8-1-2004

A Block Structured Adaptive Solution to the Shallow Water Equations

Nitin Bhagat

A BLOCK STRUCTURED ADAPTIVE SOLUTION TO THE SHALLOW WATER EQUATIONS

By

Nitin Bhagat

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computational Engineering
in the Department of Computational Engineering

Mississippi State, Mississippi

August 2004

Name: Nitin Bhagat

Date of Degree: August 07, 2004

Institution: Mississippi State University

Major Field: Computational Engineering

Major Professor: Dr. Roger L. King

Director of Thesis : Dr. Matthew Bettencourt

Title of Study: A BLOCK STRUCTURED ADAPTIVE SOLUTION TO THE SHALLOW WATER EQUATIONS

Pages in Study: 46

Candidate for Degree of Master of Science

An adaptive mesh refinement algorithm for shallow water equations is presented. The algorithm uses upwind scheme that is Godunov type and which approximately solves the Riemann problem using Roe's technique. A highly accurate solution is achieved by using the adaptive mesh refinement technique of Berger and Oliger for mesh refinement algorithm. The numerical method is second-order accurate and approximately max-min preserving by using van Leer limited-slope technique. One-dimensional nesting algorithm has been implemented successfully. Numerical results on a test problem verify the second order accuracy of the algorithm. The nested grid results yield the equivalent solution to that of the corresponding fine grid solution.

DEDICATION

To my family.

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to my thesis director Dr. Matthew Bettencourt for his invaluable advice and assistance during my studies and during my work on this thesis. He has been a good mentor and extremely patient in explaining the concepts. His constant encouraging words and ideas have inspired me to complete this project.

I would also like to thank Dr. Roger King for providing me with valuable support as an advisor, without his help, this project would have never been completed. I would also like to thank Dr. Clarence Burg for his suggestions and technical advice.

In addition I would like to thank Dr. David Shaw and Dr. Mark Janus and the Engineering Research Center for providing financial support and facilities to this research effort.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS, AND NOMENCLATURE

$\sim$ approximately

$\vec{}$ vector

O Order of

$_o$ reference value

$'$ perturbed value

SWE Shallow Water Equations

AMR Adaptive Mesh Refinement

ADCIRC Advanced Circulation Model for Coastal Ocean Hydrodynamics

SHORECIRC Quasi-3D Near-shore Circulation Model

POM Princeton Ocean Model

POP Parallel Ocean Program

NCOM Navy Coastal Ocean Model

MICOM Miami Isopycnal Coordinate Model

Λ eigenvalues

Ω angular velocity

$\pounds$ norm

α coefficient of thermal expansion

β coefficient of saline concentration

η perturbation height

κ diffusion coefficient

ν**,** μ coefficient of kinematic, dynamic viscosity

ϕ latitude of earth

ρ density

τ normal, shear stress

A, B Jacobian matrices

I identity matrix

L, R left, right eigenvectors

W characteristic variables

L length scale

Q heat gained per unit mass

P**,** R prolongation, restriction operator

S salinity

T temperature, time scale

U horizontal velocity scale

V specific volume

W vertical velocity scale

e internal energy

f**,**$f*$ Coriolis parameter, inverse Coriolis parameter

g density

h total depth

p pressure

r radius of earth's curvature

u**,** v**,** w velocity components

CHAPTER I

INTRODUCTION

1.1 Shallow Water Flows

Many geophysical flows are nearly horizontal, since the aspect ratios of the oceans and the atmosphere are small. The average depth of the ocean is $\sim$4km and therefore the vertical scale of the motions is O(1km). For large scale motions, the horizontal scale is O(1000km). Therefore, the ratio of the vertical scale to the horizontal scale is O(10^3)[15].

Hence, for many practical applications, these flows can be assumed to be two-dimensional. This assumption allows considerable simplification in the mathematical formulation and numerical solution

However, they are not exactly two-dimensional. The flow exhibits a three-dimensional structure due to bottom friction. Moreover, density stratification due to differences in temperature or salinity causes variation in the third(vertical) direction. Yet, in many geophysical flows these three-dimensional effects are not essential and it is sufficient to consider the depth-averaged form, which is two-dimensional form in the horizontal plane. This restricted form is commonly indicated by the term "shallow water equations" (SWE).

It could be argued that it is no longer necessary to make this kind of approximations in the present era of high-speed computations, where fully three-dimensional and time

dependent flows can be simulated. However, for many practical applications that is just too much detail and using the two-dimensional depth averaged shallow water equations gives essentially the same information at much lower cost.

1.2 Importance of Shallow Water Equations

The shallow water equations are still very important in modern geophysical modeling. First, they provide an efficient solution to many important processes at a much reduced cost over a fully three-dimensional model. Models such as Advanced Circulation Model for Coastal Ocean Hydrodynamics (ADCIRC)[16] or Quasi-3D Near-shore Circulation Model (SHORECIRC)[14] are either fully two-dimensional models or contain a column based third dimension approximation (2.5 dimensional). These models have been used to predict important processes such as storm surge prediction, coastal flooding and oil spill prediction. These problems exhibit a time critical nature where the solution from a fully three-dimensional model may be too late.

The shallow water equations also play an important role in most three dimensional ocean models. The bulk of the ocean models, Princeton Ocean Model (POM) [18], Parallel Ocean Program (POP) [2], Navy Coastal Ocean Model (NCOM) [17], all utilize the hydrostatic approximation. This approximation assumes that the pressure in the fluid is only a function of depth ($p = \rho g h$). This assumption greatly simplifies the solution of three-dimensional Navier-Stokes equations. Instead of solving for velocity with the constraint $\nabla \cdot \vec{u} = 0$ which determines the value of pressure, pressure field is updated explicitly in

terms of the sea surface height, neglecting the velocity divergent part. This turns a global problem into a local problem. Thus, the problem is broken into two parts, the external mode, or depth averaged, and the internal mode. The external mode is then solved using the shallow water equations and this solution is used to drive the internal mode. Typically the external mode has a much lower time step than the internal mode and thus sub-cycling is applied as in POM[18] or the semi-implicit approach of Casullie and Chang[6] is used in NCOM[17]. Both of these approaches require significant computational resources and would benefit from more efficient solvers.

Another related area where the shallow water equations play an important role is in the development of isopycnal ocean models. The Miami Isopycnal Coordinate Model (MICOM)[1] is one such layered isopycnal model. In isopycnal models the ocean is divided into layers of constant density. In a layer, flow properties are assumed to be constant and the solution is advanced in time. This results in solving equations similar to the shallow water equations for each layer. Techniques useful for the shallow water equations may also be applicable to many-layered models.

1.3 Need for Adaptive Mesh Refinement

Shallow water flows present multiple spatial scales. From the point of view of numerical representation, there are locations where high resolution is needed, while coarser resolution may be sufficient in other areas. However, such structures evolve in time and space.

For example, shipping channels in Gulf of Mexico change fresh water profile of the gulf and thus the full flow. Warm water entrainment along the keys change the flow patterns along the Atlantic coastal region. Poor resolution along the shelf breaks can cause stability problems with hydrostatic models, since they require smearing of shelf breaks. High resolution resolves this on the physical scale.

Because of the computational cost it will be prohibitive to cover the whole ocean with this fine mesh size. One way of overcoming this difficulty is to use an adaptive resolution which can reduce the computational cost while conserving the advantages of high resolution.

Nested model grids may be adaptive and movable or static. In order to follow evolving oceanic features, such as , hurricane tracking, wave fronts and propagating eddies, it is beneficial to apply the adaptive mesh refinement(AMR) techniques. The mesh refinement algorithm used in the present work uses the hierarchical structured grid approach of Berger and Oliger[5]. This AMR framework uses classical algorithms on regular Cartesian grids of different resolutions arranged hierarchically.

In the literature, both one-way and two-way coupling between the course grid and the fine grid have been considered. In one-way nesting, information is interpolated from the course grid to the fine grid, but there is no feedback from the fine grid. Fox and Maskell[10] compared one-way and two-way nesting and concluded that two-way interaction gives a more correct solution. Also, the two-way coupling is required for the geometry resolution to be felt by the coarse grid and therefore is more favorable.

In order to provide boundary conditions for the fine grid, the course grid must be interpolated to the fine grid. Studies show that lower order interpolation creates large phase errors whereas higher order interpolation may create overshooting. Quadratic interpolation is used in the present approach. These and other issues have been addressed in detail later in the appropriate section of the report.

1.4 Organization of the Report

With this objective, the report is organized as follows. In Chapter 2 two-dimensional shallow water equations are derived from Navier-Stokes covering the various approximations that are made. These equations are then further analyzed leading to the development of the eigensystem of the equations in Chapter 3. In the same chapter the numerical method for one-dimensional equations are described and the algorithm for numerical solution is discussed. Chapter 4 analyzes various aspects of adaptive mesh refinement technique. The algorithm for numerical implementation in one-dimensional form is also discussed in the same chapter. Numerical results are presented and discussed in Chapter 5. The report summarizes with the Chapter 6 on conclusion and the future work.

