NUMERICAL SIMULATION OF TSUNAMI ASSOCIATED WITH THE INDONESIAN TSUNAMI 2004

by

JOAN GOH YAH RU

Dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science in Mathematics

April 2007

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Md. Fazlul Karim for teaching me and helping me a lot to accomplish this project.

I would also like to thank Prof. Madya Ahmad Izani Md. Ismail and also Prof. Gauranga Deb. Roy, who have supported this work throughout, as well as for the extensive use I have made of their experience in this subject area and for my informative periods of study during this work.

CONTENTS

LIST OF FIGURES

SIMULASI BERANGKA TSUNAMI BERKENAAN DENGAN TSUNAMI INDONESIA 2004

ABSTRAK

Tsunami Indonesia (Disember 26, 2004) disahkan sebagai tsunami yang paling kuat di dunia ini dalam masa 40 tahun yang lalu. Tsunami ini telah menyebabkan kebinasaan dan telah membawa malapetaka nasional ke beberapa kawasan di Lautan Hindi, ini termasuklah Semenanjung Malaysia dan Utara Sumatera. Beberapa model berasaskan persamaan air-cetek tak linear dalam bentuk sistem koordinat silinder telah dibangunkan dan hasilnya telah dikembangkan dalam kajian sebelumnya (Roy et al., 2007). Dalam kajian tersebut, skima pembezaan terhingga (masa ke hadapan dan ruang di tengah) telah digunakan untuk menyelesaikan persamaan berkenaan. Dalam kajian ini, simulasi berangka telah dipersembahkan dengan bantuan model ini untuk mensimulasikan tsunami Disember 26, 2004. Pengamiran tegak terhadap persamaan air-cetek ini diselesaikan dengan mengunakan pengiraan lompat katak ke atas sistem grid bergilir-gilir. Gelombang tsunami sampai di Pulau Pinang, Malaysia dan Utara Sumatera dalam masa 240 minit dan 30 minit masing-masing dan ini adalah bersepadan dengan peristiwa tsunami sebenar pada Disember 26, 2004. Kajian ini juga memberi amplitude gelombang tsunami dan masa ketibaannya. Hasil model tersebut adalah menyahihkan bidang tinjauan dengan data yang didapati dalam laman web. Penarikan balik air pertama dari kawasan pantai juga diperhatikan. Didapati sifat

sumber juga memberi kesan penting ke atas penarikan balik air pertama dari kawasan pantai.

Kata isyarat: Malaysia, Negara Thai, Model air cetek, Disember 26, 2004 tsunami di Sumatera, Penyebaran tsunami dan ombak.

ABSTRACT

The Indonesian tsunami (December 26, 2004) is now established to be the strongest in the world over the past 40 years. It resulted in devastations amounting to national calamities in several parts of the Indian Ocean including Peninsular Malaysia and North Sumatra. A numerical model based on the nonlinear shallow-water equations in cylindrical polar coordinate system was developed and the results presented in a previous study (Roy et al 2007). In that study finite difference scheme (forward in time and central in space) was used to solve the equations. In this study numerical simulations are performed with the help of this model to simulate the tsunami of December 26, 2004. The vertically integrated shallow water equations are solved using leap-frog calculations on a staggered grid system. The tsunami waves reach Penang Island in Malaysia and North Sumatra within 240 minutes and 30 minutes respectively and those are in reasonable agreement with the real tsunami event of December 26, 2004. This study also gives the amplitude of the tsunami waves and its arrival time. The model results are validated with the field observations and the data available in USGS website. The initial withdrawal of water from the beach is also examined. It is also found that the nature of the source condition has a significant effect on the initial withdrawal of water from the coast.

Key words: Malaysia, Thailand, Shallow water model, December 26, 2004 tsunami at Sumatra, Tsunami propagation and surge.

CHAPTER 1

INTRODUCTION

1.1 Background

A tsunami is a series of ocean waves of extremely long wave length and long period generated in a body of water by an impulsive disturbance that displaces the water. These impulses can originate from undersea landslides, volcanoes and impacts of objects from outer space (such as meteorites, asteroids, and comets), but mostly, submarine earthquakes. Most of the tsunamis are caused by underwater earthquakes. However, not all under earthquakes cause tsunamis, only which earthquake over 6.75 on the *Richter scale* cause a tsunami (Tsunami[Online]). The energy generated by the earthquake is transmitted through the water. This energy in these seismic sea waves can travel virtually unnoticed in the deep oceans as the wave height may be only twelve inches. When this energy reaches the shallow waters of coastlines, bays or harbors, it forces the water into a giant wave. Some tsunamis may reach heights of 100 feet or more (Tsunami[Online]).

Factors such as earthquakes, mass movements above or below water, submarine landslides, volcanic eruptions and other underwater explosions, and large meteorite impacts may generate a tsunami.

Usually, underwater earthquakes cause tsunamis. These often occur at

subduction zones (places where a tectonic plate that carries an ocean is gradually slipping under a continental plate). When sections of the plates that have been locked together for a while move suddenly under the strain, part of the sea floor can snap upward suddenly, while other areas sink downward. Immediately after such an under water earthquake, the shape of the sea surface reflects the new contours of the sea floor. Some areas of water are pushed upwards, and others sink (Tsunami[Online]). Submarine landslides and the collapses of volcanic edifices may also disturb the overlying water column as sediment and rocks slide downslope and are redistributed across the sea floor. A violent submarine volcanic eruption can uplift the water column and form a tsunami as well. Once the event which initiates the tsunami occurs, the potential energy that results from pushing water above mean sea level is then transferred to horizontal propagation of the tsunami wave (kinetic energy).

The propagation and run-up of tsunami waves which generated by a landslide due to the eruption of the Mt. St. Augustine Volcano, Alaska in 1883 has been studied by Kienle et al. (1996). The model is based on the nonlinear shallow water equations and it is solved by the finite difference method. This model also simulates the run-up heights and inundation (the rising of a body of water and its overflowing onto normally dry land) patterns. The MOST (Method of Splitting Tsunami) Model has been developed by Titov and Gonzalez (1997) which simulates the tsunami generation, transoceanic propagation, and also the inundation of dry land; the generation process is based on elastic deformation theory (Okada, 1985). Elastic deformation is a non-permanent deformation of a body in which the stresses do not exceed its elastic limit. Once the stresses are no longer applied, the material returns to its original shape. Inelastic deformation leaves material permanently deformed. Zahibo et al. (2003) performed the numerical simulation of potential tsunamis in the

Caribbean Sea in the framework of the nonlinear shallow water theory. In the study of Kowalik and Whitmore (1991), the tsunamis have been simulated associated with the 1952 Kamchatka and 1986 Andreanof Islands earthquakes along Adak of Alaska. Very recently, a global model has been developed by Kowalik et al. (2005) to simulate the generation, propagation and run-up of the tsunami associated with the 26 December 2004. The spherical polar shallow water model which they used is incorporating a very fine mesh resolution of one minute with about 200 million grid points.

