# A Moving Mesh Finite Element Method for the Shallow Water Equations

David Skinner

August 19, 2007

## Abstract

In this dissertation a moving mesh method finite element method is used to approximate moving boundary solutions to the shallow water equations. An Arbitrary Lagrangian Eulerian method is applied to an existing finite element scheme. Some exact solutions to the shallow water equations in a parabolic basin are shown for comparison. An investigation is conducted into the accuracy of the method. It is concluded that further work is required to address the build up of numerical errors around the boundary.

## Acknowledgements

I would like to thank Mike Baines and Emmanuel Hanert for the assistance thay have provided as joint supervisors during this dissertation. I would also like to thank the Natural Environmental Research Council for providing the financial assistance that made it possible to complete the programme of study.

## Declaration

I confirm that this work is my own and the use of all other material from other sources has been properly and fully acknowledged.

David Skinner

# Contents

**1 Introduction** **1**

**2 Analytic Solutions to the Shallow Water Equations** **3**

2.1 Shallow water equations . . . . . . . . . . . . . . . . 3

2.2 General solution . . . . . . . . . . . . . . . . . . . . . . 5

2.3 Planar Surface . . . . . . . . . . . . . . . . . . . . . . . 8

2.4 Curved Surface . . . . . . . . . . . . . . . . . . . . . . 10

2.5 Numerical integration of the ODE system . . . . . 14

**3 A Finite element method** **17**

3.1 Background . . . . . . . . . . . . . . . . . . . . . . . . 17

3.2 Eulerian formulation . . . . . . . . . . . . . . . . . . . 17

3.3 Nondimensionalization . . . . . . . . . . . . . . . . . 18

3.4 Finite element discretization . . . . . . . . . . . . . 20

3.4.1 The non-conforming mixed $P_1^{NC}$-$P_1$ discretization . . . . . . . . . . . . . . . . . . . . . . . 22

3.4.2 Mass lumping . . . . . . . . . . . . . . . . . . 24

3.5 Time discretization . . . . . . . . . . . . . . . . . . . 25

**4 A Moving mesh method 27**

4.1 Background . . . . . . . . . . . . . . . . . . . . . . . 27

4.2 ALE method . . . . . . . . . . . . . . . . . . . . . . . 28

4.3 Derivation of mesh velocities . . . . . . . . . . . . . . 29

4.3.1 Integral form in a moving frame . . . . . . . 29

4.3.2 Introduction of monitor function . . . . . . 30

4.3.3 Weak form . . . . . . . . . . . . . . . . . . . . . 30

4.3.4 Introduction of velocity potential . . . . . . 31

4.3.5 Finite element Discretization . . . . . . . . . 32

4.3.6 Stiffness matrix assembly . . . . . . . . . . . 33

4.3.7 Solving for Q . . . . . . . . . . . . . . . . . . . 35

4.4 Moving the mesh . . . . . . . . . . . . . . . . . . . . . 36

4.5 Equations in the moving frame . . . . . . . . . . . . 36

**5 Numerical Results 38**

5.1 Planar surface . . . . . . . . . . . . . . . . . . . . . . . 38

5.2 Curved surface . . . . . . . . . . . . . . . . . . . . . . . 41

5.3 Error analysis . . . . . . . . . . . . . . . . . . . . . . . 43

**6 Investigation 45**

6.1 Errors at the boundary . . . . . . . . . . . . . . . . . 45

6.2 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . 49

**7 Summary and further work 51**

7.1 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . 51

7.2 Further investigation of boundary errors . . . . . . 52

7.3 Comparison with other solutions . . . . . . . . . . . 52

7.4 Adding vorticity to the mesh movement . . . . . . . 52

# List of Figures

2.1 Depth profile showing $h$ and $\eta$. The total depth is $H = h + \eta$. . . . . . . . . . . . . . . . . . . . . . . . 5

2.2 An exact planar surface solution at $t = 0$s. . . . . . . 11

2.3 An exact planar surface solution at $t = 4000$s. . . . 11

2.4 An exact curved surface solution at $t = 0$s. . . . . . 13

2.5 An exact curved surface solution at $t = 3500$s. . . . 13

3.1 The $P_1$ and $P_1^{NC}$ shape functions . . . . . . . . . . 23

5.1 A numerical planar surface solution at $t = 0$s. . . . 39

5.2 A numerical planar surface solution at $t = 2000$s. . 39

5.3 A numerical planar surface solution at $t = 3000$s. . 40

5.4 A numerical planar surface solution at $t = 4000$s. . 40

5.5 A numerical curved surface solution at $t = 0$s and $t = 2000$s. . . . . . . . . . . . . . . . . . . . . . . . 41

5.6 A numerical curved surface solution at $t = 4000$s. . 42

5.7 A numerical curved surface solution at $t = 6000$s. . 42

5.8 An error plot for $\Delta t = 10$s . . . . . . . . . . . . . . . 44

5.9 An error plot for $\Delta t = 5$s . . . . . . . . . . . . . . . . 44

6.1 Mesh velocities at $t = \Delta t$ . . . . . . . . . . . . . . . . 46

6.2 Physical velocities at $t = \Delta t$ . . . . . . . . . . . . . . 46

6.3 Physical velocities at $t = \Delta t$ on rotated mesh . . . . 48

6.4 Physical velocities at $t = \Delta t$ with advection terms removed . . . . . . . . . . . . . . . . . . . . . . . . . . 48

6.5 Boundary elements . . . . . . . . . . . . . . . . . . . . 50

# Chapter 1

# Introduction

Fluid dynamics is a field in which moving boundary problems are an integral part. Transient interfaces between fluids and solid structures are commonplace. Geophysical and environmental fluid dynamics, in particular, involve many interesting moving boundary problems, such as the modelling of flood plains, and tidal flows in coastal regions. This project seeks to use an Arbtitrary Lagrangian-Eulerian(ALE) finite element method to approximate moving boundary solutions to the shallow water equations.

Due to the nonlinearity of even the simplest formulation of the equations for fluid motion, exact solutions to problems with moving boundaries are rare. In chapter 2, some exact solutions to the shallow water equations, proposed by W.C. Thacker [1], are introduced. The general solution for nonlinear normal mode oscillatory motion in a parabolic basin is presented. Specific solutions for two sets of initial conditions are then derived. In the first case, a solution for which the surface of the fluid remains planar is shown. The second case is for a solution in which the surface of the fluid is parabolic. A 4th order Runge Kutta scheme is also presented for the numerical integration of the system of ordinary differential equations derived for the general case. These solutions are used to compare with the results the ALE finite element method. Chapter 3 introduces a finite element method, Le Roux et al. [3], for the shallow water equations. The shallow water equations are presented in nondimensional form. The finite element discretization of the equations is discussed. Chapter 4 introduces the ALE moving mesh method that is used to approximate the moving boundary problems outlined in chapter 2. The method used to derive the mesh velocities is shown and the implementation within the existing finite element model is discussed. The results obtained using the ALE method are presented in chapter 5, they are compared with the results from the exact solutions in chapter 2. Chapter 6 is an investigation into the results obtained in chapter 5, in particular, the cause of the large numerical errors is investigated. Finally, chapter 7 provides a summary of the work carried out and discusses further work that could be pursued.

# Chapter 2

# Analytic Solutions to the Shallow Water Equations

## 2.1 Shallow water equations

The shallow water equations are a simplification of the full equations of motion for an incompressible fluid. The key assumption is that the flow is homogeneous along the vertical axis. The hydrostatic balance equation (2.1) states that gravity balances the pressure gradient in the vertical direction.

$$\frac{\partial p}{\partial z} = -\rho g \tag{2.1}$$

where $p$ is pressure of the fluid, $z$ is the vertical axis, $\rho$ is the density of the fluid and $g$ is the gravitational acceleration. If $\rho$ is assumed to be constant then it follows that the horizontal pressure gradient is independent of $z$. The horizontal flow is assumed to be independent of $z$, so incompressibility implies that the vertical velocity is linear in $z$.

As a result of the assumptions made, the shallow water equations are only accurate if the horizontal scale is much greater than the vertical scale. They can be used to describe the motion of the atmosphere or the oceans and can represent many types of motion including Rossby waves and gravity waves.

The Eulerian formulation of the nonlinear shallow water equations are as follows

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv + g\frac{\partial \eta}{\partial x} = 0 \tag{2.2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu + g\frac{\partial \eta}{\partial y} = 0 \tag{2.3}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}\left[u\left(h+\eta\right)\right] + \frac{\partial}{\partial y}\left[v\left(h+\eta\right)\right] = 0 \tag{2.4}$$

where $u$ and $v$ are the horizontal velocity components corresponding to the orthogonal directions $x$ and $y$. $h$ is the depth of the fluid when it is in equilibrium. $\eta$ is the displacement of the fluid from its equilibrium state, which is positive above, and negative below, the equilibrium level, so that $h+\eta$ is equal to the total depth of the fluid, $H$. The shoreline is therefore given by $H = 0$. Figure 2.1 shows the depth profile. $f$ is the Coriolis parameter. The terms involving $f$ represent the effects of the Coriolis 'force'. This is a fictitious force, as opposed to a physical force, that appears when the equations of fluid motion are cast in a rotating frame of reference. In equations (2.2) and (2.3) they account for the rotation of the earth.

The assumptions that Thacker makes require that the basin is an elliptical parabaloid given by

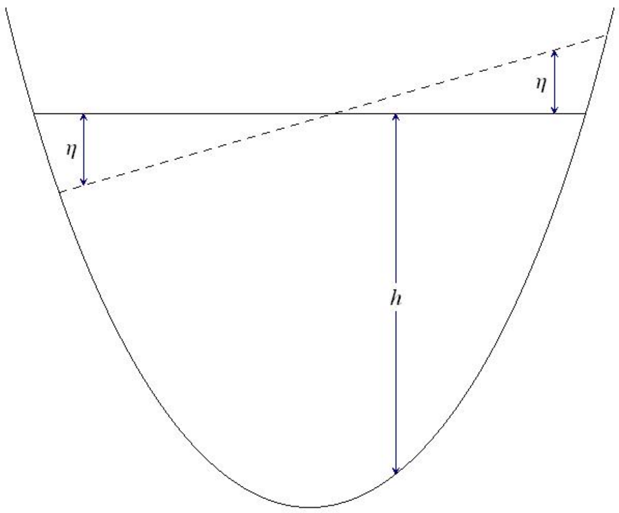

Figure 2.1: Depth profile showing $h$ and $\eta$. The total depth is $H = h + \eta$.

$$h = h_0 \left(1 - \frac{x^2}{L^2} - \frac{y^2}{l^2}\right) \tag{2.5}$$

where $h_0$ is the equilibrium depth at the centre of the basin. In this project, only the situation where $L = l$ is considered.

## 2.2 General solution

Thacker uses a similar approach to previous work done by F.K. Ball [2] in that assumptions are made about the nature of the motion and then solved for in a basin for which that motion should be possible. Ball formulates the problem in Lagrangian terms, whereas Thacker formulates it in Eulerian terms. Both approaches require the motion to occur in a parabolic basin and obtain similar results.

