SOME SOLUTIONS OF THE SHALLOW WATER WAVE EQUATIONS

Joe Sampson

A Thesis Presented for the Degree of Doctor of Philosophy

Mathematics Discipline
Faculty of Engineering and Industrial Sciences
Swinburne University of Technology
Melbourne, Australia

2008

Declaration

The candidate hereby declares that the work in this thesis, presented for the award of the degree of Doctor of Philosophy submitted in the Mathematics Discipline, Faculty of Engineering and Industrial Sciences, Swinburne University of Technology:

is that of the candidate alone and has not been submitted previously, in whole or in part, in respect of any other academic award and has not been published in any form by any other person except where due reference is given, and

has been carried out during the period from March 1999 to March 2007 under the supervision of Professor Alan Easton and Dr Manmohan Singh.

Joe Sampson

Certification

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Professor Alan Easton

Dr Manmohan Singh

COPYRIGHT

Author: Joe Sampson

Title: Some solutions of the shallow water wave equations

Discipline: Mathematics

Faculty: Engineering and Industrial Sciences

Degree: Ph.D.

Joe Sampson

Date: May 2008

Abstract

This thesis is concerned with the solution of the shallow water wave equations. To solve the equations for a real life domain of flow a numerical solution is required. Analytical solutions are useful for testing numerical solutions.

Previous research in this area is examined. Existing analytical solutions of the nonlinear shallow water equations involving moving boundaries do not include bottom friction. Existing analytical solutions of the nonlinear shallow water equations involving fixed boundaries can be modified. The SLM (Selective Lumped Mass) scheme for numerically solving the shallow water equations has been used to accurately model tides both in bays with fixed boundaries and bays with moving boundaries. The SLM scheme for domains with moving boundaries has not been fully explained in published papers.

Analytical solutions are established for moving boundary shallow water equations involving nonlinear continuity for unforced frictional flow in parabolic canals, circular paraboloids and elliptical paraboloids, and both for forced frictional flow and unforced frictionless flow in a parabolic canal. Analytical solutions are established for one dimensional nonlinear frictionless shallow water wave flow in a basin of constant depth, with a fixed boundary and with a sinusoidal input at the open boundary.

The SLM scheme is coded in Visual C ++ and validated against an analytical solution, a convergence study is carried out for the SLM scheme and a computer program is written in Visual C ++ to generate finite element meshes. The SLM scheme is used to model one dimensional nonlinear shallow water flow in a basin of constant depth, with fixed boundary and with sinusoidal input at the open boundary; results obtained were close to those in the analytical solution.

The SLM scheme when applied to a moving boundary domain has been modified, with the resultant scheme used to model forced frictionless flow and forced frictional flow in a parabolic canal, with results close to those of the analytical solutions developed.

The SLM scheme is applied to model accurately the existing tidal heights and currents in Port Phillip Bay, Victoria, Australia. The SLM scheme is applied to model the effect of proposed channel deepening on the tides in Port Phillip Bay; the effect is small, increasing the tidal heights by at most seven millimetres.

The conclusion of this thesis is that successful analytical and numerical solutions of the shallow water equations are developed both for domains with fixed boundaries and domains with moving boundaries.

Acknowledgments

I would like to thank Alan Easton, my main supervisor, for the extensive help that he has given me throughout my work on this thesis, giving up many hours to help me. Even after he moved from Australia to Papua New Guinea he continued to be my supervisor, responding quickly to my emails. I am also grateful to Alan for letting me use numerous Mathematica programs that he wrote. I am thankful to Alan for introducing me to finite element tidal modelling and analytical solutions of the shallow water wave equations.

I am also thankful to my other supervisor, Manmohan Singh, for the many hours of help that he gave me, providing many useful ideas.

Finally I thank Swinburne University of Technology for time release and computational facilities.

Contents

List of Figures

List of Tables

Table of symbols

English letters

a constant

a_1 amplitude of the water elevation

a_j amplitude of tidal constituent j

a_α constant coefficients of basis functions at node α

A constant

$\mathbf{A}$ element matrix

$\mathbf{A}^e$ global mass matrix

b_2 constant

b_α coefficients of x for basis functions at node α

B constant

$\mathbf{B}$ global mass matrix

$\mathbf{B}^e$ element matrix

c_α coefficients of y for basis functions at node α

c wavespeed

c_1 constant

C constant

d_2 constant

d_m function of s_r

D constant

$\mathbf{D}$ global mass matrix

$\mathbf{D}^e$ element matrix

E constant

E the number of elements in domain Ω

f Coriolis parameter

f_1 function of x

F constant

g acceleration due to gravity

g_1 function of x

G constant

h depth of bed below datum

h_0 constant

$\mathbf{h}$ vector of depth below datum

h_α depth of bed below datum at node α

$\mathbf{h}_{av}$ average node depth vector

$\mathbf{h}^e_{av}$ average node depth element vector

H total fluid depth

H_α total depth at node α

$\mathbf{H}$ global mass matrix

$\mathbf{H}^e$ element matrix

j $\sqrt{-1}$

j counter

k constant

K constant

K wind stress parameter

K_1 tidal constituent

L constant

$\mathbf{L}$ global mass matrix

$\mathbf{L}^e$ element matrix

m constant

M_2 tidal constituent

M constant

$\mathbf{M}$ global mass matrix

$\mathbf{M}^e$ element matrix

$\bar{\mathbf{M}}$ lumped mass matrix

$\bar{\mathbf{M}}^e$ lumped mass element matrix

$\widetilde{\mathbf{M}}$ selective lumped mass matrix

$\widetilde{\mathbf{M}}^e$ selective lumped mass element matrix

n constant

n Manning's constant

N the total number of nodes in domain Ω

N_2 tidal constituent

$\mathbf{N}$ global mass matrix

$\mathbf{N}^e$ element matrix

O_1 tidal constituent

p parameter for flow in a circular paraboloid

p_1 function of x

p_2 function of x

p_3 function of x

P constant

$\mathbf{P}$ global mass matrix

q constant

q_1 constant

Q constant

$\mathbf{Q}$ global mass matrix

r constant

R constant

s constant

s_r selective lumping parameter

S constant

S_2 tidal constituent

$\mathbf{S}^e$ element matrix

t time

T period

$\mathbf{T}^e$ element matrix

u velocity in the x direction

u_0 a function of t

U_α U-velocity at node α

U vertically averaged velocity in the x direction

$\dot{\mathbf{U}}$ vector of acceleration in the x direction

v velocity in the y direction

v_0 a function of t

V vertically averaged velocity in the y direction

V_α V-velocity at node α

$\dot{\mathbf{V}}$ vector of acceleration in the y direction

w weight function

W wind speed 10 m above the water surface

x Eastward space coordinate

$\mathbf{X}^e$ element matrix

y Northward space coordinate

$\mathbf{Y}^e$ element matrix

$\dot{\mathbf{z}}$ vector of the time derivative of the water level

$\dot{\mathbf{z}}^e$ element vector of the time derivative of the water level

Greek letters

α node number

β counter

γ constant

γ counter

γ_j phase of tidal constituent j

δt time step

δx space step in x-direction

δy space step in y-direction

Δ^e area of a triangular element

ζ water surface elevation above a reference datum

ζ_0 a function of t

ζ_1 a function of t

ζ_2 a function of t

ζ_α water level at node α

ζ_S water level specified on the open sea boundary

κ constant

λ constant

ν horizontal eddy viscosity coefficient

τ bottom friction parameter

τ^e bottom friction term for element

τ_L land boundary of Ω, domain of flow

τ_S sea boundary of Ω, domain of flow

ϕ angle of the wind direction

ϕ_α basis function

ω angular frequency

ω_j angular frequency of tidal constituent j

ψ constant

Ω constant

Ω two dimensional domain of flow

Chapter 1

Summary of the thesis

This thesis develops new solutions of the shallow water wave equations, some analytical and some numerical.

