(2.135) to give η^{n+1} and $[E(U^{pred}),\ E(V^{pred})]$. The corrector advance is completed by solving the implicit Coriolis equation ,

$$[\ \mathbf{I}\ -\ icor\Delta t\mathbf{C}\ ]\ \mathbf{V}^{n+1} = E(\mathbf{V}^{pred})\ . \tag{2.176}$$

The matrix equation in the predictor step (2.173) and the corrector step (2.176) are solved algebraically avoiding the need for a numerical matrix solver.

$$\mathbf{V}^{pred} = \frac{1}{1+(icor\Delta tf)^2}\ [\ \mathbf{I}\ +\ icor\Delta t\ \mathbf{C}\ ]\ E(\mathbf{V}^n) \tag{2.177}$$

$$\mathbf{V}^{n+1} = \frac{1}{1+(icor\Delta tf)^2}\ [\ \mathbf{I}\ +\ icor\Delta t\ \mathbf{C}\ ]\ E(\mathbf{V}^{pred}) \tag{2.178}$$

The fully explicit case is recovered when $icor = 0$. The individual expressions for U^{n+1} and V^{n+1} are obtained by expanding $\mathbf{V}$ and $\mathbf{C}$.

$$U^{pred} = \frac{1}{1+(icor\Delta tf)^2}[\ E(U^n)\ +\ (icor\Delta tf)E(V^n)\ ] \tag{2.179}$$

$$V^{pred} = \frac{1}{1+(icor\Delta tf)^2}[\ E(V^n)\ -\ (icor\Delta tf)E(U^n)\ ] \tag{2.180}$$

$$U^{n+1} = \frac{1}{1+(icor\Delta tf)^2}[\ E(U^{pred})\ +\ (icor\Delta tf)E(V^{pred})\ ] \tag{2.181}$$

$$V^{n+1} = \frac{1}{1+(icor\Delta tf)^2}[\ E(V^{pred})\ -\ (icor\Delta tf)E(U^{pred})\ ] \tag{2.182}$$

2.7.4 Baroclinic Semi-Implicit Coriolis and Vertical Viscosity

The Coriolis and vertical viscosity terms in the discrete baroclinic equations (2.100)-(2.101) are separated into semi-implicit components. The Coriolis notation follows that used in Section 2.7.3, and the vertical viscosity is

$$VISC^z(u) = (1 - ivisc)\ EVISC^z(u_k^m) + ivisc\ IVISC^z(u_k^{n+1}) \tag{2.183}$$

$$VISC^z(v) = (1 - ivisc)\ EVISC^z(v_k^m) + ivisc\ IVISC^z(v_k^{n+1}) \tag{2.184}$$

where

$$EVISC^z(u_k^m) = \delta^z(\text{eVisc\_Flux\_Ub}_k) \tag{2.185}$$

$$EVISC^z(v_k^m) = \delta^z(\text{eVisc\_Flux\_Vb}_k) \tag{2.186}$$

$$IVISC^z(u_k^{n+1}) = \delta^z(\text{iVisc\_Flux\_Ub}_k) \tag{2.187}$$

$$IVISC^z(v_k^{n+1}) = \delta^z(\text{iVisc\_Flux\_Vb}_k)\ , \tag{2.188}$$

and

$$\text{eVisc\_Flux\_Ub}_k = \zeta_k\ \delta^z(u_{k+1}^m) \tag{2.189}$$

$$\text{eVisc\_Flux\_Vb}_k = \zeta_k\ \delta^z(v_{k+1}^m) \tag{2.190}$$

$$\text{iVisc\_Flux\_Ub}_k = \zeta_k\ \delta^z(u_{k+1}^{n+1}) \tag{2.191}$$

$$\text{iVisc\_Flux\_Vb}_k = \zeta_k\ \delta^z(v_{k+1}^{n+1})\ . \tag{2.192}$$

The implicit portion of vertical viscosity is represented as *ivisc*. Leapfrog steps evaluate advection terms using (u^n, v^n), explicit viscosity terms using (u^{n-1}, v^{n-1}), and pressure gradient terms using time averaged density $(\frac{1}{4}\rho^{n-1} + \frac{1}{2}\rho^n + \frac{1}{4}\rho^{n+1})$. Euler forward-

backward steps evaluate all explicit terms with (u^n, v^n, ρ^n) in the predictor step, followed by $(u^{pred}, v^{pred}, \rho^{pred})$ in the corrector step.

The time marching procedure is similar to that used in sections 2.7.2 and 2.7.3. A leapfrog step or Euler forward-backward substep first advances the baroclinic equations (2.100)-(2.101) to an explicit state, $E(\mathbf{V}_k)$. The step or substep is completed by solving for the implicit components in the matrix equation

$$[\ \mathbf{I} - icor\Delta\tau\mathbf{C} - ivisc\Delta\tau\mathbf{D}\ ]\ \mathbf{V}_k^M = E(\mathbf{V}_k) \tag{2.193}$$

$$\mathbf{V}_k^n = \begin{bmatrix} u_k^n \\ v_k^n \end{bmatrix} \tag{2.194}$$

$$\mathbf{D} = \begin{bmatrix} IVISC^z & 0 \\ 0 & IVISC^z \end{bmatrix}, \tag{2.195}$$

where the matrix $\mathbf{C}$ is defined in equation (2.175), and $\Delta\tau$ is $2\Delta t$ or Δt for leapfrog or Euler forward-backward steps respectively. The updated state, $\mathbf{V}_k^M$, is $\mathbf{V}_k^{pred}$ for predictor Euler forward-backward steps, and $\mathbf{V}_k^*$ for leapfrog and corrector Euler forward-backward steps. The state $\mathbf{V}_k^*$ represents the velocity field plus an artificial depth mean component (see Section 2.5). The matrix, $[\ \mathbf{I} - icor\Delta\tau\mathbf{C} - ivisc\Delta\tau\mathbf{D}\ ]$, requires both velocity components to be solved simultaneously. This scenario is avoided by operator splitting [64] which simplifies the operation otherwise required to invert the matrix.

$$[\ \mathbf{I} - icor\Delta\tau\mathbf{C} - ivisc\Delta\tau\mathbf{D}\ ] = [\ \mathbf{I} - icor\Delta\tau\mathbf{C}\ ][\ \mathbf{I} - ivisc\Delta\tau\mathbf{D}\ ] + O(\Delta\tau^2) \tag{2.196}$$

Thus, the implicit Coriolis and viscous operations are applied separately while retaining

second order accuracy in time.

$$\mathbf{V}_k^M \;=\; [\,\mathbf{I} - ivisc\Delta\tau\mathbf{D}\,]^{-1}[\,\mathbf{I} - icor\Delta\tau\mathbf{C}\,]^{-1}E(\mathbf{V}_k) \tag{2.197}$$

The Coriolis operation is the same as that performed on the barotropic equations.

$$u_k^{**} \;=\; \frac{1}{1+(icor\Delta tf)^2}[\,E(u_k) \;+\; (icor\Delta\tau f)E(v_k)\,] \tag{2.198}$$

$$v_k^{**} \;=\; \frac{1}{1+(icor\Delta tf)^2}[\,E(v_k) \;-\; (icor\Delta\tau f)E(u_k)\,] \tag{2.199}$$

The implicit viscous operation results in two separate tridiagonal systems,

$$A_k u_{k-1}^M \;+\; B_k u_k^M \;+\; C_k u_{k+1}^M \;=\; u_k^{**} \tag{2.200}$$

$$A_k v_{k-1}^M \;+\; B_k v_k^M \;+\; C_k v_{k+1}^M \;=\; v_k^{**}\;, \tag{2.201}$$

where

$$A_k \;=\; D_k\left(\frac{1}{\Delta z_{k-1}^w}\right)(1 \;-\; \delta_{k1}) \tag{2.202}$$

$$B_k \;=\; 1 \;-\; A_k \;-\; C_k \tag{2.203}$$

$$C_k \;=\; D_k\left(\frac{1}{\Delta z_k^w}\right) \tag{2.204}$$

$$D_k \;=\; -\frac{ivisc\;\Delta\tau\;\zeta_k}{\Delta z_k^{TU}}\;. \tag{2.205}$$

Surface flux is set to zero using the Kroneker delta, δ_{k1}, and the tridiagonal solver in [57] is used to solve for each velocity component.

The updated velocity field, $(u^M, v^M) = (u^*, v^*)$, is the baroclinic mode plus a depth mean component (see Section 2.5). The true baroclinic velocities are obtained by

subtracting off the depth mean.

$$u'_{i,j,k} = u^*_{i,j,k} - \frac{1}{H_{i,j}} \sum_{k=1}^{k_{max}} u^*_{i,j,k} \, \Delta z_k^{TU} \tag{2.206}$$

$$v'_{i,j,k} = v^*_{i,j,k} - \frac{1}{H_{i,j}} \sum_{k=1}^{k_{max}} v^*_{i,j,k} \, \Delta z_k^{TU} \tag{2.207}$$

The true velocity field is obtained by summing the baroclinic and the barotropic velocities when the two systems are at the same simulation time.

$$u_{i,j,k} = u'_{i,j,k} + \frac{1}{H_{i,j}} U_{i,j} \tag{2.208}$$

$$v_{i,j,k} = v'_{i,j,k} + \frac{1}{H_{i,j}} V_{i,j} \tag{2.209}$$

2.7.5 Subcycling

The largest stable barotropic time step,

$$\Delta t^{\mathrm{BT}} < \frac{1}{2\sqrt{\Omega^2 + gH\left(\frac{1}{(R\cos\phi_{\mathtt{MAX}}\Delta\lambda)^2} + \frac{1}{(R\Delta\phi)^2}\right)}}, \tag{2.210}$$

is limited by the Earth's rotational velocity (Ω), the speed of gravity waves ($\sqrt{gH}$), and the horizontal grid resolution nearest the grid pole ($\cos\phi_{\mathtt{MAX}}\Delta\lambda$, $\Delta\phi$). This is typically much smaller than the maximum baroclinic and tracer time steps,

$$\Delta t^{\mathrm{nonBT}} < \min\left[\frac{1}{U_{\mathtt{MAX}}\left(\frac{1}{R\cos\phi_{\mathtt{MAX}}\Delta\lambda} + \frac{1}{R\Delta\phi}\right)}, \frac{1}{4\nu\left(\frac{1}{(R\cos\phi_{\mathtt{MAX}}\Delta\lambda)^2} + \frac{1}{(R\Delta\phi)^2}\right)}\right], \tag{2.211}$$

where $U_{\mathtt{MAX}}$ is the maximum advective velocity, and ν is the horizontal diffusion coefficient. A large computational expense occurs if tracer and baroclinic time steps are set equal to the barotropic time step.

A subcycling technique is used to allow non-barotropic time steps to be greater than the barotropic time step. The tracer equations are first advanced from step n to $n+1$. The baroclinic terms on the right hand side of equations (2.100)-(2.101) are calculated along with the barotropic forcing terms (2.136)-(2.137). The baroclinic equations are next advanced from step n to $n+1$. The forcing terms remain constant as the barotropic equations are advanced $\left(\Delta t^{\mathrm{nonBT}}/\Delta t^{\mathrm{BT}}\right)$ times, and the process is repeated.

2.7.6 Null Space Filtering

The spatial discretization of the barotropic equations will allow a checkerboard pattern in the solution of the surface height field due to a null space solution to the centered differencing [23]. Once the pattern is present in the discrete solution, it will persist and possibly cause an instability after large numbers of time steps unless it is filtered out. The checkerboard pattern is damped by applying a filter which compares the Laplacian at each grid point in the + and × directions. The Laplacian operations are written in flux form to account for zero flux at land/water boundaries.

$$\mathrm{FilterFlux\_w}_{i,j} = (\eta_{i,j} - \eta_{i-1,j})\ M^T_{i-1,j,1} M^T_{i,j,1} \tag{2.212}$$

$$\mathrm{FilterFlux\_s}_{i,j} = (\eta_{i,j} - \eta_{i,j-1})\ M^T_{i,j-1,1} M^T_{i,j,1} \tag{2.213}$$

$$\mathrm{FilterFlux\_sw}_{i,j} = (\eta_{i,j} - \eta_{i-1,j-1})\ M^U_{i-1,j-1,1} \tag{2.214}$$

$$\mathrm{FilterFlux\_nw}_{i,j} = (\eta_{i-1,j+1} - \eta_{i,j})\ M^U_{i-1,j,1} \tag{2.215}$$

The Laplacians are given by

$$\begin{aligned} \nabla^2_+ \eta_{i,j} = [\ &\mathrm{FilterFlux\_w}_{i+1,j} - \mathrm{FilterFlux\_w}_{i,j} + \\ &\mathrm{FilterFlux\_s}_{i,j+1} - \mathrm{FilterFlux\_s}_{i,j}\ ] \end{aligned} \tag{2.216}$$

and

$$\nabla^2_{\times}\eta_{i,j} = \frac{1}{2}[\ \text{FilterFlux\_sw}_{i+1,j+1} - \text{FilterFlux\_sw}_{i,j} + \text{FilterFlux\_ns}_{i,j} - \text{FilterFlux\_ns}_{i+1,j-1}\ ]\ . \tag{2.217}$$

The filter is constructed from the difference in the two operations.

$$\eta_{i,j} = \eta_{i,j} + \lambda_{CB}\left(\nabla^2_{+}\eta_{i,j} - \nabla^2_{\times}\eta_{i,j}\right) \tag{2.218}$$

The checkerboard pattern is fully damped when $\lambda_{CB} = 0.125$. See [23, 78] for more details. This filter can be applied as frequently as after each barotropic time step. Less frequent applications of the filter will contribute less computational expense, but will also allow more opportunity for a null space solution to grow.

Chapter 3

Adaptive Mesh Refinement Implementation

This chapter describes the extensions made to the traditional AMR method [9, 8, 7] that are essential for its application to the numerical ocean model described in Chapter 2. Attention is specifically given to curvilinear coordinates, time stepping, staggering of variables, and regridding. Code design primarily consists of modifications to LLNL's AMR software library known as SAMRAI (Structured Adaptive Mesh Refinement Application Infrastructure) [38, 39]. This object oriented library can manage most AMR procedures (e.g., refinement, coarsening, dynamic regridding, time stepping, flux matching, load balancing, etc.) allowing users to implement standard AMR models by providing the numerical method and refinement criteria. Several characteristics of the AMR Ocean Model are unfortunately beyond the scope of SAMRAI. Modifications were made to components of SAMRAI while leaving the library's basic object oriented design intact. Data communi-

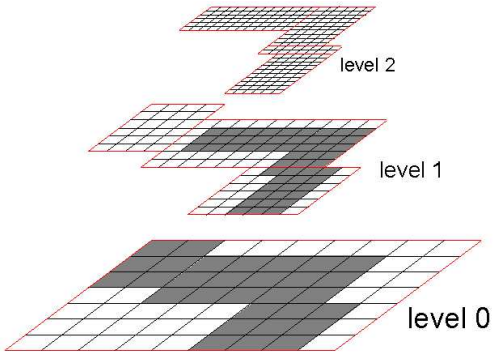


Figure 3.1: An AMR hierarchy consisting of a level-(0) and two levels of refinement. Gray areas represent cells refined by a factor of 2 in the x and y directions. Refinement levels are subdivided into refinement patches outlined in red.

cation and memory management within SAMRAI were not altered. The AMR concepts discussed in this chapter may also be added to software libraries other than SAMRAI.

3.1 The AMR Hierarchy

The basic idea of the AMR method is to refine localized areas in need of higher resolution instead of refining the entire domain. Data are maintained on a hierarchy of refinement levels with the root level defined as level-(0) (see Figure 3.1). The refinement ratios for level-(ℓ),

$$(r_\ell^\lambda, r_\ell^\phi) = (\frac{\Delta\lambda_\ell}{\Delta\lambda_{\ell+1}}, \frac{\Delta\phi_\ell}{\Delta\phi_{\ell+1}}) \ , \tag{3.1}$$

are defined here in longitude-latitude coordinates as the ratios of angular grid cell width between levels ℓ and $\ell+1$. Each level is divided into rectangular patches which may change size and position as the solution progresses in time. Communication between refinement levels occurs through refine and coarsen operations constructed as direct inverses of one

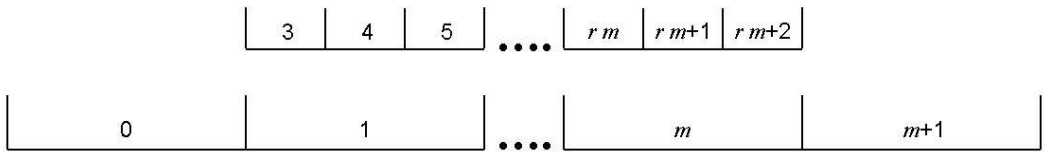


Figure 3.2: One dimensional index relationship between cell centered data on levels ℓ (bottom) and $\ell + 1$ (top) with a refinement ratio of $r_\ell = 3$.

another. The refine-coarsen inverse property is expressed in one dimension as

$$\textit{Refine}(\Phi_0) \quad \longrightarrow \quad \varphi_0, \varphi_1, ..., \varphi_{r-1} \tag{3.2}$$

$$\Phi_0 \quad \longleftarrow \quad \textit{Coarsen}(\varphi_0, \varphi_1, ..., \varphi_{r-1}) \ , \tag{3.3}$$

where $\varphi_0, \varphi_1, ..., \varphi_{r-1}$ are cell centered data on level-(ℓ+1) over-lying Φ_0 located on level-(ℓ).

Data on each refinement level resides in a different index space. Figure 3.2 shows the index relationship for cell centered data on levels ℓ and $\ell + 1$ in one dimension. Data at index m on level-(ℓ) are covered by level-($\ell + 1$) data with indices in the range $i = r_\ell\, m$ .... ($r_\ell\, m$+r_ℓ−1). The same relation is independently true for indices in two dimensions. The data on level-(ℓ) at index, $[m,\ n]$, is covered by level-($\ell + 1$) data with indices in the range $[i,\ j] = [r_\ell^\lambda m,\ r_\ell^\phi n]$ .... $[r_\ell^\lambda(m+1) - 1,\ r_\ell^\phi(n+1) - 1]$.

3.1.1 Refinement and Coarsening in Cartesian Coordinates

The refine and coarsen operations are discussed first in one dimensional Cartesian coordinates, and then expanded to two dimensions. Refined data, u_i, on level-($\ell + 1$) are defined using linear interpolation

$$\varphi_i \quad = \quad \Phi_m \ + \ \Phi^x \left[\ (i + \frac{1}{2})\Delta x_{\ell+1} - \frac{r_\ell^x}{2}\Delta x_{\ell+1} \ \right] \ , \tag{3.4}$$

where the refinement ratio is $r_\ell^x = \frac{\Delta x_\ell}{\Delta x_{\ell+1}}$, and Φ^x is an approximation to $\partial \Phi_m / \partial x$. Cell width weighted averaging,

$$\Phi_m \;=\; \frac{1}{\Delta x_\ell} \sum_{i=r_\ell^x m}^{r_\ell^x (m+1)-1} \varphi_i \; \Delta x_{\ell+1} \;, \tag{3.5}$$

is used to define coarsened data, Φ_m. The refine-coarsen inverse property (3.2)-(3.3) is easily demonstrated by substituting (3.4) into (3.5) and applying the summation equality $\sum_{i=a}^{b}(i) \;=\; \frac{a+b}{2}(b-a+1)$. Any definition of Φ^x will obey the refine-coarsen inverse property, although a slope limiter [72] is often used to preserve monotonicity when using upwind numerical methods. The above refine and coarsen operations are easily extended to two dimensions with bi-linear refinement,

$$\varphi_{i,j} \;=\; \Phi_{m,n} \;+\; \Phi^x \; \Delta x_{\ell+1} \left[\; (i+\frac{1}{2}) - \frac{r_\ell^x}{2} \; \right] \;+\; \Phi^y \; \Delta y_{\ell+1} \left[\; (j+\frac{1}{2}) - \frac{r_\ell^y}{2} \; \right] \;, \tag{3.6}$$

and cell area weighted averaging,

$$\Phi_{m,n} \;=\; \frac{1}{\Delta x_\ell \; \Delta y_\ell} \sum_{i=r_\ell^x m}^{r_\ell^x (m+1)-1} \; \sum_{j=r_\ell^y n}^{r_\ell^y (n+1)-1} \varphi_{i,j} \; \Delta x_{\ell+1} \Delta y_{\ell+1} \;. \tag{3.7}$$

3.1.2 Refinement and Coarsening in Longitude-Latitude Coordinates

The AMR Ocean Model refines the longitude and latitude directions where refinement ratios are defined by (3.1). The refine and coarsen operations are obtained by translating the two dimensional Cartesian operations (3.6)-(3.7) to longitude-latitude coordinates. The physical distances between grid cell centers are $(R \cos\phi \; \Delta\lambda, \; R\Delta\phi)$. Bi-linear refinement is expressed in longitude-latitude coordinates as

$$\varphi_{i,j} \;=\; \Phi_{m,n} \;+\; \Phi^\lambda \; (R \cos\phi_j \; \Delta\lambda_{\ell+1}) \left[\; (i+\frac{1}{2}) - \frac{r_\ell^\lambda}{2} \; \right] \;+\; \Phi^\phi \; (R \; \Delta\phi_{\ell+1}) \left[\; (j+\frac{1}{2}) - \frac{r_\ell^\phi}{2} \; \right] \;, \tag{3.8}$$

where ϕ_j is the latitudinal coordinate of $\varphi_{i,j}$. The land mask on level-(ℓ), $M_{m,n}$, dictates if Φ^λ and Φ^ϕ are defined with centered or one sided differences.

$$\Phi^\lambda = \frac{1}{\cos\phi_n} \begin{cases} \delta^\lambda(\overline{\Phi_{m,n}}^\lambda) & , \quad M_{m+1,n}\ M_{m-1,n} > 0 \\ \delta^\lambda(\Phi_{m,n}) & , \quad M_{m+1,n}\ (1 - M_{m-1,n}) > 0 \\ \delta^\lambda(\Phi_{m-1,n}) & , \quad (1 - M_{m+1,n})\ M_{m-1,n} > 0 \end{cases} \tag{3.9}$$

$$\Phi^\phi \quad = \quad \begin{cases} \delta^\phi(\overline{\Phi_{m,n}}^\phi) & , \quad M_{m,n+1}\ M_{m,n-1} > 0 \\ \delta^\phi(\Phi_{m,n}) & , \quad M_{m,n+1}\ (1 - M_{m,n-1}) > 0 \\ \delta^\phi(\Phi_{m,n-1}) & , \quad (1 - M_{m,n+1})\ M_{m,n-1} > 0 \end{cases} \tag{3.10}$$

Coarsening in longitude-latitude coordinates is performed by replacing the Cartesian cell area, $\Delta x \Delta y$, in equation (3.7) with the surface area of a longitude-latitude grid cell, $2R^2\ \Delta\lambda \cos\phi\ \sin(\Delta\phi/2)$. The land mask is again used to denote which cells are land or water.

$$\Phi_{m,n} \quad = \quad \left(\frac{M_{m,n}}{\cos\phi_n \sin(\Delta\phi_\ell/2)\ \Delta\lambda_\ell}\right) \times \sum_{i=r_\ell^\lambda m}^{r_\ell^\lambda(m+1)-1} \sum_{j=r_\ell^\phi n}^{r_\ell^\phi(n+1)-1} (\varphi_{i,j}\ M_{i,j}\ \cos\phi_j\ \sin(\Delta\phi_{\ell+1}/2)\ \Delta\lambda_{\ell+1})\ . \tag{3.11}$$

The above definitions for refinement and coarsening assume that the land mask is the same on all refinement levels. This condition is enforced for conservation purposes discussed later in Section 3.3.1.

3.2 Time Advancement

The following sections describe the time advancement procedures used by the AMR Ocean Model in detail. The traditional AMR two time-level integration scheme is described first, followed by the leapfrog AMR time integration scheme. Both procedures are used in the AMR Ocean Model. Similar descriptions of the time stepping algorithms are also found in [34]. Extra attention is given here to the flux matching procedure in order to reflect its implementation in the AMR Ocean Model.

3.2.1 Traditional Time Refinement

AMR adds complexity to time advancing PDEs given that refinement level-(ℓ) will generally permit a larger time step, Δt_ℓ, than levels finer than level-(ℓ). The situation occurs because stability conditions of explicit numerics require the time step to vary inversely with the grid spacing. This is exploited in a "time refinement" procedure, where each level's Δt_ℓ is set equal to an integer multiple of $\Delta t_{\ell+1}$ such that Δt_ℓ still satisfies the stability condition for level-(ℓ). Levels are advanced recursively starting with level-(0). An interface flux matching procedure preserves conservation where levels ℓ and $(\ell+1)$ abut.

The numerical methods typically used in AMR are explicit two time-level methods of the form

$$\mathbf{U}^{n+1} \;=\; \mathbf{U}^{n} \;+\; \Delta t\; F(\mathbf{U}), \tag{3.12}$$

where Δt is determined by the stability condition of the chosen numerical method. The time refinement ratio between two levels of refinement is defined as the ratio of the level-(ℓ)

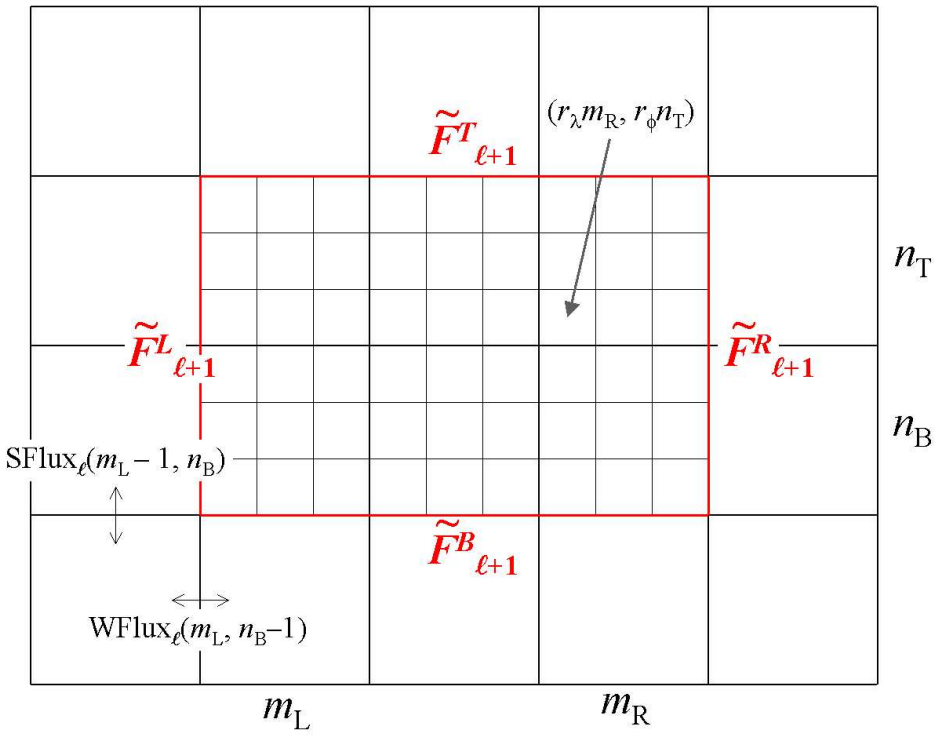


Figure 3.3: Level-ℓ cells with indices in the range $(m_L, n_B)...(m_R, n_T)$ are covered by a level-$(\ell+1)$ patch with refinement ratios $(r_\lambda, r_\phi) = (3, 3)$. For conservation, level-(ℓ) fluxes at the interface (shown in red) must equal the spatial and temporal average of level-$(\ell+1)$ interface fluxes. The flux sum variables $\tilde{F}^L_{\ell+1}$, $\tilde{F}^R_{\ell+1}$, $\tilde{F}^B_{\ell+1}$, and $\tilde{F}^T_{\ell+1}$, store time averaged level-$(\ell+1)$ fluxes at the interfaces.

and level-$(\ell+1)$ time steps,

$$r^t_\ell \;=\; \frac{\Delta t_\ell}{\Delta t_{\ell+1}} \;>\; 0\ . \tag{3.13}$$

Refinement level-(ℓ) steps forward in time to $t_\ell = \tau + \Delta t_\ell$ using equation (3.12), and level-$(\ell+1)$ advances r^t_ℓ time steps until its simulation time is equal to $t_{\ell+1} = \tau + \Delta t_\ell$. Ghost regions on level-$(\ell+1)$ patches are filled prior to each level-$(\ell+1)$ advance using either spatial interpolation when $t_{\ell+1} = \tau$, or space and time interpolation when level-(ℓ) and level-$(\ell+1)$ are at different simulation times.

Interface flux matching ensures the fluxes used by level-(ℓ) and level-($\ell+1$) agree at the interface, thereby preserving conservation [8]. Temporary flux sum variables, $\tilde{F}^{\mathrm{L}}_{\ell}$, $\tilde{F}^{\mathrm{R}}_{\ell}$, $\tilde{F}^{\mathrm{B}}_{\ell}$ and $\tilde{F}^{\mathrm{T}}_{\ell}$, are defined on refinement patch edges for levels finer than level-(0) as shown in Figure 3.3. They are initialized to zero before the first level-($\ell+1$) advance, and summed with the patch edge fluxes after each level-($\ell+1$) advance. A level-($\ell+1$) patch with indices in the range $(i_{\mathrm{L}}, j_{\mathrm{B}})...(i_{\mathrm{R}}, j_{\mathrm{T}})$ has flux sums,

$$\tilde{F}^{\mathrm{L}}_{\ell+1}(j) \quad \mathtt{+=} \quad \mathrm{WFlux}_{\ell}(i_{\mathrm{L}},\ j)\ /\ r^{t}_{\ell} \tag{3.14}$$

$$\tilde{F}^{\mathrm{R}}_{\ell+1}(j) \quad \mathtt{+=} \quad \mathrm{WFlux}_{\ell}(i_{\mathrm{R}}+1,\ j)\ /\ r^{t}_{\ell} \tag{3.15}$$

$$\tilde{F}^{\mathrm{B}}_{\ell+1}(i) \quad \mathtt{+=} \quad \mathrm{SFlux}_{\ell}(i,\ j_{\mathrm{B}})\ /\ r^{t}_{\ell} \tag{3.16}$$

$$\tilde{F}^{\mathrm{T}}_{\ell+1}(i) \quad \mathtt{+=} \quad \mathrm{SFlux}_{\ell}(i,\ j_{\mathrm{T}}+1)\ /\ r^{t}_{\ell}\ , \tag{3.17}$$

equal to the time averaged level-($\ell+1$) flux through their respective interfaces after r^{t}_{ℓ} steps.

A synchronization process occurs when level-(ℓ) and level-($\ell+1$) are at the same simulation time. Fluxes on level-(ℓ) that lie on an interface with level-($\ell+1$) are replaced by the spatial averages of the interface flux sums,

$$\mathrm{WFlux}_{\ell}(m_{\mathrm{L}}, n) \quad = \quad \sum_{j=r^{\phi}_{\ell} n}^{r^{\phi}_{\ell}(n+1)-1} \tilde{F}^{\mathrm{L}}_{\ell+1}(j)\ /\ r^{\phi}_{\ell}, \qquad (n_{\mathrm{B}} \le n \le n_{\mathrm{T}}) \tag{3.18}$$

$$\mathrm{WFlux}_{\ell}(m_{\mathrm{R}}+1, n) \quad = \quad \sum_{j=r^{\phi}_{\ell} n}^{r^{\phi}_{\ell}(n+1)-1} \tilde{F}^{\mathrm{R}}_{\ell+1}(j)\ /\ r^{\phi}_{\ell}, \qquad (n_{\mathrm{B}} \le n \le n_{\mathrm{T}}) \tag{3.19}$$

$$\mathrm{SFlux}_{\ell}(m, n_{\mathrm{B}}) \quad = \quad \sum_{i=r^{\lambda}_{\ell} m}^{r^{\lambda}_{\ell}(m+1)-1} \tilde{F}^{\mathrm{B}}_{\ell+1}(i)\ /\ r^{\lambda}_{\ell}, \qquad (m_{\mathrm{L}} \le m \le m_{\mathrm{R}}) \tag{3.20}$$

$$\mathrm{SFlux}_{\ell}(m, n_{\mathrm{T}}+1) \quad = \quad \sum_{i=r^{\lambda}_{\ell} m}^{r^{\lambda}_{\ell}(m+1)-1} \tilde{F}^{\mathrm{T}}_{\ell+1}(i)\ /\ r^{\lambda}_{\ell}, \qquad (m_{\mathrm{L}} \le m \le m_{\mathrm{R}})\ . \tag{3.21}$$

Cells on level-(ℓ) abutting each $\tilde{F}_{\ell+1}$ are re-advanced from τ to $\tau + \Delta t_{\ell}$ using equation

(3.12). Cells on level-(ℓ) overlaid by level-($\ell + 1$) cells are replaced by averaged data from the fine level using the coarsen operation described in Section 3.1. The traditional AMR algorithm is summarized in Algorithm 3.2.1, and time refinement is depicted in Figure 3.4a. See [8] for more details.

Algorithm 3.2.1. Traditional Time Advance on Level-(ℓ)
  Fill ghosts
  Step using two time-level numerical method
  if ($\ell > 0$)
    $\tilde{F}_\ell$ += [patch edge fluxes] / $r^t_{\ell-1}$
  end

  if ($\ell + 1$ exists)
    $\tilde{F}_{\ell+1} = 0.0$

    for ($sub\_step = 1 \rightarrow r^t_\ell$)
      Traditional Time Advance on Level-($\ell + 1$)
    end

    Re-advance data touching $\ell + 1$ interface using spatially averaged $\tilde{F}_{\ell+1}$
    Coarsen $\ell + 1$ data onto ℓ
  end

Algorithm 3.2.1 is traditionally used with Euler forward time integration to define equation (3.12). A similar time refinement procedure is also used for sub-stepping time integration schemes such as Euler forward-backward or Runge-Kutta. The only modification involves the addition of larger ghost regions. These regions are filled prior to a level's advance as usual using interpolation. Prediction steps advance the solution on patch interiors and ghosts. The correction step records interface fluxes and advances the interior data only. The correction step is also used to re-advance interface data during the synchronization step. These modifications keep errors from propagating from patch boundaries to the patch interior. The recursive time refinement algorithm for sub-stepping time integration is otherwise identical to Algorithm 3.2.1.

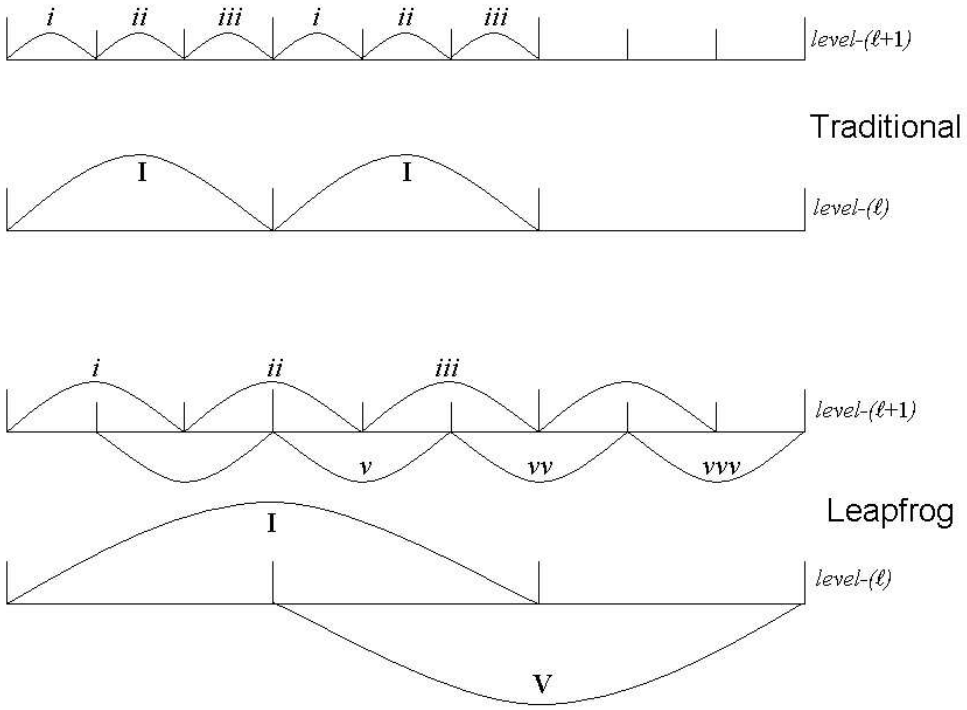


Figure 3.4: Flux matching in the presence of time refinement for traditional AMR (top), and leapfrog AMR (bottom). (Top) Time steps do not overlap and level-(ℓ) step **I** is always matched by level-($\ell + 1$) steps i, ii, and iii. (Bottom) Leapfrog time steps overlap. As a result, level-(ℓ) step **I** is matched by i, ii, and iii, but the following level-(ℓ) step **V** is matched by v, vv, and vvv.

3.2.2 Leapfrog Time Refinement

Leapfrog time advancement uses centered differences in time to approximate first order time derivatives in partial differential equations, and has the form

$$\mathbf{U}^{n+1} \;=\; \mathbf{U}^{n-1} \;+\; 2\Delta t\; F(\mathbf{U}^{n})\;. \tag{3.22}$$

Figure 3.4 illustrates how leapfrog in the context of AMR creates the need for two sets of flux sum variables. Leapfrog uses level-$(\ell+1)$ steps *i*, *ii*, and *iii* to match level-(ℓ) step **I**, and level-$(\ell+1)$ steps *v*, *vv*, and *vvv* to match level-(ℓ) step **V**. This scenario does not occur in traditional AMR because time steps do not overlap, and consequently each level-(ℓ) step **I** is matched using level-$(\ell+1)$ steps *i*, *ii*, and *iii*.

Flux matching is accomplished logistically in leapfrog AMR using primary and secondary sets of flux sum variables ($\tilde{F}_\ell$ and $\tilde{f}_\ell$) as shown in Figure 3.5 for odd and even time step ratios. The flux sums are initialized prior to the first level-$(\ell+1)$ advance as $\tilde{F}_{\ell+1} = \tilde{f}_{\ell+1}$ and $\tilde{f}_{\ell+1} = 0$. In the case of odd time refinement ratios, interface fluxes are added into the primary sum after level-$(\ell+1)$ makes an odd numbered step, and into the secondary sum after an even numbered step. The summing procedure for even time step ratios sums both the primary and the secondary on even numbered steps only. Level-(ℓ) is re-advanced at the interfaces using spatially averaged $\tilde{F}_{\ell+1}$ when the two levels are both at $t = \tau + \Delta t_\ell$.

Leapfrog requires an occasional two time-level mixing step as discussed in Section 2.7.1. This necessitates a special case of flux summing shown in Figure 3.6 for odd and even time step ratios. A level-(ℓ) mixing step is flux matched by initializing $\tilde{F}_{\ell+1}$ and $\tilde{f}_{\ell+1}$ to zero and starting level-$(\ell+1)$ with a mixing step followed by leapfrog steps. Flux sums are

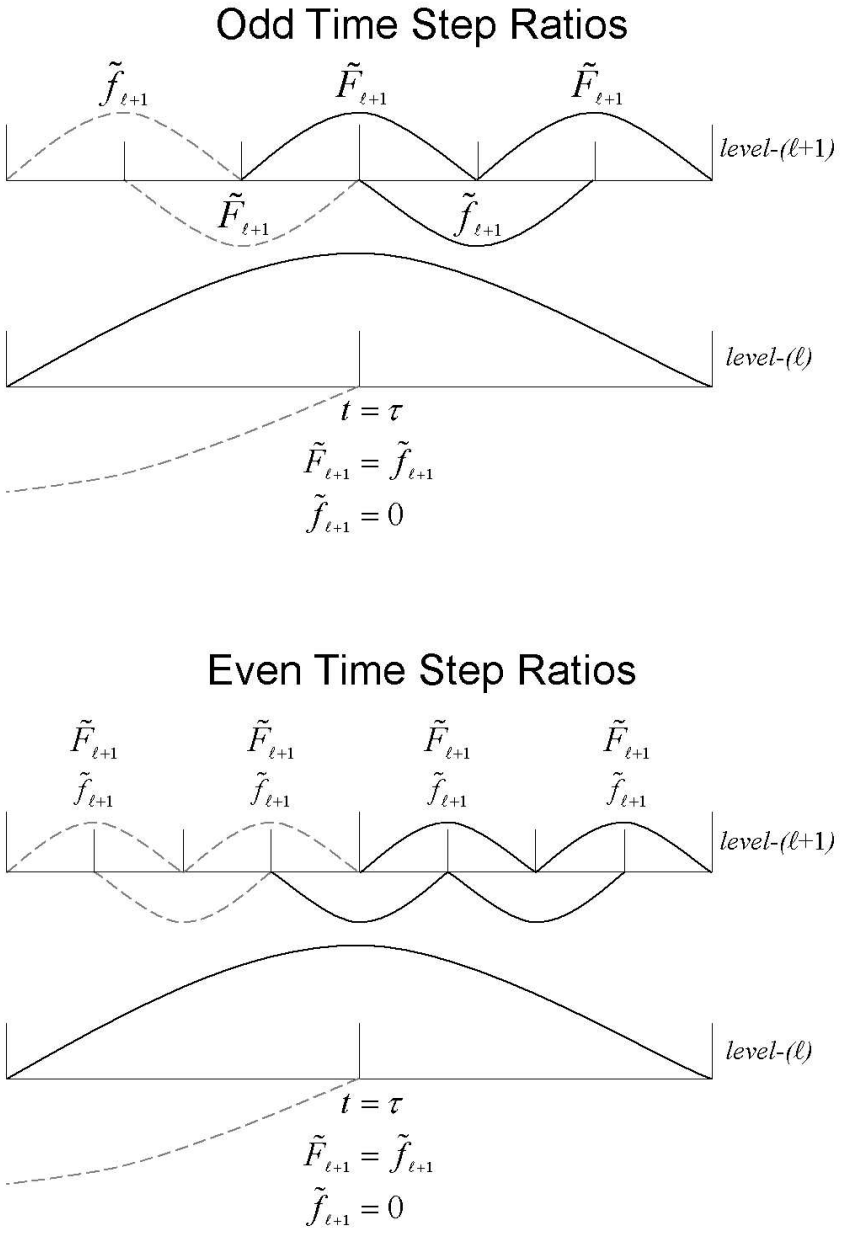


Figure 3.5: Leapfrog interface flux summing procedures for odd and even time step ratios at $t = \tau$. After a level-(ℓ) leapfrog step, $\tilde{F}_{\ell+1}$ is initialized as $\tilde{f}_{\ell+1}$ from the previous level-($\ell+1$) step, and $\tilde{f}_{\ell+1}$ is set to zero. (Top) Interface fluxes are summed in $\tilde{F}_{\ell+1}$ on odd steps and $\tilde{f}_{\ell+1}$ on even steps. (Bottom) Interface fluxes are summed in $\tilde{F}_{\ell+1}$ and $\tilde{f}_{\ell+1}$ on even steps.

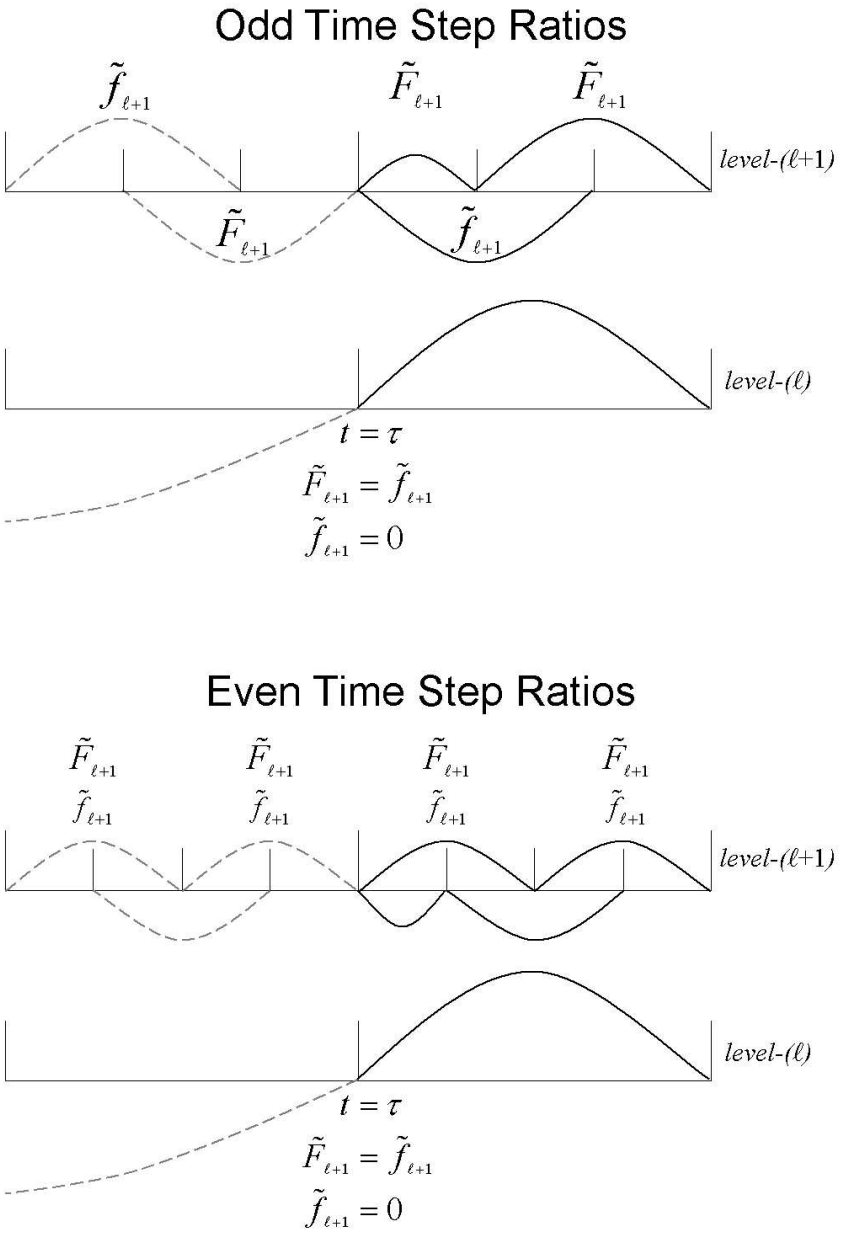


Figure 3.6: Mixing step interface flux summing procedures for odd and even time step ratios at $t = \tau$. After a level-(ℓ) mixing step, $\tilde{F}_{\ell+1}$ and $\tilde{f}_{\ell+1}$ are both initialized to zero, and level-($\ell + 1$) advances a mixing step followed by leapfrog steps. (Top) Interface fluxes are summed in $\tilde{F}_{\ell+1}$ on odd steps and $\tilde{f}_{\ell+1}$ on even steps. (Bottom) Interface fluxes are summed in $\tilde{F}_{\ell+1}$ and $\tilde{f}_{\ell+1}$ on even steps. (Top and Bottom) Sums after level-($\ell+1$) leapfrog steps are given twice the weight of those after mixing steps.

again alternated in primary and secondary for odd time refinement ratios, and summed only after even numbered steps for even time refinement ratios. Since leapfrog and mixing steps advance the solution by different time increments ($2\Delta t$ and Δt respectively), flux sums from leapfrog steps are given twice the weight as mixing steps. Thus, a parameter σ, equal to 1 for mixing steps or 2 for leapfrog steps, is multiplied by each right hand side in equations (3.14)-(3.17). The time averaged interface flux between level-(ℓ) and level-($\ell + 1$) is then $\tilde{F}_{\ell+1}/\sum\sigma$ after level-($\ell + 1$) makes r_ℓ^t steps. The leapfrog time advancement algorithm is given in Algorithm 3.2.2. The AMR Ocean Model is capable of integrating odd, even, or a mixture of odd and even time refinement ratios in the AMR hierarchy.

Algorithm 3.2.2. Leapfrog Time Advance on Level-(ℓ)

Fill ghosts
if (mixing_step$_\ell$)
  Step using Euler Forward-Backward
  mixing_step$_{\ell+1}$ = TRUE
  $\sigma = 1$
else
  Step using leapfrog
  $\sigma = 2$
end

if ($\ell > 0$)
  if [(step$_\ell$ % 2) == $\left(r^t_{\ell-1}\ \%\ 2\right)$]
    $\tilde{F}_\ell$ += $\sigma*$ [patch edge fluxes] / $r^t_{\ell-1}$
    W_ℓ += σ
  end
  if (step$_\ell$ is even)
    $\tilde{f}_\ell$ += $\sigma*$ [patch edge fluxes] / $r^t_{\ell-1}$
    w_ℓ += σ
  end
end

if ($\ell + 1$ exists)
  if (mixing_step$_l$)
    $\tilde{F}_{\ell+1} = 0.0,\ \ W_{\ell+1} = 0.0$
  else
    $\tilde{F}_{\ell+1} = \tilde{f}_{\ell+1},\ \ W_{\ell+1} = w_{\ell+1}$
  end
  $\tilde{f}_{\ell+1} = 0.0,\ \ w_{\ell+1} = 0.0$

  for ($sub\_step = 1 \rightarrow r^t_\ell$)
    Leapfrog Time Advance on Level-($\ell + 1$)
    mixing_step$_{\ell+1}$ = FALSE
  end

  Re-advance data touching $\ell + 1$ interface using spatially averaged $\left(\tilde{F}_{\ell+1}/W_{\ell+1}\right)$
  Coarsen $\ell + 1$ data onto ℓ
end

step$_\ell$ = step$_\ell$ + 1

3.2.3 Barotropic Subcycling

The barotropic subcycling technique discussed in Section 2.7.5 is implemented with time refinement in the AMR Ocean Model. The barotropic system advances with time step $\Delta t_\ell^{\mathrm{BT}}$ using Euler forward-backward and the traditional time refinement algorithm. This is simultaneously coupled to a leapfrog time integration algorithm for advancing the baroclinic and tracer systems with time step $\Delta t_\ell^{\mathrm{BC}}$. The barotropic time refinement ratio varies as $(\Delta\lambda^{-1}, \Delta\phi^{-1})$, while the time step for the baroclinic and tracer systems varies as $(\Delta\lambda^{-2}, \Delta\phi^{-2})$. The algorithm discussed here for subcycling with time refinement allows different sized time steps and time step ratios for the barotropic and non-barotropic systems. The number of subcycles on level-(ℓ),

$$\mathrm{subcycles}_\ell \;=\; \Delta t_\ell^{\mathrm{BC}} \;/\; \Delta t_\ell^{\mathrm{BT}} \;, \tag{3.23}$$

may also differ on each refinement level.

The barotropic and baroclinic time steps are first calculated on the finest refinement level. Time steps on the remaining levels are calculated in order of decreasing ℓ and adjusted if necessary such that integer values are obtained for the time refinement ratios,

$$r_\ell^{\mathrm{BT}} \;=\; \Delta t_\ell^{\mathrm{BT}} / \Delta t_{\ell+1}^{\mathrm{BT}} \tag{3.24}$$

$$r_\ell^{\mathrm{BC}} \;=\; \Delta t_\ell^{\mathrm{BC}} / \Delta t_{\ell+1}^{\mathrm{BC}} \;. \tag{3.25}$$

Each level's subcycle number must also be an integer value. Equations (3.23)-(3.25) are combined to give the subcycle condition,

$$\frac{r_\ell^{\mathrm{BT}}}{r_\ell^{\mathrm{BC}}} \;=\; \frac{\mathrm{subcycles}_{\ell+1}}{\mathrm{subcycles}_\ell} \;, \tag{3.26}$$

which must be true for each pair of refinement levels. Two procedures are available in Appendix A for adjusting time steps further such that equation (3.26) is satisfied. Both should be used with caution given that adjusted time steps which satisfy (3.26) are not guaranteed to meet stability requirements.

Time stepping may proceed after $\Delta t_\ell^{\mathrm{BT}}$ and $\Delta t_\ell^{\mathrm{BC}}$ are defined on each refinement level. The number of barotropic steps on each level, step_ℓ, is used to determine when non-barotropic components are called. Specifically, a level advances the non-barotropic systems when step_ℓ is a multiple of $\mathrm{subcycles}_\ell$. The recursive time advancing algorithm is evaluated $\mathrm{subcycles}_0$ times on level-(0) in order to complete full baroclinic and tracer steps on level-(0). The AMR subcycling algorithm is given in Algorithm 3.2.3. See Appendix A for greater detail regarding the traditional and leapfrog AMR time integration components used in the barotropic and baroclinic systems respectively.

Algorithm 3.2.3. Subcycle Advance on Level-(ℓ)

Adv_Non_BT = (step_ℓ % subcycles_ℓ == 0)

if (Adv_Non_BT)
  Fill Non-Barotropic ghosts
  Non-Barotropic Step
end

Fill Barotropic ghosts
Barotropic Step

if ($\ell > 0$)
  Sum Barotropic Fluxes
  if (Adv_Non_BT)
    Sum Non-Barotropic Fluxes
  end
end

if ($\ell + 1$ exists)
  Initialize Barotropic Flux Sums
  if (Adv_Non_BT)
    Initialize Non-Barotropic Flux Sums
  end

  for ($sub\_step = 1 \rightarrow r_\ell^{\text{BT}}$)
    Subcycle Advance on Level-($\ell + 1$)
  end

  Re-advance Barotropic data touching $\ell+1$ interface using spatially averaged Barotropic Flux Sums
  Coarsen $\ell + 1$ Barotropic data onto ℓ

  if [level-$(\ell + 1)$ has advanced $\left(r_\ell^{\text{BC}}\ \text{subcycles}_{\ell+1}\right)$ times]
    Re-advance Non-Barotropic data touching $\ell + 1$ interface using spatially averaged Non-Barotropic Flux Sums
    Coarsen $\ell + 1$ non-Barotropic data onto ℓ
  end
end

$\text{step}_\ell = \text{step}_\ell + 1$

3.3 Staggered Grid Approach

Staggered grids are often used in ocean models to reduce computational noise and better represent wave speeds. The AMR Ocean Model specifically uses a B-grid, where the velocity variables (U) are defined at the nodes of tracer variables (T). This introduces multiple control volumes which do not at first seem to translate well to an AMR environment. The majority of work in AMR software libraries like SAMRAI occurs in data communication on a single control volume between multiple refinement levels. Data is permitted to exist on the nodes and faces of a control volume, but only cell centered data are able to match fluxes at refinement level interfaces. The staggered grid is achieved in the AMR Ocean Model by defining T and U in two separate coordinated hierarchies. For each patch in the T hierarchy, a complementary patch is created in the U hierarchy and the two are arranged in space such that the cell centers of U patches lie on the nodes of T patches. Both variables are defined at cell centers in their respective patches to enable flux matching. Each hierarchy independently applies the refine and coarsen operations discussed in Section 3.1.2 to communicate their data between refinement levels.

A few issues must be noted before applying the dual hierarchy approach. Refinement interfaces for the T grid and for the U grid are spatially separated with B-Grid AMR (see Figure 3.7). This presents two possible orientations for a B-Grid refinement patch. Either U patches are interior to T patches or vice versa. Both arrangements may be used in general for B-Grid AMR, but only the former is implemented in the AMR Ocean Model. This decision was made in the early stages of research when the opposite arrangement often became unstable during benchmark tests. Also, B-Grid AMR requires all refinement ratios

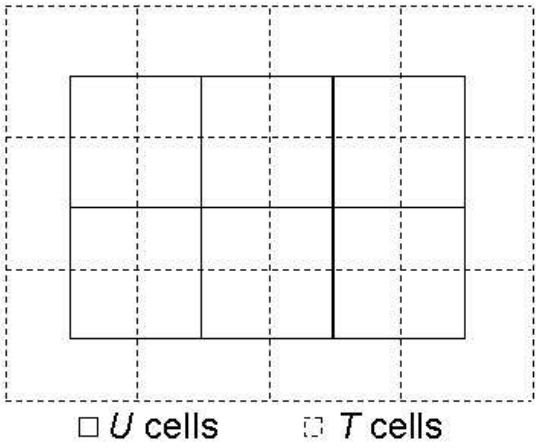


Figure 3.7: A staggered grid patch on level-0. For this work, the U patch is positioned on the interior of the T patch. The opposite arrangement could be used, but is not addressed here.

to be odd numbers [1]. Figure 3.8 illustrates how even refinement ratios create refinement levels which are no longer staggered. Odd ratios ensure that U cells are always defined on the nodes of T cells.