Assume that there are solutions for $u$ and $v$ of the form,

$$u = u_0 + u_1 x + u_2 y \tag{2.6}$$

$$v = v_0 + v_1 x + v_2 y \tag{2.7}$$

where $u_0$, $v_0$, $u_1$, $v_1$, $u_2$ and $v_2$ are functions only of time. This restricts the possible surface shape to planar or parabolic. (Thacker made attempts, that were not described in [1], to find solutions with surfaces described by polynomials of a higher degree but obtained overdetermined sets of equations. Thacker hypothesised that, 'there may not be any polynomial solutions of degree higher than second', [1].) When the assumptions, (2.6) and (2.7), are applied to the momentum equations, (2.2) and (2.3), it requires, for exact solutions, that the elevation solution has the form

$$\eta = \eta_0 + \eta_1 x + \eta_2 y + \frac{1}{2}\eta_{11}x^2 + \frac{1}{2}\eta_{22}y^2 + \frac{1}{2}\left(\eta_{12} + \eta_{21}\right)xy \tag{2.8}$$

where

$$\eta_1 = -\frac{1}{g}\left[\frac{du_0}{dt} + u_0 u_1 + v_0 u_2 - f v_0\right] \tag{2.9}$$

$$\eta_2 = -\frac{1}{g}\left[\frac{dv_0}{dt} + u_0 v_1 + v_0 v_2 + f u_0\right] \tag{2.10}$$

$$\eta_{11} = -\frac{1}{g}\left[\frac{du_1}{dt} + u_1^2 + u_2 v_1 - f v_1\right] \tag{2.11}$$

$$\eta_{22} = -\frac{1}{g}\left[\frac{dv_2}{dt} + v_2^2 + u_2 v_1 + f u_2\right] \tag{2.12}$$

$$\eta_{12} = -\frac{1}{g}\left[\frac{du_2}{dt} + u_1u_2 + u_2v_2 - fv_2\right] \tag{2.13}$$

$$\eta_{21} = -\frac{1}{g}\left[\frac{dv_1}{dt} + u_1v_1 + v_1v_2 + fu_1\right] \tag{2.14}$$

where $\eta_{12} = \eta_{21}$ and $\eta_0$ is a function of $t$.

If equations (2.5)-(2.14) are substituted into the continuity equation (2.4) and the spatial derivatives evaluated, then the time varying coefficients of the linearly independent terms must separately disappear. $u_0$, $v_0$, $u_1$, $v_1$, $u_2$, $v_2$ and $h_0$ must, therefore, satisfy the following equations

$$\frac{d\eta_0}{dt} + (u_1 + v_2)(h_0 + \eta_0) + u_0\eta_1 + v_0\eta_2 = 0 \tag{2.15}$$

$$\frac{d\eta_1}{dt} + (2u_1 + v_2)\eta_1 + v_1\eta_2 + u_0\left(\eta_{11} - \frac{2h_0}{L^2}\right) + v_0\eta_{12} = 0 \tag{2.16}$$

$$\frac{d\eta_2}{dt} + (2v_2 + u_1)\eta_2 + u_2\eta_1 + v_0\left(\eta_{22} - \frac{2h_0}{l^2}\right) + u_0\eta_{12} = 0 \tag{2.17}$$

$$\frac{d\eta_{11}}{dt} + (3u_1 + v_2)\left(\eta_{11} - \frac{2h_0}{L^2}\right) + 2v_1\eta_{12} = 0 \tag{2.18}$$

$$\frac{d\eta_{22}}{dt} + (3v_2 + u_1)\left(\eta_{22} - \frac{2h_0}{l^2}\right) + 2u_2\eta_{12} = 0 \tag{2.19}$$

$$\frac{d\eta_{12}}{dt} + (2u_1 + v_2)\eta_{12} + u_2\left(\eta_{11} - \frac{2h_0}{L^2}\right) + v_1\left(\eta_{22} - \frac{2h_0}{l^2}\right) = 0 \tag{2.20}$$

These six equations and the requirement that $\eta_{12} = \eta_{21}$, which amounts to

$$\frac{d}{dt}\left(v_1 - u_2\right) + \left(u_1 + v_2\right)\left(v_1 - u_2 + f\right) = 0 \tag{2.21}$$

determine the seven unknown functions of time, $u_0$, $v_0$, $u_1$, $v_1$, $u_2$, $v_2$ and $h_0$. Of these seven ordinary differential equations, (2.15) and (2.21) are 1st order and (2.16)-(2.20) are 2nd order. As such, twelve initial conditions are required to obtain a unique solution. These initial conditions correspond to the initial fields of $u$, $v$ and $\eta$. The twelve time varying coefficients of (2.6), (2.7) and (2.8), namely, $u_0$, $v_0$, $u_1$, $v_1$, $u_2$, $v_2$, $h_0$, $h_1$, $h_2$, $h_{11}$, $h_{22}$ and $h_{12}$, are the initial conditions that fully define these fields.

## 2.3 Planar Surface

The first type of motion for which these equations are solved is the simplest. The velocities are assumed to vary in time but not space, $u_1 = u_2 = v_1 = v_2 = 0$, so $\eta_{11} = \eta_{22} = \eta_{12} = \eta_{21} = 0$. The functions of time that must be solved for are then, $u_0$, $v_0$ and $h_0$. Equations (2.18) to (2.21) are satisfied for all time, and equations (2.15) to (2.17) reduce to

$$\frac{dh_0}{dt} - \frac{1}{g}\left[u_0\frac{du_0}{dt} + v_0\frac{dv_0}{dt}\right] = 0 \tag{2.22}$$

$$\frac{d^2u_0}{dt^2} - f\frac{dv_0}{dt} + \frac{2gh_0}{L^2}u_0 = 0 \tag{2.23}$$

$$\frac{d^2v_0}{dt^2} + f\frac{du_0}{dt} + \frac{2gh_0}{L^2}v_0 = 0 \tag{2.24}$$

The equations that define the velocity components, (2.23) and (2.24), are linear and have constant coefficients. Therefore, $u_0$ and $v_0$ vary sinusoidally with frequency, $\omega$, that satisfies

$$\left(\omega^2 - \frac{2gh_0}{L^2}\right)^2 - f^2\omega^2 = 0 \tag{2.25}$$

There are two solutions that satisfy these equations. These solutions determine similar fluid motion, in which the surface of the fluid is planar and tilted with respect to the $z$-axis. The fluid rotates around the centre of the basin, maintaining the tilting angle in relation to the $z$-axis. One solution corresponds to clockwise motion, the other to anticlockwise motion. The solution for clockwise motion is presented below, and this is the solution that will be used for comparison with the ALE method.

$$u = -\lambda\omega \sin \omega t \tag{2.26}$$

$$v = -\lambda\omega \cos \omega t \tag{2.27}$$

$$\eta = 2\lambda\frac{h_0}{L}\left(\frac{x}{L}\cos \omega t - \frac{y}{L}\sin \omega t - \frac{\lambda}{2L}\right) \tag{2.28}$$

$$\omega = \frac{f}{2} + \left[\frac{f^2}{4} + \frac{2gh_0}{L^2}\right]^{\frac{1}{2}} \tag{2.29}$$

$\lambda$ is a constant that determines the amplitude of the motion. The shoreline remains circular in the $x - y$ plane, and the centre of the circle traces a circular orbit around the centre of the basin. The shoreline is given by the points $(x, y)$ that satisfy

$$(x - \lambda \cos \omega t)^2 + (y + \lambda \sin \omega t)^2 = L^2 \tag{2.30}$$

Figures 2.2 and 2.3 show the elevation field plotted in the y-z plane and the x-y plane.

## 2.4 Curved Surface

The second type of motion looked at in this project is restricted to divergence from, convergence towards, and rotation about the centre of the basin (in the $x - y$ plane). So, it is assumed that $u_0 = v_0 = 0$. The functions of time that must be solved for are $u_1$, $u_2$, $v_1$, $v_2$ and $\eta_0$. Equations (2.16) and (2.17) are now satisfied for all time. Further assumptions are made, namely that $u_1 = v_2$ and $u_2 = -v_1$. The three unknown functions are now determined by

$$\frac{d^2 u_1}{dt^2} + \left(\frac{8gh_0}{L^2} + f^2\right) u_1 + 6u_1 \frac{du_1}{dt} + 4u_1^3 = 0 \tag{2.31}$$

$$\frac{dv_1}{dt} + 2u_1 \left(v_1 + \frac{f}{2}\right) = 0 \tag{2.32}$$

$$\frac{d\eta_0}{dt} + 2u_1 (\eta_0 + h_0) = 0 \tag{2.33}$$

These have simple, exact solutions,

$$u_1 = \frac{\omega}{2} \frac{A \sin \omega t}{1 - A \cos \omega t} \tag{2.34}$$

$$v_1 = \left(v_{10} + \frac{f}{2}\right) \frac{1 - A}{1 - A \cos \omega t} - \frac{f}{2} \tag{2.35}$$

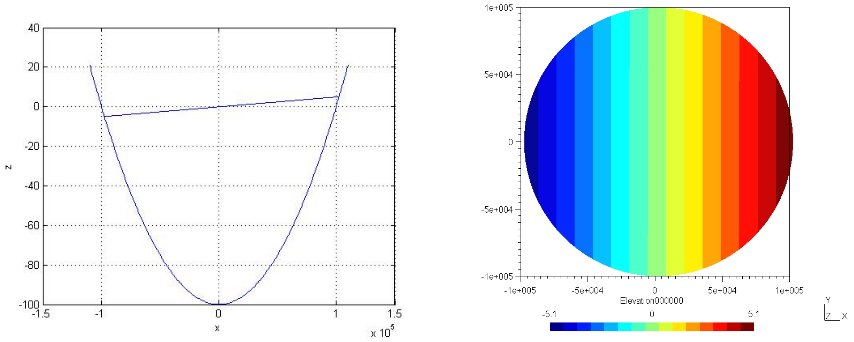

Figure 2.2: The exact planar solution, equation 2.28, of the elevation $\eta$ plotted at $t = 0$s with $\lambda = 2500$, $L = 100000$m and $h_0 = 100$m. The chart on the left shows the profile of the solution at $x = 0$. The chart on the right shows a surface plot of $\eta$ in the $x - y$ plane. The scaling runs from -5.1(blue) to 5.1(red). The surface is tilted so part of the boundary falls outside the axis box that fits tangentially around the equilibrium surface boundary.