When the tsunami generate, the tsunami wave will travels outward in all directions from the source with a very high velocity. As a result, the people of the surrounding coastal belts must be warned about the arrival time and intensity of wave amplitude immediately once the tsunami generate. By the reason of tsunami propagate in a high speed, the time lag between the generation and inundation along a coastal belt is subsequently small. Thus, there is no much time is available for model simulation purpose. Hence, any global model which takes a long time for simulation (like Kowalik's) is not effective for real time simulation. For this purpose, a regional model may be more effective and efficient as it takes less simulation time and it is possible to incorporate fine resolution (small grid size) in the numerical scheme. Therefore, a more detail information about intensity and other aspects of the tsunami around the interested region may be obtained.

1.2 Indonesian Tsunami 2004

The 2004 Indian Ocean earthquake, known by the scientific community as the Sumatra-Andaman earthquake, was a great undersea earthquake that occurred at 00:58:53 UTC (07:58:53 local time) December 26, 2004, which was a M = 9.0 mega

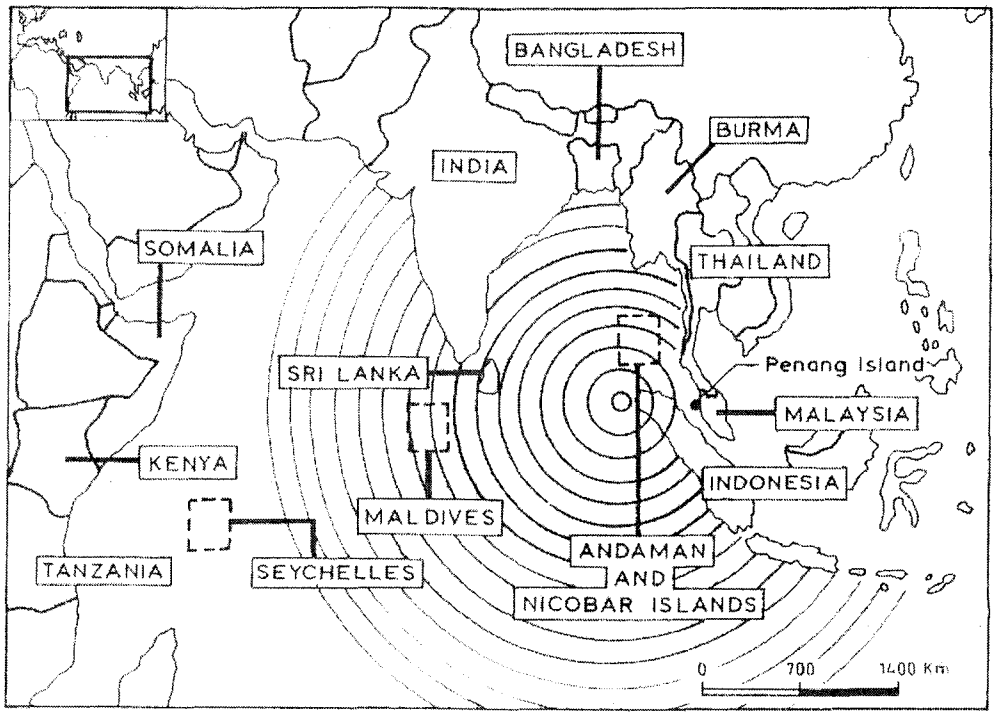

Fig. 1.1: Major affected regions around the Indian Ocean by tsunami on 26 December 2004 generated near Sumatra.

thrust earthquake occurred along 1000 km of the subduction zone west of Sumatra, Indonesia and Thailand in the Indian Ocean.

The earthquake was originally reported as 9.0 on the *Richter scale*, but has been revised and upgraded by some scientists to 9.3 in February 2005. This earthquake was reported to be the longest duration of faulting ever observed, lasting between 500 and 600 seconds, and it was large enough that it caused the entire planet to vibrate at least half an inch, or over a centimeter. It also triggered earthquakes in other locations as far away as Alaska (2004 Indian Ocean Earthquake[Online]).

The tsunami caused tremendous loss of life and property along the coastal regions surrounding the Indian Ocean including west coasts of Peninsular Malaysia

and North Sumatra (Fig. 1.1). The west coasts of peninsular Malaysia and North Sumatra are in vulnerable positions for tsunami due to a source close to Sumatra. So, it is necessary that tsunamis are studied in detail and prediction models be developed to simulate propagation and to estimate amplitude along the coastal belts.

1.3 Objectives

The west coast of peninsular Malaysia including Penang is vulnerable to the effects of seismic sea waves, or tsunamis, generated along the active subduction zone of Sumatra and that was demonstrated on December 26, 2004. More recent earthquakes have also been reported.

Roy et al. (1999) developed a polar coordinate shallow water model to compute tide and surge due to tropical storms along the coast of Bangladesh. The model of Roy et al. (1999) later was improved by Haque et al. (2003) to achieve a finer resolution in the numerical scheme along the coastal belt of Bangladesh. Following the approach of Roy et al. (1999) and Haque et al. (2003), Roy et al. (2007) developed a Cylindrical Polar model to simulate tsunami wave propagation and to estimate water level and other related aspects along the coast of Penang Island and North Sumatra associated with 26 December 2004 tsunami source at Sumatra. In that study, they have used finite difference scheme (forward in time and central in space) to solve the vertically integrated shallow water equations and boundary conditions.

In the present study, the model of Roy et al. (2007) has been used to simulate the propagation of 26 December 2004 tsunami wave towards the Penang Island in Malaysia and North Sumatra and to estimate the water levels along its coastal belts.