CHAPTER II

GOVERNING EQUATIONS

2.1 Shallow Water Equations Derived

In this section, Governing equations for shallow water flows is developed from Navier-stokes equations. The derivation has been discussed in depth by Beniot Cushman-Roisin[9].

2.1.1 Governing Equations

Assumption: *If L is the length scale and r is the radius of curvature of the earth, $L \ll r$.*

Hence curvature terms can be ignored.

2.1.1.1 Momentum Equations

Statement: *Mass times acceleration equals sum of forces.*

$$\text{x}: \quad \rho\left(\tfrac{du}{dt} + f_* w - f v\right) = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}(\tau_{xx}) + \frac{\partial}{\partial y}(\tau_{xy}) + \frac{\partial}{\partial z}(\tau_{xz}) \tag{2.1}$$

$$\text{y}: \quad \rho\left(\tfrac{dv}{dt} + f u\right) = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x}(\tau_{yx}) + \frac{\partial}{\partial y}(\tau_{yy}) + \frac{\partial}{\partial z}(\tau_{yz}) \tag{2.2}$$

$$\text{z}: \quad \rho\left(\tfrac{dw}{dt} - f_* u\right) = -\frac{\partial p}{\partial z} - g\rho + \frac{\partial}{\partial x}(\tau_{zx}) + \frac{\partial}{\partial y}(\tau_{zy}) + \frac{\partial}{\partial z}(\tau_{zz}) \tag{2.3}$$

where

ρ	density
u, v, w	velocity components in x, y, z directions respectively
f	$= 2\Omega \sin\phi$ Coriolis parameter
f_*	$= 2\Omega \cos\phi$ reciprocal Coriolis parameter
	Ω angular velocity, ϕ latitude
p	pressure
τ	normal and shear stresses due to friction
g	gravitational acceleration
	effective gravitational force = true gravitational + centrifugal force
$\frac{d}{dt}$	$= \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z}$ total/substantial derivative

2.1.1.2 Mass Conservation

Statement: *Mass is conserved.*

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0 \tag{2.4}$$

2.1.1.3 Energy Conservation

Statement: *Internal energy gained is equal to the heat received minus the mechanical work performed.*

$$\frac{de}{dt} = Q - p\frac{dV}{dt}$$

where

e = $C_v T$ internal energy

C_v heat capacity at constant volume

V = $\frac{1}{\rho}$ specific volume

Q heat gained per unit mass

T temperature

Assumption: *No internal sources.* Thus, heat gained is due to lateral diffusion. Using Fourier law:

$$\rho\, Q = k\nabla^2 T$$

And using the mass conservation Eqn.2.4, Energy equation becomes

$$\rho C_v \frac{dT}{dt} + p\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) = k\nabla^2 T \tag{2.5}$$

2.1.1.4 Equation of State

- For air:
  Assumption: *Air as an ideal gas*
  $$\rho = \frac{p}{RT} \tag{2.6}$$
  where
  R = $C_p - C_v$
  C_p heat capacity at constant pressure

- For water:
  Assumption: *Water as an incompressible fluid*, hence density is independent of pressure. Assuming density to be *linearly dependent* of temperature and salinity:
  $$\rho = \rho_o\left[1 - \alpha(T - T_o) + \beta(S - S_o)\right] \tag{2.7}$$
  where

S	salinity
α	coefficient of thermal expansion
β	coefficient of saline concentration
ρ_o, T_o, S_o	reference values of density, temperature and salinity

- Local Salt Budget:
  Statement: *Sea water parcels conserve their salt content except in the face of diffusion.*

$$\frac{dS}{dt} = \kappa_s \nabla^2 S \tag{2.8}$$

  where
  κ_s coefficient of salt diffusion

2.1.2 Boussinesq Approximation

Statement: *Fluid density varies, but not greatly, around a mean value.* This variation is caused by the existing stratification and/or fluid motions.

$$\rho = \rho_o + \rho'(x, y, z, t), \qquad \rho' \ll \rho_o \tag{2.9}$$

Argument: In ocean, the variation in density within one ocean basin rarely exceeds 3 kg/m^3. Even in eusteries, where fresh river water eventually turns into salty sea waters, the relative density difference is less than 2%. In atmosphere, most of the density variations can be attributed to isostatic pressure effects, leaving only a moderate variability due to buoyancy effects. Weather patterns are confined to lowest layer(troposphere), approximately 10 km thick, within which density variations responsible for winds are no more than 5%.

2.1.2.1 Continuity Equation

Using the assumption, continuity equation can be written as:

$$\rho_o\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)+\rho'\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right) + \left(\frac{\partial \rho'}{\partial t}+u\frac{\partial \rho'}{\partial x}+v\frac{\partial \rho'}{\partial y}+w\frac{\partial \rho'}{\partial z}\right)=0$$

For geophysical flows, relative variation of density in time and space are usually smaller than that of velocity. Thus only the first group of terms is retained. Thus:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{2.10}$$

2.1.2.2 Momentum Equation

Assumption: *Newtonian fluid* (viscous stresses are proportional to velocity gradients) and with the use of reduced continuity equation Eqn.(2.10), stress components can be written as:

$$\tau_{xx}=\mu\left(\frac{\partial u}{\partial x}+\frac{\partial u}{\partial x}\right)\quad \tau_{xy}=\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\quad \tau_{xz}=\mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)$$
$$\tau_{yy}=\mu\left(\frac{\partial v}{\partial y}+\frac{\partial v}{\partial y}\right)\quad \tau_{yz}=\mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)$$
$$\tau_{zz}=\mu\left(\frac{\partial w}{\partial z}+\frac{\partial w}{\partial z}\right)$$

where μ is the coefficient of dynamic viscosity.

Consider momentum in $x-$direction. Using the Boussinesq approximation, Eqn(2.9), wherever ρ' occurs, ρ_o dominates. Thus Eqn(2.1) becomes:

$$\frac{du}{dt}+f_*w-fv=-\frac{1}{\rho_o}\frac{\partial p}{\partial x}+\nu\nabla^2 u$$

Similarly, the $y-$momentum equation Eqn(2.2) becomes:

$$\frac{dv}{dt} + fu = -\frac{1}{\rho_o}\frac{\partial p}{\partial y} + \nu\nabla^2 v$$

In the $z-$momentum equation, the term ρg accounts for the weight of the fluid that causes increase(decrease) of pressure with depth(height). With the ρ_o part of the density goes a hydrostatic pressure p_o, a function only of z:

$$p = p_o + p'(x, y, z, t)p_o(z) = p_o - \rho_o g z$$

so that

$$\frac{dp_o}{dz} = -\rho_o g$$

And thus the vertical momentum equation reduces to:

$$\frac{dw}{dt} - f_* u = -\frac{1}{\rho_o}\frac{\partial p'}{\partial z} - \frac{\rho' g}{\rho_o} + \nu\nabla^2 w$$

2.1.2.3 Energy Equation

Continuity of volume eliminates the middle term in the energy equation and with the use of Boussinesq approximation, the energy equation becomes:

$$\rho C_v \frac{dT}{dt} = k\nabla^2 T \quad \text{or}$$

$$\frac{dT}{dt} = \kappa_T \nabla^2 T \tag{2.11}$$

where κ_T is coefficient of heat diffusivity. This is isomorphic to the salt equation, Eqn(2.8).

For sea-water, the two equations, Eqn(2.11) for temperature and Eqn(2.8) for salinity, combine to determine the evolution of density. Further simplification can be obtained if the salt and heat diffusivities (κ_s and κ_t) can be taken to be equal. Thus:

$$\frac{d\rho'}{dt} = \kappa \nabla^2 \rho' \tag{2.12}$$

where $\kappa = \kappa_S = \kappa_T$.

2.1.3 Scaling Analysis

Statement: *Geophysical flows are naturally occurring, large scale flows attributed by the presence of effects of ambient rotation and stratification.*

Fluid can be anticipated to feel the effects of ambient rotation if:

$$\frac{\text{time of one revolution}}{\text{motion time scale}} = \frac{2\pi/\Omega}{T} \lesssim 1 \quad \text{or} \quad T \gtrsim \frac{1}{\Omega}$$

and for velocity and length scales:

$$\frac{U}{L} \lesssim \Omega$$

Statement: *Geophysical flows are typically confined to domains that are much wider than they are thick.*

Thus,

$$H \ll L$$

2.1.3.1 Continuity Equation

$$\frac{\frac{U}{L}}{\frac{\partial u}{\partial x}} + \frac{\frac{U}{L}}{\frac{\partial v}{\partial y}} + \frac{\frac{W}{H}}{\frac{\partial w}{\partial z}} = 0 \tag{2.13}$$

Three cases:

1. $W/H \gg U/L$: Eqn(2.13) reduces to $\frac{\partial w}{\partial z} = 0$ or $W =$const. Since there is a bottom somewhere, flow must be supplied by lateral convergence, this is not possible.
2. $W/H \ll U/L$: Eqn(2.13) reduces to $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$. Acceptable, convergence in one horizontal direction is balanced by divergence in the other horizontal direction.
3. $W/H \sim U/L$: implying three way balance is also possible.

