Adaptive Multigrid Solution of the Shallow Water Equations

Brittany L. Mitchell
Scott R. Fulton

May 2000
(revised 8/8/2000)

Abstract

This paper describes the formulation and testing of a two-dimensional tropical cyclone track model which solves the shallow water equations with open boundary conditions using local mesh refinement. An Arakawa C-grid variable staggering scheme is used to correctly model the physics of the problem. The shallow water equations are discretized in space using second-order centered finite difference approximations. A semi-implicit time discretization scheme is utilized to satisfy the stability condition while allowing longer time steps for efficiency. A Helmholtz problem for the geopotential is derived from the shallow water equations and solved by a multigrid method. Numerical results show that the multigrid method converges at the rate predicted by smoothing analysis, and that the boundary conditions used permit gravity waves to freely propagate between adaptive patches and the coarser base grid. Compared to uniform-grid results, using local mesh refinement can achieve either the same accuracy with up to an order of magnitude less computational work or up to an order of magnitude improvement in accuracy with the same computational work.

Technical Report No. 2000-02
Department of Mathematics and Computer Science
Clarkson University, Potsdam, New York

This work was supported by the
Office of Naval Research
Marine Meteorology and Atmospheric Effects Program
Grants N00014-98-1-0103 and N00014-98-1-0368
Contents

1 Introduction ............................................. 1

2 Governing Equations .................................. 2

3 Discretization ......................................... 4
   3.1 Time discretization .................................. 4
   3.2 Space discretization .................................. 6
   3.3 The discrete Helmholtz problem ....................... 10
   3.4 Summary of discretization ............................. 12

4 Multigrid Method ....................................... 13
   4.1 Gauss-Seidel relaxation ............................... 13
   4.2 A two-grid method ..................................... 15
   4.3 Grid transfers .......................................... 16
   4.4 The multigrid V-cycle .................................. 16
   4.5 The Full Multigrid algorithm ......................... 17
   4.6 Local mesh refinement .................................. 18

5 Numerical Results ..................................... 19
   5.1 Multigrid convergence .................................. 19
   5.2 Gravity wave experiment .............................. 20
   5.3 Sample run ............................................. 23
   5.4 Accuracy vs. efficiency .............................. 23

6 Conclusions ............................................ 26
1 Introduction

A first attempt at numerical weather prediction was made by Richardson in the early 1920s. Despite his failed attempts, he documented his work in a book[17] which became the starting point for numerical meteorology. In 1928, Courant, Fredrichs, and Lewy partially explained Richardson’s problems, showing that space and time discretizations needed to meet certain stability criteria. In 1945, the first electronic computer was constructed; using this to solve the absolute vorticity conservation equation, Charney, Fjortoft, and Von Neumann[5] made the first numerical weather prediction. Since then, the field of numerical weather prediction has blossomed due to improved understanding of atmospheric dynamics, development of new numerical techniques, and continually increasing computational speed and storage capacity.

Since the full equations describing atmospheric motions (e.g., conservation of momentum, mass, and energy) are quite complicated, solving them usually requires some sort of dynamical approximation. The simplest dynamical models are barotropic, i.e., two-dimensional models based on the assumption that the flow does not vary in the vertical. In some situations these simple models are adequate for practical forecasting. For example, barotropic hurricane track models[6, 7] are routinely used for operational forecasting, since in many situations they perform as well as more complicated baroclinic (three-dimensional) models.

Likewise, even after making various dynamical approximations, the resulting equations are usually too complicated to solve exactly, and some sort of numerical approximation must be used. Most limited-area models (those not encompassing the global atmosphere) use some sort of finite-difference (or occasionally finite-element) approximation. In most situations, the resolution required varies across the domain. For example, in the problem of interest here—that of a tropical cyclone moving in a larger-scale flow—the spatial scales of variation differ by at least two orders of magnitude between the small-scale vortex and the surrounding flow. In such situations, uniform resolution is inherently inefficient, and some sort of local mesh refinement is called for.

One promising method combines multigrid processing[2] and adaptive mesh refinement[1], resulting in an adaptive multigrid method. Such a method has been applied to the simplest possible dynamical model of hurricane motion, namely, the nondivergent barotropic model (which simply expresses the conservation of absolute vorticity). The resulting MUDBAR model has been shown to be accurate and efficient [9, 10]. The goal of this paper is to extend this method to the next level of dynamical complexity, namely, the shallow water equations. While these equations are still barotropic (two-dimensional), they incorporate more dynamics (e.g., gravity-inertia waves), and serve both as a better test bed for numerical methods and as the basis of practical models (cf. [7]). A similar model has been proposed for ocean modeling[18]; the boundary conditions, discretization, and overall goal are somewhat different from the present study.

This paper is organized as follows. Section 2 describes the governing equations and boundary conditions in continuous form, and section 3 describes the time and space discretizations. The multigrid method used to solve the resulting Helmholtz problem is described in section 4. Section 5 describes the numerical results, and Section 6 details the conclusions of this work.
2 Governing Equations

The shallow water equations govern the flow of a layer of incompressible fluid over a flat boundary. The principal assumption behind the shallow water equations is that the horizontal length scale of the flow is much greater than the height of the fluid (for example, the ocean can be considered a shallow fluid because it is hundreds of miles wide, while its depth rarely exceeds two miles). While the atmosphere is not an incompressible fluid, the shallow water equations serve as a useful approximation, including the important effects of horizontal advection and propagation of gravity waves while maintaining the simplicity of a two-dimensional system. For a derivation of the shallow water equations from the incompressible Navier-Stokes equations, see [15].

In the shallow water system the dependent variables are the velocity components $u$ and $v$ in the $x$ (east) and $y$ (north) directions, respectively, and the geopotential $\phi = gz$, where $z$ is the free surface height as shown in Fig. 1 and $g$ is the (constant) acceleration due to gravity. The equations of conservation of momentum and mass take the form

$$u_t + uu_x + v u_y -fv + \phi_x = 0,$$

$$v_t + uv_x + v v_y + fu + \phi_y = 0,$$

$$\phi_t + (u_x + v_y)(\phi - \Phi) + u\phi_x + v\phi_y + \Phi(u_x + v_y) = 0,$$

where $t$ is time and subscripts denote partial derivatives. The Coriolis parameter $f$ expresses the effects of the Earth’s rotation; here we will make the so-called $\beta$-plane approximation, using the linear approximation $f(y) = f_0 + \beta y$, where $f_0 = 2\omega \sin(\theta_0)$ is the value of $f$ at the latitude $\theta_0$ (where $y = 0$), and $\beta = (2\omega /a) \cos(\theta_0)$ is $df/dy$ there ($\omega$ and $a$ are the rotation rate and radius of the Earth, respectively). In the geopotential equation (3), $\Phi$ represents a constant (positive) reference value of geopotential; this splitting allows us to isolate the terms which give rise to gravity-inertia waves and treat them implicitly (see section 3).

![Figure 1: Variables in the shallow water system](image)
The problem domain is a rectangle \( \Omega = [-L_x, L_x] \times [-L_y, L_y] \), centered at \((x, y) = (0, 0)\) where \( f = f_0 \). As there are no physical boundaries in the real atmosphere, we use open boundary conditions which allow gravity-inertia waves to propagate out of the domain. If linearized about a basic state with \( u = 0, v = 0, \) and \( \phi = \Phi \) (and \( f \) is taken to be zero), the solutions of (1)–(3) can be written in terms of characteristic quantities \( u + \phi/c \) and \( u - \phi/c \) propagating with speed \( c \) in the \(+x\) and \(-x\) directions and \( v + \phi/c \) and \( v - \phi/c \) propagating with speed \( c \) in the \(+y\) and \(-y\) directions (where \( c = \sqrt{\Phi} \)). Specifying only the incoming quantities leads to the open boundary conditions

\[
\begin{align*}
    u(-L_x, y, t) + \frac{\phi(-L_x, y, t)}{c} &= u_s + \frac{\phi_s}{c}, \\
    u(+L_x, y, t) - \frac{\phi(+L_x, y, t)}{c} &= u_s - \frac{\phi_s}{c}, \\
    v(x, -L_y, t) + \frac{\phi(x, -L_y, t)}{c} &= v_s + \frac{\phi_s}{c}, \\
    v(x, +L_y, t) - \frac{\phi(x, +L_y, t)}{c} &= v_s - \frac{\phi_s}{c},
\end{align*}
\]

where \( u_s, v_s, \) and \( \phi_s \) are functions specified at the boundary. Since \( f \) is nonzero in the model, we must also specify the tangential component of the flow (i.e., \( v \) at \( x = \pm L_x \) and \( u \) at \( y = \pm L_y \)) where there is inflow. These open boundary conditions were used in a spectral shallow water model[12] and were shown by Oliger and Sundström[16] to give a well-posed problem.
3 Discretization

To solve the shallow water equations numerically we must discretize them. In this section we give the details of the time discretization (section 3.1) and space discretization (section 3.2). The resulting system of discrete equations is reduced to a single equation of the Helmholtz type (section 3.3), which will then be solved by a multigrid method as discussed in section 4.