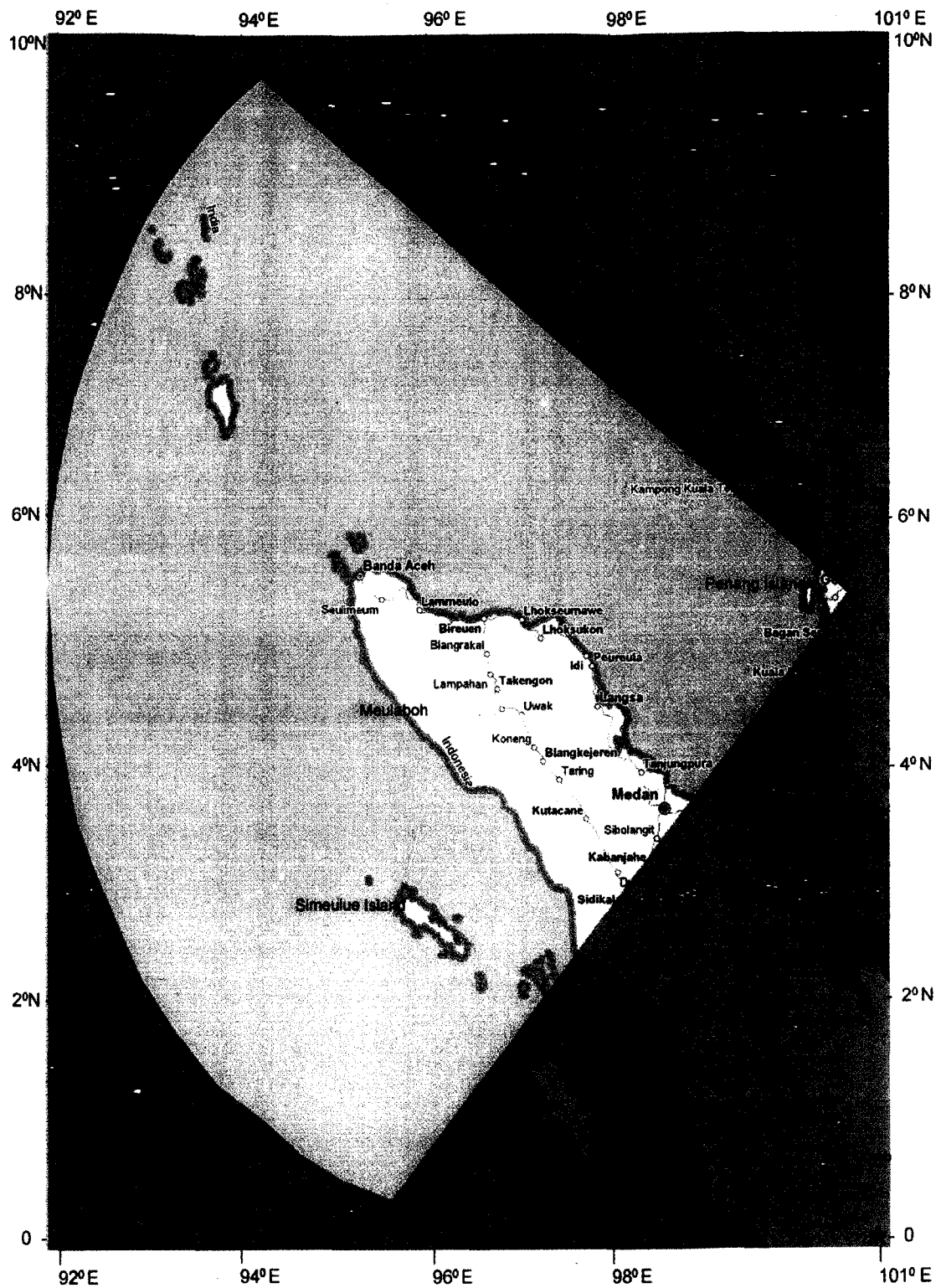

Fig. 1.2: Model domain including pole of the coordinate system and boundaries

The model has also been applied to compute other related aspects of tsunami surrounding the both the Islands. In particular, leap-frog method on a staggered grid system has been used to solve the equations in this study. The pole in this model is set at the main land of Penang (100.5°*E*) and extending the model region up to west of Sumatra Island (92°*E*) as shown in Figure 1.2.

1.4 Outline of Dissertation

The aim of this work is to simulate the effect of Indonesian tsunami 2004 towards Penang Island in Malaysia and North Sumatra using the nonlinear Polar coordinate shallow water model of Roy et al (2007) using leap-frog calculations on a staggered grid system. This thesis consists of six chapters. In the first chapter, the background of this study tsunami and its characteristics especially the character of Indonesian Tsunami 2004 and also the objective of this study is discussed. In chapter 2 the governing equations and also the boundary conditions are discussed. Numerical approximations are described in chapter 3. The source generation mechanism (rupture process) is described in chapter 4. Particular attention is paid to the initial conditions of the tsunami source. In chapter 5 results of simulations are presented. Finally in chapter 6 results are summarized by offering the conclusions of this study.

CHAPTER 2

GOVERNING EQUATIONS AND BOUNDARY CONDITIONS

2.1 Introduction of Shallow Water Model

The shallow water equations are a set of equations that describe the flow below a horizontal pressure surface in a fluid. The flow that describe in these equations is the horizontal flow caused by changes in the height of the pressure surface of the fluid. Shallow water equations can be used in atmospheric and oceanic modeling. Shallow water equation models cannot encompass any factor that varies with height since they have only one vertical level (Shallow Water Equations[Online]).

2.2 Vertically Integrated Shallow Water Equations

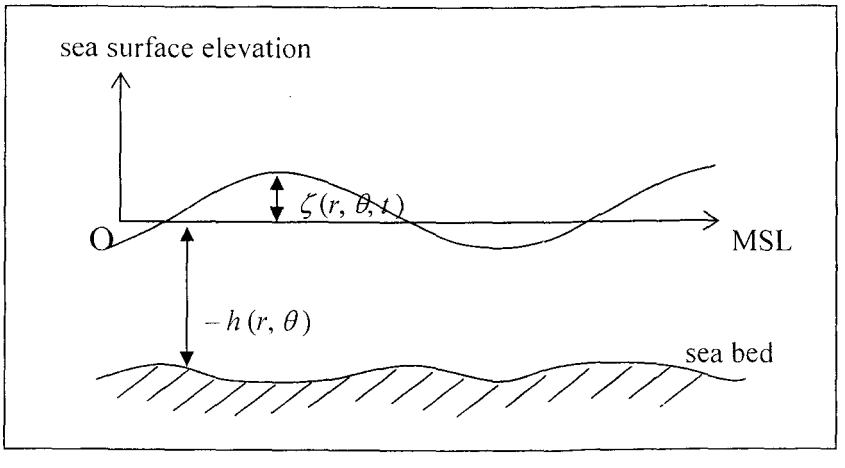

Fig. 2.1: The displaced position of the sea surface and the position of the sea floor

In order to derive the vertically integrated shallow water equation, a system of cylindrical polar coordinates is used in which the origin, O, is in the undisturbed level of the sea surface which is considered as the $r\theta$ - plane and O_z is directed vertically upwards. Let consider the displaced position of the free surface as $z = \zeta(r, \theta, t)$ and the position of the sea floor as $z = -h(r, \theta)$ so that the total depth of the fluid layer is $\zeta + h$. According to Roy et al. (2007), the vertically integrated nonlinear shallow water equations are

$$\frac{\partial \zeta}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}[r(\zeta + h)v_r] + \frac{1}{r}\frac{\partial}{\partial \theta}[(\zeta + h)v_\theta] = 0 \tag{1}$$

$$\frac{\partial v_r}{\partial t} + v_r\frac{\partial v_r}{\partial r} + \frac{v_\theta}{r}\frac{\partial v_r}{\partial \theta} - fv_\theta = -g\frac{\partial \zeta}{\partial r} - \frac{F_r}{\rho(\zeta + h)} \tag{2}$$

$$\frac{\partial v_\theta}{\partial t} + v_r\frac{\partial v_\theta}{\partial r} + \frac{v_\theta}{r}\frac{\partial v_\theta}{\partial \theta} + fv_r = -\frac{g}{r}\frac{\partial \zeta}{\partial \theta} - \frac{F_\theta}{\rho(\zeta + h)} \tag{3}$$

where

v_r = radial component of velocity of the sea water

v_θ = tangential component of velocity of the sea water

F_r = radial component of frictional resistance at the sea bed

F_θ = tangential component of frictional resistance at the sea bed

f = Coriolis parameter $= 2\Omega\sin\varphi$

Ω = angular speed of the earth

φ = latitude of the location

g = acceleration due to gravity

The parameterization of the frictional resistance at the sea bed, F_r and F_θ, are done by the conventional quadratic law :

$$F_r = \rho\, C_f v_r \sqrt{v_r^2 + v_\theta^2} \quad \text{and} \quad F_\theta = \rho\, C_f v_\theta \sqrt{v_r^2 + v_\theta^2} \tag{4}$$