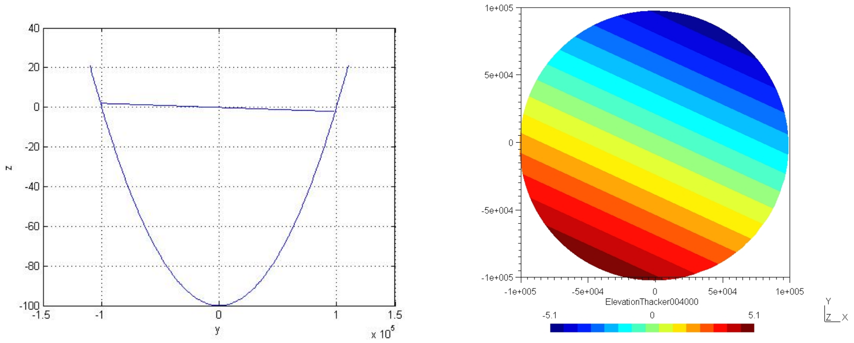

Figure 2.3: This is the same solution as 2.2 plotted at $t = 4000$s.

$$\eta_0 = (\lambda + h_0)\,\frac{1 - A}{1 - A\cos\omega t} - h_0 \tag{2.36}$$

$$\omega = \left(\frac{8gh_0}{L^2} + f^2\right)^{\frac{1}{2}} \tag{2.37}$$

where

$$A = \frac{(h_0 + \lambda)^2 - h_0^2}{(h_0 + \lambda)^2 + h_0^2} \tag{2.38}$$

$v_{10}$ is the initial value of $v_1$. In this case, $v_{10} = \frac{f\lambda}{2h_0}$. $\lambda$ is, again, the amplitude of the motion and also specifies the initial value of $\eta_0$. The surface of the fluid is a parabola of revolution,

$$\eta = \eta_0 + \frac{1}{2}h_{11}\left(x^2 + y^2\right) \tag{2.39}$$

The complete solution for the motion is given by

$$u = \frac{1}{1 - A\cos\omega t}\left[\frac{1}{2}\omega x A\sin\omega t - \frac{1}{2}fy\left(\left(1 - A^2\right)^{\frac{1}{2}} + A\cos\omega t - 1\right)\right] \tag{2.40}$$

$$v = \frac{1}{1 - A\cos\omega t}\left[\frac{1}{2}\omega y A\sin\omega t + \frac{1}{2}fx\left(\left(1 - A^2\right)^{\frac{1}{2}} + A\cos\omega t - 1\right)\right] \tag{2.41}$$

$$\eta = h_0\left[\frac{(1 - A^2)^{\frac{1}{2}}}{1 - A\cos\omega t} - 1 - \frac{x^2 + y^2}{L^2}\left(\frac{1 - A^2}{(1 - A\cos\omega t)^2}\right)\right] \tag{2.42}$$

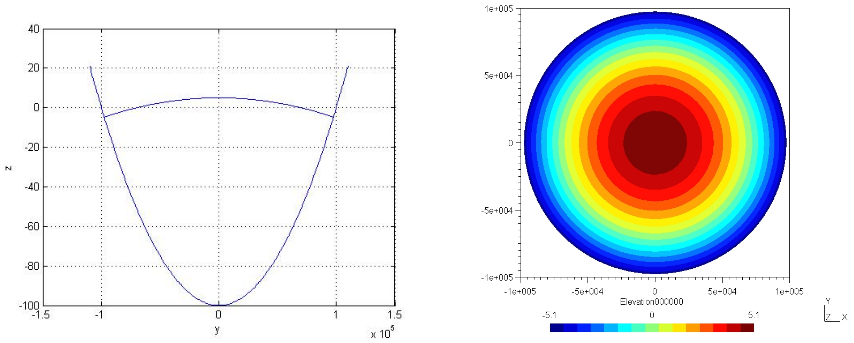

Figure 2.4: The exact curved solution, equation 2.42, of the elevation $\eta$ plotted at $t = 0$s with $\lambda = 5$, $L = 100000$m and $h_0 = 100$m. The chart on the left shows the profile of the solution at $x = 0$. The chart on the right shows a surface plot of $\eta$ in the $x - y$ plane. The scaling runs from -5.1(blue) to 5.1(red).The surface is convex so the boundary has contracted within the equilibrium boundary.

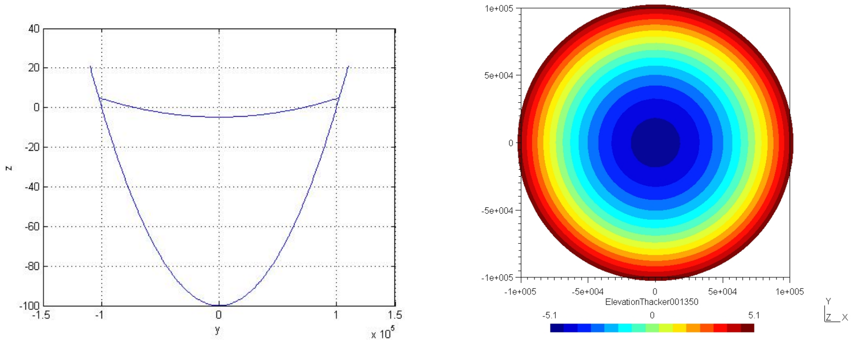

Figure 2.5: This is the same solution as 2.4 plotted at $t = 3500$s. The surface is now concave so the boundary has expanded to be outside the equilibrium boundary.

Providing $\lambda > 0$, the surface of the fluid starts as a convex parabola, with the centre of the fluid above the equilibrium level and the circular shoreline below. The circular shoreline, centred at the origin in the $x - y$ plane, therefore has a smaller radius than the equilibrium shoreline. (If $\lambda < 0$, the inverse is true and the surface starts off as a concave parabola.) As the solution evolves, the centre of the fluid drops and the shoreline rises and expands. The radius, $R$, of the shoreline, as a function of time, is given by

$$R\left(t\right) = L\left[\frac{h_0}{h_0 + \eta_0\left(t\right)}\right]^{\frac{1}{2}} \tag{2.43}$$

## 2.5 Numerical integration of the ODE system

The specific solutions presented in the previous two sections are the solutions used to validate the model presented in chapter 4. Thacker presents some other specific solutions in [1]. Solutions are given for the planar and parabolic surfaces in parabolic channels, where $l >> L$. A solution is also given for a parabolic flood wave, where $h_0 = 0$, which represents a parabolic mound of water being dropped onto a planar surface. Further solutions for comparison can be obtained by numerically integrating the ordinary differential equations obtained in section 2.3. Equations (2.9)-(2.20) can be rearranged so that they are of the form

$$\frac{d}{dt}\begin{pmatrix} u_0 \\ \vdots \\ \eta_{12} \end{pmatrix} = \boldsymbol{f}\left(u_0, u_1, u_2, v_0, v_1, v_2, \eta_0, \eta_1, \eta_2, \eta_{11}, \eta_{22}, \eta_{12}\right)$$

This is now a system of 12 first order ordinary differential equations that, given appropriate initial values for the 12 unknowns, can be numerically integrated using any number of techniques. The 12 dependent variables can then be used to reconstruct the approximated solution across the domain at each, or any, time step. The method used to solve the system of ODEs in this project is the 4th-order Runge-Kutta, RK4, scheme.

$$\boldsymbol{W}_{n+1} = \boldsymbol{W}_n + \frac{h}{6}\left(\boldsymbol{K}_1 + 2\boldsymbol{K}_2 + 2\boldsymbol{K}_3 + \boldsymbol{K}_4\right) \tag{2.44}$$

$$\boldsymbol{K}_1 = \boldsymbol{f}\left(t_n, \boldsymbol{W_n} + \frac{1}{2}\boldsymbol{K_1}\right) \tag{2.45}$$

$$\boldsymbol{K}_2 = \boldsymbol{f}\left(t_n + \frac{h}{2}, \boldsymbol{W_n} + \frac{1}{2}\boldsymbol{K_2}\right) \tag{2.46}$$

$$\boldsymbol{K}_3 = \boldsymbol{f}\left(t_n + \frac{h}{2}, \boldsymbol{W_n} + \frac{1}{2}\boldsymbol{K_3}\right) \tag{2.47}$$

$$\boldsymbol{K}_4 = \boldsymbol{f}\left(t_n, \boldsymbol{W_n} + \frac{1}{2}\boldsymbol{K_3}\right) \tag{2.48}$$

where $h = \Delta t$, $n$ is the timestep and

$$\boldsymbol{W}_n = \begin{pmatrix} u_0 \\ \vdots \\ \eta_{12} \end{pmatrix}_n$$

The RK4 scheme is $O(\Delta t^4)$ accurate and this must be taken into account when using the numerically integrated solutions for comparison with the ALE solutions.

# Chapter 3

# A Finite element method

## 3.1 Background

The moving mesh method used in this project is an extension of the finite element method proposed by Le Roux et al. [3]. The results have been obtained by applying the moving mesh method to the existing code used by Le Roux et al. [3]. This chapter briefly explains the formulation of the finite element approximation used in [3].

## 3.2 Eulerian formulation

The method that shall be adapted uses an Eulerian scheme to solve the equations. The Eulerian scheme approximates the solution on a fixed spatial grid. It can be thought of as monitoring the evolution of the system as it passes through these fixed points in space. This is opposed to a Lagrangian scheme where the evolution of the system is monitored at points that advect with the solution. The solution is integrated along the characteristics and, in this case, can be thought of as tracking the motion of individual parcels of fluid.

The Eulerian formulation of the shallow water equations in vector form are as follows

$$\frac{\partial \eta}{\partial t} + \nabla . \left[ \left( h + \eta \right) \boldsymbol{u} \right] = 0 \tag{3.1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} . \nabla \boldsymbol{u} + f \boldsymbol{k} \times \boldsymbol{u} = -g \nabla \eta \tag{3.2}$$

where $\boldsymbol{k}$ is a unit vector in the vertical direction.

## 3.3 Nondimensionalization

In this project, the equations are solved using dimensionless variables. In order to do this, equations (3.1) and (3.2) must be nondimensionalized.

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{U} \tilde{\boldsymbol{u}} \\ \eta &= E \tilde{\eta} \\ h &= H_s \tilde{h} \\ \boldsymbol{x} &= L \tilde{\boldsymbol{x}} \\ t &= T \tilde{t} \end{aligned}$$

where $s$ is the dimensional variable, $\tilde{s}$ is the dimensionless variable and $S$ is the scaling factor. Equation (3.1) with the dimensional variables replaced by their dimensionless counterparts, and scaling factors, is

$$\frac{E}{T}\frac{\partial\tilde{\eta}}{\partial\tilde{t}} + \frac{U}{L}\nabla.\left[\left(H_s\tilde{h} + E\tilde{\eta}\right)\tilde{\boldsymbol{u}}\right] = 0 \tag{3.3}$$

This can be rearranged so that

$$\frac{\partial\tilde{\eta}}{\partial\tilde{t}} + \frac{TH_sU}{EL}\nabla.\left[\left(\tilde{h} + \frac{E}{H_s}\tilde{\eta}\right)\tilde{\boldsymbol{u}}\right] = 0 \tag{3.4}$$

In order to prevent the divergence term from swamping the time derivative term, $T$ can be chosen so that the total scaling factor is equal to 1.

$$T = \frac{EL}{H_sU} \tag{3.5}$$

The same process is applied to equation (3.2), replacing the dimensional variables by their dimensionless counterparts and scaling factors,

$$\frac{U}{T}\frac{\partial\tilde{\boldsymbol{u}}}{\partial\tilde{t}} + \frac{UU}{L}\tilde{\boldsymbol{u}}.\nabla\tilde{\boldsymbol{u}} + Uf\boldsymbol{k}\times\tilde{\boldsymbol{u}} = -\frac{gE}{L}\nabla\tilde{\eta} \tag{3.6}$$

This can be rearranged so that

$$\frac{1}{Tf}\frac{\partial\tilde{\boldsymbol{u}}}{\partial\tilde{t}} + \frac{U}{Lf}\tilde{\boldsymbol{u}}.\nabla\tilde{\boldsymbol{u}} + \boldsymbol{k}\times\tilde{\boldsymbol{u}} = -\frac{gE}{fUL}\nabla\tilde{\eta} \tag{3.7}$$

This time the pressure gradient term must be prevented from swamping the solution, so $U$ is chosen so that the total scaling factor is equal to 1.

$$U = \frac{gE}{fL} \tag{3.8}$$

The dimensionless variables are used throughout the project. From this point onwards, for ease of notation, the tildes shall be dropped from the dimensionless variables.