A subtle issue of the dual hierarchy approach occurs in indexing. The indices (i, j) on level-0 correspond to cells which touch on T and U patches. Specifically, data with indices (i, j) of a U patch lie on the upper right node of (i, j) on a T patch (see Figure 3.8). This arrangement is desired on all refinement levels, but only occurs naturally on level-0. For finer levels where the ratio to level-0 is defined as $(\rho_\ell^\lambda, \rho_\ell^\phi) = (\frac{\Delta\lambda_0}{\Delta\lambda_\ell}, \frac{\Delta\phi_0}{\Delta\phi_\ell})$, there is an index offset

$$\begin{aligned} \text{offset}_\ell^\lambda &= (\rho_\ell^\lambda - 1) \ / \ 2 \\ \text{offset}_\ell^\phi &= (\rho_\ell^\phi - 1) \ / \ 2 \end{aligned} \tag{3.27}$$

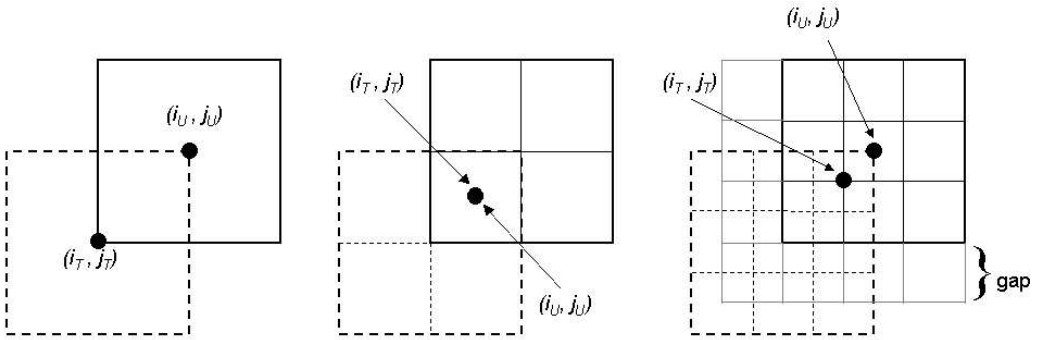


Figure 3.8: (Left) The U cell (i_u, j_u) lies naturally on the upper right node of the T cell (i_t, j_t) on level-0. (Middle) Even refinement ratios collocate T and U cells. (Right) Odd refinement ratios keep the grid staggered, shift (i_t, j_t) by $(\text{offset}_1^\lambda,\ \text{offset}_1^\phi)$, and create a "staggered grid gap".

and indices on U and T patches have the relation

$$\begin{aligned} i_t &= i_u + \text{offset}_\ell^\lambda \\ j_t &= j_u + \text{offset}_\ell^\phi \end{aligned} \tag{3.28}$$

Using this convention, the U cell (i_u, j_u) always lies on the upper right node of T cell (i_t, j_t).

Complementary patches are created in the T hierarchy for each patch in the U hierarchy. A refined U cell (i_u, j_u) on level-(ℓ) requires T cells (i_t, j_t), (i_t+1, j_t), (i_t, j_t+1), and (i_t+1, j_t+1) to be refined as well (see Figure 3.8). If the refinement ratio is 3 in each direction, there will be 9 U cells on level-$(\ell+1)$, and a total of 36 T cells on level-$(\ell+1)$. A "staggered grid gap" is created as a result, and extra ghost cells must be added around the border of the U patch in order to perform the desired numerics on the outer most level-$(\ell+1)$ T cells. The extra ghost cell width is

$$\begin{aligned} \text{Gap}_\ell^\lambda &= (r_{\ell-1}^\lambda - 1) \ / \ 2 \\ \text{Gap}_\ell^\phi &= (r_{\ell-1}^\phi - 1) \ / \ 2 \end{aligned} \tag{3.29}$$

where r_ℓ^λ and r_ℓ^ϕ are defined in equation (3.1). The amount of ghost rows added to tracer patches is the number needed by the numerical method on a non-AMR grid. Velocity patches require both the number of ghost rows needed for the non-AMR grid numerics and those required to fill the staggered grid gap.

This construction allows B-Grid AMR to be performed without the need for writing a separate set of communication algorithms for nodes and cell centers. U and T data are each treated as cell centered, but reside in two separate refinement hierarchies. Consequently, both data types are able to use the same hierarchy operations such as refinement, coarsening, time advancement, and flux matching. U and T patches only require communication with each other during the calculation of fluxes.

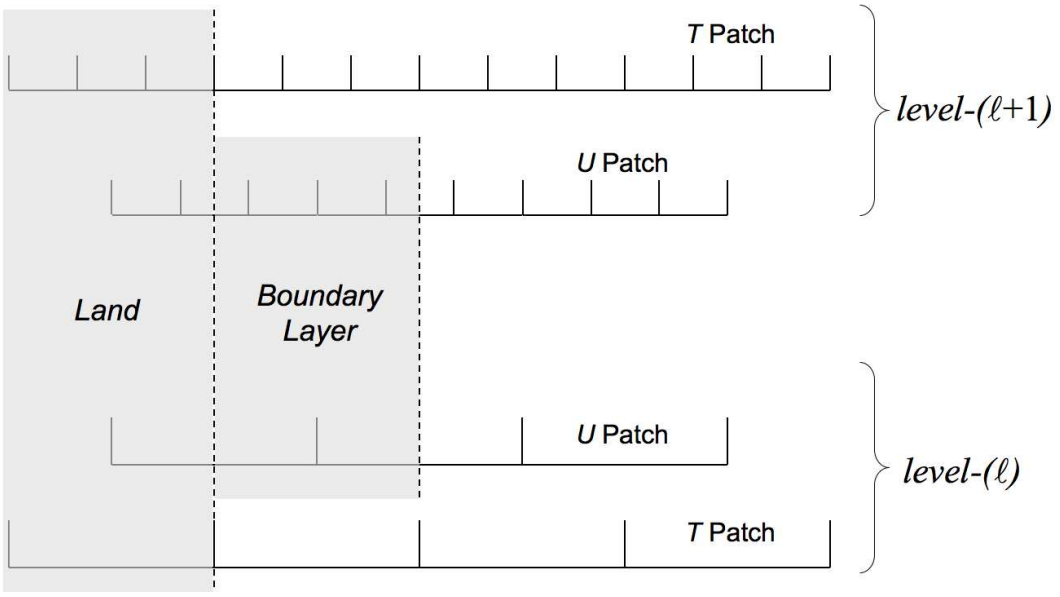


Figure 3.9: One dimensional example of T and U patch refinement in the presence of land. There are more U cells in the boundary layer on level-$(\ell + 1)$.

3.3.1 Topography

The AMR Ocean Model sets topography at the start of a simulation in a separate dual-hierarchy refined over the entire domain. This eliminates the need for calculating topography "on the fly" as refinement patches change size and position. Topography data are distributed among multiple processors using SAMRAI's load balancing routines. The ocean depth and land mask are set on level-(0) T cells from data read from a file. T cells on levels finer than level-(0) (see Figure 3.9) are given the same depth and mask as their underlying coarse cell. The U cell depth and mask are set using equations (2.78) and (2.79) on each refinement level.

This arrangement keeps the T cell topography uniform on all refinement levels and retains the refine-coarsen inverse (3.2)-(3.3) for all T cell variables. The U cell topography varies with refinement in order to better resolve flow fields in the boundary layer. The refine operation given by equation (3.8) will not properly refine the boundary layer unless a modification is made to the definition of Φ^λ and Φ^ϕ. Variables defined on U cells are therefore refined using

$$\Phi^\lambda = \frac{1}{\cos\phi_n} \begin{cases} \delta^\lambda(\overline{\Phi_{m,n}}^\lambda) & , \quad M_{m,n} > 0 \\ \delta^\lambda(\Phi_{m,n}) & , \quad M_{m+1,n}\,(1-M_{m-1,n})\,(1-M_{m,n}) > 0 \\ \delta^\lambda(\Phi_{m-1,n}) & , \quad (1-M_{m+1,n})\,M_{m-1,n}\,(1-M_{m,n}) > 0 \end{cases} \tag{3.30}$$

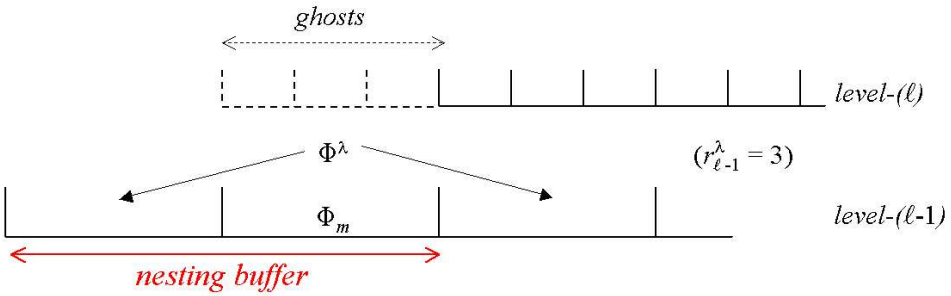


Figure 3.10: One dimensional example of a level-(ℓ) patch nested within level-($\ell - 1$) by 2 level-($\ell - 1$) cells (red). The refine operator uses interior level-($\ell - 1$) cells to fill level-(ℓ) ghosts.

and

$$\Phi^{\phi} \quad = \quad \begin{cases} \delta^{\phi}(\overline{\Phi_{m,n}}^{\phi}) & , \quad M_{m,n+1} \; > \; 0 \\ \delta^{\phi}(\Phi_{m,n}) & , \quad M_{m,n+1} \; (1 - M_{m,n-1}) \; (1 - M_{m,n}) \; > \; 0 \\ \delta^{\phi}(\Phi_{m,n-1}) & , \quad (1 - M_{m,n+1}) \; M_{m,n-1} \; (1 - M_{m,n}) \; > \; 0 \end{cases} \tag{3.31}$$

in the refinement operation. The above definition allows refinement of the fluid boundary layer but does not retain the refine-coarse inverse property globally. This is not an issue of concern given that the numerical model itself does not conserve momentum in the boundary layer.

3.3.2 Ghost Cells

The ghost cell widths assigned to state variables (see Table 3.1) are based on spatial discretization, temporal discretization, and refinement ratio. The SAMRAI library does not allow ghost cell widths to change during the simulation, and the maximum ghost cell width is always available as a result. For example, the maximum tracer ghost rows, 2, is only required for mixing steps but is also available on leap frog steps. Also, refinement

State Variable		*Ratio To Level-(ℓ-1)* 1	3	5
Tracers	(T, S)	2	2	2
Surface Height	(η)	2	2	2
Baroclinic Velocity	(u, v)	2	3	4
Barotropic Velocity	(U, V)	2	3	4
Barotropic Forcing	(X, Y)	1	2	3

Table 3.1: Maximum ghost cell widths required by state variables for non-AMR simulations, and AMR simulations with refinement ratios equal to 3 and 5. The latter two cases require extra U cell ghost rows to fill the staggered grid gap.

levels finer than level-(1) are nested within coarser levels such that the refinement operator (3.8) uses interior data of the underlying coarse level (see Figure 3.10). The number of level-(ℓ) ghost rows and the refinement ratio, $r_{\ell-1}$, determine the number of level-($\ell - 1$) cells which define the nesting buffer. This is in general given by the relation,

$$\text{nesting} \;=\; \frac{\text{ghosts} - 1}{r_{\ell-1}} \;+\; 2 \;, \tag{3.32}$$

where the fraction represents integer division. The minimum nesting of 2 is obtained when (ghosts $\leq$ $r_{\ell-1}$), and Table 3.1 shows this condition is true for the ghost rows required by the AMR Ocean Model. Thus, a nesting buffer width of 2 level-($\ell - 1$) cells is imposed on refinement levels finer than level-(1).

3.4 Regridding

A regrid procedure is used periodically in AMR to change positions of refinement patches so that localized areas of interest stay resolved. This involves specifying a new set of cells for refinement and creating a new set of patches. New refinement patch locations are determined during the regrid based on a specified set of refinement criteria. A test

is performed on each level-(ℓ) cell based on some refinement criteria. Cells are tagged as "1" or "0" depending on if additional refinement is or is not required, respectively. A clustering algorithm creates a new set of refinement patches for level-($\ell + 1$). A load balancing algorithm subdivides patches further and assigns a processor to each patch. Cells that were not refined before the regrid are initialized by interpolation from the next coarser level. The clustering and load balancing algorithms will generally guarantee that most, but not all level-(ℓ) cells with tag "1" will be refined. It is sometimes more efficient to negate and/or add refinement on the regridded level in order to better cluster refinement patches which must be within a specified minimum and maximum patch size. See [10, 38, 39] for more details on the clustering and load balancing algorithms implemented by SAMRAI.

Traditional AMR allows each refinement level, ℓ, to regrid independently after a specified number of steps, R_ℓ. Leapfrog AMR time integration requires all refinement levels to regrid in unison. If a level changes location at $t = \tau$ as a result of a regrid, the solution at $t < \tau$ is not guaranteed to be defined everywhere after the regrid. This is not a problem with traditional AMR time integration since the numerical method only uses $\mathbf{U}^n$ defined at $t = \tau$ to advance to $\mathbf{U}^{n+1}$. With leapfrog AMR, data at $t < \tau$ and $t = \tau$ must coincide on the same index space. Ignoring this requirement creates an inconsistency when flux sums are initialized as $\tilde{F}_{\ell+1} = \tilde{f}_{\ell+1}$ because $\tilde{F}_{\ell+1}$ is defined on the new hierarchy and $\tilde{f}_{\ell+1}$ is defined on the previous hierarchy. To avoid this discrepancy, the AMR Ocean Model performs a mixing step on level-(0) as the first step after each regrid. This forces the AMR hierarchy to time step as though it were the first step in the simulation thus avoiding the index space discrepancy.

Another alternative makes use of a "pre-advance" after each regrid. The data at $t = \tau - \Delta t_0$ is saved as a temporary state ($\mathbf{U}^{tmp}$) on all levels. $\mathbf{U}^n$ and $\mathbf{U}^{tmp}$ are regridded at $t = \tau$ so that they both lie on the same index space. The hierarchy then pre-advances levels finer than the level-(0) from $t = \tau - \Delta t_0$ to $t = \tau$, starting with a mixing step on each level. Early test cases of the leapfrog AMR time stepping made use of the pre-advance [34] until it proved to add significant increases in overhead and algorithm complexity. A level-(0) mixing step after each regrid is more favorable for the AMR Ocean Model and is therefore used in place of a pre-advance.

The regrid adds mixing steps to leapfrog AMR which can be undesirable unless the regrid interval is properly specified. For example, a time refinement ratio of 3 between levels with a regrid every third level-0 step will mix level-1 after 9 steps and level-2 after 27 steps. It is common to mix every 17 steps in non-AMR leapfrog simulations. Thus, a regrid after three level-0 steps will mix levels 0 and 1 more frequently and will mix level-2 less frequently than is typical for leapfrog without AMR. Since the fluid motion is relatively slow for geophysical flows, the hierarchy can be regridded less frequently in the AMR Ocean Model. Doing so will reduce mixing steps and overhead associated with the regrid.

3.4.1 Refinement Criteria

Refinement criteria can be specified in several ways. The simplest method involves prior knowledge of the solution which allows refinement to be placed in a specific location during the regrid. This should be used in practice whenever possible to minimize overhead created from automated refinement criteria. The latter is typically used with AMR since many continuum problems contain unpredictable solutions. This requires the creation of

temporary variables during the regrid. The cell tags are stored in the variable $\text{Tags}_{i,j}$ which is initialized to 0. One or more calculations are performed to determine which indices of $\text{Tags}_{i,j}$ are set to 1 indicating the need for refinement. Each calculation is stored in a variable, $\text{Tag\_Calc}_{i,j,c}$, where c ranges from 1 to the number of tag calculations. Common tag calculations include gradient, vorticity, and order of accuracy error estimations.

The AMR Ocean Model's gradient tag calculation implemented on a scalar $U_{i,j}$ stores the maximum approximation of the derivative in the $\hat{\lambda}$ and $\hat{\phi}$ directions,

$$\text{Tag\_Calc}_{i,j,c} \;=\; \max\left(\left|U^{\lambda}_{i,j}\right|,\ \left|U^{\phi}_{i,j}\right| \right) \ , \tag{3.33}$$

where $U^{\lambda}_{i,j}$ and $U^{\phi}_{i,j}$ are defined in equations (3.30) and (3.31). The gradient calculation applied to a velocity variable, $\vec{V}_{i,j} = u_{i,j}\hat{\lambda} + v_{i,j}\hat{\phi}$, stores the maximum of $\left|u^{\lambda}_{i,j}\right|$, $\left|u^{\phi}_{i,j}\right|$, $\left|v^{\lambda}_{i,j}\right|$, and $\left|v^{\phi}_{i,j}\right|$. A tag calculation based on the $\hat{z}$ component of vorticity is defined as

$$\text{Tag\_Calc}_{i,j,c} \;=\; \left|u^{\phi}_{i,j} - v^{\lambda}_{i,j}\right| \ . \tag{3.34}$$

The velocity magnitude can also be used as a tag calculation,

$$\text{Tag\_Calc}_{i,j,c} \;=\; \sqrt{u^2_{i,j} + v^2_{i,j}} \ . \tag{3.35}$$

The tag variable is set to 1 for each cell that has a tag calculation greater than a specified threshold,

$$\text{Tags}_{i,j} \;=\; \textbf{true}, \qquad \textbf{if } \left(\text{Tag\_Calc}_{i,j,c} \;>\; \mu_{\ell,c}\right) \ . \tag{3.36}$$

This process can be performed using any number of tag calculations. The area of interest is better captured as more tag calculations are performed, but overhead is also increased.

The tagging procedure can be amended to place an upper limit, $\text{max\_frac\_tagged}_\ell$, on the number of cells tagged on level-(ℓ). The maximum number of tagged cells permitted on level-(ℓ) is then

$$\text{upper\_tag\_limit}_\ell \quad = \quad \text{max\_frac\_tagged}_\ell \; N_\lambda N_\phi \; , \tag{3.37}$$

where (N_λ, N_ϕ) are the number of level-(ℓ) cells in the $(\hat{\lambda}, \hat{\phi})$ directions present when level-(ℓ) covers the entire domain. The level's $\text{Tag\_Calc}_{i,j,c}$ variable is first reduced to the maximum of all tag calculations.

$$\text{Tag\_Calc}_{i,j} \quad = \quad \max_c \left(\text{Tag\_Calc}_{i,j,c} \right) \tag{3.38}$$

Cells with $\text{Tags}_{i,j} = 1$ are then ranked using Quicksort [57] from 1 for the smallest $\text{Tag\_Calc}_{i,j}$ to $\sum_{i,j} \text{Tags}_{i,j}$ for the largest $\text{Tag\_Calc}_{i,j}$. Cells with $\text{Tags}_{i,j} = 1$ are set to zero if their rank is less than or equal to $\left(\sum_{i,j} \text{Tags}_{i,j} \; - \; \text{upper\_tag\_limit}_\ell \right)$. Note that $\text{max\_frac\_tagged}_\ell = 0$ sets all $\text{Tags}_{i,j}$ equal to zero, and $\text{max\_frac\_tagged}_\ell = 1$ leaves $\text{Tags}_{i,j}$ unaltered.

3.4.2 Modifications to the SAMRAI Clustering Algorithm

The nesting buffer discussed in Section 3.3.2 places a potential constraint on the amount of refinement as level numbers increase. The AMR Ocean Model compensates for the nesting buffer by using a tag buffer. The (i^{th}, j^{th}) index of $\text{Tags}_{i,j}$ is set to 1 if any neighboring index is 1. This is written logistically in Algorithm 3.4.1.

Algorithm 3.4.1. Tag Buffering

```
if ( Tags_{i,j} == 1 )

   for ( i' = i - 1 → i + 1 )
      for ( j' = j - 1 → j + 1 )
         Buffered_Tags_{i',j'} = 1
      end
   end

else

   Buffered_Tags_{i,j} = 0

end
```

The numerics of the AMR Ocean Model were noted to behave poorly when refinement level-$(\ell+1)$ U patches are positioned apart by 1 level-(ℓ) cell (see Figure 3.11). Such situations create an excess of feedback between flux matching and ghost filling which can severely corrupt the solution when using large time step refinement ratios or large subcycling numbers. These scenarios are avoided during the regrid by modifying the refinement patches suggested by SAMRAI's clustering algorithm where level-$(\ell+1)$ U patches are separated by 1 level-(ℓ) cell. The patch with fewer cells is grown by r_ℓ level-$(\ell+1)$ cells in the direction of its neighbor that produces the gap. If this alteration invades the nesting buffer, it is negated and the patch with the most cells is shortened by r_ℓ thus widening the level-(ℓ) gap to 2.

3.5 Applying the AMR Method in the AMR Ocean Model

The concepts described in this chapter are theoretically sufficient to implement the numerical ocean model from Chapter 2 in an AMR environment. Several implementation

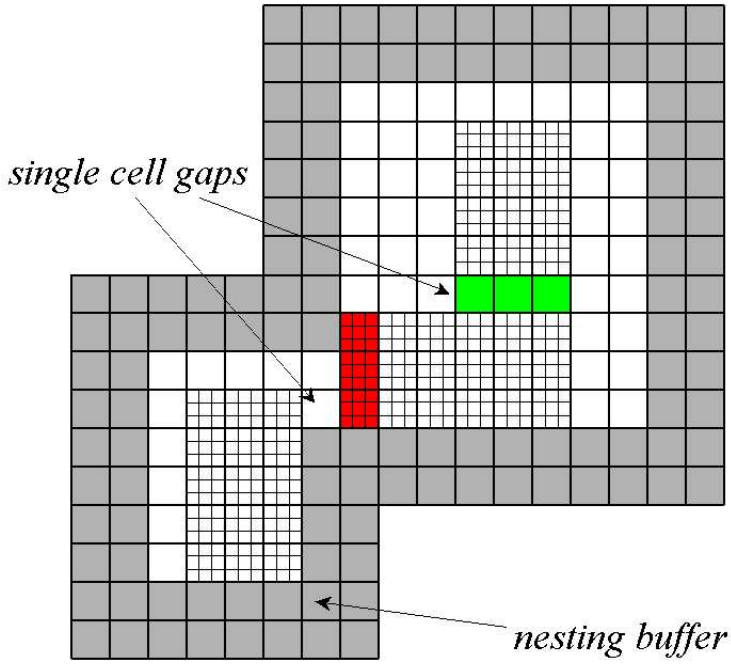


Figure 3.11: Examples of U refinement patches on level-$(\ell+1)$ oriented such that they are separated by a distance equal to a single level-(ℓ) cell. These scenarios are avoided without invading the nesting buffer (gray). One gap is removed by extending a level-$(\ell+1)$ patch to cover the cells shown in green. The other gap is widened by removing the level-$(\ell+1)$ cells shown in red.

options are specified before this is in fact possible. These options can in some instances be altered if a more sophisticated model is desired. The choices made here are suitable for the applications given in Chapter 5 and Chapter 6.

The surface tracer restoring and wind stress data are read from files and spatially interpolated to each refinement level's resolution. Tracer fields are initialized either from spatially interpolated file data or set to a constant. Velocity fields are initialized to zero. Topography is read from file data as explained in Section 3.3.1.

The interface flux matching procedure described in Section 3.2 is applied to all non-pressure flux quantities in the spatial discretization section of Chapter 2. Pressure flux

matching is neglected on the basis that hydrostatic pressure gradients are due to external forces rather than fluid motion itself. Pressure terms are therefore characterized as source terms and are not given special attention at interfaces.

Advective velocities calculated on the faces of T and U cells are matched at interfaces in order to preserve continuity. A zero divergence velocity field at sub-surface depths ($k > 1$) is maintained after flux matching on level-(ℓ) by re-calculating the vertical T and U velocities on cells that abut level-($\ell+1$). The zero advective flux boundary condition at the surface ($k = 1$) creates a velocity field which is rarely divergence free. Artificial numerical sources and sinks are created as a result which are offset in traditional ocean models by surface restoring boundary conditions. This is often insufficient in the AMR Ocean Model when combined with the additional numerical artifacts created by AMR interfaces. These effects are dampened by applying the checkerboard filter described in Section 2.7.6 to tracer fields in the surface layer.

Numerical dispersion is created by leap frog AMR in advection simulations. This is demonstrated in the following chapter, and a smoothing operation is shown to dampen dispersion when applied near AMR interfaces. The AMR Ocean Model applies the smoother in Section 4.2 after each leap frog step to the outer two rows of T cells on level-($\ell+1$) patches where a refinement interface with level-(ℓ) is present.

Chapter 4

Benchmark Tests

A numerical model must undergo multiple tests in order for simulations to be conducted with confidence. Standard benchmark tests help the model developer find errors in computer code and point out the strengths and weaknesses of the numerical model. Accuracy is typically evaluated by comparing the model's solution to an analytic solution if available. Tests which have no analytic solution can demonstrate the model's accuracy by comparing the numerical solution at various resolutions.

The first three tests presented in this chapter contain far fewer degrees of freedom than a full ocean model. The results primarily reflect the AMR algorithms described in Chapter 3. Refinement is time varying, but specified according to the analytic solution. The remaining tests are geophysical and demonstrate the AMR Ocean Model's accuracy with increased refinement using various refinement criteria. Run time performance is evaluated in Chapters 5 and 6 using simulations with more typical ocean model parameters.

4.1 Advection with Dynamic Refinement

The leapfrog AMR time advancing scheme is compared to single-step and multi-step time integrators for the 2D advection problem

$$\frac{\partial \varphi}{\partial t} = -u\frac{\partial \varphi}{\partial x} - v\frac{\partial \varphi}{\partial y} \tag{4.1}$$

where φ is an unknown within a constant velocity field $\vec{V} = (u, v)$. For an initial condition $\varphi_0(x, y)$, the analytic solution at time t is given by $\varphi(x, y, t) = \varphi_0(x - ut,\ y - vt)$. The advection test is performed on a periodic grid of unit length with advection velocity $(u, v) = (1, 1)$. A Gaussian initial condition,

$$\varphi_0(x,\ y) = \exp\left(\frac{-(x-\frac{1}{2})^2 - (y-\frac{1}{2})^2}{g_r^2}\right), \tag{4.2}$$

with $g_r = 0.08$ is advanced with 2 levels of refinement for five cycles. The level-(0) resolution is 60×40, and each level's refinement ratio is a factor of 3 in each dimension. Fine (540×360) and coarse (60×40) grid simulations are also conducted without AMR for comparison. The refinement operation (3.6) is used for leapfrog and multi-step time integration. The single-step time integrator requires preservation of monotonicity, which is satisfied when calculating Φ^x and Φ^y using the van Leer [72] slope limiter. The third order Runge Kutta method is used as the multi-step integrator since the second order Runge Kutta method is unstable for pure advection as shown in [37].

Average and differencing operator notation similar to that described in Section 2.6.2 is used to discretize the advection equation:

$$\overline{q_{i,j}}^x = \frac{1}{2}(q_{i,j} + q_{i-1,j}) \tag{4.3}$$

$$\delta^x q_{i,j} = \frac{q_{i+1,j} - q_{i,j}}{\Delta x} \tag{4.4}$$

Each numerical method is written in conservative flux form so that equation (4.1) becomes

$$\left.\frac{\partial \varphi}{\partial t}\right|_{i,j} \;=\; \delta^x(\text{Flux\_w}_{i,j}) \;+\; \delta^y(\text{Flux\_s}_{i,j}) \tag{4.5}$$

where the flux terms for leapfrog and Runge Kutta are

$$\text{Flux\_w}_{i,j} \;=\; -u\;\overline{\varphi_{i,j}}^{x} \tag{4.6}$$

$$\text{Flux\_s}_{i,j} \;=\; -v\;\overline{\varphi_{i,j}}^{y}\;. \tag{4.7}$$

Fluxes in the single-step integrator are defined using the Godunov method [18] which in the x-direction is

$$\text{Flux\_w}_{i,j} \;=\; -u\left(\varphi_{i,j} \;+\; \frac{1}{2}(\Delta x - u\Delta t)\Delta^x\varphi_{i,j} \;-\; \frac{\Delta t\; v}{2}\delta^y\varphi_{i,j-1}\right)\;. \tag{4.8}$$

Here, the derivative term $\Delta^x\varphi_{i,j}$ is calculated using the same slope limiter as described above for Φ^x and Φ^y.

The time step for each method is calculated as a fraction of the maximum stable time step. Leapfrog and third order Runge Kutta time steps are

$$\Delta t_{LF} \;=\; \frac{\mu}{\sqrt{u^2+v^2}\;\sqrt{\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}}} \tag{4.9}$$

$$\Delta t_{RK} \;=\; \sqrt{3}\;\Delta t_{LF}\;, \tag{4.10}$$

and the Godunov time step is

$$\Delta t_{GD} \;=\; \mu\;\min\left(\frac{\Delta x}{u},\;\frac{\Delta y}{v}\right)\;. \tag{4.11}$$

The parameter μ in (4.9) and (4.11) is set to 0.90.

It is desirable that the AMR hierarchy be constructed so that the solution has no sharp gradients at refinement interfaces. This can be accomplished by regridding after each step, but will add undesired overhead. A more efficient approach is to make refinement patches larger and regrid less frequently. The increased patch size is problem dependent and will vary with the grid resolution, time step, and solution velocity. The refinement patches for the advection test are chosen to be large enough to maintain refinement in areas where the solution has steeper gradients for three level-(0) steps. The refinement patch positions are shifted during each regrid by $3\Delta t_0 \vec{V}$ rounded to the closest integer, where the quantity $\vec{V}$ is again the analytic velocity. As a result, the leapfrog AMR simulations will perform mixing steps on the fine level every 27 steps. This same frequency is therefore used for mixing steps in the non-AMR runs. The leapfrog test makes use of the pre-advance option (see Section 3.4) during each regrid. This is permissible for test cases like advection which model a relatively small number of variables.

Figure 4.1 shows snapshots of the Gaussian advection for 1 cycle. Table 4.1 lists the RMS error and run time of each numerical method as applied on AMR and fine resolution grids. The RMS error is given by

$$RMS \quad = \quad \frac{\|\mathfrak{q} - q\|}{\|\mathfrak{q}\|}, \tag{4.12}$$

where $\mathfrak{q}$ and q are the analytic and numeric solutions respectively. The AMR error is very close to that of a fine grid resolution while yielding a significant reduction in run time. The error with leapfrog is slightly higher than that using Godunov and lower than that using Runge Kutta for the cases using fine resolution or AMR. Run times using leapfrog are shorter than those using Runge Kutta and Godunov in non-AMR cases. This is due to

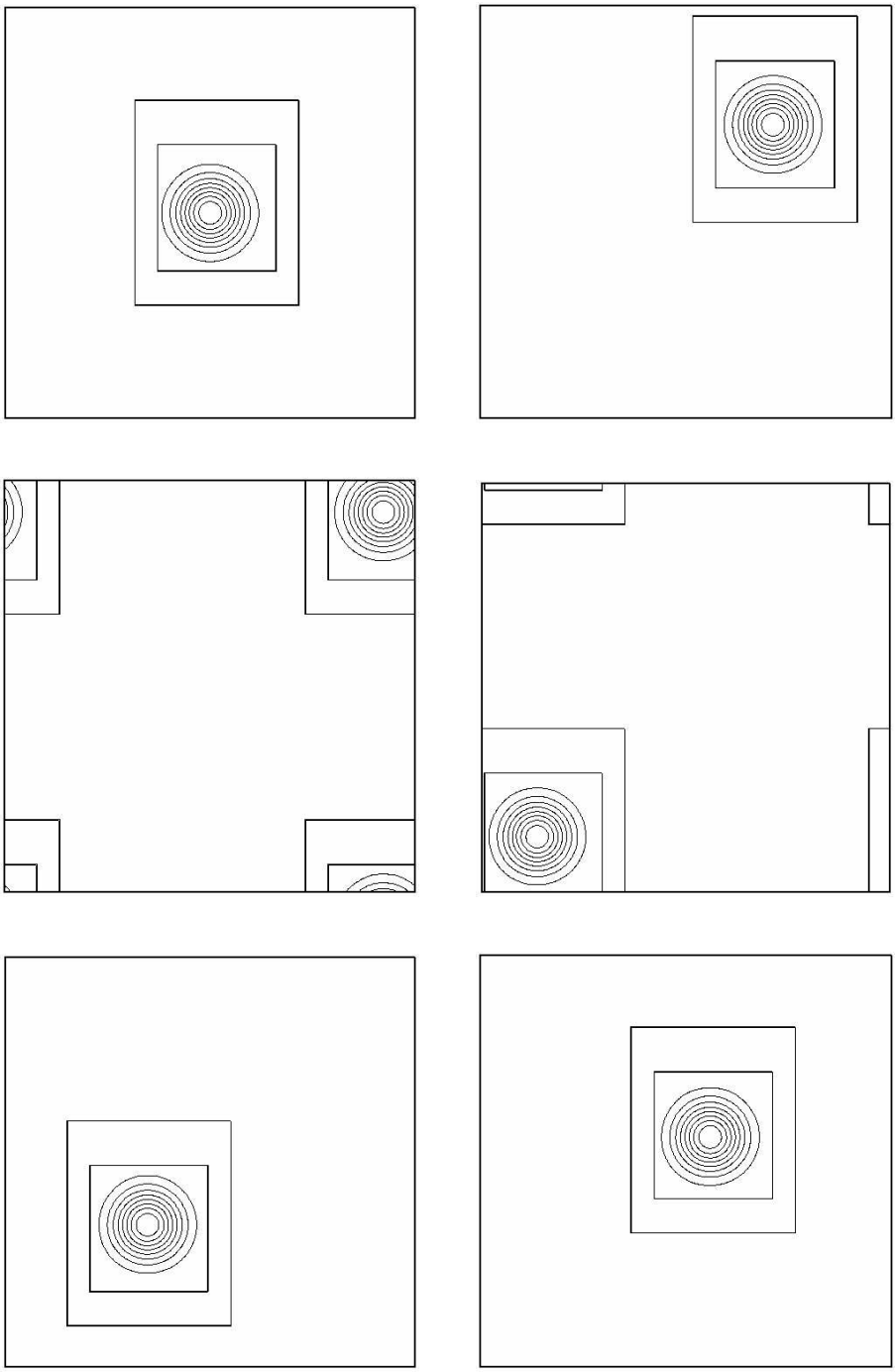


Figure 4.1: One complete advection cycle using AMR with leapfrog time integration.

	RMS (%)		Run Time (sec)	
	Fine	AMR	Fine	AMR
Godunov	0.172	0.256	358	71
Leapfrog	0.416	0.422	194	32
Runge Kutta	0.520	0.596	331	27

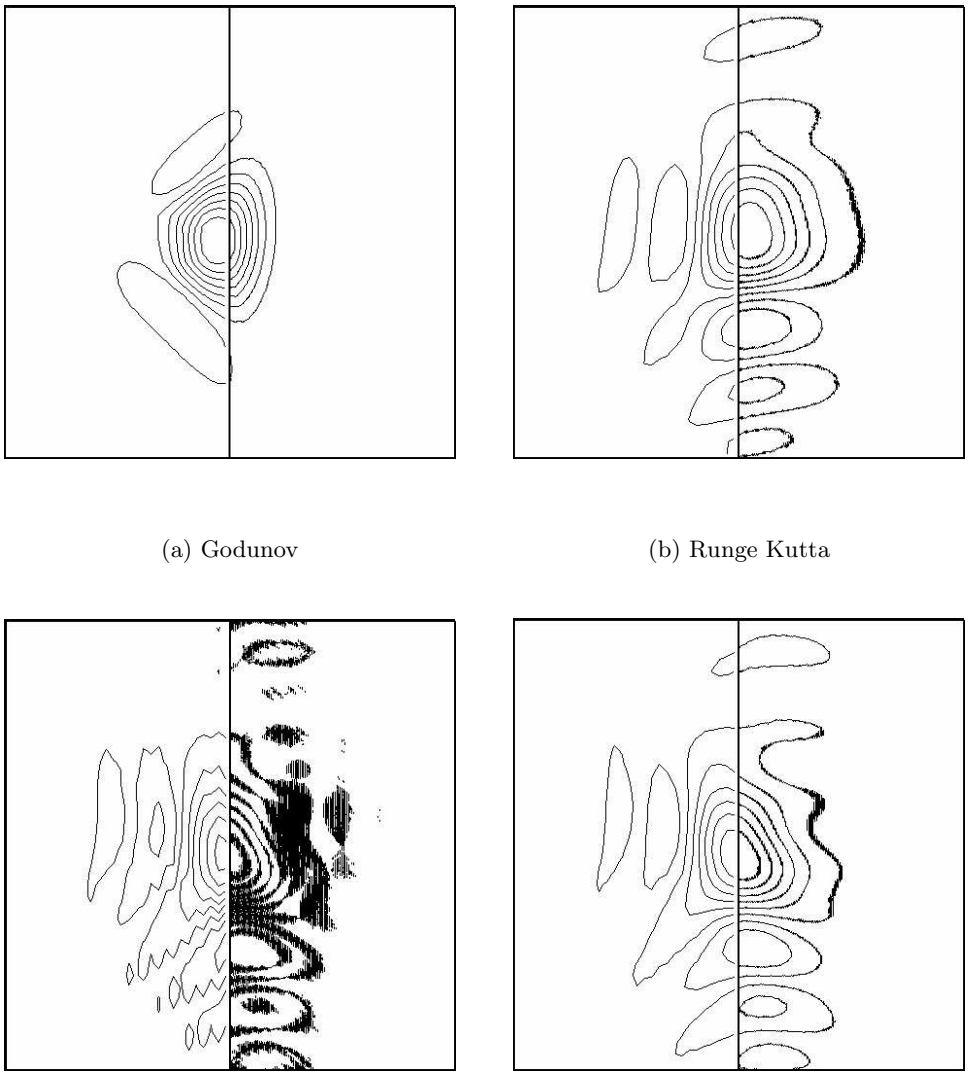
Table 4.1: RMS errors and run times for the Gaussian advection tests with dynamic refinement after 5 cycles. AMR produces an RMS error which is comparable to a uniform fine grid resolution while yielding a much shorter run time.

the fact that leapfrog does not require multiple time steps or slope limiting even though a smaller Δt is used. AMR with leapfrog has a slightly longer run time than Runge Kutta due to extra computation in the pre-advance at every third level-(0) step. This overhead can be reduced for slow-moving flows by regridding less frequently.

4.2 Advection Through a Coarse/Fine Interface

The 2D advection test demonstrates how AMR can obtain high spatial accuracy without increasing the resolution of the entire domain. The objective of moving refined regions is to keep refinement boundaries in regions where the solution has no sharp gradients and thereby minimize errors produced at coarse/fine interfaces. This is accomplished in scalar advection by properly choosing the regrid frequency and patch size. However, when modeling a more complex system with multiple variables (e.g. the ocean) it is likely that some variables will have small gradients at interfaces while other variables will have steeper gradients.

A test was made to demonstrate how leapfrog, Runge Kutta, and Godunov nu-



(a) Godunov

(b) Runge Kutta

(c) Leapfrog

(d) Leapfrog with interface smoothing

Figure 4.2: Advection through coarse/fine interfaces. The left side of the domain has grid spacing $(\Delta x,\ \Delta y) = (\frac{1}{60},\ \frac{1}{40})$ and the right side of the domain is refined by a factor of 9 in each spatial direction. The results shown are after 4 complete cycles.

merics behave when a solution has steeper gradients at refinement interfaces. A Gaussian initial condition (4.2) is advected ten times through a fine region using $\vec{V} = (1, 1)$. The same discretization and time steps are used for each integrator as in Section 4.1. The resolution of level-(0) is set to 60×40. The refinement hierarchy is not moved with the Gaussian, but instead, the right side of the domain is refined in a single level by a factor of 9 in each spatial direction. This creates a coarse (left) to fine (right) interface at $x = 0.5$ and a fine to coarse interface at $x = 0.0$.

Figures 4.2a and 4.2b show that Godunov and Runge Kutta behave reasonably well in the presence of an interface. The leapfrog case in Figure 4.2c shows an excess of reflection unless an interface smoother is used at the edges of level-(1) (Figure 4.2d). The smoother adds contributions from the nearest neighbors of each cell and is of the form

$$\begin{aligned} \text{Ave\_X}_{i,j} &= \frac{1}{4}(q_{i-1,j} + 2q_{i,j} + q_{i+1,j}) \\ \text{Ave\_Y}_{i,j} &= \frac{1}{4}(q_{i,j-1} + 2q_{i,j} + q_{i,j+1}) \\ q_{i,j} &= (1 - \lambda_s)q_{i,j} \; + \; \frac{\lambda_s}{2}\left(\text{Ave\_X}_{i,j} \; + \; \text{Ave\_Y}_{i,j}\right) \end{aligned} \tag{4.13}$$

Here, λ_s is set to 0.5, and the smoother is applied to the outer 2 rows of cells on patches where refinement interfaces are present on levels finer than level-(0). The RMS error for each integrator in the interface test is given in Figure 4.3. The reflection error produced by leapfrog grows quickly unless the interface smoother is applied.

4.3 Barotropic Modon

A geophysical benchmark [65, 44, 11, 59] is presented to test the dual hierarchy approach to B-Grid AMR. The shallow water equations are implemented with horizontal

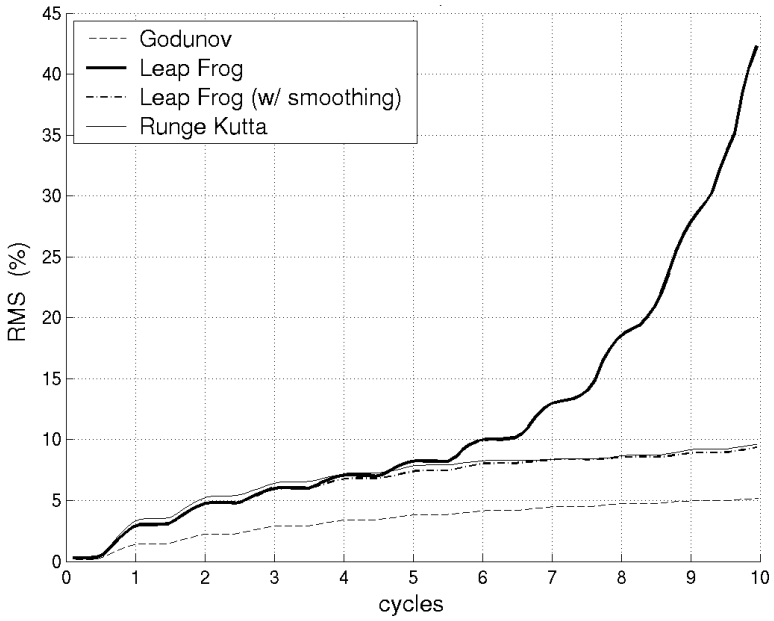


Figure 4.3: RMS error of advection through a refined region using Godunov, leapfrog, and Runge Kutta numerics. Leapfrog produces an excess of reflection that causes rapid RMS error growth first noticeable near the 5^{th} pass through the interface. A smoother applied at the interface keeps leapfrog RMS error on the same order as Runge Kutta.

refinement:

$$\frac{\partial u}{\partial t} = -(\vec{V}\cdot\vec{\nabla})u + fv - g\frac{\partial \eta}{\partial x} \tag{4.14}$$

$$\frac{\partial v}{\partial t} = -(\vec{V}\cdot\vec{\nabla})v - fu - g\frac{\partial \eta}{\partial y} \tag{4.15}$$

$$\frac{\partial \eta}{\partial t} = -H(\vec{\nabla}\cdot\vec{V}) \tag{4.16}$$

where f is the Coriolis parameter here given by the beta-plane approximation ($f \simeq f_0 + \beta y$), g is the gravitational constant, η is surface height, and H is the depth of the basin. Refer to [31] and [74] for more discussion concerning the shallow water equations. The pressure gradient, continuity, and advection terms above are similar to those used in the barotropic and baroclinic equations of the AMR Ocean Model. The spatial discretization of these terms therefore mimics the AMR Ocean Model as well. Velocities are defined at U points

and the surface height at T points. The equations are expressed in discrete form as

$$\left.\frac{\partial u}{\partial t}\right|_{i_u,j_u} = \delta^x(\text{Flux\_Uw}_{i,j}) + \delta^y(\text{Flux\_Us}_{i,j}) + (f_0+\beta y)v_{i_u,j_u} \tag{4.17}$$

$$\left.\frac{\partial v}{\partial t}\right|_{i_u,j_u} = \delta^x(\text{Flux\_Vw}_{i,j}) + \delta^y(\text{Flux\_Vs}_{i,j}) - (f_0+\beta y)u_{i_u,j_u} \tag{4.18}$$

$$\left.\frac{\partial \eta}{\partial t}\right|_{i_t,j_t} = \delta^x(\text{Flux\_hw}_{i,j}) + \delta^y(\text{Flux\_hs}_{i,j}) \tag{4.19}$$

where flux terms in the $\hat{x}$ direction are

$$\text{Flux\_Uw}_{i,j} = -(\overline{u_{i_u,j_u}}^x \, \overline{u_{i_u,j_u}}^x) - g\,\overline{\eta_{i_t,j_t+1}}^y \tag{4.20}$$

$$\text{Flux\_Vw}_{i,j} = -(\overline{u_{i_u,j_u}}^x \, \overline{v_{i_u,j_u}}^x) \tag{4.21}$$

$$\text{Flux\_hw}_{i,j} = -H\,\overline{u_{i_u,j_u}}^y \tag{4.22}$$

and those in the $\hat{y}$ direction are

$$\text{Flux\_Us}_{i,j} = -(\overline{v_{i_u,j_u}}^y \, \overline{u_{i_u,j_u}}^y) \tag{4.23}$$

$$\text{Flux\_Vs}_{i,j} = -(\overline{v_{i_u,j_u}}^y \, \overline{v_{i_u,j_u}}^y) - g\,\overline{\eta_{i_t+1,j_t}}^x \tag{4.24}$$

$$\text{Flux\_hs}_{i,j} = -H\,\overline{u_{i_u,j_u}}^x \ . \tag{4.25}$$

The shallow water equations are tested using a barotropic modon [47, 65, 11]. This analytic solution is a double vortex which moves unchanging to the right with constant velocity equal to $c = \beta a^2$, where a is the modon radius. The initial condition is given by the stream function

$$\psi(r,\theta) = \beta a^3 \sin\theta \begin{cases} \frac{1}{k^2}\frac{J_1(k\frac{r}{a})}{J_1(k)} - (1+\frac{1}{k^2})\frac{r}{a} & , \quad r \le a \\ -\frac{K_1(\frac{r}{a})}{K_1(1)} & , \quad r > a \end{cases} \tag{4.26}$$

where J_1 and K_1 are Bessel functions of order one, k is the interior region wavenumber (3.9226), and r and θ are the radius and angle from the center of the double vortex. The

initial velocities, $\vec{V}_0$, are calculated from the stream function

$$\begin{aligned} u_0 &= -\frac{\partial \psi}{\partial y} \\ v_0 &= \frac{\partial \psi}{\partial x}, \end{aligned} \tag{4.27}$$

giving

$$u_0(r,\theta) = \beta a^2 \begin{cases} -\frac{\sin^2\theta}{2kJ_1(k)}(J_0(\frac{kr}{a}) - J_2(\frac{kr}{a})) - \frac{a}{kr}\frac{\cos^2\theta}{kJ_1(k)}J_1(\frac{kr}{a}) + (1+\frac{1}{k^2}) & , \quad r \le a \\ -\frac{\sin^2\theta}{2K_1(1)}(K_0(\frac{r}{a}) + K_2(\frac{r}{a})) + \frac{a}{r}\frac{\cos^2\theta}{K_1(1)}K_1(\frac{r}{a}) & , \quad r > a \end{cases} \tag{4.28}$$

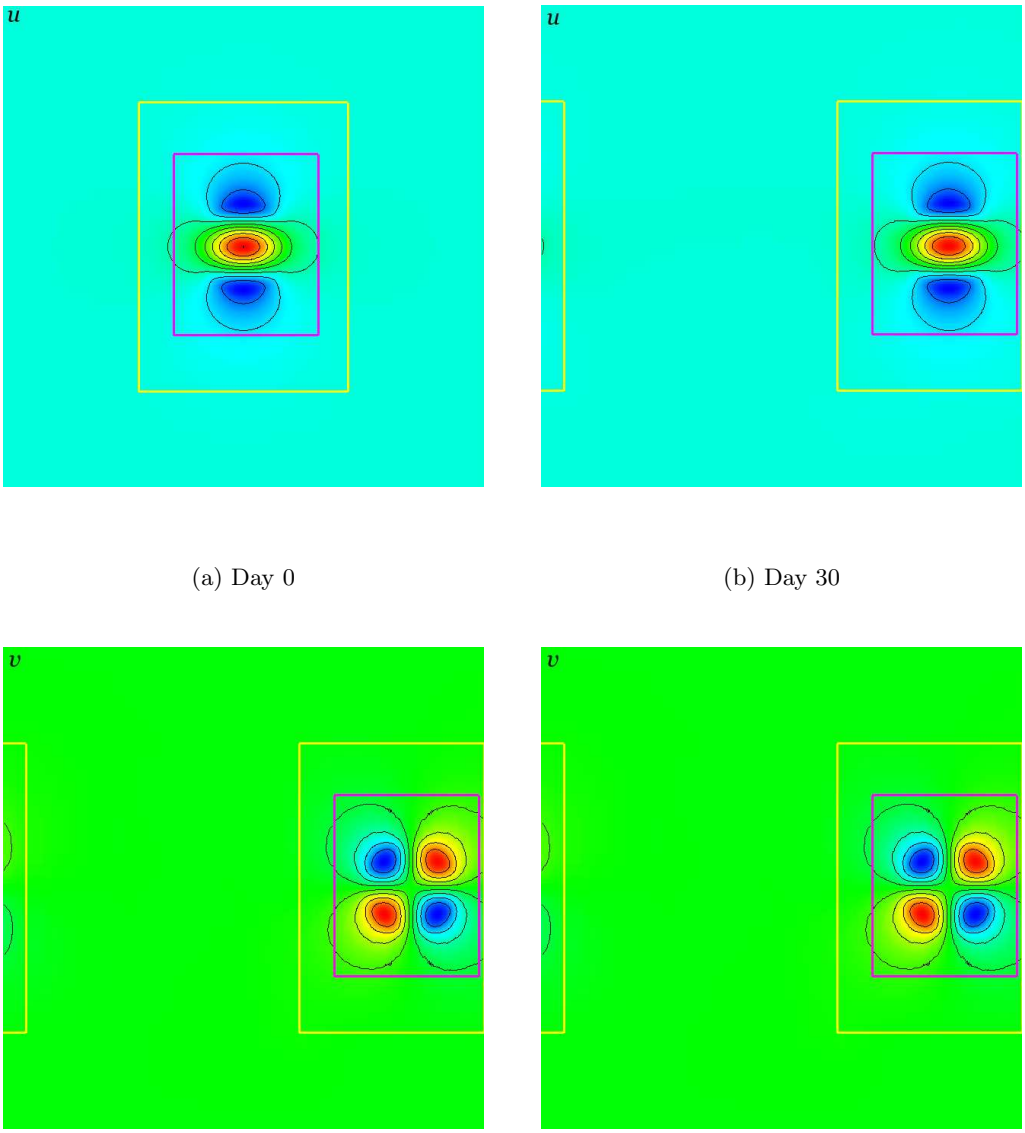
$$v_0(r,\theta) = \beta a^2 \begin{cases} \frac{\cos\theta\sin\theta}{2kJ_1(k)}(J_0(\frac{kr}{a}) - J_2(\frac{kr}{a})) - \frac{a}{kr}\frac{\cos\theta\sin\theta}{kJ_1(k)}J_1(\frac{kr}{a}) & , \quad r \le a \\ -\frac{\cos\theta\sin\theta}{2K_1(1)}(K_0(\frac{r}{a}) + K_2(\frac{r}{a})) - \frac{a}{r}\frac{\cos\theta\sin\theta}{K_1(1)}K_1(\frac{r}{a}) & , \quad r > a \end{cases} \tag{4.29}$$

The modon radius is set at $a = 75km$, and a constant depth of $4km$ is used. A periodic basin of length $750km \times 750km$ is centered at a latitude of $\phi = 38.5°$. The resulting beta plane Coriolis terms are $f_0 = 2\Omega\sin(\phi) = 9.0887 \times 10^{-5} sec^{-1}$ and $\beta = 2\frac{\Omega}{R}\cos(\phi) = 1.8753 \times 10^{-8} sec^{-1} km^{-1}$.

Leapfrog time integration advances the solutions from the initial state at day 0 to day 30. Non-AMR simulations perform a mixing step every 29^{th} step. The regrid interval for AMR runs is also set to 29 level-0 steps. This value is larger than used for the advection case since the modon solution moves more slowly. Refinement regions retain their initial size, and change position at regrids according to the analytic modon velocity. The time step, Δt, for equations (4.17)-(4.19) using leapfrog time integration is given by

$$\Delta t = \frac{1}{2}\frac{\mu}{\sqrt{f_{max}^2 + (c+\sqrt{gH})^2(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2})}} \tag{4.30}$$

where the parameter μ is set to 0.90. The variable η is filtered after each step using the



(a) Day 0

(b) Day 30

Figure 4.4: u (top) and v (bottom) for the barotropic modon at day 0 and day 30 using AMR with leapfrog time integration and a staggered grid.

checkerboard filter (2.218). Levels finer than level-0 are smoothed at the interfaces by applying the interface smoother (4.13) to the outer 2 rows of cells on T patches.

The solutions for u and v at the beginning and end of the AMR simulation are shown in Figure 4.4. The errors of u and v are computed against the analytic solutions given by (4.28) and (4.29). The total error is defined as the RMS of the velocity error vector

$$RMS = \frac{\left\| \sqrt{(u-u_0)^2 + (v-v_0)^2} \right\|}{\left\| \sqrt{u_0^2 + v_0^2} \right\|} \tag{4.31}$$

The reference solutions in this case are only the exact solutions for the barotropic modon on an infinite domain with a rigid lid. However, the model domain used here is finite and periodic, but 10 times larger than the modon radius. The effects of the free surface and the periodic domain on RMS errors are found to be small at the fine resolution. This suggests the model solutions converge close to the analytic solutions, which therefore serve as good reference solutions. This was also noted by [65].

Figure 4.5a shows the errors for coarse (60×40), fine (540×360), and AMR runs. The errors are similar to those obtained with the AMR shallow water model of [11] in which the vorticity and stream function are directly computed. Similar results were found by [65] for the barotropic modon test in a non-moving nested grid model. In addition, the AMR model reduced run time by a factor of 15 from that of a globally fine grid . The RMS error slightly increases from 1.40% for the fine grid to 1.78% with the AMR grid.