where C_f = coefficients of friction, ρ = sea water density

2.3 Boundary Conditions

The coastal belts of the Malaysia main land and Sumatra islands are the closed boundaries where the normal components of the current are taken as zero. The radiation type of boundary condition allows the disturbance, generated within the model area, to go out through the open boundary. The analysis area is bounded by the radial lines $\theta = 0°$, $\theta = \Theta = 92°$ through O and the circular arc $r = R$. The radiation boundary conditions for the southern, northern and western open sea boundaries, are respectively given by Roy et al. (2007) as

$$v_\theta + \sqrt{g/h}\,\zeta = 0 \quad \text{along} \quad \theta = 0 \tag{5}$$

$$v_\theta - \sqrt{g/h}\,\zeta = 0 \quad \text{along} \quad \theta = \Theta \tag{6}$$

$$v_r - \sqrt{g/h}\,\zeta = 0 \quad \text{along} \quad r = R \tag{7}$$

2.4 Transformation for Uneven Resolution along Radial Direction

The polar coordinate system automatically ensures finer resolution along tangential direction near the region of the Pole. A uniform grid of size $\Delta\theta$ can be generated in the tangential direction by a set of radial lines through the Pole, by setting the Pole suitably at the location where fine resolution is required. The arc distance between any two consecutive radial lines decreases towards the Pole and increases away from the Pole. Thus, in terms of arc distance, an uneven resolution can be obtained in the tangential direction although uniform grid size $\Delta\theta$ is used (Roy et al., 2007).

As stated by Haque et al. (2003), an uneven resolution along radial direction, fine to coarse in the positive radial direction can be attained by using the transformation as the following:

$$\eta = c \ln\left(1 + \frac{r}{r_0}\right) \tag{8}$$

where c = scale factor

r_0 = constant of the order of total radial distance

Following by the transformations, the relationship between Δr and $\Delta\eta$ can be achieved as below:

$$\Delta r = \frac{r + r_0}{c} \Delta\eta \tag{9}$$

According to this relation, a variable Δr can be obtained by fix the value of $\Delta\eta$. By using a constant value of $\Delta\eta$, Δr can be increased by increase the value of r. By this way, the uneven resolution (fine to coarse) in the radial direction in the physical domain can be attained while in computational domain the resolution remains uniform. The Jacobian of the given transformation is as follows :

$$J = \begin{vmatrix} \frac{\partial r}{\partial \eta} & \frac{\partial \theta}{\partial \eta} \\ \frac{\partial r}{\partial \theta} & \frac{\partial \theta}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \frac{r_0}{c} e^{\eta/c} & 0 \\ 0 & 1 \end{vmatrix} = \frac{r_0}{c} e^{\eta/c} \neq 0 \tag{10}$$

and the operator for the derivative is given by

$$\frac{\partial}{\partial r} \equiv \frac{c e^{-\eta/c}}{r_0} \frac{\partial}{\partial \eta} \tag{11}$$

By using the transformation (8), the equations (1)-(3) can be transformed into the equations as follows :

$$\frac{\partial \zeta}{\partial t} + \frac{ce^{-\eta/c}}{r_0 (e^{\eta/c} - 1)} \frac{\partial}{\partial \eta} \left\{ (\zeta + h)\, (e^{\eta/c} - 1) v_r \right\} + \frac{1}{r_0 (e^{\eta/c} - 1)} \frac{\partial}{\partial \theta} \left\{ (\zeta + h) v_\theta \right\} = 0 \tag{12}$$

$$\frac{\partial v_r}{\partial t} + v_r \frac{ce^{-\eta/c}}{r_0} \frac{\partial v_r}{\partial \eta} + \frac{v_\theta}{r_0 (e^{\eta/c} - 1)} \frac{\partial v_r}{\partial \theta} - f v_\theta = -\frac{gce^{-\eta/c}}{r_0} \frac{\partial \zeta}{\partial \eta} - \frac{C_f v_r \sqrt{v_r^2 + v_\theta^2}}{(\zeta + h)} \tag{13}$$

$$\frac{\partial v_\theta}{\partial t} + v_r \frac{ce^{-\eta/c}}{r_0} \frac{\partial v_\theta}{\partial \eta} + \frac{v_\theta}{r_0 (e^{\eta/c} - 1)} \frac{\partial v_\theta}{\partial \theta} + f v_r = -\frac{g}{r_0 (e^{\eta/c} - 1)} \frac{\partial \zeta}{\partial \theta} - \frac{C_f v_\theta \sqrt{v_r^2 + v_\theta^2}}{(\zeta + h)} \tag{14}$$

The boundary conditions (5) – (7) remain the same under this transformation.

2.5 Grid Generation and Numerical Scheme

2.5.1 Grid Generation

In this dissertation, the polar coordinate grid system is generated through the intersection of a set of straight lines, given by θ = *constant* through the Pole O (5° 22.5′ N, 100° 30′ E) and concentric circles, with centre as O, given by r = *constant*. The angle between any two consecutive straight gridlines through O is $\Delta\theta$ and the space between any two consecutive circular gridlines is Δr, which increases in the positive radial direction.

After the transformation (8), both $\Delta\theta$ and $\Delta\eta$ become uniform. The discrete grid points (η_i, θ_j) in the transformed domain are given by

$$\eta_i = (i-1)\Delta\eta, \qquad i = 1, 2, 3, \ldots, M \tag{15a}$$

$$\theta_j = (j-1)\Delta\theta, \qquad j = 1, 2, 3, \ldots, N \tag{15b}$$

where $\Delta\eta$ is a constant whereas Δr increases with the increase of r according to the equation (9).

The sequence of discrete time instants is given by

$$t_k = k\ \Delta t, \qquad k = 1, 2, 3, \ldots \tag{16}$$

In the physical domain N gridlines meet at the Pole; however, in the computation of domain this point is considered as N distinct grid points (it is generated automatically). There will be no computation due to the pole is set at the land. Thus, there also would not have any problem of instability during numerical computation (Roy et al., 2007).

2.5.2 Numerical Scheme

A staggered leap-frog scheme (central in both time and space) has been used to discretize the equations (12)-(14) and the boundary conditions (5)-(7). These equations are solved by an explicit method using staggered grid. The normal components of the velocity along the closed boundaries are taken as zero. To ensure the CFL stability criterion of the numerical scheme, the time step is taken as 5 seconds. Throughout the physical domain, the values of the friction coefficient are taken uniformly (Roy et al., 2007).

CHAPTER 3

NUMERICAL METHODS

3.1 Finite Difference Approximations

3.1.1 The Finite Difference Grid

Let $U = U(x_i, y_j)$ denote the value of U at the point (x_i, y_j). The solution domain in x-y space is covered by a *rectangular grid*, with grid spacing of Δx and Δy being assumed uniform. As shown in Figure 3.1, the first quadrant of the x-y plane is divided into uniform rectangles by grid lines parallel to the x-axis given by $x = x_i$ where $x_i = i\Delta x$, and grid lines parallel to the y-axis given by $y = y_j$ where $y_j = j\Delta y$.