Thus, vertical velocity is constrained by:

$$W \lesssim \frac{H}{L} U$$

and since $H \ll L$

$$W \ll U$$

2.1.3.2 Momentum Equation

Order of magnitudes for the x-momentum equation can be written as:

$$\begin{array}{cccccccccc} \frac{\partial u}{\partial t} & +u\frac{\partial u}{\partial x} & +v\frac{\partial u}{\partial y} & +w\frac{\partial u}{\partial z} & +f_* w & -fv & = -\frac{1}{\rho_o}\frac{\partial p}{\partial x} & +\nu\frac{\partial^2 u}{\partial x^2} & +\nu\frac{\partial^2 u}{\partial y^2} & +\nu\frac{\partial^2 u}{\partial z^2} \\ \frac{U}{T} & \frac{U^2}{L} & \frac{U^2}{L} & \frac{WU}{H} & \Omega W & \Omega U & \frac{p}{\rho_o L} & \frac{\nu U}{L^2} & \frac{\nu U}{L^2} & \frac{\nu U}{H^2} \end{array}$$

As compared to the term ΩU, ΩW, $\frac{\nu U}{L^2}$ and $\frac{\nu U}{L^2}$ are small, and hence can be neglected.

Thus, x-momentum equation becomes:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} - fv = -\frac{1}{\rho_o}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u}{\partial z^2} \tag{2.14}$$

Similarly, y-momentum equation reduces to:

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} + fu = -\frac{1}{\rho_o}\frac{\partial p}{\partial y} + \nu\frac{\partial^2 v}{\partial z^2} \tag{2.15}$$

For z-momentum equation:

$$\frac{\partial w}{\partial t} \quad +u\frac{\partial w}{\partial x} \quad +v\frac{\partial w}{\partial y} \quad +w\frac{\partial w}{\partial z} \quad +f_* u \quad = -\frac{1}{\rho_o}\frac{\partial p}{\partial z} \quad -g\frac{\rho}{\rho_o} \quad +\nu\frac{\partial^2 w}{\partial x^2} \quad +\nu\frac{\partial^2 w}{\partial y^2} \quad +\nu\frac{\partial^2 w}{\partial z^2}$$

$$\frac{W}{T} \quad \frac{UW}{L} \quad \frac{UW}{L} \quad \frac{W^2}{H} \quad \Omega U \quad \frac{p}{\rho_o H} \quad \frac{g\Delta\rho}{\rho_o} \quad \frac{\nu W}{L^2} \quad \frac{\nu W}{L^2} \quad \frac{\nu W}{H^2}$$

The first four terms and the friction terms are small compared to the ΩU which itself is found to be small. In summary, the vertical momentum equation reduces to the simple hydrostatic relation as shown:

$$0 = -\frac{1}{\rho_o}\frac{\partial p}{\partial z} - \frac{g\rho}{\rho_o} \tag{2.16}$$

However, pressure p is already a small perturbation to a much larger pressure, and is in hydrostatic balance. Thus, large scale geophysical flows tend to be fully hydrostatic even in the presence of substantial motions.

Thus $z-$momentum equation gives:

$$dp = -g\rho dz \quad \text{or} \tag{2.17}$$

$$p = -g\rho(\eta - z) \tag{2.18}$$

which is hydrostatic relation. Thus:

$$\left.\begin{aligned} \frac{\partial p}{\partial x} &= g\rho\frac{\partial \eta}{\partial x} \\ \frac{\partial p}{\partial y} &= g\rho\frac{\partial \eta}{\partial y} \end{aligned}\right\} \text{ independent of } z \tag{2.19}$$

Statement: *If the horizontal flow field is initially independent of depth, it will remain so at all future times.*

Therefore, horizontal momentum equations become:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv = -\frac{1}{\rho_o}\frac{\partial p}{\partial x} \tag{2.20a}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu = -\frac{1}{\rho_o}\frac{\partial p}{\partial y} \tag{2.20b}$$

Whereas in the continuity equation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.21}$$

first two terms are independent of z but do not necessarily add up to zero. A vertical velocity varying with depth can exist, enabling the flow to support two-dimensional divergence.

An integration over the entire fluid depth yields:

$$\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \int_b^{b+h} dz + [w]_b^{b+h} = 0 \tag{2.22}$$

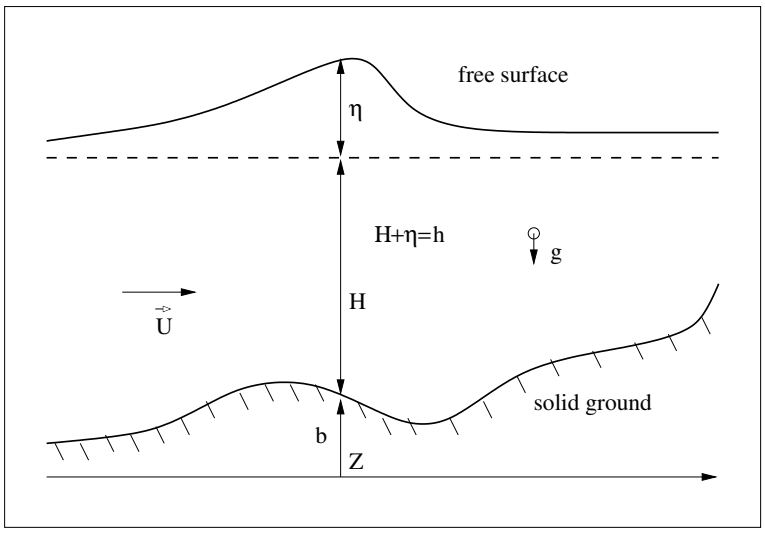


Figure 2.1 Shallow water flow

Applying the boundary conditions that fluid particles cannot leave the surface and bottom results in the continuity equation as shown:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(uh) + \frac{\partial}{\partial y}(vh) = 0 \tag{2.23}$$

For a homogeneous fluid the dynamic pressure, which is independent of depth, is:

$$p = \rho_o g(h + b)$$

Thus, for a flat bottom the system of equations becomes:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv = -g\frac{\partial h}{\partial x} \tag{2.24}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu = -g\frac{\partial h}{\partial y} \tag{2.25}$$

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(uh) + \frac{\partial}{\partial y}(vh) = 0 \tag{2.26}$$

This system of equations is called *shallow water equations.*

CHAPTER III

NUMERICAL IMPLEMENTATION

The conservative form of two-dimensional shallow water equations is given by:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{G}}{\partial x} + \frac{\partial \mathbf{H}}{\partial y} + \mathbf{S} = 0 \tag{3.1}$$

The vectors **U, G** and **H** can be expressed in terms of the primary variables, u, v and h as:

$$\mathbf{U} = \begin{bmatrix} h \\ uh \\ vh \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} uh \\ u^2h + gh^2/2 \\ uvh \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} vh \\ vuh \\ v^2h + gh^2/2 \end{bmatrix} \tag{3.2}$$

and **S** is the source term represented as:

$$\mathbf{S} = \begin{bmatrix} 0 \\ -fv \\ fu \end{bmatrix}$$

The system is hyperbolic in nature with three real and distinct eigenvalues as shown in the following section.

3.1 Eigensystem

To implement the numerical scheme based on the above formulation, eigensystem is derived as shown: Equation (3.1) can be written in the quasi-linear form:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} + \mathbf{S} = 0 \tag{3.3}$$

where

$$\mathbf{A} = \frac{\partial \mathbf{G}}{\partial \mathbf{U}} \quad \text{and} \quad \mathbf{B} = \frac{\partial \mathbf{H}}{\partial \mathbf{U}} \tag{3.4}$$

Thus, the Jacobian matrix **A** and **B** can be computed as:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + gh & 2u & 0 \\ -uv & v & u \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ -v^2 + gh & 0 & 2v \end{bmatrix} \tag{3.5}$$

The eigenvalues of **A, B**, defined as:

$$|\mathbf{A} - \lambda \mathbf{I}| = 0 \qquad |\mathbf{B} - \lambda \mathbf{I}| = 0$$

are given by

$$\Lambda_{\mathbf{A}} = \begin{matrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{matrix} = \begin{matrix} v \\ u + \sqrt{gh} \\ u - \sqrt{gh} \end{matrix} \qquad \Lambda_{\mathbf{B}} = \begin{matrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{matrix} = \begin{matrix} u \\ v + \sqrt{gh} \\ v - \sqrt{gh} \end{matrix}$$

Since the eigenvalues of the each matrix are real and distinct, there exists a linearly independent set of eigenvectors. The right eigenvectors, **R**, for the matrix **A** and **B** are defined as:

$$(\mathbf{A} - \lambda \mathbf{I})\, r = 0 \quad (\mathbf{B} - \lambda \mathbf{I})\, r = 0$$