3.1 Time discretization

The shallow water system admits two distinct types of motion: slow, quasi-steady flow approximately in geostrophic balance and fast, propagating gravity-inertia waves. Since in the cases we wish to study the former is the type of motion of interest—and there is little energy in the latter—it is advantageous to discretize the terms associated with these two modes separately. Treating the terms associated with gravity waves implicitly and the rest explicitly yields a semi-implicit discretization; this idea was introduced in 1971 by Kwizak and Robert[13] and has been used extensively since then.

Here we will use the leapfrog method and the trapezoidal method for the terms to be treated explicitly and implicitly, respectively. Using a superscript $n$ to denote (approximate) values at time $t_n = n\Delta t$, the time-discrete approximations to (1)–(3) are

\[
\frac{u^{n+1} - u^{n-1}}{2\Delta t} + (u^n u_x^n + v^n u_y^n - f v^n) + \left(\frac{\phi_x^{n+1} + \phi_x^{n-1}}{2}\right) = 0, \tag{8}
\]

\[
\frac{v^{n+1} - v^{n-1}}{2\Delta t} + (u^n v_x^n + v^n v_y^n + f u^n) + \left(\frac{\phi_y^{n+1} + \phi_y^{n-1}}{2}\right) = 0, \tag{9}
\]

\[
\frac{\phi^{n+1} - \phi^{n-1}}{2\Delta t} + (\phi^n - \Phi)(v_x^n + v_y^n) + u^n \phi_x^n + v^n \phi_y^n + \Phi \left(\frac{u^{n+1} + u^{n-1}}{2} + \frac{v^{n+1} + v^{n-1}}{2}\right) = 0. \tag{10}
\]

Regarding the values at $t_n$ and $t_{n-1}$ as known, this gives a system of equations to be solved for the values at $t_{n+1}$.

For the initial time step ($n = 0$) the values at $t_n$ are obtained from the specified initial conditions, but the values at $t_{n-1}$ are not known. Consequently, we replace the leapfrog method for this initial step with a second-order Runge-Kutta method, with the first stage using an Euler step (of length $\frac{1}{2}\Delta t$) given by

\[
\frac{u^{n+1/2} - u^n}{\frac{1}{2}\Delta t} + (u^n u_x^n + v^n u_y^n - f v^n) + \left(\frac{\phi_x^{n+1/2} + \phi_x^n}{2}\right) = 0, \tag{11}
\]

\[
\frac{v^{n+1/2} - v^n}{\frac{1}{2}\Delta t} + (u^n v_x^n + v^n v_y^n + f u^n) + \left(\frac{\phi_y^{n+1/2} + \phi_y^n}{2}\right) = 0. \tag{12}
\]
\[
\frac{\phi^{n+1/2} - \phi^n}{\frac{1}{2}\Delta t} + (\phi^n - \Phi)(u^n + v^n) + u^n \phi^n_x + v^n \phi^n_y + \Phi \left( \frac{u^{n+1/2} + u^n}{2} + \frac{v^{n+1} + v^n}{2} \right) = 0, \quad (13)
\]

and the second stage using a leapfrog step (also of length $\frac{1}{2}\Delta t$) given by
\[
\frac{u^{n+1} - u^n}{\Delta t} + (u^{n+1/2} u^{n+1/2} + v^{n+1/2} v^{n+1/2} - f_{v^{n+1/2}}) + \left( \frac{\phi^{n+1} + \phi^n}{2} \right) = 0, \quad (14)
\]
\[
\frac{v^{n+1} - v^n}{\Delta t} + (u^{n+1/2} v^{n+1/2} + v^{n+1/2} v^{n+1/2} + f_{u^{n+1/2}}) + \left( \frac{\phi^{n+1} + \phi^n}{2} \right) = 0, \quad (15)
\]
\[
\frac{\phi^{n+1} - \phi^n}{\Delta t} + (\phi^{n+1/2} - \Phi)(u^{n+1/2} + v^{n+1/2})
\]
\[
+ u^{n+1/2} \phi^{n+1/2} + v^{n+1/2} \phi^{n+1/2} + \Phi \left( \frac{u^{n+1} + u^n}{2} + \frac{v^{n+1} + v^n}{2} \right) = 0. \quad (16)
\]

In each of the above cases we obtain a system of three equations for the unknowns $u$, $v$, and $p$ at a new time level. Putting known quantities on the right, we can write each of the systems (8)–(10), (11)–(13), and (14)–(16) in the form
\[
\tilde{u} + \eta \Delta t \tilde{\phi}_x = U(\eta \Delta t, u, v, \tilde{u}, \tilde{\phi}), \quad (17)
\]
\[
v + \eta \Delta t \tilde{\phi}_y = V(\eta \Delta t, u, v, \tilde{u}, \tilde{\phi}), \quad (18)
\]
\[
\tilde{\phi} + \Phi \eta \Delta t (\tilde{u}_x + \tilde{v}_y) = P(\eta \Delta t, u, v, \tilde{u}, \tilde{v}), \quad (19)
\]
where the operators $U$, $V$, and $P$ have the form
\[
U(\eta \Delta t, u, v, \tilde{u}, \tilde{\phi}) := \tilde{u} - \eta \Delta t \tilde{\phi}_x - 2\eta \Delta t (uv_x + vu_y - f v), \quad (20)
\]
\[
V(\eta \Delta t, u, v, \tilde{u}, \tilde{\phi}) := \tilde{v} - \eta \Delta t \tilde{\phi}_y - 2\eta \Delta t (uv_x + vu_y + f u), \quad (21)
\]
\[
P(\eta \Delta t, u, v, \tilde{u}, \tilde{v}) := \tilde{\phi} - \Phi \eta \Delta t (\tilde{u}_x + \tilde{v}_y)
\]
\[
- 2\eta \Delta t \left[ (\phi - \Phi) (u_x + v_y) + u \phi_x + v \phi_y \right]. \quad (22)
\]

Here, $(\tilde{u}, \tilde{v}, \tilde{\phi})$ are the unknowns at the new time level, $(u, v, \phi)$ are the current values, and $(\tilde{u}, \tilde{v}, \tilde{\phi})$ are previous values (for the leapfrog steps), with $\eta$ denoting the length of the step. Thus, for the full leapfrog step (8)–(10) we have
\[
(\tilde{u}, \tilde{v}, \tilde{\phi}) = (u^{n+1}, v^{n+1}, \phi^{n+1}), \quad (\tilde{u}, \tilde{v}, \tilde{\phi}) = (u^{n-1}, v^{n-1}, \phi^{n-1}), \quad \eta = 1, \quad (23)
\]
for the initial Euler step (11)–(13) with $n = 0$ we have
\[
(\tilde{u}, \tilde{v}, \tilde{\phi}) = (u^{n+1/2}, v^{n+1/2}, \phi^{n+1/2}), \quad (\tilde{u}, \tilde{v}, \tilde{\phi}) = (u^n, v^n, \phi^n), \quad \eta = \frac{1}{2}, \quad (24)
\]
and for the initial leapfrog step (14)–(16) with $n = 0$ we have
\[
(\tilde{u}, \tilde{v}, \tilde{\phi}) = (u^{n+1}, v^{n+1}, \phi^{n+1}), \quad (\tilde{u}, \tilde{v}, \tilde{\phi}) = (u^n, v^n, \phi^n), \quad \eta = \frac{1}{2}. \quad (25)
\]
3.2 Space discretization

To discretize the problem in space we use second-order finite differences on a single uniform grid (for now) with mesh spacing $h$, given by

$$\Omega_h = \{(x_i, y_j) = (x_0 + i h, y_0 + j h) : 0 \leq i \leq M, \, 0 \leq j \leq N\}. \quad (26)$$

We use the Arakawa C-grid staggering for the variables $u$, $v$, and $\phi$ as shown in Fig. 2. This arrangement of variables has been shown to give the best representation of the geostrophic adjustment process[14]. Note that values of velocity are also carried at the “ghost points”, i.e., points outside the computational domain; these will be used in discretizing the boundary conditions.