Shallow water wave models are applicable where the water depth is much less than the horizontal scale of motion. They are used to represent the flow of water waves in coastal seas and estuaries (e.g. tides or tsunamis or storm surges). Such models are used to predict the water velocity and water height at various points within a region of flow at different times during a long period of time (e.g. model a tide in a bay every 100 seconds over a period of 24 hours).

Shallow water wave models are simultaneous nonlinear partial differential equation models representing conservation of mass and conservation of momentum. In some circumstances the nonlinear terms can be left out of the equations. Exact solutions of the shallow water wave models have been found only for a small number of domains of flow. All such domains of flow have simple geometric shapes. As real life domains of flow do not have simple geometric shapes, analytical (exact) solutions cannot be found for such domains. For domains of flow without exact solutions the equations must be solved numerically (i.e. approximately). One technique is to use the finite element method in space and the finite difference method in time. Another technique is to use the finite difference method in space and in time. Analytical solutions for domains of flow with simple geometric shapes are very useful for testing numerical solutions as they give the height and velocity for every point in the domain and every time, whereas for real life domains there is only limited experimental data.

One aim of this thesis is to develop new analytical solutions of the nonlinear shallow water wave equations both for domains with moving boundaries and for domains with fixed boundaries. Another aim of this thesis is to develop a numerical model to accurately model nonlinear shallow water wave flow in domains with fixed boundaries and domains with moving boundaries.

There exist analytical solutions of both linear and nonlinear shallow water wave equations for domains with fixed boundaries. Also, there exist analytical solutions of both linear and nonlinear moving boundary shallow water wave equations. None of the moving boundary solutions contain friction.

Some moving boundary analytical solutions of the nonlinear shallow water wave equations involving nonlinear continuity and linear friction for unforced flow in a two dimensional circular paraboloidal basin, a two dimensional elliptical paraboloidal basin and a one dimensional parabolic canal will be developed in this thesis. Also, some moving boundary analytical solutions of the nonlinear shallow water wave equations involving nonlinear continuity and both for no friction and linear friction for forced flow in a one dimensional bed with quadratically varying depth will be obtained. Also, some analytical solutions of the one dimensional nonlinear shallow water wave equations for a basin with a horizontal bed and constant rectangular cross section will be developed in this thesis; the basin is closed at one end and at the other end has an open sea boundary, at which there is sinusoidal forcing.

The numerical method that is used in this thesis is the Selective Lumped

Mass (SLM) model, developed by Kawahara, Hirano and Tsubota [46]. The SLM model can be applied both to domains with fixed boundaries and domains with moving boundaries. With the moving boundary scheme some elements are wet for part of the time and dry for part of the time. The SLM method has the advantage of giving explicit solutions of the shallow water wave equations. It has the disadvantage of being subject to node to node oscillations [29]. However, these oscillations can be minimised by using elements with sides that are small compared with the wavelength [29]. The SLM model is implemented in this thesis in a manner that the quadratic friction term does not become infinite at a moving shoreline. In addition, a wetting and drying scheme for domains with moving boundaries has been developed.

The SLM model is coded in Visual C++ and validated both against analytical solutions for flow in a basin of constant depth and in a basin of linearly varying depth. Convergence studies for a basin of constant depth have already been published [29, 46]. In this thesis a convergence study of the SLM model is carried out for a rectangular basin of linearly varying depth using analytical solutions of the one dimensional linear shallow water wave equations. The convergence study uses a computer program written in Visual C++ to generate meshes. A wetting and drying scheme is applied to both moving boundary forced frictionless flow and moving boundary forced linear frictional flow above a bed with quadratically varying depth. The results are compared with the results from analytical solutions developed in this thesis.

The SLM model is applied to model accurately the existing tidal heights and tidal currents in Port Phillip Bay, Victoria, Australia; the SLM model is also applied to model the effects of proposed channel deepening on the tides in Port Phillip Bay.

Computer programs written in Visual C++ or Mathematica by the author of this thesis are used extensively as well as some Mathematica packages written by Alan Easton, the main supervisor of the thesis.

The structure of this thesis is outlined below. Chapter 2 presents the shallow water equations, and then reviews existing analytical and numerical solutions of these equations which are related to this thesis. Chapter 3 presents some new analytical solutions of the nonlinear shallow water equations for domains with moving boundaries. Chapter 4 discusses some new analytical solutions of the nonlinear shallow water equations for domains with fixed boundaries. Chapter 5 discusses the SLM (selective lumped mass model) in detail plus the results of a convergence study of the SLM. Chapter 6 discusses a wetting and drying scheme which is a modification of the scheme of Kawahara, Hirano and Tsubota; the new scheme has been applied to moving boundary forced frictional flow in a bed of quadratically varying depth. The results are compared with the results from analytical solutions developed in this thesis. Chapter 7 presents the SLM method to model the existing tidal heights and tidal currents in Port Phillip Bay. The SLM model is applied to model the effects of proposed channel deepening on the tides in Port Phillip Bay. In Chapter 8 the conclusions of this thesis are presented with suggestions for further research.

Chapter 2

Shallow water wave models

2.1 Introduction

Shallow water wave models are applicable where the water depth is much less than the horizontal scale of motion. Shallow water models are also called long wave models as the wavelength is long compared with the depth. They are used to represent the flow of water waves in coastal seas and estuaries (e.g. of tides or tsunamis or storm surges). They can also be used to model lake flows. Such models are used to predict the water velocity and water height at various points within a region of flow at different times. The time period of interest for a tidal model is 12 to 24 hours and the variation over a month or a season may be important. The time period of interest for tsunamis is usually 15 to 30 minutes, while the period of interest for storm surges is several days.

Shallow water wave models are important because they describe real life situations that are critical to large numbers of people. For example, a knowledge of tides and tidal currents is important for shipping. For example, an estimate of the effect of the proposed deepening of some shipping channels in Port Phillip Bay, Victoria, Australia on tides and storm surges is important for neighbouring communities. A knowledge of water levels and currents in bays and estuaries is important for an understanding of the local environment (e.g. the aftermath of an oil spill) and the wellbeing of coastal communities. Shallow water wave models are one way of predicting tides and tidal currents in a given coastal region. Global warming could inundate coastal areas (e.g. near Townsville, Australia) and knowledge of the present low and high tides would help in understanding how warming would affect sea levels in a region. Another example is tsunamis (often incorrectly called tidal waves), which can be caused by earthquakes or underwater volcanoes. These can kill large numbers of people. If these are predicted, lives can be saved. One might also want to know in hindsight the effects of a tsunami, e.g. the 2004 tsunami in the Indian Ocean. Or one might want to forecast what might happen if a tsunami were to arrive at a certain coastal area. Another example is storm surges. In Bangladesh many people are killed by storm surges (increase of water level caused by storms) as the sea inundates low lying land. This is a substantial problem for the government of Bangladesh and they need more information about these surges.