There exists then, one eigenvector for each eigenvalue. A set of right eigenvectors of each matrix is given as the columns of the matrix:

$$\mathbf{R_A} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & u+\sqrt{gh} & u-\sqrt{gh} \\ 1 & v & v \end{bmatrix} \quad \mathbf{R_B} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & u & u \\ 0 & v+\sqrt{gh} & v-\sqrt{gh} \end{bmatrix}$$

The inverse of these matrices is:

$$\mathbf{R_A^{-1}} = \frac{1}{2\sqrt{gh}} \begin{bmatrix} 2v\sqrt{gh} & 0 & -2\sqrt{gh} \\ u-\sqrt{gh} & -1 & 0 \\ -u-\sqrt{gh} & 1 & 0 \end{bmatrix} \tag{3.6}$$

$$\mathbf{R_B^{-1}} = \frac{1}{2\sqrt{gh}} \begin{bmatrix} 2u\sqrt{gh} & -2\sqrt{gh} & 0 \\ -v+\sqrt{gh} & 0 & 1 \\ v+\sqrt{gh} & 0 & -1 \end{bmatrix} \tag{3.7}$$

Noting that, this is also a set of left eigenvectors of represented as rows of the above matrix $\mathbf{R^{-1}}$ i.e.

$$\mathbf{L} = \mathbf{R^{-1}}$$

Thus, the matrix **A** and **B** has now been diagonalized and is given by:

$$\mathbf{A} = \mathbf{R_A \Lambda_A L_A} \quad \mathbf{B} = \mathbf{R_B \Lambda_B L_A}$$

where the matrix $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues given by Eqn(3.1) as the diagonal elements. Thus, the eigensystem of **A** is given by the eigenvalues of Eqn(3.1), the right eigenvectors which are columns of the similarity matrix **R** and the left eigenvectors which are the rows of the matrix **L**.

3.2 Numerical solution to Shallow water equations

Shallow water equations, as seen from above analysis have hyperbolic character. Hyperbolic equations admit discontinuous and smooth solutions. Even for the case in which the initial conditions are smooth, the non-linear character combined with the hyperbolic type of equations can lead to discontinuous solutions in finite time. The non-linear character of the shallow water equations means that analytical solutions to these equations are limited to only very special cases. Numerical methods are generally used to obtain solutions to practical problems.

Local initial value problems which involve discontinuous neighboring states are known as the *Riemann* problems. Numerical schemes based on the solution are generally known as Godunov-type schemes[11]. This approach solves a series of problems where each Riemann problem corresponds to a cell face. Following the Godunov's approach, Roe[19] developed a computationally efficient method for solving the Riemann problems. The main advantages are that they are robust and accurately capture the location of discontinuities such as shocks and contact surfaces.

3.3 Discretization Methods

In Ocean models, the governing equations are solved principally in three ways: finite-difference, spectral or spectral transform and finite-element methods[15]. The zonal and meridional continuity of the atmospheric envelope makes it inherently well adapted to spectral and spectral transform methods. Hence these methods are preferred in the atmo-

sphere. The availability of fast transforms makes this technique feasible, since transformation from spectral to physical space and back is necessary for data insertion and calculation of nonlinear terms. On the other hand, because of the meridional boundaries, and their complicated shape, the ocean basins are inherently ill suited to spectral techniques. Finite difference methods are more prevalent here. Finite-element methods originated in the coastal engineering community are becoming increasingly popular because of their nearly unlimited flexibility in adapting the grid locally to any desired resolution. However, the finite-difference technique is the current commonly used method for basin scale and global ocean modeling and this method is implemented in the present work.

3.4 Single grid algorithm

The notations used in discretization scheme are as shown in the Fig. 3.1. For one-dimensional case, the governing equations for shallow water flows can be written as:

$$h_t + (uh)_x = 0 \tag{3.8}$$

And the momentum equation is:

$$\begin{aligned} u_t + uu_x &= -gh_x \\ &= -gH_x - g\eta_x \end{aligned}$$

Here

$$H = \eta - h; \quad \text{where } h \text{ is relative depth}$$

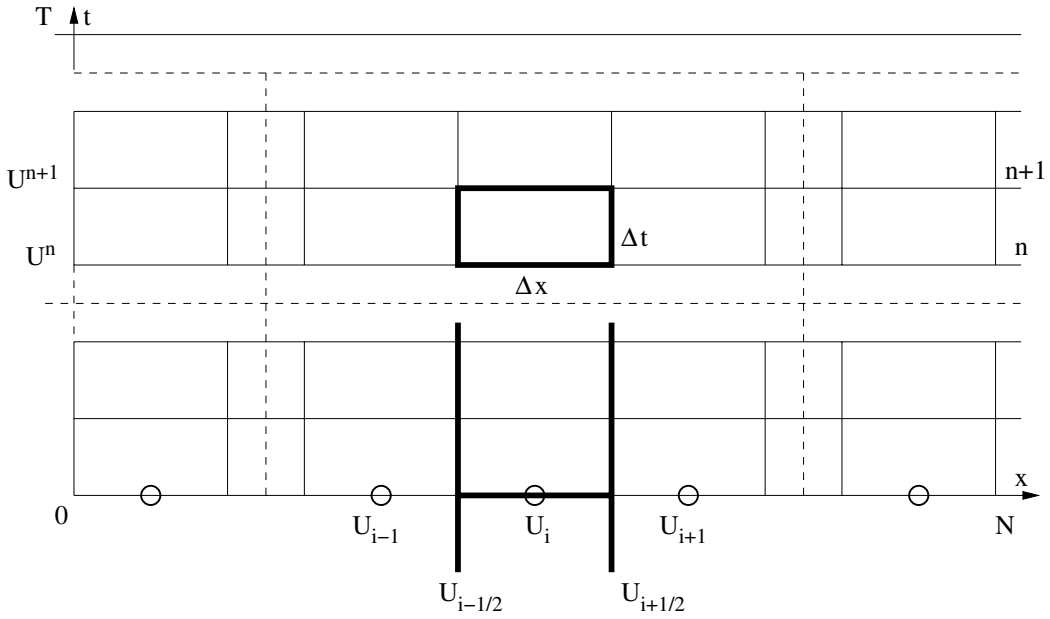


Figure 3.1 1D time-space discretization for SWE

In matrix form these equations can be written as:

$$\begin{bmatrix} \eta \\ u \end{bmatrix}_t + \begin{bmatrix} u & H \\ g & u \end{bmatrix} \times \begin{bmatrix} H \\ u \end{bmatrix}_x = \begin{bmatrix} 0 \\ -gh_x \end{bmatrix} \tag{3.9}$$

These equations are coupled in terms of primitive variables u and η/h.

The basic algorithm for the solution of one-dimensional shallow water equations is as shown in Fig.(3.4)

Steps to solve:

1. Using change of variables, the primitive variables can be written in terms of auxiliary variables, such that the governing equations are decoupled.

   In compact form, Eqns(3.9) without the source term can be rewritten as:

$$\mathbf{U}_t + \mathbf{A}\mathbf{U}_x = 0 \tag{3.10}$$

where $\mathbf{A}$ represents the matrix for flux Jacobian. $\mathbf{A}$ has two real and distinct eigenvalues which are:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} u + \sqrt{gh} \\ u - \sqrt{gh} \end{bmatrix}$$

and a complete set of linearly independent eigenvectors are:

$$\mathbf{L} = \frac{1}{2\sqrt{gh}} \begin{bmatrix} \sqrt{g} & \sqrt{h} \\ -\sqrt{g} & \sqrt{h} \end{bmatrix} \quad \text{and} \quad \mathbf{R} = \begin{bmatrix} \sqrt{h} & -\sqrt{h} \\ \sqrt{g} & \sqrt{g} \end{bmatrix}$$

Thus, matrix $\mathbf{A}$ can be diagonalized as:

$$\mathbf{A} = \mathbf{R\Lambda L} \quad \text{or} \quad \mathbf{\Lambda = LAR} \quad \left(\text{and } \mathbf{L = R^{-1}}\right) \tag{3.11}$$

The existence of inverse matrix $\mathbf{L}$ makes it possible to define a new set of dependent variables $\mathbf{W}$ via the transformation:

$$\mathbf{W = LU} \quad \text{or} \quad \mathbf{U = RW}$$

where the *characteristic variables* $\mathbf{W}$ turn out to be:

$$\mathbf{W} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \frac{1}{2\sqrt{g}} \begin{bmatrix} u + \sqrt{gh} \\ u - \sqrt{gh} \end{bmatrix}$$

so that the system of equations, when represented in terms of $\mathbf{W}$, becomes completely decoupled. The corresponding derivatives for new variables are:

$$\mathbf{U_t = LW_t}, \qquad \mathbf{U_x = LW_x}$$

Substitution of these expressions into Eqn(3.10) and after multiplying from the left by $\mathbf{R}$ and using Eqn(3.11) gives the so called canonical or characteristic form of system of equations:

$$\mathbf{W_t + \Lambda W_x} = 0 \tag{3.12}$$

When written in this case, the full system becomes:

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_t + \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_x = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.13}$$

For convenience, let $w_1 = r$ and $w_1 = s$. Thus, the speed with which the information travels is max($|u| + \sqrt{gh}$) and this bounds the time(Δt) for each iteration.

$$\Delta t = \frac{\Delta x}{\left(\max(|u| + \sqrt{gh})\right)} \tag{3.14}$$

2. Estimate the face values of r and s based on the center values and upwind them. This is achieved using two approaches: first order implementation and second order implementation.