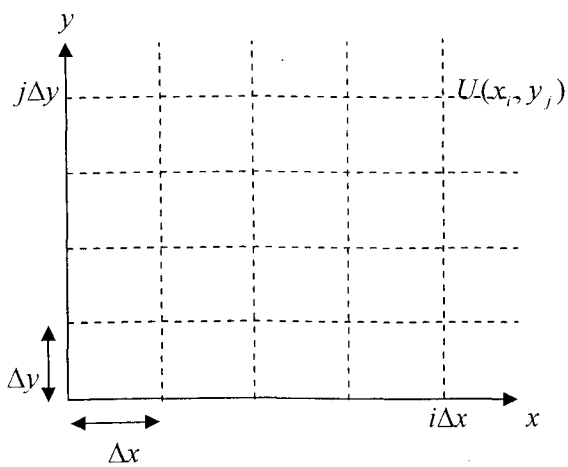

Fig. 3.1: The finite difference grid in the solution region.

Finite difference methods will now be developed which determine approximately the values of U at interior points of the solution domain where the lines of $x_i=i\Delta x$ and $y_j=j\Delta y$ intersect. For the purpose of this study, grid points within the solution domain are termed interior.

3.1.2 Finite Difference Approximation to Derivatives

To develop a method of calculating the value of U at each *interior* grid point (the x and y derivatives of U at (i,j)), grid points must be expressed in terms of values of U at nearby grid points. The most popular way to generate these approximations is through the use of Taylor series.

The Taylor series expansion of U about (x_i, y_j) is :

$$U(x_i + \Delta x, y_j) = \sum_{m=0}^{\infty} \frac{(\Delta x)^m}{m!} \frac{\partial^m U(x_i, y_j)}{\partial x^m}$$

or

$$U(x_i + \Delta x, y_j) =$$

$$U(x_i, y_j) + \frac{\Delta x}{1!}\frac{\partial U}{\partial x}(x_i, y_j) + \frac{\Delta x^2}{2!}\frac{\partial^2 U}{\partial x^2}(x_i, y_j) + \frac{\Delta x^3}{3!}\frac{\partial^3 U}{\partial x^3}(x_i, y_j) + \frac{\Delta x^4}{4!}\frac{\partial^4 U}{\partial x^4}(x_i, y_j) + \ldots \tag{17}$$

Suppose the series on the right hand side of (17) beginning with the third term is truncated. If Δx is sufficiently small, the fourth and higher terms can be assumed are much smaller than the third term. Therefore, it can be rewritten as :

$$U(x_i + \Delta x, y_j) = U(x_i, y_j) + \frac{\Delta x}{1!}\frac{\partial U}{\partial x}(x_i, y_j) + O(\Delta x^2) \tag{18}$$

The term $O(\Delta x^2)$ means the sum of the truncated terms (*i.e.* the truncation error) is in absolute terms at most a constant multiple of Δx^2.

Divide (18) by Δx and rearrange to give

$$\frac{\partial U}{\partial x}(x_i, y_j) = \frac{U(x_i + \Delta x, y_j) - U(x_i, y_j)}{\Delta x} + O(\Delta x) \tag{19}$$

which yields the following approximation

$$\frac{\partial U}{\partial x}(x_i, y_j) \approx \frac{U(x_i + \Delta x, y_j) - U(x_i, y_j)}{\Delta x} \tag{20}$$

Notice that the right hand side of expression (20) uses the value $U(x_i + \Delta x, y_j)$ which is forward of $U(x_i, y_j)$. Thus it is said to be the *forward difference approximation* to $\frac{\partial U}{\partial x}$ at (x_i, y_j) and is said to be the first order accurate or $O(\Delta x)$ accurate (Noye, 1982).

Applying the Taylor series expansion for $U(x_i - \Delta x, y_j)$ about (x_i, y_j) leads to

$$U(x_i - \Delta x, y_j) =$$
$$U(x_i, y_j) - \frac{\Delta x}{1!}\frac{\partial U}{\partial x}(x_i, y_j) + \frac{\Delta x^2}{2!}\frac{\partial^2 U}{\partial x^2}(x_i, y_j) - \frac{\Delta x^3}{3!}\frac{\partial^3 U}{\partial x^3}(x_i, y_j) + \frac{\Delta x^4}{4!}\frac{\partial^4 U}{\partial x^4}(x_i, y_j) - \ldots \tag{21}$$

Again, the series on the right hand side of (21) beginning with the third term is decided to be truncated. If Δx is sufficiently small, as the previous assumption, the fourth and higher terms are much smaller than the third term. Thus,

$$U(x_i - \Delta x, y_j) = U(x_i, y_j) - \frac{\Delta x}{1!}\frac{\partial U}{\partial x}(x_i, y_j) + O(\Delta x^2) \tag{22}$$

Divide (22) by Δx and rearrange to yield

$$\frac{\partial U}{\partial x}(x_i, y_j) = \frac{U(x_i, y_j) - U(x_i - \Delta x, y_j)}{\Delta x} + O(\Delta x) \tag{23}$$

With the approximation is expressed as

$$\frac{\partial U}{\partial x}(x_i, y_j) \approx \frac{U(x_i, y_j) - U(x_i - \Delta x, y_j)}{\Delta x} \tag{24}$$

Observe that the right hand side expression of (24) uses the value $U(x_i - \Delta x, y_j)$ at a previous level to $U(x_i, y_j)$. Thus it is said to be the *backward difference approximation* to $\frac{\partial U}{\partial x}$ at (x_i, y_j) and is said to be the first order accurate or *O(Δx)* accurate (Noye, 1982).

By subtracting (21) from (17), it leads to

$$U(x_i + \Delta x, y_j) - U(x_i - \Delta x, y_j) = 2\Delta x \frac{\partial U}{\partial x}(x_i, y_j) + O(\Delta x^3) \tag{25}$$

Divide (25) with $2\Delta x$ and rearrange to give

$$\frac{\partial U}{\partial x}(x_i, y_j) = \frac{U(x_i + \Delta x, y_j) - U(x_i - \Delta x, y_j)}{2\Delta x} + O(\Delta x^2) \tag{26}$$

Which the approximation is

$$\frac{\partial U}{\partial x}(x_i, y_j) \approx \frac{U(x_i + \Delta x, y_j) - U(x_i - \Delta x, y_j)}{2\Delta x} \tag{27}$$

Because the $i\Delta x$ is centered between the levels $(x_i + \Delta x)$ and $(x_i - \Delta x)$ at which the values $U(x_i + \Delta x, y_j)$ and $U(x_i - \Delta x, y_j)$ occur, the expression on the right hand side of (27) is said to be the *central difference approximation* to $\frac{\partial U}{\partial x}$ at (x_i, y_j) and is said to be second order accurate or $O(\Delta x^2)$ accurate (Noye, 1982).

In similar way, finite difference approximation to the second order derivative can be derived.