In this chapter the shallow water equations, which are used to model shallow water flow, will be discussed. This chapter contains a review of literature that is relevant to this thesis. First there is a review of some of the analytical solutions of the shallow water wave equations both for domains with fixed boundaries and domains with moving boundaries. Secondly, there is a review of some of the numerical solutions of the shallow water wave equations both for domains with fixed boundaries and domains with moving boundaries, with particular emphasis on the latter.

2.2 The shallow water wave equations

The two-dimensional depth-averaged shallow water wave equations are a set of nonlinear partial differential equations [52, 108]. In Cartesian form they consist of the conservation of momentum equation in the East direction

$$\frac{\partial U}{\partial t}+U\frac{\partial U}{\partial x}+V\frac{\partial U}{\partial y}-fV-\nu\left(\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}\right)+\tau U+g\frac{\partial \zeta}{\partial x}-\frac{KW^2\cos\phi}{H}=0, \tag{2.1}$$

the conservation of momentum equation in the North direction

$$\frac{\partial V}{\partial t}+U\frac{\partial V}{\partial x}+V\frac{\partial V}{\partial y}+fU-\nu\left(\frac{\partial^2 V}{\partial x^2}+\frac{\partial^2 V}{\partial y^2}\right)+\tau V+g\frac{\partial \zeta}{\partial y}-\frac{KW^2\sin\phi}{H}=0, \tag{2.2}$$

and the continuity (conservation of mass) equation

$$\frac{\partial \zeta}{\partial t}+\frac{\partial(h+\zeta)U}{\partial x}+\frac{\partial(h+\zeta)V}{\partial y}=0, \tag{2.3}$$

where $\zeta(x,y,t)$ is the height of the water surface above a horizontal datum, $z=-h(x,y)$ is the bottom surface, $H(x,y,t)=h(x,y)+\zeta(x,y,t)$ is the total depth of the fluid, f is the Coriolis parameter, W is the wind speed 10 m above the water surface, K is a function of W, $u(x,y,z,t)$ is the East component of the water velocity, $U(x,y,t)$ is the depth averaged velocity component of the water current to the East defined by $U=\int_{-h}^{\zeta} u\,dz$, $v(x,y,z,t)$ is the North component of the water velocity, $V(x,y,t)$ is the depth averaged velocity component of the water current to the North defined by $V=\int_{-h}^{\zeta} v\,dz$, ϕ is the angle of the wind direction measured anticlockwise from the East, τ is the water bed friction parameter, ν is the horizontal eddy viscosity, g is the acceleration due to gravity and t is the time. The friction parameter, τ, is either given the value zero for frictionless flow, a constant positive value for linear friction or for quadratic friction

$$\tau=\frac{n^2 g\sqrt{U^2+V^2}}{H^{4/3}}, \tag{2.4}$$

where n is Manning's coefficient of bottom roughness. Equation (2.4) is based on Manning's formula for open-channel flow [15]. The advective (convective) terms are the terms $U\frac{\partial U}{\partial x}$, $V\frac{\partial U}{\partial y}$, $U\frac{\partial V}{\partial x}$ and $V\frac{\partial V}{\partial y}$, the Coriolis terms are the terms $-fV$ and fU, the viscosity terms are the terms $-\nu(\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2})$ and $-\nu(\frac{\partial^2 V}{\partial x^2}+\frac{\partial^2 V}{\partial y^2})$ and the wind stress terms are the terms $-\frac{KW^2\cos\Phi}{H}$ and $-\frac{KW^2\sin\Phi}{H}$.

A vertical cross-section of a body of shallow water is shown in Figure 2.1. Sometimes the horizontal datum is taken to be mean sea level at a certain place. Sometimes the datum is taken to be the geoid, the equipotential surface of the Earth's gravity field which best fits, using the least squares method, global mean sea level. Solution of the shallow water wave equations (2.1), (2.2) and (2.3) is sought for $\zeta(x,y,t)$, $U(x,y,t)$ and $V(x,y,t)$ in a two dimensional domain Ω. If the domain is the datum surface for a bay, the boundary τ comprises two parts: the land boundary, τ_L, and the open sea boundary, τ_S. The shallow water wave equations can also be written in polar form [67] and in spherical form [50, 52]. The shallow water equations (2.1), (2.2)and (2.3) are called the primitive equations. Sometimes the continuity equation is replaced by a second order equation called the WCE (Wave Continuity Equation) [69], or the GWCE (Generalised Wave Continuity Equation) [49], which is a modification of the WCE.

Exact solutions of the shallow water wave equations are known only for a small number of domains of flow, in some cases for linearised forms of the equations; all such domains of flow have simple geometric shapes. These models are discussed in Section 2.3. As real life domains of flow do not have simple geometric shapes, analytical (exact) solutions cannot be found for such domains. For these domains of flow the equations must be solved numerically (i.e. approximately). These models are discussed in Section 2.4. Although analytical solutions have been developed only for domains of

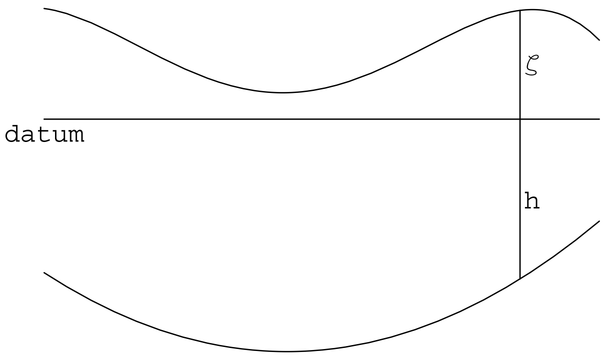

Figure 2.1: Shallow water flow cross section

flow with simple geometric shapes they are very useful for testing numerical solutions as they solve for the height and velocity of the wave for every point in the domain and every time, whereas for real life domains there is only limited experimental data. Conversely, numerical models are also useful in validating analytical models.

2.3 Analytical solutions

Analytical solutions of the shallow water wave equations exist both for domains with fixed boundaries and domains with moving boundaries. Some of these solutions are for linearised forms of the shallow water wave equations while others are for the nonlinear shallow water wave equations.

The linearised equations used ignore the advective (convective) terms (i.e. the terms $U\frac{\partial U}{\partial x}$, $V\frac{\partial U}{\partial y}$, $U\frac{\partial V}{\partial x}$ and $V\frac{\partial V}{\partial y}$ in the momentum equations), the Coriolis terms, the viscosity terms and the wind stress terms. They assume that the oscillations of the water surface, ζ, are small compared to the total depth, $h + \zeta$, and hence that the nonlinear continuity terms $\frac{\partial(h+\zeta)U}{\partial x}$ and $\frac{\partial(h+\zeta)V}{\partial y}$ are respectively replaced with the linear continuity terms $\frac{\partial(hU)}{\partial x}$ and $\frac{\partial(hV)}{\partial y}$. They also use a linear friction term.