3. Update the primitive variables at the cell centers. This is done in two sub-steps. First the characteristic variables are converted back into the primitive variables at the cell edges using:

$$\mathbf{U} = \mathbf{RW}; \quad \text{or} \quad \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \frac{1}{2\sqrt{g}} \begin{bmatrix} \sqrt{h}(w_1 - w_2) \\ \sqrt{g}(w_1 + w_2) \end{bmatrix}$$

Based on these face values, update the primitive variables at the cell center.

$$h^{n+1} = h^n - \frac{\Delta t}{\Delta x}\left(h_{i+\frac{1}{2}} u_{i+\frac{1}{2}} - h_{i-\frac{1}{2}} u_{i-\frac{1}{2}}\right) \tag{3.15}$$

$$u^{n+1} = u^n - \frac{1}{2}\left(\frac{\Delta t}{\Delta x}\right)^2 \left(u^2_{i+\frac{1}{2}} - u^2_{i-\frac{1}{2}}\right) - g\frac{\Delta t}{\Delta x}\left(h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}}\right) \tag{3.16}$$

Here the *half* values (*e.g.* $u_{i\pm 1/2}$) are the appropriate face values for the cell.

3.4.1 Flux differencing: First order implementation

First order implementation uses the following scheme for calculation of the edge values.

$$w_i^{n+1} = w_i^n + \lambda_i^n \frac{\Delta t}{\Delta x} \Delta w_{i+1/2}^{n+1/2} \tag{3.17}$$

denoting $w_L = w_i^n$ and $w_R = w_{i+1}^n$, Δw is determined in the following way:

$$\Delta w = \begin{cases} w_L & \text{if } w_L, w_R > 0 \\ w_R & \text{if } w_L, w_R < 0 \\ w_L & \text{if } (w_L > 0, w_R < 0) \;\&\; (w_L > -w_R) \\ w_R & \text{if } (w_L > 0, w_R < 0) \;\&\; (w_L < -w_R) \\ 0 & \text{if } (w_L < 0, w_R > 0) \end{cases} \tag{3.18}$$

3.4.2 Flux Differencing: Second order implementation

The idea behind making the scheme second order is to calculate $\Delta w_{i+1/2}^{n+1/2}$ in such a way so as to have a local truncation error of $O(\Delta t^2)$. This can be done by using piecewise linear profiles for variables w, instead of constant linear profiles, as described in [7].

Such an estimate can be obtained by using van Leer's algorithm as shown:

$$w_i^{n+1} = w_i^n + \lambda_i^n \frac{\Delta t}{\Delta x} \Delta_i w_{i+1/2}^{n+1/2}$$

where

$$w_{i+1/2}^{n+1/2} = w_i + \frac{1}{2}\left(1 - \lambda_i \frac{\Delta t}{\Delta x}\right) \overline{\Delta}_i w$$

and

$$\overline{\Delta}_i w = \begin{cases} \delta \text{ sign}\left(w_{i+1} - w_{i-1}\right) & \text{if } \Delta_i w_{i+1} \Delta_i w > 0 \\ 0 & \text{otherwise} \end{cases}$$

such that:

$$\delta = \min\left(2\left|w_{i+1} - w_i\right|, 2\left|w_i - w_{i-1}\right|, \frac{1}{2}\left|w_{i+1} - w_{i-1}\right|\right)$$

procedure 1D_SWE_Solve()

1: Initial conditions

2: **while** (time<final_time) **do**

3: estimate Δt

4: primitive variables $\Longrightarrow$ auxiliary variables

5: calculate auxiliary values at the cell edges

6: at the cell edges: auxiliary variables $\Longrightarrow$ primitive variables

7: update primitive variables

8: apply boundary conditions

9: **end while**

CHAPTER IV

ADAPTIVE MESH REFINEMENT

4.1 Nesting Algorithm

Shallow water flows have structures that vary in scales. When solving these problems, high grid resolution is needed to adequately solve the equations. However, there are often large portions of the domain where high levels of refinement represents waste of computational effort. Limitations on computational resources often force a compromise on grid resolution, resulting in under-resolution of the problem. By locally refining the mesh only where needed, Adaptive Mesh Refinement allows concentration of effort where it is needed, allowing better resolution of the problem.

Unfortunately, local refinement introduces several added issues like matching the solution at grid interfaces. These will be examined in more depth.

The Nesting algorithm is based on the AMR method used by Berger and Oliger[5]. This method attains a given accuracy with a minimum amount of work. The base grid consists of one coarse-resolution grid covering the entire domain of computation, and the fine-resolution grid is embedded to resolve a feature which needs high resolution.

Any number of methods can be used to identify regions where refinement is desired. An error estimation procedure can be used to automatically determine where the solution

resolution is inadequate, and the grid generation procedure can create fine-grid patches dynamically in these regions. The present approach uses local truncation error presented by Berger and Oliger[5].

If desired, these refined regions may be refined still further where necessary to adequately resolve the solution.

4.2 Discretization

Nested grids are generated using the block structured approach. In this strategy, patches or sub-domains are refined instead of individual cells. This way, areas that need more accuracy can be covered with a relatively small number of refined blocks. This results in more internal area for the amount of coarse-fine interface processing. While a block structured approach results in some unnecessary refinement, the advantages of this approach are better efficiency of data access and better amortization of the overhead costs of irregular operations, such as inter-level communication, over a large number of regular operations on a grid.

This block structured approach was first developed by Berger and Oliger[5] for hyperbolic PDEs, and was extended to shock hydrodynamics by Berger and Colella[4]. Bell et al[3] found this approach for three-dimensional systems of conservation laws.

In general, there will be different levels of refinement, $0 \leq \ell \leq \ell_{max}$. Level 0 is the coarsest level that covers the entire domain. Finer levels are made up from one or more refined patches which do not cover the entire domain.

Let Ω^ℓ denote the union of these patches, $\partial\Omega^\ell$ denote the boundary of level ℓ. Where they do not extend all the way to a physical boundary, refined patches need to use information from coarse grids to provide boundary conditions. Exchange of information between the grids adds constraint on the nesting technique. Thus, a refined grid must be bordered by a physical boundary where normal physical boundary conditions apply, another grid at the same level of refinement (where boundary conditions are simply copied from the adjoining grid), or grids with only one level of refinement less (where boundary conditions will be interpolated from the coarse level. Written formally,

$$R\left(P\left(\Omega^{\ell+1}\right)\right) = \Omega^{\ell+1}$$

$$P\left(\Omega^{\ell+1}\right) \subset \Omega^{\ell}$$

where P is the projection/prolongation operator, $\ell \rightarrow \ell-1$ and R is the refining/restriction operator $\ell \rightarrow \ell+1$. Boundaries of $\Omega^{\ell+1}$ can intersect boundaries of Ω^ℓ only at the boundary of the domain. Fig(4.2) shows a properly nested grid.

It is important to note that grids are not patched into the coarse grid. Rather, a fine grid is overlaying over a coarse grid. Each grid is defined independently of the other grids, with its own solution vector, storage, etc. In this way, each patch grid can be integrated (almost) independently of the other grids. Also, it allows for the possibility of using moving grids even if the coarsest grid is stationary.

Each level of grids is refined by a factor defined as Refinement ratio from the level below it. This refinement ratio of spatial and temporal resolution between the two grids is an integer and is given as:

$$\text{Refinement Ratio} = \frac{\Delta x_{\text{coarse}}}{\Delta x_{\text{fine}}} = \frac{\Delta t_{\text{coarse}}}{\Delta t_{\text{fine}}} \tag{4.1}$$

With large values of refinement ratio, much of the fine scale phenomena on the fine grid are not resolvable on the coarse grid and this may cause increased disturbances of signals passing from the fine grid to the coarse grid thus resulting in instabilities and unsmooth solutions at the interfaces. This not only increases discretization errors at the interface but also require unnecessarily large portion of the domain to be refined. On the other hand, lower values increase the required number of levels and interfaces and thus increasing the wasted storage devoted to coarse cells underlying fine grids.

The basic algorithm is summarized in Section(4.2). The solution in the coarse grid is advanced one time step. Then this solution is interpolated in time and space to provide boundary conditions for the fine grid. The solution at the fine grid is then updated. This coarse grid solution is corrected using the fine grid solution as the final step of each iteration. Both the coarse grid and fine grid have initial conditions at the initialization step.