Figure 4.5b shows that a similar error to the AMR case can be achieved near day 30 using a resolution of (480×320) without AMR. Such a resolution has a longer run time than AMR by a factor of 10. On the other hand, the highest resolution which will give a run time comparable to the AMR case is (255×170). The error using this resolution is a

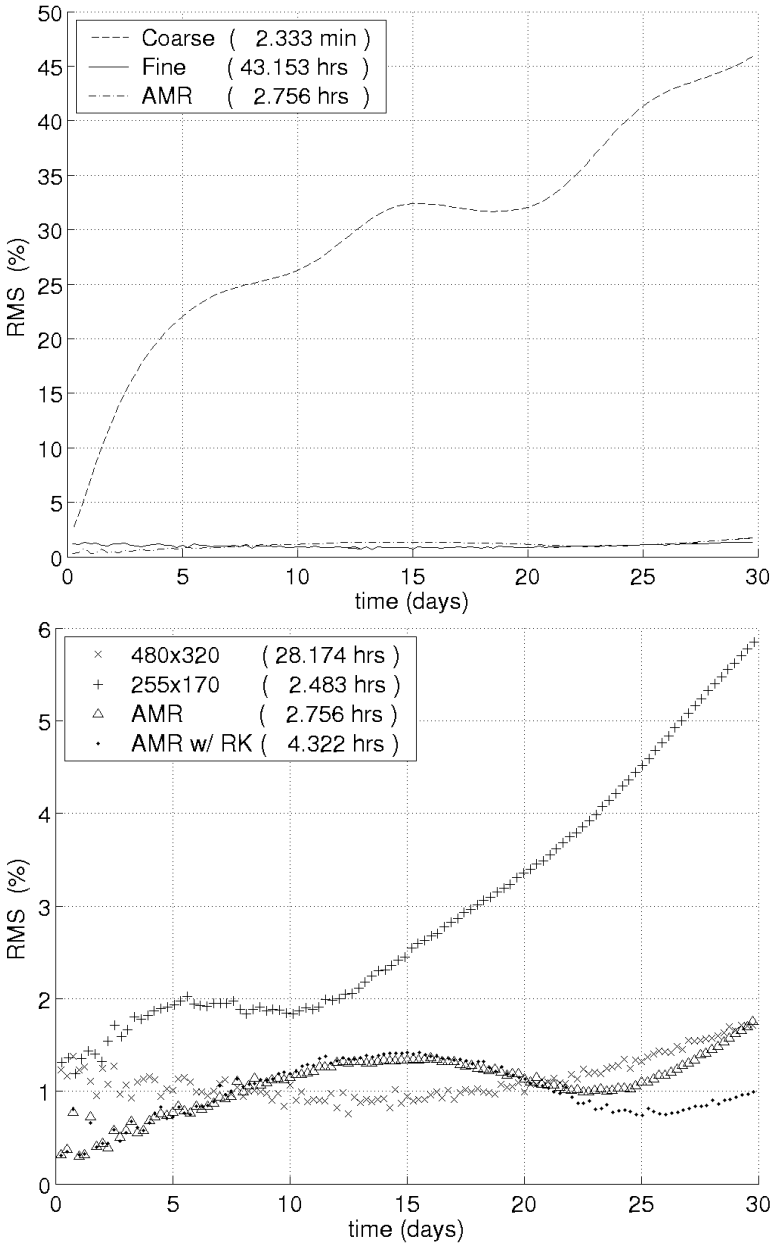


Figure 4.5: RMS error plots and run times for the barotropic modon tests. (Top) Comparison between the AMR simulation and the two non-AMR simulations with resolution equal to the level-0 (coarse) and level-2 (fine) of the AMR simulation. (Bottom) Comparison between the leapfrog simulation, the Runge-Kutta simulation, and the two non-AMR simulations with comparable error (480×320) and similar run time (255×170). Leapfrog AMR yields a small RMS error with the advantage of a short run time.

factor of 3.3 higher than the AMR case at day 30. A final AMR test with Runge Kutta time integration produces a similar RMS error to leapfrog AMR, but at the expense of a run time 1.6 times longer. Thus, leapfrog AMR yields a small RMS error with the advantage of a short run time for the shallow water equations.

4.4 Limited Basin With Flat Topography

Benchmark tests of ocean models are often performed on a limited basin. Basic geophysical flows are obtained with run times that are orders of magnitude shorter than those of full scale global ocean simulations. A limited basin is used here to test accuracy of the AMR Ocean Model with increased refinement based on various criteria. The model domain is 63 degrees in longitudinal width and spans from 20 degrees north to 62 degrees north in latitude. Both basin test cases compare non-AMR simulations to tests with uniform horizontal resolution equal to $3^\circ \times 3^\circ$ (coarse), $1^\circ \times 1^\circ$ (standard), and $\frac{1}{3}^\circ \times \frac{1}{3}^\circ$ (fine). AMR simulations use the coarse and standard horizontal resolutions for level-(0) and level-(1), respectively, making the refinement ratio between the levels equal to 3 in each spatial

Layer Number	Bottom of Layer Depth (m)	Layer Thickness (m)
1	100.0	100.0
2	347.14	247.14
3	867.01	519.87
4	1760.49	893.48
5	2999.99	1239.5

Table 4.2: Depth layers for the limited basin with flat topography.

Resolution	Δt^{BC} (minutes)	Δt^{BT} (minutes)	Subcycles ($\Delta t^{\mathrm{BC}}/\Delta t^{\mathrm{BT}}$)
level-(0), coarse	756	6	126
level-(1), standard	84	2	42
fine	9.33333	0.666666	14

Table 4.3: Time steps used for the limited basin with flat topography.

direction. Depth is discretized for all simulations with 5 layers of variable thickness shown in Table 4.2. The ocean floor topography is flat giving the basin a constant 3000 meter depth.

Diffusion is modeled with horizontal and vertical coefficients equal to 2×10^7 $\frac{cm^2}{s}$ and 0.2 $\frac{cm^2}{s}$, respectively. The horizontal and vertical viscosity coefficients are 10^9 $\frac{cm^2}{s}$ and 20 $\frac{cm^2}{s}$, respectively. The Coriolis, vertical diffusion, and vertical viscosity terms are modeled implicitly ($icor = idiff = ivisc = 1$). The interface smoother in equation (4.13) is applied to the AMR interfaces of the surface height field.

The time steps shown in Table 4.3 are calculated using equations (2.210)-(2.211), and then adjusted such that the AMR subcycling rule (3.26) is satisfied. The time refinement ratios for the AMR simulations are 3 for the barotropic system, and 9 for the baroclinic and tracer systems. A regrid frequency of 3 level-(0) steps enables regions of refinement to react to frequent changes in fluid motion dominated by the barotropic mode. This forces a mixing step on every 27^{th} level-(1) step. Non-AMR simulations perform mixing steps every 27 steps as well.

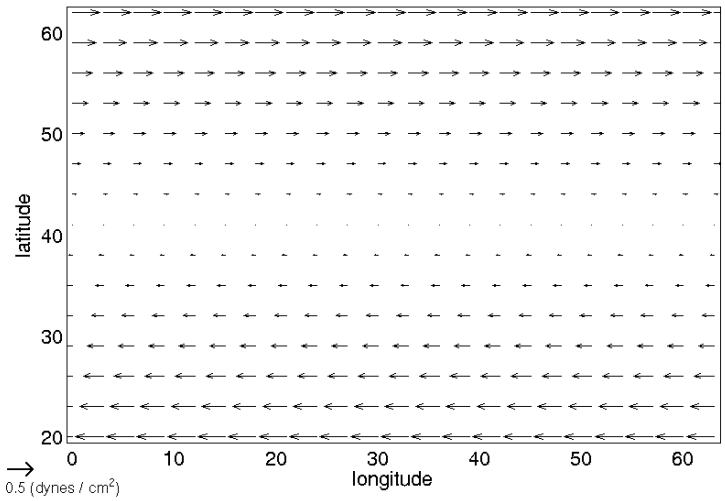


Figure 4.6: Wind stress profile for the single-gyre tests.

4.4.1 Single-Gyre Circulation

The tests presented here simulate a clockwise circulating current with strongest intensity along the western boundary of the basin. They will be referenced as the "single-gyre" tests. Western boundary currents in ocean basins have been studied for over half of a century [67, 48]. Simplified analytical models are shown to produce such currents at steady state when a latitudinally varied zonal wind stress is applied to the surface [73, 31]. Western boundary currents are also obtained with numerical ocean models using similar surface boundary conditions [78].

The single-gyre tests initialize the ocean basin at rest with temperature and salinity set equal to the constants 3.5 °C and 35.0 psu, respectively. These tracer values are retained throughout the simulation to maintain a constant basin density. The effects due to spatial variations in density are removed as a result, and the time scale necessary to reach a steady state is reduced. The applied surface wind stress (see Figure 4.6) has zero meridional

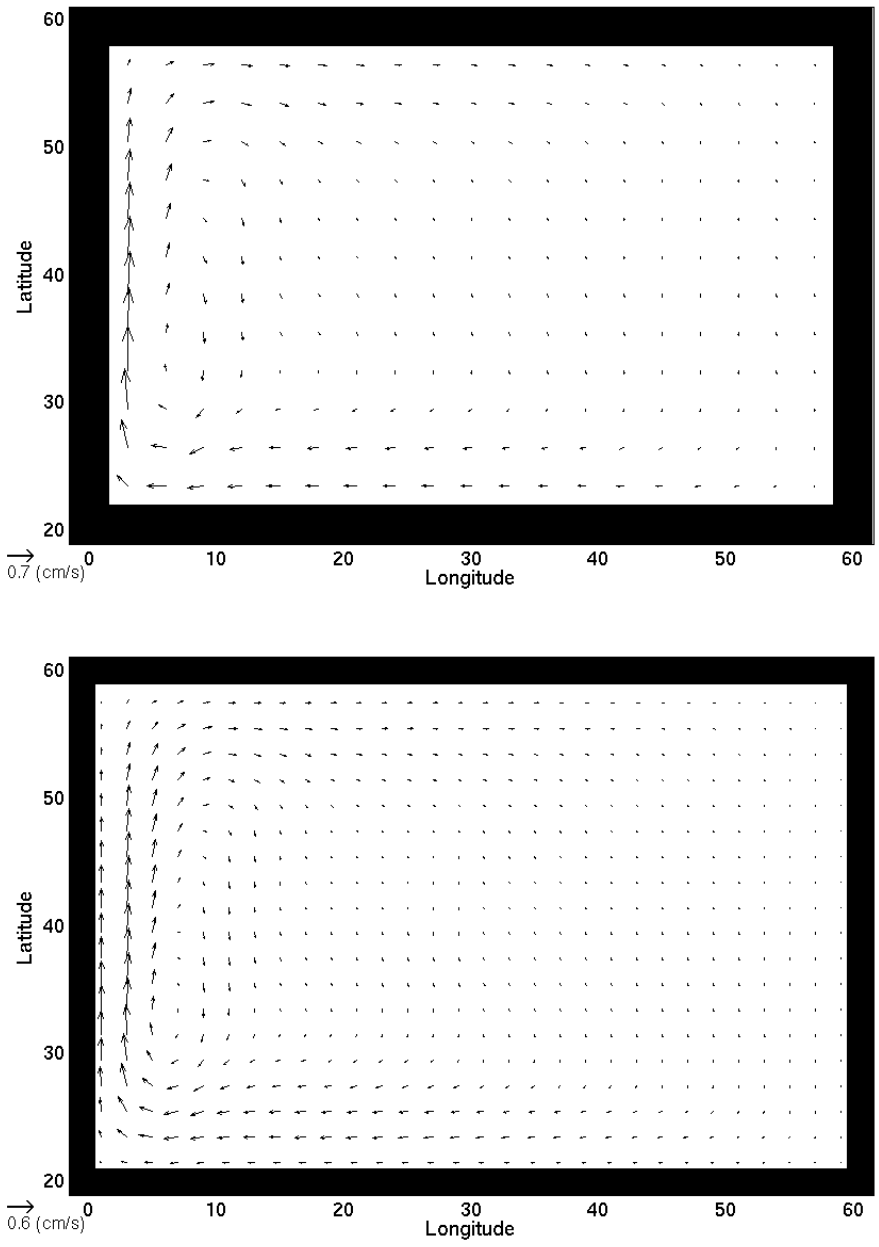


Figure 4.7: Single-gyre final state barotropic circulation for coarse (top) and standard (bottom) resolutions.

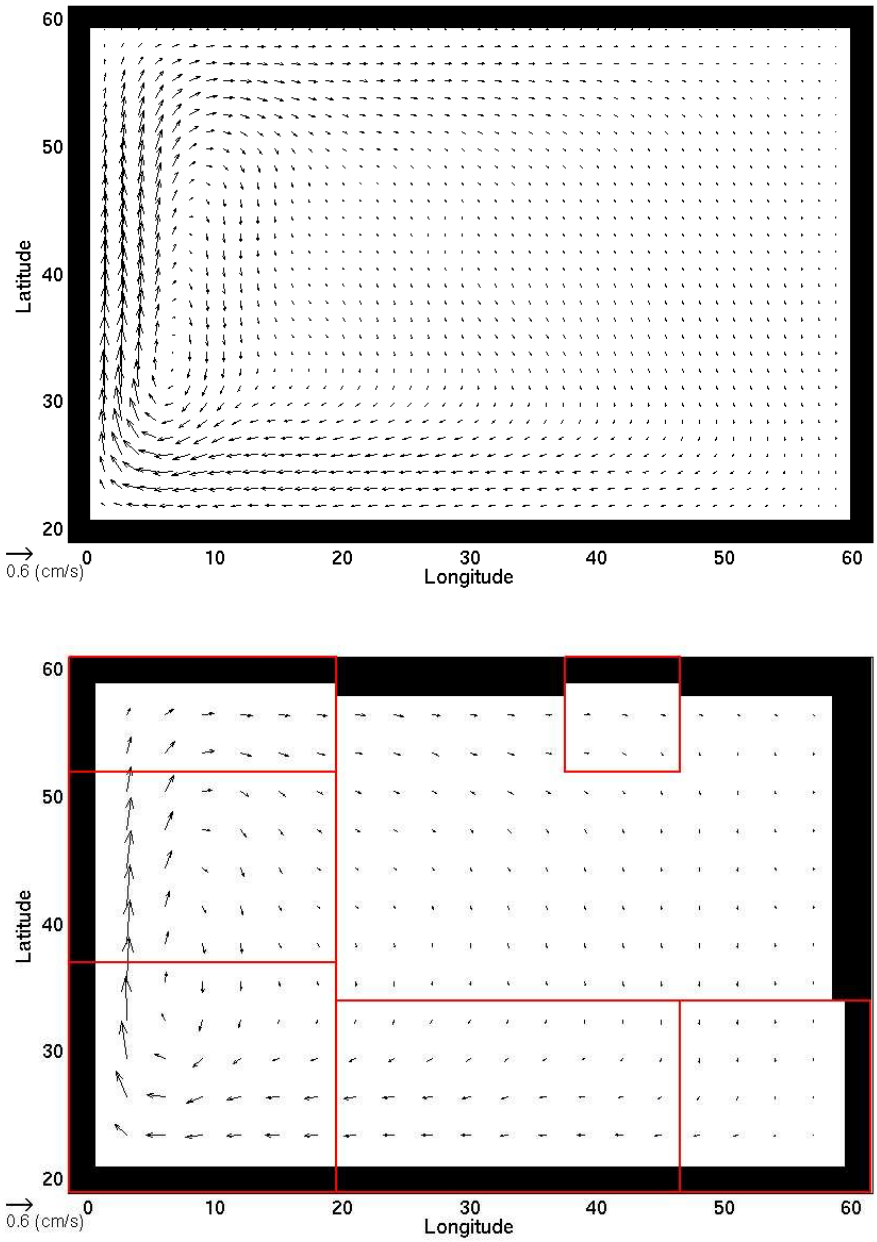


Figure 4.8: (top) Single-gyre final state barotropic circulation for the fine resolution. (bottom) An example of the same circulation using AMR where regrids perform the BVG tag calculation and tag the top 30% of level-(0) cells. Circulation is shown on level-(0) and red boxes indicate the edges of level-(1) refinement patches.

component, and has magnitude varying linearly with latitude from -0.5 dynes/cm^2 to 0.5 dynes/cm^2.

AMR tests define their refinement criteria using one of three tag calculations: barotropic velocity gradient (BVG), barotropic vorticity magnitude (BVM), and surface height gradient (SHG). Runs are performed for each AMR criterion using the top 10%, 30%, and 50% of the tag calculation as explained in Section 3.4.1. The locations and sizes of level-(1) refinement patches are determined by the clustering algorithm within SAMRAI which is dependent on the number of cells tagged for refinement. This parameter is therefore used to control the amount of increased refinement leaving the actual percentage of domain refinement up to SAMRAI.

The single-gyre circulation at day 150 is shown in Figure 4.7 and Figure 4.8. The maximum velocity magnitude of the coarse resolution is slightly higher than that for the standard, fine, and AMR cases. Boundary layer refinement as described in Section 3.3.1 is seen in the AMR case. The circulation generated with AMR has similar characteristics as those generated without AMR. The AMR simulations not shown in Figure 4.8 also produce similar circulation patterns.

A comparison of kinetic energy density as a function of time obtained with and without AMR is shown in Figure 4.9. The non-AMR runs have little variation in time after day 100 indicating a steady state. AMR simulations oscillate due to frequent regrids of the AMR hierarchy. The time averaged and standard deviation of kinetic energy density after day 100 are shown in Table 4.4. The BVM runs have mean values after day 100 that are closer to the standard mean than the BVG and SHG compared with equal amounts of

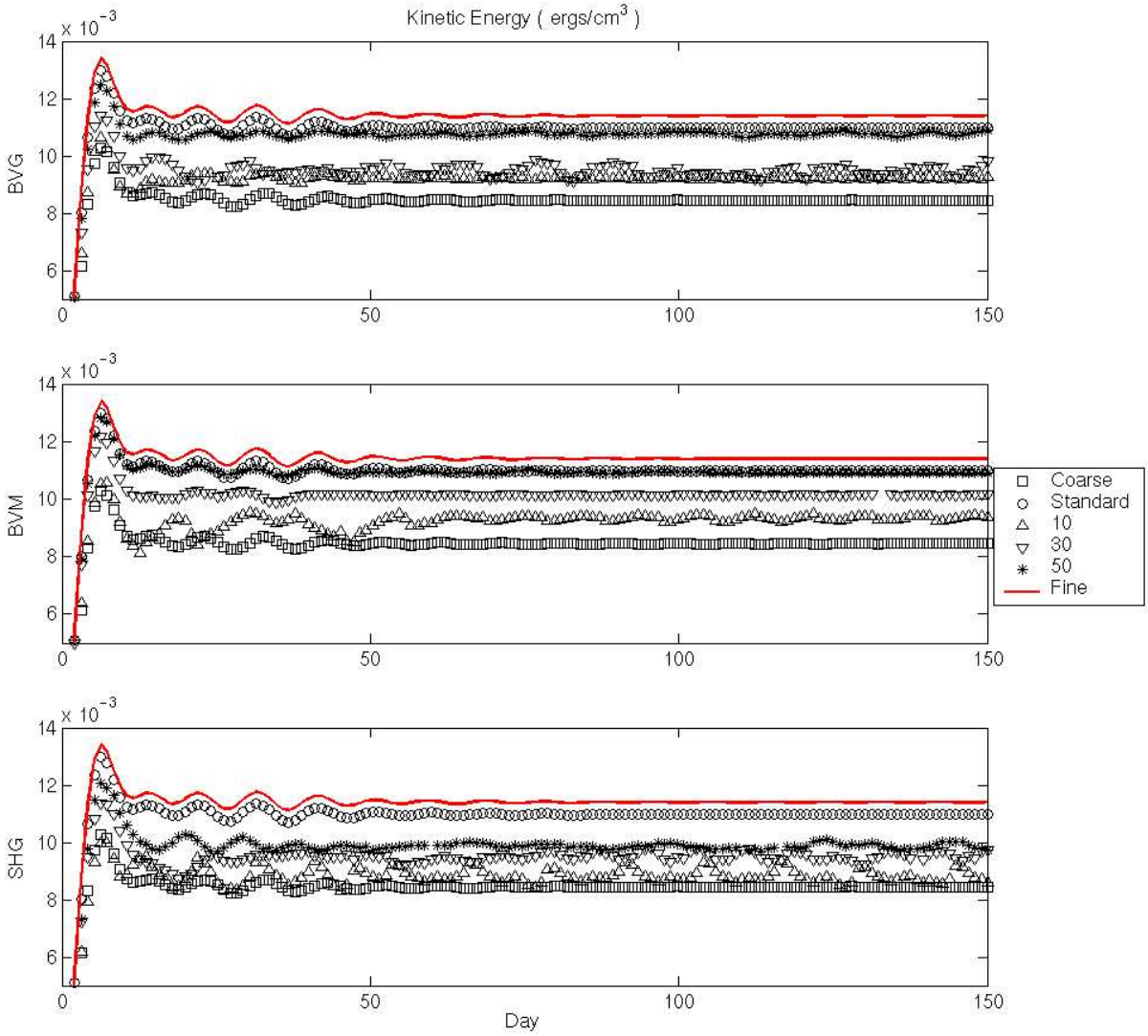


Figure 4.9: Kinetic energy density verses time for the single-gyre tests. The graphs compare refinement criteria based on barotropic velocity gradients (top), barotropic vorticity magnitude (middle), and surface height gradients (bottom) to coarse, standard, and fine resolutions.

Resolution	Mean KE density ($\times 10^{-3}$ ergs/cm^3)	Standard Deviation
Coarse	8.444	0.004
Standard	10.977	0.004
Fine	11.395	0.004
BVG 10%	9.227	0.029
BVG 30%	9.436	0.150
BVG 50%	10.763	0.061
BVM 10%	9.349	0.077
BVM 30%	10.144	0.005
BVM 50%	10.916	0.015
SHG 10%	8.823	0.216
SHG 30%	9.525	0.142
SHG 50%	9.866	0.102

Table 4.4: The mean and standard deviation of kinetic energy density after day 100 for the single-gyre tests.

tagging. The standard deviation represents a measure of the oscillations in kinetic energy density about the mean. The non-AMR simulations oscillate by at least an order of magnitude less than the AMR simulations. The BVM criterion has a lesser degree of oscillation than the BVG or SHG criteria when tagging 30% and 50% of the domain.

The coarse and standard velocity errors are calculated by comparing each resolution's velocity to the fine grid velocity averaged to the coarse and standard resolutions, respectively. Errors in AMR simulations are calculated by averaging the fine resolution data to the level-(0) resolution for level-(0) data, and to the level-(1) resolution for level-(1) data. The total velocity error is defined as the RMS of the velocity error vector,

$$RMS \;=\; \frac{\left\| \sqrt{(u-\overline{u})^2 \;+\; (v-\overline{v})^2} \right\|}{\left\| \sqrt{\overline{u}^2 \;+\; \overline{v}^2} \right\|}\,, \tag{4.32}$$

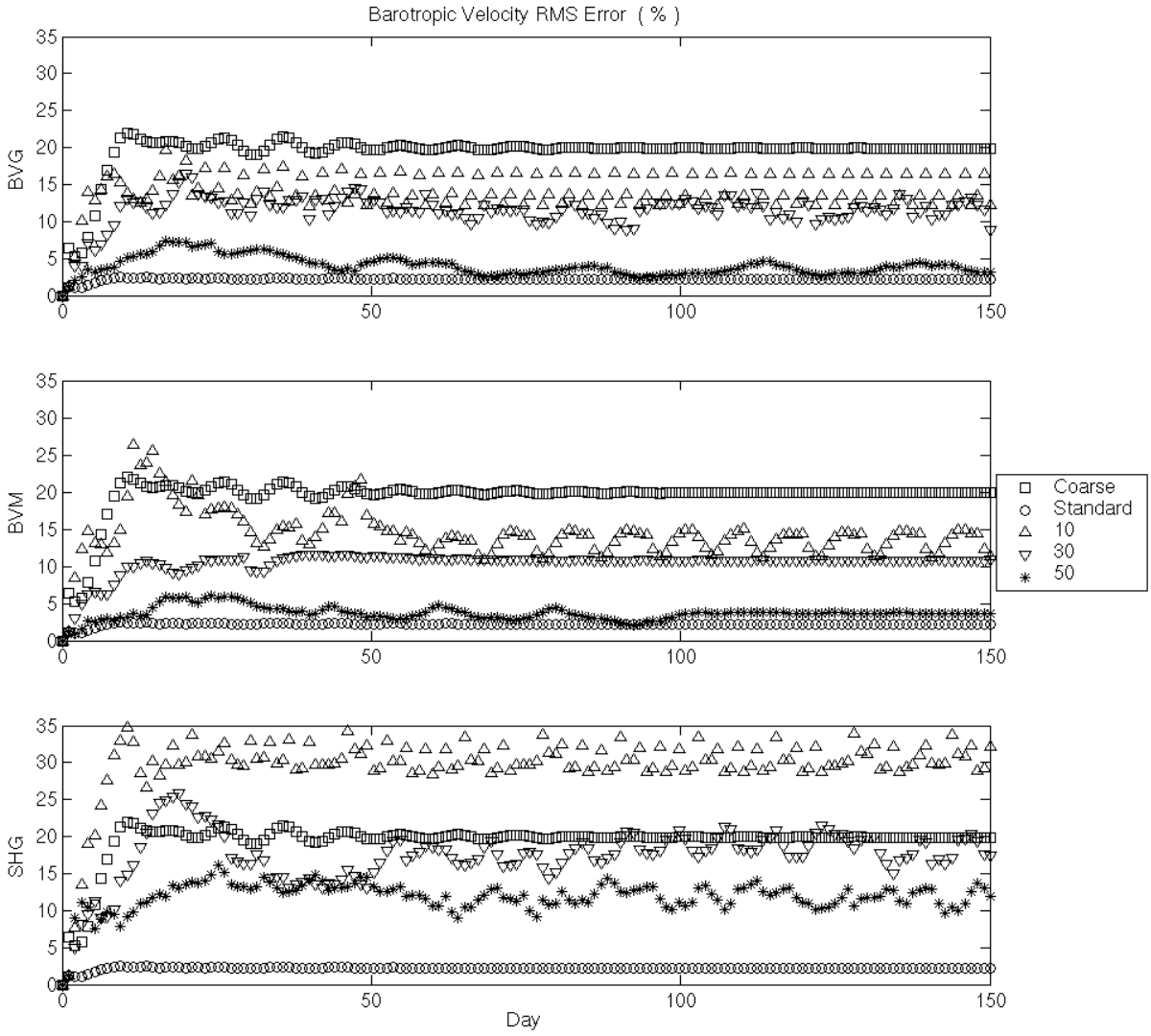


Figure 4.10: Barotropic velocity error verses time for the single-gyre tests.

Resolution	Mean BT Velocity Error (%)	Standard Deviation
Coarse	19.92	0.03
Standard	2.22	0.004
BVG 10%	14.13	1.77
BVG 30%	11.64	1.13
BVG 50%	3.54	0.55
BVM 10%	13.38	1.29
BVM 30%	10.77	0.06
BVM 50%	3.72	0.04
SHG 10%	30.56	1.51
SHG 30%	18.68	1.55
SHG 50%	11.85	1.12

Table 4.5: The mean and standard deviation of the barotropic velocity error after day 100 for the single-gyre tests.

where $\overline{u}$ and $\overline{v}$ are the averaged fine resolution velocity components, and $\|q\|$ is the 2-norm of the quantity q.

The time evolution of barotropic velocity errors is shown in Figure 4.10. Each graph compares the errors produced by AMR with a specific refinement criterion against the coarse and standard errors. Oscillations are again seen in AMR runs making it more useful to analyze the mean and standard deviation of the errors (see Table 4.5). The oscillations represented by standard deviations are consistently reduced for BVG and BVM runs as refinement is increased. BVM runs give the least amount of oscillation of the three refinement criteria compared with equal amounts of tagging. Each refinement criterion approaches the mean error of the standard mean error as refinement is increased. The SHG runs are the most oscillatory and have mean errors the farthest from the standard mean

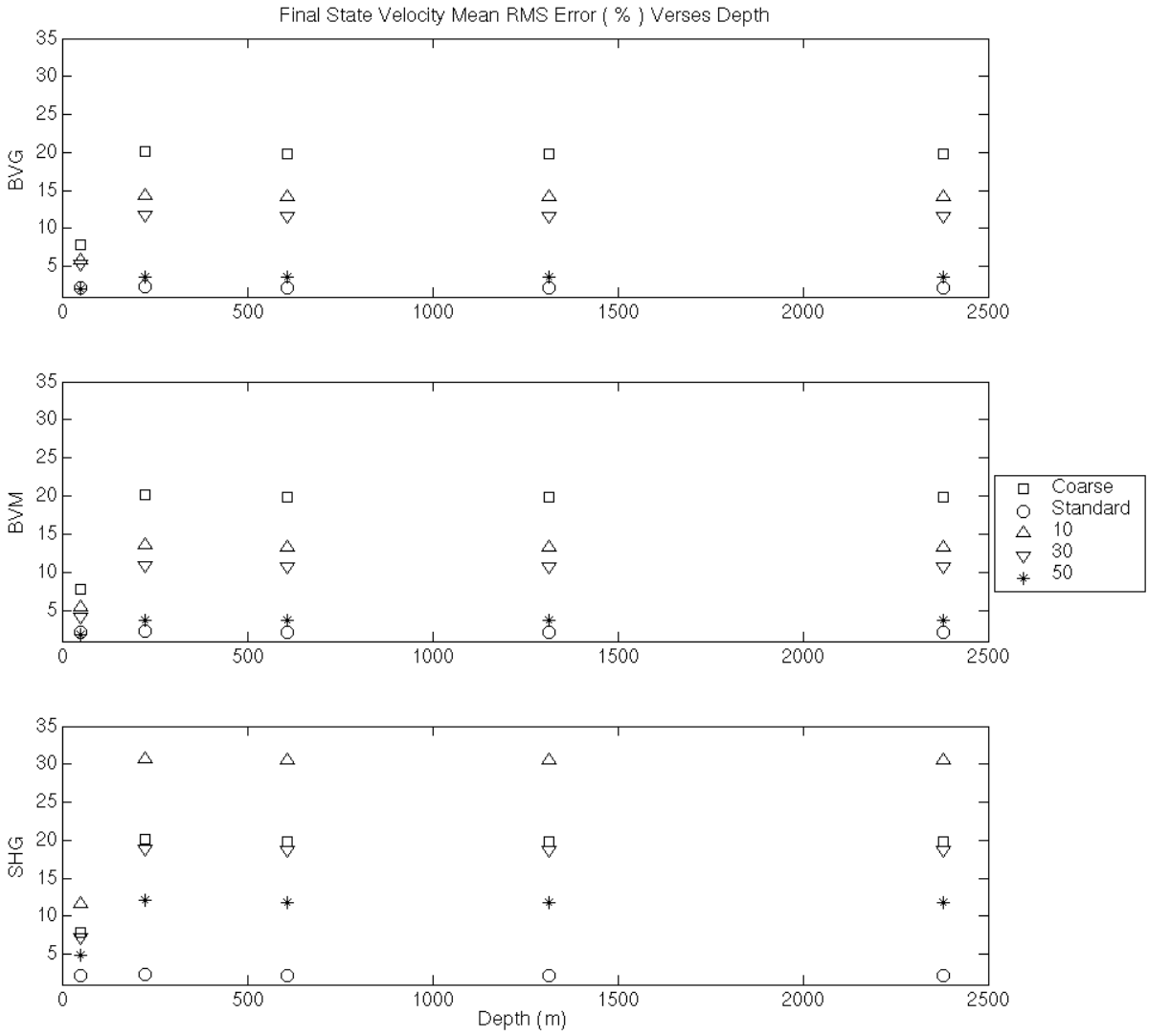


Figure 4.11: Mean velocity error after day 100 verses depth for the single-gyre tests.

error when compared to the BVG and BVM runs with equal amounts of tagging.

The mean error in the total horizontal velocity field after day 100 is shown in Figure 4.11 as a function of depth. The standard deviations are nearly the same as seen in the barotropic velocity and are therefore not shown for clarity. Mean velocity errors are smallest near the surface and increase with depth. AMR mean errors approach the standard mean errors as tagging is increased. The BVG and BVM mean errors are nearly identical and always less than the SHG mean errors when compared with equal amounts of tagging.

The integrated northward barotropic transport expressed in Sverdrups ($1Sv = 10^6 \frac{m^3}{s}$) is defined here as

$$\psi_{i_t} \quad = \quad 10^{-6} \; R\Delta\lambda \; \sum_{j_t} \cos\phi_{j_t} \overline{V_{i_u,j_u}}^{\lambda\phi} \; , \tag{4.33}$$

where the average operation and indexing are as defined in Section 2.6.2 and Section 3.3, respectively. The transport is defined at T cell centers to remove ambiguity associated with the boundary layer. Comparisons of transport shown in Figure 4.12 are performed at the coarse resolution to remove the bias from the $\Delta\lambda$ dependency. The fine resolution transport is not shown for clarity. The BVG and BVM runs produce transports close to the standard values near the western boundary. The transports from the SHG run at 10% deviate the most from the standard transports near the western boundary of all the AMR runs.

Surface height fields of the single-gyre tests at day 150 are shown in Figure 4.13 and Figure 4.14. The coarse, standard, fine, and AMR surface heights have similar contours, but the magnitude of the AMR surface height is higher than seen in the non-AMR cases. The source of this increase is not known, but is believed to result from surface height smoothing at AMR interfaces. The numerics of the AMR Ocean Model uses only the derivatives

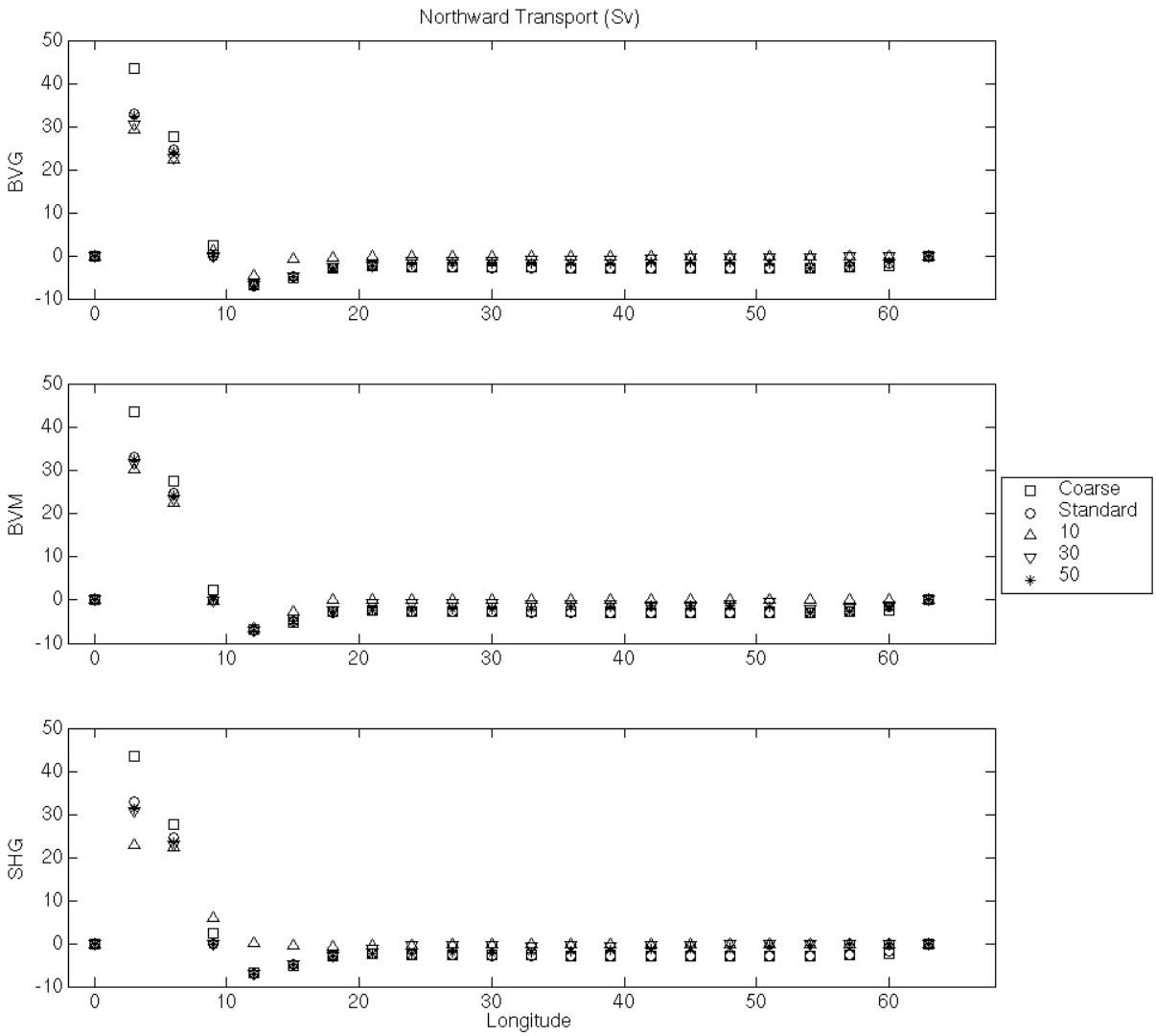


Figure 4.12: Mean integrated northward barotropic transport after day 100 for the single-gyre tests.

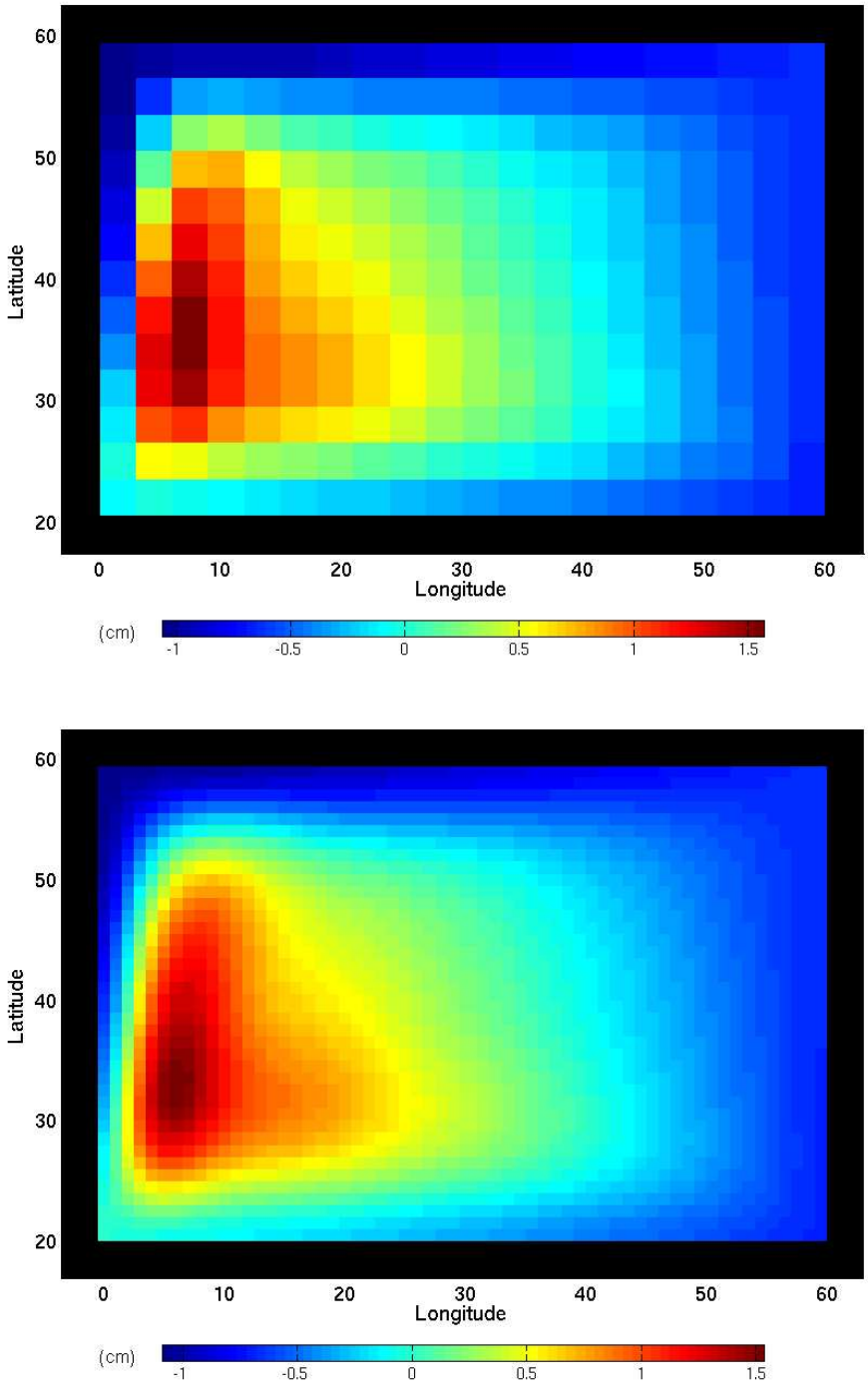


Figure 4.13: Surface height final state for the single-gyre coarse (top) and standard (bottom) resolutions.

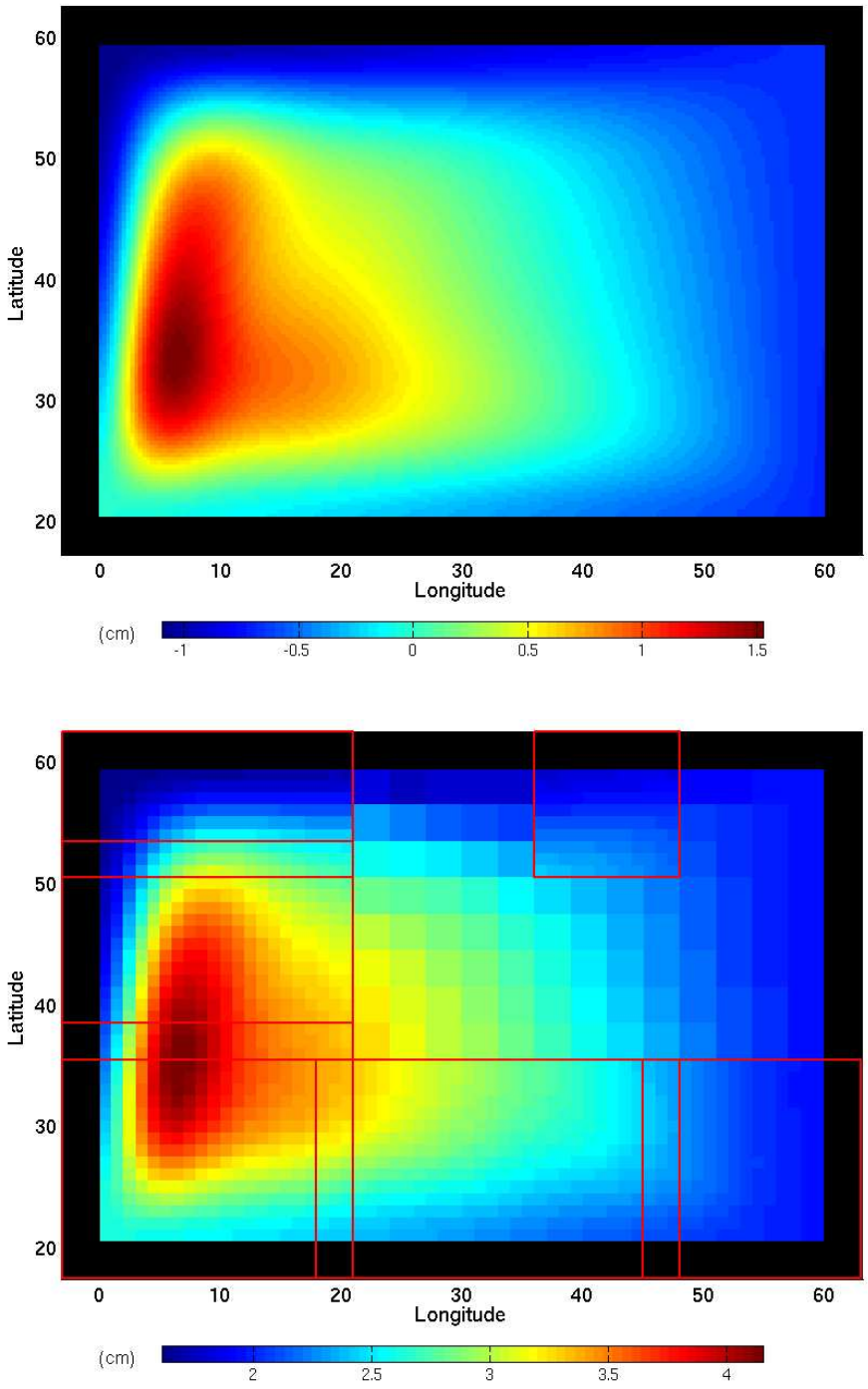


Figure 4.14: (top) Surface height final state for the single-gyre fine resolution. (bottom) An example of the surface height final state using AMR where regrids use the BVG tag calculation and tag the top 30% of level-(0) cells. Red boxes indicate the edges of level-(1) refinement patches.

of surface height (see Section 2.5). The increase in surface height for the AMR runs is therefore negligible provided the errors in the surface height gradient reduce as refinement is increased.

The surface height gradient,

$$\vec{\nabla}\eta_{i,j} \quad = \quad (\delta^{\lambda}\eta_{i,j})\ \hat{\lambda} \quad + \quad (\delta^{\phi}\eta_{i,j})\ \hat{\phi}\ , \tag{4.34}$$

is calculated for each test case where the quantities $\delta^{\lambda}\eta_{i,j}$ and $\delta^{\phi}\eta_{i,j}$ are defined using second order centered differences or one sided extrapolations.

$$\delta^{\lambda}\eta_{i,j} = \frac{1}{2\Delta\lambda} \begin{cases} \eta_{i+1,j} - \eta_{i-1,j} & , \quad M^T_{i+1,j}\ M^T_{i-1,j} > 0 \\ -3\eta_{i,j} + 4\eta_{i+1,j} - \eta_{i+2,j} & , \quad M^T_{i+1,j}\ (1 - M^T_{i-1,j}) > 0 \\ 3\eta_{i,j} - 4\eta_{i-1,j} + \eta_{i-2,j} & , \quad (1 - M^T_{i+1,j})\ M^T_{i-1,j} > 0 \end{cases} \tag{4.35}$$

$$\delta^{\phi}\eta_{i,j} = \frac{1}{2\Delta\phi} \begin{cases} \eta_{i,j+1} - \eta_{i,j-1} & , \quad M^T_{i,j+1}\ M^T_{i,j-1} > 0 \\ -3\eta_{i,j} + 4\eta_{i,j+1} - \eta_{i,j+2} & , \quad M^T_{i,j+1}\ (1 - M^T_{i,j-1}) > 0 \\ 3\eta_{i,j} - 4\eta_{i,j-1} + \eta_{i,j-2} & , \quad (1 - M^T_{i,j+1})\ M^T_{i,j-1} > 0 \end{cases} \tag{4.36}$$

The surface height gradient error calculations are similar to those performed for the velocity error. Each $\vec{\nabla}\eta_{i,j}$ computed for the coarse, standard, and AMR runs is compared to the fine grid surface height gradient averaged to the appropriate resolution. The total surface height gradient error is defined as the RMS of the error vector,

$$RMS \quad = \quad \frac{\left\| \sqrt{\left(\delta^{\lambda}\eta - \overline{\delta^{\lambda}\eta}\right)^2 \ + \ \left(\delta^{\phi}\eta - \overline{\delta^{\phi}\eta}\right)^2} \right\|}{\left\| \sqrt{\overline{\delta^{\lambda}\eta}^2 \ + \ \overline{\delta^{\phi}\eta}^2} \right\|}\ , \tag{4.37}$$

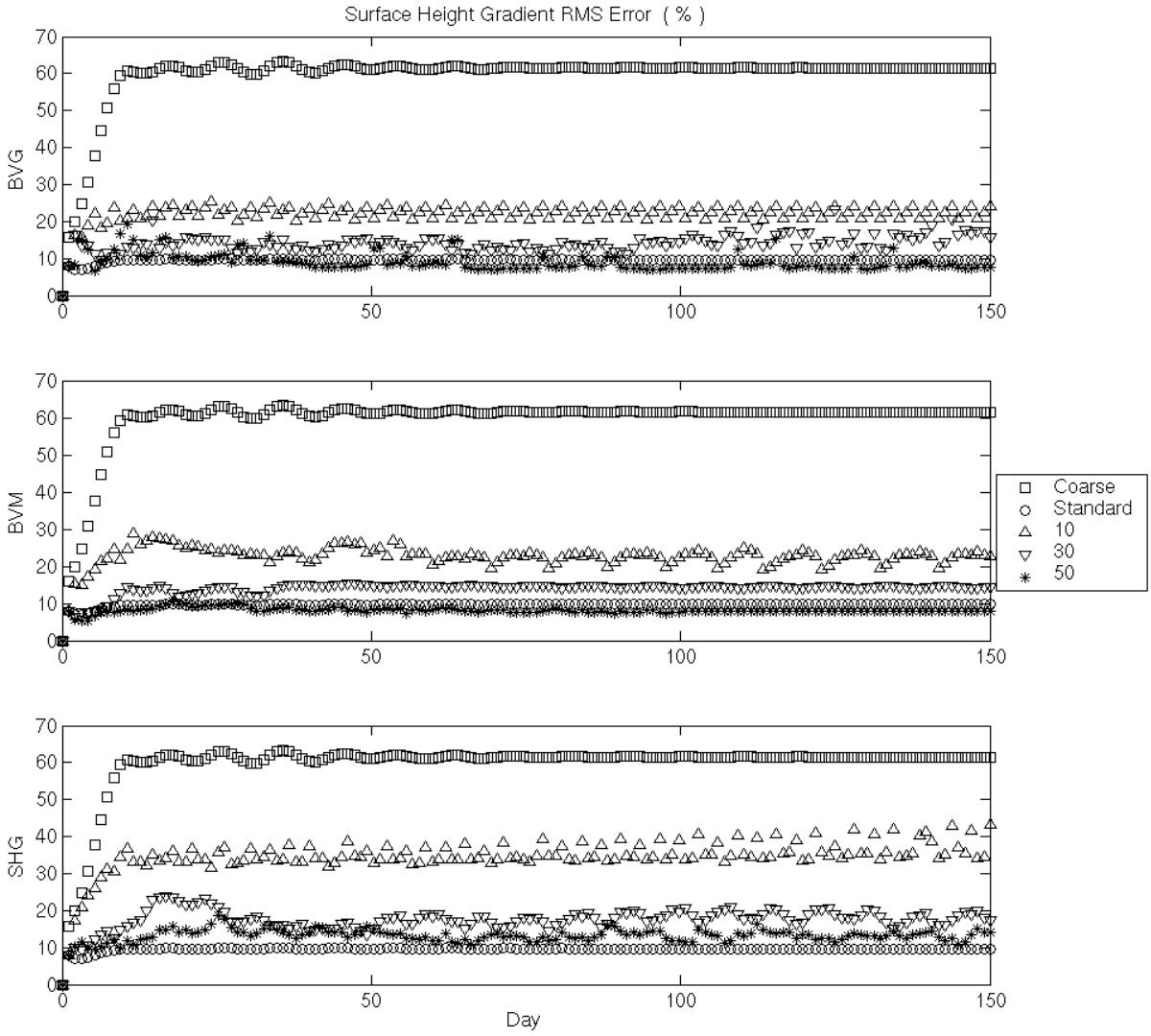


Figure 4.15: Surface height gradient error verses time for the single-gyre tests.

Resolution	Mean $\vec{\nabla}\eta$ Error (%)	Standard Deviation
Coarse	61.55	0.03
Standard	9.70	0.003
BVG 10%	22.47	1.34
BVG 30%	15.68	2.12
BVG 50%	8.12	1.58
BVM 10%	22.50	1.52
BVM 30%	14.42	0.20
BVM 50%	7.87	0.04
SHG 10%	36.89	3.00
SHG 30%	18.37	1.48
SHG 50%	13.20	1.09

Table 4.6: The mean and standard deviation of the surface height gradient error after day 100 for the single-gyre tests.

where $\overline{\delta^{\lambda}\eta}$ and $\overline{\delta^{\phi}\eta}$ are the averaged fine resolution gradient components.

The time evolution of surface height gradient errors is seen in Figure 4.15, and the mean and standard deviation of the errors after day 100 are shown in Table 4.6. The refinement criteria each show decreasing trends in the mean error as refinement is increased. Oscillations representations are again seen in AMR errors which consistently decrease for the BVM and SHG criteria as refinement is increased. The BVM run at 50% has a standard deviation on the same order of magnitude as the standard run.

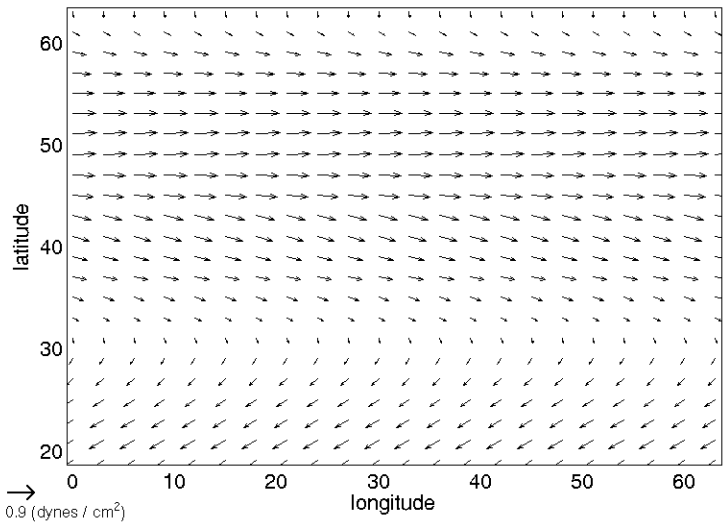


Figure 4.16: Wind stress profile for the double-gyre tests.

4.4.2 Double-Gyre Circulation

AMR accuracy is further tested using an alternate geostrophic flow within a limited basin. This is accomplished by repeating the single-gyre tests with a modified surface wind stress field shown in Figure 4.16. The new field results from taking the zonal and annual means of the Hellerman and Rosenstein data set [33, 32] on the domain defined from 71 degrees west to 7 degrees east and 19 degrees north to 63 degrees north. The zonal and meridional components vary with latitude, but no longitudinal variation is present as was the case in the single-gyre test. This forcing is known to produce a pair of counter circulating currents [78] and is therefore referenced here as the "double-gyre" test.

Simulations of the double-gyre circulation are performed using coarse, standard, fine, and AMR resolutions. Temperature and salinity are again held constant at 3.5 °C and 35.0 psu, respectively, in order to remove spatial variations in density. Refinement criteria is

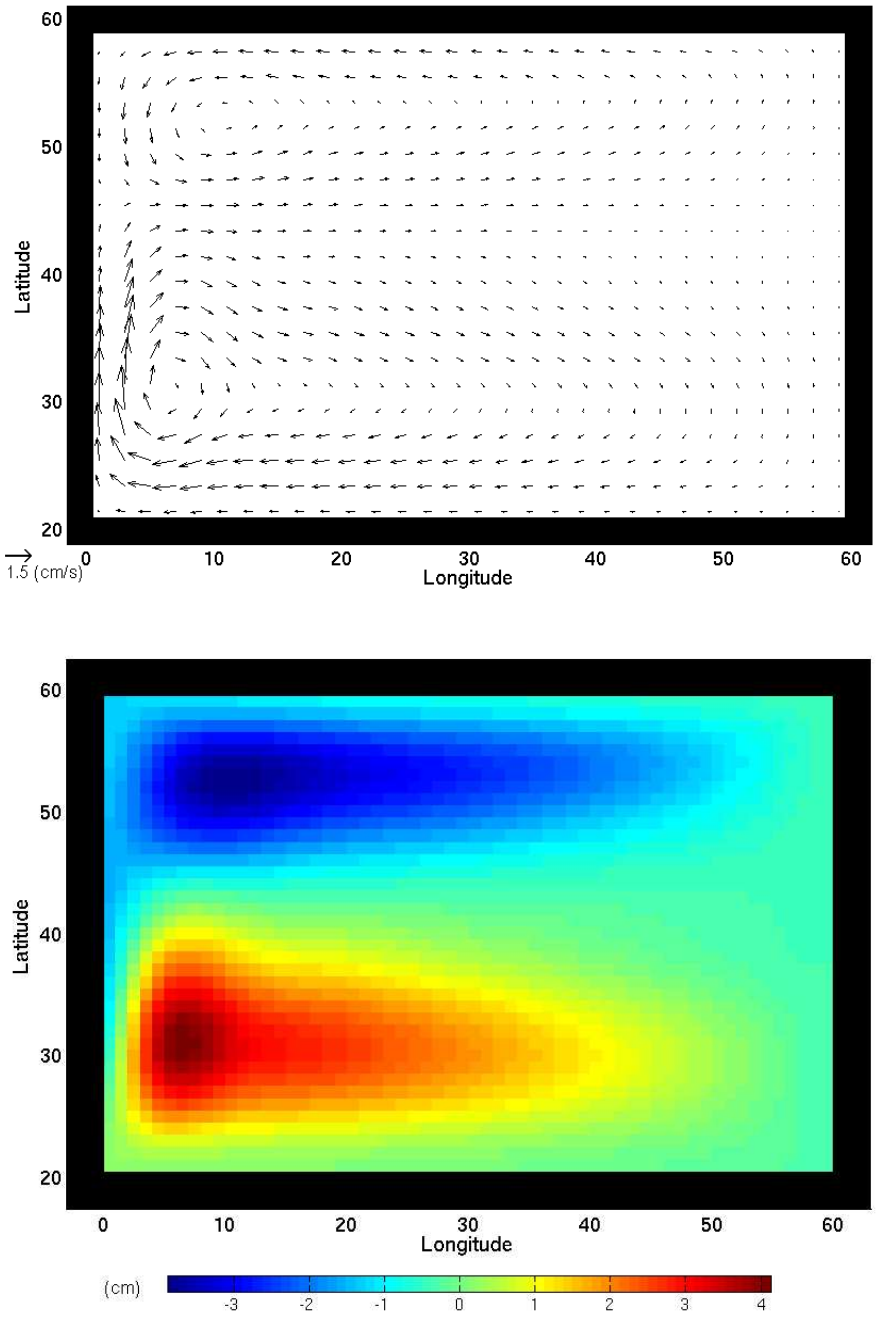


Figure 4.17: Double-gyre final state barotropic circulation (top) and surface height (bottom) for the standard resolution.