2.3.1 Fixed boundary analytical solutions

Lamb [55] solved linearised forms of the shallow water wave equations to model tidal flow in canals of different shapes for frictionless flow. Ippen [39] solved linearised forms of the shallow water wave equations to model tidal flow in canals of different shapes both for frictionless flow and linear frictional flow. Lynch and Gray [67] developed analytical solutions of the linearised shallow water wave equations. The equations included linear friction and a wind stress term. Solutions obtained included for the case where there is no wind stress. The domains considered were of constant depth and linearly varying depth and quadratically varying depths with cartesian and polar geometries. The domains were periodically forced. The authors stated that their analytical solutions should prove useful for comparison with numerical models. The advantage of these analytical solutions over previous analytical solutions is that most previous analytical solutions were for frictionless flow and where they were for frictional flow were for simple cases. One of Lynch and Gray's analytical solutions will be used in Chapter 5 of this thesis to test against a numerical model. Prandle and Rahman [80] modelled tidal oscillations with linear friction in canals of varying depth and cross-section.

Some authors have considered tidal oscillations in estuaries when the water elevation, ζ, is not small compared with the mean water depth. Tidal oscillations in one dimensional estuaries were investigated by Airy in 1845 (discussed in [55] and [83]) using the method of successive approximation. Wind stresses, bottom friction, viscosity and the Coriolis forces were omitted. Flow is in the x direction only. The resulting shallow water equations with advective terms included are nonlinear. The estuary is of constant depth and is of semi-infinite length and the tidal oscillation is specified at the open sea. The solutions for ζ and U have terms involving ω, the angular frequency of the tidal oscillation, plus terms involving twice the frequency, 2ω, the latter terms representing overtides, or tides of the second order. Airy's analysis did not extend to tides of higher order, i.e. those whose frequencies are three, four or more times that of that of the forced frequency. Airy's solution is discussed further in chapter 4. Kreiss [54] included linear friction in analysing nonlinear oscillations in a tidal channel of finite length. He had a tidal oscillation of angular frequency, ω, at the open sea boundary. He obtained a solution for the velocity including a second order overtide using the perturbation method. Proudman [82] included quadratic friction in analysing nonlinear oscillations due to tide and surge in a channel of finite length. He had a prescribed incident wave of first order at the mouth plus a reflected wave of first order plus second order terms which reduce to a reflected wave at the mouth. Knight [51] extended Proudman's work. Gallagher and Munk [27], Kabbaj and Provost [43], and DiLorenzo [17] all found second order solutions for quadratic frictional tidal flow in channels of finite length using perturbation methods. Friedrichs and Madsen [26] produced a table showing the ratio of local acceleration to friction and of advection to friction in twelve estuaries, in ten of which both of these ratios are very small but both are significant in the other two. They found approximate analytical solutions for quadratic frictional tidal flow in estuaries in which the local acceleration and advection were assumed to be insignificant.

2.3.2 Moving boundary analytical solutions

This subsection is concerned with moving boundary analytical solutions of the shallow water wave equations. Because in the vicinity of a moving boundary the oscillations of the water surface, ζ, are comparable to the total depth, $h + \zeta$, the shallow water wave equations need to include nonlinear continuity terms to accurately model moving boundary flow. Nevertheless, there do exist moving boundary solutions of linearised forms of the shallow water wave equations (e.g. [48, 76]).

Carrier and Greenspan [11] obtained moving boundary analytical solutions of the nonlinear shallow water equations for a water wave climbing a linearly sloping beach. Carrier and Greenspan's proof is discussed in detail in [42] and [116]. The procedure is discussed in detail here to show the complications involved in computing the analytical solutions for later comparison with new solutions.

The equations of motion to be solved are based on equations (2.1), (2.2) and (2.3), with flow assumed only in the x-direction and without any Coriolis, viscosity, friction or wind terms. The resultant equations are

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + g\frac{\partial \zeta}{\partial x} = 0, \tag{2.5}$$

and

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (h+\zeta)U}{\partial x} = 0. \tag{2.6}$$

The equations are then rewritten in terms of u', ζ', x' and t', which are successively dimensionless forms of u, ζ, x and t. Then two new variables, σ and λ are introduced. These are defined in terms of the dimensionless variables (with primes dropped)

$$\sigma = 4\sqrt{\zeta - x} \tag{2.7}$$

and

$$\lambda = 2(u+t). \tag{2.8}$$

Then equations (2.5) and (2.6) transform to the equations

$$\frac{\partial \sigma}{\partial t} + u\frac{\partial \sigma}{\partial x} + \left(\frac{\sigma}{4}\right)\frac{\partial \lambda}{\partial x} = 0, \tag{2.9}$$

$$\frac{\partial \lambda}{\partial t} + u\frac{\partial \lambda}{\partial x} + \left(\frac{\sigma}{4}\right)\frac{\partial \sigma}{\partial x} = 0. \tag{2.10}$$

The hodograph transformation, i.e transforming the equations so that the independent and dependent variables are interchanged, [42] is used on equations (2.9) and (2.10), giving

$$\frac{\partial x}{\partial \lambda} - u\frac{\partial t}{\partial \lambda} + \left(\frac{\sigma}{4}\right)\frac{\partial t}{\partial \sigma} = 0, \tag{2.11}$$

$$\frac{\partial x}{\partial \sigma} - u\frac{\partial t}{\partial \sigma} + \left(\frac{\sigma}{4}\right)\frac{\partial t}{\partial \lambda} = 0. \tag{2.12}$$

Substituting $t = \frac{\lambda}{2} - u$ into these equations gives

$$\left(x + \frac{u^2}{2}\right)_\lambda - \frac{\sigma}{4}u_\sigma - \frac{u}{2} = 0, \tag{2.13}$$

$$\left(x + \frac{u^2}{2} + \frac{\sigma^2}{16}\right)_\sigma - \left(\frac{\sigma u}{4}\right)_\lambda = 0, \tag{2.14}$$

where the derivatives are represented by subscripted variables.

It follows from equation (2.14) that there is a variable $\phi(\sigma, \lambda)$, such that

$$x + \frac{u^2}{2} + \frac{\sigma^2}{16} = \frac{\phi_\lambda}{4}, \tag{2.15}$$

and

$$\frac{\sigma u}{4} = \frac{\phi_\sigma}{4}. \tag{2.16}$$

Substituting (2.15) and (2.16) into (2.13) gives a linear equation

$$(\sigma\phi_\sigma)_\sigma - \sigma\phi_{\lambda\lambda} = 0. \tag{2.17}$$

Thus, the original nonlinear equations have been reduced to a linear equation. The instantaneous shoreline is at $\sigma = 0$. Far away from the shoreline nonlinear effects are small and $\sigma = 4$ [100].