Time stepping used in the algorithm keeps equal Courant numbers on all levels so that the quality of the wave propagation relative to the mesh size will be approximately the same, (i.e. time step is refined with the same factor as spatial resolution).

4.2.1 Refluxing

The algorithm incorporates a refluxing step, in which edge fluxes for coarse grid cells at the coarse-fine interface are adjusted to equal those computed for neighboring fine cells, thus enforcing conservation. Since two way interaction (between coarse and fine grid solution) is being implemented, the coarse grid solution is updated as follows.

The fluxes at the interface of the fine grid and coarse grid are stored, so called “coarse flux” and the “fine flux” at the interface. Fluxes at the interface are computed from the fine grid solution and summed over all the fine steps (subcycles = refinement ratio) that constitute a coarse time step. This gives so called “fine flux”. The corresponding “coarse flux” through the interface is also computed, using the coarse solution only. Then the difference between the fine flux and coarse flux is uniformly distributed to correct the coarse values along the interfaces in order to ensure conservation.

4.2.2 Quadratic Interpolation

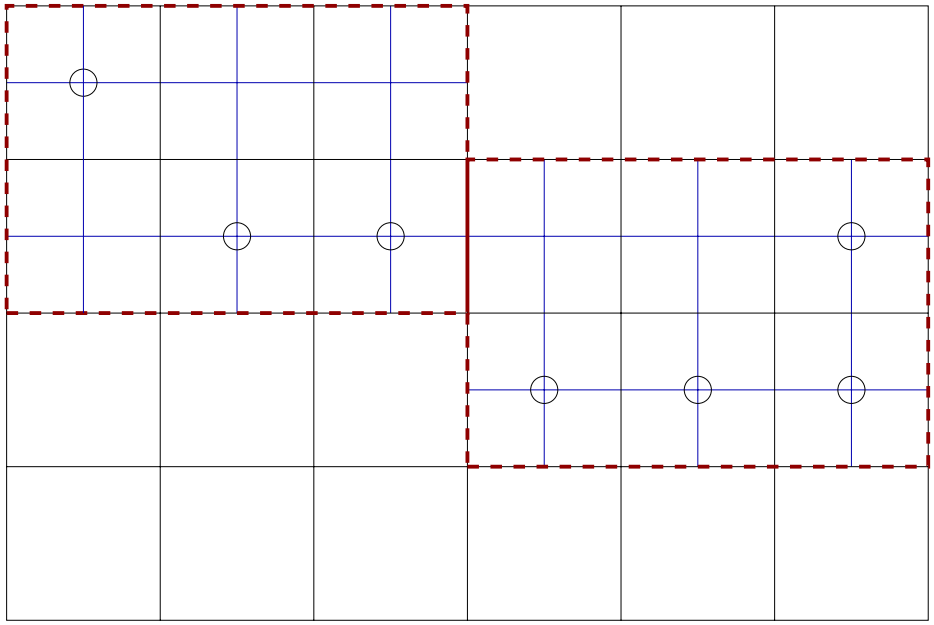
Quadratic interpolation is used wherever possible to link the coarse and fine grids. The way this is achieved is shown in Fig(4.4). Coarse grid values are interpolated to get the intermediate values. These intermediate values are then used with fine cells to get ghost cell values for computing coarse-fine fluxes.

If one of the coarse grid cells in the usual stencil is covered by a finer grid, then the stencil is shifted so that only the coarse cells in $(\Omega^l - P(\Omega^{l+1}))$, are used in the interpolation parallel to the coarse-fine interface. Fig(4.5) shows the use of quadratic interpolation

stencils for one such special case of fine grid corners. In this case, since the left coarse cell is covered by a fine grid, a shifted coarse grid stencil is used to get the intermediate values. The final interpolation is performed as before to get the ghost cell values.

If a suitable coarse grid stencil parallel to the interface does not exist, then the order of interpolation is dropped and the available coarse cells are used.

Interpolating using both coarse and fine grids and using the same fluxes for both coarse and fine grids links the two grids enough to fix the coarse-fine interface problem.



◯ Tagged Cells

Figure 4.1 Block structured refinement of cells

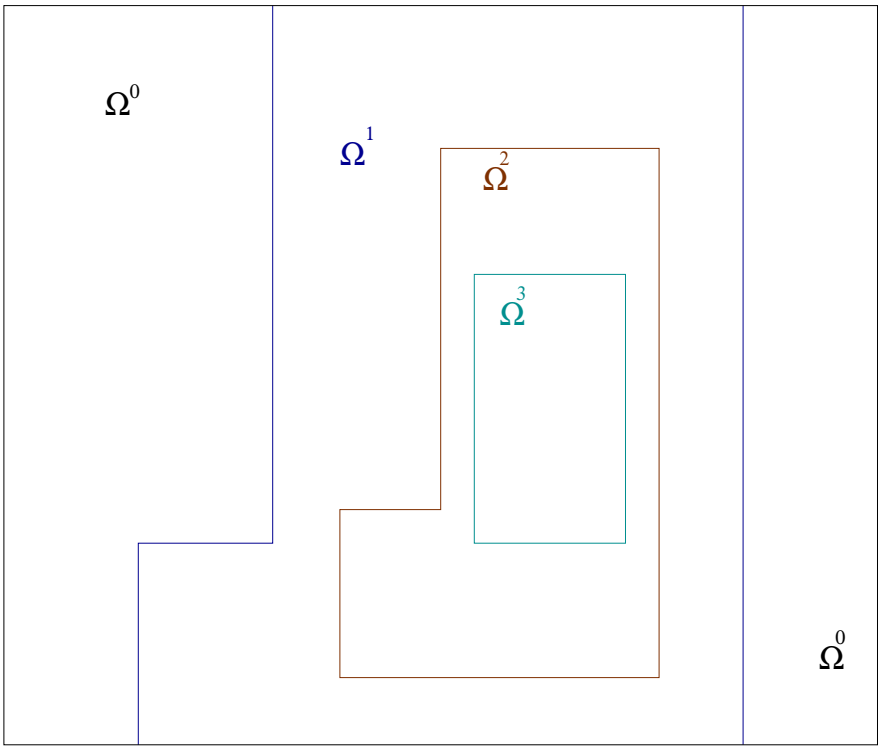


Figure 4.2 A properly nested grid

procedure Nested_SWE_Solve()

```
1:  set refinement ratio
2:  Initialize coarse grid
3:  Initialize fine grid
4:  while (time<final_time) do
5:     estimate Δt
6:     primitive variables ⟹ auxiliary variables
7:     calculate auxiliary values at the edges
8:     at the edges: auxiliary variables ⟹ primitive variables
9:     update primitive variables
10:    apply boundary conditions
11:    reset the fluxes at the boundary of fine grid
12:    for subcycle=1 to subcycle=refinement ratio do
13:       get the ghost cell values of the fine grid
14:       (fine grid) primitive variables ⟹ auxiliary variables
15:       (fine grid) calculate auxiliary values at the edges
16:       (fine grid) at the edges: auxiliary variables ⟹ primitive variables
17:       (fine grid) update primitive variables
18:       get the fluxes across the both the fine boundaries
19:    end for
20:    reflux primitive variables using the fluxes from fine grid
21:    correct the coarse grid, where it overlaps with fine grid
22: end while
```