## 3.4 Finite element discretization

The finite element method is a way of approximating the solution to a boundary value problem. The boundary value problem must be rephrased in its weak form and then discretized in finite dimensional space. Typically, in two dimensions, this involves a triangulation of the domain into piecewise linear functions.

The weak form of equations (3.1) and (3.2) are defined so that the elevation solution is continuous everywhere but the velocity solution can be discontinuous between elements. The weak form is defined as thus so that the $P_1^{NC}$ shape functions, discussed in the next section, can be used. The integral forms of equations (3.1) and (3.2) are multiplied by test functions $\omega_\eta$ and $\boldsymbol{\omega}_u$ respectively.

$$\sum_{e=1}^{N_E} \int_{\Omega_e} \left( \frac{\partial \eta}{\partial t} \omega_\eta - (h+\eta)\, \boldsymbol{u}.\nabla \omega_\eta \right) d\Omega + \sum_{e=1}^{N_E} \int_{\partial \Omega_e} (h+\eta)\, \omega_\eta \boldsymbol{u}.\boldsymbol{n}_e d\Gamma = 0 \tag{3.9}$$

$$\sum_{e=1}^{N_E} \int_{\Omega_e} \left( \frac{\partial \boldsymbol{u}}{\partial t}.\boldsymbol{w}_u - \left(\nabla.\left(\boldsymbol{u}\boldsymbol{w}_u\right)\right).\boldsymbol{u} + f\left(\boldsymbol{k} \times \boldsymbol{u}\right).\boldsymbol{w}_u + g\nabla\eta.\boldsymbol{w}_u \right) d\Omega \tag{3.10}$$

$$+ \sum_{e=1}^{N_E} \int_{\Omega_e} \left(\boldsymbol{u}\boldsymbol{u}.\boldsymbol{n}_e\right).\boldsymbol{w}_u d\Gamma + \sum_{l=1}^{N_\Gamma} \int_{\Gamma_l} [\boldsymbol{u}] . [\boldsymbol{a}\left(\boldsymbol{w}_u\right)] d\Gamma = 0$$

where $N_E$ and $N_\Gamma$ are the number of elements and the number of interelement boundaries, respectively. $\boldsymbol{n}_e$ is the outward normal vector to the boundary of each element $\delta\Omega_e$. $[b] = b_{|\Omega_e} - b_{|\Omega_f}$ is the jump of $\boldsymbol{b}$ on an interior edge $\Gamma_l$ and $b_{|\Omega_e}$ denotes the restriction of $\boldsymbol{b}$ on $\Omega_e$. The final integral in (3.10) appears because the velocity can be discontinuous between elements. It weakly imposes the continuity of the velocity. The function $\boldsymbol{a}$ imposes continuity of the solution along the characteristics.

$$\boldsymbol{a}\left(\boldsymbol{w}_u\right) = \begin{cases} \boldsymbol{u}.\boldsymbol{n}\left(\gamma - 1/2\right)\boldsymbol{w}_u, & on \quad \Omega_e \\ \boldsymbol{u}.\boldsymbol{n}\left(\gamma + 1/2\right)\boldsymbol{w}_u, & on \quad \Omega_f \end{cases}$$

where $\gamma \in [-1/2, 1, 2]$. Choosing $\gamma = 0$ gives a centred advection scheme. In this project an upwind advection scheme is used, Le Roux et al. [4] and Houston et al. [6], which is obtained by choosing

$$\gamma = \frac{1}{2}\frac{\boldsymbol{u}(\boldsymbol{x}).\boldsymbol{n}(\boldsymbol{x})}{|\boldsymbol{u}(\boldsymbol{x}).\boldsymbol{n}(\boldsymbol{x})|} \tag{3.11}$$

The elevation solution is continuous so there are no jump terms in the finite element discretization of equation (3.9). The boundary integral in (3.9) is assumed to be equal to zero to enforce mass conservation in the finite element discretization. With some standard algebra, Le Roux et al. [4] and Houston et al. [6], the weak form can therefore be rewritten as

$$\sum_{e=1}^{N_E} \int_{\Omega_e} \left( \frac{\partial \eta}{\partial t}\omega_\eta - (h + \eta)\,\boldsymbol{u}.\nabla\omega_\eta \right) d\Omega = 0 \tag{3.12}$$

$$\sum_{e=1}^{N_E} \int_{\Omega_e} \left( \frac{\partial \boldsymbol{u}}{\partial t} . \boldsymbol{w}_u - \left( \nabla . \left( \boldsymbol{u} \boldsymbol{w}_u \right) \right) . \boldsymbol{u} + f \left( \boldsymbol{k} \times \boldsymbol{u} \right) . \boldsymbol{w}_u + g \nabla \eta . \boldsymbol{w}_u \right) d\Omega$$
$$+ \sum_{l=1}^{N_\Gamma} \int_{\Omega_e} \langle \boldsymbol{u}\boldsymbol{u}.\boldsymbol{n} \rangle_\gamma . \left[ \boldsymbol{w}_u \right] d\Gamma \tag{3.13}$$

where $\langle b \rangle_\gamma$ is the weighed average of $b$ on the segment $\Gamma_l$

$$\langle b \rangle_\gamma = \left( \frac{1}{2} + \gamma \right) b_{|\Omega_e} + \left( \frac{1}{2} - \gamma \right) b_{|\Omega_f} \tag{3.14}$$

The finite element approximation is obtained by replacing $\eta$ and $\boldsymbol{u}$ by

$$\eta \approx \eta^h = \sum_{i=1}^{N_V} \eta_i \omega_i \tag{3.15}$$

$$\boldsymbol{u} \approx \boldsymbol{u}^h = \sum_{j=1}^{N_S} \boldsymbol{u}_j \psi_j \tag{3.16}$$

where $N_V$ and $N_S$ are the number of vertices and segments respectively. $\eta_i$ and $\boldsymbol{u}_i$ are the nodal values for the elevation and velocity respectively and $\omega_i$ and $\psi_i$ are the shape functions associated with them. $\omega_j$ replaces the test function $w_\eta$ in (3.12). $(\psi_i, 0)$ and $(0, \psi_i)$ replace the test function $\boldsymbol{w}_u$ in (3.13).

### 3.4.1 The non-conforming mixed $P_1^{NC}$-$P_1$ discretization

The elevation variable is approximated using standard linear conforming $P_1$ shape functions, as shown in figure 3.1. The elevation nodes are at the vertices of this triangulation and therefore the elevation field is continuous everywhere. The velocity variable is approximated using linear non-conforming $P_1^{NC}$ shape functions, also shown in figure 3.1. The velocity nodes are situated at the midpoints of the sides of each element. The velocity field is only continuous at the nodes and discontinuous at all other points around the element boundary. These nonconforming shape functions have the orthogonality property

$$\int_\Omega \psi_p \psi_q d\Omega = \frac{A_q}{3}\delta_{pq} \tag{3.17}$$

where $A_q$ is the area of support of $\psi_q$ and $\delta_{pq}$ is the Kronecker delta. This orthogonality property greatly reduces the computation required.

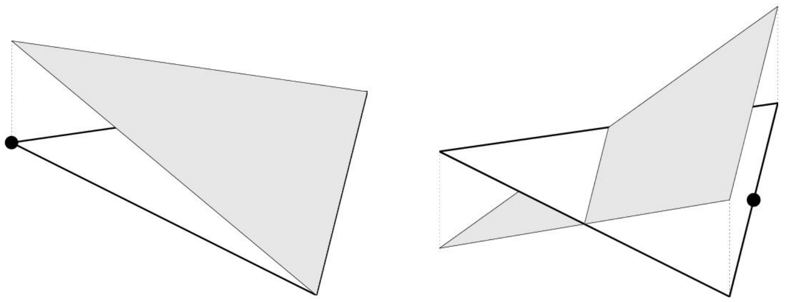

Figure 3.1: The $P_1$ shape function is shown on the left and the $P_1^{NC}$ shape function on the right.

A global linear approximation of the product of $\boldsymbol{u}^h$ with $f$ is made to simplify the algebra. $f_i$ represents the value of the coriolis parameter at a velocity node. $f$ varies smoothly across the domain so the effect on the accuracy is small.

$$\int_{\Omega(e)} f\left(\boldsymbol{k}\times\boldsymbol{u}^h\right)\psi_j d\Omega = \int_{\Omega(e)} f\sum_{i=1}^{N_s}\left(\boldsymbol{k}\times\boldsymbol{u}_i\right)\psi_i\psi_j d\Omega \tag{3.18}$$

$$\approx \int_{\Omega(e)}\sum_{i=1}^{N_s}\underbrace{f_i\left(\boldsymbol{k}\times\boldsymbol{u}_i\right)\psi_i}_{(f\boldsymbol{k}\times\boldsymbol{u})^h}\psi_j d\Omega$$

The space discretized finite element formulation of equations (3.1) and (3.2) are therefore

$$\sum_{e=1}^{N_E}\int_{\Omega_e}\left(\frac{\partial\eta^h}{\partial t}\omega_i-\left(h+\eta^h\right)\boldsymbol{u}^h.\nabla\omega_i\right)d\Omega=0 \tag{3.19}$$

$$\sum_{e=1}^{N_E}\int_{\Omega_e}\left(\frac{\partial\boldsymbol{u}^h}{\partial t}\psi_j-\boldsymbol{u}^h\nabla.\left(\boldsymbol{u}^h\psi_j\right)+\left(f\boldsymbol{k}\times\boldsymbol{u}\right)^h\psi_j+g\nabla\eta^h\psi_j\right)d\Omega$$

$$+\sum_{l=1}^{N_\Gamma}\int_{\Gamma_l}\left\langle\boldsymbol{u}^h\boldsymbol{u}^h.\boldsymbol{n}\right\rangle_\gamma\left[\psi_j\right]d\Gamma=0 \tag{3.20}$$

### 3.4.2 Mass lumping

A rigorous treatment of the integrals in equations (3.19) and (3.20) requires that a mass matrix is formed at each timestep. This mass matrix is a sparse matrix and would require a linear solver, such as the conjugate gradient method, to invert it. To avoid this inversion of the mass matrix, and the large amount of computation it necessitates, the mass matrix is lumped. The integrals that require this treatment are

$$\int_{\Omega(e)}\omega_i\omega_j d\Omega=\frac{A_e}{12}\begin{pmatrix}2&1&1\\1&2&1\\1&1&2\end{pmatrix}$$

If it is assumed that the mass of each element is concentrated at the nodes, then the off diagonal components can be added to the diagonal components to create a diagonal mass matrix.

$$\int_{\Omega(e)} \omega_i \omega_j d\Omega \approx \frac{A_e}{3} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The integrals in the momentum equation do not require this treatment as the orthogonality property, (3.17), of the nonconforming shape functions ensure that the mass matrix is already diagonal.

$$\int_{\Omega(e)} \psi_i \psi_j d\Omega = \frac{A_e}{3} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Le Roux et. al [5] have studied the effect of mass lumping on the shallow water equations. They have shown that, for the $P_1 - P_1^{NC}$ shape function pairing, the effect on the propogation of gravity waves is limited. The lower frequency Rossby waves are slowed down by the mass lumping. However, Le Roux et. al [5] concluded that the 2D shallow water equations, with the $P_1 - P_1^{NC}$ shape function pairing, could be mass lumped without significantly compromising the model's accuracy and dispersion properties.