Resolution	Mean KE density ($\times 10^{-2}$ ergs/cm^3)	Standard Deviation
Coarse	4.297	0.001
Standard	5.884	0.001
Fine	6.093	0.001
BVG 10%	4.167	0.168
BVG 30%	4.723	0.083
BVG 50%	5.522	0.019
BVM 10%	4.113	0.107
BVM 30%	4.747	0.079
BVM 50%	5.680	0.056
SHG 10%	3.611	0.134
SHG 30%	4.096	0.185
SHG 50%	4.676	0.154

Table 4.7: The mean and standard deviation of kinetic energy density after day 100 for the double-gyre tests.

based on the barotropic velocity gradients (BVG), barotropic vorticity magnitude (BVM), or surface height gradients (SHG). Each criterion is tested using the top 10%, 30%, and 50% of the tag calculation.

The double-gyre barotropic circulation and surface height at day 150 are shown in Figure 4.17 for the standard resolution. A pair of currents are present with counter-clockwise circulation in the northern half of the basin and stronger clockwise circulation in the southern half of the basin. All other comments regarding the barotropic circulation and surface height final states for the coarse, fine, and AMR resolutions are the same as mentioned in single-gyre tests. These resolutions are therefore not shown for clarity.

The kinetic energy density plots are similar to those obtained in the single-gyre test and are not shown for clarity. The means and standard deviations of kinetic energy

Resolution	Mean BT Velocity Error (%)	Standard Deviation
Coarse	16.81	0.01
Standard	1.98	0.002
BVG 10%	22.56	2.35
BVG 30%	14.56	0.76
BVG 50%	6.81	0.29
BVM 10%	24.58	2.32
BVM 30%	14.74	1.18
BVM 50%	6.21	0.63
SHG 10%	42.79	1.20
SHG 30%	23.66	1.99
SHG 50%	21.50	2.14

Table 4.8: The mean and standard deviation of the barotropic velocity error after day 100 for the double-gyre tests.

density after day 100 are shown in Table 4.7. AMR runs approach the standard mean as refinement is increased. The BVG and BVM criteria show consistent decreases in oscillation with increased refinement.

The mean and standard deviation in barotropic velocity errors after day 100 are shown in Table 4.8. The majority of the runs have errors close to or greater than the coarse mean error, but increases in tagging lessen the error produced by each criterion. The standard deviations in the BVG and BVM runs decrease as refinement increases. The SHG mean errors are always greater than the coarse mean errors. The model is therefore more sensitive to the refinement criteria than was seen in the single-gyre tests. The same can be said of the error in the final state velocities as a function of depth shown in Figure 4.18.

The integrated northward transports are shown in Figure 4.19. The double-gyre

Resolution	Mean $\vec{\nabla}\eta$ Error (%)	Standard Deviation
Coarse	50.35	0.01
Standard	8.18	0.002
BVG 10%	25.93	2.37
BVG 30%	15.08	0.96
BVG 50%	10.25	0.37
BVM 10%	34.26	1.77
BVM 30%	19.87	2.30
BVM 50%	11.02	0.99
SHG 10%	38.87	1.63
SHG 30%	24.36	2.24
SHG 50%	21.43	2.09

Table 4.9: The mean and standard deviation of the surface height gradient error after day 100 for the double-gyre tests.

transports are greater in magnitude than those in the single-gyre tests due to stronger surface wind stresses. The BVG and BVM runs produce transports close to the standard values near the western boundary. The SHG run at 10% has values farthest from the standard near the western boundary.

The temporal mean and standard deviation of surface height gradient error after day 100 are given in Table 4.9. AMR runs produce errors less than the coarse and become closer to the standard as refinement is increased. The BVG criterion gives the lowest AMR errors when compared to BVM and SHG using equal amounts of tagging. Oscillations consistently decrease for the BVG runs as refinement is increased, but not for the BVM or SHG runs.

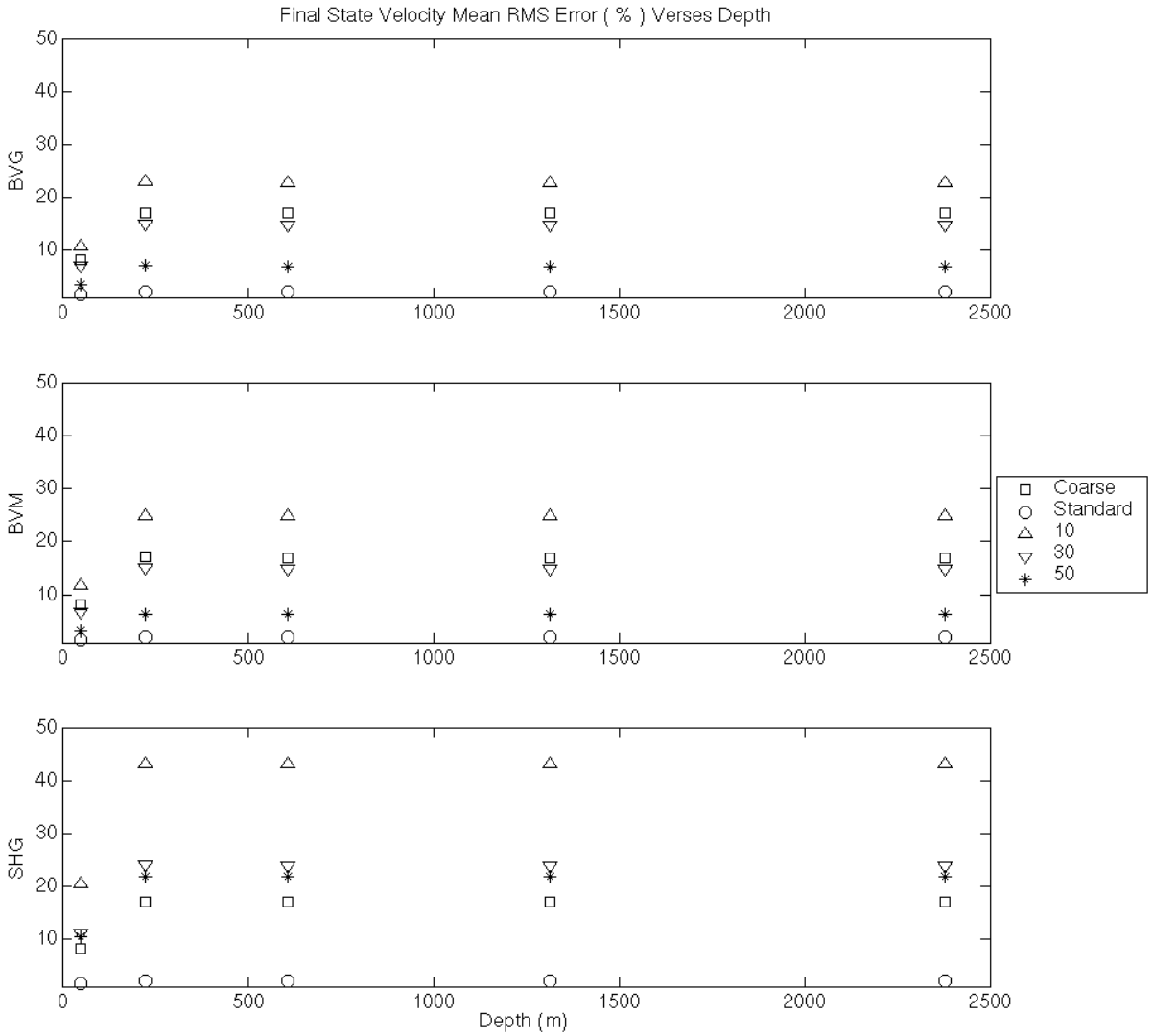


Figure 4.18: Final state velocity error verses depth for the double-gyre tests.

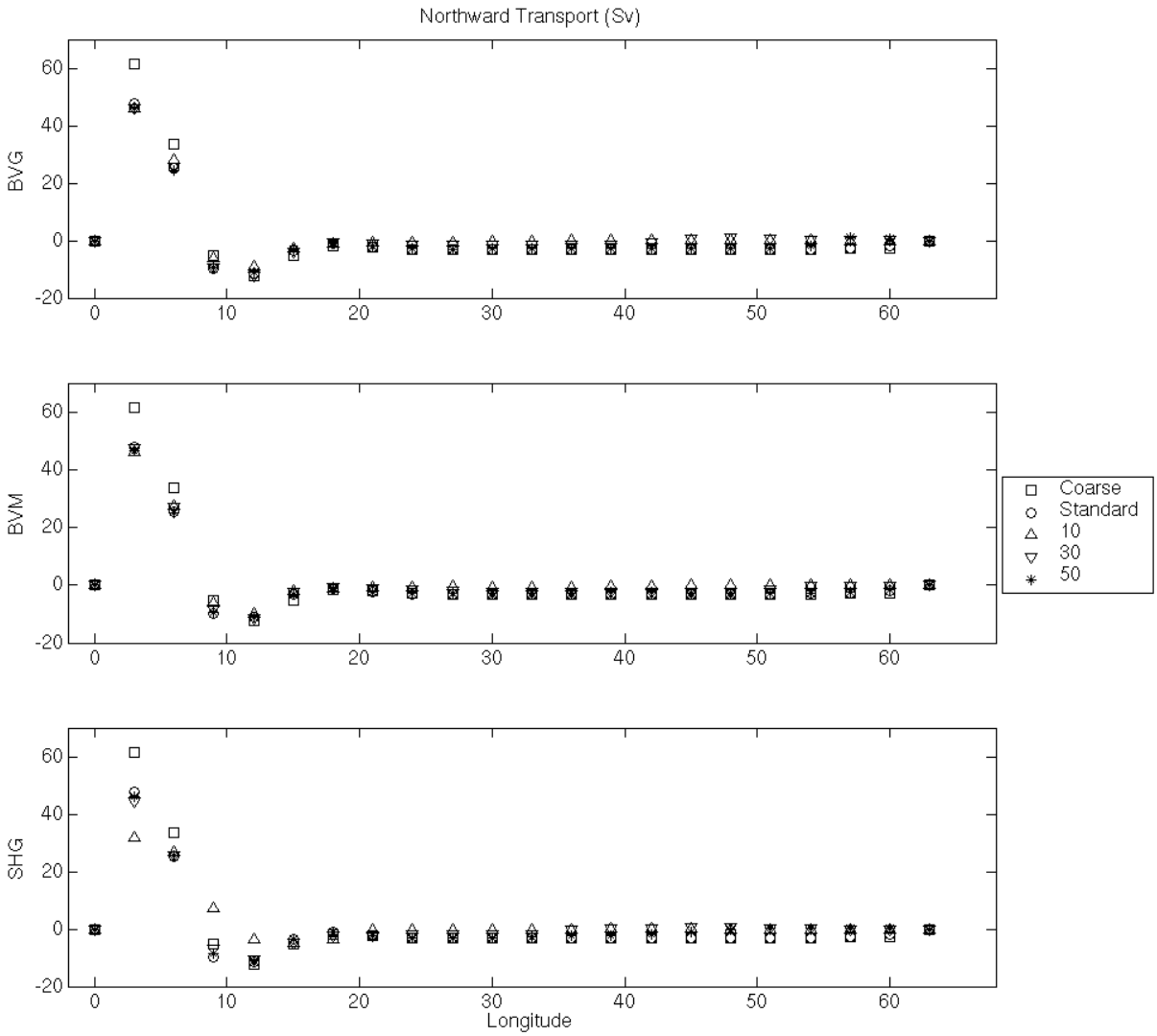


Figure 4.19: Integrated northward barotropic transport of the final state double-gyre tests.

4.5 Summary of Test Cases

The 2D advection with dynamic refinement test demonstrates that AMR with leapfrog time integration produces RMS errors that are of the same order of magnitude as single and multi-step time integrators. The run time using leapfrog is shorter than that using Godunov. Leapfrog and Runge Kutta run times are of the same order. The second advection test demonstrates the effects of leapfrog time integration when advancing a solution through a coarse/fine interface. Leapfrog produces more interface noise than single or multi-step integrators. Introducing an interface smoother reduces effects of interface noise on solution error.

The barotropic modon test validates the B-grid AMR approach. The shallow water equations are discretized with velocities and surface height defined in U and T patches, respectively. The RMS errors of the AMR and fine grids were less than 2% after advancing a barotropic modon with leapfrog time integration for 30 simulated days. These results are similar to those of [11] and [65].

The limited basin tests demonstrate the AMR Ocean Model's ability to dynamically adapt refinement regions in response to the barotropic flow field. AMR with refinement based on velocity gradients or vorticity produce mean errors that are consistently reduced and approach the standard mean errors as refinement is increased. AMR errors oscillate more about the mean than non-AMR errors due to frequent regrids of the AMR hierarchy. Surface height fields take on unrealistic values in the presence of AMR, but gradients in surface height are improved with increased refinement. AMR produces velocity fields consistent with those produced without AMR. Mean velocity errors are reduced and approach

the standard mean errors as refinement is increased. AMR runs using barotropic vorticity magnitude and barotropic velocity gradient refinement criteria consistently produce smaller mean errors than the surface height gradient criterion. Refinement criteria dependent on the flow field is therefore recommended for AMR simulations of geophysical flows. The double-gyre tests require 50% of the domain to be tagged before mean velocity errors are close to the standard mean errors. This suggests large amounts of refinement might be necessary for AMR applied during the spin up phase of a simulation. It is speculated from these results that AMR is more suitable during spin up when the flow fields have more isolated features and/or different refinement criteria is used. However, further testing is recommended to before such arguments are confirmed.

Chapter 5

Simulations of the North Atlantic

5.1 Carbon Sequestration Model

Sequestering carbon dioxide by direct injection from near shore power plants into the ocean has been proposed for the past decade as a means of reducing man-made atmospheric emissions. The idea is based on the premise that such gases would eventually end up in the ocean through the planet's carbon cycle [16]. Power plant CO_2 emissions equilibrate with the ocean surface on a scale of one year or less. Hundreds of years pass before the CO_2 mixes into the deep ocean where it reacts with carbonate minerals on time scales of 1000's of years. The net reaction of CO_2 and ocean carbonates sediments such as limestone is given by

$$CO_2 \; + \; H_2O \; + \; CaCO_3 \;\; \longrightarrow \;\; Ca^{2+} \; + \; 2HCO_3^- \; . \tag{5.1}$$

The anthropogenic CO_2 is then essentially buried in the deep ocean as carbonate sediments.

Direct injection CO_2 sequestration would by pass the mixing phases of the natural

carbon cycle by harnessing emissions from near shore power plants and transporting them to the middepth and deep ocean through an elongated pipe [27, 52]. Global ocean models have been used to simulate the fate of CO_2 injected into the ocean at various locations. The models provide a means of estimating 1.) the amount of CO_2 escaping back to the atmosphere, and 2.) the resulting changes in ocean chemistry. Wickett et al. [76] illustrated the dependency on horizontal grid resolution when calculating pH changes near injection sites and suggested using localized resolution increases to improve calculations.

The perturbation of CO_2 concentration in the ocean due to sequestration is modeled in the AMR Ocean Model as a passive tracer, C, and is propagated using the tracer transport equation described in Section 2.6.3,

$$\frac{\partial C_{i,j,k}}{\partial t} = \left[-ADV(C_{i,j,k}) + DIFF^{\lambda\phi}(C_{i,j,k}) + DIFF^{z}(C_{i,j,k}) + S_{i,j,k} \right] M^{T}_{i,j,k} , \tag{5.2}$$

with an added source term, $S_{i,j,k}$. This approach assumes that C represents small deviations in the ocean's mean inorganic carbon concentration and that the outgassing of C has a negligible impact on atmospheric CO_2 [36]. The outgassing of CO_2 to the atmosphere is represented by the flux of C through the ocean surface and is modeled by a restoring term defined in the model by equation (2.97), where C^{surf} is set to zero and the restoring time scale, τ_C, is set to one year [36]. The change in atmospheric CO_2 concentration due to outgassing is not modeled.

It is assumed that CO_2 instantly and completely dissolves in seawater upon injection primarily by the reaction

$$CO_2 + H_2O \longrightarrow H^+ + HCO_3^- . \tag{5.3}$$

The dissociation of the reverse reaction or secondary reactions are not considered in this study. The pH change is calculated based on this reaction for each grid cell,

$$\Delta \mathrm{pH}_{i,j,k} \;=\; -\log\left(10^{-\widehat{\mathrm{pH}}_{i,j,k}} \;+\; C_{i,j,k}\right) \;-\; \widehat{\mathrm{pH}}_{i,j,k}\;, \tag{5.4}$$

where $C_{i,j,k}$ is expressed in units of (mol/L) and $\widehat{\mathrm{pH}}_{i,j,k}$ is the pH of the unperturbed grid cell. The latter parameter is estimated for this case study since the AMR Ocean Model does not model background free hydrogen. Future studies could also calculate pH changes relative to data set estimations of ocean carbon chemistry [55]. Given that equation (5.3) predicts an increase in acidity, the pH change calculation (equation 5.4) becomes an upper bound when $\widehat{\mathrm{pH}}_{i,j,k}$ is defined as the most basic pH naturally occurring in the ocean ($\widehat{\mathrm{pH}}_{i,j,k} \lesssim 8.2$). The pH changes reported in these CO_2 sequestration studies therefore serve as worst-case scenarios.

5.2 Spin Up Simulations

The model is advanced to a near equilibrium, or "spin up", state prior to simulating CO_2 direct injections. Such simulations are performed here in a closed basin characteristic of the North Atlantic. The horizontal domain spans from 100° west to 0° longitude, and 3° south to 65° north latitude. Two non-AMR simulations are performed with horizontal resolution equal to $4^\circ \times 2^\circ$ (coarse) and $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ (standard), respectively. The computational domain is much smaller than the global ocean which enables comparisons of AMR and standard simulation results. The tests presented in this study would gain increased credibility if comparisons were made against results from a $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ resolution. Such a resolution would require months of CPU time due to the parallel performance limitations discussed later in

Section 6.3, and is therefore not investigated.

Depth is discretized with 23 layers of increasing thickness shown in Table 5.1. Satellite measured topography data [63] are read from a file and spatially averaged to tracer cell centers. Single grid cell islands are given a depth of 51.25 meters and channels with width less than 3 tracer grid cells are given depth 0. A Gaussian smoother is applied 5 times before discretizing depth to the nearest depth layer bottom. Land masks and velocity cell topography are defined using the procedure described in Section 3.3.1. The coarse resolution topography shown in Figure 5.1a is used in all North Atlantic Basin simulations to prevent conservation problems associated with refining topography as discussed in Section 3.3.1.

5.2.1 Surface Forcing and Parameterization

Monthly surface wind stresses obtained from the Hellerman and Rosenstein data set [33] are read from files and spatially interpolated to velocity cell centers. The data are further interpolated in time prior to each baroclinic time step to remove temporal discontinuities. The annual mean of the surface wind stress is shown in Figure 5.1b. The same interpolation procedure is applied to the temperature and salinity restoring fields which are calculated from Levitus [45], and applied with a time scale of 30 days.

The horizontal and vertical momentum viscosity coefficients are 10^9 $\frac{cm^2}{s}$ and 20 $\frac{cm^2}{s}$, respectively. The horizontal tracer diffusion coefficient is 2×10^7 $\frac{cm^2}{s}$, and vertical tracer diffusion coefficients vary linearly with depth from 0.2 $\frac{cm^2}{s}$ at the surface to 1.3 $\frac{cm^2}{s}$ at the ocean floor. The Coriolis, vertical diffusion, and vertical viscosity terms are modeled implicitly ($icor = idiff = ivisc = 1$).

Layer Number	Bottom of Layer Depth (m)	Layer Thickness (m)
1	25.0	25.0
2	51.25	26.25
3	81.75	30.5
4	118.5	36.75
5	161.75	43.25
6	212.5	50.75
7	275.0	62.5
8	360.0	85.0
9	377.5	117.5
10	627.5	150.0
11	807.5	180.0
12	1017.5	210.0
13	1262.5	245.0
14	1557.5	295.0
15	1922.5	365.0
16	2350.0	427.5
17	2800.0	450.0
18	3250.0	450.0
19	3700.0	450.0
20	4150.0	450.0
21	4600.0	450.0
22	5050.0	450.0
23	5500.0	450.0

Table 5.1: Depth layers for the North Atlantic simulations.

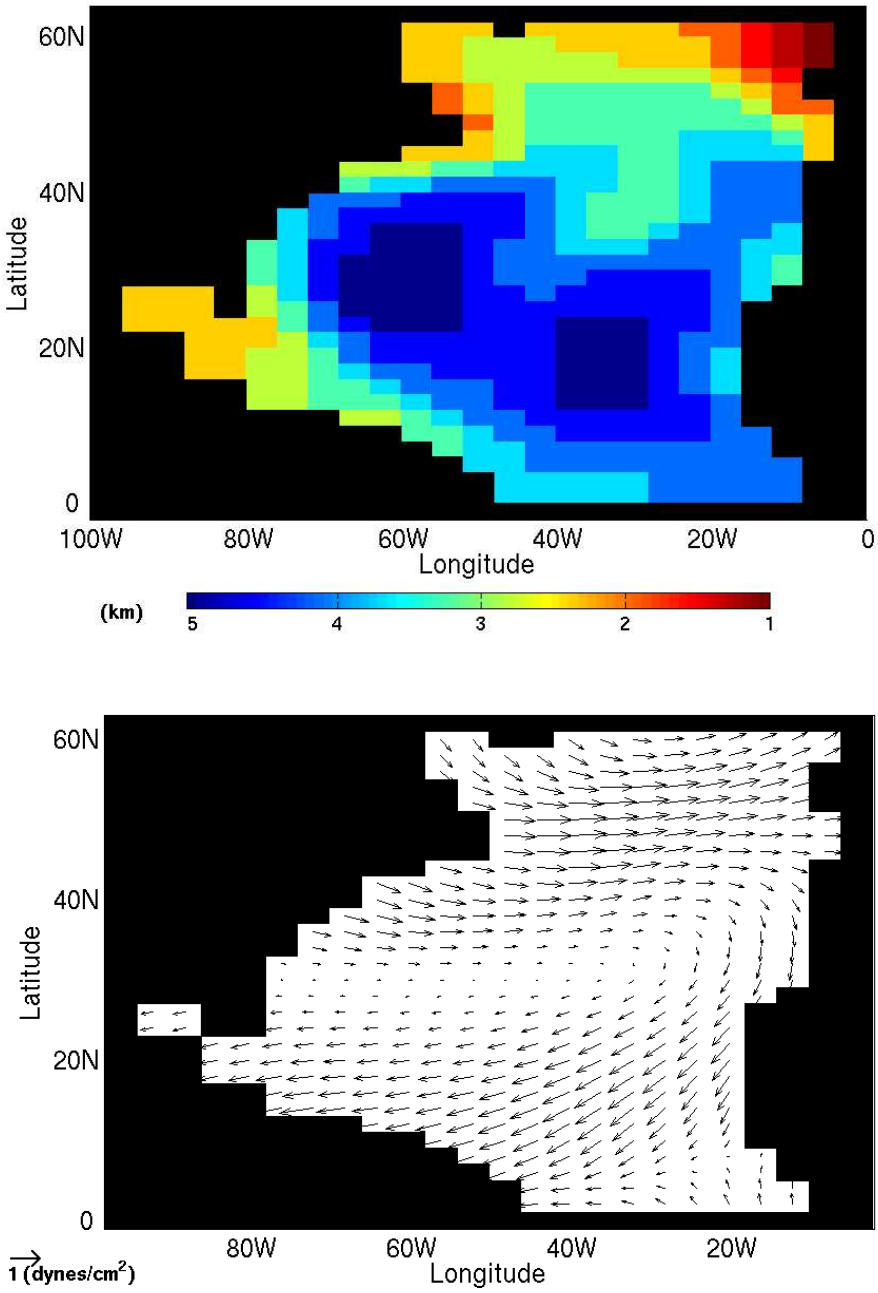


Figure 5.1: Topography (top) and the annual mean of the monthly wind stress (bottom) used in the coarse simulation of the North Atlantic Basin simulations.

5.2.2 Time Advancement

The basin is initialized at rest with temperature (Figure 5.2) and salinity (Figure 5.3) fields spatially interpolated from the Levitus annual means. The time steps shown in Table 5.2 are less than the maximum stable time steps (equations 2.210-2.211). The time steps are identical to those used later in the Global Ocean simulations making them slightly smaller than necessary for modeling the North Atlantic. This was done to make the results obtained here more applicable to the results of the Global Ocean simulations in the next chapter. The coarse and standard simulations are advanced for 100 simulated years to a spin up state. Acceleration techniques for advancing tracers, such as time step splitting and deep ocean acceleration, are not implemented. AMR is not applied since the limited basin benchmark tests (Section 4.4) showed that large portions of the domain would require refinement to produce a globally accurate flow field.

5.2.3 Barotropic Transport at Spin Up

Local barotropic transports expressed in Sverdrups $\left(1Sv = 10^6 \frac{m^3}{s}\right)$ are defined as

$$\Psi_{i,j} = \Psi^E_{i,j}\hat{\lambda} + \Psi^N_{i,j}\hat{\phi} \tag{5.5}$$

$$\Psi^E_{i,j} = 10^{-6}\ R\ \Delta\phi\ H^T_{i,j}\ \overline{\left(\frac{U_{i,j}}{H^U_{i,j}}\right)}^{\lambda\phi}\ \hat{\lambda} \tag{5.6}$$

$$\Psi^N_{i,j} = 10^{-6}\ R\ \cos\phi^T_j\ \Delta\lambda\ H^T_{i,j}\ \overline{\left(\frac{V_{i,j}}{H^U_{i,j}}\right)}^{\lambda\phi}\ \hat{\phi} \tag{5.7}$$

and are located at tracer cell centers. The magnitude of a cell's transport is dependent on its resolution. The eastward and northward components of the barotropic transport ($\Psi^E_{i,j}, \Psi^N_{i,j}$) calculated on resolutions finer than the coarse resolution are therefore reduced to the coarse

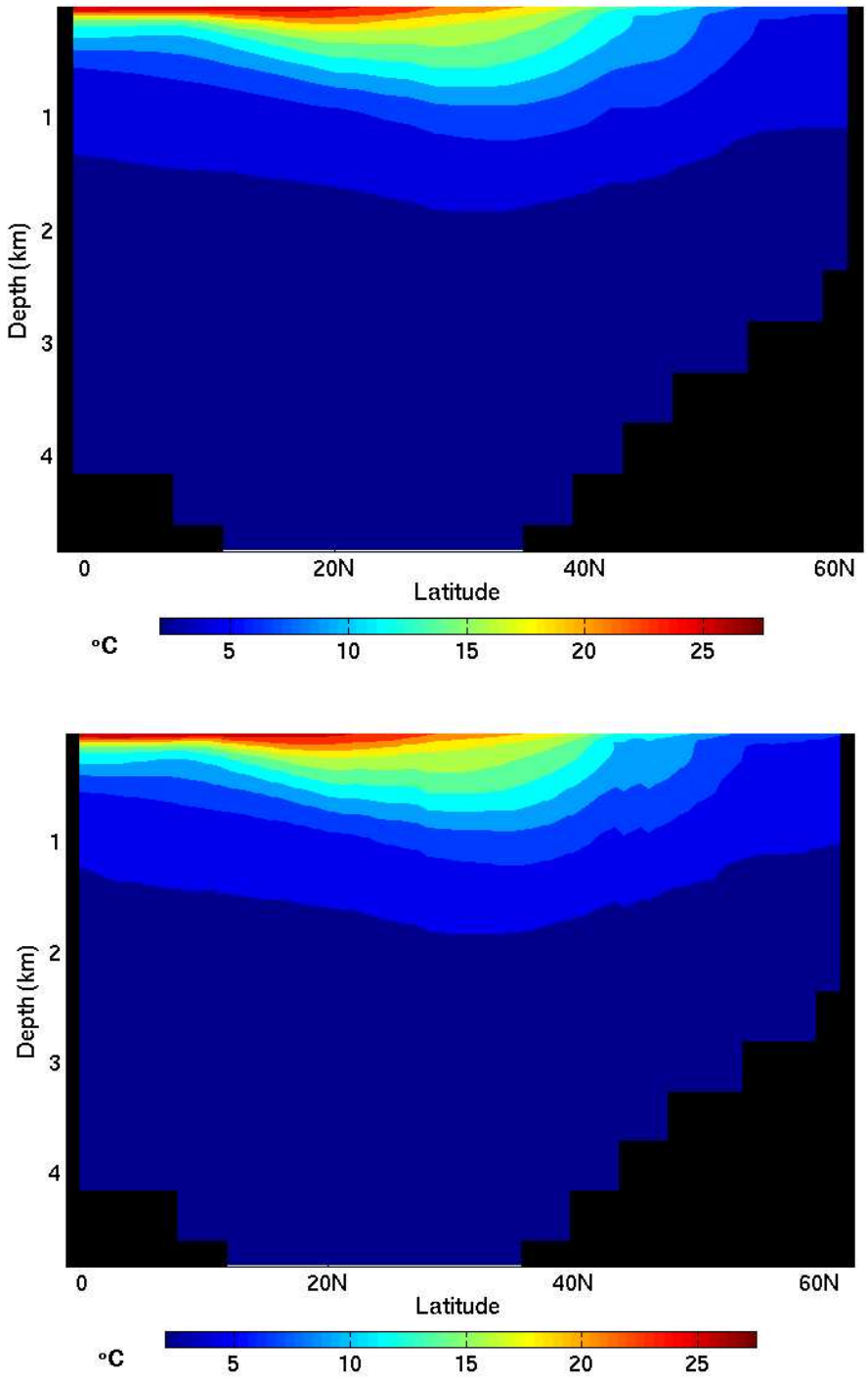


Figure 5.2: Zonal mean initial temperature of the coarse (top) and standard (bottom) simulations for the North Atlantic Basin simulations.

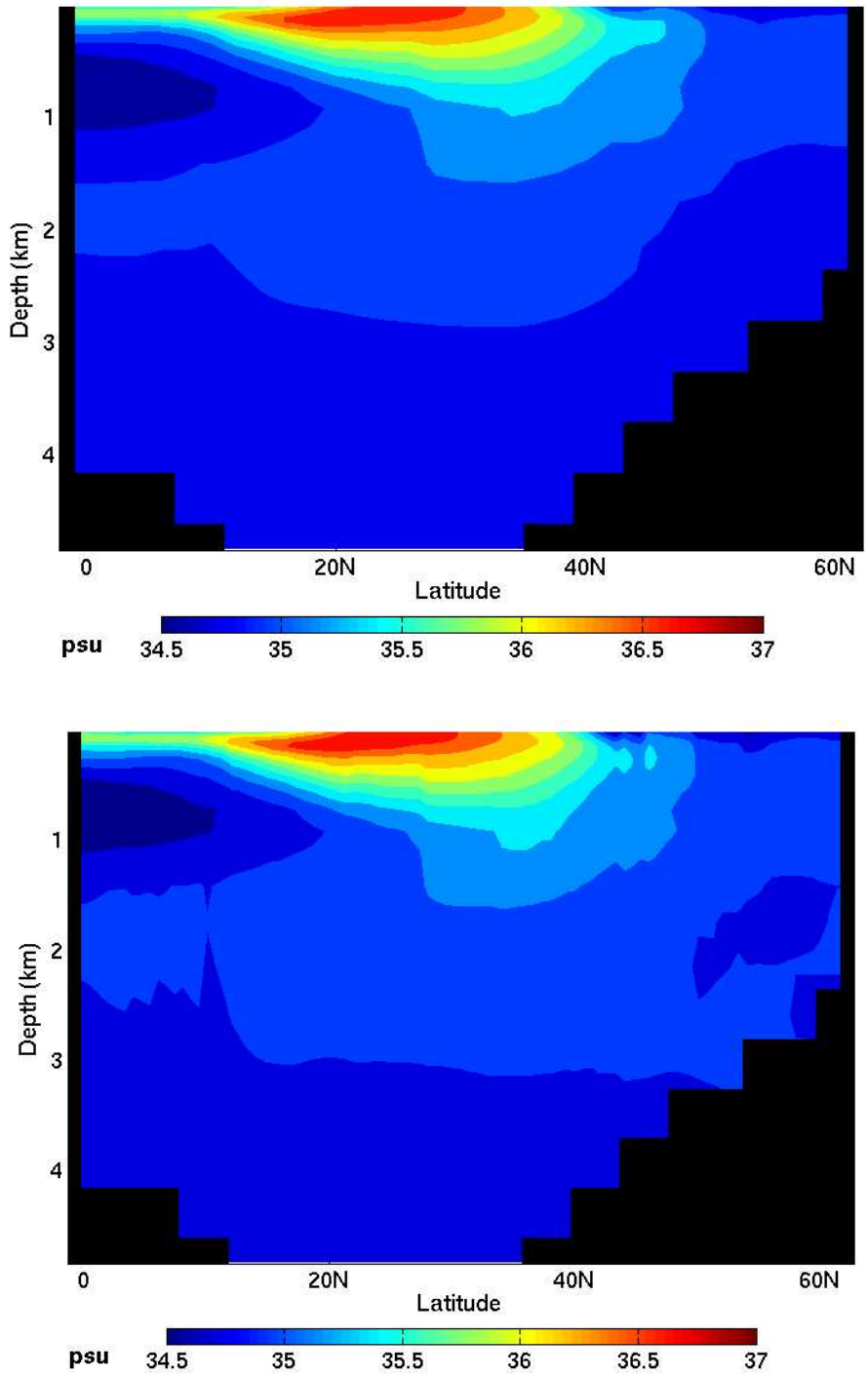


Figure 5.3: Zonal mean initial salinity of the coarse (top) and standard (bottom) simulations for the North Atlantic Basin simulations.

Resolution	Δt^{BC} (minutes)	Δt^{BT} (minutes)	Subcycles ($\Delta t^{\mathrm{BC}}/\Delta t^{\mathrm{BT}}$)
coarse, level-(0)	306	3.4	90
standard, level-(1)	102	1.7	60
level-(2)	11.3333	0.56666	20

Table 5.2: Time steps used in AMR and non-AMR simulations.

resolution ($\overline{\Psi^E_{i,j}}, \overline{\Psi^N_{i,j}}$) to enable comparisons across equal volumes. These operations are given by

$$\overline{\Psi^E_{i,j}} = \frac{1}{r_\lambda} \sum_{i'=ir_\lambda}^{(i+1)r_\lambda-1} \sum_{j'=jr_\phi}^{(j+1)r_\phi-1} \Psi^E_{i',j'} \tag{5.8}$$

$$\overline{\Psi^N_{i,j}} = \frac{1}{r_\phi} \sum_{i'=ir_\lambda}^{(i+1)r_\lambda-1} \sum_{j'=jr_\phi}^{(j+1)r_\phi-1} \Psi^N_{i',j'} \tag{5.9}$$

where (r_λ, r_ϕ) are the ratios of the longitude and latitude grid spacings to the coarse longitude and latitude grid spacings, respectively.

Similar barotropic transport fields are produced by the coarse and standard simulations at year 100 as seen in Figure 5.4. A clockwise rotating gyre is centered roughly about 30° north latitude and has strongest intensity along the eastern United States. The Gulf Stream is slightly more defined by the standard simulation as are the two smaller counter clockwise rotating gyres located near the northern and southern domain boundaries, respectively.

5.2.4 Temperature and Salinity at Spin Up

Depth profiles of the annual mean temperature and salinity during the spin up phase are displayed in Figure 5.5 and Figure 5.6, respectively. The two simulations have

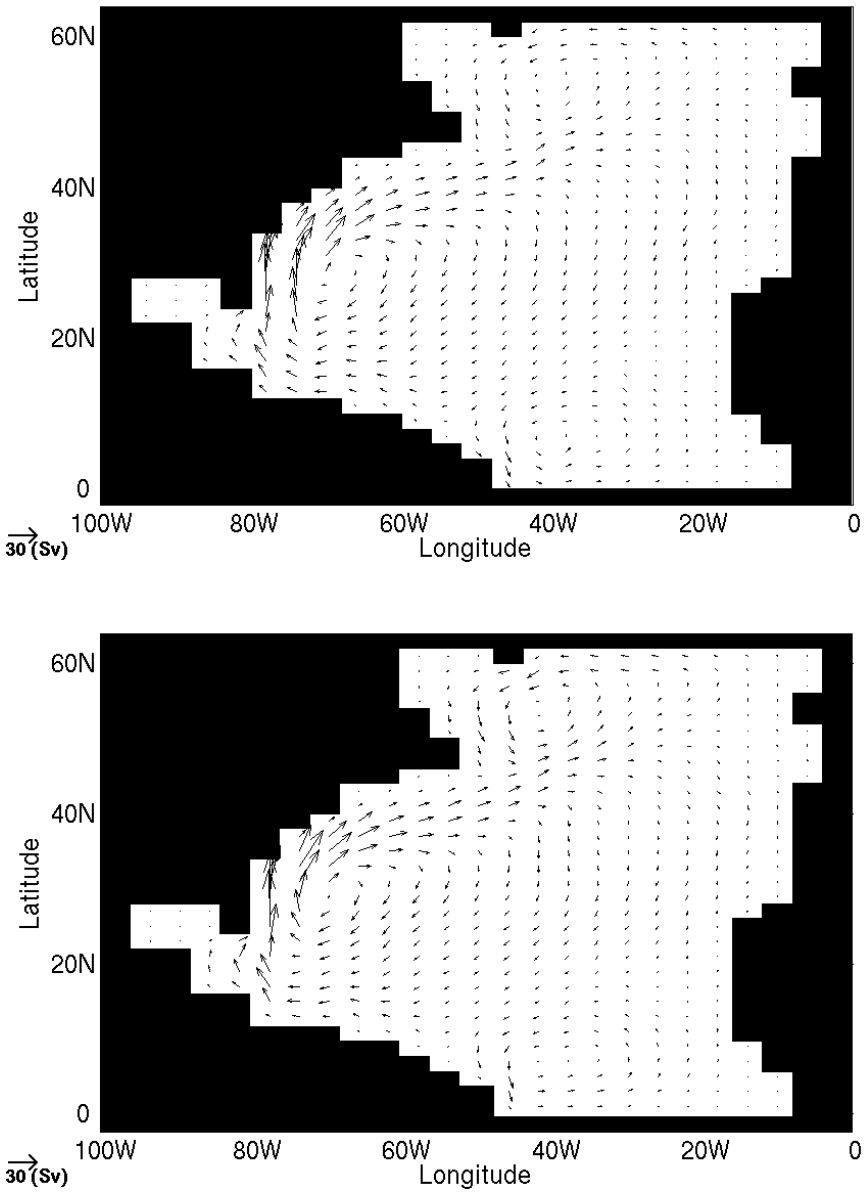


Figure 5.4: Barotropic transport at year 100 for the coarse (top) and standard (bottom) simulations.

nearly identical temperature profiles, and the standard simulation is slightly more fresh than the coarse simulation. The fields tend to increase with time which is also reported by [77] during the first several hundred years of full global ocean simulations. Bryan-Cox-Semtner ocean models typically require at least one thousand simulated years before achieving true steady states for these fields [22, 25, 24, 77]. The annual mean temperature and salinity in Figure 5.5 and Figure 5.6 are still in transition on time scales of thousands of years, but are relatively steady on the scale of several decades by year 100. These fields are therefore believed to be in equilibrium in this study where a 20 year simulation is desired after spin up.

The annual and zonal mean temperature (Figure 5.7) and salinity at year 100 (Figure 5.8) show consistencies between coarse and standard simulations. However, the fields differ from the Levitus annual means in Figure 5.2 and Figure 5.3. The temperature is too diffused and too warm near the northern and southern boundaries. The salinity field indicates the domain is in general under freshed, especially at shallower depths. This is again attributed to the slow transition of the fields towards a true spin up state. Similar Bryan-Cox-Semtner ocean models produce unphysical temperature and salinity fields even after thousands of simulated years [22, 25, 24, 77]. Thus, little is gained by integrating to such time scales for the applications in this study.

The local northward heat transport expressed in Pettawatts ($1PW = 10^{15}W$) is defined here as

$$\begin{aligned} \psi^H_{i,j} &= (10^{-15})\, C_p\, R\Delta\lambda\, \times \\ &\sum_k \left[\rho_{i,j,k}\, \Delta z^T_k \cos\phi_j\, \overline{(\text{Adv\_Flux\_Ts}_{i,j+1,k} - \text{Diff\_Flux\_Ts}_{i,j+1,k})}^{\phi} \right] , \end{aligned} \tag{5.10}$$

where C_p is the specific heat of water $\left(4180\ \frac{J}{kg\ ^\circ C}\right)$ and the flux terms are as defined in Section 2.6.3. The transport is defined at tracer cell centers and the standard simulation transports are reduced to the coarse resolution using an operation analogous with equation (5.9).

Since deviations in salinity are small compared to its mean ($S_0 = 35.2$ psu), the flux of freshwater is approximated as directly proportional to the salinity flux (Flux_FW $=$ $-$Flux_S$/S_0$) [22]. The local northward freshwater transport is then calculated in Sverdrups as

$$\psi^{FW}_{i,j} = -\left(\frac{10^{-6}}{S_0}\right) R\Delta\lambda \sum_k \left[\Delta z^T_k \cos\phi_j \overline{\left(\text{Adv\_Flux\_Ss}_{i,j+1,k} - \text{Diff\_Flux\_Ss}_{i,j+1,k}\right)}^{\phi} .\right] , \tag{5.11}$$

The standard freshwater transports are again reduced to the coarse resolution.

Meridional sums of the annual mean transports of barotropic volume, heat, and fresh water are shown in Figure 5.9 for the final year of spin up. The largest transports occur near 75° west longitude which corresponds to the longitudinal location of the intensified western boundary current in Figure 5.4. The coarse simulation maximum barotropic volume transport agrees well with that of the standard simulation. The maximum heat and freshwater transports of the coarse simulation are slightly less in magnitude than the standard simulation. The coarse and standard transports are in close agreement at longitudes outside of the transport peaks. The circulation fields produced by the coarse and standard simulations after 100 simulated years are essentially the same given the close agreement between their tracer depth profiles and annual mean northward transports.

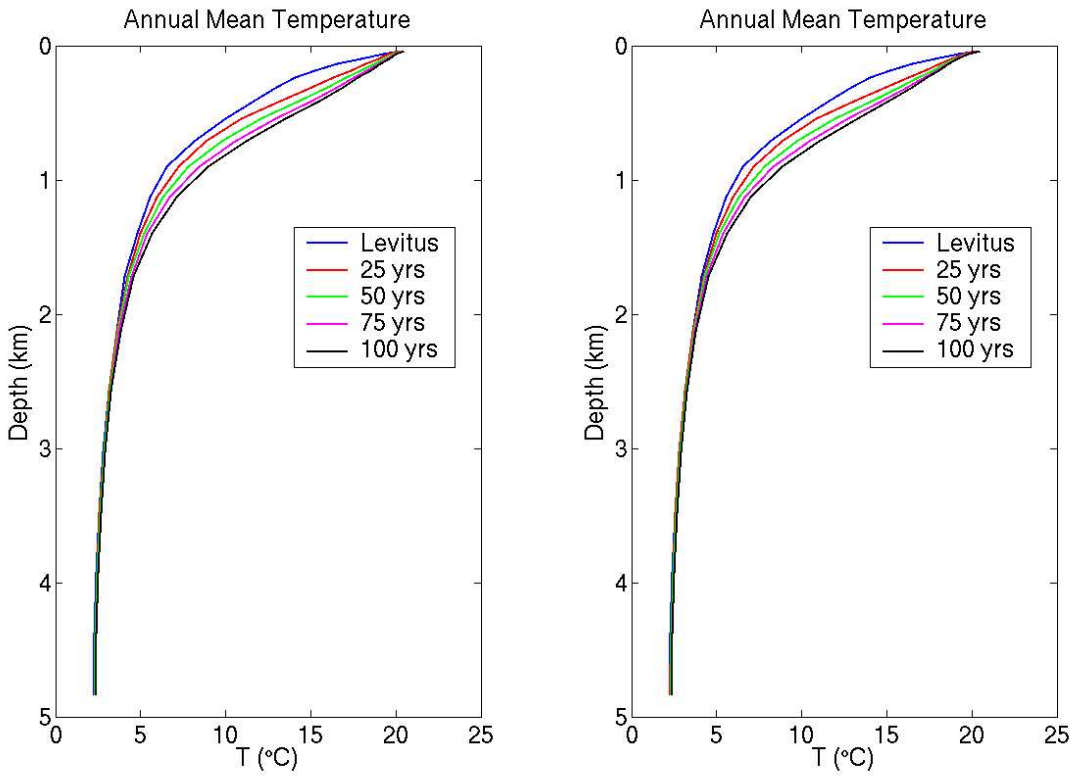


Figure 5.5: Depth profiles of the annual mean temperature for the coarse simulation (left) and the standard simulation (right) during the spin up phase of the North Atlantic Basin simulations.

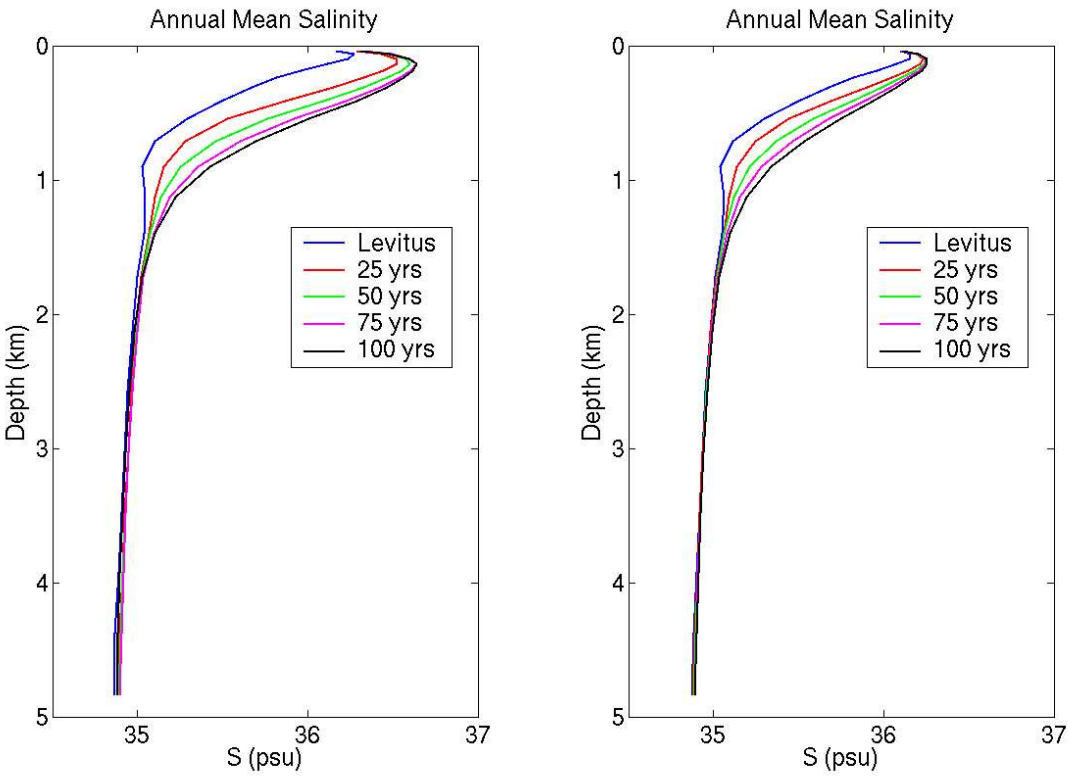


Figure 5.6: Depth profiles of the annual mean salinity for the coarse simulation (left) and the standard simulation (right) during the spin up phase of the North Atlantic Basin simulations.

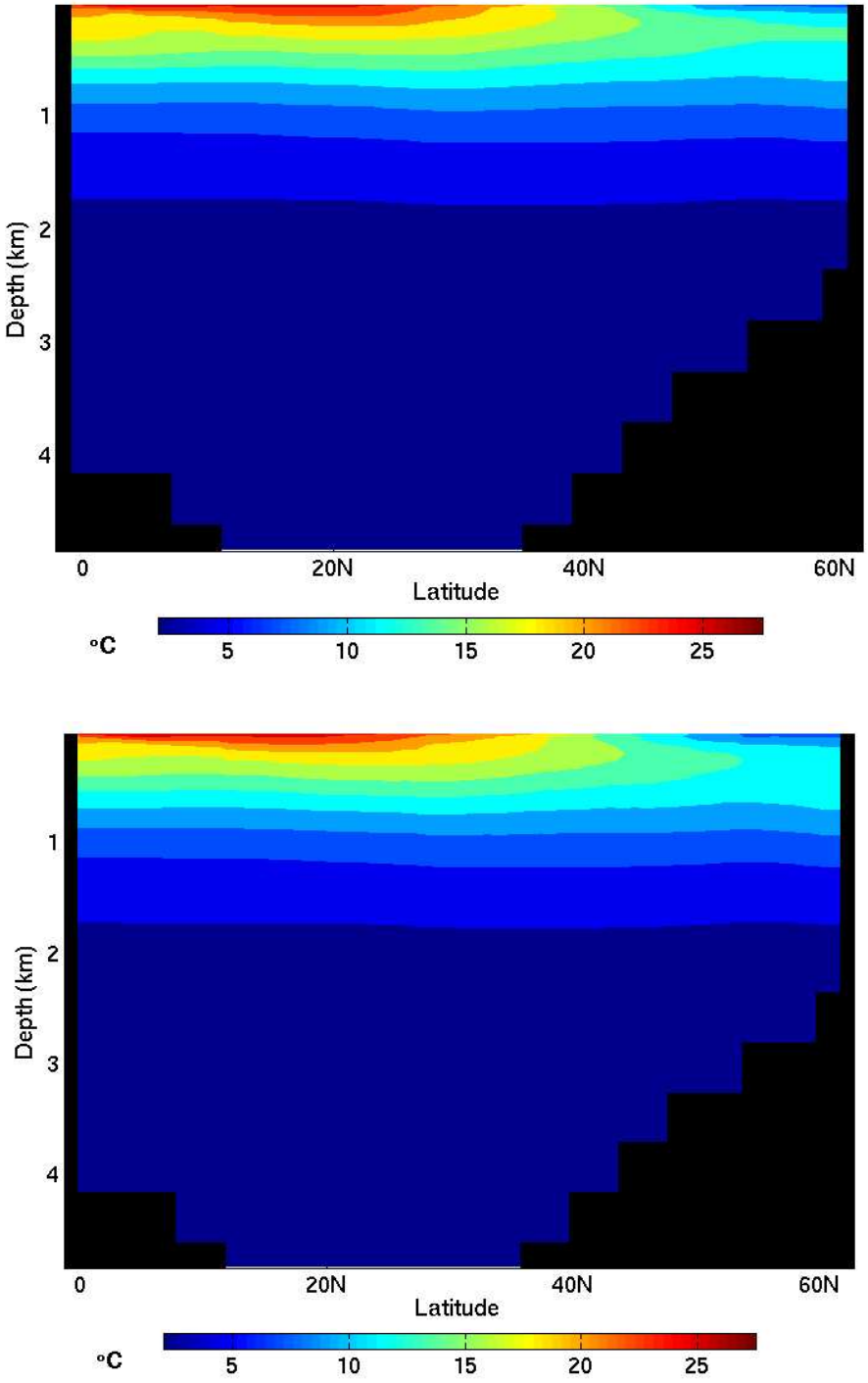


Figure 5.7: Annual and zonal mean temperature of the coarse (top) and standard (bottom) simulations for year 100 of the North Atlantic Basin simulations.

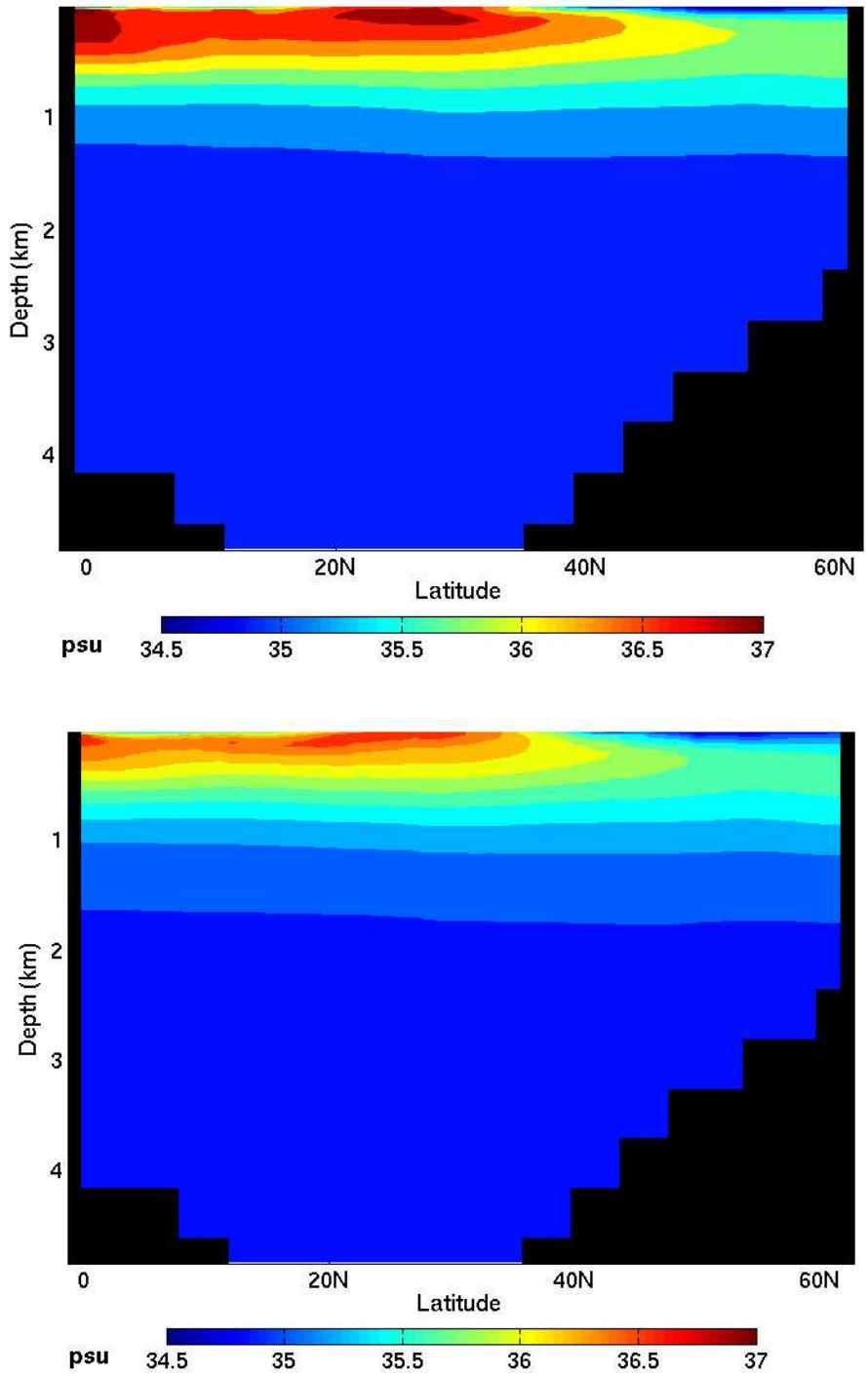


Figure 5.8: Annual and zonal mean salinity of the coarse (top) and standard (bottom) simulations for year 100 of the North Atlantic Basin simulations.