Figure 4.3 One-dimensional AMR Algorithm

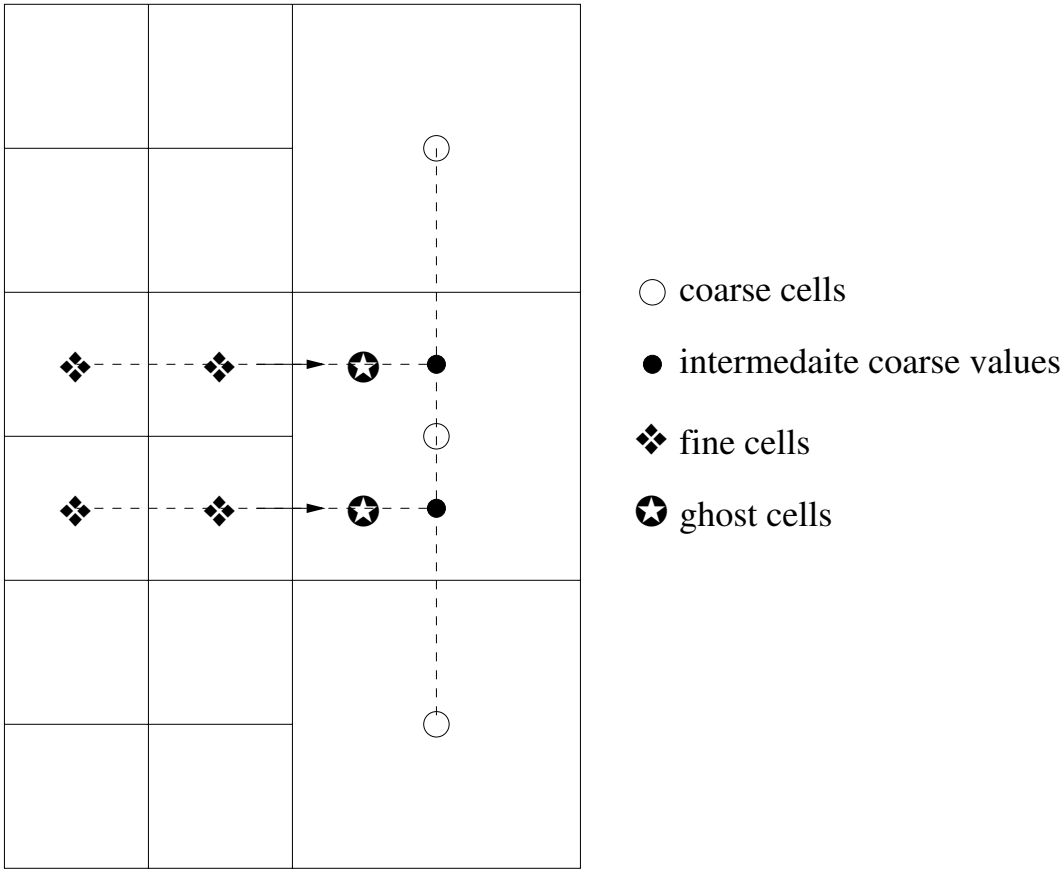


Figure 4.4 Quadratic interpolation

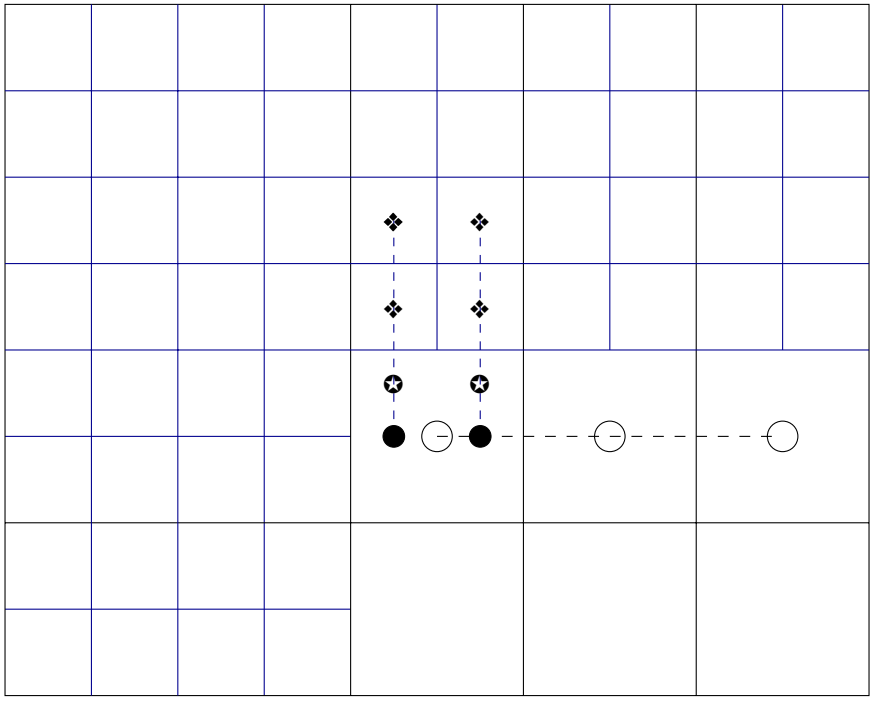


Figure 4.5 Quadratic interpolation (special case)

CHAPTER V

RESULTS

5.1 Single Grid Results

To convey the proof of concept, the algorithm(3.4) described in Chapter(III) has been implemented in one-dimension. The present section shows the numerical results obtained for the single-grid version of the scheme. A stationary wave is considered as a test case for the test run.

The plot in Fig(5.1) shows the result of a wave with the initial condition and the final result.

The grid consists of 64 points and the depth field is initialized as:

$$h = 0.1He^{-50(x-0.5)^2} + H; \tag{5.1}$$

and the velocity field is initialized by specifying:

$$u = 2\sqrt{gh} - 2\sqrt{gH} - \sqrt{gH} \tag{5.2}$$

These initial conditions make the wave approximately stationary. The final result consists of the solution after a non-dimensional time period of 1.

The table computes the $\pounds 1$ and $\pounds 2$ Norms for different grids ranging from 32 to 256 grid points, to depict the order of the scheme. Thus, from the ratio of norms we conclude that the second case of edge calculations indeed is second order accurate.

Table 5.1 Norms for single grid case

GRID	Steps	$\pounds 1$ Norm $\frac{\lvert(h_{f(avg)}-h_c)\rvert}{N}$	$\pounds 2$ Norm $\sqrt{\frac{\Sigma(h_{f(avg)}-h_c)^2}{N}}$	Order ($\pounds 1$)	Order ($\pounds 2$)
32	63	2.08695e-05	8.09821e-06	0.308755	0.232484
64	126	6.44358e-06	1.8827e-06	0.372893	0.245603
128	253	2.40277e-06	4.62397e-07	0.45797	0.287777
256	507	1.1004e-06	1.33067e-07		

5.2 Results for Nesting Algorithm

The goal of adaptive mesh refinement is to have accuracy of the fine grid with a reduced computation cost. Thus, AMR scheme should compute the results such that the difference between the modified solution and fine solution is very small as compared to the difference between the coarse solution and the fine solution. Where modified solution is the solution obtained using adaptive mesh refinement technique, coarse solution is the result of single-grid implementation and the fine solution is the single-grid implementation with the grid size defined by refinement ratio.

As a proof, the nesting algorithm(Section 4.2) described in chapter(III) has been implemented in one-dimension. The test case considered here is same, *i.e.*, a stationary wave.

Fig(5.2) shows the result with the initial condition and the final result. The grid consists of 32 points. For coarse grid the depth field is initialized as:

$$h = 0.1He^{-50(x-0.5)^2} + H; \tag{5.3}$$

and the velocity field is initialized specifying:

$$u = 2\sqrt{gh} - 2\sqrt{gH} - \sqrt{gH} \tag{5.4}$$

These initial conditions make the wave stationary. Same initial conditions are applied to fine grid as well. The final result consists of the solution after a time period of 1.

The plot shows the solution obtained using coarse grid and the solution obtained using the nesting/AMR technique. Here the number of grid points is 32 for coarse and the refinement ratio is 2.

The table computes the $\pounds 1$ and $\pounds 2$ Norms for different grids ranging from 32 through 256 grid points, to depict the effect of nesting technique on the solution.

Table 5.2 Norms for adaptive mesh refinement case

GRID	Steps	$\pounds 1$ Norm $\frac{\lvert(h_m - h_c)\rvert}{N}$	$\pounds 2$ Norm $\sqrt{\frac{\Sigma(h_m - h_c)^2}{N}}$	$\pounds 1$ Norm $\frac{\lvert(h_{f(avg)} - h_m)\rvert}{N}$	$\pounds 2$ Norm $\sqrt{\frac{\Sigma(h_{(avg)} - h_m)^2}{N}}$
32	63	2.44323e-05	9.19444e-06	5.03465e-06	1.8331e-06
64	126	8.0671e-06	2.27586e-06	1.76424e-06	4.76764e-07
128	253	3.08911e-06	5.71052e-07	7.78192e-07	1.25476e-07
256	507	1.46155e-06	1.73289e-07	3.91802e-07	4.27089e-08

It is seen from the values of norms for modified solution, fine solution and the coarse solution, that the difference between the modified solution and the fine solution is much smaller as compared to the difference between fine solution and the coarse solution. Thus, it can be concluded that the solution obtained by the nesting technique is more accurate.