Carrier and Greenspan obtained a number of solutions of (2.17). One solution of (2.17) is

$$\phi = AJ_0(\sigma)\cos\lambda. \tag{2.18}$$

Using (2.7), (2.8), (2.15), (2.16) and (2.18), exact solutions for $\zeta(x,t)$ and $u(x,t)$ can be found in terms of σ and λ as

$$\zeta = \frac{\phi_\lambda}{4} - \frac{u^2}{2} = -\frac{A}{4}J_0(\sigma)\sin\lambda - \frac{A^2J_1^2(\sigma)}{2\sigma^2}\cos^2\lambda \tag{2.19}$$

$$u = \frac{\phi_\sigma}{\sigma} = -\frac{AJ_1(\sigma)\cos\lambda}{\sigma} \tag{2.20}$$

$$x = -\frac{\sigma^2}{16} + \frac{\phi_\lambda}{4} - \frac{u^2}{2} = -\frac{\sigma^2}{16} - \frac{A}{4}J_0(\sigma)\sin\lambda - \frac{A^2J_1^2(\sigma)}{2\sigma^2}\cos^2\lambda \tag{2.21}$$

$$t = \frac{\lambda}{2} - u = \frac{\lambda}{2} + \frac{AJ_1(\sigma)\cos\lambda}{\sigma} \tag{2.22}$$

The resulting wave running up the sloping beach is periodic for $0 < A \leq 1$. To find the resultant wave at any time t, a value of σ is substituted in (2.22) to find λ numerically using iteration. Then the values of σ and λ are substituted in equations (2.19), (2.20) and (2.21) to find x, u and ζ. Then using another value of σ another value of λ is found, and so on. Then the process is repeated for other values of t.

The analytical solutions of Carrier and Greenspan (including (2.19) to (2.22)) have been used to test many numerical models [10, 14, 16, 34, 40, 52, 61, 74, 81, 87, 95, 96, 107, 115, 116]. A number of other analytical solutions have been found that are modifications of Carrier and Greenspan's solutions. A prime motivation for these solutions is to provide further tests for numerical techniques and codes.

Ball [5] found exact moving boundary solutions of the nonlinear shallow water equations in Lagrangian form for flow in a parabolic trough and in a paraboloid of revolution. Sielecki and Wurtele [96] tested their numerical model against these solutions. Keller and Keller [48] obtained moving boundary analytical solutions of the linearised shallow water equations for a periodic water wave travelling across an ocean of constant depth then running up a uniform plane beach. Shuto [92] found exact moving boundary solutions of the linearised shallow water equations in Lagrangian form for flow on a sloping beach. His results for runup height were found to be in close agreement to experimental results. Tuck and Hwang [105] investigated the generation of waves on a linear slope to simulate a tsunami due to seismic disturbances, using linearised equations. They also considered runup using nonlinear equations which they transformed into linear equations which they solved.

Thacker [102], using Eulerian equations, obtained exact moving boundary frictionless solutions of the shallow water wave equations similar to those obtained by Ball [5] using Lagrangian equations. Thacker's approach, like Ball's, was to make assumptions about the nature of the motion and then to solve for the basin shape in which that motion should be possible. First Thacker assumed that the velocity was a function of time only. This implied that the water surface was a plane for all time and that such flow could take place in a circular paraboloid, parabolic canal or elliptical paraboloid. For the circular paraboloid the moving shoreline is a circle in the xy plane, with the centre of the circle orbiting the centre of the basin. Also, Thacker considered flow in a circular paraboloid with the velocity assumed to be a function of position and time. This implied that the water surface is an oscillating paraboloid. Finally, Thacker obtained a solution corresponding to a flood wave caused by a parabolic mound spreading over a plane. A number of numerical models have been tested against Thacker's exact solutions [6, 35, 61, 77, 112]. There will be further discussion of Thacker's model in Chapter 3 in which new moving boundary shallow water wave solutions are presented. These are modifications of Thacker's solutions.

Sachdev, Paliannapan and Sarathy [88] built on Thacker's work, producing periodic solutions for frictionless flow involving the Coriolis force in parabolic canals, circular paraboloids and elliptical paraboloids. Johns [40] expressed Carrier and Greenspan's [11] exact solution for moving boundary periodic flow in a simpler form, which involved periodic forcing at the open sea boundary. Li and Raichlen [62] studied the runup of solitary waves on a uniform plane beach connected to an open beach of constant depth. They obtained an analytical nonlinear solution using an approach based on that of

Carrier and Greenspan [11]. The solution was in close agreement to experimental results, giving slightly better results for maximum runup than the approximate linear theory of Synolakis [100]. Kanoglu [44] solved analytically Carrier and Greenspan's equations for the evolution of waves climbing sloping beaches for a number of different initial waveforms.

2.4 Numerical solutions

For domains of flow without exact solutions the equations must be solved numerically (i.e. approximately). One technique is to use the finite difference method in space and in time. Another technique is to use the finite element method in space and the finite difference method in time, often referred to as the finite element method. The finite element numerical method has the advantage over the finite difference numerical method that it can represent the boundaries and topography of domains of flow more accurately. In all numerical models one must specify initial conditions, i.e. state the water heights and velocities in the region at the starting time of the calculation. One might assume a cold start, i.e. initially the water in the region is calm. As this is unrealistic this will cause transient solutions. One must also specify boundary conditions, i.e. state what physical conditions must be satisfied on the boundaries of the region of flow. For example, for a tidal flow one can specify the height of the water on the open sea boundary, τ_S , at different times and that no water can flow across the seashore boundary, τ_L. This is not always a good representation, as water can flow through a pervious sea wall. Often it is sufficiently accurate to assume that the shoreline is fixed. However, in some regions of flow significant tidal flats may be covered and uncovered during a tidal period, meaning that the shoreline changes position substantially during any day. Also, tsunamis flow inland, often a distance of several kilometres. As well, extreme water levels due to storm surges can lead to water flow inland.

At a moving boundary the boundary condition is that the total depth of water, H, is zero. Specifying the height of the water on the sea boundary is an appropriate procedure if only tides are represented. It is also correct to use the velocity rather than the height.

In a finite difference or finite element numerical model the domain of flow is broken up into a grid, with the water depth below the datum specified at the nodes. These nodes are joined in elements or grid blocks or cells, which are usually square in finite difference schemes and usually triangular in finite element schemes. The water height, ζ, and velocity components, U and V, are usually calculated at each node at the end of each time step, δt.

With some moving boundary schemes the elements are fixed; at the end of each time step some elements will be wet and some will be dry. For example, in some finite element schemes an element is classified as wet only if all its nodes are wet, otherwise an element is taken to be dry. With schemes with fixed elements a decision must be made at the end of each time step whether any nodes change status from wet to dry or dry to wet. The usual way of deciding whether a node changes from wet to dry is to see whether its calculated total depth is at or below some minimum value, e.g. zero and then the total depth is set to the minimum value and the velocity to zero. There are various methods used to decide whether a previously dry node becomes wet, with some based on the continuity equation and some based on whether the water at an adjacent wet node becomes higher than the land height of the previously dry node.

With some moving boundary schemes elements with sides on the moving boundary change shape over time with nodes initially on the shoreline moving with the shoreline over time.

Some numerical models have been found to have artificial short wave length waves in their solution. These so called $2\delta x$ (or node to node) oscillations, of wavelength twice δx, the grid spacing [66], have sometimes been found to be substantial in models based on the primitive form of the shallow water equations but much reduced when the continuity equation is replaced by the wave continuity equation [31].