## 3.5 Time discretization

In order to apply the finite element method, equations (3.19) and (3.20) must be discretized in time. The time discretization presented in this section is of the original equations, (3.1) and (3.2), for ease of notation. The explicit 3rd order Adams-Bashforth scheme is used. The scheme is $O(\Delta t^3)$ accurate and conditionally stable for both advective and diffusive processes. Durran [7] shows that the scheme, with sufficiently small timestep, provides excellent damping of the computational modes compared with limited damping of the physical modes. The scheme is a linear multistep method and requires the storage of the solution at three timesteps.

$$\frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_n}{\Delta t} = \frac{1}{2}\left(23b_u^n - 16b_u^{n-1} + 5b_u^{n-2}\right) \tag{3.21}$$

$$b_u^n = -\boldsymbol{u}_n.\nabla \boldsymbol{u}_n - f\boldsymbol{k} \times \boldsymbol{u}_n - g\nabla\eta_n \tag{3.22}$$

$$\frac{\eta^{n+1} - \eta^n}{\Delta t} = \frac{1}{2}\left(23b_\eta^n - 16b_\eta^{n-1} + 5b_\eta^{n-2}\right) \tag{3.23}$$

$$b_\eta^n = -\nabla.\left[\left(h + \eta_n\right)\boldsymbol{u}_n\right] \tag{3.24}$$

# Chapter 4

# A Moving mesh method

## 4.1 Background

One of the major advantages of the finite element method, over the finite difference method for instance, is that unstructured meshes can be used. Most physical problems involve irregular shaped domains that can be more precisely modelled using unstructured meshes. Unstructured meshes can also be adjusted to give higher resolution, and therefore greater accuracy, at areas of interest within the domain. Many problems in fluid dynamics involve domains that change shape. Also, it is often not known a priori where the areas of interest will be within the domain. For these reasons, methods that dynamically adapt the mesh to track the boundary, or to follow areas of interest, are sought in fluid dynamics.

There are three main adaptive mesh methods used in numerical ocean modelling, Piggott et al. [12], $h$ adaptivity, $r$ adaptivity and $p$ adaptivity. $h$ adaptivity modifies the mesh by adding, or removing, grid points at areas of the domain that require, respectively, increased or reduced resolution. Each time the mesh is updated, the solution must be interpolated onto the new grid points. The advantage of this method is that, in a highly transient flow, the resolution of the grid can be altered very quickly, however, the interpolation can be computationally expensive. With $r$ adaptivity, the mesh contains a fixed number of gridpoints. These gridpoints are moved within the domain to satisfy some requirement of the flow. The advantage of $r$ adaptivity, with respect to fluid dynamics, is that areas of interest, such as eddies, move with the flow and so unnecessary interpolation is not required at each time step to resolve them. $p$ adaptivity seeks to improve the accuracy at certain areas within the domain by increasing the polynomial order of approximation at those points. In this project, a form of $r$ adaptivity is used to move the mesh with the moving boundary.

## 4.2 ALE method

There are two classical descriptions of motion for systems such as the shallow water equations. The Lagrangian description, which tracks individual particles as the system evolves, and the Eulerian description, where the evolution of the system is tracked from fixed reference points in space. These two descriptions of motion have advantages and disadvantages when applied to computational fluid dynamics. The Lagrangian description is physically fundamental, as most physical laws are expressed in Lagrangian terms. As such, numerical approximations to physical systems like the shallow water equations are often simpler, and more accurate, when stated in Lagrangian terms. The problem that arises, is that the computational domain can become distorted quickly as the nodes track the motion of the fluid. The Eulerian description, with its fixed computational nodes, does not have this problem, although a higher grid resolution will often be required to attain the same level of accuracy as the Lagrangian method. The method used to move the mesh in this project is the Arbitrary Lagrangian-Eulerian(ALE) method. This method seeks to use the advantages of both the Eulerian and Lagrangian methods.

## 4.3 Derivation of mesh velocities

In this project, the mesh velocities are obtained by conserving the integral of a monitor function over the patch of elements surrounding each node. The monitor function is the total depth of the fluid, $H = \eta + h$, so that the volume of fluid in a control volume is conserved. The Eulerian form of the continuity equation, (3.1), must be transformed into a moving frame of reference with velocity $\boldsymbol{q}$.

### 4.3.1 Integral form in a moving frame

The Reynolds' Transport Theorem states that the rate of change of $H$ is equal to the rate of change of $H$ within the control volume and the net rate of change of $H$ through the control surface. Application of the divergence theorem leads to

$$\frac{d}{dt}\int_{\Omega(t)} H d\Omega = \int_{\Omega(t)} \frac{\partial H}{\partial t} d\Omega + \oint_{\partial\Omega(t)} H\boldsymbol{q}.\boldsymbol{n} ds \tag{4.1}$$

$$= \int_{\Omega(t)} \left( \frac{\partial H}{\partial t} + \nabla . \left( H\boldsymbol{q} \right) \right) d\Omega$$

where

$$\frac{\partial H}{\partial t} = \frac{\partial \eta}{\partial t} \tag{4.2}$$

The integral form of the continuity equation, 3.1, in the moving frame of reference can therefore be written as

$$\frac{d}{dt} \int_{\Omega(t)} H d\Omega - \int_{\Omega(t)} \nabla . \left( H\boldsymbol{q} \right) d\Omega = - \int_{\Omega(t)} \nabla . \left[ \left( H \right) \boldsymbol{u} \right] d\Omega \tag{4.3}$$

### 4.3.2 Introduction of monitor function

The method used to move the mesh is based on the conservation of the integral of $H$.

$$\int_{\Omega(t)} H d\Omega = C \tag{4.4}$$

where $C$ is constant in time. So

$$\frac{d}{dt} \int_{\Omega(t)} H d\Omega = 0 \tag{4.5}$$

The first term in equation 4.3 is therefore zero. With the volume of water conserved, equation 4.3 becomes

$$- \int_{\Omega(t)} \nabla . \left( H\boldsymbol{q} \right) d\Omega = - \int_{\Omega(t)} \nabla . \left[ \left( H \right) \boldsymbol{u} \right] d\Omega \tag{4.6}$$

### 4.3.3 Weak form

In order to apply a finite element method, equation 4.3 must be generalised to the weak form. A test function $\omega$ is introduced that moves with velocity $\boldsymbol{q}$ which therefore satisfies the advection equation

$$\frac{\partial \omega}{\partial t} + \boldsymbol{q}.\nabla \omega = 0 \tag{4.7}$$

Using a generalisation of equation (4.3)

$$\frac{d}{dt}\int_{\Omega(t)} \omega H d\Omega - \int_{\Omega(t)} \omega \nabla. \left(H\boldsymbol{q}\right) d\Omega = -\int_{\Omega(t)} \omega \nabla. \left[\left(H\right)\boldsymbol{u}\right] d\Omega \tag{4.8}$$

the weak form of equation (4.6) is

$$-\int_{\Omega(t)} \omega \nabla. \left(H\boldsymbol{q}\right) d\Omega = -\int_{\Omega(t)} \omega \nabla. \left(H\boldsymbol{u}\right) d\Omega \tag{4.9}$$

This can be integrated by parts to obtain

$$-\oint_{\partial\Omega(t)} \omega H \boldsymbol{q}.\boldsymbol{n} d\Gamma + \int_{\Omega(t)} H\boldsymbol{q}.\nabla \omega d\Omega = -\int_{\Omega(t)} \omega \nabla. \left(H\boldsymbol{u}\right) d\Omega \tag{4.10}$$

### 4.3.4 Introduction of velocity potential

In the problem being solved, $H = 0$ on the boundary, so the boundary integral in 4.10 disappears. To determine $\boldsymbol{q}$ uniquely, additional constraints are needed. If the vorticity, $\nabla \times \boldsymbol{q}$, is defined, then given $H$, it is possible to uniquely determine the velocity $\boldsymbol{q}$ using Helmholtz' Theorem. For the initial investigation the mesh velocity $\boldsymbol{q}$ is defined as irrotational, $\nabla \times \boldsymbol{q} = 0$. A velocity potential $\phi$, therefore exists that satisifies

$$\boldsymbol{q} = \nabla \phi \tag{4.11}$$

So, substituting 4.11 into 4.10 gives

$$\int_{\Omega(t)} H\nabla\phi.\nabla\omega d\Omega = -\int_{\Omega(t)} \omega\nabla.\left(H\boldsymbol{u}\right)d\Omega \tag{4.12}$$

### 4.3.5 Finite element Discretization

In order to discretize equation 4.12 for the finite element method, the test function $\omega$ is replaced by a compact basis function $\omega_i(x, y)$. In this case, $\omega_i$ is the $P_1$ shape function discussed in chapter 3.

$$\omega = \omega_i(x, y) \tag{4.13}$$

The continuous variables, $\phi$ and $H$, are replaced by their discretized approximations, $\Phi$ and $\widehat{H}$ as

$$\phi \approx \Phi = \sum_{j=1}^{N} \Phi_j\omega_j \tag{4.14}$$

$$\nabla\Phi = \sum_{j=1}^{N} \Phi_j\nabla\omega_j \tag{4.15}$$

$$H \approx \widehat{H} = \sum_{k=1}^{N} \widehat{H}_k\omega_k \tag{4.16}$$

So the finite element discretization of equation 4.12 is

$$\sum_{j=1}^{N}\left[\int_{\Omega(t)} \nabla\omega_i.\nabla\omega_j\widehat{H}d\Omega\right]\Phi_j = -\sum_{j=1}^{N}\int_{\Omega(t)} \omega_i\nabla.\widehat{H}\boldsymbol{u}d\Omega \tag{4.17}$$

The right hand side of 4.17 can be integrated. The left hand side forms a weighted stiffness matrix which is assembled element by element to get a linear system of the form

$$K(\widehat{H})\boldsymbol{\Phi} = \boldsymbol{f} \tag{4.18}$$

### 4.3.6 Stiffness matrix assembly

The element contribution to the stiffness matrix is

$$K_{ij}^e = \int_{\Omega(t)} \widehat{H}\nabla\omega_i.\nabla\omega_j d\Omega \tag{4.19}$$

Over a triangular element, $\Delta ABC$, with angles $\alpha$, $\beta$ and $\gamma$, and sides $a$, $b$ and $c$, the off diagonal element $K_{AB}$ is equal to

$$K_{AB}^e = \nabla\omega_A.\nabla\omega_B \int_{\Omega(t)} \widehat{H} d\Omega \tag{4.20}$$

$$\widehat{H}_e = \sum_{k=1}^{3} \widehat{H}_k \tag{4.21}$$

$$K_{AB}^e = \nabla\omega_A.\nabla\omega_B \widehat{H}_e \frac{A(e)}{3} \tag{4.22}$$

where $A(e)$ is the area of the element.