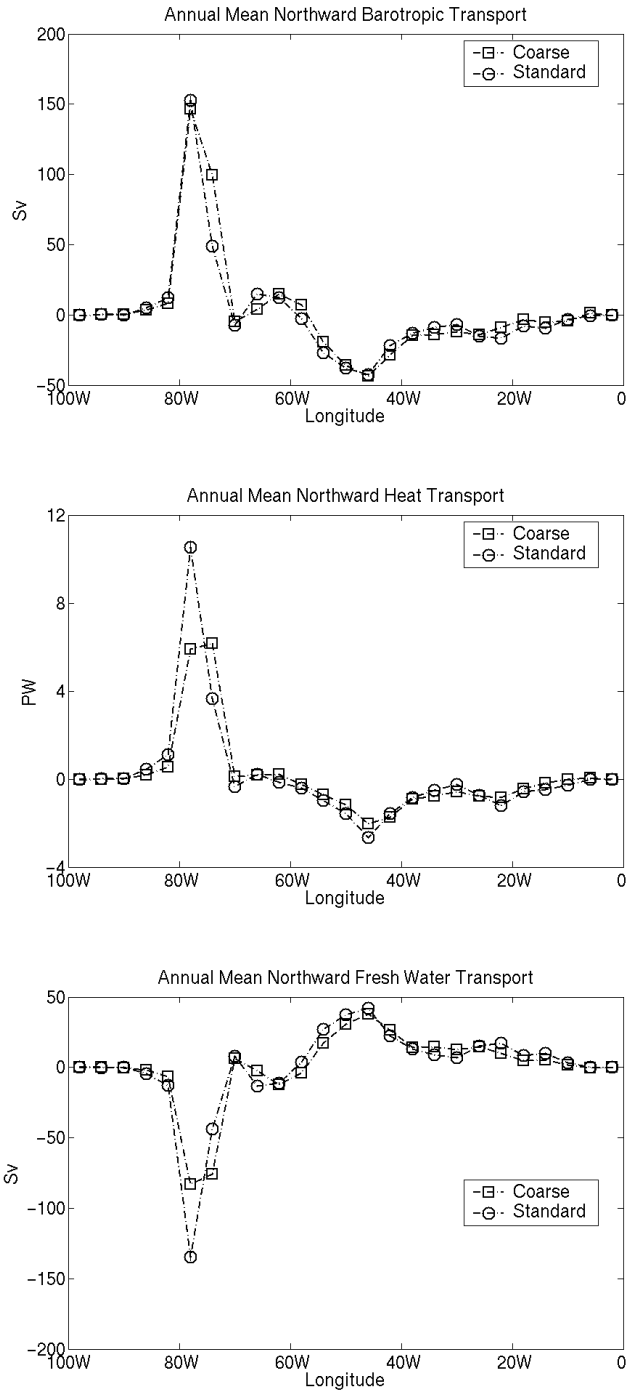


Figure 5.9: Meridional sums of the annual mean northward transports of barotropic volume (top), heat (middle), and freshwater transport (bottom) at year 100 of the North Atlantic Basin simulations.

5.3 CO_2 Direct Injection

CO_2 direct injection is modeled as discussed in Section 5.1. A point source located at 68° west longitude and 27° north latitude is continuously injected at a rate, R_I, equal to $4 \times 10^{12} \frac{g}{yr}$ for 20 simulated years. This corresponds roughly to the emissions generated by a 1 giga-watt power plant located on the eastern coast of the United States near New York City. "Shallow" and "deep" injection scenarios are modeled with the point source depth set to 710 meters and 2575 meters, respectively. The source term in equation (5.2) is defined as the quotient of the mass injection rate and the volume of the injection cell with indices (i_s, j_s, k_s),

$$S_{i,j,k} = \frac{R_I}{\Delta\Omega_{j,k}} \ \delta_{i,i_s} \ \delta_{j,j_s} \ \delta_{k,k_s} \ , \tag{5.12}$$

where $\Delta\Omega_{j,k} = R^2 \cos\phi_j^T \ \Delta\lambda\Delta\phi\Delta z_k^T$ and δ is the Kronecker delta. The magnitude of the source term is low enough to still consider the tracer term a perturbation [36], and is also valid for injection rates with magnitude less than 4×10^{12} g C / yr.

CO_2 direct injection is modeled at the coarse, standard, and AMR resolutions with refinement by factors of 3 in the horizontal directions. An AMR case referenced as "AMR 1" is modeled with level-(0) and level-(1) resolutions equal to the coarse and standard resolutions, respectively. A second AMR case referenced as "AMR 2" has refinement levels level-(0) and level-(1) with resolutions equal to the coarse and standard resolutions, respectively, and a second refinement level with longitude-latitude spacing equal to $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$. The coarse and standard simulations are initialized as their respective spin up states. The AMR simulations are initialized as the coarse spin up state on level-(0), and spatial interpolations of the coarse spin up state on finer levels.

Time steps are set to those shown in Table 5.2. The AMR time refinement ratios between level-(0) and level-(1) are 2 and 3 for the barotropic and non-barotropic systems, respectively. The time refinement ratios between level-(1) and level-(2) are 3 and 9 for the barotropic and non-barotropic systems, respectively. The time steps are numerically stable and satisfy the AMR subcycling rule given by equation (3.26). The interface smoother in equation (4.13) is applied to the AMR interfaces of the surface height field. The interface normal component of the smoother is applied to temperature and salinity fields at AMR interfaces in the top 2 depth layers.

5.3.1 Refinement Criteria

Initial refinement patches are positioned ad hoc on levels finer than level-(0). A single patch on level-(1) refines a 5 grid cell by 5 grid cell section of level-(0) centered about the injection site. A similar initial refinement patch is positioned on level-(2). The AMR hierarchy is regridded every 27 level-(0) steps ($\sim$ 5.7 days) and new refinement levels are created based on refinement criteria. The ideal AMR refinement criteria produces refinement patches such that all variables have relatively flat gradients at the refinement interfaces. This is a difficult, if not impossible, task when applied to an ocean model. Refinement criteria are therefore applied in this study with the primary intent of keeping the passive tracer field highly resolved near the injection source while creating minimal distortions in the flow field. This is accomplished by basing refinement criteria on both the passive tracer concentration and on the circulation near the injection source. Effects of various refinement criteria are not evaluated in this study.

A regrid of level-(ℓ) at time t evaluates the first refinement criterion by performing

a tag calculation on each cell defined as the ratio of the cell's CO_2 mass and the total CO_2 mass injected,

$$\text{Tag\_Calc}_{i,j,c1} \;=\; \frac{C_{i,j,k}\;\Delta\Omega_{j,k}}{t\;\mathrm{R}_I}\;. \tag{5.13}$$

Values obtained by this calculation are unitless and range from 0 to 1. The calculation is applied at the injection depth to better ensure refinement around the injection source. Cells are tagged for refinement by setting the variable, $\text{Tags}_{i,j,c1}$, to 1 on cells where the tag calculation is greater than $\mu_{\ell,1}$. Only the top $\text{max\_frac\_tagged}_\ell$ cells are tagged by applying the procedure described in Section 3.4.1 to prevent excessive refinement. The greatest masses of CO_2 are located near the injection site in this study, and the first criterion places refinement there as a result. In future studies, other criteria could achieve similar results by basing refinement on quantities such as concentration or the gradient in C.

Steep barotropic velocity gradients near the injection site are next tagged for refinement using a second refinement criterion. The tag calculation is the maximum gradient of each cell's barotropic integrated velocity components,

$$\text{Tag\_Calc}_{i,j,c2} \;=\; \max\Big(\text{grad}^\lambda(U_{i,j}),\;\text{grad}^\phi(U_{i,j}),\;\text{grad}^\lambda(V_{i,j}),\;\text{grad}^\phi(V_{i,j})\Big)\;. \tag{5.14}$$

The gradients are calculated using absolute one sided differences,

$$\text{grad}^\lambda(U_{i,j}) \;=\; \frac{\max\left(\;|\mathtt{U}_{i,j}-\mathtt{U}_{i-1,j}|,\;|\mathtt{U}_{i+1,j}-\mathtt{U}_{i,j}|\;\right)}{R\;\cos\phi_j^U\;\Delta\phi} \tag{5.15}$$

$$\text{grad}^\phi(U_{i,j}) \;=\; \frac{\max\left(\;|\mathtt{U}_{i,j}-\mathtt{U}_{i,j-1}|,\;|\mathtt{U}_{i,j+1}-\mathtt{U}_{i,j}|\;\right)}{R\;\Delta\phi} \tag{5.16}$$

$$\text{grad}^\lambda(V_{i,j}) \;=\; \frac{\max\left(\;|\mathtt{V}_{i,j}-\mathtt{V}_{i-1,j}|,\;|\mathtt{V}_{i+1,j}-\mathtt{V}_{i,j}|\;\right)}{R\;\cos\phi_j^U\;\Delta\phi} \tag{5.17}$$

$$\text{grad}^\phi(V_{i,j}) \;=\; \frac{\max\left(\;|\mathtt{V}_{i,j}-\mathtt{V}_{i,j-1}|,\;|\mathtt{V}_{i,j+1}-\mathtt{V}_{i,j}|\;\right)}{R\;\Delta\phi}\;, \tag{5.18}$$

Refinement Level	$\mu_{\ell,1}$ (unitless)	max_frac_tagged$_\ell$ (%)	$\mu_{\ell,2}$ (s^{-1})	Δc1
level-(0)	0.98	15	3×10^{-8}	3
level-(1)	0.90	5	9×10^{-8}	3

Table 5.3: Parameters for AMR refinement criteria in the North Atlantic CO_2 direct injection simulations.

where $(\mathtt{U}_{i,j}, \mathtt{V}_{i,j}) = (U_{i,j}, V_{i,j})/H^U_{i,j}$. A cell on level-($\ell$) is tagged for refinement if Tag_Calc$_{i,j,c2}$ is greater than $\mu_{\ell,2}$. This criterion is applied only to cells that are within Δc1 cell lengths of a cell previously tagged for refinement by the first refinement criterion. This is given by

$$\text{Tags}_{i,j,c2} = \begin{cases} 1 \; - \; \prod_{i',j'} \left(1 - \text{Tags}_{i',j',c1}\right) & , \quad \textbf{if } \left(\text{Tag\_Calc}_{i,j,c2} \; > \; \mu_{\ell,c2}\right) \\ 0 & , \quad \text{otherwise} \end{cases} \tag{5.19}$$

where the indices (i', j') range from $(i - N_{c1}, j - N_{c1})$ to $(i + N_{c1}, j + N_{c1})$.

The refinement parameters applied during regrids of the CO_2 direct injection simulations are shown in Table 5.3. The two tag variables on level-(ℓ) are clustered separately using SAMRAI. This gives two sets of boxes describing the sizes and locations of new refinement patches on level-($\ell + 1$) (i.e., one set from Tags$_{i,j,c1}$ and one set from Tags$_{i,j,c2}$). The two sets are combined prior to creating new patches on level-($\ell + 1$). This creates the potential for some unwanted overlapping among neighboring patches, but is sparsely observed in the simulations presented here.

5.3.2 Sequestration Results

Column inventories of injected CO_2 at year 20 are shown in Figure 5.10 for the shallow injection case and in Figure 5.11 for the deep injection case. The first and second

refinement levels in the AMR cases are outlined in green and magenta, respectively. The shallow case is more advective due to its proximity to stronger surface currents. The passive tracer propagation is primarily diffusive in the deep injection case because of weaker currents in the deep ocean. The tracer distribution in the deep case is more localized which allows AMR regrids to construct simpler refinement patch clusters and to keep more of the injected CO_2 refined.

The local differences of coarse and AMR column inventories relative to the standard column inventories are shown in Figure 5.12. All differences are normalized by the maximum standard value. The largest differences occur within approximately 7 coarse resolution grid cells of the injection site. Differences are greatest in the deep injection simulations along the western boundary. This is believed to result from deep ocean currents which circulate opposite the Gulf Stream during the spring months. The shallow injection differences are greatest at the injection site, and are larger in magnitude than the deep injection cases.

The global normalized RMS errors are computed of the CO_2 perturbation relative to the standard simulation for the coarse and AMR 1 simulations as a function of time. Errors are not computed for the AMR 2 simulations since level-(2) is presumably more accurate than the standard simulation and comparisons between the two are not valid measurements of error. The coarse data are compared to the standard data by cell area weighted averaging to the coarse resolution. The same is done for comparisons on level-(0) of the AMR 1 simulations. Direct comparisons are made between the standard and level-(1) data given that their resolutions are equal. Figure 5.13 displays the errors plotted as

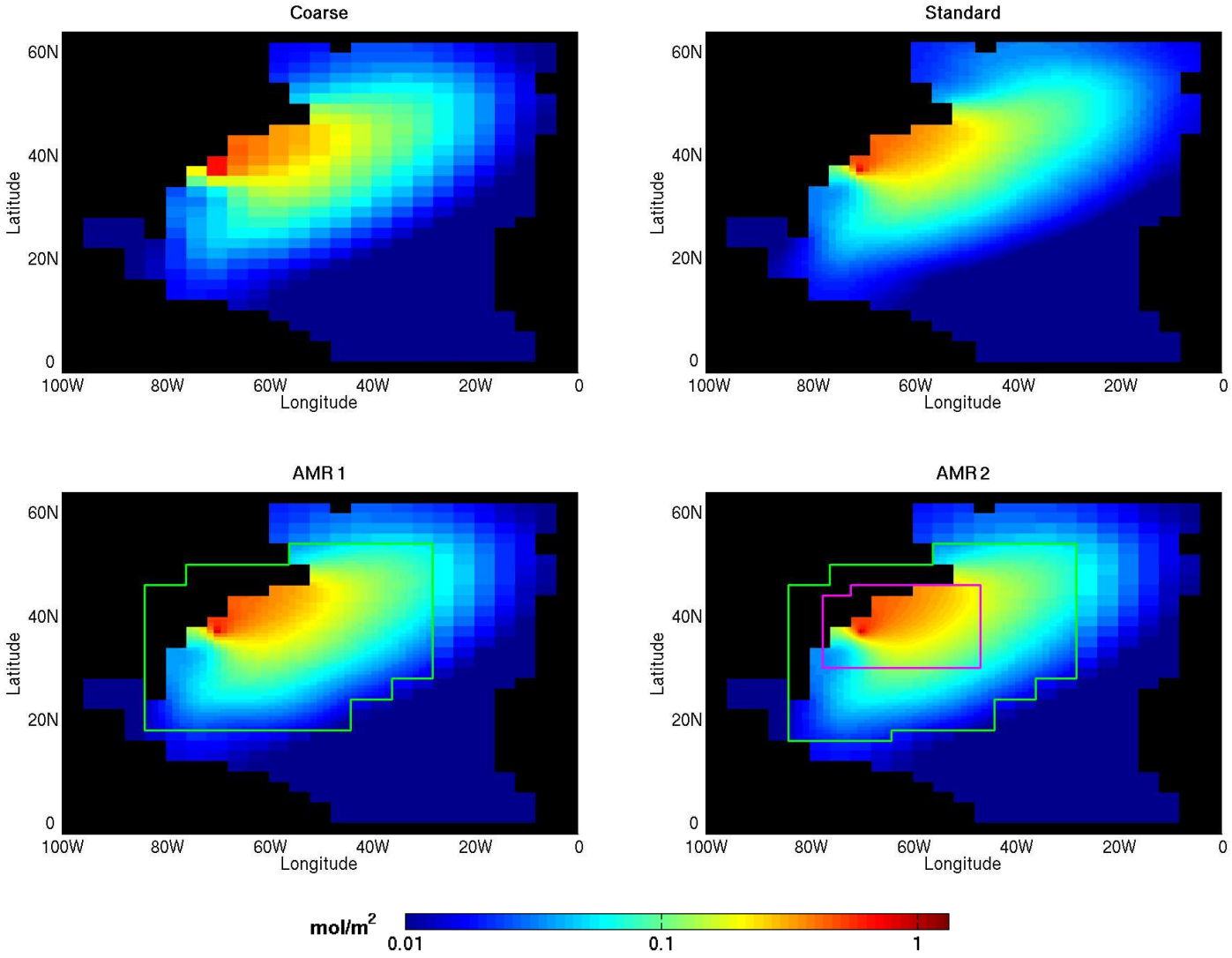


Figure 5.10: Vertically integrated CO_2 after 20 years of continuous injection at a depth of 710 meters.

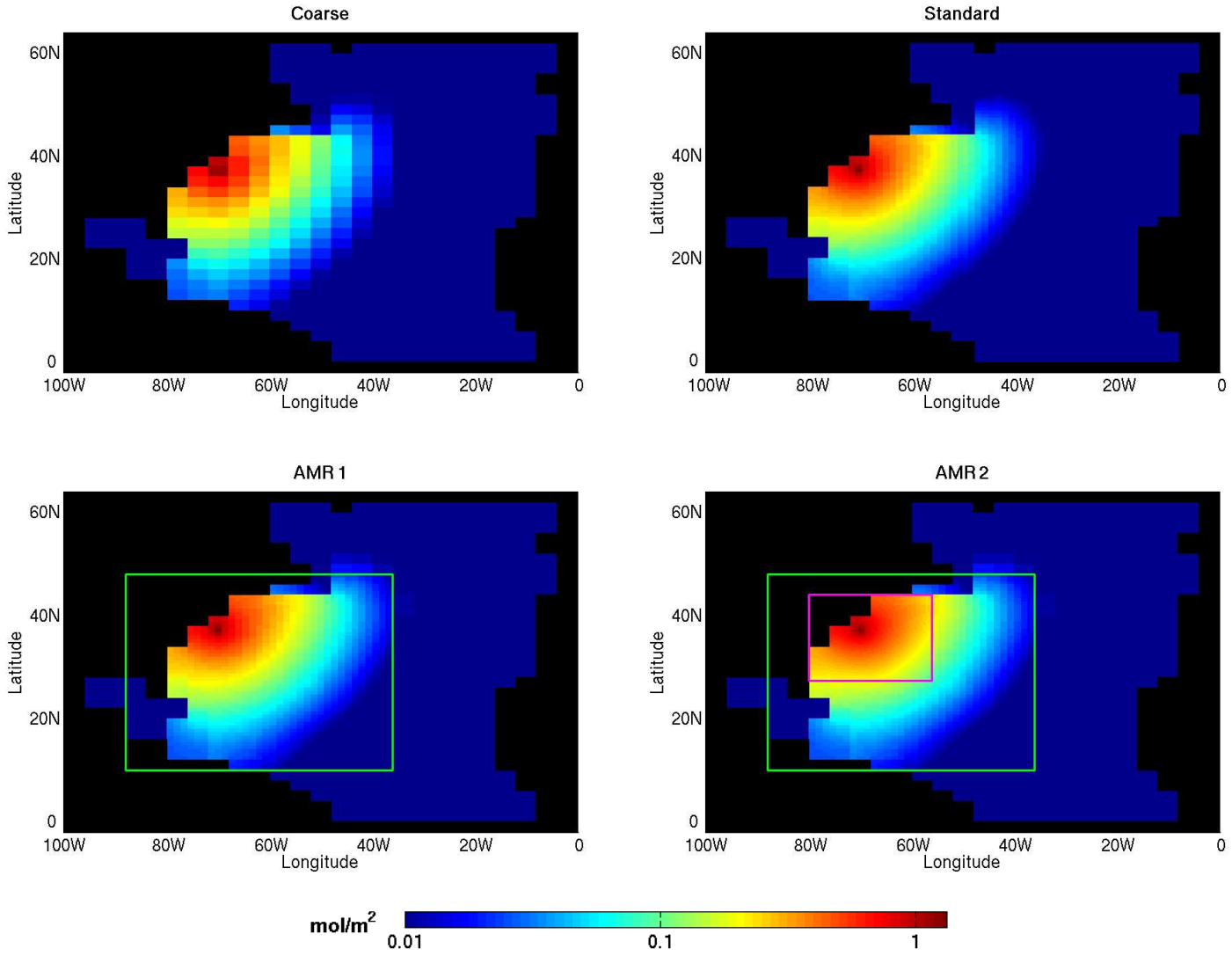


Figure 5.11: Vertically integrated CO_2 after 20 years of continuous injection at a depth of 2575 meters.

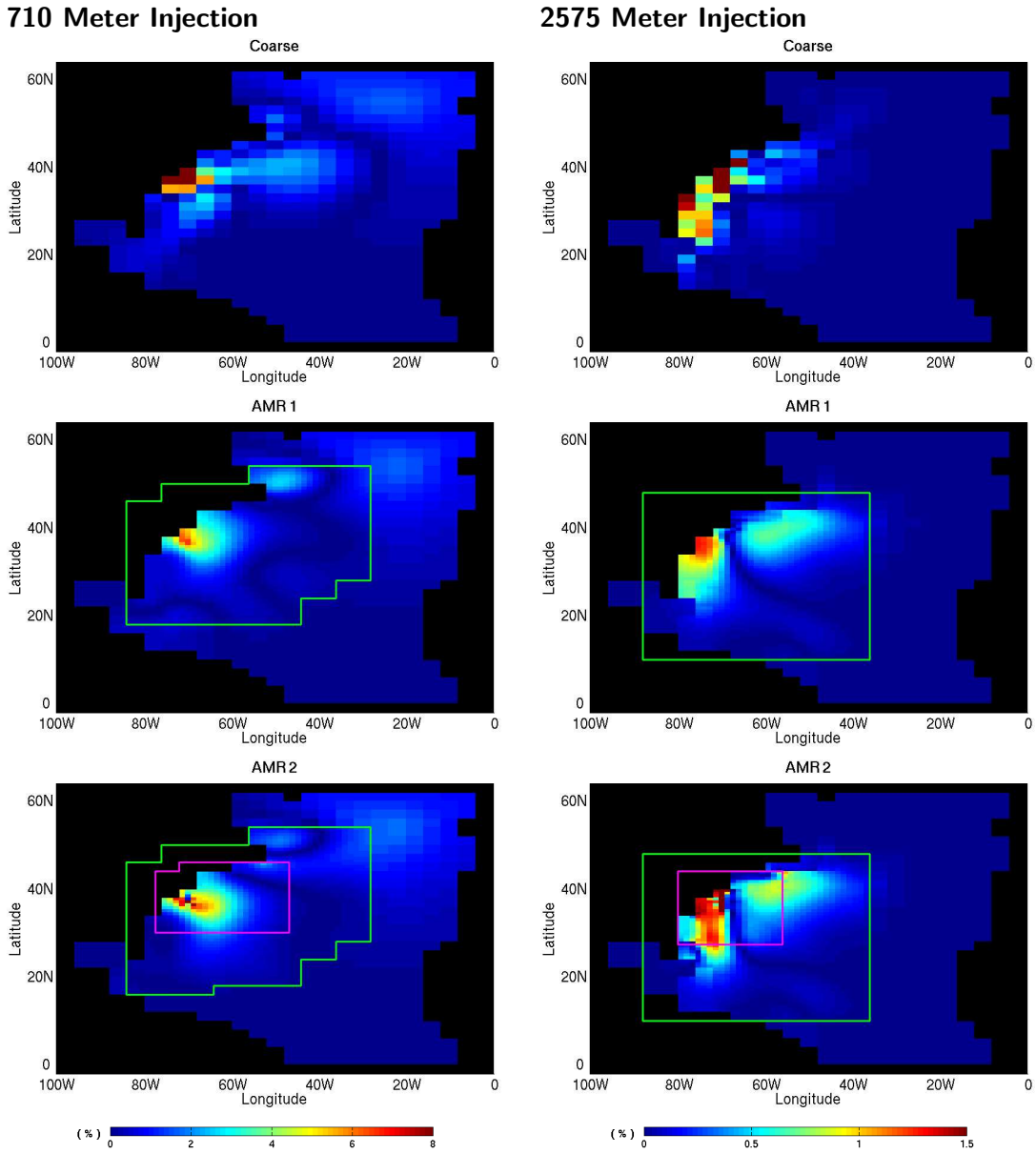


Figure 5.12: Local differences in vertically integrated CO_2 for the coarse, AMR 1, and AMR 2 simulations relative to the standard simulation after 20 years of continuous injection. The values displayed for the shallow (left) and deep (right) injection cases are normalized by the maximum column inventories in the shallow and deep standard cases, respectively.

a function of time for the deep and shallow injection cases. The coarse errors are largest in the first year given that the majority of CO_2 is located near the injection site where local errors are the largest. The coarse global errors decrease over time as more CO_2 is propagated away from the injection site to regions where the local errors are significantly reduced. The AMR 1 global errors are much lower than the coarse global errors and tend to increase over time. This is anticipated given that refinement interfaces produce small errors in the circulation field which build up throughout the simulation. It is possible that the AMR 1 simulations will eventually have global errors larger than the coarse simulation global errors if applied over time scales far longer than 20 years.

Figure 5.14 displays the depth profiles of errors in CO_2 concentration at year 20. Standard simulation data are again reduced to the coarse resolution where necessary. The error of a simulation at a particular depth is defined as the root mean square of its differences relative to the reduced standard data at the same depth. The AMR 1 errors are less than the coarse errors at all depths for both injection depths. The shallow injection cases have smaller errors than the deep injection cases in the deep ocean. The deep injection errors are less than the shallow injection errors at shallow depths. In each injection case, the largest errors occur at the injection depth and is the basis for applying the first refinement criterion only to cells at the injection depth.

Spatial distributions of CO_2 surface fluxes at year 20 are shown in Figure 5.15 for the shallow injection case and in Figure 5.16 for the deep injection case. Positive flux represents transfers of CO_2 from the ocean to the atmosphere. The shallow injection surface fluxes are significantly higher than the deep injection surface fluxes as expected [17]. The

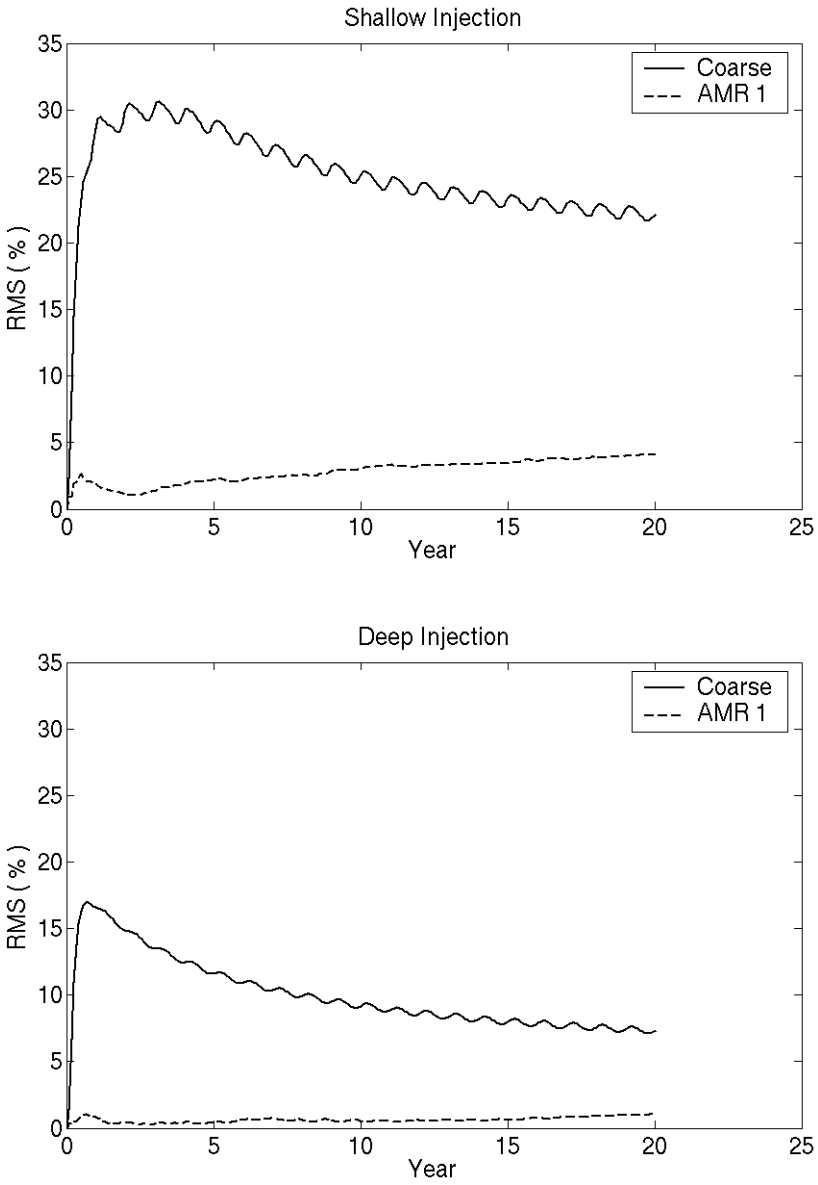


Figure 5.13: Global RMS passive tracer errors as a function of time for the shallow (top) and deep (bottom) injections.

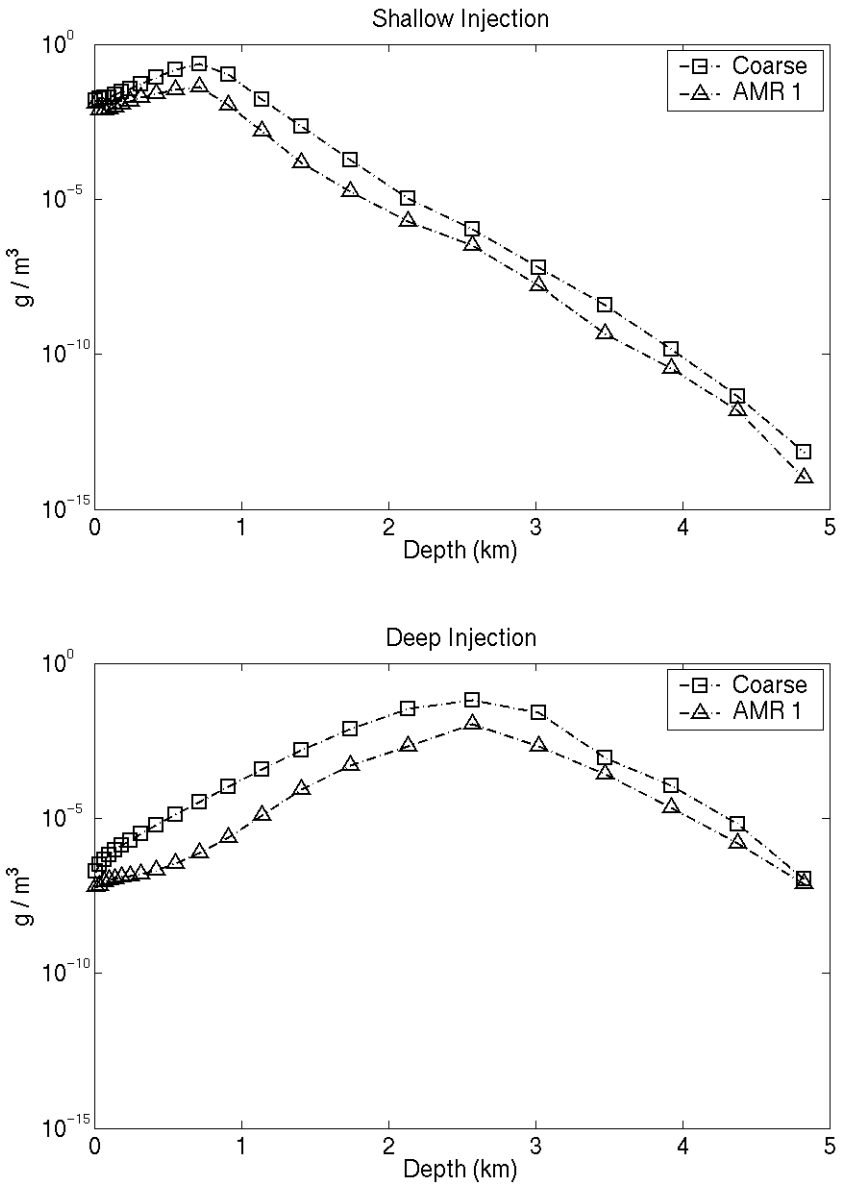


Figure 5.14: RMS passive tracer errors as a function of depth at year 20 for the shallow (top) and deep (bottom) injections.

northern half of the basin releases at least 100 times more CO_2 than the southern half in both injection cases. This is expected given the strong influences of the Gulf Stream at shallow depths.

The spatial distributions of the coarse, AMR 1, and AMR 2 differences in surface flux relative to the standard surface flux are shown in Figure 5.17. AMR refinement is located near the injection sites and therefore does not attempt to improve the accuracy at the surface. Even so, AMR surface fluxes have smaller differences than the coarse surface fluxes. The deep injection surface flux differences are larger than those in the shallow injection case given that the deep injection CO_2 must travel farther to get to the surface which permits more opportunity for error growth.

The total outgassing of injected CO_2 from the ocean to the atmosphere is plotted as a function of time in Figure 5.18 along with the deviations in total outgassing relative to the standard cases. More outgassing results in the shallow injection cases than the deep injection cases as expected. Outgassing deviations in the shallow injection cases are greatest during the first year. Any outgassing during this time is small and located near the injection site. This suggests that coarse and AMR simulations produce vertical transports near the shallow injection site that are consistently different than that produced in the standard simulation. The AMR deviations are less than the coarse deviations in the shallow cases for simulation times after the first year. The deviations in outgassing for the deep cases show less agreement than the shallow cases. The vertical transports in the deep ocean therefore vary for all simulations. The inconsistencies between the deep injection outgassing deviations fortunately occur in this study when the total outgassing is relatively

small and therefore have little effect on the accumulated outgassing after 20 years. This is not guaranteed to be true for other CO_2 deep injection simulations.

The percentages of sequestered CO_2 remaining in the ocean after 20 years of continuous injection are shown in Table 5.4. In the shallow injection simulations, the AMR predictions are closer to the standard prediction than is the coarse. The shallow injection coarse predictions agree with the standard prediction to within 1%. All resolutions predict approximately 99.99% of the deep injected CO_2 will remain in the ocean. The horizontal resolution is therefore not crucial for predicting outgassing in CO_2 direct injection simulations on time scales of 2 decades.

The biological impact of CO_2 sequestration is assessed through pH changes resulting from the increase in ocean acidity. As explained in Section 5.1, the maximum possible pH changes are reported given that the AMR Ocean Model is not equipped with a means of modeling background pH. Future studies of pH changes could also be calculated relative to data set estimations of ocean carbonate chemistry [55]. Figure 5.20 shows the fraction of ocean volume verses the maximum calculated pH change for each simulation in the shallow and deep injection cases. For a given pH change, the volume fraction displayed is the fraction of ocean volume with a pH change at least that large.

Figure 5.20 represents about 1% of the total volume in the ocean basin (1.43×10^{17} m^3). Calculations of pH changes lower than those displayed agree well in predicted volume fraction among all simulations. All simulations agree well at large volume scales, but predictions become more dependent on resolution when looking at volume scales approaching that of the injection grid cell. The AMR 2 simulations predict larger maximum pH changes

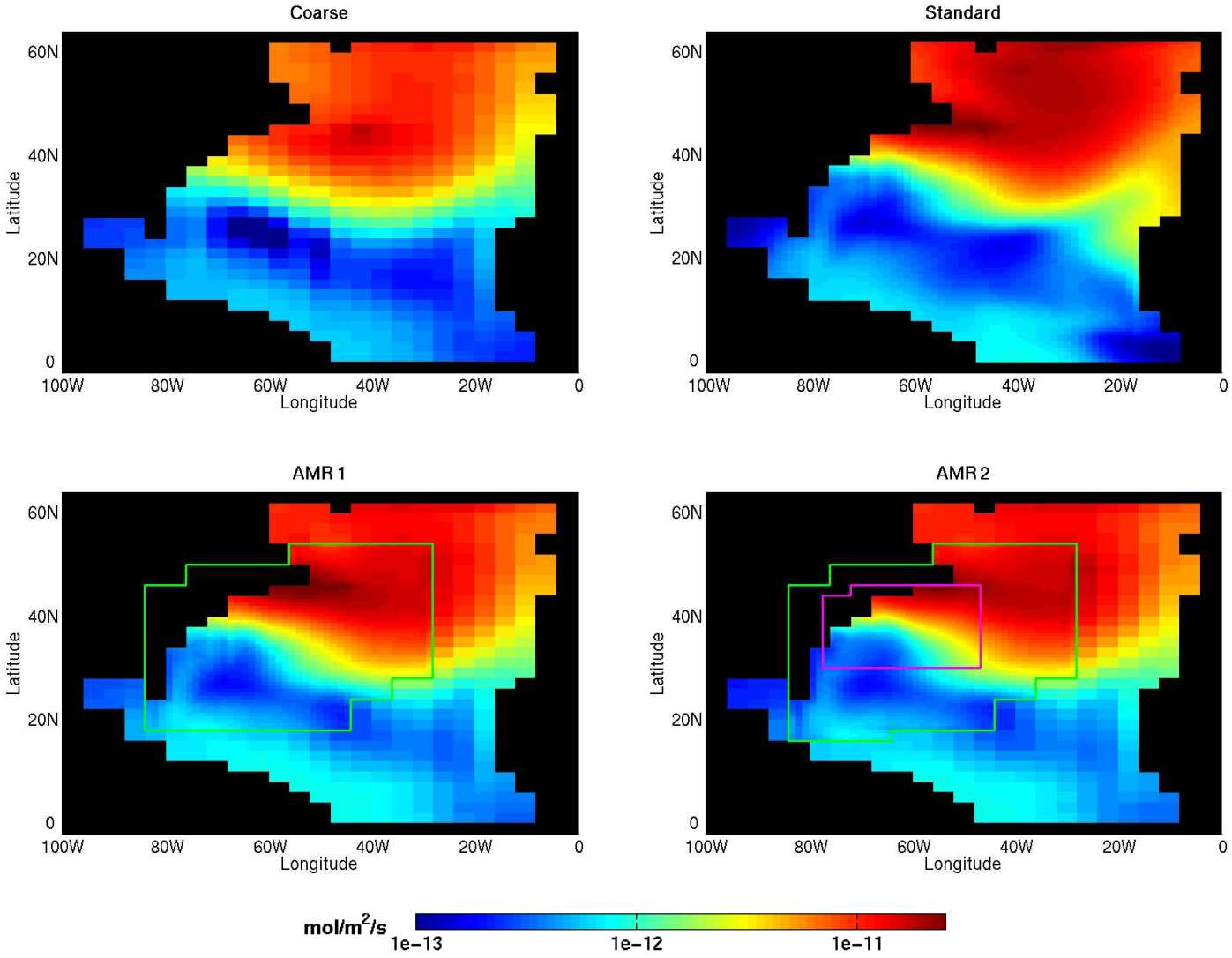


Figure 5.15: Surface flux of sequestered CO_2 after 20 years of continuous injection at a depth of 710 meters.

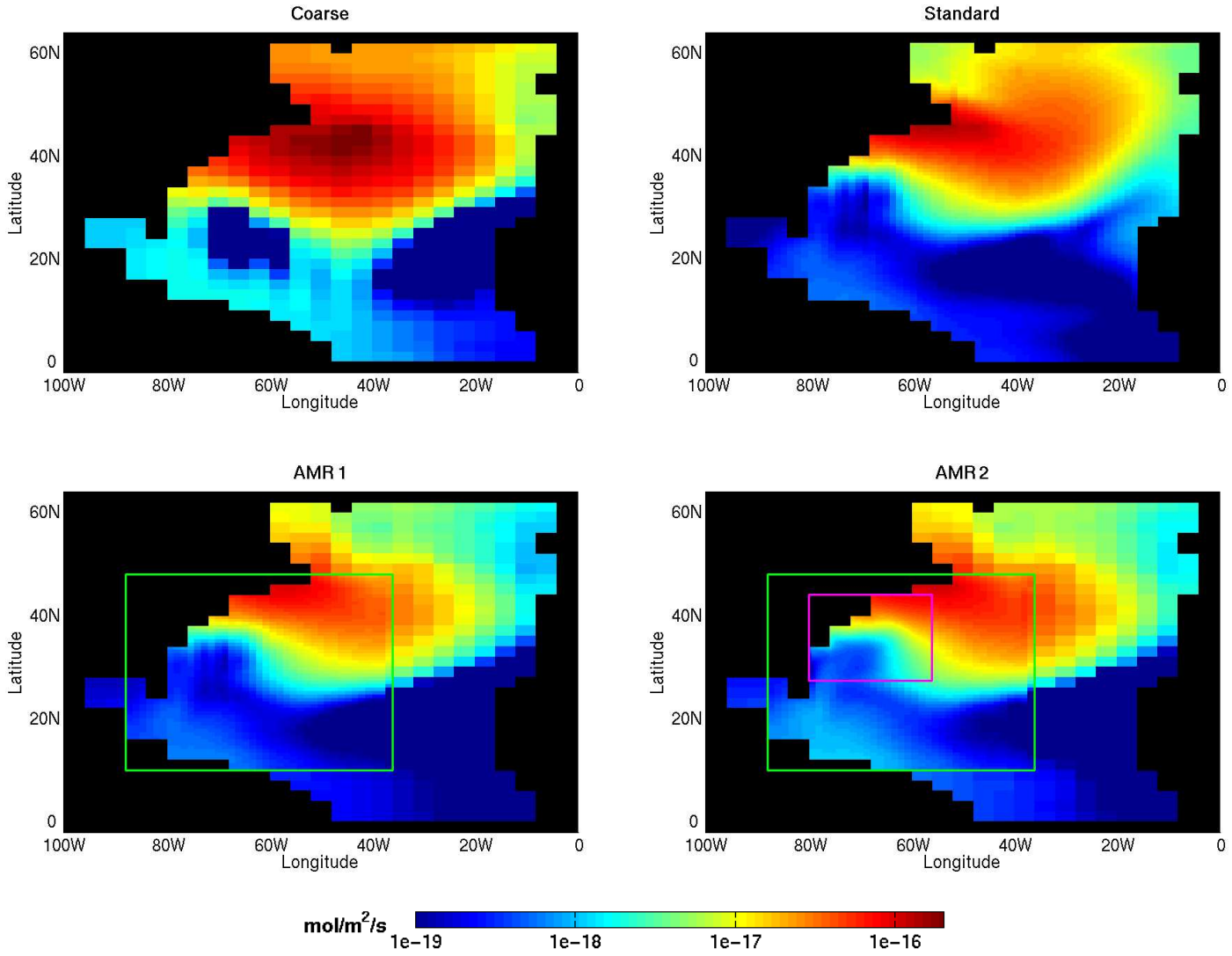


Figure 5.16: Surface flux of sequestered CO_2 after 20 years of continuous injection at a depth of 2575 meters.

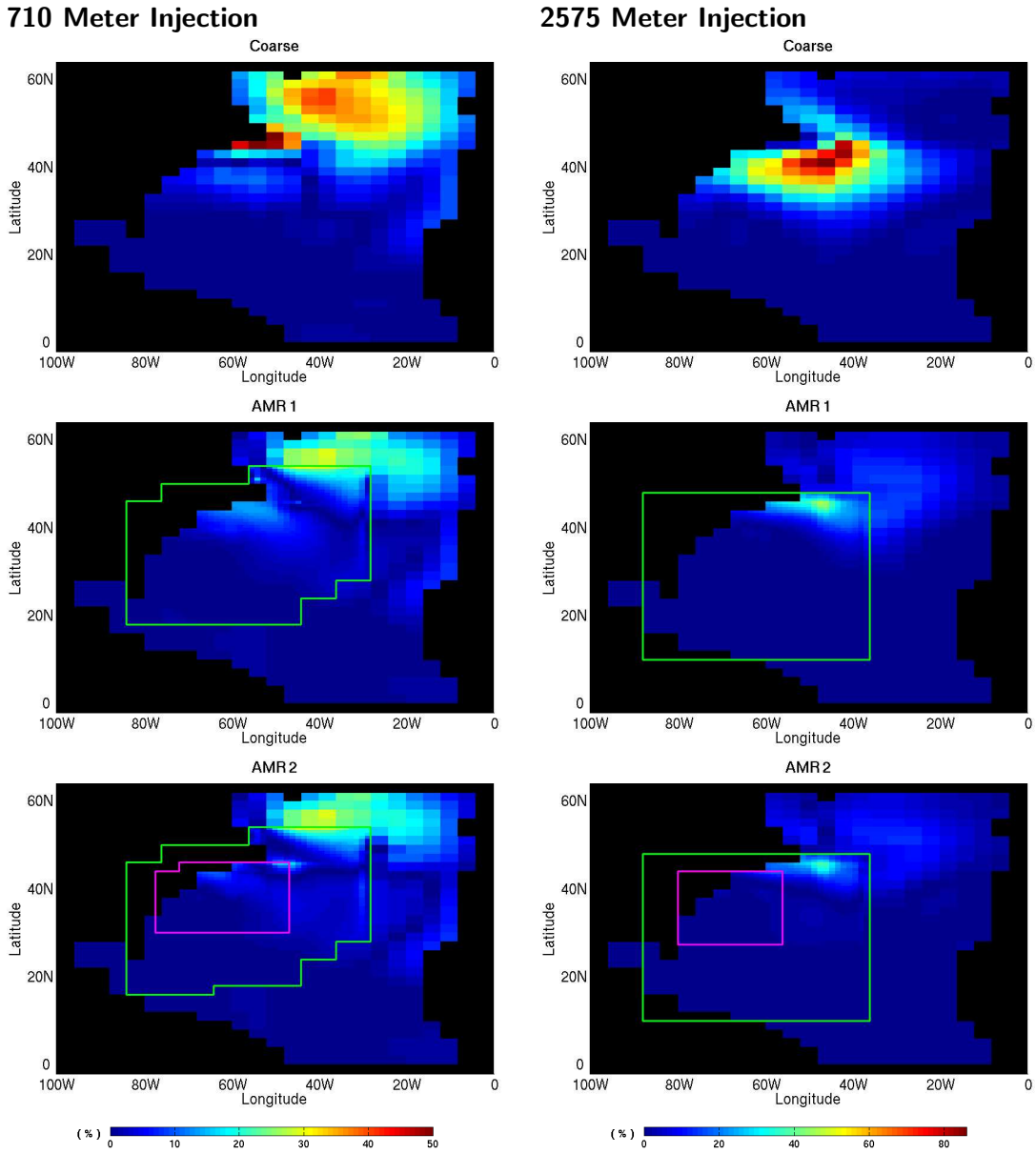


Figure 5.17: Local differences in CO_2 surface flux for the coarse, AMR 1, and AMR 2 simulations relative to the standard simulation after 20 years of continuous injection. The values displayed for the shallow (left) and deep (right) injection cases are normalized by the maximum column inventories in the shallow and deep standard cases, respectively.

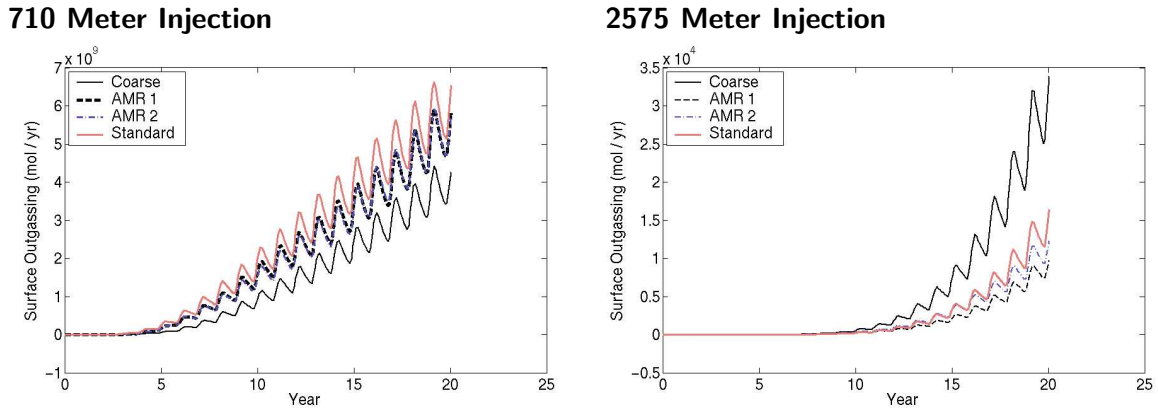


Figure 5.18: Total surface flux of sequestered CO_2 as a function of time for the shallow (left) and deep (right) injections.

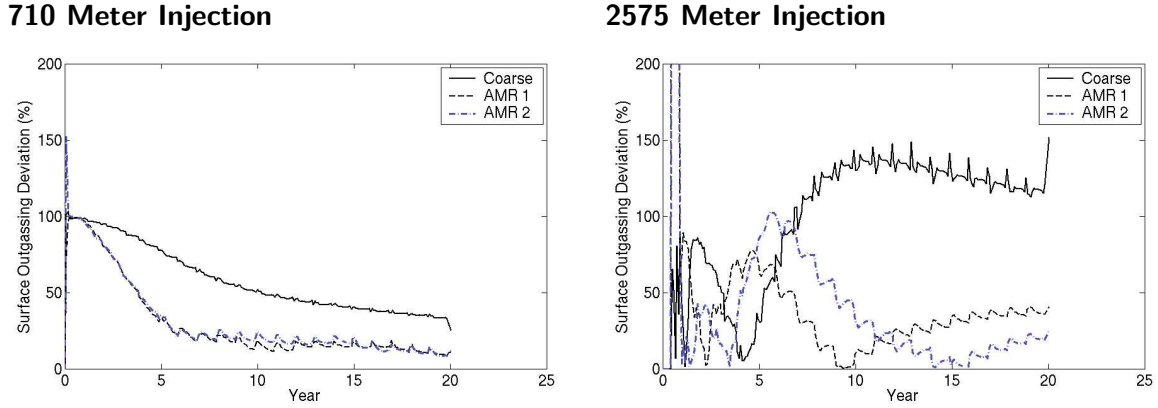


Figure 5.19: Error in total CO_2 surface flux relative to the standard case as a function of time for the shallow (left) and deep (right) injections.

Injection Depth (meters)	Coarse (%)	Standard (%)	AMR 1 (%)	AMR 2 (%)
710	98.60	97.61	98.02	98.06
2575	99.99	99.99	99.99	99.99

Table 5.4: Percentages of sequestered CO_2 remaining in the ocean after 20 years of continuous injection.

than are possible with the other tested simulations. This is because CO_2 is injected into a smaller grid cell volume giving a higher CO_2 concentration which results in a higher change in pH. The predictions given by the standard simulation at large and small volume scales are also achieved by the AMR 1 simulation.

Similar trends are seen in Figure 5.21 which displays the fraction of ocean volume verses increased CO_2 concentration. The largest increases occur over volumes the size of each simulation's injection grid cell. This again illustrates the importance of increased localized resolution for better resolving details near the injection site. The AMR 1 simulations predict increases in CO_2 concentration that are nearly identical to those predicted by the standard simulation. The AMR 2 simulations provide the highest resolution near the injection site and predict the largest increases in CO_2 concentration at the smallest volume scales.

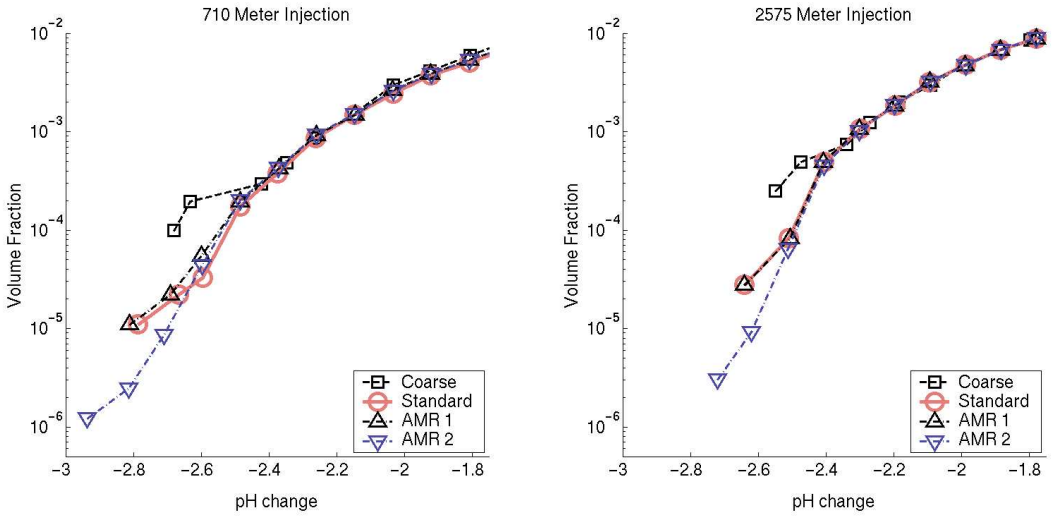


Figure 5.20: The fraction of ocean volume versus pH change. For a given pH change, the fraction displayed is the fraction of the ocean with a pH change at least that large.

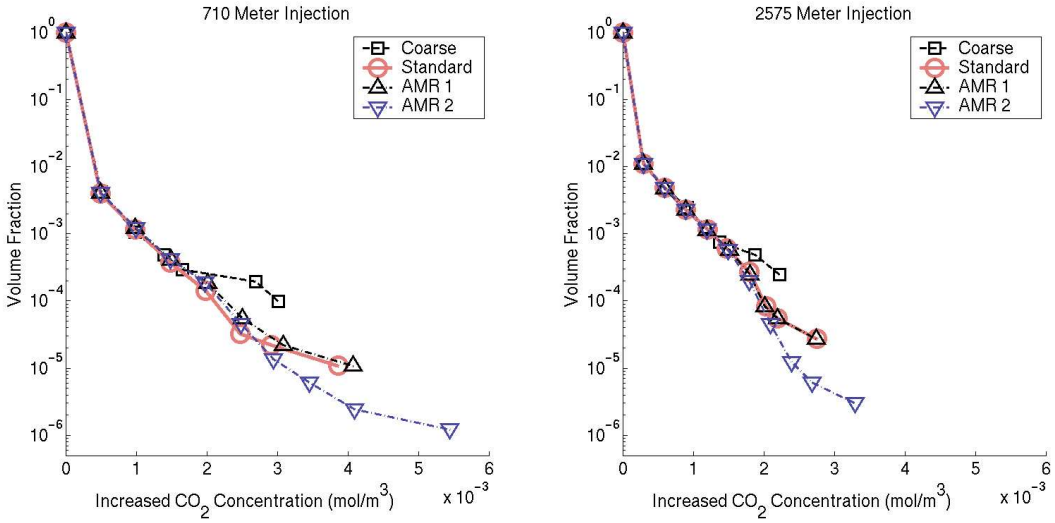


Figure 5.21: The fraction of ocean volume versus increased CO_2 concentration. For a given increase in concentration, the fraction displayed is the fraction of the ocean with an increased at least that large.

5.3.3 Ocean Model Results

Barotropic transports shown in Figure 5.22 are calculated at year 20 as discussed in Section 5.2.3 and reduced to the coarse resolution so that comparisons among various simulations are made across equal volumes. All simulations produce the Gulf Stream circulation with patterns similar to those seen in the spin up states (Figure 5.4). Recall that the AMR refinement levels are constructed with refinement criteria primarily based on CO_2 concentration. More refinement of the Gulf Stream is seen in the shallow injection AMR cases given that the CO_2 propagation is more advective than in the deep injection case. Comparisons of the transport magnitudes show that the AMR cases tend to slightly dampen the maximum barotropic circulation.