![Variable staggering on the Arakawa C-grid](image)

Figure 2: Variable staggering on the Arakawa C-grid

Using second-order centered finite difference, the space-discretized form of the implicit system (17)–(19) is

$$\begin{align*}
\bar{u}_{i+1/2,j} + \eta \Delta t \left( \frac{\bar{\phi}_{i+1,j} - \bar{\phi}_{i,j}}{h} \right) &= U_{i+1/2,j}, \quad 0 \leq i < M, \quad 0 \leq j \leq N \quad (27) \\
\bar{v}_{i,j+1/2} + \eta \Delta t \left( \frac{\bar{\phi}_{i,j+1} - \bar{\phi}_{i,j}}{h} \right) &= V_{i,j+1/2}, \quad 0 \leq i \leq M, \quad 0 \leq j < N \quad (28) \\
\bar{\phi}_{i,j} + \Phi \eta \Delta t \left( \frac{\bar{u}_{i+1/2,j} - \bar{u}_{i-1/2,j}}{h} + \frac{\bar{v}_{i,j+1/2} - \bar{v}_{i,j-1/2}}{h} \right) &= P_{i,j}, \quad 0 \leq i \leq M, \quad 0 \leq j < N, \quad (29)
\end{align*}$$

where the subscripts $i$ and $j$ refer to values at the points $(x_i, y_j)$. Note that these equations are applied at the interior points and boundary points but not the ghost points.
For the right-hand side, we must discretize (20)–(22) differently on the interior and boundaries. At the interior points, we obtain

\[
U_{i+1/2,j} := \tilde{u}_{i+1/2,j} - \eta \Delta t \left( \frac{\tilde{\phi}_{i+1,j} - \tilde{\phi}_{i,j}}{h} \right) - 2\eta \Delta t \left[ \frac{u_{i+3/2,j} - u_{i-1/2,j}}{2h} \right] + \left( \frac{u_{i+1,j+1/2} + u_{i+1,j-1/2} + v_{i,j+1/2} + v_{i,j-1/2}}{4} \right) \left( \frac{u_{i+1/2,j+1} - u_{i+1/2,j-1}}{2h} - f \right),
\]

(30)

\[
V_{i,j+1/2} := \tilde{v}_{i,j+1/2} - \eta \Delta t \left( \frac{\tilde{\phi}_{i,j+1} - \tilde{\phi}_{i,j}}{h} \right) - 2\eta \Delta t \left[ \frac{v_{i,j+3/2} - v_{i,j-1/2}}{2h} \right] + \left( \frac{u_{i+1/2,j+1} + u_{i+1/2,j} + u_{i-1/2,j+1} + u_{i-1/2,j}}{4} \right) \left( \frac{v_{i+1,j+1/2} - v_{i-1,j+1/2}}{2h} + f \right),
\]

(31)

and

\[
P_{i,j} := \tilde{\phi}_{i,j} - \Phi \eta \Delta t \left( \frac{\tilde{u}_{i+1/2,j} - \tilde{u}_{i-1/2,j} + \tilde{v}_{i,j+1/2} - \tilde{v}_{i,j-1/2}}{h} \right)
\]

\[- 2\eta \Delta t \left( \phi_{i,j} - \Phi \right) \left( \frac{u_{i+1/2,j} - u_{i-1/2,j} + v_{i,j+1/2} - v_{i,j-1/2}}{h} \right) + \left( \frac{u_{i+1/2,j} + u_{i-1/2,j}}{2} \right) \left( \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h} \right) + \left( \frac{v_{i,j+1/2} + v_{i,j-1/2}}{2} \right) \left( \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h} \right). \]

(32)

At the boundaries, we use discretizations which are identical except that any differences requiring values at non-existent points outside the boundaries are replaced by one-sided differences; this approach is analogous to that proposed by Elvius and Sundström[8]. Thus, at the west boundary, the new expressions for \( V \) and \( P \) are

\[
V_{0,j+1/2} := \tilde{v}_{0,j+1/2} - \eta \Delta t \left( \frac{\tilde{\phi}_{0,j+1} - \tilde{\phi}_{0,j}}{h} \right) - 2\eta \Delta t \left[ \frac{v_{0,j+3/2} - v_{0,j-1/2}}{2h} \right] + \left( \frac{u_{-1/2,j} + u_{-1/2,j+1} + u_{1/2,j} + u_{1/2,j+1}}{4} \right) \left( \frac{v_{1,j+1/2} - v_{0,j+1/2}}{h} + f \right),
\]

(33)

and

\[
P_{0,j} := \tilde{\phi}_{0,j} - \Phi \eta \Delta t \left( \frac{\tilde{u}_{1/2,j} - \tilde{u}_{-1/2,j} + \tilde{v}_{0,j+1/2} - \tilde{v}_{0,j-1/2}}{h} \right)
\]

\[- 2\eta \Delta t \left( \phi_{0,j} - \Phi \right) \left( \frac{u_{1/2,j} - u_{-1/2,j} + v_{0,j+1/2} - v_{0,j-1/2}}{h} \right) + \left( \frac{u_{1/2,j} + u_{-1/2,j}}{2} \right) \left( \frac{\phi_{1,j} - \phi_{0,j}}{h} \right) + \left( \frac{v_{0,j+1/2} + v_{0,j-1/2}}{2} \right) \left( \frac{\phi_{0,j+1} - \phi_{0,j-1}}{2h} \right). \]

(34)
At the east boundary, the new expressions for \( V \) and \( P \) are

\[
V_{M,j+1/2} := \tilde{v}_{M,j+1/2} - \eta \Delta t \left( \frac{\tilde{\phi}_{M,j+1} - \tilde{\phi}_{M,j}}{h} \right) - 2\eta \Delta t \left[ v_{M,j+1/2} \left( \frac{v_{M,j+3/2} - v_{M,j+1/2}}{2h} \right) \right.
+ \left( \frac{u_{M-1/2,j} + u_{M-1/2,j+1} + u_{M+1/2,j} + u_{M+1/2,j+1}}{4} \right) \left( \frac{v_{M,j+1/2} - v_{M-1,j+1/2}}{h} \right) + f \right] \]

and

\[
P_{M,j} := \tilde{\phi}_{M,j} - \Phi \eta \Delta t \left( \frac{\tilde{u}_{M+1/2,j} - \tilde{u}_{M-1/2,j} + \tilde{v}_{M,j+1/2} - \tilde{v}_{M,j-1/2}}{h} \right)
- 2\eta \Delta t \left[ \left( \phi_{M,j} - \Phi \right) \left( \frac{u_{M+1/2,j} - u_{M-1/2,j} + v_{M,j+1/2} - v_{M,j-1/2}}{h} \right) \right.
+ \left( \frac{u_{M+1/2,j} + u_{M-1/2,j}}{2} \right) \left( \frac{\phi_{M,j} - \phi_{M-1,j}}{h} \right)
+ \left( \frac{v_{M,j+1/2} + v_{M,j-1/2}}{2} \right) \left( \frac{\phi_{M,j+1} - \phi_{M,j-1}}{2h} \right) \right]. \tag{36}
\]