By adding (17) and (21), it becomes

$$U(x_i + \Delta x, y_j) + U(x_i - \Delta x, y_j) = 2U(x_i, y_j) + \Delta x^2 \frac{\partial^2 U}{\partial x^2}(x_i, y_j) + O(\Delta x^4) \tag{28}$$

Divide (28) with Δx^2 and manipulating gives

$$\frac{\partial^2 U}{\partial x^2}(x_i, y_j) = \frac{U(x_i + \Delta x, y_j) - 2U(x_i, y_j) + U(x_i - \Delta x, y_j)}{\Delta x^2} + O(\Delta x^2) \tag{29}$$

with the approximation is expressed as

$$\frac{\partial^2 U}{\partial x^2}(x_i, y_j) \approx \frac{U(x_i + \Delta x, y_j) - 2U(x_i, y_j) + U(x_i - \Delta x, y_j)}{\Delta x^2} \tag{30}$$

The right hand side expression of (30) is said to be the *central difference approximation* to $\frac{\partial^2 U}{\partial x^2}$ at (x_i, y_j) and is said to be second order accurate or $O(\Delta x^2)$ accurate.

In similar manner as described earlier, by using the same multivariable Taylor's expansion, we can develop the finite difference approximations for U about y as :

$\frac{\partial U}{\partial y}(x_i, y_j) \approx \frac{U(x_i, y_j + \Delta y) - U(x_i, y_j)}{\Delta y}$ which is the *forward difference approximation* to $\frac{\partial U}{\partial y}$ at (x_i, y_j).

$\frac{\partial U}{\partial y}(x_i, y_j) \approx \frac{U(x_i, y_j) - U(x_i, y_j - \Delta y)}{\Delta y}$ which is the *backward difference approximation* to $\frac{\partial U}{\partial y}$ at (x_i, y_j).

$\frac{\partial U}{\partial y}(x_i, y_j) \approx \frac{U(x_i, y_j + \Delta y) - U(x_i, y_j - \Delta y)}{2\Delta y}$ which is the *central difference approximation* to $\frac{\partial U}{\partial y}$ at (x_i, y_j).

$\frac{\partial^2 U}{\partial y^2}(x_i, y_j) \approx \frac{U(x_i, y_j + \Delta y) - 2U(x_i, y_j) + U(x_i, y_j - \Delta y)}{\Delta y^2}$ which is the *central difference approximation* to $\frac{\partial^2 U}{\partial y^2}$ at (x_i, y_j).

Finite difference approximations can also be obtained for mixed derivatives ($\frac{\partial^2 U}{\partial x \partial y}$) but this will not be pursued here.

3.2 Discretization and Finite Difference Scheme in Cylindrical Polar Coordinate System

As stated by Roy at el. (2005), the governing equations are discretized by finite differences (forward in time and central in space) and can be solved by a conditionally stable semi-implicit method. The following notations is used for the purpose of discretization : For any dependent variable χ (r, θ, t),

$$\chi(r_i, \theta_j, t_k) = \chi_{ij}^k$$

$$\frac{1}{2}(\chi_{i+1j}^k + \chi_{i-1j}^k) = \overline{\chi_{ij}^k}^r$$

$$\frac{1}{2}(\chi_{ij+1}^k + \chi_{ij-1}^k) = \overline{\chi_{ij}^k}^\theta$$

$$\frac{1}{4}(\chi_{i+1j}^k + \chi_{i-1j}^k + \chi_{ij+1}^k + \chi_{ij-1}^k) = \overline{\chi_{ij}^k}^{r\theta}$$

Thus, the discretized form of equation (12) is

$$\frac{\zeta_{ij}^{k+1}-\zeta_{ij}^{k}}{\Delta t}+\frac{ce^{-\frac{\eta_i}{c}}}{r_0(e^{\frac{\eta_i}{c}}-1)}\left[\frac{(\zeta_{i+1j}^{k}+h_{i+1j})(e^{\frac{\eta_{i+1}}{c}}-1)v_{r(i+1)j}^{k}-(\zeta_{i-1j}^{k}+h_{i-1j})(e^{\frac{\eta_{i-1}}{c}}-1)v_{r(i-1)j}^{k}}{2\Delta\eta}\right]$$
$$+\frac{1}{r_0(e^{\frac{\eta_i}{c}}-1)}\left[\frac{(\zeta_{ij+1}^{k}+h_{ij+1})v_{\theta(i,j+1)}^{k}-(\zeta_{ij-1}^{k}+h_{ij-1})v_{\theta(i,j-1)}^{k}}{2\Delta\theta}\right]=0$$

from which ζ_{ij}^{k+1} is computed for i = 2, 4, 6, ... , M-2 and j = 3, 5, 7, ..., N-2. The boundary condition (5) is discretized as

$$v_{r(M-1)j}^{k}+(g/h_{M-1j})^{1/2}\frac{1}{2}(\zeta_{M-2j}^{k+1}+\zeta_{Mj}^{k+1})=0 \tag{31}$$

from which ζ_{Mj}^{k+1} is computed for j = 1, 3, 5, ..., N. The boundary condition (6) is discretized as

$$v_{\theta(i,2)}^{k}+(g/h_{i2})^{1/2}\frac{1}{2}(\zeta_{i1}^{k+1}+\zeta_{i3}^{k+1})=0 \tag{32}$$

from which ζ_{i1}^{k+1} is computed for i = 2, 4, 6, ..., M-2. The boundary condition (7) is discretized as

$$v_{\theta(i,N-1)}^{k}-(g/h_{iN-1})^{1/2}\frac{1}{2}(\zeta_{iN-2}^{k+1}+\zeta_{iN}^{k+1})=0 \tag{33}$$

from which ζ_{iN}^{k+1} is computed for i = 2, 4, 6, ..., M-2. The discretized form of equation (13) is

$$\frac{v_{r(i,j)}^{k+1}-v_{r(i,j)}^{k}}{\Delta t}+\overline{v_{r(i,j)}^{k}}^{\eta}\frac{ce^{-\frac{\eta_i}{c}}}{r_0}\left[\frac{v_{r(i+1,j)}^{k}-v_{r(i-1,j)}^{k}}{2\Delta\eta}\right]+\frac{\overline{v_{\theta(i,j)}^{k}}^{\theta}}{r_0(e^{\frac{\eta_i}{c}}-1)}\left[\frac{v_{r(i,j+1)}^{k}-v_{r(i,j-1)}^{k}}{2\Delta\theta}\right]-f\overline{v_{\theta(i,j)}^{k}}^{\eta\theta}$$
$$=-g\frac{ce^{-\frac{\eta_i}{c}}}{r_0}\left(\frac{\zeta_{i+1j}^{k+1}-\zeta_{i-1j}^{k+1}}{2\Delta\eta}\right)-\frac{C_f v_{r(i,j)}^{k+1}((v_{r(i,j)}^{k})^2+(\overline{v_{\theta(i,j)}^{k}}^{\eta\theta})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{\eta}+h_{ij}} \tag{34}$$

from which $v_{r(i,j)}^{k+1}$ is computed for i = 3, 5, 7, ..., M -1 and j = 3, 5, 7, ..., N - 2. Note that in the last term $v_{r(i,j)}^{k+1}$ is in advanced time level and this ensures a semi-implicit nature of the numerical method. Similarly, the discretized form of equation (14) is