Errors are computed of the barotropic velocity as a function of time as discussed in Section 4.4 and are displayed in Figure 5.23. AMR errors are more oscillatory than the coarse errors as anticipated from the limited basin tests in Section 4.4. An AMR transition period near year 0 is seen where high errors quickly fall off before becoming more consistent with time. This indicates the circulation field re-equilibrates during the first 1-2 years after AMR is introduced. Future CO_2 direct injection simulations could avoid using this transitioning circulation field by introducing the initial AMR refinement patches and spinning up the model for two additional years prior to simulating direct injections. The deep injection errors are always less than the coarse simulation after year 6. The shallow injection errors are at times slightly greater than the coarse simulation errors, but no obvious growth trend is apparent. The barotropic circulation of the AMR 1 shallow injection case is therefore more representative of the coarse simulation than it is of the standard simulation.

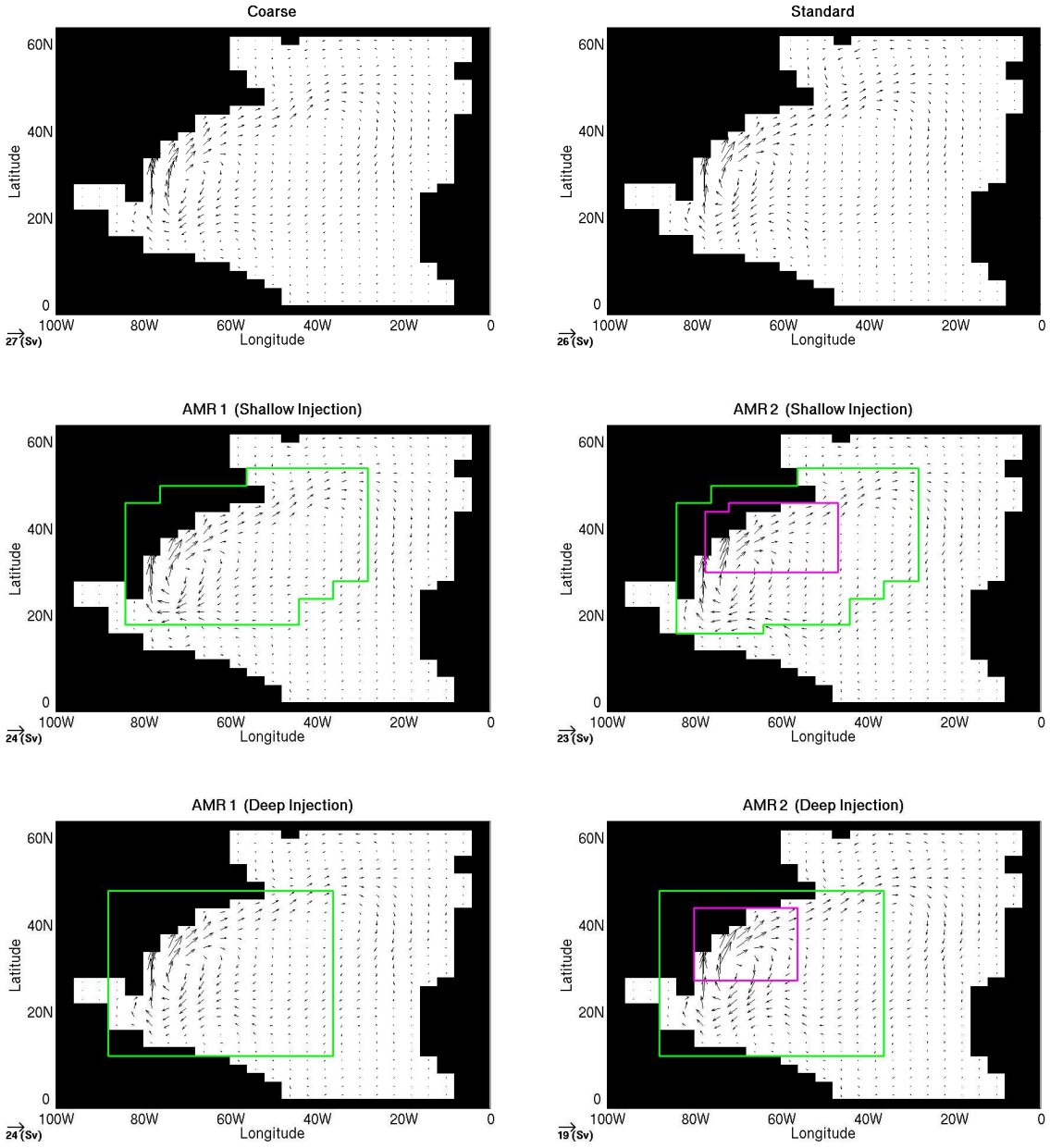


Figure 5.22: Barotropic Transports at year 20 for the AMR simulations.

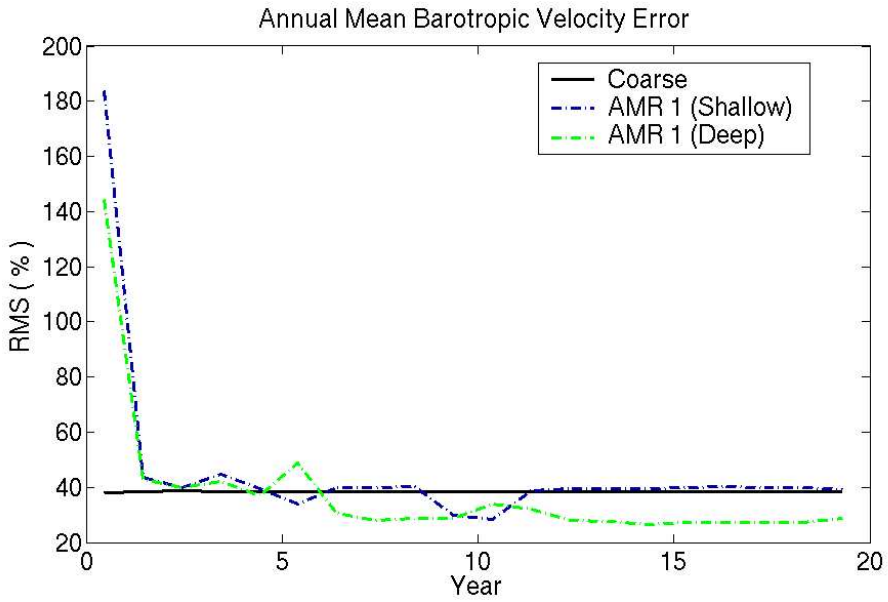


Figure 5.23: Annual mean barotropic velocity error verses time for the sequestration phase of the North Atlantic Basin simulations.

Depth profiles of the annual mean temperature and salinity at year 20 are displayed in Figure 5.24. The deep ocean means are close to Levitus in all simulations. Shallow depths are consistently warmer than Levitus and are under freshed by various amounts. The standard simulation salinity means are closer to the Levitus means at shallow depths than are the coarse or AMR simulations. The coarse simulation is the least fresh and the AMR means are closer to the standard than are the coarse means.

The annual and zonal mean temperatures at year 20 are displayed in Figure 5.25, and show very little discrepancy among the different simulations. Salinity annual and zonal means at year 20 have slightly more deviation as seen in Figure 5.26. The coarse simulation has a build up of salt near the southern boundary and the standard simulation is the least fresh at the mid-latitudes of the domain. The AMR simulations have under freshed regions that stretch from the southern domain boundary to the mid-latitudes of the domain. The

temperature and salinity zonal means show similarities to their respective zonal means at spin up (Figure 5.7 and Figure 5.8). The fields therefore have a relatively consistent annual cycle for the two decades of simulated CO_2 direct injection.

The RMS errors at year 20 are computed of the velocity, temperature, and salinity fields for each depth layer and are displayed in Figure 5.25. Errors in the deep ocean are much smaller than those seen near the surface. The AMR 1 circulation errors are less than the coarse circulation errors near the surface, and are slightly larger than the coarse circulation errors in the deep ocean. Temperature and salinity errors have local maxima just below the surface which are slightly larger in the AMR cases. Outside the local maxima, the AMR 1 temperature and salinity errors are less than the coarse errors.

The temperature and fresh water transports are calculated at year 20 as discussed in Section 5.2.4 and reduced to the coarse simulation to make comparisons across equal volumes. Figure 5.28 shows the meridionally summed transports of barotropic volume, heat, and fresh water for all tested simulations. The largest transports occur near 75° west longitude which corresponds to the longitudinal coordinate where the magnitude of the Gulf Stream is the strongest. The AMR transports of temperature and salinity in this region show improvements over the coarse simulation. The maximum AMR barotropic transports in the region are slightly less than the coarse and standard transports.

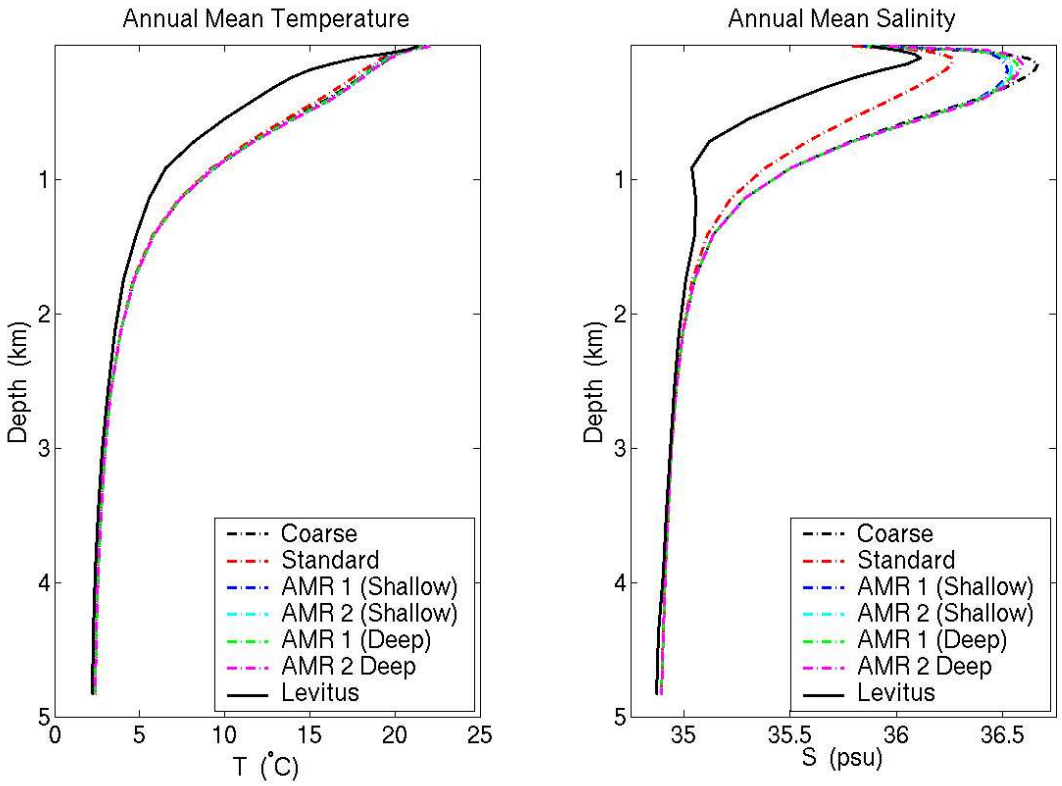


Figure 5.24: Depth profiles of the annual mean temperature (left) and salinity (right) at year 20.

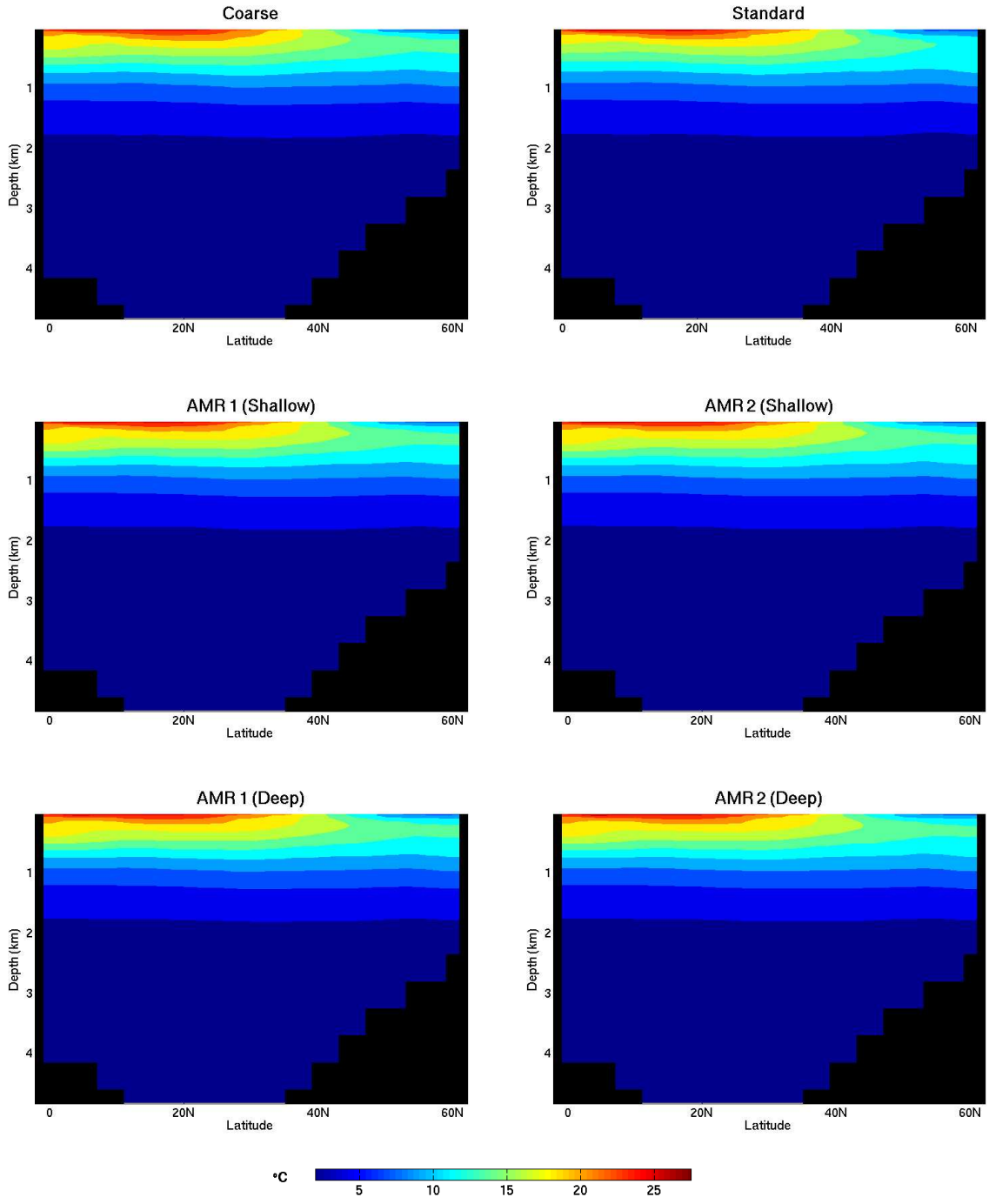


Figure 5.25: Annual and zonal mean temperature verses depth at year 20.

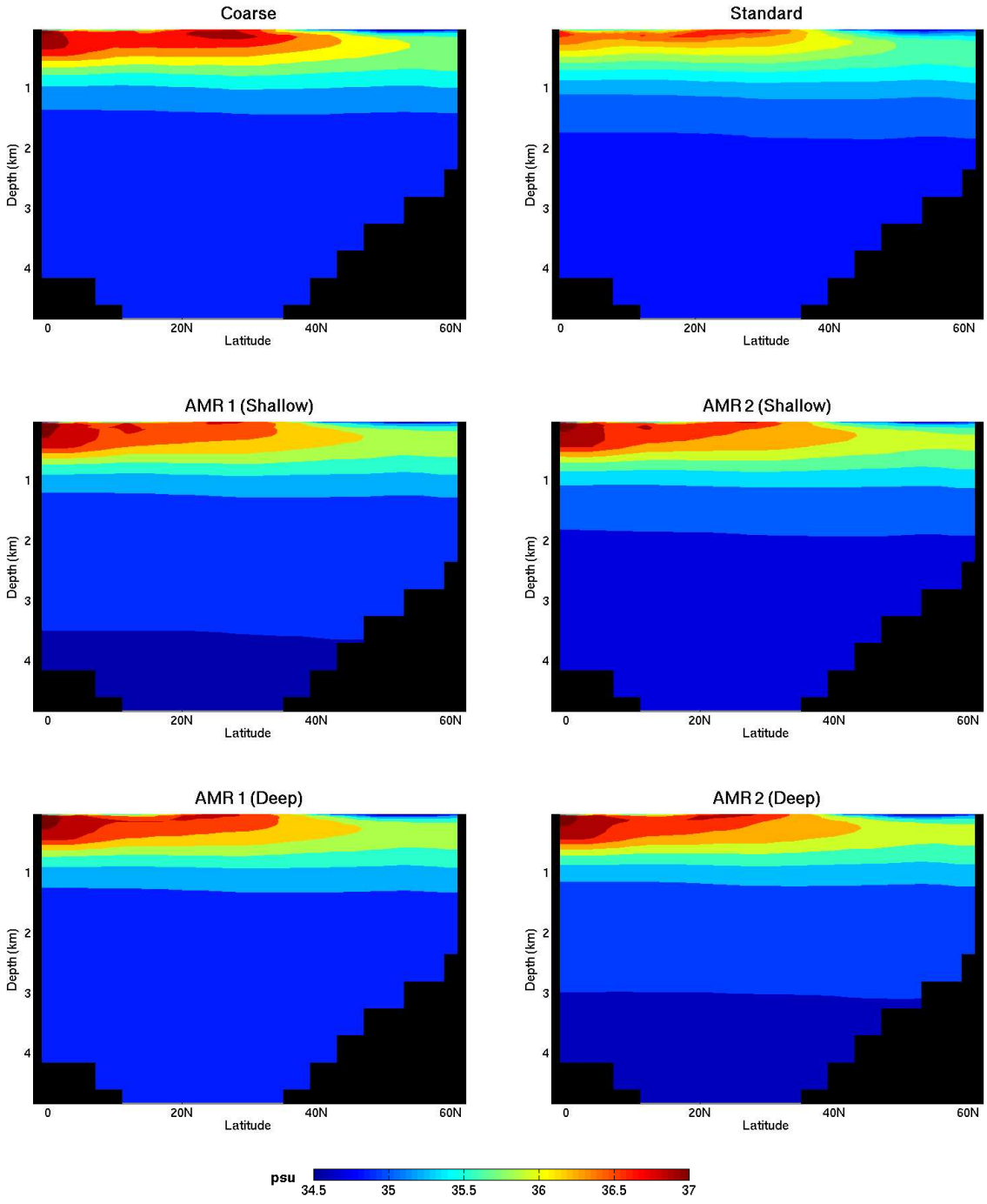


Figure 5.26: Annual and zonal mean salinity verses depth at year 20.

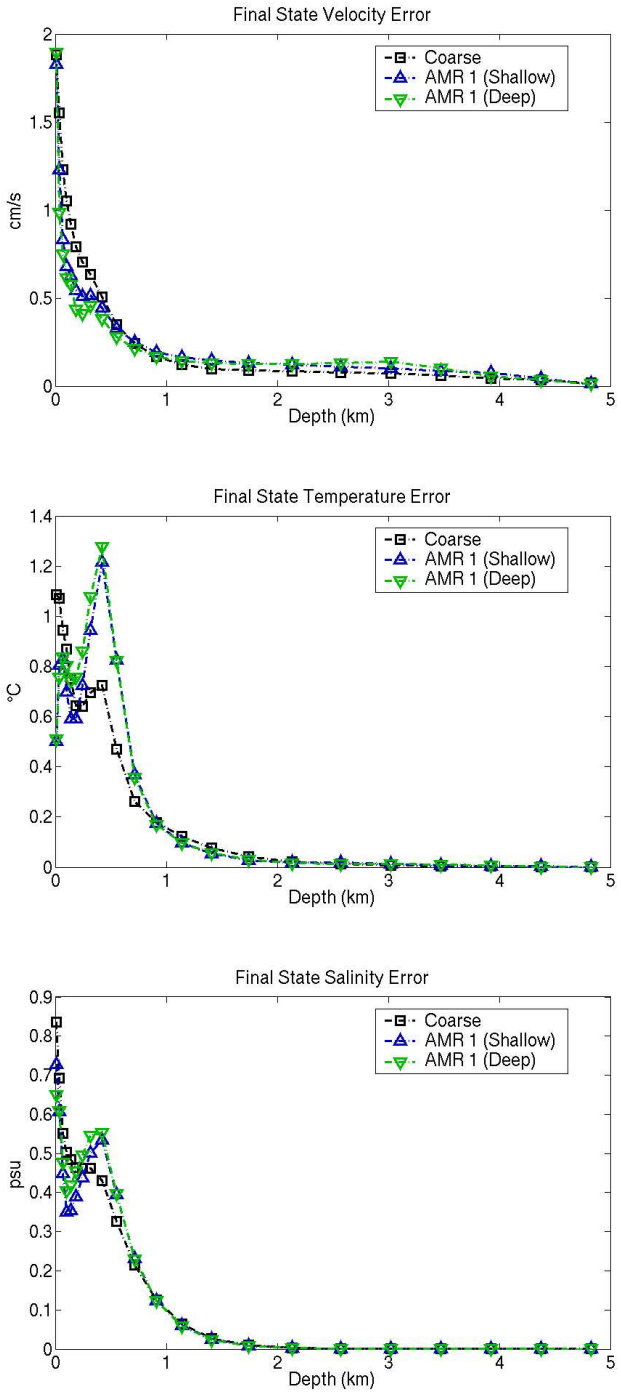


Figure 5.27: Depth profiles of errors in the total velocity field (top), temperature (middle), and salinity (bottom) at year 20.

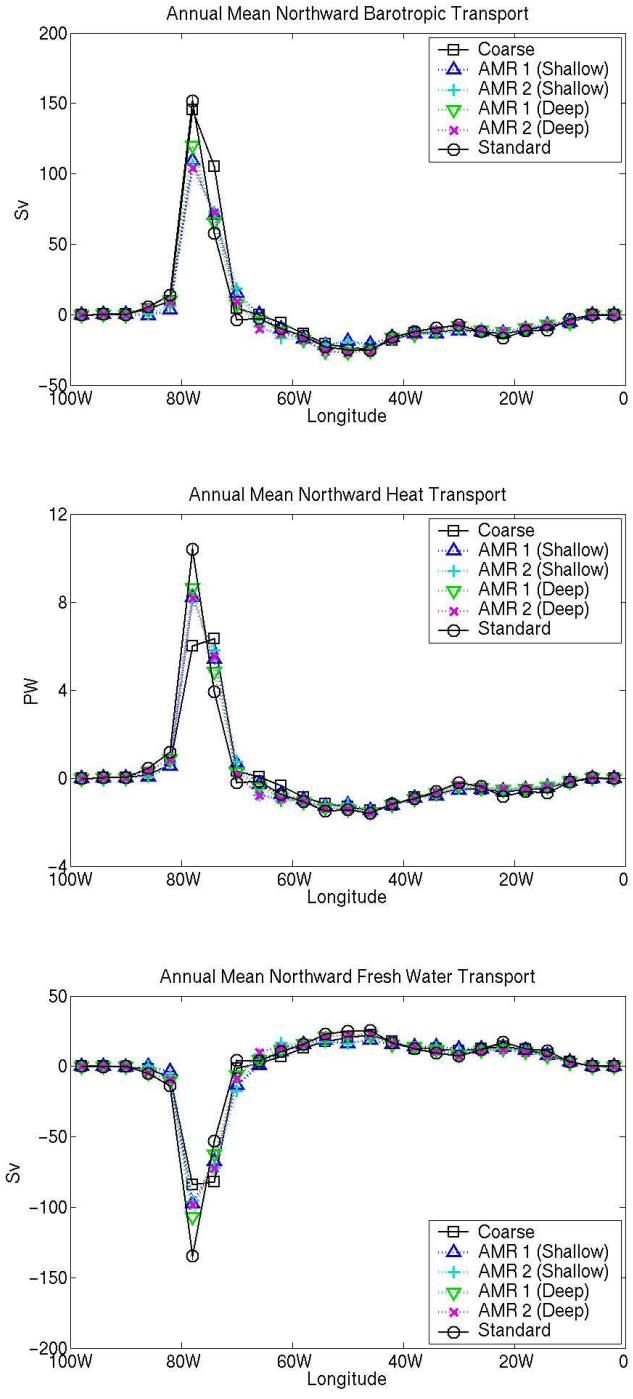


Figure 5.28: Meridional sums of the annual mean northward transports of barotropic volume (top), heat (middle), and freshwater transport (bottom) at year 20.

5.4 Run Times

The AMR Ocean Model has parallel processing capabilities with processor communication invoked using standard message-passing interfacing (MPI) as described in Section 6.3. The domain is decomposed and load balanced using functionality built into the SAMRAI library. Only the standard resolution simulations in this study were conducted in parallel and used 8 processors. All other simulations were ran in serial. Comparisons of simulation rates and run time increases relative to the coarse simulation are given in Figure 5.29. The theoretical increase in run time for a given simulation is the ratio of its total computations per CPU relative to that of the coarse simulation. For a simulation with an arbitrary number of refinement levels, the total computations per CPU is defined here as

$$\text{COMP} \quad = \quad \sum_{\ell,r} \; 27 \; \left(\frac{\Delta t_0^{\text{nonBT}}}{\Delta t_\ell^{\text{nonBT}}} \right) \; (23 + \text{substeps}_\ell) \; \left(\frac{N_{\ell,r}}{p} \right) \tag{5.20}$$

where 27 is the regrid frequency, 23 is the number of vertical layers in the domain, $N_{\ell,r}$ is the number of cells on level-(ℓ) after the r^{th} regrid, and p is the number of processors used in the simulation.

The standard and AMR simulations have longer run times than predicted due to overhead associated with the model. Each AMR simulation is within a factor of 2 of its predicted increase in run time, and the standard simulation is a factor of 2.8 longer than predicted. The standard theoretical run time is less than the AMR run times due to the use of parallel processing by the standard simulations. The AMR 1 deep injection case has very little overhead and a shorter actual run time than the standard case as a result. Even though the AMR simulations in this study typically have run times longer than the standard simulation, they still have a run time advantage over the standard case given their

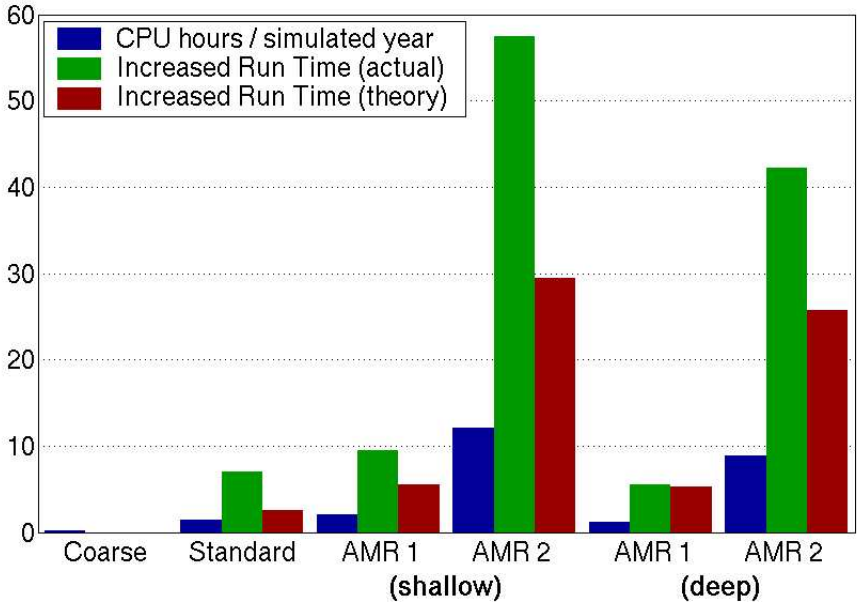


Figure 5.29: Simulation run times expressed in CPU hours per simulated year, and the actual and theoretical increases in run time relative to the coarse simulation run time.

use of a coarse resolution spin up state. The increased AMR run times are only present during the 20 year direct injection simulations following the spin up simulation. The run times of the AMR simulations can be reduced in future studies by refining smaller portions of the domain. This should be done with caution given that the results presented in this study may not be applicable to dramatic changes made in the refinement criteria.

5.5 Summary of North Atlantic Simulations

CO_2 sequestration simulations were presented of the North Atlantic using two uniform resolutions and two AMR resolutions. Continuous injection is simulated for 20 years near New York at injection depths of 710 meters and 2575 meters. The simulations do not resolve mesoscale eddies, and the same physical parameters are used for each. AMR refinement criteria is based on injected CO_2 mass and barotropic velocity gradients near injected CO_2 . The passive tracer representing sequestered CO_2 is more localized in the deep injection simulations which allows simplified refinement. The global RMS errors show that AMR resolutions with one level of refinement give a significant increase in accuracy over the coarse resolution. AMR global RMS errors slowly increase with time, but are significantly lower than the coarse global RMS errors after 20 simulated years of continuous injection.

AMR with one refinement level predicts close to the same pH changes and increases in CO_2 concentration as the standard resolution at both large and small volume scales. AMR with two levels of increased refinement provides a means of predicting details of pH changes and increased CO_2 concentration at scales smaller than are capable by the coarse and standard resolutions. Simulations with AMR show improvements over the coarse resolution in surface outgassing of CO_2 even though refinement is not always oriented in regions of high surface flux. All deep injection simulations predict a CO_2 retention of 99.9% after 20 years of continuous direct injection. AMR provides a more accurate prediction of CO_2 retention than the coarse resolution in the shallow injection case. The coarse predicted retention is within 1% of the standard prediction of 97.61%.

AMR applied with the primary intent of increasing resolution about the passive tracer has positive and negative effects on the ocean state. A transition period occurs just after AMR is introduced which lasts for roughly one simulated year before the ocean once again becomes steady with time. Barotropic circulation is more accurate in the deep injection cases. Little change is observed at year 20 in the coarse and AMR mean temperatures relative to the standard means. The AMR mean salinities near the surface are slightly improved over the coarse mean salinities. Comparisons of errors in the final state velocity, temperature, and salinity fields show that reductions in circulation error occur at depths where state tracer errors increase, and vice-versa. Calculations of northward heat and freshwater transports show improvements with AMR near the intensified western boundary current. AMR barotropic volume transports are slightly damped near the western boundary current. In general, AMR as applied here offers the possibility of improving circulation, and the instances where errors increase are not observed as dramatic nor persistent.

This study is a first step towards global ocean CO_2 direct injection simulations using AMR with multiple injection sites. The deep injection case in particular provides a highly localized solution that is well captured by refinement. Results may vary for injections at other locations on the globe due to different topography, circulation, and surface forcing. AMR simulations have the advantage of placing high resolution around injection sites, and can be initialized with a coarse resolution spin up. The run time performance of the AMR Ocean Model could be improved with more investigation of parallel processing. Further testing of refinement criteria could also improve performance and/or numerical accuracy.

Chapter 6

Simulations of the Global Ocean

6.1 Spin Up Simulations

Simulations are performed of the Global Ocean with model parameters similar to those applied in the North Atlantic basin simulations. The domain spans 360° longitude with a periodic boundary condition at the eastern and western domain edges. The southern pole is naturally covered by land and the southern domain boundary is set to 80° south latitude. The time scales of the Global Ocean simulations are short enough to ignore the relatively small influences of the Arctic Ocean. The northern domain boundary is set to 80° north and grid cells above the Bering Sea ($60° < \lambda < 260°, \phi > 65°$) are modeled as land. This approach of ignoring latitudes near the poles ($|\phi| > 80°$ in this case) is often used in models with latitude-longitude based grids to decrease the total number of grid cells in the domain and to increase the maximum stable time step of the simulation.

The domain resolution is $4° \times 2°$ (coarse) in the longitude and latitude directions, respectively. Depth is discretized according to the depth layers shown in Table 5.1. The

topography (Figure 6.1a), surface wind stress (Figure 6.1b), and restoring fields for temperature and salinity are defined from observed data using the procedures discussed in Section 5.2.1. All physical parameters discussed in Section 5.2.1 are also applied in the global simulations.

The global ocean grid contains 9.375 more grid cells than the North Atlantic coarse resolution grid. This necessitates parallel processing to achieve results with reasonable execution times. The number of processors implemented in the global ocean simulations is based on the optimal run time rates given in Section 6.3. A non-AMR simulation with resolution equal to $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ has an optimal run time rate of approximately 3.9 simulated years per day. Approximately 28 days of CPU time is required for a 100 year spin up simulation followed by a 10 year CO_2 sequestration simulation. Several months of real time is necessary to complete such tasks due to additional time spent waiting for available computing resources. This time scale is impractical and motivates the use of alternate simulation techniques like AMR that drastically reduce run time. It is for this reason that a simulation with uniform resolution equal to $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ or finer is not conducted in the global ocean simulations. It is therefore assumed that the results obtained in the North Atlantic simulations in Chapter 5 are applicable on a global scale.

6.1.1 Time Advancement

The basin is initialized at rest with temperature and salinity fields (Figure 6.3) spatially interpolated to the Levitus annual means. The model is advanced for 100 simulated years using time steps shown in Table 6.1. Acceleration techniques for advancing tracers, such as time step splitting and deep ocean acceleration, are not implemented. The

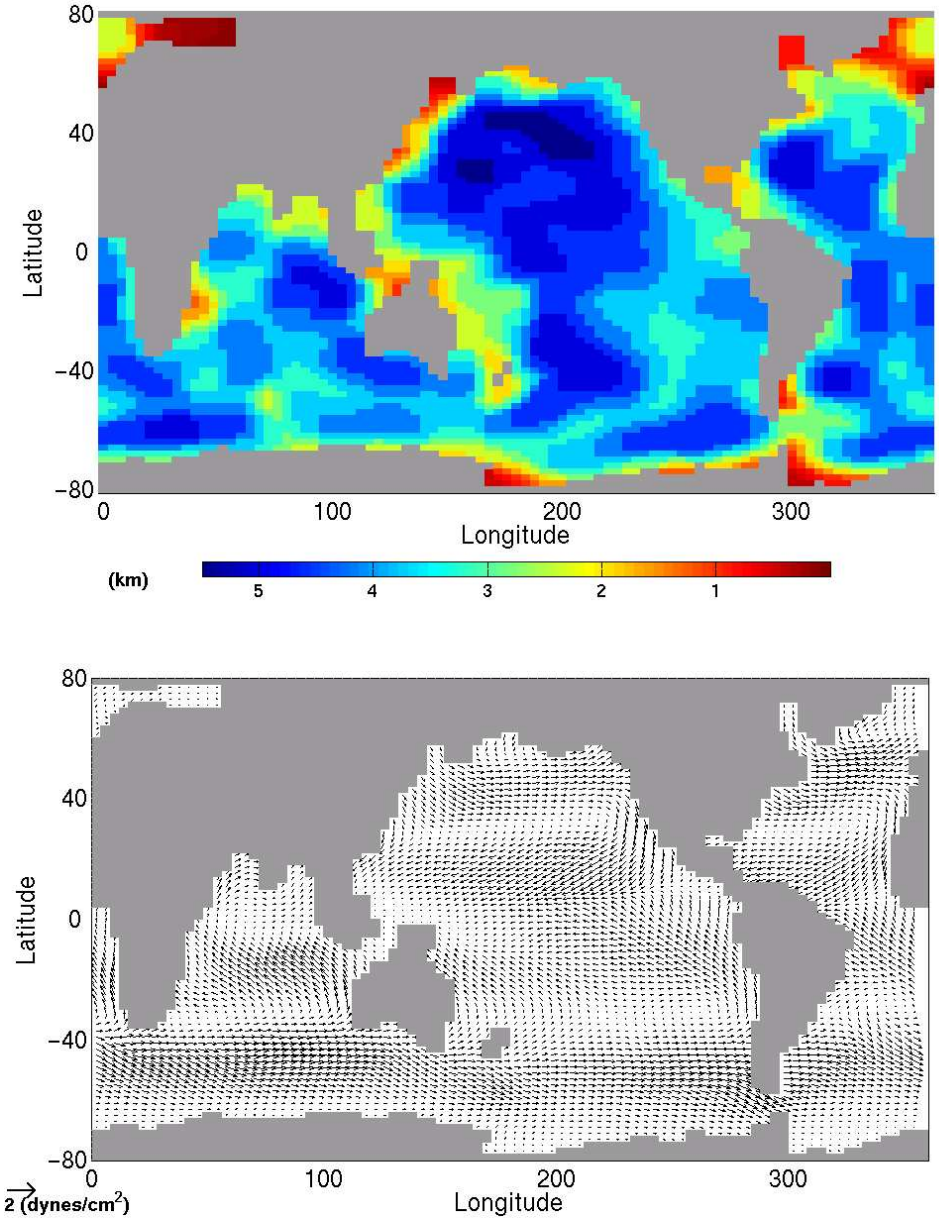


Figure 6.1: T cell topography (top) and U cell annual mean wind stress (bottom) used in the coarse resolution Global Ocean simulations.

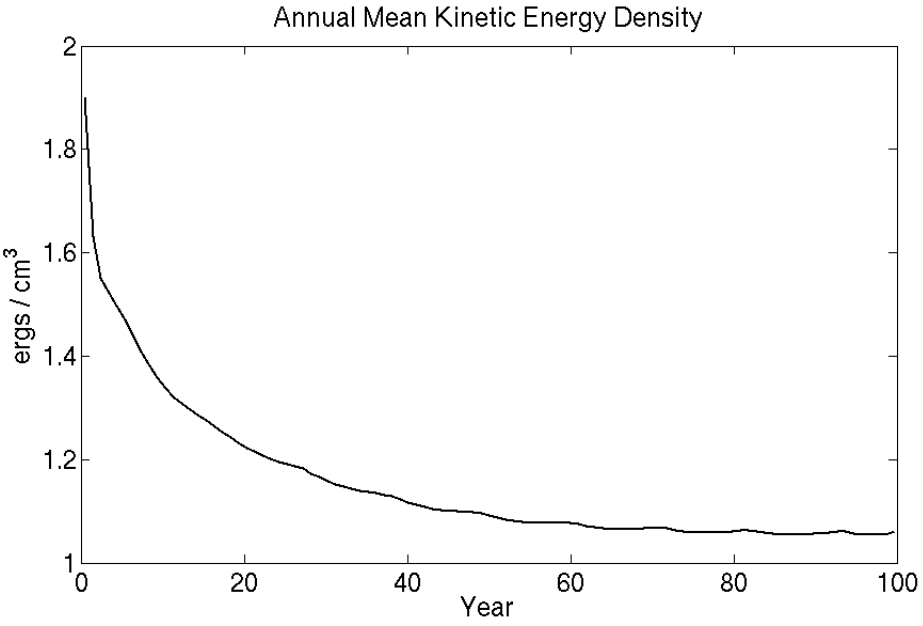


Figure 6.2: Annual mean kinetic energy density as a function of time during the spin up phase of the Global Ocean simulations.

simulation uses 8 processors to achieve the optimal run time rate discussed in Section 6.3. AMR is not applied during spin up since the limited basin benchmark tests (Section 4.4) showed that large portions of the domain would require refinement to produce a globally accurate flow field. The annual mean kinetic energy density seen in Figure 6.2 shows a relatively steady annual circulation cycle by year 100.

Resolution	Δt^{BC} (minutes)	Δt^{BT} (minutes)	Subcycles ($\Delta t^{\mathrm{BC}}/\Delta t^{\mathrm{BT}}$)
coarse, level-(0)	306	3.4	90
level-(1)	102	1.7	60
level-(2)	11.3333	0.56666	20

Table 6.1: Time steps used in AMR and non-AMR simulations.

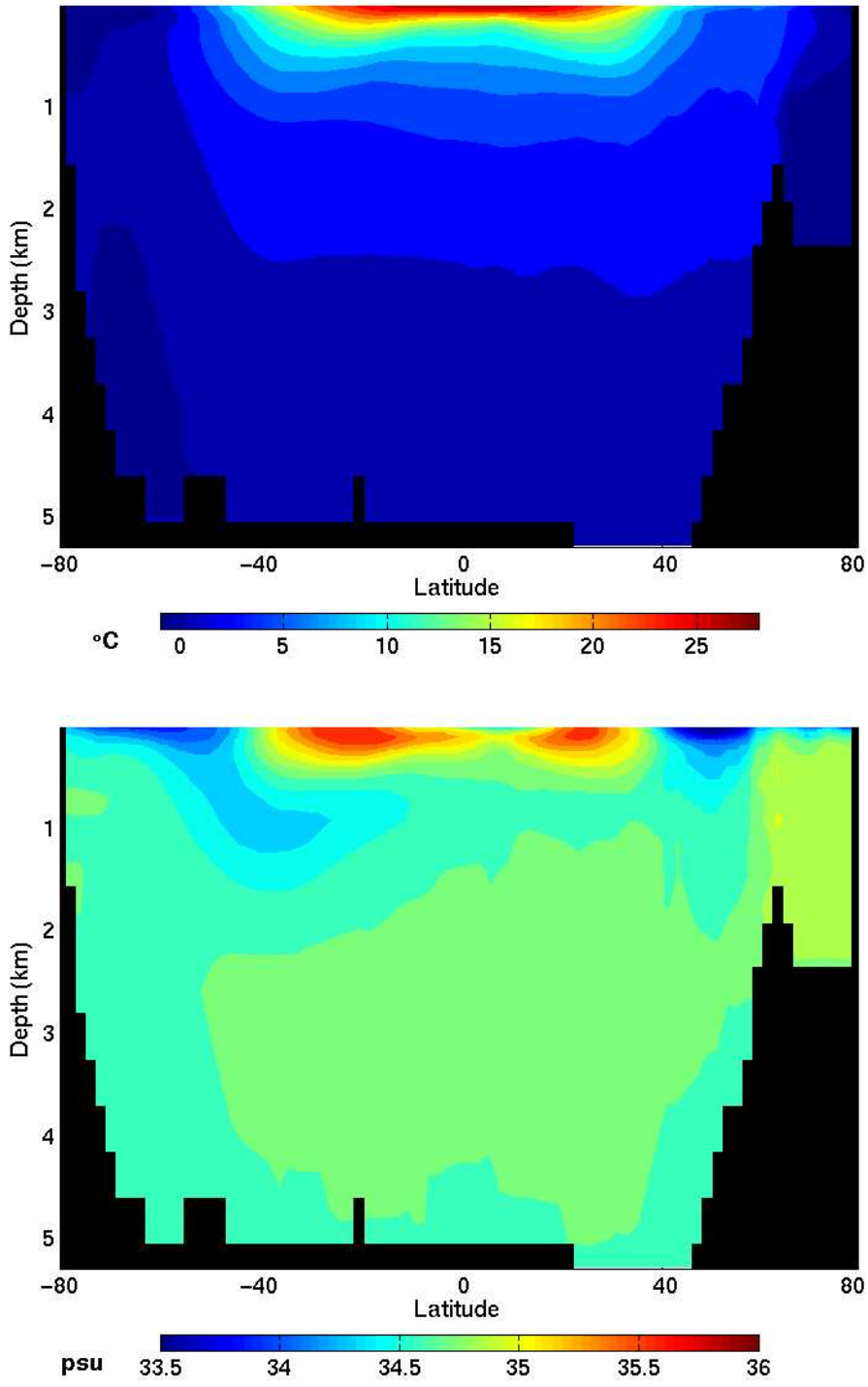


Figure 6.3: Zonal mean initial temperature (top) and salinity (bottom) for the Global Ocean simulations.

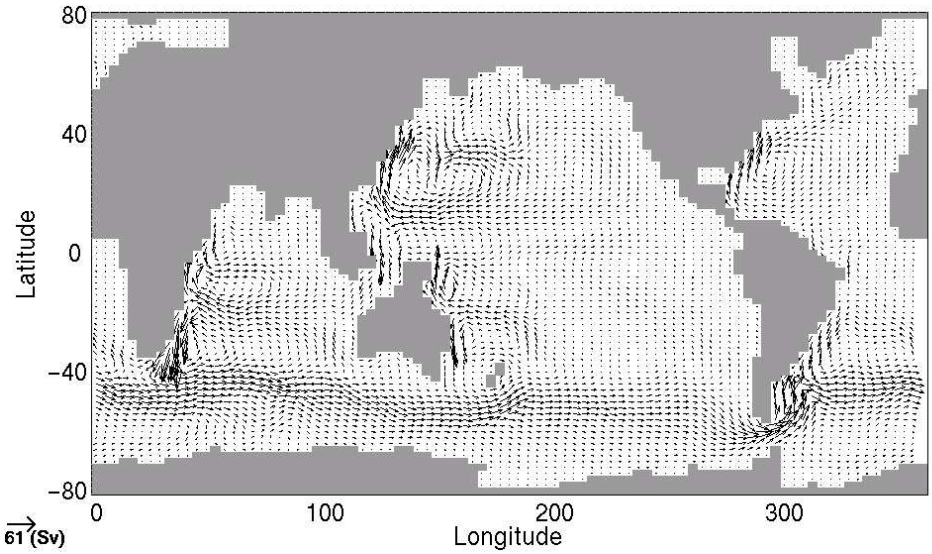


Figure 6.4: Barotropic transport at year 100 for the Global Ocean simulations.

6.1.2 Barotropic Transport at Spin Up

The barotropic transport at spin up is calculated as discussed in Section 5.2.3 and is shown in Figure 6.4. Several gyres are seen with strong western boundary currents. The Antarctic Circumpolar current is visible in the Southern Hemisphere with a strong flow through the Drake Passage. The Gulf Stream is seen off the eastern coast of the United States. The Kuriosho current is visible along the eastern coast of Asia, and the Agulhas current is strongest near the eastern coast of South Africa.

6.1.3 Temperature and Salinity at Spin Up

Depth profiles of the annual mean temperature and salinity during the spin up phase are displayed in Figure 6.5. The two resolutions have nearly identical temperature profiles, and the standard resolution is slightly more fresh than the coarse resolution. The

fields tend to increase with time which is also reported by [77] during the first several hundred years of spin up. Bryan-Cox-Semtner ocean models typically require at least one thousand simulated years before achieving true steady states for these fields [22, 25, 24, 77]. The annual mean temperature and salinity in Figure 6.5 are still in transition on time scales of thousands of years, but are relatively steady on the scale of several decades by year 100. These fields are therefore believe to be in equilibrium for this study where a 10 year simulation is desired after spin up.

The annual and zonal mean temperature and salinity at spin up (Figure 6.6) show consistencies between coarse and standard resolutions. However, the fields differ from the Levitus annual means in Figure 6.3. The temperature is too diffused and too warm near the northern and southern boundaries. The salinity fields indicate the domain is in general too salty, especially at shallower depths. This is again attributed to the slow transition of the fields towards a true spin up state. Similar Bryan-Cox-Semtner ocean models produce unphysical temperature and salinity fields even after thousands of simulated years [22, 25, 24, 77]. Thus, little is gained by integrating to such time scales for the applications in this study.

The northward transports of heat and salinity are calculated as explained in Section 5.2.4. Meridional sums of the annual mean transports of barotropic volume, heat, and freshwater are displayed in Figure 6.7 for the final year of spin up. Local extrema coincide longitudinally with the intensified western boundary currents seen in Figure 6.4.

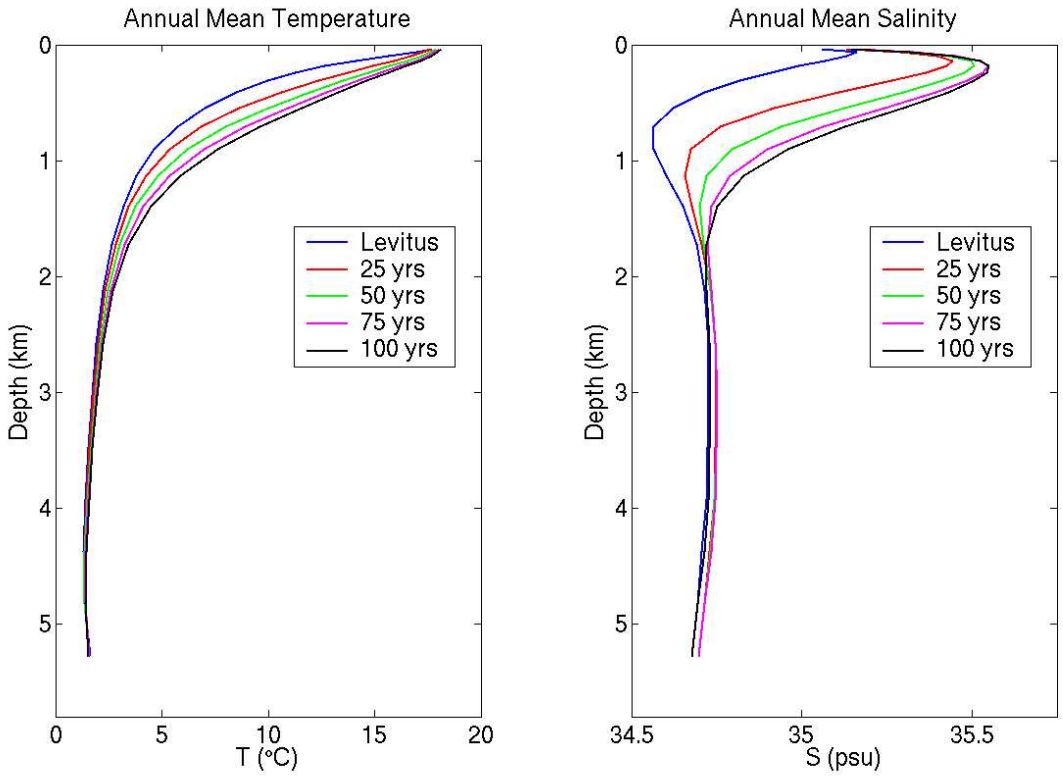


Figure 6.5: Depth profiles of the annual mean temperature (left) and salinity (right) during the spin up phase of the Global Ocean simulations.

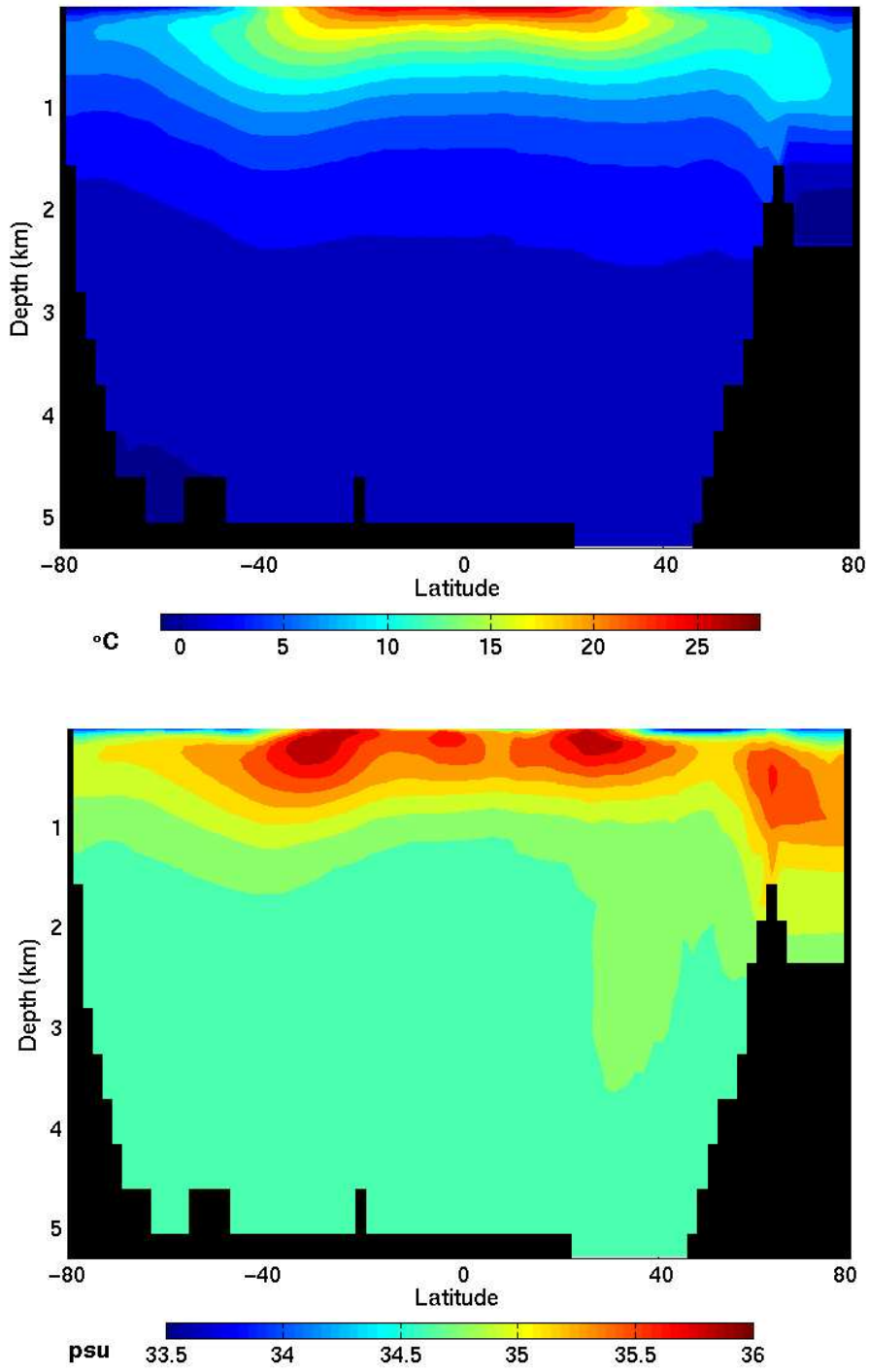


Figure 6.6: Annual and zonal mean temperature (top) and annual and zonal mean salinity (bottom) for year 100 of the Global Ocean simulations.

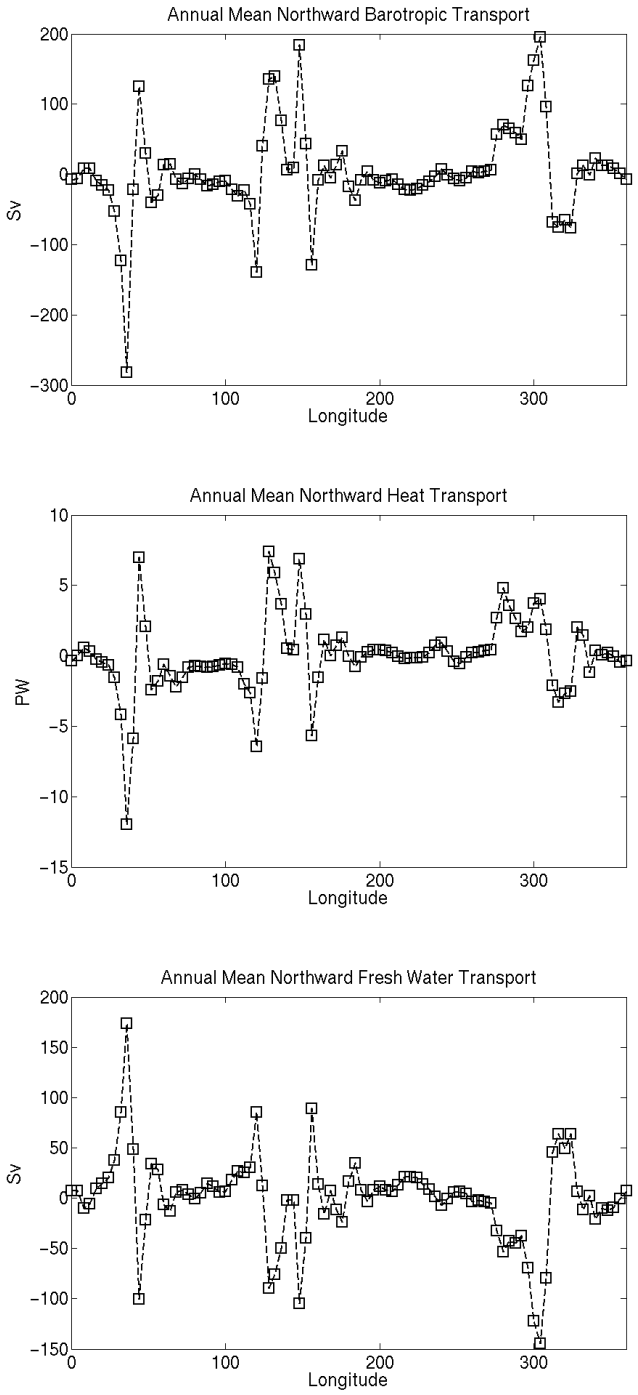


Figure 6.7: Meridional sums of the annual mean northward transports of barotropic volume (top), heat (middle), and freshwater transport (bottom) at year 100 of the Global Ocean simulations.