At the south boundary, the new expressions for \( U \) and \( P \) are

\[
U_{i+1/2,0} := \tilde{u}_{i+1/2,0} - \eta \Delta t \left( \frac{\tilde{\phi}_{i+1,0} - \tilde{\phi}_{i,0}}{h} \right) - 2\eta \Delta t \left[ u_{i+1/2,0} \left( \frac{u_{i+1,0} - u_{i,0}}{2h} \right) \right.
+ \left( \frac{v_{i,-1/2} + v_{i,1/2} + v_{i+1,1/2} + v_{i+1,-1/2}}{4} \right) \left( \frac{u_{i+1,2} - u_{i+1,0}}{h} \right) - f \right] \]

and

\[
P_{i,0} := \tilde{\phi}_{i,0} - \Phi \eta \Delta t \left( \frac{\tilde{u}_{i+1/2,0} - \tilde{u}_{i-1/2,0} + \tilde{v}_{i,1/2} - \tilde{v}_{i,-1/2}}{h} \right)
- 2\eta \Delta t \left[ \left( \phi_{i,0} - \Phi \right) \left( \frac{u_{i+1,0} - u_{i-1,0} + v_{i,1/2} - v_{i,-1/2}}{h} \right) \right.
+ \left( \frac{u_{i+1,0} + u_{i-1,0}}{2} \right) \left( \frac{\phi_{i+1,0} - \phi_{i-1,0}}{h} \right)
+ \left( \frac{v_{i,1} + v_{i,-1}}{2} \right) \left( \frac{\phi_{i+1} - \phi_{i-1}}{2h} \right) \right]. \tag{38}
\]

At the north boundary the new expressions for \( U \) and \( P \) are

\[
U_{i+1/2,N} := \tilde{u}_{i+1/2,N} - \eta \Delta t \left( \frac{\tilde{\phi}_{i+1,N} - \tilde{\phi}_{i,N}}{h} \right) - 2\eta \Delta t \left[ u_{i+1/2,N} \left( \frac{u_{i+1,N} - u_{i,N}}{2h} \right) \right.
+ \left( \frac{v_{i,N-1/2} + v_{i,N+1/2} + v_{i+1,N+1/2} + v_{i+1,N-1/2}}{4} \right) \left( \frac{u_{i+1,2,N} - u_{i+1,2,N-1}}{h} \right) - f \right] \]

\[
+ \left( \frac{u_{i,N-1/2} + u_{i,N+1/2} + v_{i+1,N+1/2} + v_{i+1,N-1/2}}{4} \right) \left( \frac{u_{i+1,2,N} - u_{i+1,2,N-1}}{h} \right) \left( \frac{v_{i+1,2,N} - v_{i+1,2,N-1}}{2h} \right) \right] \tag{39}
\]
and
\[ P_{i,N} := \bar{\phi}_{i,N} - \Phi \eta \Delta t \left( \frac{\bar{u}_{i+1/2,N} - \bar{u}_{i-1/2,N} + \bar{v}_{i,N+1/2} - \bar{v}_{i,N-1/2}}{h} \right) \]
\[ - 2\eta \Delta t \left[ (\phi_{i,N} - \Phi) \left( \frac{u_{i+1/2,N} - u_{i-1/2,N} + v_{i,N+1/2} - v_{i,N-1/2}}{h} \right) \right] \]
\[ + \left( \frac{u_{i+1/2,N} + u_{i-1/2,N}}{2} \right) \left( \frac{\phi_{i+1,N} - \phi_{i-1,N}}{2h} \right) + \left( \frac{v_{i,N+1/2} + v_{i,N-1/2}}{2} \right) \left( \frac{\phi_{i,N} - \phi_{i,N-1}}{h} \right) \] \hspace{1cm} (40)

Likewise, at the corner points we must use one-sided differences for the derivatives of \( \phi \). At the southwest corner the new expression for \( P \) is
\[ P_{0,0} := \bar{\phi}_{0,0} - \Phi \eta \Delta t \left( \frac{\bar{u}_{1/2,0} - \bar{u}_{-1/2,0} + \bar{v}_{1,0} - \bar{v}_{-1,0}}{h} \right) \]
\[ - 2\eta \Delta t \left[ (\phi_{0,0} - \Phi) \left( \frac{u_{1/2,0} - u_{-1/2,0} + v_{1,0} - v_{-1,0}}{h} \right) \right] \]
\[ + \left( \frac{\bar{u}_{1/2,0} + \bar{u}_{-1/2,0}}{2} \right) \left( \frac{\phi_{1,0} - \phi_{0,0}}{h} \right) + \left( \frac{\bar{v}_{1,0} + \bar{v}_{-1,0}}{2} \right) \left( \frac{\phi_{1,0} - \phi_{0,0}}{h} \right) \], \hspace{1cm} (41)

at the southeast corner the new expression for \( P \) is
\[ P_{M,0} := \bar{\phi}_{M,0} - \Phi \eta \Delta t \left( \frac{\bar{u}_{M+1/2,0} - \bar{u}_{M-1/2,0} + \bar{v}_{M,1/2} - \bar{v}_{M,-1/2}}{h} \right) \]
\[ - 2\eta \Delta t \left[ (\phi_{M,0} - \Phi) \left( \frac{u_{M+1/2,0} - u_{M-1/2,0} + v_{M,1/2} - v_{M,-1/2}}{h} \right) \right] \]
\[ + \left( \frac{\bar{u}_{M+1/2,0} + \bar{u}_{M-1/2,0}}{2} \right) \left( \frac{\phi_{M+1,0} - \phi_{M-1,0}}{h} \right) + \left( \frac{\bar{v}_{M,1/2} + \bar{v}_{M,-1/2}}{2} \right) \left( \frac{\phi_{M,1} - \phi_{M,0}}{h} \right) \], \hspace{1cm} (42)

at the northwest corner the new expression for \( P \) is
\[ P_{0,N} := \bar{\phi}_{0,N} - \Phi \eta \Delta t \left( \frac{\bar{u}_{1/2,N} - \bar{u}_{-1/2,N} + \bar{v}_{0,N+1/2} - \bar{v}_{0,N-1/2}}{h} \right) \]
\[ - 2\eta \Delta t \left[ (\phi_{0,N} - \Phi) \left( \frac{u_{1/2,N} - u_{-1/2,N} + v_{0,N+1/2} - v_{0,N-1/2}}{h} \right) \right] \]
\[ + \left( \frac{\bar{u}_{1/2,N} + \bar{u}_{-1/2,N}}{2} \right) \left( \frac{\phi_{1,N} - \phi_{0,N}}{h} \right) + \left( \frac{\bar{v}_{0,N+1/2} + \bar{v}_{0,N-1/2}}{2} \right) \left( \frac{\phi_{0,N} - \phi_{0,N-1}}{h} \right) \], \hspace{1cm} (43)

and (finally!) at the northeast corner the new expression for \( P \) is
\[ P_{M,N} := \bar{\phi}_{M,N} - \Phi \eta \Delta t \left( \frac{\bar{u}_{M+1/2,N} - \bar{u}_{M-1/2,N} + \bar{v}_{M,N+1/2} - \bar{v}_{M,N-1/2}}{h} \right) \]
\[-2\eta \Delta t \left[ (\phi_{M,N} - \Phi) \left( \frac{u_{M+1/2,N} - u_{M-1/2,N} + v_{M,N+1/2} - v_{M,N-1/2}}{h} \right) \right] \]
\[+ \left( \frac{u_{M+1/2,N} + u_{M-1/2,N}}{2} \right) \left( \frac{\phi_{M,N} - \phi_{M-1,N}}{h} \right) \]
\[+ \left( \frac{v_{M,N+1/2} + v_{M,N-1/2}}{2} \right) \left( \frac{\phi_{M,N} - \phi_{M,N-1}}{h} \right) \]. \quad (44)

### 3.3 The discrete Helmholtz problem

To facilitate solving the discrete implicit system (27)–(29) we reduce it to a single equation of the Helmholtz type for \(\tilde{\phi}\) as follows. First, at the interior points (0 < \(i\) < \(M\), 0 < \(j\) < \(N\)) we can substitute for \(\bar{u}\) and \(\bar{v}\) in (29) from (27) and (28) to obtain

\[
(1 + 4\gamma^2)\tilde{\phi}_{i,j} - \gamma^2 \left( \tilde{\phi}_{i+1,j} + \tilde{\phi}_{i-1,j} + \tilde{\phi}_{i,j+1} + \tilde{\phi}_{i,j-1} \right) = g_{i,j}
\]

\[= P_{i,j} - c\gamma \left[ U_{i+1/2,j} - U_{i-1/2,j} + V_{i,j+1/2} - V_{i,j-1/2} \right], \quad (45)\]

where \(\gamma = c\eta \Delta t / h\) is the Courant number. This equation applies at the interior gridpoints only. Note that the same equation is obtained if we first eliminate \(u\) and \(v\) between (8)–(10) and then discretize in space, rather than discretize first and then eliminate as done here[15].