Reid and Bodine [85] numerically modelled a storm surge in Galveston Bay, Texas, using a finite difference scheme. Advection was considered negligible for most of the bay and was not included in the model except at certain regions (submerged barriers and narrow channels) where the effect was included implicitly through the use of nonlinear discharge relations. Grid blocks could flood or dry. Their model included formulas for flooding rates at the shoreline boundary based on empirical expressions for flow over weirs. The model was first tested with a tide and was found to fit the measured tide with a suitable bottom friction factor. The scheme was then used to model water levels and velocities due to Hurricane Carla (September 1961), with the wind stress values used based on meteorological data from the U.S. Weather Bureau. A tidal oscillation was imposed at the open sea boundary. Comparison of the modelled and observed water levels at four places for Hurricane Carla showed a good fit. The scheme was then used to model water levels and velocity due to Hurricane Cindy (1963). Again, comparison of the modelled and observed levels at four sites for Hurricane Carla showed a good fit.

The paper by Leendertse [60] involves the simulation of water quality in well-mixed estuaries and coastal seas. The model, which is finite difference and moving boundary, solves the shallow water equations for the motion of tides including motion on tidal flats and the advective diffusion equation representing the movement of dissolved waste constituents. The numerical solutions are finite difference. The computational procedures were tested in modelling tidal levels and velocities in Jamaica Bay, Long Island, New York, a bay which has many tidal flats. In addition, the dispersion of coliform bacteria from a large number of time-varying sources was computed. Leendertse's land-water boundary moved in discrete steps. Leendertse noted that if discrete changes are made local discontinuities are generated which then radiate from the location of change as small waves through the system. He approached this problem as follows. The search for the new boundary is made at larger intervals than the time step. The computational noise generated by the boundary decays during this interval. If in a particular field a cross-section decreases to less than a preset small value during ebb, then that field is taken out of the computation. As negative cross sections are not allowed to occur, this preset value is close to zero, but positive. During rising water levels, grid points are added to the computation if the average of the adjacent fields which are under water are larger than the preset value. The water level of the newly wetted field is taken to be the average of the adjacent fields. Flooding can occur from one to four sides of a grid field. The tidal flow was forced by a prescribed sinusoidal vertical tide at an inlet. A diagram is shown of tidal velocity vectors at ebb tide.

Sielecki and Wurtele [96] tested three different finite difference schemes against some moving boundary shallow water analytical solutions, including solutions by Ball [5] and Carrier and Greenspan [11]. The three schemes each gave results that were highly accurate. The position of the shoreline at a given time was estimated by an extrapolation of the water level, based on the continuity equation, from the last two underwater grid points to the first underground grid point. The velocity was calculated at underwater points only and at underground nodes was set to zero.

Ramming [84] modelled the tide in the Outer Elbe River, Germany, using a finite difference scheme. No details of the numerical scheme were given. Some points in the river are dry for part of the day and wet for part of the day. Comparison of the computed and observed time variation of water level at three points in the river showed good agreement. Two of the points were dry for part of the day.

Apelt, Gout and Szewczyk [3] presented a finite difference model to simulate the dispersion and transport of pollutants in bays and estuaries. The equations solved are the shallow water wave equations for tidal flow, with allowance made for a mean water level which is not horizontal, and a pollutant equation. The regions modelled include intertidal flats. There is a drying of various regions during tidal motion. Depth values in the vicinity of the moving land-sea boundary must be adjusted so that no instability is introduced by spurious negative depths entering the computation. There was no explanation of the drying and wetting scheme. The model was tested for a number of cases, with good agreement with experimental data. In addition, the tides were simulated for a small bay off the northeastern coast of Australia, which has intertidal flats. Two different approaches were used. The first approach made the water surface over the newly wetted areas horizontal initially, at a level corresponding to the average of the nearby points over deeper water. This approach seemed to work well but generated some small disturbances. The second approach set the water surface over the newly wetted areas to have the same gradient in the direction of the main tidal flow as exists in the adjacent deeper water areas. The second approach gave improved results.

Flather and Heaps [24] presented a finite difference scheme for modelling the M_2 tide in Morecambe Bay, England, which contains large areas of sandbanks exposed at low tide. They used a minimum water depth for the denominator in the quadratic frictional term (2.4), which would otherwise have a singularity at the shoreline, where the depth is zero. Tests are carried out at the end of each time interval to determine whether a grid point is wet or dry; if a grid point is dry its current is set to zero. These tests allow flow at a grid point only when certain conditions on both depth and elevation gradient are satisfied. Three schemes were used; two involving advection, the other without it. Flather and Heaps pointed out that Charnock and Crease [12] presented a dimensional analysis showing that advection is important in the equations of motion when the water elevation is comparable to the mean depth, which implies that advection is important for flow near a moving boundary. Advective terms were removed at points close to the open sea boundary to remove grid scale oscillations. All schemes gave stable solutions. It was found that leaving advection out at all points caused a variation in amplitude of at most 2 per cent. Comparison of the amplitude and phase of water elevation at the one point where data was available showed a good fit. Comparison of computed and modelled velocity ellipses at one point showed a good fit using one scheme and a bad fit using the other two schemes. Diagrams showing velocity vectors and dry areas in Morecambe Bay were presented. Flather and Heaps [25] refined the techniques of their earlier article, allowing for partial wetting or drying of a cell. They used coarse and fine grid models, the latter giving more accurate results.

Herrling [33] presents a finite element model to simulate flow in an estuary with tidal flats. He has a fixed grid with three types of elements: dry, partly flooded or flooded. The first type has all its nodes dry, the second has at least one dry and one wet node and the last has all its nodes wet. The latter two elements are used in the computation. An extrapolation is used to find the actual boundary. The model was first tested for frictional forced flow in a rectangular basin with linearly varying slope and open to the sea. There is no analytical solution against which to test the modelled results. The second test modelled flow in the Jade estuary in Germany. There was no comparison with experimental results.

Runchal [87] developed a finite difference model suitable for storm surges and tidal run-up. His model involves the wetting and drying of cells. There are two ways in which the run-up/draw-down conditions were implemented. The first, implicit, is partly based on Leendertse's scheme [60]. The second is explicit. He tested his models against a number of moving boundary analytical solutions, including solutions by Carrier and Greenspan [11]. The model gave results close to the analytical solutions for most cases. It was found that when the advective term was modelled by the central difference scheme in testing against one of Carrier and Greenspan's solutions that a spurious solution of wave length $2\delta x$ was superimposed on the true solution if δx was above a certain value.

Ages and Woollard [1] modelled the tides on the Fraser River, Canada, using a one dimensional finite difference model. The model was calibrated with fifteen tidal gauges, with the amplitude and phase of water level at all places in good agreement with the modelled values. They had no data for current velocity against which they could test their model.

Gray and Lynch [31] investigated the finite element representations of the one dimensional linearised shallow water equations in conjunction with ten different time stepping schemes using a Fourier analysis. One of the schemes involved replacing the continuity equation with the wave continuity equation (WCE), a second order partial differential equation obtained from both the continuity and momentum equations. They investigated the stability of the time stepping schemes and errors in wave amplitudes and phase. Oscillations of wavelength $2\delta x$ (node to node oscillations) arose in the computed solutions for some schemes. The schemes that suppressed these waves contained second derivatives in space of the water level and velocity. The analysis by Lynch and Gray looked only at linearised shallow water equations. The effects of nonlinear terms (e.g. advection) on stability and node to node oscillations were not examined.