6.2 CO_2 Direct Injection

CO_2 direct injection is modeled as discussed in Section 5.1 using the standard OCMIP [5] protocol for multiple injections located near New York, San Francisco, the Bay of Biscay, Rio de Janeiro, Tokyo, Jakarta, and Bombay. Each injection is modeled as a single grid cell point source (equation 5.12) with CO_2 continuously injected at a rate of $4 \times 10^{12} \frac{g}{yr}$. "Shallow" and "deep" injection scenarios are modeled with the injection depth set to 710 meters and 2575 meters, respectively.

CO_2 direct injection is modeled at the coarse and AMR resolutions with refinement by factors of 3 in the horizontal directions. An AMR case referenced as "AMR 1" is modeled with level-(0) and level-(1) resolutions equal to the coarse and $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ resolutions, respectively. A second AMR case referenced as "AMR 2" has refinement levels level-(0) and level-(1) with resolutions equal to the coarse and $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ resolutions, respectively, and a second refinement level with resolution equal to $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$. The ocean state of the coarse simulation is initialized as the spin up state. The AMR simulations are initialized as the coarse spin up state on level-(0), and spatial interpolations of the coarse spin up state on finer levels.

Time steps are set to those shown in Table 6.1. The AMR time refinement ratios between level-(0) and level-(1) are 2 and 3 for the barotropic and non-barotropic systems, respectively. The time refinement ratios between level-(1) and level-(2) are 3 and 9 for the barotropic and non-barotropic systems, respectively. The time steps are numerically stable and satisfy the AMR subcycling rule given by equation (3.26). The interface smoother in equation (4.13) is applied to the AMR interfaces of the surface height field. The interface

normal component of the smoother is applied to temperature and salinity fields at AMR interfaces in the top 2 depth layers.

Longitude-latitude based grids such as the one used here increase resolution as grid cell latitudes approach the poles. The model's largest stable time step is inversely proportional to the smallest grid cells. AMR refinement levels finer than level-(0) are able to use larger time steps by not refining regions near the poles. Refinement is only implemented in AMR simulations of the global ocean between 60° south and 60° north ($|\phi| < 60°$). This increases the maximum stable time steps on levels finer than level-(0) by factors of 1.3 and 4.2 for the barotropic and baroclinic systems, respectively. Coarse simulations show that the majority of injected CO_2 stays within the bounds of these latitudes. Also, the resolution of level-(0) grid cells at latitudes $|\phi| > 60°$ is at least twice as fine as the grid cells at the equator. Thus, the need for refinement at latitudes where $|\phi| > 60°$ is less crucial.

AMR simulations are advanced using 8 processors. Parallel processing of refinement level communication (e.g., refinement, coarsening) is determined by SAMRAI. The number of processors used during the time step advance varies for each refinement level. Advances on level-(0) utilize 8 processors. Advances on level-(1) utilize 4 processors when refining less than 30% of the tracer domain, and 8 processors otherwise. Advances on level-(2) use 4 processors when refining less than 4% of the tracer domain, and 8 processors otherwise. These thresholds are based on the performance test results given in Section 6.3.

Initial refinement patches are positioned ad hoc on levels finer than level-(0). Patches on level-(1) refine 5 grid cell by 5 grid cell sections of level-(0) centered about the injection sites. Similar initial refinement patches are positioned on level-(2). Additional

level-(1) patches are placed about strong western boundary currents near injection sites. The AMR hierarchy is regridded every 27 level-(0) time steps using refinement criteria identical to that applied in the North Atlantic simulations (see Section 5.3.1). Cells on level-(ℓ) are first tagged based on CO_2 mass. Steep barotropic velocity gradients in the vicinity of the injected CO_2 are next tagged for refinement. The tags are clustered by SAMRAI and a new set of refinement patches is created on level-($\ell + 1$).

6.2.1 Sequestration Results

Column inventories of CO_2 at year 10 are shown in Figure 6.8 for the shallow injection case and in Figure 6.9 for the deep injection case. The first and second refinement levels in the AMR cases are outlined in green and magenta, respectively. The shallow case is more advective due to its proximity to stronger surface currents. The passive tracer propagation is primarily diffusive in the deep injection case because of weaker currents in the deep ocean. The tracer distribution in the deep case is more localized which allows AMR regrids to construct simpler refinement patch clusters refine more CO_2 .

Spatial distributions of CO_2 surface fluxes at year 10 are shown in Figure 6.10 for the shallow injection case and in Figure 6.11 for the deep injection case. Positive flux represents transfers of CO_2 from the ocean to the atmosphere. The shallow injection surface fluxes are significantly higher than the deep injection surface fluxes as expected [17]. The largest surface fluxes are seen in the North Atlantic and near Antarctica. This is expected given the strong influences of ocean currents in these regions as well as large amounts of vertical convective mixing. The AMR deep injection cases show larger surface fluxes than the coarse resolution. Very little CO_2 escapes back to the atmosphere after 10 years of

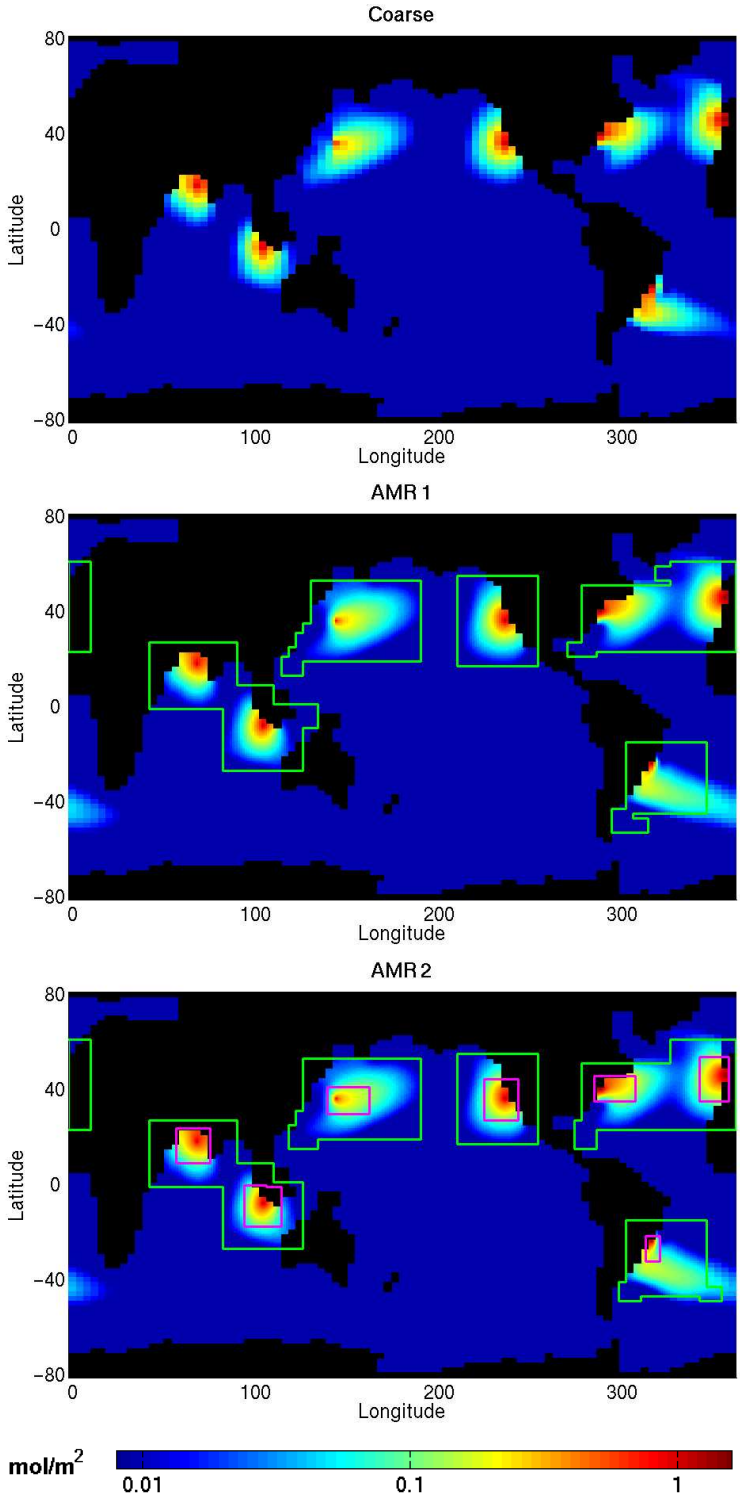


Figure 6.8: Vertically integrated CO_2 after 10 years of continuous injection at a depth of 710 meters. Note the logarithmic scale.

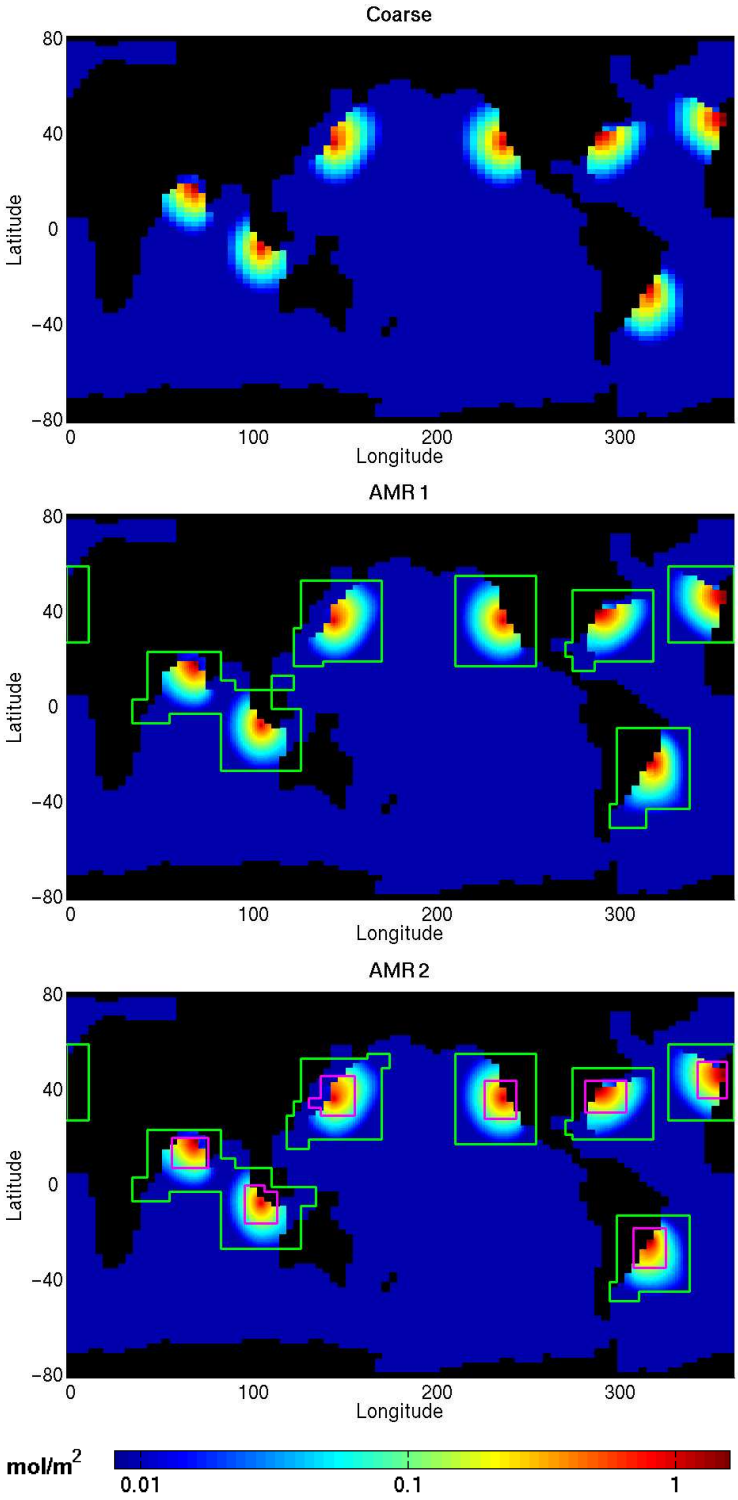


Figure 6.9: Vertically integrated CO_2 after 10 years of continuous injection at a depth of 2575 meters. Note the logarithmic scale.

continuous injection due to the relatively small magnitudes in surface fluxes. A retention of 99.99% is calculated for all simulations.

The biological impact of CO_2 sequestration is assessed through pH changes resulting from the increase in ocean acidity. As explained in Section 5.1, the maximum possible pH changes are reported given that the AMR Ocean Model is not equipped with a means of modeling background pH. Future studies of pH changes could also be calculated relative to data set estimations of ocean carbonate chemistry [55]. Figure 6.12 shows the fraction of ocean volume verses the maximum calculated pH change for each simulation in the shallow and deep injection cases. For a given pH change, the volume fraction displayed is the fraction of ocean volume with a pH change at least that large.

Figure 6.12 represents about 0.3% of the total volume in the ocean basin (1.68×10^{18} m^3). All simulations agree well at large volume scales, but predictions become more dependent on resolution when looking at volume scales approaching that of the injection grid cell. The AMR 2 simulations predict larger maximum pH changes than are possible with the other tested simulations. This is because CO_2 is injected into a smaller grid cell volume giving a higher CO_2 concentration which results in a higher change in pH.

Similar trends are seen in Figure 6.13 which displays the fraction of ocean volume verses increased CO_2 concentration. The largest increases occur over volumes the size of each simulation's injection grid cell. This again illustrates the importance of increased localized resolution for better resolving details near the injection site. The AMR 2 simulations provide the highest resolution near the injection site and predict the largest increases in CO_2 concentration at the smallest volume scales.

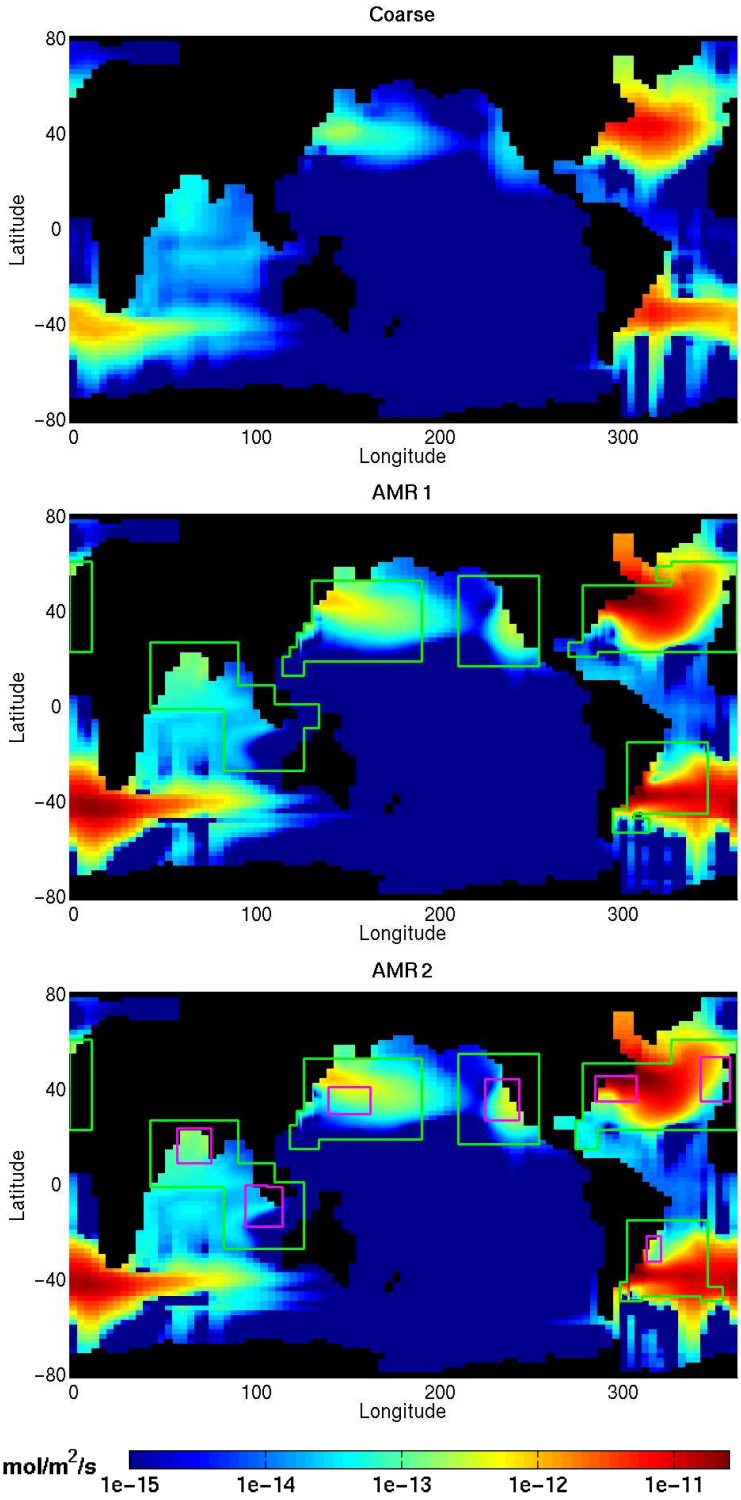


Figure 6.10: Surface Flux of CO_2 after 10 years of continuous injection at a depth of 710 meters. Note the logarithmic scale.

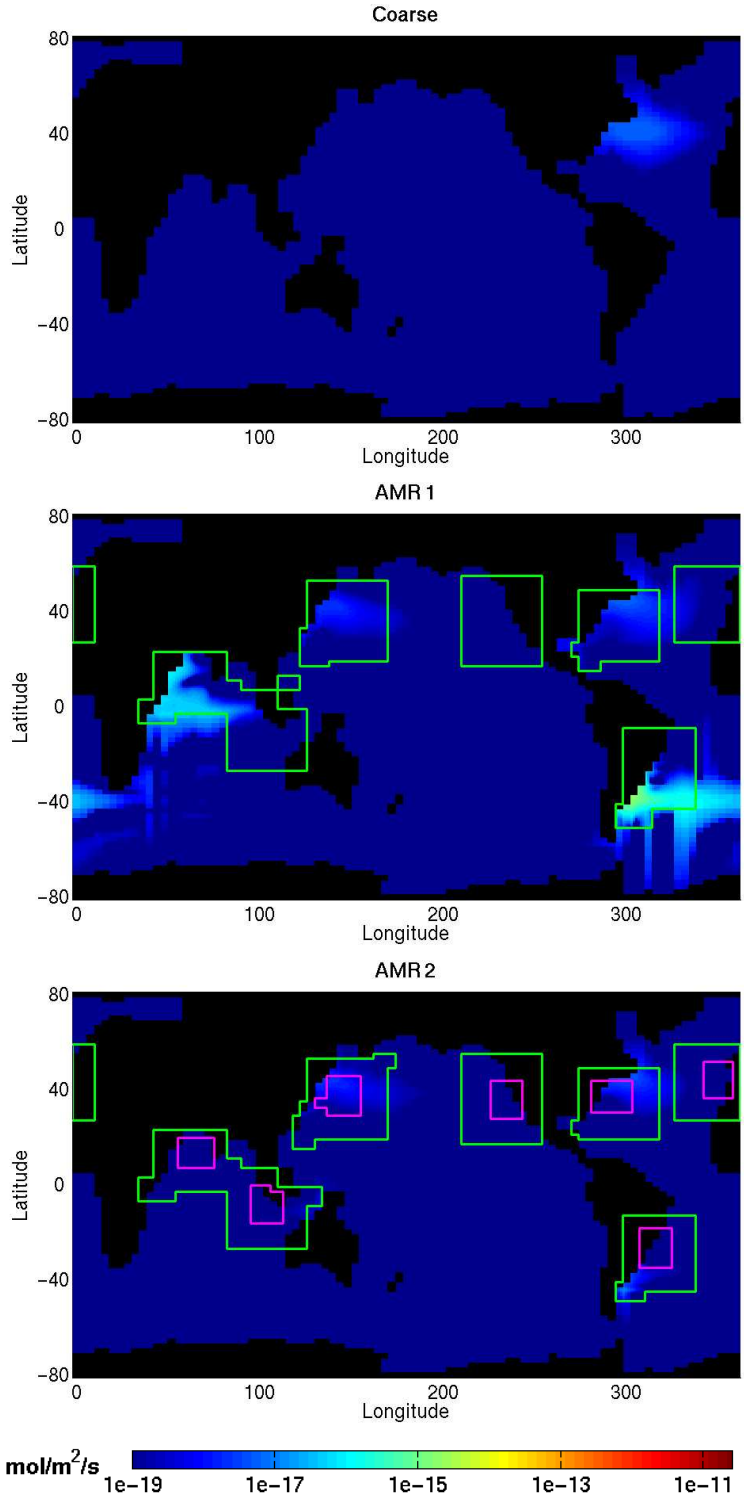


Figure 6.11: Surface Flux of CO_2 after 10 years of continuous injection at a depth of 2575 meters. Note the logarithmic scale.

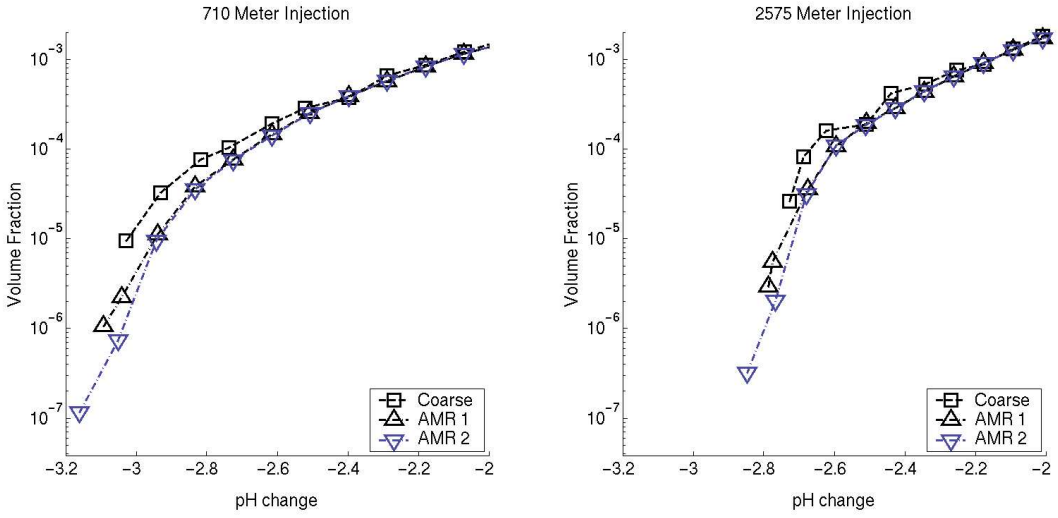


Figure 6.12: Volume of ocean versus pH change. For a given pH change, the volume displayed is the amount of volume with a pH change at least that large.

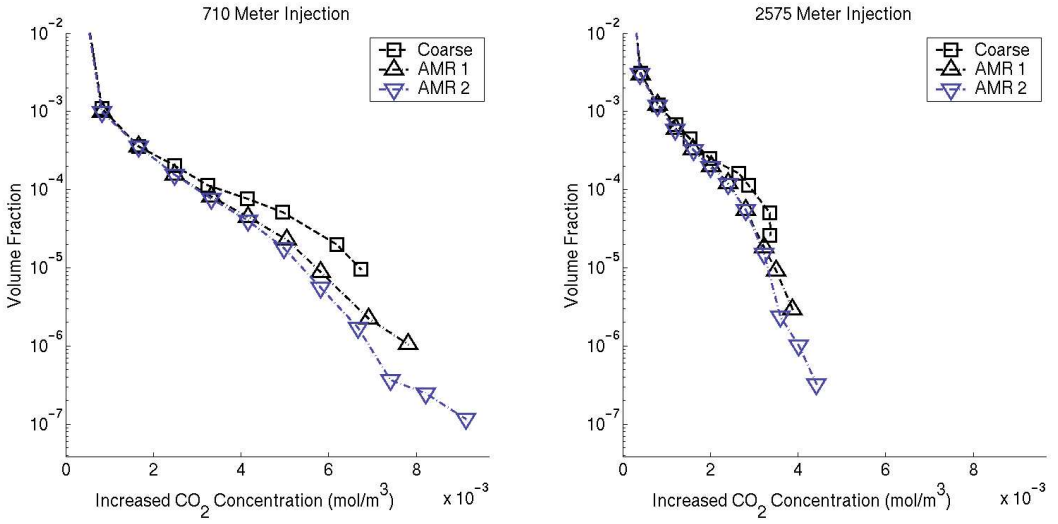


Figure 6.13: Volume of ocean versus increased CO_2 concentration. For a given increase in concentration, the volume displayed is the amount of volume with an increase at least that large.

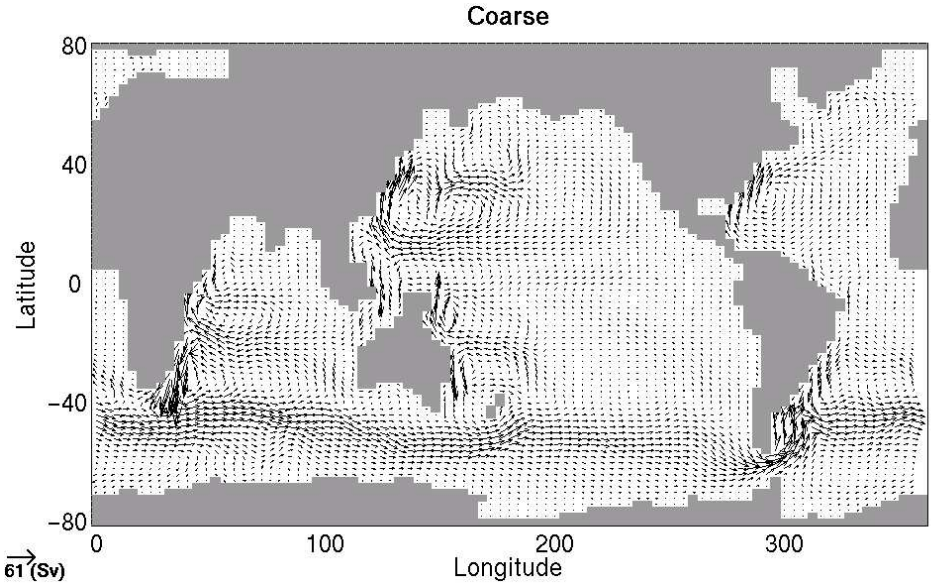


Figure 6.14: Barotropic Transport of the coarse resolution at year 10.

6.2.2 Ocean Model Results

Barotropic transports shown in Figure 6.14 and Figure 6.15 are calculated at year 10 as discussed in Section 5.2.3 and reduced to the coarse resolution so that comparisons among various resolutions are made across equal volumes. All resolutions produce circulation patterns similar to the spin up circulation (Figure 6.4). Recall that the AMR refinement levels are constructed with refinement criteria primarily based on CO_2 concentration. More refinement is seen in the shallow injection AMR cases given that the CO_2 propagation is more advective than in the deep injection case. Refinement of intensified western boundary currents near the injection sites is also seen.

Depth profiles of the annual mean temperature and salinity at year 10 are displayed in Figure 6.16. The deep ocean means are close to Levitus in all resolutions. Shallow depths

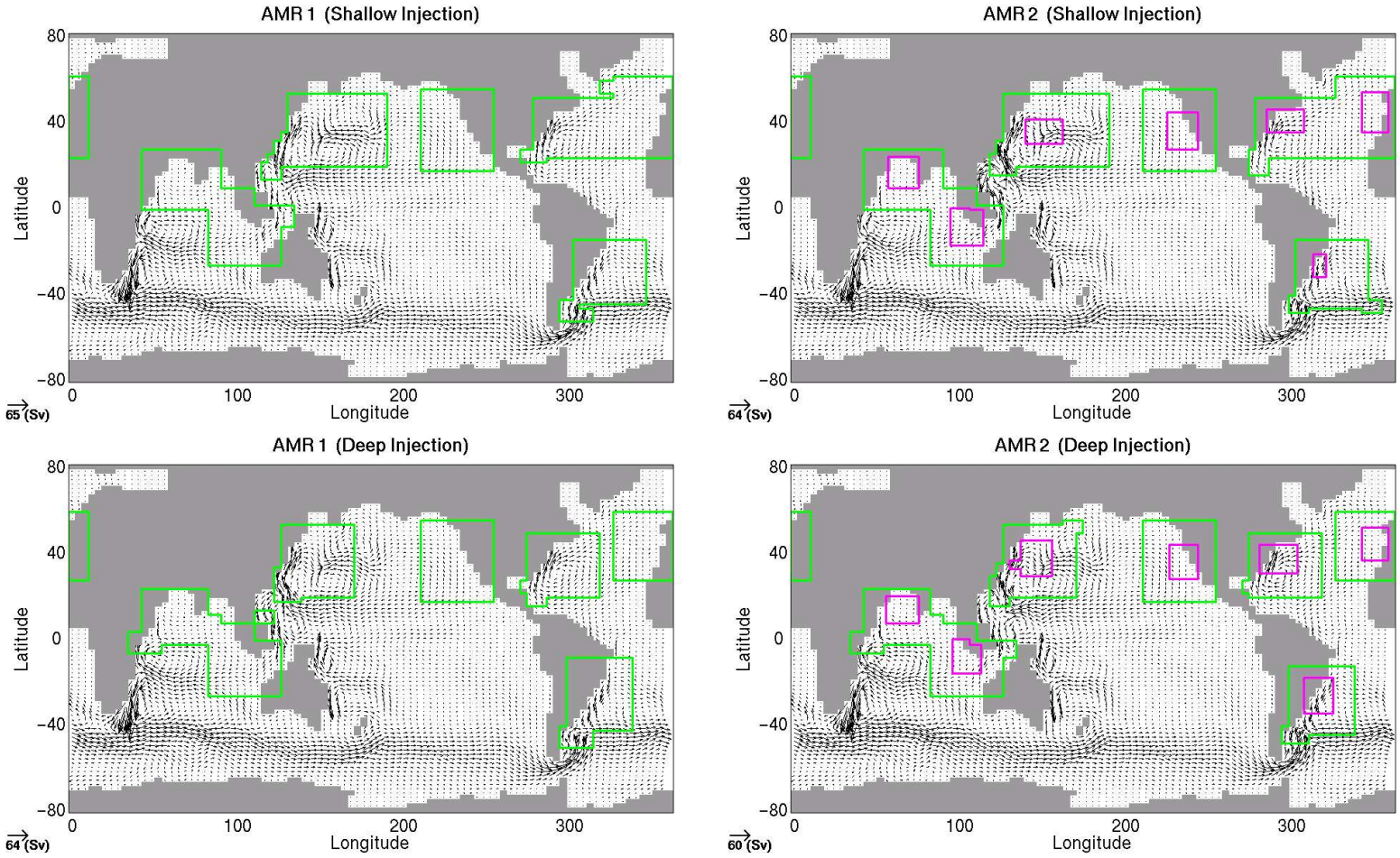


Figure 6.15: Barotropic Transports of the AMR resolutions at year 10.

are consistently warmer than Levitus and are under freshed. The annual and zonal mean temperature at year 10 are displayed in Figure 6.17, and show very little discrepancy among the different simulations. Salinity annual and zonal means at year 10 have slightly more deviation as seen in Figure 6.18. All resolutions show a build up of salt present near the equator at the surface. The anomaly near 60° north latitude is due to the zonal mean at latitudes where land is present in the North Pacific and high salinity is present in the North Atlantic. The temperature and salinity zonal means show similarities to their respective zonal means at spin up (Figure 6.6). The fields therefore remain relatively consistent for the two decades of simulated CO_2 sequestration.

The temperature and fresh water transports are calculated at year 10 as discussed in Section 5.2.4 and reduced to the coarse resolution to make comparisons across equal volumes. Figure 6.19 shows the meridionally summed transports of barotropic volume, heat, and fresh water for all tested resolutions. Local maxima coincide with the longitudinal coordinates of the intensified western boundary currents. Strong agreement is seen among all resolutions.

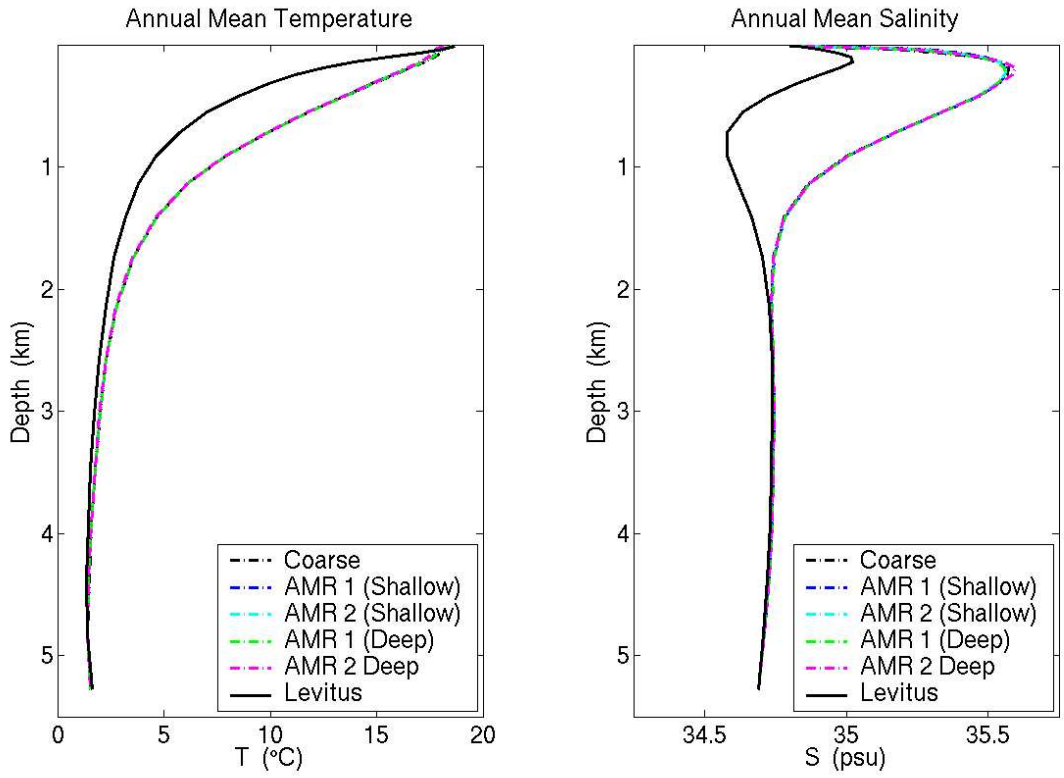


Figure 6.16: Depth profiles of the annual mean temperature (left) and salinity (right) during at year 10 of the Global Ocean simulations.

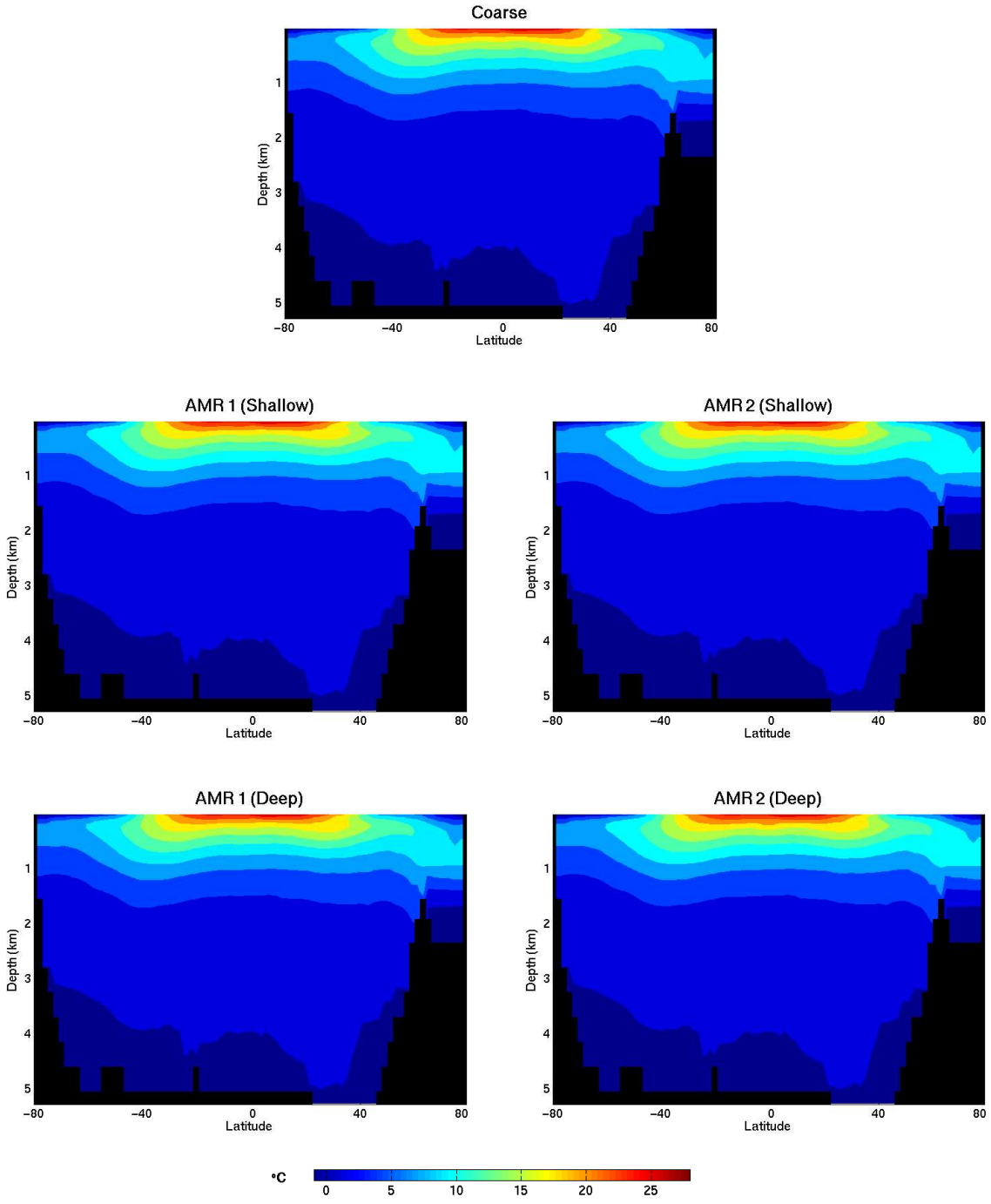


Figure 6.17: Annual and zonal mean temperature verses depth at year 10.

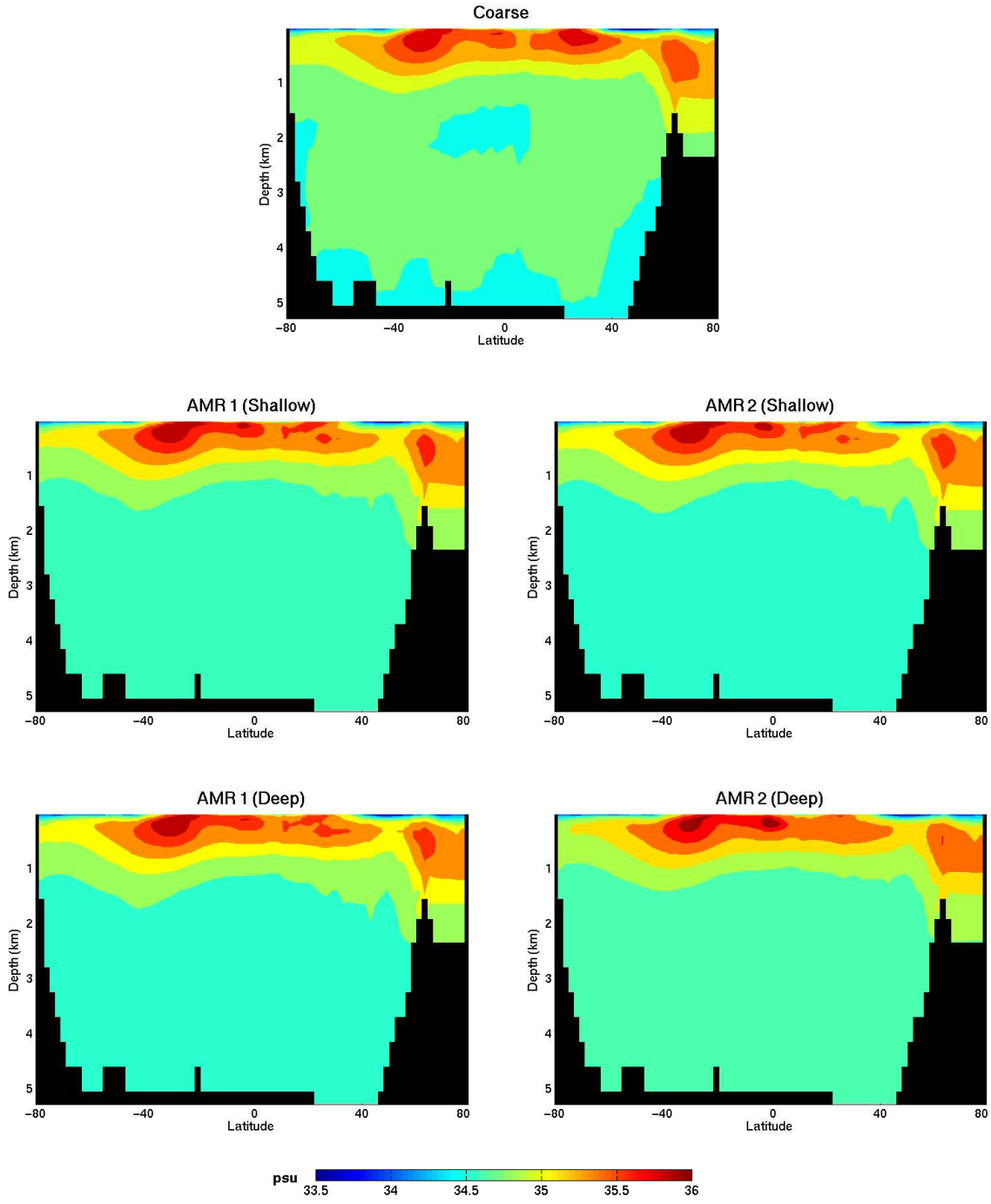


Figure 6.18: Annual and zonal mean salinity verses depth at year 10.

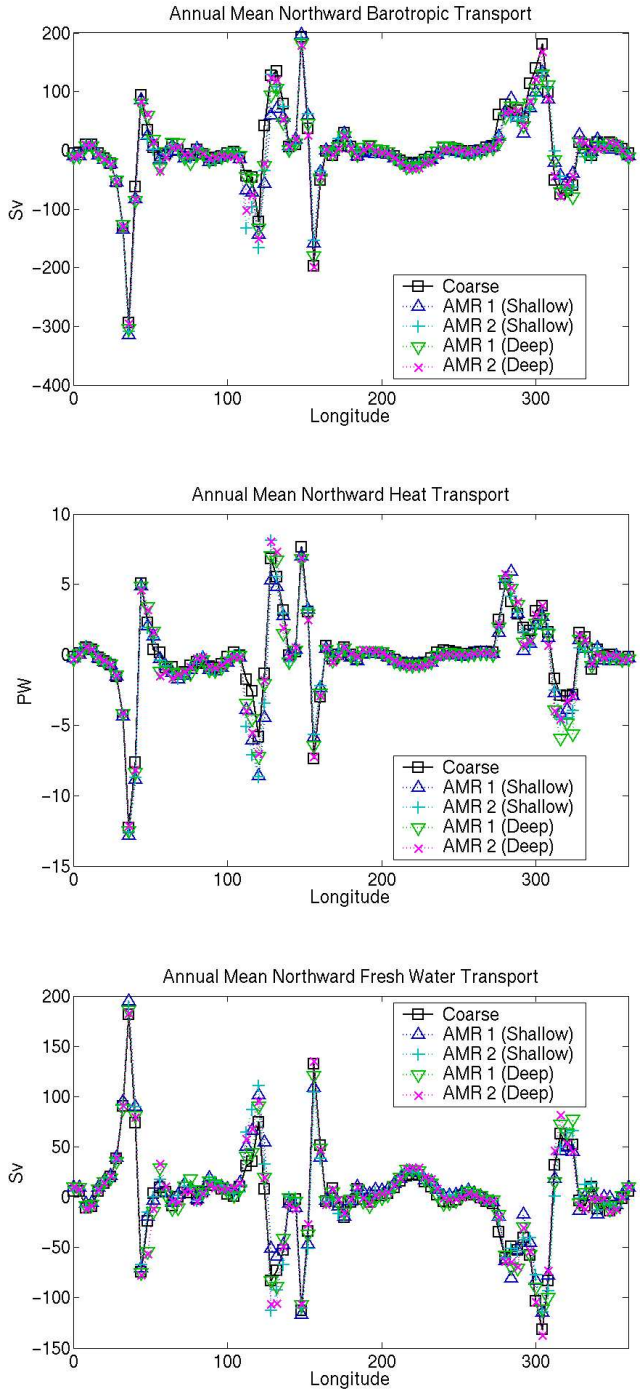


Figure 6.19: Meridional sums of the annual mean northward transports of barotropic volume (top), heat (middle), and freshwater transport (bottom) at year 10.

6.3 Parallel Performance

Simulation run times are highly dependent on how efficiently the model utilizes multiple CPUs. Several simulations were performed using various numbers of processors in order to determine the optimal run time rate for non-AMR and AMR simulations. These tests were conducted on the LLNL Multiprogrammatic Capability Cluster (MCR). This machine consists of a collection of CPU nodes, each with two 2.4-GHz Pentium 4 Xeon processors and 4 GB of RAM. The AMR Ocean Model communicates information between processors using the standard message-passing interface (MPI). The domain and all numerical and AMR parameters in this section are identical to those used in the Global Ocean simulations.

A $4^\circ \times 2^\circ$ resolution (coarse) is advanced 1000 time steps using 1, 2, 4, 8, 16, 32, and 64 processors. Similar tests are conducted for $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ (standard) and $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ (fine) resolutions. The three resolutions are equal to those used by level-(0), level-(1), and level-(2) in the Global Ocean AMR simulations. Table 6.2 shows the run time rates obtained for each resolution using various numbers of processors. The $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ case was unable to run with 1 or 2 processors due to memory constraints. The optimal run times are achieved for coarse, standard, and fine cases using 8, 16, and 32 processors, respectively.

Testing parallel performance is less straightforward with AMR given that the size and positions of refinement patches significantly impact run time. These factors are constantly changing as refinement regions follow areas of interest. AMR parallelism for the OCMIP application is tested on a series of simulations with various amounts of refinement. Each test case consists of refinement patches centered about the CO_2 injection sites. Res-

olution on level-(1) and level-(2) are equal to the standard and fine non-AMR resolutions, respectively. Refinement patches are kept static as the test cases are advanced 27 level-(0) time steps with various numbers of processors. This neglects the overhead due to regridding which makes the optimal simulation for each refinement orientation an upper bound. It is also unlikely that an AMR simulation will consistently generate AMR hierarchies with refinement patches identical to those used in the performance tests. Placing refinement around injection sites, however, makes the time integration representative of the OCMIP CO_2 direct injection simulations. Similar tests should be conducted for other applications known to produce different clusters of refinement patches.

Figure 6.20 shows the run time rates obtained when the domain is refined by a single refinement level. Each test uses 8 processors on level-(0), and level-(1) is tested with 1, 2, 4, 8, 16, and 32 processors. The amount of refinement covering the tracer domain ranges from 3.7% to 35.3%. Test cases with up to 16.9% of refinement yield the fastest rates when level-(1) uses 4 processors. Using one level of refinement with 4 processors gives a shorter run time than a standard resolution non-AMR simulation (see Table 6.2) when

Processors	$\mathbf{4^\circ \times 2^\circ}$	$\mathbf{\frac{4}{3}^\circ \times \frac{2}{3}^\circ}$	$\mathbf{\frac{4}{9}^\circ \times \frac{2}{9}^\circ}$
1	31.5	0.7	NA
2	30.3	0.8	NA
4	48.1	1.6	0.001
8	57.1	2.6	0.027
16	51.1	3.9	0.076
32	30.8	3.4	0.11
64	9.7	1.3	0.08

Table 6.2: Simulation rates in simulated years per CPU day for $4^\circ \times 2^\circ$, $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$, and $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ resolutions.

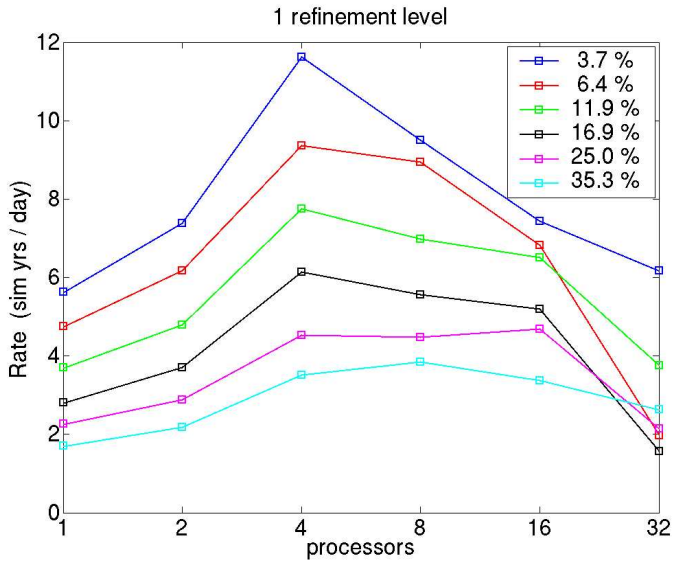


Figure 6.20: Simulation rate verses number of processors for AMR simulations with 1 level of refinement. The legend indicates the percentage of the tracer domain covered by level-(1).

the tracer domain is refined by less than 35.3%.

Similar tests are performed for an AMR hierarchy with two refinement levels. The graphs in Figure 6.21 and Figure 6.22 show the run time rates for different amounts of level-(1) and level-(2) refinement. Each test uses 8 processors on level-(0). The number of processors is limited to 4 on level-(1) when level-(1) covers less than 30% of the tracer domain. Refinement level-(2) is tested with various numbers of processors.

Each graph in Figure 6.21 and Figure 6.22 varies level-(1) refinement and uses the same level-(2) refinement for each of its tests. The graphs show that run time rates decrease as the amount of level-(2) refinement increases. The graphs also indicate little dependence of run time rate on the amount of level-(1) refinement when level-(2) covers more than 2.7% of the tracer domain. Optimal run time rates are obtained with 4 processors on tests with less than 3% of level-(2) refinement, and 16 processors on tests with more than 3% of

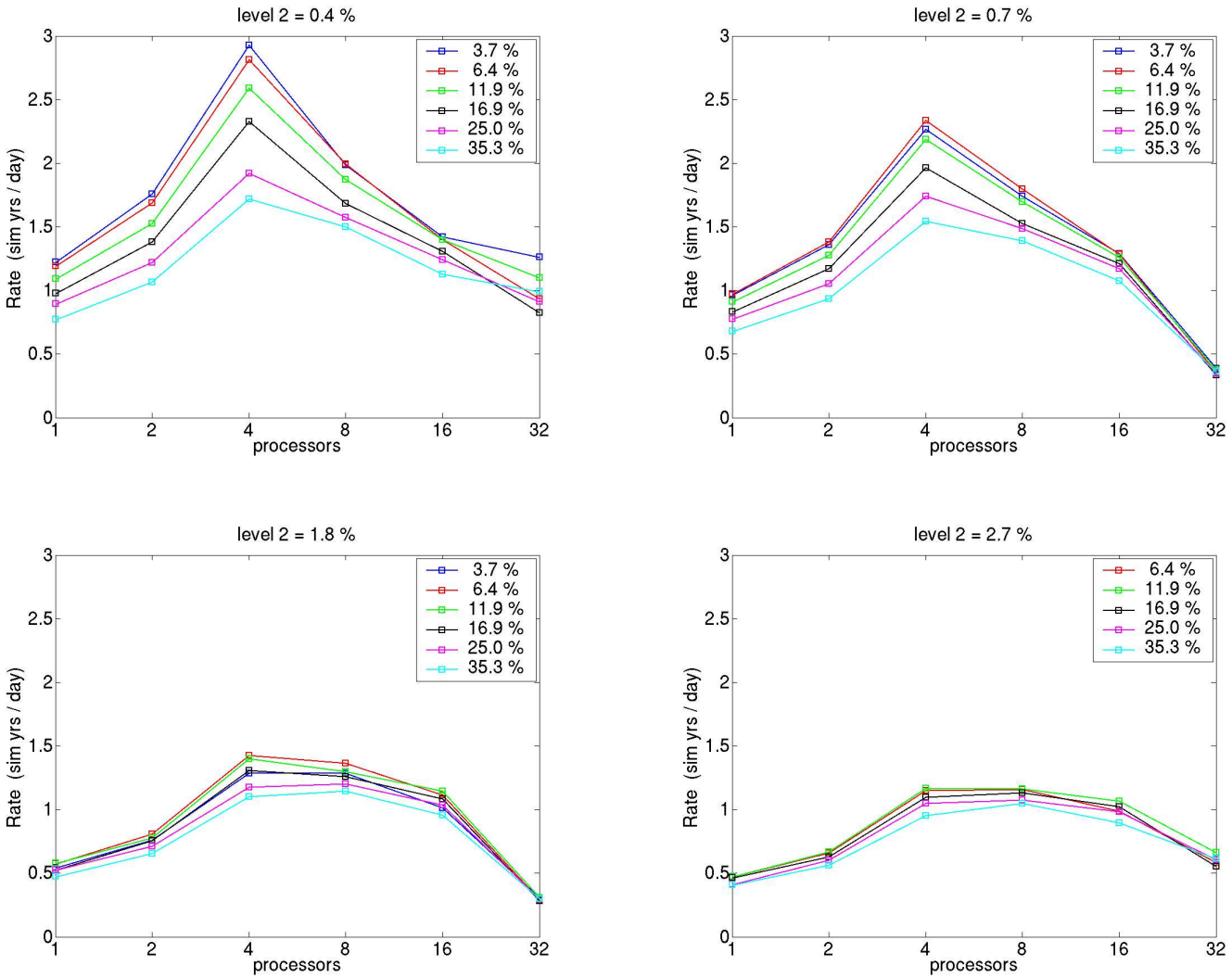


Figure 6.21: Simulation rate verses number of processors for AMR simulations with 2 levels of refinement. The legends indicate the percentage of the tracer domain covered by level-(1).