At the boundaries we must take a slightly different approach, since (27) and (28) cannot be used to substitute for the ghost-point values of \(\bar{u}\) and \(\bar{v}\). Rather, we obtain equations involving these ghost-point values from the boundary conditions (4)–(7), which we discretize as

\[
\left( \frac{\tilde{u}_{1/2,j} + \tilde{u}_{-1/2,j}}{2} \right) + \frac{\tilde{\phi}_{0,j}}{c} = U_{-1/2,j} := \left( \frac{u_s + \phi_s}{c} \right)_{0,j}, \quad (46)
\]

\[
\left( \frac{\tilde{u}_{M+1/2,j} + \tilde{u}_{M-1/2,j}}{2} \right) - \frac{\tilde{\phi}_{M,j}}{c} = U_{M+1/2,j} := \left( \frac{v_s + \phi_s}{c} \right)_{M,j}, \quad (47)
\]

\[
\left( \frac{\tilde{v}_{i,1/2} + \tilde{v}_{i,-1/2}}{2} \right) + \frac{\tilde{\phi}_{i,0}}{c} = V_{i,-1/2} := \left( \frac{u_s - \phi_s}{c} \right)_{i,0}, \quad (48)
\]

\[
\left( \frac{\tilde{v}_{i,N+1/2} + \tilde{v}_{i,N-1/2}}{2} \right) - \frac{\tilde{\phi}_{i,N}}{c} = V_{i,N+1/2} := \left( \frac{v_s - \phi_s}{c} \right)_{i,N}. \quad (49)
\]

Solving these for the ghost-point values of \(\tilde{u}\) and \(\tilde{v}\) and using these together with (27) and (28) at the interior and boundary points, we obtain the following discrete equations for \(\tilde{\phi}\):
on the west boundary,
\[
(1 + 2\gamma + 4\gamma^2)\tilde{\phi}_{0,j} - \gamma^2(\tilde{\phi}_{0,j+1} + 2\tilde{\phi}_{1,j} + \tilde{\phi}_{0,j-1}) = g_{0,j}
\]
\[
= P_{0,j} - c\gamma \left(2U_{1/2,j} - 2U_{-1/2,j} + V_{0,j+1/2} - V_{0,j-1/2}\right),
\]
(50)
on the east boundary,
\[
(1 + 2\gamma + 4\gamma^2)\tilde{\phi}_{M,j} - \gamma^2(\tilde{\phi}_{M,j+1} + 2\tilde{\phi}_{M-1,j} + \tilde{\phi}_{M,j-1}) = g_{M,j}
\]
\[
= P_{M,j} - c\gamma \left(2U_{M+1/2,j} - 2U_{M-1/2,j} + V_{M,j+1/2} - V_{M,j-1/2}\right),
\]
(51)
on the south boundary,
\[
(1 + 2\gamma + 4\gamma^2)\tilde{\phi}_{i,0} - \gamma^2(\tilde{\phi}_{i+1,0} + 2\tilde{\phi}_{i,1} + \tilde{\phi}_{i-1,0}) = g_{i,0}
\]
\[
= P_{i,0} - c\gamma \left(U_{i+1/2,0} - U_{i-1/2,0} + 2V_{i,1/2} - 2V_{i,-1/2}\right),
\]
(52)
on the north boundary,
\[
(1 + 2\gamma + 4\gamma^2)\tilde{\phi}_{i,N} - \gamma^2(\tilde{\phi}_{i+1,N} + 2\tilde{\phi}_{i,N-1} + \tilde{\phi}_{i-1,N}) = g_{i,N}
\]
\[
= P_{i,N} - c\gamma \left(U_{i+1/2,N} - U_{i-1/2,N} + 2V_{i,N+1/2} - 2V_{i,N-1/2}\right),
\]
(53)
at the southwest corner,
\[
(1 + 4\gamma + 4\gamma^2)\tilde{\phi}_{0,0} - 2\gamma^2(\tilde{\phi}_{1,0} + \tilde{\phi}_{0,1}) = g_{0,0}
\]
\[
= P_{0,0} - 2c\gamma \left(U_{1/2,0} - U_{-1/2,0} + V_{0,1/2} - V_{0,-1/2}\right),
\]
(54)
at the southeast corner,
\[
(1 + 4\gamma + 4\gamma^2)\tilde{\phi}_{M,0} - 2\gamma^2(\tilde{\phi}_{M-1,0} + \tilde{\phi}_{M,1}) = g_{M,0}
\]
\[
= P_{M,0} - 2c\gamma \left(U_{M+1/2,0} - U_{M-1/2,0} + V_{M,1/2} - V_{M,-1/2}\right),
\]
(55)
at the northwest corner,
\[
(1 + 4\gamma + 4\gamma^2)\tilde{\phi}_{0,N} - 2\gamma^2(\tilde{\phi}_{1,N} + \tilde{\phi}_{0,N-1}) = g_{0,N}
\]
\[
= P_{0,N} - 2c\gamma \left(U_{1/2,N} - U_{-1/2,N} + V_{0,N+1/2} - V_{0,N-1/2}\right),
\]
(56)
and (finally!) at the northeast corner,
\[
(1 + 4\gamma + 4\gamma^2)\tilde{\phi}_{M,N} - 2\gamma^2(\tilde{\phi}_{M-1,N} + \tilde{\phi}_{M,N-1}) = g_{M,N}
\]
\[
= P_{M,N} - 2c\gamma \left(U_{M+1/2,N} - U_{M-1/2,N} + V_{M,N+1/2} - V_{M,N-1/2}\right).
\]
(57)
To write the discrete Helmholtz problem obtained above more compactly we use the idea of “grid functions”, i.e., collections of values on the grid $\Omega_h$ with the appropriate staggering. We denote such grid functions using the superscript $h$. For example, $u^h, v^h, \phi^h$ denote the sets of values $u_{i+1/2,j}, v_{i,j+1/2},$ and $\phi_{i,j}$, respectively, as shown in Fig. 2. With this notation, the discrete Helmholtz problem becomes simply

$$L^h \psi^h = g^h \quad \text{on } \Omega_h,$$

(58)

where $L^h$ is a linear operator expressing the left-hand side of (45) and (50)–(57) and $g^h$ is the grid function consisting of corresponding values of the right-hand side. Since the linear operator $L^h$ is strictly diagonally dominant, this discrete problem has a unique solution $\psi^h$ for any right-hand side $g^h$.

### 3.4 Summary of discretization

Putting the pieces together, a single time step of the model consists of the following steps:

1. Using the current solution $(u^h, v^h, \phi^h)$ and previous solution $(\bar{u}^h, \bar{v}^h, \bar{\phi}^h)$ [see (23)–(25)], evaluate the right-hand side $(U^h, V^h, P^h)$ of the implicit system (27)–(29) using equations (30)–(44).

2. Compute the right-hand side $g^h$ of the Helmholtz problem (58) using equations (45) and (50)–(57).

3. Solve the Helmholtz problem (58) for the new geopotential $\psi^h$.

4. Compute the corresponding velocity components $\bar{u}^h$ and $\bar{v}^h$ from (27) and (28).

5. Replace the tangential component of the flow ($\bar{u}$ or $\bar{v}$) on the boundary by the specified flow $(u_s$ or $v_s$) where there is inflow.
4 Multigrid Method

To solve the Helmholtz equation (58) we use a multigrid method. A brief overview of multigrid methods and their application to the Helmholtz problem which arises in our model follows in this section. Much of this section is based on the review paper of Fulton et al.[11], with applications made to our specific problem. More detailed introductions to multigrid methods are given in [2], [3], and [4].