Lynch and Gray [69] extended their WCE (the wave continuity equation) [31] to include nonlinear terms. They developed a finite element model based on this equation and the momentum equations. They tested their numerical model against their analytical solution of the two dimensional linearised shallow water equations for flow in an annular region [67]. Results obtained with linear triangles and quadratic quadrilaterals compared well with analytical solutions. Node to node oscillations in both the circumferential and radial directions were fairly small.

A number of papers have been published on modelling the tides in Port Phillip Bay, Victoria, Australia. These include papers by Easton [19], Black, Hatton and Rosenberg [9], Walker [109], Hubbert and McIntosh [37] and Lawson and Treloar [56, 57]. All of these papers used the shallow water equations except for Walker's, which used a three dimensional model. Modelling of the tides in Port Phillip Bay will be discussed in detail in chapter 7.

Hibberd and Peregrine [34] modelled run-up on a beach using an explicit finite difference scheme. The shoreline was obtained by extrapolation from nearby points. Their scheme was tested against analytical solutions for runup by Carrier and Greenspan [11] and Spielvogel [97]. There was close agreement between the numerical scheme and the analytical solutions except in regions where they believed that discretisation gave insufficient resolution.

Lynch [66] presented a substantial literature review on finite element numerical models of shallow water flow; most of the models were for domains with fixed boundaries.

Lynch and Gray [68, 70] developed a finite element technique for solving moving boundary flow problems using Galerkin's method. The method involved continuous grid deformation during simulation. They first discussed two common alternative numerical approaches to the moving boundary problem and the possible errors in such approaches. The first and simpler approach is to construct an imaginary vertical barrier near the moving shoreline and assume no flux through the vertical barrier. This is a good approximation provided that the real shoreline does not move very far from the assumed shoreline. If the boundary motion is significant compared to the spatial discretisation various errors can arise including the reflection of waves from the fixed boundary. The second and more complicated approach is to consider each element in the domain to be either wet or dry at any computed time. A problem with this approach is that while in actuality it might take something in the order of twenty to two hundred time steps for an element to wet, the approach will wet an element in one time step. Lynch and Gray considered that this instantaneous wetting may generate spurious waves at the boundary.

The application of the Galerkin finite element technique by Lynch and Gray to flow with moving boundaries using a continuously deforming grid, in which each element is wet at any computed time, involves basis functions which are functions of time and space. In contrast the application of the Galerkin finite element technique by some other authors (e.g. Kawahara, Hirano and Tsubota [46]) to flow with moving boundaries using a fixed grid, in which an element is either wet or dry at any computed time, involves basis functions which are functions of space only. Lynch and Gray's scheme involves the changing in the shape of elements which have a side on the moving boundary. For these elements terms extra to those used in the Galerkin scheme for fixed grids are required. Lynch and Gray's scheme was used to model flow in two canals and one harbour. The results were stable over time. The first canal, one dimensional with constant bathymetry, was subject to a surge at the entrance. The surge increased towards an asymptotic value over time. The moving boundary condition was applied at the landward boundary and the solution compared to that obtained by applying a no-flux condition at the point where the beach begins. The surface elevation as a function of time at a point approximately half way along the canal both for the moving boundary case and the no flux case were compared. It was found over time that both solutions agreed quite well. This agreed with the assumption that the authors made - that while the detailed solution at the beach boundary should depend heavily on the choice of boundary condition, the solutions in the interior should be substantially the same. The second example involved a canal with linear bathymetry subject to sinusoidal forcing at the seaward end. This problem was solved using two dimensional linear triangles. The surface elevation as a function of time for the moving boundary and fixed boundary case was compared at a number of nodes. The further away a node was from the moving boundary the closer were the surface elevation graphs for the two cases to each other. The final problem involved a rectangular bay. At the entrance a storm surge, which increased towards an asymptotic value over time, was imposed. One boundary was moving and the rest fixed. At the moving boundary quadratic friction was computed based on a depth of one foot. The positions over time of nodes on the moving boundary were plotted. Boundary points closest to the entrance move the most over time. Velocity vectors in the harbour were plotted over time. The velocities were as expected.

Shuto and Goto [94] solved numerically a linearised set of shallow water equations in Lagrangian form. Numerical finite difference results for wave runup were compared with analytical results for a simple topography and the agreement was good. Numerical results were also calculated for other simple topographies for which there is no analytical solution. The computations were found to be stable. The authors considered that the equations used were suitable for analysing tsunami run-up.

Goto [30] derived a linear set and a nonlinear set of shallow water equations in Lagrangian form. These equations were solved for wave runup using an explicit finite difference scheme. No details of the scheme were provided. In the numerical experiment water overlays a uniform slope connected to a channel of constant depth. The initial condition is that the water particles are not moving and the water surface is horizontal in the channel. The offshore boundary condition is that a sinusoidal progressive wave train enters from the farthest end of the channel of constant depth. Analytical solutions of runup height for the linear case were derived by Shuto [93]. The numerical results, both for the linear and nonlinear case, were compared with the analytical solutions. For the linear case the analytical and numerical results for runup height at different times were in good agreement. The nonlinear results differed by 10 to 20 percent from the linear analytical results. The wave profile for both the linear wave and the nonlinear wave was shown on a diagram as numerically computed at two different times. They were moderately similar, with the shoreline at one time being the same for both waves but quite different at the other time.

Yeh and Chou [111] developed a moving boundary finite difference storm surge model. At water-land boundaries the normal velocity is set equal to zero. The water-land boundaries are not fixed but allowed to move depending on the storm surge elevations. During a rising surge a grid point is added to the computation system if the surge elevation of any of its neighbouring points is above the bottom of the grid point. During the receding surge a grid point is taken out of the computation if its total water depth decreases below a preset value. However, if any of its four neighbouring points has a surge elevation above its bottom value the removal is overruled. A Fourier analysis of a linearised version of the shallow water equations showed that the model is conditionally stable. In four numerical experiments the scheme was tested against a fixed boundary scheme and involved both hypothetical and actual storms. In most cases the moving boundary scheme gave better results than the fixed boundary scheme.

Pearson [74] used a finite difference moving boundary solution of the shallow water equations in Lagrangian form. He compared his numerical solution with an analytical one dimensional solution by Carrier and Greenspan [11]. With 100 mesh points the analytical and numerical results corresponded to better than three significant figures, both for the shoreline position and the wave shape. Pearson also modelled a slowly rising, then periodic wave incident on a coastal depth profile representative of a portion of the West coast of the U.S. The results for this case were checked by repeating the calculation with several different mesh spacings in space and time. A graph showed the wave profile at different times, while another showed beach runup as a function of time.