$$\frac{v_{\theta(i,j)}^{k+1}-v_{\theta(i,j)}^{k}}{\Delta t}+\overline{v_{r(i,j)}^{k}}^{\eta}\frac{ce^{-\frac{\eta_i}{c}}}{r_0}[\frac{v_{\theta(i+1)j}^{k}-v_{\theta(i-1)j}^{k}}{2\Delta\eta}]+\frac{\overline{v_{\theta(i,j)}^{k}}^{\theta}}{r_0(e^{\frac{\eta_i}{c}}-1)}[\frac{v_{\theta(i,j+1)}^{k}-v_{\theta(i,j-1)}^{k}}{2\Delta\theta}]+f\overline{v_{r(i,j)}^{k}}^{\eta\theta}$$

$$=-\frac{g}{r_0(e^{\frac{\eta_i}{c}}-1)}(\frac{\zeta_{ij-1}^{k+1}-\zeta_{ij-1}^{k+1}}{2\Delta\theta})-\frac{C_f v_{\theta(i,j)}^{k+1}(\overline{(v_{\theta(i,j)}^{k})^{\eta\theta}})^2+(v_{\theta(i,j)}^{k})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{\theta}+h_{ij}} \tag{35}$$

from which $v_{\theta(i,j)}^{k+1}$ is computed for i = 2, 4, 6, ..., M - 2 and j = 2, 4, 6, ..., N - 1. As before, in the last term $v_{\theta(i,j)}^{k+1}$ is in advanced time level and this ensures a semi-implicit nature of the numerical method.

3.3 The Staggered Leapfrog Scheme

3.3.1 Leapfrog Scheme

The leapfrog scheme uses centered differences in both time and space. Let consider the linear wave equation with unity wave speed, i.e.:

$$\frac{\partial T}{\partial t}+\frac{\partial T}{\partial x}=0$$

The finite difference approximation of this scheme for the given wave equation is:

$$\frac{T_i^{n+1}-T_i^{n-1}}{2\Delta t}+\frac{T_{i+1}^{n}-T_{i-1}^{n}}{2\Delta x}=0$$

This is an explicit scheme, which requires data at two different time levels to find T at the next level.

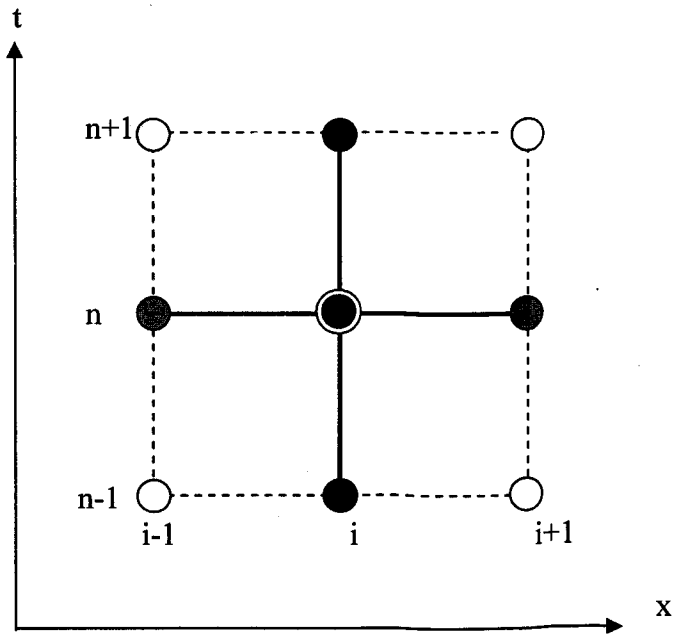

Fig. 3.2: The computation grid for leapfrog scheme

By using the leapfrog method, the discretized form of equation (12) is :

$$\frac{\zeta_{ij}^{k+1}-\zeta_{ij}^{k-1}}{2\Delta t}+\frac{ce^{-\frac{\eta_i}{c}}}{r_0(e^{\frac{\eta_i}{c}}-1)}\left[\frac{(\zeta_{i+1j}^{k}+h_{i+1j})(e^{\frac{\eta_{i+1}}{c}}-1)v_{r(i+1)j}^{k}-(\zeta_{i-1j}^{k}+h_{i-1j})(e^{\frac{\eta_{i-1}}{c}}-1)v_{r(i-1)j}^{k}}{2\Delta\eta}\right]$$
$$+\frac{1}{r_0(e^{\frac{\eta_i}{c}}-1)}\left[\frac{(\zeta_{ij+1}^{k}+h_{ij+1})v_{\theta(i,j+1)}^{k}-(\zeta_{ij-1}^{k}+h_{ij-1})v_{\theta(i,j-1)}^{k}}{2\Delta\theta}\right]=0$$

Manipulate $2\Delta t$ and rearranging it, gives :

$$\zeta_{ij}^{k+1}=\zeta_{ij}^{k-1}-\frac{\Delta t}{r_0(e^{\frac{\eta_i}{c}}-1)}\left\{ce^{-\frac{\eta_i}{c}}\left[\frac{(\zeta_{i+1j}^{k}+h_{i+1j})(e^{\frac{\eta_{i+1}}{c}}-1)v_{r(i+1)j}^{k}-(\zeta_{i-1j}^{k}+h_{i-1j})(e^{\frac{\eta_{i-1}}{c}}-1)v_{r(i-1)j}^{k}}{\Delta\eta}\right]\right.$$
$$\left.+\frac{(\zeta_{ij+1}^{k}+h_{ij+1})v_{\theta(i,j+1)}^{k}-(\zeta_{ij-1}^{k}+h_{ij-1})v_{\theta(i,j-1)}^{k}}{\Delta\theta}\right\} \qquad (36)$$

From equation (13), the leapfrog method gives :

$$\frac{v_{r(i,j)}^{k+1}-v_{r(i,j)}^{k-1}}{2\Delta t}+\overline{v_{r(i,j)}^{k}}^{\eta}\frac{ce^{-\frac{\eta_i}{c}}}{r_0}[\frac{v_{r(i+1,j)}^{k}-v_{r(i-1,j)}^{k}}{2\Delta\eta}]+\frac{\overline{v_{\theta(i,j)}^{k}}^{\theta}}{r_0(e^{\frac{\eta_i}{c}}-1)}[\frac{v_{r(i,j+1)}^{k}-v_{r(i,j-1)}^{k}}{2\Delta\theta}]-f\overline{v_{\theta(i,j)}^{k}}^{\eta\theta}$$

$$=-g\frac{ce^{-\frac{\eta_i}{c}}}{r_0}(\frac{\zeta_{i+1j}^{k+1}-\zeta_{i-1j}^{k+1}}{2\Delta\eta})-\frac{C_f v_{r(i,j)}^{k+1}((v_{r(i,j)}^{k})^2+(\overline{v_{\theta(i,j)}^{k}}^{\eta\theta})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{\eta}+h_{ij}}$$