$$K_{AB} = \frac{1}{H_A}.\frac{1}{H_B}\cos\left(\pi - \gamma\right)\widehat{H}_e \frac{A(e)}{3} \tag{4.23}$$

$$K_{AB} = \frac{a}{2A(e)}.\frac{b}{A(e)}\cos\left(\pi - \gamma\right)\widehat{H}_e \frac{A(e)}{3} \tag{4.24}$$

$$K_{AB} = \frac{1}{12}\frac{ab}{A(e)}\cos\left(\pi - \gamma\right)\widehat{H}_e \tag{4.25}$$

$$K_{AB} = \frac{1}{6}\frac{\cos\left(\pi - \gamma\right)}{\sin(\gamma)}\widehat{H}_e \tag{4.26}$$

$$K_{AB} = -\frac{\widehat{H}_e}{6} \cot \gamma \tag{4.27}$$

The same derivation applies to all the off-diagonal components of the element matrix. The diagonal components are derived as following. Since

$$\nabla \omega_A + \nabla \omega_B + \nabla \omega_C = 0 \tag{4.28}$$

so equation 4.22 applied to a diagonal element, $B = A$, becomes

$$K_{AA} = \widehat{H}_e \frac{A(e)}{3} \nabla \omega_A . \left( -\nabla \omega_B - \nabla \omega_C \right) \tag{4.29}$$

$$K_{AA} = \frac{\widehat{H}_e}{6} \cot \left( \beta + \gamma \right) \tag{4.30}$$

Again, the same derivation applies to the other two diagonal components of the element matrix.

$$K^e = \frac{H_e}{6} \begin{pmatrix} \cot(\beta + \gamma) & -\cot \gamma & -\cot \beta \\ -\cot \gamma & \cot(\alpha + \gamma) & -\cot \alpha \\ -\cot \ \beta & -\cot \alpha & \cot(\alpha + \beta) \end{pmatrix}$$

The element matrices are then assembled into the full stiffness matrix, each component of the 3 × 3 element matrix being added to the full matrix component that corresponds to the computational nodes involved. The mesh used in the investigation contains 1353 nodes. The full stiffness matrix $K$ is a sparse matrix, most of the components are zero. To avoid storing all of these zero components a column position array, $K_p$ is used. The non zero values of $K$ are read into a computational $K_c$ matrix. These values are placed in the first available column in $K_c$ in the same row as their theoretical position in $K$. The theoretical column position is then placed in the corresponding component in $K_p$. This reduces the number of components stored from 1830609 in $K$ to 2760 in $K_c$ and $K_p$. It also reduces the computation required in the linear solver by a similar magnitude.

In order to make the matrix $K$ non-singular, and therefore ensure that 4.18 has a unique solution, a value of $\Phi$ must be specified at one node. The position of this node, and the value specified, are unimportant because it is only the gradient of $\Phi$ that is required. The linear system can be solved using any linear solver, although the conjugate gradient method is a good choice as $K$ is symmetric and positive definite.

### 4.3.7 Solving for Q

Once $\Phi$ has been obtained by solving 4.18, the mesh velocity **q** must be derived from equation 4.11. Discretizing equation 4.11 into finite element form using 4.13, 4.15. The left hand side of the equation can be approximated using the $P_1^{NC}$ shape functions

$$\boldsymbol{q} \approx \boldsymbol{Q} = \sum_{j=1}^{N} \boldsymbol{Q}_j \psi_j \tag{4.31}$$

so that

$$\sum_{j=1}^{N} \left( \int_{\Omega(e)} \psi_i . \psi_j d\Omega \right) \boldsymbol{Q}_j = \sum_{j=1}^{N} \left( \int_{\Omega(e)} \omega_i d\Omega \right) \nabla \omega_j \Phi_j \tag{4.32}$$

The left hand side forms a mass matrix. When implementing the method this mass matrix was initially lumped, see section 3.4.2. When the method was found to be unstable, the full mass matrix was applied. This did not fully resolve all the problems but did allow the model to run for a longer period of time before blowing up.

$$\frac{A_e}{12}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}\boldsymbol{Q}_i = \int_{\Omega(e)} \omega_i d\Omega \nabla\Phi_i(e)$$

where $A(e)$ is the area of the element and

$$\nabla\Phi_i(e) = \sum_{j=1}^{3} \nabla\omega_j\Phi_j \tag{4.33}$$

The mass matrix is assembled element by element and solved using the conjugate gradient method.

## 4.4 Moving the mesh

The new coordinates of the mesh points are obtained by integrating the mesh velocities using the forward Euler method.

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta t\boldsymbol{q} \tag{4.34}$$

## 4.5 Equations in the moving frame

With the mesh velocities derived, and the grid points moved, the governing equations must be derived with respect to the moving frame of reference. The continuity equation is simple

$$\frac{d}{dt}\int_{\Omega(t)} \omega H d\Omega = 0 \tag{4.35}$$

$$\int_{\Omega(t)} \omega H d\Omega = C \tag{4.36}$$

where $C$ is constant in time. In order to conserve the volume of water in a patch of elements $C$ is set as the integral of $H$ at $t = 0$.

$$\int_{\Omega(t)} \omega H d\Omega = \int_{\Omega(0)} \omega(0) H d\Omega \tag{4.37}$$

The total depth of the fluid at each node, at time $t$, is therefore given by a weighted average of the areas of support

$$H_i(t) = \frac{\sum_{k=1}^{M} Ae_k(0) H_{ke}(0)}{\sum_{k=1}^{M} Ae_k(t)} \tag{4.38}$$

where $M$ is the number of supporting elements, $Ae_k(t)$ and $Ae_k(0)$ are the area of element $k$ at time $t = t$ and $t = 0$ respectively, and $H_{ke}(0)$ is the average height of element $k$ at $t = 0$.

The momentum equations, 3.2, are also modified to account for the moving frame of reference. An advective term, that moves with velocity $\boldsymbol{q}$ is included to account for the movement of the mesh. The mesh velocities $\boldsymbol{q}$ are interpolated onto the $P_1^{NC}$ nodes before the calculation takes place.

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}.\nabla \boldsymbol{u} + f\boldsymbol{k} \times \boldsymbol{u} - \nabla.(\boldsymbol{q}\boldsymbol{u}) = -g\nabla\eta \tag{4.39}$$

The momentum equations are implemented in the same way as before. The extra term is approximated using the same finite element technique and the resulting ODE's are time integrated using the 3rd order Adams-Bashforth scheme.

# Chapter 5

# Numerical Results

## 5.1 Planar surface

In this section, the results from the ALE method described in chapter 4 are compared with the exact planar surface solution described in section 2.3. Throughout this section $L = 100000$m, $h = 100$m, $f = 0.0001$, $\Delta t = 5$s and $\lambda = 2500$. Figure 5.1 shows the initial solution at $t = 0$. The mesh used in the model is displayed along side. Figure 5.2 shows the solution after 2000 seconds, 400 time steps. The exact solution is displayed along side for comparison. Qualitatively, the approximate solution is similar to the exact solution. Close inspection shows that the isolines do not match perfectly. Figure 5.3 shows the solution after 3000 seconds, 600 time steps. Numerical errors can clearly be seen in the bottom right of the plot. In these plots, the exact solution is evaluated at the same points as the approximated solution. The boundary of the exact solution is therefore defined by the boundary of the finite element approximation. The boundary of the exact solution may therefore

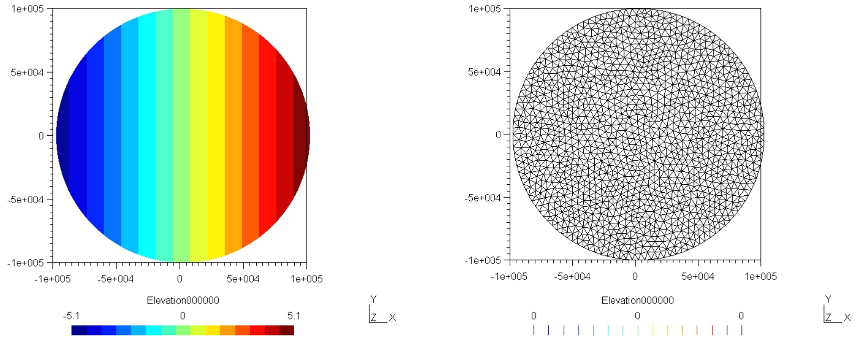

Figure 5.1: This is the planar solution plotted at $t = 0$. The plot on the right shows the initial mesh.

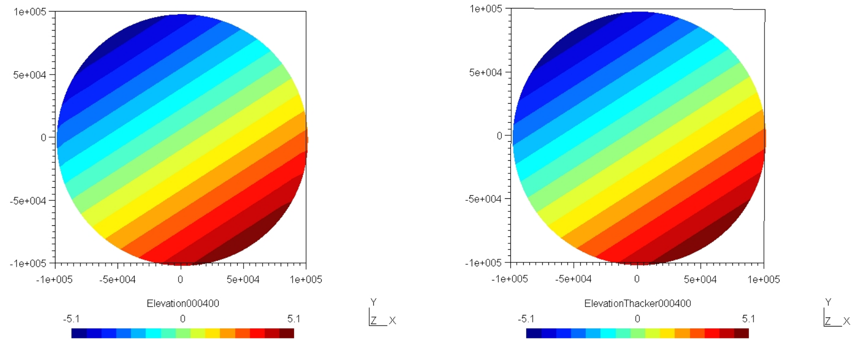

Figure 5.2: This is the planar solution plotted at $t = 2000$. The plot on the left shows the solution approximated using the ALE method. The plot on the right shows the exact solution.

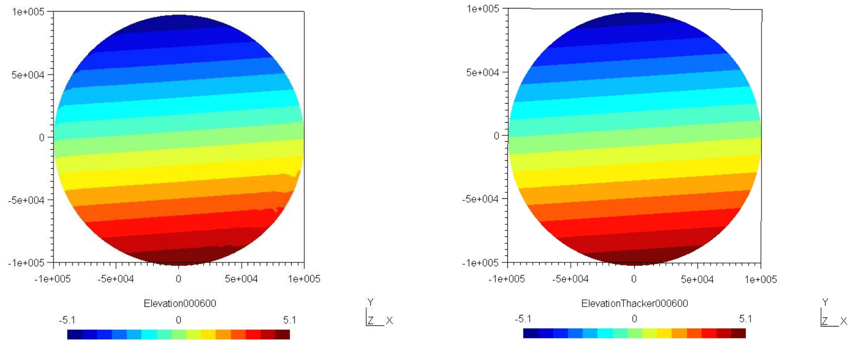

Figure 5.3: This is the planar solution plotted at $t = 3000$. The plot on the left shows the solution approximated using the ALE method. The plot on the right shows the exact solution.

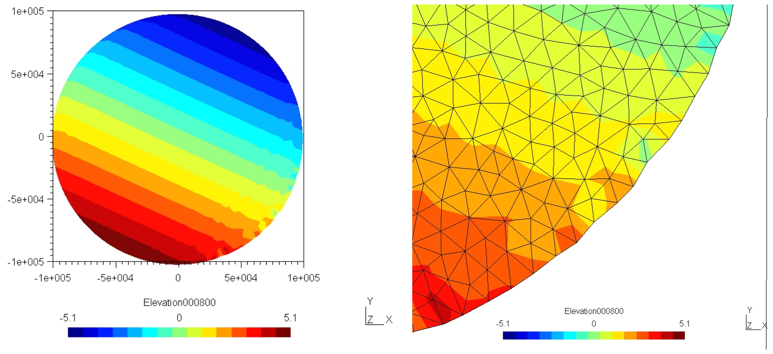

Figure 5.4: This is the planar solution plotted at $t = 4000$. The plot on the left shows the solution approximated using the ALE method. The plot on the right shows a close up of the mesh losing its shape at the boundary.

be incorrect. Figure 5.4 shows the solution after 4000 seconds, 800 timesteps. The numerical errors have increased and the structure of the mesh has become inconsistent. The left hand plot shows a close up of the boundary, the boundary points are no longer equidistant from the centre of the domain. The solution blows up before $t = 1000$.