The basic multigrid idea is to approximate a continuous problem on a set of overlapping uniform grids with varying mesh sizes and then cycle between these grids to produce a solution with optimal efficiency. On each grid, error is reduced using a simple relaxation scheme. Multigrid methods offer both efficiency and generality. Typically a multigrid algorithm solves a problem to the level of truncation error in just 4–10 work units\(^1\). While multigrid methods are well-suited for elliptic problems, such as the Helmholtz problem (58), this same efficiency has been obtained for a wide class of problems (including general elliptic boundary value problems, singular perturbation and nonelliptic problems, optimization problems, and integral equations). Adaptive local mesh refinement can be combined with multigrid processing using the Full Approximation Scheme (FAS), which also permits solving some nonlinear problems with similar efficiencies.

Developing a multigrid algorithm for a particular problem involves choosing a relaxation scheme, a set of grids, grid transfers, and a control algorithm. A brief description of these components and of how they are utilized in solving (58) will be given in the following sections.

4.1 Gauss-Seidel relaxation

The heart of a multigrid method is the relaxation scheme used to smooth the error (i.e., to reduce the high wavenumber error components) on each grid. The relaxation scheme is the most problem-dependent part of a multigrid method, and has the most effect on the overall efficiency. Here we use point Gauss-Seidel relaxation, which can be described as follows. Starting with an approximation \( \hat{\phi}^h \) to the true (discrete) solution \( \phi^h \) of (58), a single iteration (“sweep”) of relaxation consists of updating each value \( \hat{\phi}_{i,j}^h \) in turn so that the discrete equation at the corresponding point is exactly satisfied using the current values at the surrounding points. In practice, the resulting new approximation can written over the old approximation \( \hat{\phi}^h \) immediately, so we do not introduce new notation for it here.

While treating the points in lexicographic order is natural and easy, here we use so-called “red-black” ordering, where the colors refer to those of a checkerboard. A single red-black relaxation sweep consists of first relaxing at all “red” points (where \( i + j \) is even), and then at all “black” points (where \( i + j \) is odd). While this ordering is marginally harder to implement and has no real advantage as a solver on a single grid, it is twice as effective as as smoother in a multigrid method[3].

\(^1\)A work unit is the computational work for one relaxation sweep on the finest grid.
The relaxation equations for (58) are obtained by solving (45) and (50)–(57) for the “central” value:

\[
\hat{\phi}_{i,j}^{h} \leftarrow \frac{1}{1 + 4\gamma^2} \left[ g_{i,j} + \gamma^2 (\hat{\phi}_{i+1,j}^{h} + \hat{\phi}_{i-1,j}^{h} + \hat{\phi}_{i,j-1}^{h} + \hat{\phi}_{i,j+1}^{h}) \right]
\]

(59)

for the interior points,

\[
\hat{\phi}_{0,j}^{h} \leftarrow \frac{1}{1 + 2\gamma + 4\gamma^2} \left[ g_{0,j} + \gamma^2 (\hat{\phi}_{0,j-1}^{h} + 2\hat{\phi}_{1,j}^{h} + \hat{\phi}_{0,j+1}^{h}) \right]
\]

(60)

at the west boundary,

\[
\hat{\phi}_{M,0}^{h} \leftarrow \frac{1}{1 + 2\gamma + 4\gamma^2} \left[ g_{M,j} + \gamma^2 (\hat{\phi}_{M,j-1}^{h} + 2\hat{\phi}_{M-1,j}^{h} + \hat{\phi}_{M,j+1}^{h}) \right]
\]

(61)

at the east boundary,

\[
\hat{\phi}_{i,0}^{h} \leftarrow \frac{1}{1 + 2\gamma + 4\gamma^2} \left[ g_{i,0} + \gamma^2 (\hat{\phi}_{i-1,0}^{h} + 2\hat{\phi}_{i,1}^{h} + \hat{\phi}_{i+1,0}^{h}) \right]
\]

(62)

at the south boundary,

\[
\hat{\phi}_{i,N}^{h} \leftarrow \frac{1}{1 + 2\gamma + 4\gamma^2} \left[ g_{i,N} + \gamma^2 (\hat{\phi}_{i-1,N}^{h} + 2\hat{\phi}_{i,N-1}^{h} + \hat{\phi}_{i+1,N}^{h}) \right]
\]

(63)

at the north boundary,

\[
\hat{\phi}_{0,0}^{h} \leftarrow \frac{1}{1 + 4\gamma + 4\gamma^2} \left[ g_{0,0} + 2\gamma^2 (\hat{\phi}_{1,0}^{h} + \hat{\phi}_{0,1}^{h}) \right]
\]

(64)

at the southwest corner,

\[
\hat{\phi}_{M,0}^{h} \leftarrow \frac{1}{1 + 4\gamma + 4\gamma^2} \left[ g_{M,0} + 2\gamma^2 (\hat{\phi}_{M-1,0}^{h} + \hat{\phi}_{M,1}^{h}) \right]
\]

(65)

at the southeast corner,

\[
\hat{\phi}_{0,N}^{h} \leftarrow \frac{1}{1 + 4\gamma + 4\gamma^2} \left[ g_{0,N} + 2\gamma^2 (\hat{\phi}_{1,N}^{h} + \hat{\phi}_{0,N-1}^{h}) \right]
\]

(66)

at the northwest corner, and (finally!)

\[
\hat{\phi}_{M,N}^{h} \leftarrow \frac{1}{1 + 4\gamma + 4\gamma^2} \left[ g_{M,N} + 2\gamma^2 (\hat{\phi}_{M-1,N}^{h} + \hat{\phi}_{M,N-1}^{h}) \right]
\]

(67)

at the northeast corner.

Relaxation is a local process, and as such is effective at smoothing the error, i.e., reducing the high-wavenumber error components. Indeed, local mode analysis[2] shows that for the relaxation described above the amplitude of each such error component is effectively reduced by at least the “smoothing factor” \( \bar{\mu} = 0.25 \) per sweep (when \( \gamma \) is large; for smaller \( \gamma \), \( \bar{\mu} \) is even smaller).
4.2 A two-grid method

After a few relaxation sweeps, the error is smooth and thus no longer needs the original (fine) grid to properly represent it. Thus, the key idea in multigrid methods is coarse grid correction, i.e., using a coarser grid to solve for the smooth components of the error. This can be illustrated in a simple two-grid method as follows (see Fig. 3). Suppose that a few relaxation sweeps (say $\nu_1$ of them) have been carried out for the problem (58) on the grid $\Omega_h$, yielding an approximate solution $\hat{\phi}^h$. After performing this relaxation, the error is smooth (i.e., the high wavenumber modes have been reduced substantially and further relaxation would reduce the low wavenumber modes very slowly). Now the error $e^h = \phi^h - \hat{\phi}^h$ satisfies the residual equation

$$L^h e^h = r^h$$

where

$$r^h = g^h - L^h \hat{\phi}^h$$

is the residual. Since $e^h$ is smooth, (68) can be approximated on a coarser grid $\Omega_{2h}$ with mesh spacing $2h$ by

$$L^{2h} e^{2h} = I_{2h}^h r^h.$$  \hspace{1cm} (70)

Here $L^{2h}$ is the same discrete operator as $L^h$ except for the grid spacing $2h$, $e^{2h}$ is the grid $\Omega_{2h}$ approximation to $e^h$, and the right hand side is obtained by computing $r^h$ from (69) and transferring it to the coarse grid by some fine-to-coarse transfer operator $I_{2h}^h$ (as detailed in the next section). Assuming that (70) can be solved, the result $e^{2h}$ is an approximation to the error $e^h$ and hence can be transferred back to the fine grid and added to the previous approximate solution $\hat{\phi}^h$ to obtain a new approximation

$$\hat{\phi}^h \leftarrow \hat{\phi}^h + I_{2h}^h e^{2h},$$

where $I_{2h}^h$ represents the coarse-to-fine grid transfer (see next section). Finally, the resulting approximation can be improved by further relaxation (say $\nu_2$ sweeps), and the whole cycle can be repeated as needed to solve the problem iteratively.