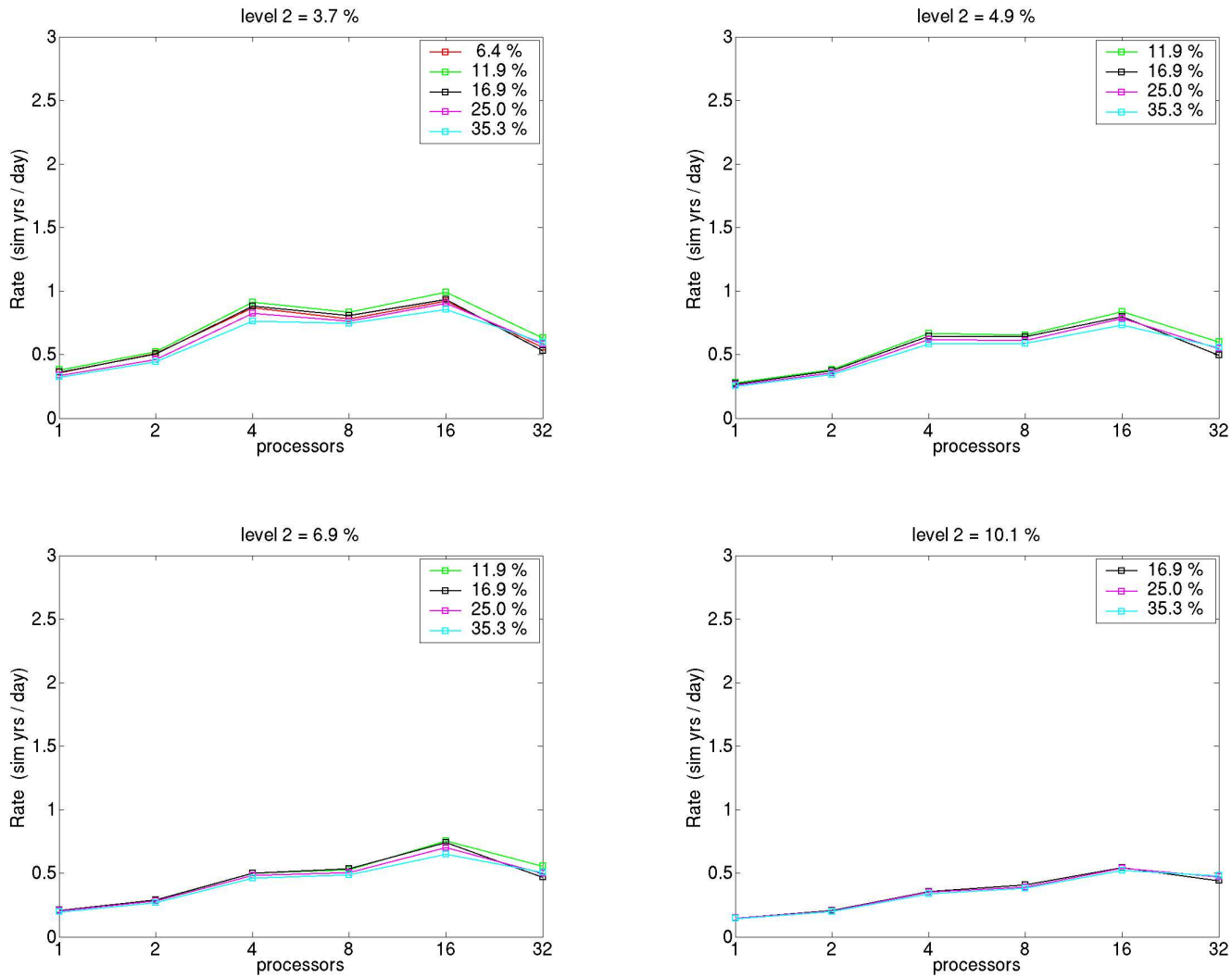


Figure 6.22: Simulation rate verses number of processors for AMR simulations with 2 levels of refinement. The legends indicate the percentage of the tracer domain covered by level-(1).

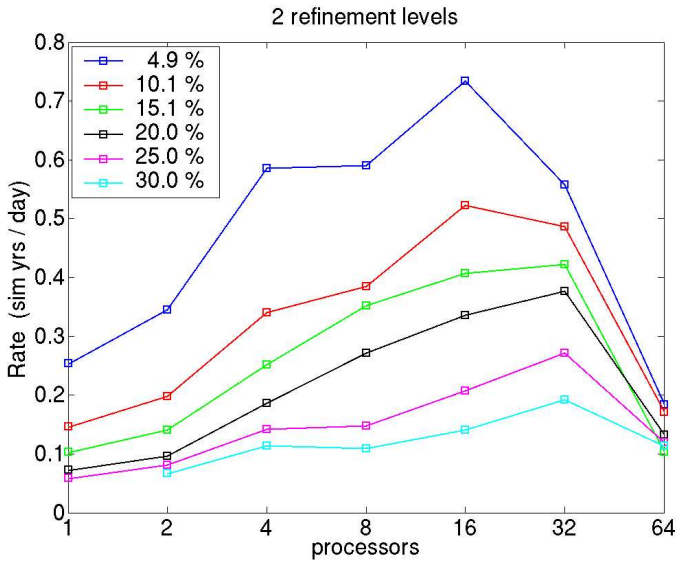


Figure 6.23: Simulation rate verses number of processors for AMR simulations with 2 level of refinement. The legend indicates the percentage of the tracer domain covered by level-(2).

level-(2) refinement.

Figure 6.23 shows rates obtained for various amounts of level-(2) refinement where level-(1) is fixed at 40% of the tracer domain. Tests with less than 15% of level-(2) refinement yield the fastest rates when using 16 processors. Tests with greater than or equal to 15% of level-(2) refinement perform best with 32 processors. An AMR simulation with 2 levels of refinement and 32 processors will execute at a rate of 0.19 simulated years per CPU day which is nearly double that of a fine resolution simulation (see Table 6.2).

The parallelism tests demonstrate the ability of the AMR Ocean Model to achieve results in a manner that is more computationally efficient than increasing resolution globally. A single refinement level AMR simulation can cover up to 25% of the tracer domain with level-(1) and produce results at a rate nearly equal to using a $\frac{4}{3}^{\circ} \times \frac{2}{3}^{\circ}$ global resolution. Similarly, a two refinement level AMR simulation can cover up to 30% of the tracer

domain with level-(2) and produce results at twice the rate as a $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ global resolution. AMR simulations also require fewer processors for optimal run time rates making them less demanding of computing resources compared to $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ and $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ resolutions.

6.4 Run Times

The simulation rates and increases in run time relative to the coarse resolution are displayed in Figure 6.24 and Figure 6.25. The figures show the same data on different vertical scales. The theoretical increases in CPU run time are calculated as described in Section 5.4. A simulation with a uniform $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ resolution takes nearly 15 times longer than a coarse resolution. The $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ resolution increases CPU run time by a factor of 35 compared to the $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ resolution. These dramatic increases are a reflection of the less than satisfactory usage of parallel processing in the AMR Ocean Model and motivates improvements in parallelism. The AMR 1 simulations require slightly less CPU time and slightly more overhead than a simulation with uniform $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$ resolution. The AMR 2 simulations significantly reduce CPU run time and overhead when compared to a simulation with uniform $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ resolution. The AMR simulations are also initialized from a coarse spin up state which also reduces their CPU run times significantly during the spin up phase.

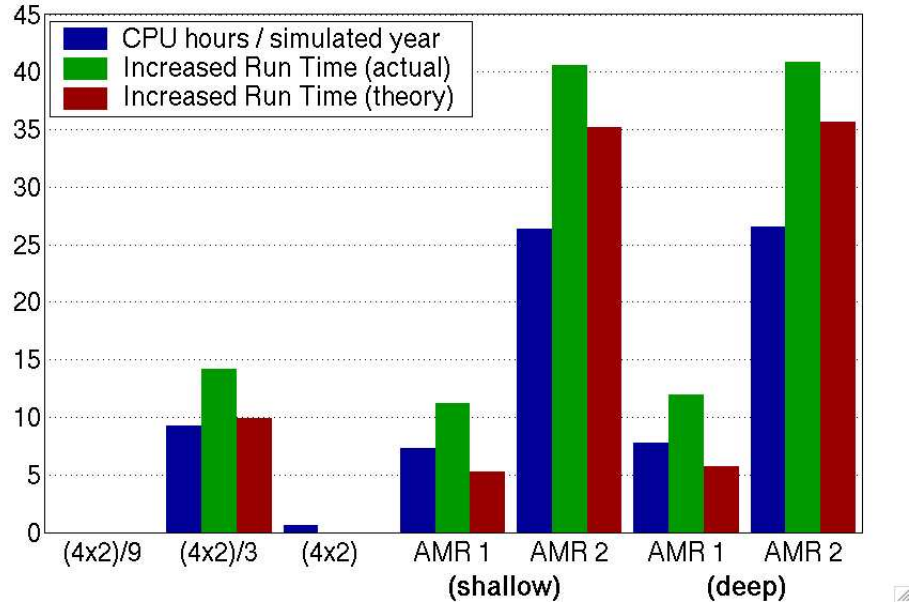


Figure 6.24: Simulation run times expressed in CPU hours per simulated year, and the actual and theoretical increases in run time relative to the coarse simulation run time. The vertical scale shown is smaller than the maximum values of $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ case to facilitate visual comparisons.

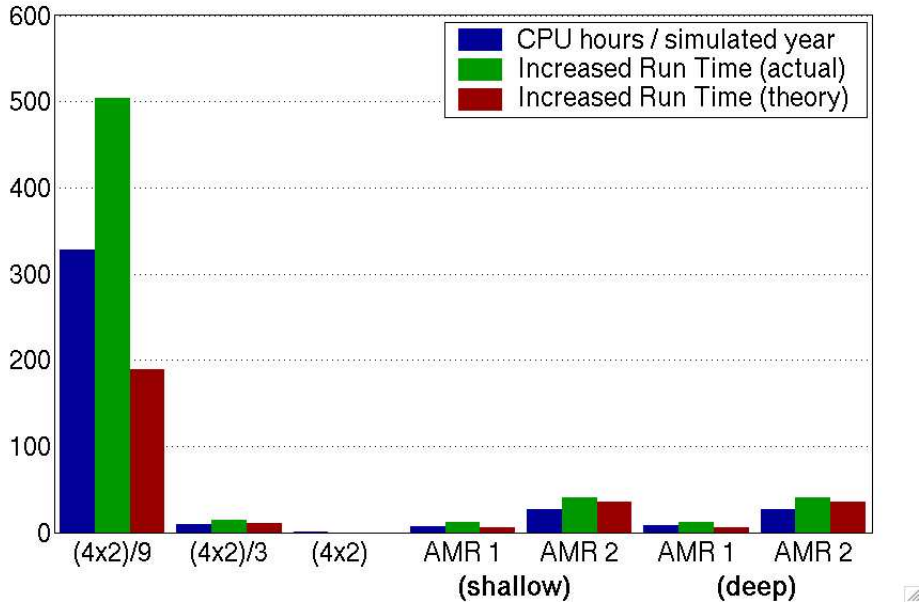


Figure 6.25: The same as Figure 6.24 with the vertical scale determined by the maximum of the $\frac{4}{9}^\circ \times \frac{2}{9}^\circ$ data.

6.5 Summary of Global Ocean Simulations

CO_2 sequestration simulations were conducted with multiple injection sites in the Global Ocean using coarse and AMR resolutions. Continuous injection is simulated for one decade at injection sites determined by the OCMIP protocol and injection depths of 710 meters and 2575 meters. The simulations do not resolve mesoscale eddies, and the same physical parameters are used for each. AMR refinement criteria are based on the injected CO_2 mass and barotropic velocity gradients near injected CO_2 . The passive tracer representing the CO_2 perturbation is more localized in the deep injection simulations which allows simplified refinement.

At large volume scales, the AMR simulations agree well with the coarse simulations on predictions of maximum changes in pH and increased CO_2 concentration. AMR simulations provide a means of predicting details in pH change and increased CO_2 concentration at volume scales smaller than are capable by the coarse resolution. All resolutions predict 99.99% of the sequestered CO_2 will remain in the ocean after one decade. Discrepancies in the spatial distributions of surface fluxes suggest the coarse and AMR resolutions could agree less on CO_2 retention after long time scales ($\sim$100 years). The AMR Ocean Model is not capable of simulations on such time scales until improvements are made that will shorten run times.

Slight AMR influences are observed in the ocean state after 10 simulated years. The magnitude of barotropic circulation varies slightly among AMR simulations at year 10, but circulation patterns agree well. Virtually no differences are seen in the temperatures produced by the various simulations. All predict temperatures at shallow depths which are

warmer than the Levitus annual means. Salinity fields all show increases from the Levitus annual means at the surface and vary slightly among the various simulations. Northward transports of barotropic volume, heat, and fresh water agree well among all resolutions and show local extrema at longitudes that coincide with intensified western boundaries.

Results of parallelism tests and the simulation run times suggest that performance improvements to the AMR Ocean Model should be made before conducting global CO_2 direct injection simulations much longer than a decade. It is still possible to conclude from these results that a significant reduction in run time is achieved when using AMR rather than a globally uniform standard or fine resolution. The AMR simulations in this study require fewer processors and are initialized with a coarse resolution spin up state. As a result, the AMR simulations are less demanding of computational resources than uniform resolutions equal to or finer than $\frac{4}{3}^\circ \times \frac{2}{3}^\circ$.

Chapter 7

Conclusions and Future Work

This dissertation has demonstrated the effectiveness of adaptive mesh refinement (AMR) applied in the field of numerical ocean modeling. The method addresses the need for increasing resolution in localized areas in a manner that is computationally more efficient than increasing resolution uniformly on the entire domain. New computational techniques were presented that allow numerical methods common in ocean models to be applied in AMR simulations. The LLNL software library SAMRAI (Structured Adaptive Mesh Refinement Application Infrastructure) was altered to include the AMR algorithms developed for ocean modeling. The ocean model was built on top of the amended SAMRAI library. Components of various existing ocean models were assembled to create a fully functional AMR Ocean Model. To the best of my knowledge, this is the first Bryan-Cox-Semtner ocean model that successfully implements traditional structured adaptive mesh refinement.

Strategies for AMR using leap frog time integration and B-grid staggering were validated with standard benchmark tests. It was discovered that leap frog AMR can produce

excessive numerical dispersion when sharp solution gradients are advected through an AMR interface. Smoothing the solution at the interface reduced the dispersion effects. The scheme for B-grid AMR was successfully applied to a geophysical simulation of a barotropic modon.

Limited basin tests showed that AMR greatly effects the solution accuracy when attempting to refine the flow field during spin up. Criteria based on barotropic velocity gradients and barotropic vorticity produced better results than criteria based on surface height gradients. All criteria showed trends of convergence as the amount of refinement was increased. It was concluded from these tests that AMR with the tested refinement criteria is not suitable during spin up simulations given that large amounts of the domain require refinement to produce a globally accurate flow field.

The AMR Ocean Model was applied in CO_2 sequestering simulations. A set of AMR refinement criteria was developed for such applications to refine injection sites and strong currents local to sequestered CO_2 . AMR applied to this application enables multiple point sources to be dynamically refined. Such capabilities are not possible with predeceasing nested grid models which use static localized increases in resolution. CO_2 sequestration simulations of the North Atlantic with a single point source confirm an increase in numerical accuracy when using AMR with one refinement level. AMR simulations confirm coarse resolution predictions of the amount of CO_2 retained by the ocean after two decades of continuous injection. Global Ocean simulations with multiple point sources reveal details in sequestered CO_2 concentration at small spatial scales. Such results are virtually unattainable with uniform high resolutions due to extremely long run times. Minimal changes in the ocean circulation, temperature, or salinity fields are observed when using AMR after

spin up.

Given that the AMR Ocean Model is the first of its kind, future work is desired before a full production model is possible. First, additional modeling techniques might improve the model's representation of the physical ocean. These include alternate vertical mixing schemes, inclusion of isopycnal mixing schemes, alternate barotropic time stepping techniques, and a model of sea ice. Next, improvements in parallel performance and run time efficiency are highly desired. This requires some sections of code to be reorganized for optimization. Alternate load balancing schemes could also be beneficial. Finally, an ocean model with completely adaptive resolution should refine topography. Implementing partial cells in a way that does not severely restrict the time step is a possible solution. This would require most of the AMR Ocean Model to be rewritten using a finite volume approach.

Given the current available computing resources, similar studies of efficient improvements to climate and ocean model accuracy will be seen in the near future. Although physical improvements are currently required, the model presented in this study could eventually be expanded to include an atmospheric component. Such a model would be beneficial for studying localized weather systems such as tropical storms and hurricanes and would improve advanced warning capabilities for severe weather conditions.

Appendix A

Details of Subcyling Time Refinement

The algorithms in this appendix contain the details of the those discussed in Section 3.2.3. Two procedures are given for adjusting the barotropic and non-barotropic time step ratios,

$$r_\ell^{\mathrm{BT}} = \Delta t_\ell^{\mathrm{BT}} / \Delta t_{\ell+1}^{\mathrm{BT}} \tag{A.1}$$

$$r_\ell^{\mathrm{BC}} = \Delta t_\ell^{\mathrm{BC}} / \Delta t_{\ell+1}^{\mathrm{BC}} \ , \tag{A.2}$$

such that the subcycle rule,

$$\frac{r_\ell^{\mathrm{BT}}}{r_\ell^{\mathrm{BC}}} = \frac{\mathrm{subcycles}_{\ell+1}}{\mathrm{subcycles}_\ell} \ , \tag{A.3}$$

is satisfied on each refinement level. The adjustment procedures are implemented prior to the first advance of the simulation. Time stepping is then performed with a traditional time refinement algorithm for the barotropic system coupled to a leapfrog time refinement

algorithm for the baroclinic and tracer systems. This coupled algorithm is given in detail at the end of this appendix.

A.1 Coarsest to Finest Time Step Adjustment

The *CoarsestToFinest* time step adjustment procedure shown in Algorithm A.1.1, starts by adjusting Δt_0^{BT} or Δt_0^{BC} such that $\mathrm{subcycles}_0$ is an integer. The remaining r_ℓ^{BT} are adjusted if necessary in order of increasing ℓ such that $r_\ell^{\mathrm{BT}}\mathrm{subcycles}_\ell$ is an integer multiple of r_ℓ^{BC}. The subcycles and time steps on level-$(\ell+1)$ are reset using (A.3) and (A.1)-(A.2) respectively.

Algorithm A.1.1. Δt Adjustment - *CoarsestToFinest*

$\mathrm{subcycles}_0 =$ closest integer multiple of $\left(\Delta t_0^{\mathrm{BC}}/\Delta t_0^{\mathrm{BT}}\right)$

if ($\Delta t_0^{\mathrm{BC}}/\mathrm{subcycles}_0 \; > \; \Delta t_0^{\mathrm{BT}}$)
  $\Delta t_0^{\mathrm{BC}} = \mathrm{subcycles}_0 \; \Delta t_0^{\mathrm{BT}}$
else
  $\Delta t_0^{\mathrm{BT}} = \Delta t_0^{\mathrm{BC}} \; / \; \mathrm{subcycles}_0$
end

for ($\ell = 1 \rightarrow$ finest)
  while [$(r_\ell^{\mathrm{BT}}\mathrm{subcycles}_\ell \; \% \; r_\ell^{\mathrm{BC}}) \; > \; 0$]
    $r_\ell^{\mathrm{BT}} \; = \; r_\ell^{\mathrm{BT}} \; + \; 1$
  end

  $\mathrm{subcycles}_{\ell+1} \; = \; r_\ell^{\mathrm{BT}}\mathrm{subcycles}_\ell \; / \; r_\ell^{\mathrm{BC}}$

  $\Delta t_{\ell+1}^{\mathrm{BC}} \; = \; \Delta t_\ell^{\mathrm{BC}} \; / \; r_\ell^{\mathrm{BC}}$

  $\Delta t_{\ell+1}^{\mathrm{BT}} \; = \; \Delta t_\ell^{\mathrm{BT}} \; / \; r_\ell^{\mathrm{BT}}$
end

A.2 Finest to Coarsest Time Step Adjustment

The *FinestToCoarsest* time step adjustment procedure is shown in Algorithm A.2.1. It starts by adjusting $\Delta t^{\mathrm{BT}}_{\mathrm{finest}}$ or $\Delta t^{\mathrm{BC}}_{\mathrm{finest}}$ such that $\mathrm{subcycles}_{\mathrm{finest}}$ is an integer. The remaining r^{BC}_{ℓ} are adjusted if necessary in order of decreasing ℓ such that $r^{\mathrm{BC}}_{\ell}\mathrm{subcycles}_{\ell+1}$ is an integer multiple of r^{BT}_{ℓ}. The subcycles and time steps on level-(ℓ) are set using (A.3) and (A.1)-(A.2) respectively.

Algorithm A.2.1. Δt Adjustment - *FinestToCoarsest*

$\mathrm{subcycles}_{\mathrm{finest}}$ = closest integer multiple of $\left(\Delta t^{\mathrm{BC}}_{\mathrm{finest}}/\Delta t^{\mathrm{BT}}_{\mathrm{finest}}\right)$

if $\left(\ \Delta t^{\mathrm{BC}}_{\mathrm{finest}}/\mathrm{subcycles}_{\mathrm{finest}} \ > \ \Delta t^{\mathrm{BT}}_{\mathrm{finest}}\ \right)$
  $\Delta t^{\mathrm{BC}}_{\mathrm{finest}} = \mathrm{subcycles}_{\mathrm{finest}}\ \Delta t^{\mathrm{BT}}_{\mathrm{finest}}$
else
  $\Delta t^{\mathrm{BT}}_{\mathrm{finest}} = \Delta t^{\mathrm{BC}}_{\mathrm{finest}}\ /\ \mathrm{subcycles}_{\mathrm{finest}}$
end

for $(\ \ell = \mathrm{finest} - 1\ \rightarrow 0\ )$
  while $[\ (r^{\mathrm{BC}}_{\ell}\mathrm{subcycles}_{\ell+1}\ \%\ r^{\mathrm{BT}}_{\ell})\ >\ 0\ ]$
    $r^{\mathrm{BC}}_{\ell}\ =\ r^{\mathrm{BC}}_{\ell}\ +\ 1$
  end

  $\mathrm{subcycles}_{\ell}\ =\ r^{\mathrm{BC}}_{\ell}\mathrm{subcycles}_{\ell+1}\ /\ r^{\mathrm{BT}}_{\ell}$

  $\Delta t^{\mathrm{BC}}_{\ell}\ =\ \Delta t^{\mathrm{BC}}_{\ell+1}\ r^{\mathrm{BC}}_{\ell}$

  $\Delta t^{\mathrm{BT}}_{\ell}\ =\ \Delta t^{\mathrm{BT}}_{\ell+1}\ r^{\mathrm{BT}}_{\ell}$
end

A.3 Detailed Subcycle Advancement With Time Refinement

The algorithm shown below advances level-(ℓ) by a single barotropic step, step_ℓ. It must be called subcycles_0 times from level-(0) in order to perform a complete non-barotropic step on level-(0). Traditional time refinement is used to advance the barotropic system with flux sum variables denoted by $\tilde{F}_\ell^{BT}$. The non-barotropic systems advance with leapfrog time refinement and use primary and secondary flux sums denoted by $\tilde{F}_\ell$ and $\tilde{f}_\ell$ respectively.

Algorithm A.3.1. Subcycle Advance on Level-(ℓ)

```
Adv_Non_BT = ( step_ℓ % subcycles_ℓ == 0 )

if  ( Adv_Non_BT )
  Fill Non-Barotropic ghosts
  if  ( mixing_step_l )
    Non-Barotropic Step using Euler forward-backward
    mixing_step_{ℓ+1} = TRUE
    σ = 1
  else
    Non-Barotropic Step using leapfrog
    σ = 2
  end
end

Fill Barotropic ghosts
Barotropic Step

if  ( ℓ > 0 )
  F̃_ℓ^BT += [BT patch edge fluxes] / r_{ℓ-1}^BT
  if  ( Adv_Non_BT )
    if  [ (step_ℓ % 2)  ==  (r_{ℓ-1}^BC % 2) ]
      F̃_ℓ += σ* [non-BT patch edge fluxes] / r_{ℓ-1}^BC
      W_ℓ += σ
    end
    if  ( step_ℓ is even )
      f̃_ℓ += σ* [non-BT patch edge fluxes] / r_{ℓ-1}^BC
      w_ℓ += σ
    end
  end
end
```

if ($\ell + 1$ exists)
  $\tilde{F}^{BT}_{\ell+1} = 0.0$
  if (Adv_Non_BT)
    if (mixing_step$_l$)
      $\tilde{F}_{\ell+1} = 0.0,\ \ W_{\ell+1} = 0.0$
    else
      $\tilde{F}_{\ell+1} = \tilde{f}_{\ell+1},\ \ W_{\ell+1} = w_{\ell+1}$
    end
    $\tilde{f}_{\ell+1} = 0.0,\ \ w_{\ell+1} = 0.0$
  end

  for ($sub\_step = 1 \rightarrow r^{\rm BT}_{\ell}$)
    Subcycle Advance on Level-($\ell + 1$)
    if (Adv_Non_BT)
      mixing_step$_{\ell+1}$ = FALSE
    end
  end

  Re-advance Barotropic data touching $\ell + 1$ interface using spatially averaged $\tilde{F}^{BT}_{\ell+1}$
  Coarsen $\ell + 1$ Barotropic data onto ℓ

  if [(step$_{\ell+1}$/subcycles$_{\ell+1}$) % $r^{\rm BC}_{\ell}$ == 0]
    Re-advance non-Barotropic data touching $\ell + 1$ interface using spatially averaged $\left(\tilde{F}_{\ell+1}\ /\ W_{\ell+1}\right)$
    Coarsen $\ell + 1$ non-Barotropic data onto ℓ
  end
end

step$_{\ell}$ = step$_{\ell}$ + 1

Appendix B

Coordinate System Transformations

The numerical model described in Chapter 2 discretizes the sphere on a longitude-latitude (long-lat) grid. It is possible to orient the grid such that its longitude and latitude coordinates do not coincide with those of the Earth (i.e., the grid and Earth axes do not coincide). The following describes transformations for scalars and vectors between the "grid" coordinate system and the "Earth" coordinate system. These operations are not implemented in the research presented in this thesis. They are included here as supplemental material for future reference.

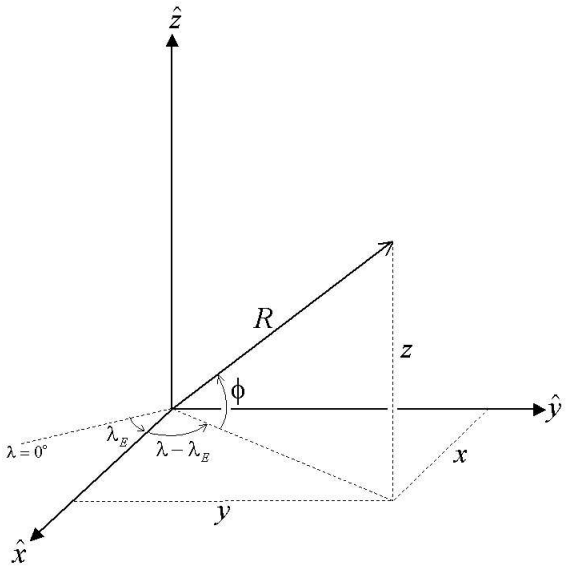


Figure B.1: Relation between Cartesian and long-lat coordinates.

B.1 Cartesian and Long-Lat Coordinates

Figure B.1 shows the two coordinate systems. The radial coordinate is constant ($r = R$) and the $\hat{x}$ axis intersects the long-lat equator at $(\lambda, \phi) = (\lambda_E, 0)$. The transformation from Cartesian coordinates to long-lat coordinates is given by

$$\lambda = \lambda_E + \arctan \frac{y}{x} \tag{B.1}$$

$$\phi = \arcsin \frac{z}{R} \, . \tag{B.2}$$

The reverse transformation from long-lat to Cartesian coordinates is

$$x = R \, \cos\phi \, \cos(\lambda - \lambda_E) \tag{B.3}$$

$$y = R \, \cos\phi \, \sin(\lambda - \lambda_E) \tag{B.4}$$

$$z = R \, \sin\phi \, . \tag{B.5}$$

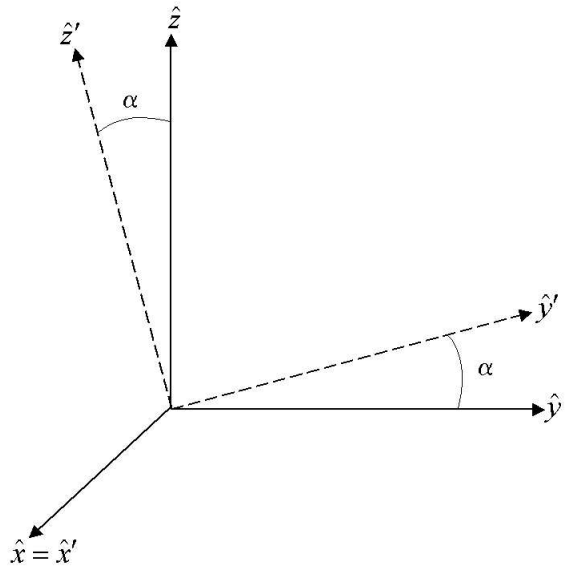


Figure B.2: Grid and Earth coordinate systems in Cartesian coordinates.

B.2 Grid to Earth Coordinates

Figure B.2 shows the Earth coordinate system $(\hat{x}, \hat{y}, \hat{z})$ and the grid coordinate system $(\hat{x}', \hat{y}', \hat{z}')$. The $\hat{x}$ and $\hat{x}'$ axes coincide, and the $\hat{y}'$ and $\hat{z}'$ axes are offset from the $\hat{y}$ and $\hat{z}$ axes by an angle α respectively. The Cartesian coordinate (x', y', z') in the grid system is transformed to the Earth system by

$$x = x' \tag{B.6}$$

$$y = y' \cos\alpha - z' \sin\alpha \tag{B.7}$$

$$z = y' \sin\alpha + z' \cos\alpha \; . \tag{B.8}$$

The grid system coordinates are expressed in long-lat coordinates using Equations B.3-B.5.

$$x = R\cos\phi'\cos(\lambda' - \lambda_E) \tag{B.9}$$

$$y = R\cos\phi'\sin(\lambda' - \lambda_E)\cos\alpha - R\sin\phi'\sin\alpha \tag{B.10}$$

$$z = R\cos\phi'\sin(\lambda' - \lambda_E)\sin\alpha + R\sin\phi'\cos\alpha \tag{B.11}$$

Equations B.1 and B.2 are used to translate the Earth coordinates to long-lat coordinates resulting in the transformation from grid long-lat coordinates to Earth long-lat coordinates.

$$\lambda = \lambda_E + \arctan\left(\frac{\cos\phi'\sin(\lambda' - \lambda_E)\cos\alpha - \sin\phi'\sin\alpha}{\cos\phi'\cos(\lambda' - \lambda_E)}\right) \tag{B.12}$$

$$\phi = \arcsin\left(\sin\phi'\cos\alpha + \cos\phi'\sin(\lambda' - \lambda_E)\sin\alpha\right) \tag{B.13}$$

B.3 Earth to Grid Coordinates

Figure B.2 is used to translate coordinates in the Earth system to the grid system. The Cartesian coordinate transformation is

$$x' = x \tag{B.14}$$

$$y' = y\cos\alpha + z\sin\alpha \tag{B.15}$$

$$z' = -y\sin\alpha + z\cos\alpha\ . \tag{B.16}$$

The Earth system coordinates are written in long-lat coordinates giving

$$x' = R\cos\phi\cos(\lambda - \lambda_E) \tag{B.17}$$

$$y' = R\cos\phi\sin(\lambda - \lambda_E)\cos\alpha + R\sin\phi\sin\alpha \tag{B.18}$$

$$z' = -R\cos\phi\sin(\lambda - \lambda_E)\sin\alpha + R\sin\phi\cos\alpha\ . \tag{B.19}$$

The long-lat coordinate transformation from the Earth system to grid system is obtained by writing the grid system in long-lat coordinates.

$$\lambda' = \lambda_E + \arctan\left(\frac{\cos\phi\sin(\lambda-\lambda_E)\cos\alpha + \sin\phi\sin\alpha}{\cos\phi\cos(\lambda-\lambda_E)}\right) \tag{B.20}$$

$$\phi' = \arcsin\left(\sin\phi\cos\alpha - \cos\phi\sin(\lambda-\lambda_E)\sin\alpha\right) \tag{B.21}$$

B.4 Cartesian and Long-Lat Vectors

A vector, $\vec{V}$, is defined in Cartesian and long-lat coordinates as

$$\vec{V} = a\hat{x} + b\hat{y} + c\hat{z} \tag{B.22}$$

and

$$\vec{V} = s\hat{\lambda} + t\hat{\phi} + R\hat{r} \tag{B.23}$$

respectively. The vector transformation from Cartesian coordinates to long-lat coordinates is performed by expressing the unit vectors $(\hat{x},\hat{y},\hat{z})$ in terms of $(\hat{\lambda},\hat{\phi},\hat{r})$. These relations are deduced from Figure B.3.

$$\begin{bmatrix}\hat{x}\\ \hat{y}\\ \hat{z}\end{bmatrix} = \begin{bmatrix}-\sin(\lambda-\lambda_E) & -\cos(\lambda-\lambda_E)\sin\phi & \cos(\lambda-\lambda_E)\cos\phi\\ \cos(\lambda-\lambda_E) & -\sin(\lambda-\lambda_E)\sin\phi & \sin(\lambda-\lambda_E)\cos\phi\\ 0 & \cos\phi & \sin\phi\end{bmatrix}\begin{bmatrix}\hat{\lambda}\\ \hat{\phi}\\ \hat{r}\end{bmatrix} \tag{B.24}$$

The expressions for $\hat{x}$, $\hat{y}$, and $\hat{z}$ are substituted into equation (B.22) and simplifying for s, t, and R. The result is the vector transformation from Cartesian to long-lat coordinates.

$$\begin{bmatrix} s \\ t \\ R \end{bmatrix} = \begin{bmatrix} -\sin(\lambda - \lambda_E) & \cos(\lambda - \lambda_E) & 0 \\ -\cos(\lambda - \lambda_E)\sin\phi & -\sin(\lambda - \lambda_E)\sin\phi & \cos\phi \\ \cos(\lambda - \lambda_E)\cos\phi & \sin(\lambda - \lambda_E)\cos\phi & \sin\phi \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{B.25}$$

The reverse transformation from long-lat to Cartesian coordinates is the inverse of the above matrix equation.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -\sin(\lambda - \lambda_E) & -\cos(\lambda - \lambda_E)\sin\phi & \cos(\lambda - \lambda_E)\cos\phi \\ \cos(\lambda - \lambda_E) & -\sin(\lambda - \lambda_E)\sin\phi & \sin(\lambda - \lambda_E)\cos\phi \\ 0 & \cos\phi & \sin\phi \end{bmatrix} \begin{bmatrix} s \\ t \\ R \end{bmatrix} \tag{B.26}$$

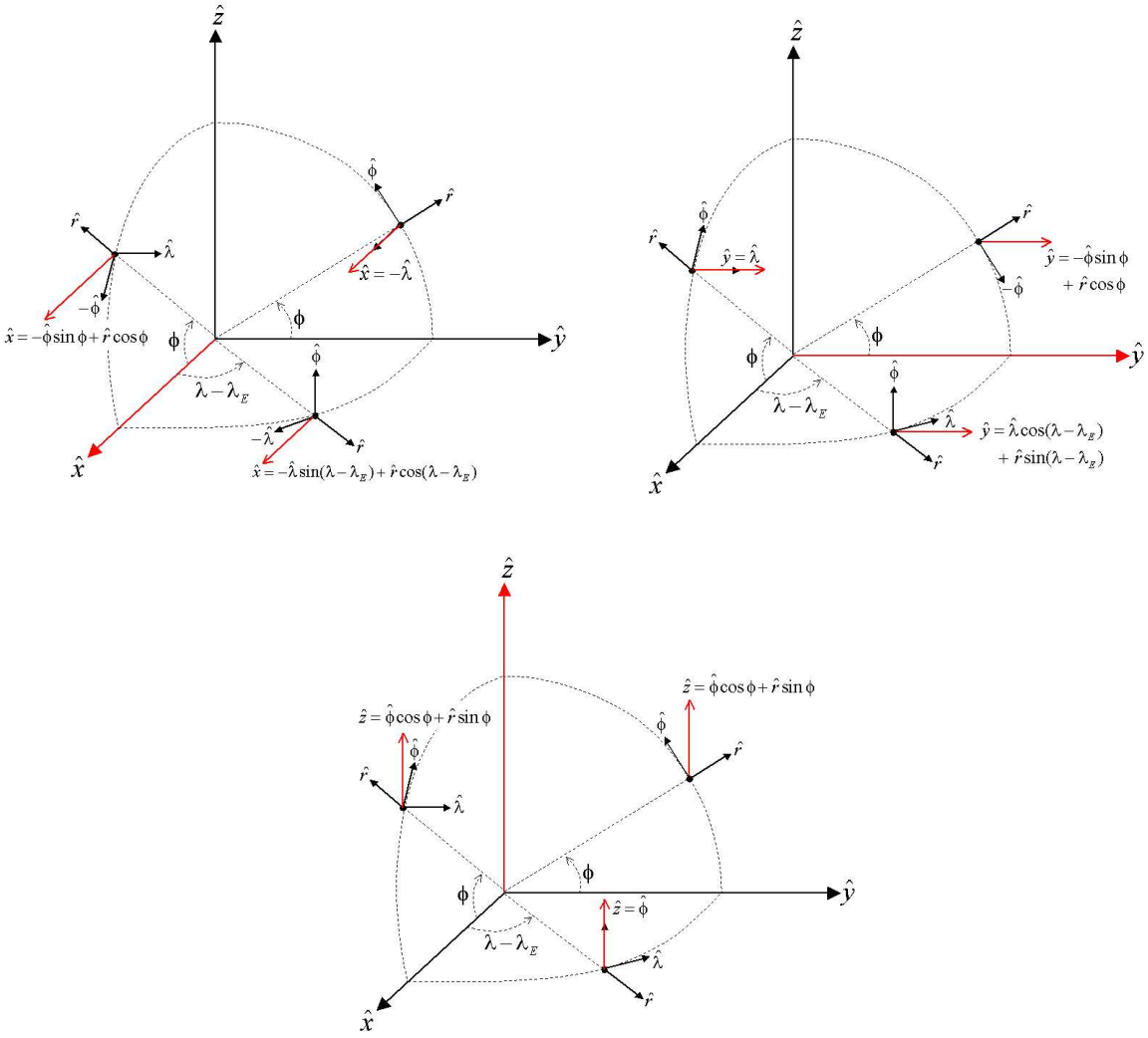


Figure B.3: Cartesian unit vector transformations to long-lat unit vectors.

B.5 Grid and Earth Cartesian Vectors

A Cartesian vector, $\vec{V}$, is defined in the Earth and grid systems as

$$\vec{V} \;=\; a\hat{x} \;+\; b\hat{y} \;+\; c\hat{z} \tag{B.27}$$

and

$$\vec{V} \;=\; a'\hat{x}' \;+\; b'\hat{y}' \;+\; c'\hat{z}' \tag{B.28}$$

respectively. The rotation from the Earth system to the grid system is performed by writing the Earth unit vectors $(\hat{x}, \hat{y}, \hat{z})$ in terms of the grid unit vectors $(\hat{x}', \hat{y}', \hat{z}')$. These relationships are deduced from Figure B.2.

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} \;=\; \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \hat{x}' \\ \hat{y}' \\ \hat{z}' \end{bmatrix} \tag{B.29}$$

The Earth vector becomes a grid vector by substituting the above expressions for $\hat{x}$, $\hat{y}$, and $\hat{z}$, into equation (B.27) and simplifying for a', b', and c'.

$$\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} \;=\; \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{B.30}$$

The transformation from a grid vector to an Earth vector is the inverse of the above matrix equation.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \;=\; \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} \tag{B.31}$$

B.6 Grid to Earth Vector

Vector transformations from the grid system to the Earth system are done in a three step process.

$$\vec{V}(s',t',R) \longrightarrow \vec{V}(a',b',c') \longrightarrow \vec{V}(a,b,c) \longrightarrow \vec{V}(s,t,R) \tag{B.32}$$

The long-lat grid vector, $\vec{V}(s',t',R)$, is first expressed in Cartesian coordinates using equation (B.26). The Cartesian grid vector, $\vec{V}(a',b',c')$, is rotated to the Earth system with equation (B.31). The transformation is completed by expressing the Cartesian Earth vector, $\vec{V}(a,b,c)$, in long-lat coordinates using equation (B.25) giving $\vec{V}(s,t,R)$. The transformations for this procedure are rewritten here for completeness.

$$\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = \begin{bmatrix} -\sin(\lambda'-\lambda_E) & -\cos(\lambda'-\lambda_E)\sin\phi' & \cos(\lambda'-\lambda_E)\cos\phi' \\ \cos(\lambda'-\lambda_E) & -\sin(\lambda'-\lambda_E)\sin\phi' & \sin(\lambda'-\lambda_E)\cos\phi' \\ 0 & \cos\phi' & \sin\phi' \end{bmatrix} \begin{bmatrix} s' \\ t' \\ R \end{bmatrix} \tag{B.33}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} \tag{B.34}$$

$$\begin{bmatrix} s \\ t \\ R \end{bmatrix} = \begin{bmatrix} -\sin(\lambda-\lambda_E) & \cos(\lambda-\lambda_E) & 0 \\ -\cos(\lambda-\lambda_E)\sin\phi & -\sin(\lambda-\lambda_E)\sin\phi & \cos\phi \\ \cos(\lambda-\lambda_E)\cos\phi & \sin(\lambda-\lambda_E)\cos\phi & \sin\phi \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{B.35}$$

B.7 Earth to Grid Vector

Vector transformations from the Earth system to the grid system are also done in a three step process.

$$\vec{V}(s,t,R) \longrightarrow \vec{V}(a,b,c) \longrightarrow \vec{V}(a',b',c') \longrightarrow \vec{V}(s',t',R) \tag{B.36}$$

The long-lat Earth vector, $\vec{V}(s,t,R)$, is first expressed in Cartesian coordinates using equation (B.26). The Cartesian Earth vector, $\vec{V}(a,b,c)$, is rotated to the grid system with equation (B.30). The transformation is completed by expressing the Cartesian grid vector, $\vec{V}(a',b',c')$, in long-lat coordinates using equation (B.25) giving $\vec{V}(s',t',R)$. The transformations for this procedure are rewritten here for completeness.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -\sin(\lambda-\lambda_E) & -\cos(\lambda-\lambda_E)\sin\phi & \cos(\lambda-\lambda_E)\cos\phi \\ \cos(\lambda-\lambda_E) & -\sin(\lambda-\lambda_E)\sin\phi & \sin(\lambda-\lambda_E)\cos\phi \\ 0 & \cos\phi & \sin\phi \end{bmatrix} \begin{bmatrix} s \\ t \\ R \end{bmatrix} \tag{B.37}$$

$$\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{B.38}$$

$$\begin{bmatrix} s' \\ t' \\ R \end{bmatrix} = \begin{bmatrix} -\sin(\lambda'-\lambda_E) & \cos(\lambda'-\lambda_E) & 0 \\ -\cos(\lambda'-\lambda_E)\sin\phi' & -\sin(\lambda'-\lambda_E)\sin\phi' & \cos\phi' \\ \cos(\lambda'-\lambda_E)\cos\phi' & \sin(\lambda'-\lambda_E)\cos\phi' & \sin\phi' \end{bmatrix} \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} \tag{B.39}$$

Bibliography

1. R. Anderson, R. Pember, and N. Elliot. An arbitrary lagrangian-eulerian method with local structured adaptive mesh refinement for shock hydrodynamics. submitted to Concurrency: 40th AIAA Aerospace Sciences Meeting and Exhibit, UCRL-JC-141625, Lawrence Livermore National Laboratory, January 2002.

2. R. W. Anderson, N. S. Elliot, and R. B. Pember. An arbitrary lagrangian-eulerian method with adaptive mesh refinement for the solution of the euler equations. *J. Comput. Phys.*, 199:598–617, September 2004.

3. A. Arakawa and V. R. Lamb. Compuational design of the baisc dynamical processes of the UCLA general circulation model. *Methods in Computational Physics*, 17:174–265, 1977.

4. V. S. Arpaci. *Convection Heat Transfer*. Prentice Hall, 1984.

5. O. Aumont and J. C. Orr. Injection - HOWTO. Technical report, Ocean Carbon Cycle Intercomparison Project, http://www.ipsl.jussieu.fr/OCMIP, 2000.

6. G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.

7. J. Bell, M. Berger, J. Saltzman, and M. Welcome. Three dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comp.*, 15:127–138, 1994.

8. M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.

9. M. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

10. M. Berger and I. Rigoutsos. An algorithm for point clustering and grid generation. *IEEE Trans. on Systems, Man and Cybernetics*, 21:1278–1286, 1991.

11. E. Blayo and L. Debreu. Adaptive mesh refinement for finite-difference ocean models: First experiments. *J. Physical Oceanography*, 29:1239–1250, 1999.

12. K. Bryan. A numerical method for the study of the circulation of the world. *J. Comput. Phys.*, 4:347–376, 1969.

13. K. Bryan and M. D. Cox. An approximate equation of state for numerical models of ocean circulation. *J. Physical Oceanography*, 2:510–514, 1972.

14. K. Bryan, S. Manabe, and R. C. Pacanowski. A global ocean-atmosphere climate model. Part II. the ocean circulation. *J. Physical Oceanography*, 5:30–46, 1975.

15. K. Calderia and P. B. Duffy. The role of the southern ocean in uptake and storage of anthropogenic carbon dioxide. *Science*, 287:620–622, 2000.

16. K. Calderia and G. Rau. Accelerating carbonate dissolution to sequester carbon dioxide in the ocean: Geochemical implications. *Geophys. Res. Letters*, 27(2):225, 2000.

17. K. Calderia, M. E. Wickett, and P. B. Duffy. Depth, radiocarbon, and the effectiveness of direct CO_2 injection as an ocean carbon sequestration strategy. *Geophys. Res. Letters*, 1, August 2002.

18. P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, 87:171–200, 1990.

19. M. D. Cox. A primitive equation, three-dimensional model of the ocean. Technical Report 1, GFDL Ocean Group, Princeton University, Princeton, NJ 08542, 1984.

20. J. A. Curry and P. J. Webster. *Thermodynamics of Atmospheres and Oceans*. Academic Press, 1999.

21. G. Danabasoglu and J. C. McWilliams. Sensitivity of the global ocean circulation to parameterizations of mesoscale tracer transports. *J. of Climate*, 8:2967–2987, December 1995.

22. G. Danabasoglu, J. C. McWilliams, and W. G. Large. Approach to equilibrium in accelerated global oceanic models. *J. of Climate*, 9:1092–1110, May 1996.

23. Eric Deleersnijder and Jean-Michel Campin. On the computation of the barotropic mode of a free-surface world ocean model. *Annales Geophysicae*, 13:675–688, 1995.

24. P. B. Duffy, K. Calderia, J. Selvaggi, and M. I. Hoffert. Effects of subgrid-scale mixing parameterizations on simulated distributionsof natural C-14, temperature, and salinity in a three-dimensional ocean general circulation model. *J. Physical Oceanography*, 27(4):498–523, April 1997.

25. P. B. Duffy, P. Eltgroth, A. J. Bourgeois, and K. Calderia. Effect of improved subgrid scale transport of tracers on uptake of bomb radiocarbon in the GFDL ocean general circulation model. *Geophys. Res. Letters*, 22(9):1065–1068, May 1995.

26. A.D. Fox and S. Maskell. Two-way interactive nesting of primitive equation ocean models with topography. *J. Physical Oceanography*, 25:2977–2996, 1995.

27. Y. Fujioka, M. Ozaki, K. Takeuchi, Y. Shindo, and H. Herzog. Cost comparison in various CO_2 ocean disposal options. *Energy Convers. and Mgmt.*, 38:S273–S277, 1997.

28. Y. Fujioka, M. Ozaki, K. Takeuchi, Y. Shindo, Y. Yanagisawa, and H. Komiyama. Ocean CO_2 sequestration at the depths larger than 3700m. *Energy Convers. and Mgmt.*, 36:551–554, 1995.

29. A.E. Gill. *Atmosphere-Ocean Dynamics*. Academic Press, 1982.

30. I. Ginis, R. A. Richardson, and L. M. Rothstein. Design of a multiply nested primitive equation ocean model. *Monthly Weather Review*, 126:1054–1079, 1998.

31. G. Haltiner and R. Williams. *Numerical Prediction and Dynamic Meteorology*. John Wiley and Sons, 2 edition, 1980.

32. S. Hankin, J. Sirott, J. Callahan, K. OBrien, and A. Manke. *A Tool For Web Access To In Situ Data Collections, Too*. National Oceanic and Atmospheric Administration, http://ferret.pmel.noaa.gov/NVODS/servlets/dataset, Jan 2002.

33. S. Hellerman and M. Rosenstein. Normal monthly wind stress over the world ocean with error estimates. *J. Physical Oceanography*, 13:1093–1104, 1983.

34. A. Herrnstein, M. Wickett, and G. Rodrigue. Structured adaptive mesh refinement using leapfrog time integration on a staggered grid for ocean models. *Ocean Modelling*, 9:283–304, 2005.

35. H. J. Herzog, K. Caldeira, and E. Adams. Carbon sequestration via direct injection. In J. H. Steele, S. A. Thorpe, and K. K. Turekian, editors, *Encyclopedia of Ocean Sciences*, volume 1, pages 408–414, San Diego, CA, 2001. Academic.

36. C. Hill, V. Bugnion, M. Follows, and J. Marshall. Evaluating carbon sequestration efficiency in an ocean circulation model by adjoint sensitivity analysis. *J. Geophys. Res.*, 109:C11005, 2004.

37. C. Hirsch. *Numerical Computation of Internal and External Flows Volume 1: Fundamentals of Numerical Discretization*. John Wiley and Sons, 1988.

38. R. Hornung. *SAMRAI Structured Adaptive Mesh Refinement Application Infrastructure*. Lawrence Livermore National Laboratory, www.llnl.gov/CASC/SAMRAI/, February 2004. UCRL-WEB-202409.

39. R. Hornung and S. Kohn. Managing application complexity in the samrai object-oriented framework. *Concurrency: Practice and Experience*, 14:347–368, 2002.

40. P. D. Killworth, D. Stainforth, D. J. Webb, and S. M. Paterson. The development of a free-surface Bryan-Cox-Semtner ocean model. *American Meteorlogical Society*, 21:1333–1348, 1991.

41. P. K. Kundu. *Fluid Mechanics*. Academic Press, Inc., 1990.

42. Y. Kurihara and M. A. Bender. Use of a movable nested-mesh model for tracking a small vortex. *Monthly Weather Review*, 108:1792–1809, 1980.

43. Y. Kurihara, G. J. Tripoli, and M. A. Bender. Design of a movable nested-mesh primitive equation model. *Monthly Weather Review*, 107:239–249, 1979.

44. M. Laugier, P. Angot, and L. Mortier. Nested grid methods for an ocean model: A comparitive study. *Internat. J. Numer. Methods Fluids*, 23:1163–1195, 1996.

45. Levitus. NODC World Ocean Atlas, NOAA-CIRES Climate Diagnostics Center, Boulder, Colorado, 1998. http://www.cdc.noaa.gov/.

46. C. Marchetti. On geoengineering and the CO_2 problem. *Clim. Change.*, 1:59–68, 1977.

47. James C. McWilliams, Glenn R. Flierl, Vitaly D. Larichev, , and Gregory M. Reznik. Numerical studies of barotropic modons. *Dynamics of Atmospheres and Oceans*, 5:219–239, 1981.

48. W. H. Munk. On the wind-driven ocean circulation. *J. Meteor.*, 7:79–93, 1950.

49. R. J. Murray. Explicit generation of orthogonal grids for ocean models. *J. Comput. Phys.*, 126:251–273, 1996.

50. J. C. Orr and et al. Ocean CO_2 sequestration efficiency from a 3-D ocean model comparison. In D. Williams, editor, *Proceedings of the 5th International Conference on Greenhouse Gas Control Technologies*, pages 469–474, Collingwood, Aust., 2001. Commonw. Sci. and Ind. Res. Org.

51. Andreas Oschlies. On spurios interactions among a mixed layer model, convective adjustment, and isopycnal mixing in ocean circulation models. *Monthly Weather Review*, 127:1920–1927, 1999.

52. M. Ozaki, K. Sonoda, Y. Fujioka O. Tsukamoto, and M. Komatsu. Sending CO_2 into deep ocean with a hanging pipe from floating platform. *Energy Convers. and Mgmt.*, 36:475–478, 1995.

53. R. Pacanowski and S. Griffies. *MOM 3.0 Manual*. NOAA/Geophysical Fluid Dynamics Laboratory, Princeton, USA 08542, May 1999. ftp://ftp.gfdl.gov/pub/GFDL_MOM3/.

54. Germana Peggion. Numerical inaccuracies across the interface of a nested grid. *Num. Meth. Part. Diff. Eqs.*, 10:455–473, 1994.

55. T. H. Peng, T. Takahashi, W. S. Broecker, and J. Olafsson. Seasonal variability of carbon dioxide, nutrients, and oxygen in the northern north atlantic surface water: Observations and a model. *Tellus, Ser. B*, 39:439–458, 1987.

56. A.L. Perkins, L.F. Smedstad, D.W. Blake, G.W. Heburn, and A.J. Wallcraft. A new nested boundary condition for a primitive equaiton ocean model. *J. Geophys. Res.*, 102:3483–3500, 1997.

57. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing*. Press Syndicate of the University of Cambridge, 2 edition, 1992.

58. G. Rau and K. Calderia. Enhanced carbonate dissolution: A means of sequestering waste CO_2 as ocean bicarbonate. *Energy Convers. and Mgmt.*, 40(17):1803–1813, 1998.

59. C. Rowley and I. Ginis. Implementiation of a mesh movement scheme in a multiply nested ocean model and its application to air-sea interaction studies. *Monthly Weather Review*, 127:1879–1896, 1999.

60. J. L. Sarmiento, J. C. Orr, and U. Siegenthaler. A perturbation simulation of CO_2 uptake in an ocean general circulation model. *J. Geophys. Res.*, 97:3621–3645, 1992.

61. A. J. Semtner. An oceanic general circulation model with bottom topography. Technical Report 9, UCLA, Department of Meteorology, 1974.

62. W. Skamarock, J. Oliger, and R. Street. Adaptive grid refinement for numerical weather prediction. *J. Comput. Phys.*, 80:27–60, 1989.

63. Peter W. Sloss. *ETOPO2 - 2 minute Worldwide Bathymetry/Topography*. National Oceanic and Atmospheric Administration, National Geophysical Data Center, Boulder, Colorado, September 2001. http://dss.ucar.edu/datasets/ds759.3/data/doc_1.

64. R. Smith and P. Gent. *Reference Manual for the Parallel Ocean Program (POP)*. LANL, NCAR, May 2002.

65. A. Spall and W. Holland. A nested primitive equation model for oceanic applications. *J. Physical Oceanography*, 21:205–220, 1991.

66. G. R. Stegen, K. H. Cole, and R. Bacastow. The influence of discharge depth and location on the sequestration of carbon dioxide. *Energy Convers. and Mgmt.*, 34:857–864, 1993.

67. H. Stommel. The westward intensification of wind-driven ocean currents. *Trans. Am. Geophys. Union.*, 29:202–206, 1948.

68. M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, 148:81–124, 1999.

69. S. Timoshenko. *An Introduction to Fluid Dynamics*. McGraw-Hill Companies, 3 edition, 1970.

70. A. M. Treguier, J. K. Dukowicz, and K. Bryan. Properties of nonuniform grids used in ocean general circulation models. *J. Geophys. Res.*, 101:20877–20881, 1996.

71. UNESCO. Tenth report of the joint panel on oceanographic tables and standards. Technical Report 36, UNESCO Technical Papers in Marine Science, Paris, 1981.

72. B. van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *J. Comput. Phys.*, 23:276–299, 1977.

73. G. Veronis. Wind-driven ocean circulation-part 1. Linear theory and perturbation analysis. *Deap Sea Research*, 13:17–29, 1966.

74. W. Washington and C. Parkinson. *An Introduction to Three-Dimensional Climate Modeling*. Oxford University Press, 1986.

75. F. W. White. *Viscous Fluid Flow*. McGraw-Hill Science/Engineering/Math, 2 edition, 1991.

76. M. E. Wickett, K. Calderia, and P. B. Duffy. Effect of horizontal grid resolution on simulations of oceanic CFC-11 uptake and direct injection of anthropogenic CO_2. *J. Geophys. Res.*, 108:3189, 2003.

77. M. E. Wickett, P. B. Duffy, and G. Rodrigue. A reduced grid for a parallel global ocean general circulation model. *Ocean Modelling*, 2:85–107, 2000.

78. Michael E. Wickett. *A Reduced Grid Method for a Parallel Global Ocean General Circulation Model*. PhD thesis, University of California, Davis, December 1999.

79. D. Zhang, H. Chang, N. Seaman, T. Warner, and J. Fritsch. A two-way interactive nesting procedure with variable terrain resolution. *Monthly Weather Review*, 114:1330–1339, 1986.

You do the Hokie Pokie and you turn yourself around, that's what it's all about!