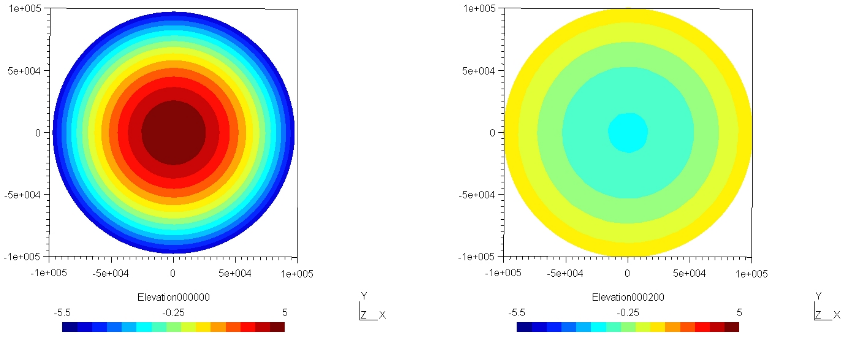

Figure 5.5: This is the curved surface solution plotted at $t = 0$ on the left, and $t = 2000$s on the right.

## 5.2 Curved surface

In this section, the results from the ALE method described in chapter 4 are compared with the exact curved surface solution described in section 2.4. Throughout this section $L = 100000$m, $h = 100$m, $f = 0.0001$, $\Delta t = 10$s and $\lambda = 5$. Figure 5.5 shows the numerically approximated solution at $t = 0$s and $t = 2000$s. During this time interval the surface of the fluid has gone from being convex to concave. Figure 5.6 shows the solution at $t = 4000$s. The numer-

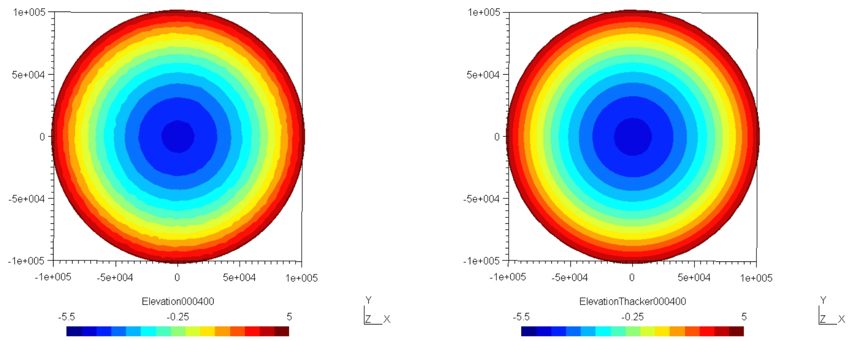

Figure 5.6: This is the curved surface solution plotted at $t = 4000$s. The plot on the left shows the solution approximated using the ALE method. The plot on the right shows the exact solution.

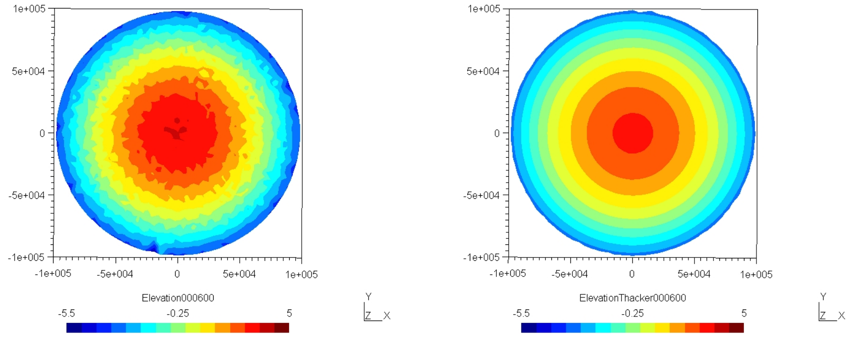

Figure 5.7: This is the curved solution plotted at $t = 6000$. The plot on the left shows the solution approximated using the ALE method. The plot on the right shows the exact solution.

ical approximation is presented alongside the exact solution. The numerical errors can be observed in the isolines of the numerical solution. Figure 5.7 shows the solution at $t = 6000$s. The numerical approximation is becoming swamped by the numerical errors. The solution blows up before $t = 7000$s.

## 5.3 Error analysis

In the previous section it was shown that the ALE method used in this project approximated the solution for a short time before blowing up. Figures 5.8 and 5.9 are plots of the local error, they show the absolute value of the difference between the ALE approximated solution and the exact solution. The plots are taken from the planar surface solution, using the same parameters as the previous section. Figure 5.8 shows the difference after 100 seconds, with a timestep of 10 seconds. Figure 5.9 shows the difference after 100 seconds, with a timestep of 10 seconds. The error is greatest at the boundaries, in the direction orthogonal to $\frac{\partial \eta}{\partial t} = 0$. The scaling of the error is approximately twice as large in figure 5.8 than in 5.9. So the error is of $O(\Delta t)$. This is the order of error to be expected when using Euler's method for the time integration of the mesh velocities.

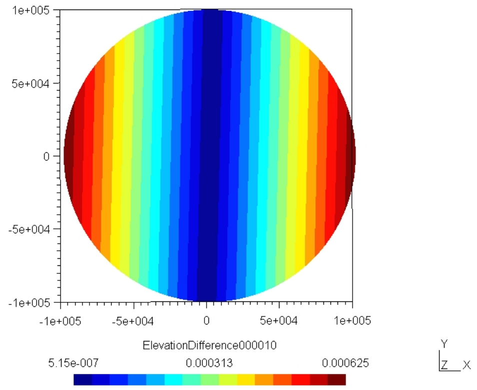

Figure 5.8: A plot of the local error at $t = 100$s with a timestep of $\Delta t = 10$s. The scale range is $5.15 \times 10^{-7}$m (blue) to $6.25 \times 10^{-4}$m(red). The maximum relative error is 0.0125%

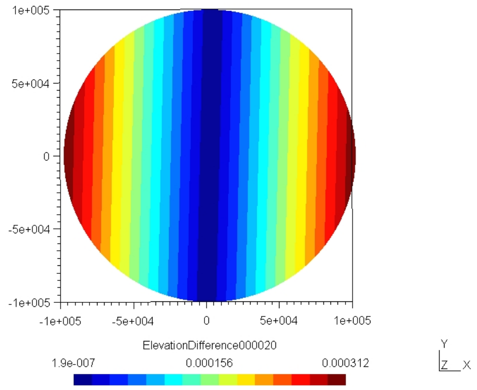

Figure 5.9: A plot of the local error at $t = 100$s with a timestep of $\Delta t = 5$s. The scale range is $1.9 \times 10^{-7}$m (blue) to $3.125 \times 10^{-4}$m(red). The maximum relative error is 0.00625%

# Chapter 6

# Investigation

## 6.1 Errors at the boundary

The results in chapter 5 show that numerical errors are introduced at the boundaries that cause the solution to blow up. In this section, the cause of these numerical errors is investigated. The investigation is conducted on the planar surface solution using the same parameter values as section 5.1. The velocities in the planar surface solution vary only in time, this means that there is no vorticity and the mesh velocity must therefore be equal to the physical velocity.

Figure 6.1 shows the initial velocity field $\boldsymbol{u}$ at $t = 0$ and the mesh velocities $\boldsymbol{q}$ calculated at the first time step. The mesh velocities $\boldsymbol{q}$ are therefore calculated from the initial values of $\eta$ and $\boldsymbol{u}$ which can have no numerical error. The mesh velocities are exactly equal to the physical velocities. No significant error is introduced when deriving the mesh velocities in the first timestep.

The magnitude of the velocity $\boldsymbol{u}$ should remain constant in time. Figure 6.2 shows the velocity field $\boldsymbol{u}$ at $t = \Delta t$. The right hand plot

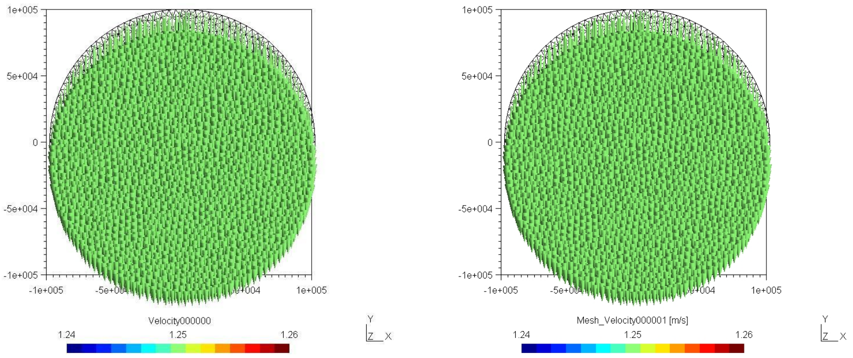

Figure 6.1: This plot shows, on the left, the velocity $\boldsymbol{u}$ at $t = 0$ and, on the right, the mesh velocity $\boldsymbol{q}$ at $t = \Delta t$.

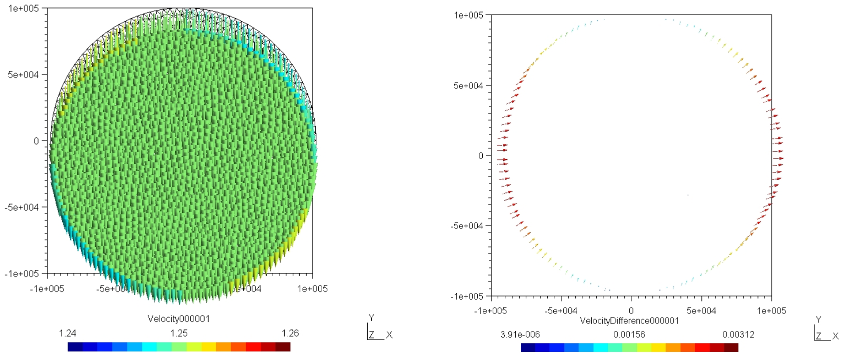

Figure 6.2: This plot shows, on the left, the velocity $\boldsymbol{u}$ at $t = \Delta t$ and, on the right, the difference between the exact velocity and the approximated velocity at $t = \Delta t$. The maximum relative error is 0.25%.

shows the difference between the numerically approximated velocities and the exact velocities. Clearly there is significant numerical error introduced during the time integration of the momentum equation. The errors are introduced along the boundary as velocity components pointing approximately in the direction of the gradient of $\eta$.

Figure 6.3 shows the same plots as figure 6.2. The only difference is that the mesh has been rotated through 90° beforehand. The mesh was rotated before the initial conditions were specified and can therefore have no effect on the physical problem. Figure 6.3 shows that there is still significant numerical error around the boundary, however, the position of the greatest numerical error has rotated through 90°. This suggests that the numerical errors are greatly influenced by the quality of the mesh.