Manipulate $2\Delta t$ and rearranging it, it gives :

$$v_{r(i,j)}^{k+1}(1+2\Delta t[\frac{C_f((v_{r(i+1,j)}^{k})^2+(\overline{v_{\theta(i,j)}^{k}}^{\eta\theta})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{\eta}+h_{ij}})=v_{r(i,j)}^{k-1}+2\Delta t\{-\overline{v_{r(i,j)}^{k}}^{\eta}\frac{ce^{-\frac{\eta_i}{c}}}{r_0}[\frac{v_{r(i+1,j)}^{k}-v_{r(i-1,j)}^{k}}{2\Delta\eta}]$$

$$-\frac{\overline{v_{\theta(i,j)}^{k}}^{\theta}}{r_0(e^{\frac{\eta_i}{c}}-1)}[\frac{v_{r(i,j+1)}^{k}-v_{r(i,j-1)}^{k}}{2\Delta\theta}]+f\overline{v_{\theta(i,j)}^{k}}^{\eta\theta}+g\frac{ce^{-\frac{\eta_i}{c}}}{r_0}(\frac{\zeta_{i+1j}^{k+1}-\zeta_{i-1j}^{k+1}}{2\Delta\eta})\}$$

Divide $1+2\Delta t[\frac{C_f((v_{r(i,j)}^{k})^2+(\overline{v_{\theta(i,j)}^{k}}^{\eta\theta})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{x}+h_{ij}}]$, it leads to

$$v_{r(i,j)}^{k+1}=\frac{v_{r(i,j)}^{k-1}+2\Delta t\{-\overline{v_{r(i,j)}^{k}}^{\eta}\frac{ce^{-\frac{\eta_i}{c}}}{r_0}[\frac{v_{r(i+1,j)}^{k}-v_{r(i-1,j)}^{k}}{2\Delta\eta}]-\frac{\overline{v_{\theta(i,j)}^{k}}^{\theta}}{r_0(e^{\frac{\eta_i}{c}}-1)}[\frac{v_{r(i,j+1)}^{k}-v_{r(i,j-1)}^{k}}{2\Delta\theta}]+f\overline{v_{\theta(i,j)}^{k}}^{\eta\theta}+g\frac{ce^{-\frac{\eta_i}{c}}}{r_0}(\frac{\zeta_{i+1j}^{k+1}-\zeta_{i-1j}^{k+1}}{2\Delta\eta})\}}{1+2\Delta t\frac{C_f((v_{r(i,j)}^{k})^2+(\overline{v_{\theta(i,j)}^{k}}^{\eta\theta})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{\eta}+h_{ij}}} \quad (37)$$

Using Leapfrog method, the equation (14) becomes :

$$\frac{v_{\theta(i,j)}^{k+1}-v_{\theta(i,j)}^{k+1}}{2\Delta t}+\overline{v_{r(i,j)}^{k}}^{\eta}\frac{ce^{-\frac{\eta_i}{c}}}{r_0}[\frac{v_{\theta(i+1)j}^{k}-v_{\theta(i-1)j}^{k}}{2\Delta\eta}]+\frac{\overline{v_{\theta(i,j)}^{k}}^{\theta}}{r_0(e^{\frac{\eta_i}{c}}-1)}[\frac{v_{\theta(i,j+1)}^{k}-v_{\theta(i,j-1)}^{k}}{2\Delta\theta}]+f\overline{v_{r(i,j)}^{k}}^{\eta\theta}$$

$$=-\frac{g}{r_0(e^{\frac{\eta_i}{c}}-1)}(\frac{\zeta_{ij-1}^{k+1}-\zeta_{ij-1}^{k+1}}{2\Delta\theta})-\frac{C_f v_{\theta(i,j)}^{k+1}((\overline{v_{\theta(i,j)}^{k}}^{\eta\theta})^2+(v_{\theta(i,j)}^{k})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{\theta}+h_{ij}}$$

Manipulate $2\Delta t$ and rearranging it, it gives :

$$v_{\theta(i,j)}^{k+1}(1+2\Delta t[\frac{C_f v_{\theta(i,j)}^{k+1}(\overline{(v_{\theta(i,j)}^{k})^{\eta\theta}})^2+(v_{\theta(i,j)}^{k})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{\theta}+h_{ij}}])$$

$$=v_{\theta(i,j)}^{k-1}-2\Delta t\{\overline{v_{r(i,j)}^{k}}^{\eta}\frac{ce^{-\frac{\eta_i}{c}}}{r_0}[\frac{v_{\theta(i+1)j}^{k}-v_{\theta(i-1)j}^{k}}{2\Delta\eta}]+\frac{\overline{v_{\theta(i,j)}^{k}}^{\theta}}{r_0(e^{\frac{\eta_i}{c}}-1)}[\frac{v_{\theta(i,j+1)}^{k}-v_{\theta(i,j-1)}^{k}}{2\Delta\theta}]+f\overline{v_{r(i,j)}^{k}}^{\eta\theta}+\frac{g}{r_0(e^{\frac{\eta_i}{c}}-1)}(\frac{\zeta_{ij-1}^{k+1}-\zeta_{ij-1}^{k+1}}{2\Delta\theta})\}$$

Divide $1+2\Delta t[\frac{C_f(\overline{(v_{r(i,j)}^{k})^{\eta\theta}})^2+(v_{\theta(i,j)}^{k})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{\theta}+h_{ij}}]$, becomes :

$$v_{\theta(i,j)}^{k+1}=\frac{v_{\theta(i,j)}^{k-1}-2\Delta t\{\overline{v_{r(i,j)}^{k}}^{\eta}\frac{ce^{-\frac{\eta_i}{c}}}{r_0}[\frac{v_{\theta(i+1)j}^{k}-v_{\theta(i-1)j}^{k}}{2\Delta\eta}]+\frac{\overline{v_{\theta(i,j)}^{k}}^{\theta}}{r_0(e^{\frac{\eta_i}{c}}-1)}[\frac{v_{\theta(i,j+1)}^{k}-v_{\theta(i,j-1)}^{k}}{2\Delta\theta}]+f\overline{v_{r(i,j)}^{k}}^{\eta\theta}+\frac{g}{r_0(e^{\frac{\eta_i}{c}}-1)}(\frac{\zeta_{ij-1}^{k+1}-\zeta_{ij-1}^{k+1}}{2\Delta\theta})\}}{1+2\Delta t[\frac{C_f(\overline{(v_{r(i,j)}^{k})^{\eta\theta}})^2+(v_{\theta(i,j)}^{k})^2)^{1/2}}{\overline{\zeta_{ij}^{k+1}}^{\theta}+h_{ij}}]} \qquad (38)$$

It can be seen that the solution $\zeta_{ij}^{k+1}, v_{r(i,j)}^{k+1}$ and $v_{\theta(i,j)}^{k+1}$ given by equation (36), (37), and (38) leaps over the solution at $\zeta_{ij}^{k}, v_{r(i,j)}^{k}$ and $v_{\theta(i,j)}^{k}$; hence, the descriptive name : leapfrog method. Leapfrog method also called a multi-step method because it uses solutions from two previous time steps (steps *k* and *k-1*) to compute the solution at the new time step (step *k+1*). In order to start the method, one must use a one-step method to approximate the solution at t_1, given the initial data at time $t_0 = 0$. Once the approximate solution is known at two consecutive time steps, formula (36), (37), and (38) can then be used to advance to future time steps.

Leapfrog scheme is used to evaluate centered differences and is said to be second order accurate in time and space or with a truncation error of $O(\Delta\eta^2, \Delta\theta^2, \Delta t^2)$. Whereas, the local truncation error for the forward time centered