![Figure 3: A two-grid method](image-url)
4.3 Grid transfers

The two-grid cycle described above requires two basic grid transfers: a fine-to-coarse transfer of the residual in (70) and a coarse-to-fine transfer of the correction in (71). For the latter we simply choose \( I_{2h} \) as bilinear interpolation. For the former, we choose \( I_{2h} \) as full weighting\footnote{[2]} as follows. First, at the interior points we use the standard full weighting

\[
I_{h}^{2h} = \frac{1}{16} \begin{bmatrix}
1 & 2 & 1 \\
2 & 4 & 2 \\
1 & 2 & 1
\end{bmatrix}.
\]

(72)

Here the stencil notation indicates the weights to be used for the fine-grid residual at the coarse-grid point and its nearest neighbors on the fine grid. Since after a red-black relaxation sweep the residuals is zero at the black points \((i + j)\) odd), here this stencil simplifies to

\[
I_{h}^{2h} = \frac{1}{16} \begin{bmatrix}
1 & 0 & 1 \\
0 & 4 & 0 \\
1 & 0 & 1
\end{bmatrix}.
\]

(73)

On the boundaries and corners the full weighting must be suitably modified. For example, on the west boundary, the stencil is

\[
I_{h}^{2h} = \frac{1}{16} \begin{bmatrix}
0 & 0 & 2 \\
0 & 4 & 0 \\
0 & 0 & 2
\end{bmatrix},
\]

(74)

and the stencil for the southwest corner is

\[
I_{h}^{2h} = \frac{1}{16} \begin{bmatrix}
0 & 0 & 4 \\
0 & 4 & 0 \\
0 & 0 & 0
\end{bmatrix},
\]

(75)

with the other boundaries and corners treated analogously.

4.4 The multigrid V-cycle

The method used to solve the residual problem (70) on the coarse grid in the two-grid method described above was left unspecified. Since this equation has the same form as the original equation (58), it can be solved by the same method; i.e., relax the grid \( \Omega_{2h} \) problem \( \nu_1 \) times, compute a correction to this problem using a still coarser grid \( \Omega_{4h} \), and relax the updated solution \( \nu_2 \) times. Continuing this process recursively through still coarser grids leads to a multigrid V-cycle as shown in Fig. 4. Here the circles denote relaxation sweeps (to reduce the error on the scale of that grid), downward arrows denote the transfer of the residual problem to the next coarser grid [cf. (70)], and upward arrows denote interpolation of the correction back to the finer grid [cf. (71)]. On the coarsest grid, the problem can be solved either directly or by many relaxation sweeps.

16
In a multigrid V-cycle, a total of $\nu = \nu_1 + \nu_2$ relaxation sweeps are performed on each level, and each Fourier component of the error is a high wavenumber component on one of the grids used. Thus (neglecting the effects of grid transfers) the error should be reduced on the scale of that grid by the factor $\tilde{\mu}^\nu$, where $\tilde{\mu}$ is the multigrid smoothing factor. Since this factor is independent of the mesh size $h$, repeated multigrid V-cycles give overall error reduction which is independent of $h$—unlike most iterative methods which converge more slowly as $h$ decreases. Here we use $\nu_1 = \nu_2 = 1$, giving an error reduction of $\tilde{\mu}^\nu = 1/16$, i.e., more than an order of magnitude per V-cycle. In addition, since the number of points on the coarse grids decreases geometrically, the total number of points (and thus the total computational work of a V-cycle) is a proportional to the number of points on the finest grid (here, about $4/3$ the work of the two sweeps on the finest grid).

### 4.5 The Full Multigrid algorithm

Using repeated V-cycles to solve the problem iteratively requires an initial approximation on the finest grid $\Omega_h$. One way to supply that is to first solve the problem on the next coarser grid $\Omega_{2h}$ (somehow), and then interpolate that solution to grid $\Omega_h$ to use as an initial approximation. Extending this idea recursively back through still coarser grids leads to the Full Multigrid (FMG) algorithm[2] as shown in Fig. 5. This figure uses the same notation as in Fig. 4, with double circles indicating the converged solution on a given level and double arrows its initial (FMG) interpolation to the next coarser level. To preserve the smoothness of this converged solution, the initial FMG interpolation must have higher-order accuracy[3], so we use bicubic interpolation here. With one V-cycle per level as shown, the algorithm is known as the 1-FMG algorithm; it should be noted that it is finite, not iterative. It can be shown heuristically that this algorithm solves the problem to the level of the truncation error, i.e., the norm of the final residual is less than the norm of the truncation error, so further iteration is not needed.
4.6 Local mesh refinement

To provide increased resolution in localized areas (in this model, the area around a tropical cyclone vortex), multigrid processing can be combined naturally with local mesh refinement[2]. This is done by superimposing grids of differing mesh size and extent. Here we use the same mesh structure as in the original (nondivergent) MUDBAR model[9], consisting of a base grid covering the entire computational grid with one or more nested grid patches with successively finer mesh sizes as shown in Fig. 6. As the details are largely the same as in the original model, they will not be repeated here. Several points deserve mention, however. First, for simplicity we use only the Berger-Oliger[1] algorithm here; this results in only one-way interaction at the grid interfaces but is somewhat simpler to implement. In particular, it does not require using Full Approximation Scheme (FAS) processing[3] (although that is included in the model code). Second, transfers of velocity components between the computational grids are slightly complicated due to the variable staggering. In all cases we use the simplest possible interpolation (i.e., injection and/or linear interpolation), so the details are straightforward (if tedious).
5 Numerical Results

This section presents numerical results showing that the multigrid solver described in section 4 works as predicted by theory, that the open boundary conditions allow gravity waves to pass through grid interfaces and lateral boundaries as desired, and that local mesh refinement produces significant gains in accuracy and efficiency.

5.1 Multigrid convergence

To verify the convergence of the multigrid solver for the Helmholtz problem (58) described in section 4, we ran it and measured the resulting reduction in the residual using the effective convergence factor per sweep defined as

$$\mu = \left( \frac{\|r_0^h\|}{\|r_0^H\|} \right)^{1/\nu}.$$ (76)
Here, \( r_0^h \) and \( r_\nu^h \) are the residuals (69) before and after the V-cycle with \( \nu \) sweeps on grid \( \Omega_h \), respectively, and \( \| \cdot \| \) denotes \( l_2 \) norm. Note that here we use the dynamic residual (computed during the sweep). The observed value of \( \mu \) should closely approximate the multigrid smoothing factor \( \bar{\mu} \), which is 0.25 in the Poisson limit (large \( \gamma \)).

Figure 7 shows the convergence history (measured by \( \mu \)) as a function of the V-cycle number for four different solutions of (58) using 12 V-cycles each. For each case, \( h = 32 \) km and \( \Phi = 10^4 \) m\(^2\)s\(^{-2}\), and thus \( c = 100 \) ms\(^{-1}\). The cases with time steps \( \Delta t \) of 1, 2, 4, and 8 minutes thus correspond to Courant numbers \( \gamma \) of 0.1875, 0.375, 0.75, and 1.5, respectively. On each line the plateau represents the asymptotic value of \( \mu \) (eventually the convergence degrades when the iteration “bottoms out” due to roundoff error). For the larger Courant numbers this is close to \( \bar{\mu} = 0.25 \) and for smaller Courant numbers it is smaller (as expected), so we conclude that the multigrid solver is working as predicted by the smoothing analysis.