The initial conditions being used for the investigation mean that there should be no vorticity in the fluid motion. The mesh velocity $\boldsymbol{q}$ derived at $t = n\Delta t$ should therefore be the same as the velocity $\boldsymbol{u}$ at $t = (n-1)\,\Delta t$. The advection terms in equation (4.39) should be identical and can therefore be removed without effecting the solution

$$\frac{\partial \boldsymbol{u}}{\partial t} + f\boldsymbol{k} \times \boldsymbol{u} = -g\nabla\eta \tag{6.1}$$

Figure 6.4 shows the velocity $\boldsymbol{u}$ at $t = \Delta t$ with the advection terms removed, using the same mesh as 6.2. The errors introduced at the boundary are identical to figure 6.2. The numerical errors are therefore introduced as a result of the time integration of the Coriolis term or the pressure gradient term in the momentum equation.

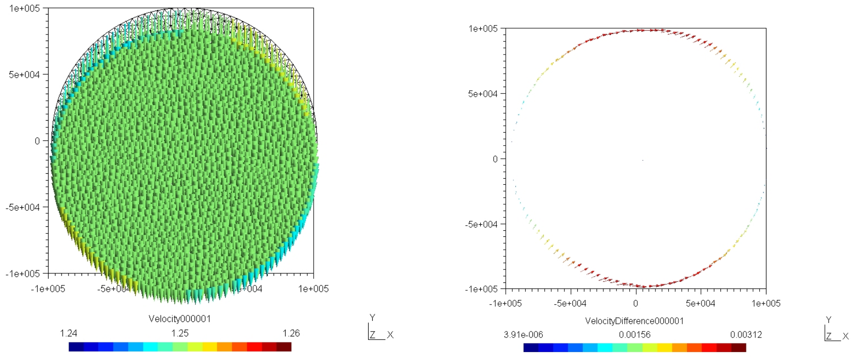

Figure 6.3: This plot shows, on the left, the velocity $\boldsymbol{u}$ at $t = \Delta t$ and, on the right, the difference between the exact velocity and the approximated velocity at $t = \Delta t$. The mesh has been rotated by 90° prior to simulation. The maximum relative error is 0.25%.

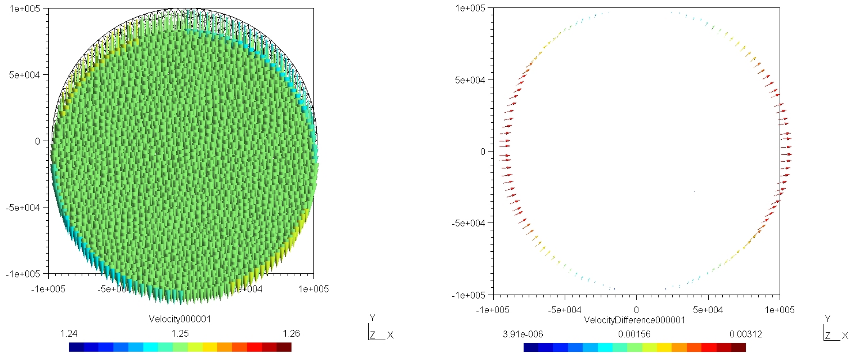

Figure 6.4: This plot shows, on the left, the velocity $\boldsymbol{u}$ at $t = \Delta t$ and, on the right, the difference between the exact velocity and the approximated velocity at $t = \Delta t$. The advection terms have been removed. The maximum relative error is 0.25%.

## 6.2 Conclusion

The investigation conducted identifies significant numerical errors appearing at the boundaries. The errors appear immediately and grow quickly, causing the solution to blow up before completing a full oscillation in both test cases. In any numerical scheme it is expected that errors will build up as a result of the approximation of derivatives. It is also the case that the boundaries are where the error is likely to have the greatest effect. Specifically to the problem modelled in this project, the bathymetry also has a greater effect on the error at the boundary. The error is dependent on the gradient of $h$ which is steepest at the boundaries. It has been identified that the quality of the mesh used in this project has a great bearing on the build up of numerical errors. Figure 6.5 shows a pair of boundary elements from the mesh used in this project. These elements are significant in that they are boundary elements that share a segment with other boundary elements. During the investigation it was observed that it was elements such as these where the maximum error would occur. The node on the boundary is only supported by the two elements it connects. It is concluded that greater resolution around the boundary, and particular avoidance of the element structure shown in 6.5, is required to improve the numerical accuracy of the scheme.

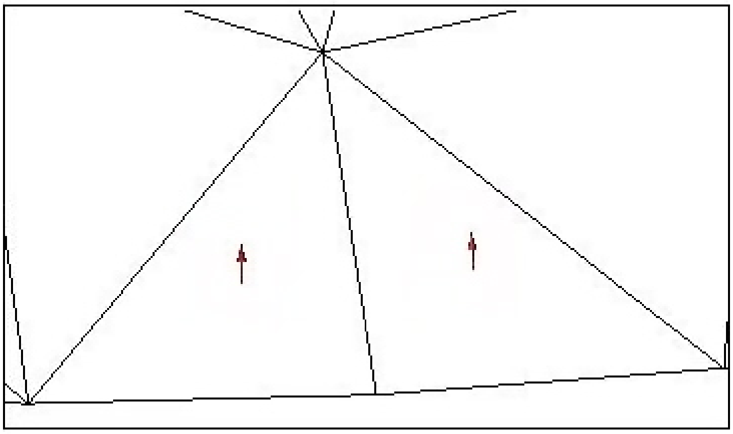

Figure 6.5: The two elements that contain the red arrows are typical 'problem' elements. The edges of the elements on the lower side of the diagram make up the boundary.

# Chapter 7

# Summary and further work

## 7.1 Summary

In this dissertation, an ALE moving mesh method has been applied to a finite element scheme for the shallow water equations. The moving mesh method is based on a conservation of mass principle. In chapter 2, some exact solutions to the shallow water equations in a parabolic basin were presented, along with a scheme for the numerical integration of the system of ordinary differential equations obtained from the general solution. Chapter 3 introduced an existing finite element method for the shallow water equations. The shallow water equations were put into nondimensional form and the finite element discretization was discussed. In chapter 4, a moving mesh method was proposed based on conserving the volume of water within a patch of elements. The implementation of this method within the finite element scheme was explained. Chapter 5 showed some numerical results and compared them with the results obtained from the exact solutions in chapter 2. In chapter 6, an investigation was conducted into the numerical errors arising at the boundary. Conclusions were drawn on the cause of these errors.

## 7.2 Further investigation of boundary errors

Further investigation into the cause of the errors at the boundaries could be conducted. Generation of a mesh with greater concentration of elements at the boundary would be desirable.

## 7.3 Comparison with other solutions

Thacker, [1], presented some other solutions not discussed in this dissertation. In particular, solutions were presented for an elongated parabolic basin, or 'canal', where $L >> l$, and a parabolic flood wave, where the bathymetry is a planar surface, $h = 0$. With a fully working model, these different solutions can be investigated. The numerically integrated solutions from chapter 2 could also be used for comparison with different kinds of motion.

## 7.4 Adding vorticity to the mesh movement

In section (4.3.4) the mesh velocities were specified to have zero vorticity, $\nabla \times \boldsymbol{q} = 0$. This may be undesirable in flows with high vorticity, where the mesh may become tangled. In this section, the idea of applying vorticity to the mesh movement is proposed. A velocity function $\boldsymbol{s}(x, y, t)$ is specified such that $\nabla \times \boldsymbol{q} = \nabla \times \boldsymbol{s}$. Equation 4.11 becomes

$$\boldsymbol{q} = \nabla \phi + \boldsymbol{s} \tag{7.1}$$

and equation 4.12 becomes

$$\int_{\Omega(t)} H \nabla \phi . \nabla \omega d\Omega = - \int_{\Omega(t)} \omega \nabla . \left( H \boldsymbol{u} \right) d\Omega + \oint \omega H \boldsymbol{s} . \boldsymbol{n} d\Gamma \tag{7.2}$$

$$+ \int_{\Omega} H \boldsymbol{s} . \nabla \omega d\Omega$$

Given $\boldsymbol{s}$, equation 7.2 can be solved for $\phi$ in the same way as before. The mesh velocities must be derived from equation 7.1

$$\int_{\Omega} \omega \boldsymbol{q} d\Omega = \int_{\Omega} \omega \nabla \phi d\Omega + \int_{\Omega} \omega \boldsymbol{s} d\Omega \tag{7.3}$$

The mesh vorticity $\boldsymbol{s}$ can be chosen to be any function of $(x, y, t)$. An interesting development would be to make $\boldsymbol{s}$ equal to minus the vorticity

$$\boldsymbol{s}\left(x, y, t\right) = - \left( \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) \tag{7.4}$$

Choosing this function for $\boldsymbol{s}$ should prevent tangling of the mesh due to the vorticity of the flow.

# Bibliography

[1] W.C. Thacker. Some exact solutions to the nonlinear shallow-water wave equations *J. Fluid Mechanics,* **107** (1981).

[2] F.K. Ball. An exact theory of simple finite shallow water oscillations on a rotating earth *Hydraulics and Fluid Dynamics,* pp. 293-305. Macmillan (1964).

[3] Emmanuel Hanert, Daniel Y. Le Roux, Vincent Legat, Eric Deleesnijder. An efficient Eulerian finite element method for the shallow water equations, *Ocean Modelling* **10** (2005).

[4] Emmanuel Hanert, Daniel Y. Le Roux, Vincent Legat, Eric Deleesnijder. Advection schemes for unstructured grid ocean modelling, *Ocean Modelling* **7** (2004).

[5] Emmanuel Hanert, Daniel Y. Le Roux, Vincent Legat, Eric Deleesnijder. Impact of mass lumping on gravity and Rossby waves in 2D finite-element shallow-water models, *Int. J. Numer. Meth. Fluids* (2007-Awaiting publication).

[6] P. Houston, C. Schwab, E. Suli. Discontinuous *hp* element methods for advection diffusion problems, *Tech. Rep. NA-00/15, Oxford University* **7** (2000).

[7] Dale R. Durran. The third order Adams Bashforth method: An attractive alternative to Leapfrog time differencing, *Monthly Weather Review* **119** (1990).

[8] M. J. Baines, M. E. hubbard, P. K. Jimack. A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries, *App. Num. Maths.* **54** (2005).

[9] M. J. Baines, M. E. hubbard, P. K. Jimack, A.C. Jones. Scale-invariant moving finite elements for nonlinear partial differential equations in two dimensions, *App. Num. Maths.* **56** (2006).

[10] K. W. Blake. Moving Mesh Methods for Non-Linear Parabolic Partial Differential Equations, *PhD Thesis* (2001).

[11] J. Donea, Antonio Huerta, J.-Ph. Ponthot, A.Rodriguez-Ferran. Arbitrary Lagrangian-Eulerian Methods, *Encyclopedia of computational mechanics. Vol 1. Ch 14. Wiley* (2004).

[12] M. D. Piggott, C. C. Pain, G. J. Gorman, P. W. Power, A. J. H. Goddard. h, r, and hr adaptivity with applications in numerical ocean modelling, *Ocean Modelling* **10** (2005).