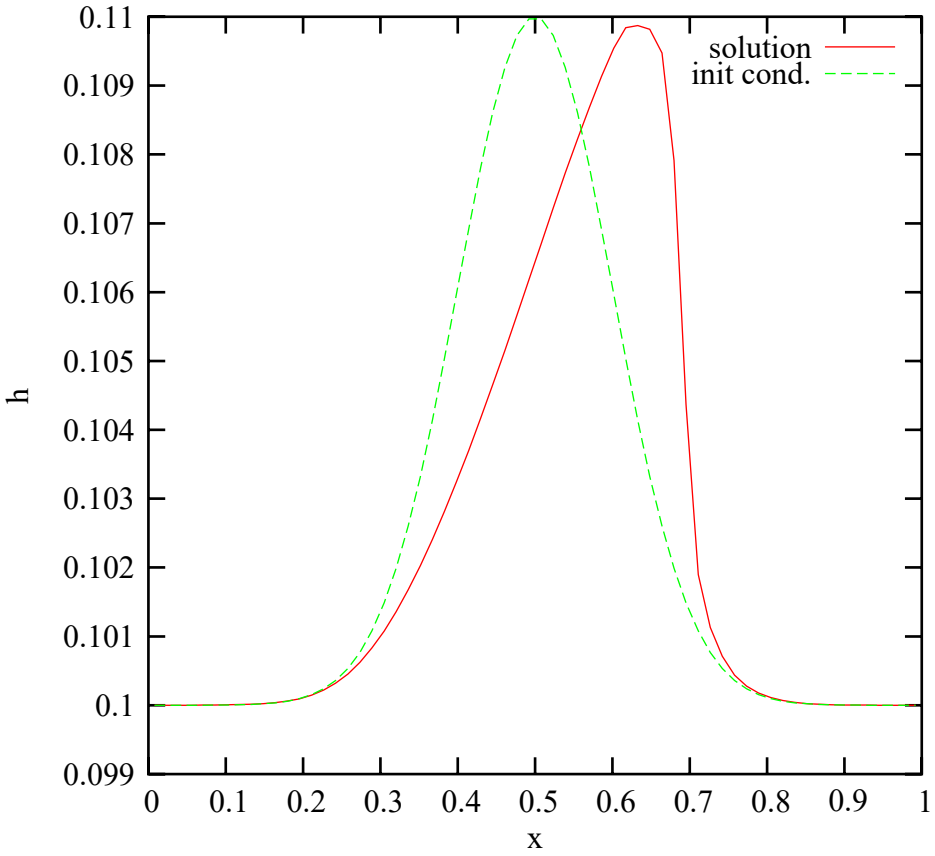


Figure 5.1 Results of 1D SWE for a stationary wave

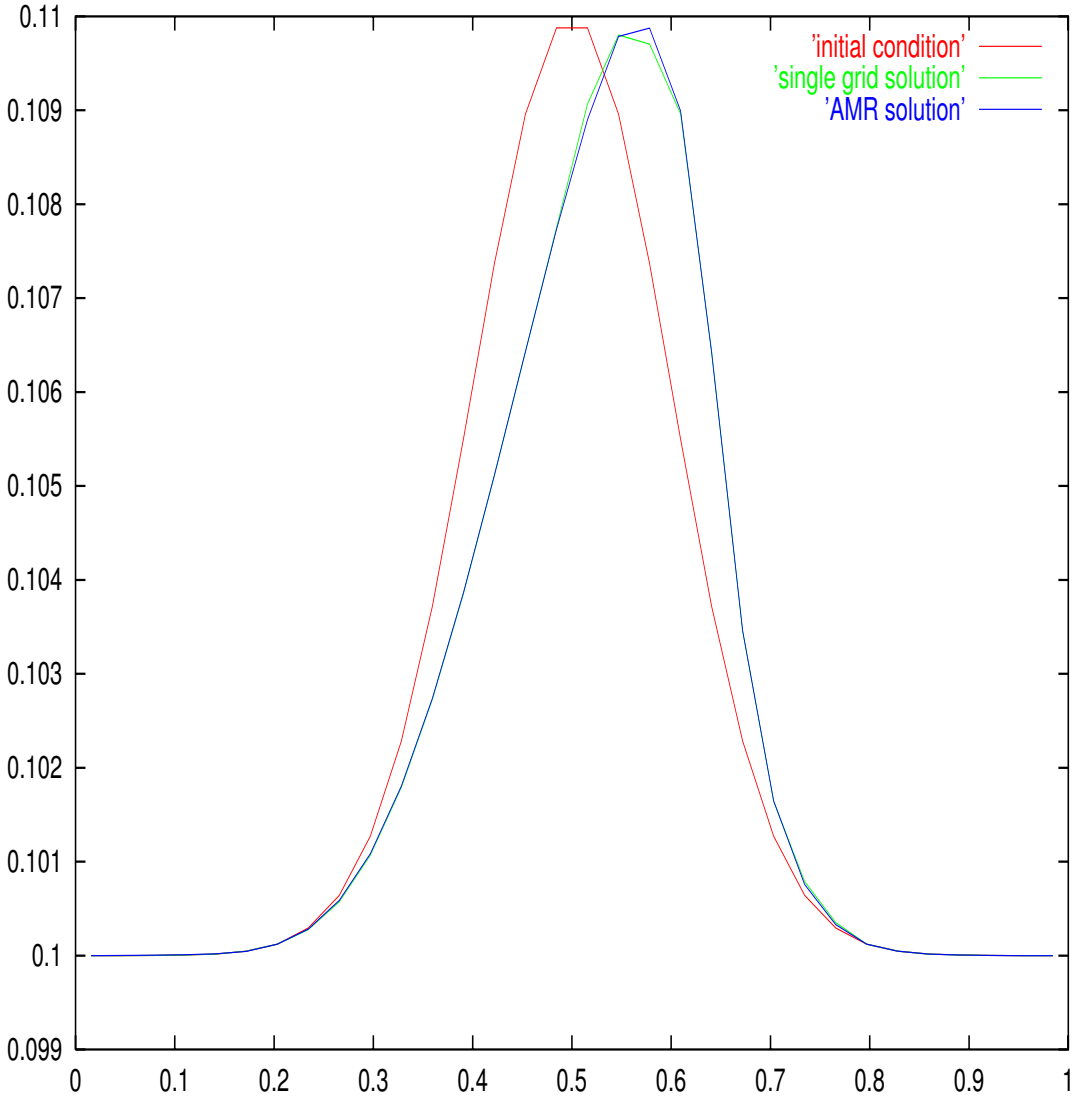


Figure 5.2 1D Nested SWE results for stationary wave

CHAPTER VI

CONCLUSION AND FUTURE WORK

6.1 Proposed work for 2D SWE

A second order one-dimensional solver for shallow water equations is developed. This method is then extended to incorporate the technique of Burger and Oliger to make the scheme adaptive. The adaptive solution is shown to exhibit the same level of accuracy as the fine with a reduced number of cells. The method can be extended to two-dimensions to provide efficient solution technique for a large class of geophysical problems.

REFERENCES

[1] *The Miami Isopycnic Coordinate Model*, http://oceanmodeling.rsmas.miami.edu/micom.

[2] *The Parallel Ocean Program*, 2003, http://climate.lanl.gov/Models/POP.

[3] S. J. Bell J., Berger M.J. and W. M., "Three dimensional adaptive mesh refinement for Hyperbolic conservation laws," *Journal of Sci. Comput.*, vol. 15(1), 1994, pp. 127–138.

[4] M. Berger and P. Colella, "Local adaptive mesh refinement for Hyperbolic partial differential equations," *Journal of Computational Physics*, vol. 82(1), 1989, pp. 64–84.

[5] M. J. Berger and J. Oliger, "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," *Journal of Computational Physics*, vol. 53, 1984, pp. 484–512.

[6] V. Casulli and R. Cheng, "Solutions of Primitive Equations for Three-dimensional Tidal Circulation," *Estuarine and Coastal Modeling III, Proceedings of 3rd International Conference*. 1994, pp. 396–406, ASCE, New York.

[7] P. Colella, "Multidimensional Upwind Methods for Hyperbolic Conservation Laws," *Journal of Computational Physics*, vol. 87, 1990, pp. 171–200.

[8] P. Colella and P. R. Woodward, "The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations," *Journal of Computational Physics*, vol. 54, 1984, pp. 174–201.

[9] B. Cushman-Roisin, *Introduction to Geophysical Fluid Dynamics*, Prentice Hall Inc., 1994.

[10] A. Fox and S. Maskell, "A nested primitive equation model of the Iceland-Fareoe front," *Journal of Geophysical Research*, vol. 101, 1996, pp. 18259–18278.

[11] S. Godunov, "Finite Difference Method for Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics," *Mat. Sbornik*, vol. 47, 1959, pp. 271–306.

[12] C. Hirsch, *Numerical Computation of Internal and External Flows: Volume 1 Fundamentals of Numerical Discretization*, John Wiley & Sons, 1988.

[13] C. Hirsch, *Numerical Computation of Internal and External Flows: Volume 2 Computational Methods for Inviscid and Viscous Flows*, John Wiley & Sons, 1988.

[14] K. H. Ib A. Svendsen and Q. Zhao, *Quasi-3D Nearshore Circulation Model SHORECIRC*, 1991, http://chinacat.coastal.udel.edu/∼kirby/programs/shorecirc/shorecirc.html.

[15] L. H. Kantha and C. A. Clayson, *Numerical Models of Oceans and Oceanic Processes*, Academic Press, 2000.

[16] R. Luettich, *A Parallel Advanced Circulation Model for Oceanic, Coastal and Estuarine Waters*, 2000, http://www.marine.unc.edu/C_CATS/adcirc/adcirc.htm.

[17] P. J. Martin, *A Description of the Navy Coastal Ocean Model*, 2000, Naval Research Laboratory.

[18] G. L. Mellor, *A Three-dimensional, Primitive Equation, Numerical Ocean Model*, 1998, http://www.aos.princeton.edu/WWWPUBLIC/htdocs.pom.

[19] P. Roe, "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, vol. 43, 1981, pp. 357–372.

[20] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer-Verlag Berlin, 1999.