![Figure 7: Convergence history for multigrid Helmholtz solver](image)

**5.2 Gravity wave experiment**

To test the performance of the boundary and interface conditions we use an experiment which includes propagating gravity waves. The initial condition consists of a hurricane-like vortex embedded in a motionless environment, with initial geopotential specified as

\[
\phi(x, y, 0) = \phi_0 + \phi_1 (1 + \epsilon) \exp \left\{ - \left[ \left( \frac{x - x_c}{x_s} \right)^2 + \left( \frac{y - y_c}{y_s} \right)^2 \right] \right\}.
\] (77)
Here $\phi_0$ is the far-field geopotential (we use the value $\phi_0 = 10^4$ m$^2$s$^{-2}$, corresponding to $c = 100$ ms$^{-1}$) and $(x_c, y_c)$, $(x_s, y_s)$, and $\phi_1$ specify the center, horizontal scale, and amplitude of the initial vortex, respectively. The parameter $\epsilon$ is included to allow an initial geostrophic imbalance (to generate gravity waves for testing). The corresponding initial velocity components are obtained from the assumption of geostrophic balance, i.e.,

$$-fu = \frac{\partial \phi}{\partial y}, \quad fu = \frac{\partial \phi}{\partial x}, \quad (78)$$

resulting in

$$u(x, y, 0) = \left[ \frac{2(y - y_c)}{fy_s^2} \right] \phi_1 \exp \left\{ - \left[ \left( \frac{x - x_c}{x_s} \right)^2 + \left( \frac{y - y_c}{y_s} \right)^2 \right] \right\} \quad (79)$$

and

$$v(x, y, 0) = -\left[ \frac{2(x - x_c)}{fx_s^2} \right] \phi_1 \exp \left\{ - \left[ \left( \frac{x - x_c}{x_s} \right)^2 + \left( \frac{y - y_c}{y_s} \right)^2 \right] \right\}. \quad (80)$$

Note that here we have dropped the factor $\epsilon$, so setting $\epsilon$ nonzero in (77) will result in an initially unbalanced vortex which will generate gravity waves. Using the values $\phi_1 = -75$ m$^2$s$^{-2}$ and $x_s = y_s = 112$ km (on a $\beta$-plane centered at latitude $\theta_0 = 20^\circ$ N, where $f = f_0 = 4.99 \times 10^{-5}$ s$^{-1}$ and $\beta = 2.15 \times 10^{-11}$ m$^{-1}$s$^{-1}$) results in a vortex with maximum wind speed about 12 ms$^{-1}$ at radius 80 km. We set $\epsilon = 0.2$ in (77) to generate initial geostrophic imbalance and thus produce propagating gravity waves.

Figure 8 shows the result of solving this problem with and without mesh refinement. The domain is a square of side length 4096 km. The left-hand panels show the solution at times $t = 4, 6$, and 8 hours computed on a uniform grid with mesh spacing $h = 32$ km, while the right-hand panels show the analogous solution computed with one grid patch with mesh spacing $h = 16$ km superimposed (the interface is shown in the Figure). In both cases the gravity wave front is clearly seen in the geopotential field (contours), propagates with the correct speed (approximately $c = 100$ ms$^{-1}$), and exits the domain with little visible reflection. With the mesh refinement, we notice two things: (1) there is no visible reflection or distortion as the wave crosses the grid interface into the coarser grid, and (2) the wave is initially better resolved (due to the fine-grid patch) and remains so even after propagating into the coarse grid. We conclude that the open boundary conditions used at the lateral boundaries and grid interfaces work as expected, and that the model can handle gravity wave propagation properly. Note, however, that these results used Courant number $\gamma = 0.56$ on each grid in order to properly represent gravity wave propagation. Other tests[15] show that with larger time steps ($\gamma > 5$) there is more wave reflection from the boundaries and interfaces as expected due to distortion of the gravity wave speed by the time discretization. This should not be an issue: while large time steps ($\gamma > 1$) are stable with semi-implicit time discretization, they should only be used in situations where there is little energy in the gravity waves (which would be distorted by the time stepping regardless of the boundary or interface treatment).
Figure 8: Gravity wave experiment, showing geopotential (contours) and velocity (vectors) as functions of $x$ and $y$ for a uniform-grid case (left) and local mesh refinement case (right).
5.3 Sample run

To investigate the performance of the model in a more realistic situation, we embed the vortex specified by (77), (79), and (80) (using $\epsilon = 0$ to achieve balanced initial flow) in an environmental flow. For the latter we use the zonal current of DeMaria[6], specified by

$$\bar{u}(y) = \bar{u}_0 \sin \left( \frac{2\pi y}{l} \right), \quad \bar{v} = 0,$$

where $\bar{u}_0$ specifies the amplitude and $l$ specifies the length scale. From (78) we compute the corresponding environmental geopotential in geostrophic balance as

$$\bar{\phi}(y) = \phi_0 - \frac{\bar{u}_0 l}{2\pi} \left[ \frac{\beta}{2\pi} \sin \left( \frac{2\pi y}{l} \right) - (f_0 + \beta y) \cos \left( \frac{2\pi y}{l} \right) \right],$$

where the $\phi_0$ is the reference geopotential. Note that this environmental flow is added to the initial vortex at $t = 0$, and used at each time step (without the vortex) to supply specified values of $u$, $v$, and $\phi$ as required for the lateral boundary conditions.

For the sample run below (and subsequent runs) the domain is a square with side length 6144 km on a $\beta$-plane centered at at latitude $20^\circ$ N. The environmental flow uses the values $\bar{u}_0 = 10$ ms$^{-1}$ and $l = 6144$ km, and the vortex uses the values $\phi_1 = -150$ m$^2$s$^{-2}$ and $x_s = y_s = 96$ km, resulting in a maximum wind speed of about 27 ms$^{-1}$ at radius 68 km.

Figure 9 shows the results of a sample 72 hour model run. This run used a base grid with mesh spacing $h = 32$ km and two patches with mesh spacing $h = 16$ km and $h = 8$ km with side length 1/2 and 1/4 of the domain size, respectively. The time step on the base grid was $\Delta t = 2$ minutes, and output is shown at $t = 0, 12, 24, 36, 48,$ and 60 hours. Since the initial conditions are only in geostrophic balance, a small gravity wave front is generated (visible at $t = 12$ hours) which leaves the domain without noticeable reflection. For the remainder of the run, the solution evolves smoothly, with no evidence of problems at the grid interfaces.

5.4 Accuracy vs. efficiency

To evaluate the accuracy and efficiency of the model with and without local mesh refinement, we compare the forecast track for various runs to that from a higher-resolution reference run. The vortex center is taken to be the location of the maximum relative vorticity

$$\zeta = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

and the track differences were averaged in time over a 72 hour model run to obtain the “mean forecast error” for the run. The domain and initial conditions are the same as for the sample run above. The reference run used a uniform grid with mesh size $h = 8$ km and the time step $\Delta t = 1$ minute.
Figure 9: Sample run, showing geopotential (contours) and velocity (vectors) as functions of $x$ and $y$, with boundaries of grid patches as indicated.
Figure 10 shows the errors in several model runs plotted as a function of the computer time needed for the run (on a SUN Ultra10 workstation). The points plotted with + signs are for uniform-grid runs with the indicated mesh size; the line joining them thus gives a benchmark against which to compare the runs with local refinement. It should be noted that the errors for the uniform-grid runs decrease like $O(h^2)$ as expected. The points plotted with $\times$ signs are for runs using local mesh refinement; they are labeled with the size of the grid patch(es) used (side lengths of $A = 1/2$, $B = 3/8$, and $C = 1/4$ of the domain size). The runs with a single patch produce essentially the same error as the $h = 16$ km run (since they have the same finest mesh size) but at a savings of up to a factor of ten in computational cost. Likewise, the runs with two patches produce errors which are smaller (by about a factor of ten) with roughly the same computational cost.

Figure 10: Plot of Accuracy Vs. Efficiency
6 Conclusions

We have successfully extended the adaptive multigrid hurricane model MUDBAR from a simple non-divergent, barotropic model to the next level of dynamical complexity, the shallow water equations. We find no significant barriers to applying the adaptive multigrid method in this setting; rather, it works essentially the same as in the simpler model. In particular:

- The multigrid solver exhibits convergence rates which are consistent with theory.
- The use of open boundary conditions allows gravity waves to pass through the lateral boundaries and the interfaces between grids without noticeable reflection.
- The use of local mesh refinement can produce up to an order of magnitude improvement in either accuracy or efficiency (compared to using a single uniform grid).

The model developed here may be useful in studying problems of hurricane motion and dynamics. The success of the adaptive multigrid method for this problem suggests that it may be useful for other fluid flow problems.
References