Holz and Nitsche [36] developed an explicit finite element scheme for modelling tidal flow in intertidal flats. They said that the typical topography of estuaries with intertidal flats is characterised by deep channels with steep banks flowing through nearly horizontal planes. To obtain a high resolution of the topography, small elements have to be used for the steep banks, while on the tidal flats and in the deep water region larger elements can be used. They compared two techniques. The first, with triangular elements, involves the removal of dry elements from the computational scheme and the wetting of dry elements by interpolation. The second, involves modifying the grid for the first technique so that a moving coastline is calculated. This involves generating at the coastline triangular and trapezoidal elements whose shapes vary over time. They found that the schemes could be made stable if second order viscosity or numerical diffusion were used. As the the quadratic friction formula implies infinite friction at the shoreline, they modified the formula near the shoreline. They found that they could smooth out undesirable disturbances by time-averaging every 10 to 30 time steps. They modelled flow in an estuary with tidal flats. They found that the velocities given by the former scheme were too high.

Johns [40] developed a finite difference scheme to solve the one dimensional shallow water equations for flow over a sloping shelf involving a continuously moving shoreline. Johns modified the analytical solutions obtained by Carrier and Greenspan [11] for a non-breaking standing wave with prescribed oscillatory forcing in which the shoreline moves up and down a sloping shelf. In the numerical scheme a variable X is defined so that the moving shoreline always corresponds to $X = 1$. The numerical solutions were compared with the modified analytical solutions for both tsunami and tidal scale oscillations. The forced oscillation for both cases was of amplitude one metre, with the former of period 15 minutes and the latter 12.4 hours. Very close agreement for shoreline displacement as a function of time was found for both oscillations.

Johns et al. [41] used two finite difference models for flow with a continuously moving lateral boundary. They modelled the water levels and velocities in the storm surge caused by the 1977 Andhra cyclone in India. One model uses a fixed coastline. The other is a deforming coastline model with a coordinate transformation used to simplify the numerical treatment of an irregular boundary configuration so that calculations are done using a rectangular grid. The two models give similar results, with the latter slightly better.

Kawahara, Hirano and Tsubota [46] developed a two dimensional shallow water finite element model. The numerical scheme for the time variation is an explicit two step scheme. As this scheme involves a combination of lumped and unlumped coefficients, the scheme is called a selective lumping scheme.

The Selective Lumped Mass matrix model (SLM) is discussed in much more detail in chapter 5. A weighting parameter, s_r, the selective lumping parameter, is used, ranging in value from 0 to less than 1. Kawahara, Hirano and Tsubota developed a CFL (Courant-Friedrichs-Lewy) stability criterion for their scheme. Goraya [29] stated that their stability criterion is incorrect and developed a new stability criterion for the SLM method using Fourier analysis. Kawahara, Hirano and Tsubota said that in practice one should use $0.85 < s_r < 0.95$ because in their numerical experiments there was not erroneous numerical damping with s_r in these limits. They applied their model to four different situations. The first case was the analysis of a solitary wave propagating along a one-dimensional channel with uniform bottom slope. The results were compared with an analytical solution. Results were computed for s_r equalling 0, 0.6 and 0.8 and various discretisations. It was found that the higher the value of s_r and the finer the mesh, the closer that the numerical solution was to the analytical solution. They concluded that to obtain reasonably accurate solutions that at least forty nodal points should be included in one half of a wave length. The time increment, δt, was chosen to be as long as possible but within the limit where stable computations are obtained. The second case was the computation of the propagation of a sinusoidal wave along a rectangular channel, with the water elevation specified at one end, AB, and a progressive wave condition specified at both ends, AB and CD. The ratio of the computed water elevation, ζ, at CD to the specified ζ at AB, was plotted against s_r, with calculations being for s_r equalling 0, 0.2, 0.4, 0.6, 0.8 and 0.9. The ratio increased with increasing s_r, being 0.85 for s_r equalling zero and 0.99 for s_r equalling 0.95. The time increment, δt, reduced with increasing s_r.

Kawahara, Hirano and Tsubota used the SLM scheme to model the tides in Osaka Bay, Japan, comparing computed results with observed results.

They used 609 nodes and 1055 triangular elements. Along the coastline the normal velocity was taken to be zero. At the two open sea boundaries tidal elevations of period 12 hours and with amplitudes and phases varying along the boundaries were specified. The observed velocity and computed velocity at a certain location were found to be in agreement for s_r equalling 0.92. The smaller the value of s_r the smaller the computed velocity, being slightly less than 40 percent of the observed value for when s_r was zero.

Kawahara, Hirano and Tsubota also modelled the tidal current flow in Yatsushiro Bay, Japan, with 759 nodes and 1279 triangular elements. Along the coastline there are several areas at which the sea bed is exposed at low tide. This is dealt with by having the triangles in such areas wet for part of the time and dry for part of the time. In the wetting and drying scheme of Kawahara, Hirano and Tsubota, at each node i of a finite element that is in the domain of computation the water elevation, ζ, and velocity components, U and V, are calculated at the end of each half time step. There are three possible outcomes:

(i) for each node in an element the total depth, $H_i > 0$;
(ii) at least one value of H_i is $H_i > 0$ and the rest of H_i have $H_i \leq 0$; and
(iii) all values of H_i are $H_i \leq 0$.

In case (i), the element is taken to be under water. In case (ii), at any nodal point at which $H_i > 0$, the water elevation, ζ_i , and the current velocity are computed and at nodal points at which $H_i \leq 0$ the water elevation is computed but the current velocity is treated as zero. In case (iii) the element is on the exposed sea bed and is omitted from the computation. Kawahara et al. do not discuss certain aspects of their calculations:

(a) whether all the nodes at $t = 0$ are subject to a cold start (e.g. if a node whose depth is above sea level has an initial zero water level this implies that the water is initially underground);

(b) how they decide when a previously dry element becomes wet plus what height the water becomes at a previously dry node that has become wet;
(c) how they deal with the problem that the quadratic friction (which appears in the momentum equations), which is inversely proportional to H_i , becomes infinite at the shoreline, where H_i equals zero.

Experimental data shows that the semi-diurnal tide in Yatsushiro Bay is predominant. Hence, the tidal elevation specified at the open sea boundaries has period 12 hours with its amplitudes the sum of the M_2 and S_2 amplitudes. The computed velocity vectors were shown at low tide, low tide plus 3 hours, high tide and high tide plus 3 hours. The shoal area is largest at low tide, decreases at low tide plus 3 hours, disappears at high tide and starts to reappear at high tide plus 3 hours; hence the model gives expected results in the tidal flats region.

Kawahara and Umetsu [47] used the SLM method to analyse river flow with a moving boundary. The two unknown variables are discharge per unit width and water elevation. They used the wetting and drying scheme of Kawahara, Hirano and Tsubota, [46]. For their first computation flow in an open channel with a solid wall was calculated, with discharge specified at the start of the channel. Plots of velocity and elevation were shown at various time steps. For the second computation, flow through a channel with three mounds on the channel bottom was computed, with a longitudinal velocity and water elevation specified at the channel entrance. Plots of velocity and elevation were shown at various time steps, showing the mounds being eventually submerged. In the final computation, flow in the Arakawa River, Japan, was computed. At the starting cross-section of the river the velocity normal to the cross-section and the elevation were specified. Plots of velocity at various times showed that some parts of the river were submerged for part of the time and exposed for part of the time.

Lewis and Adams [61] developed an explicit one-dimensional finite difference scheme to model tsunami flooding. The model includes quadratic friction and advection. The stability of a linearised version of the scheme was derived using the von Neumann method. The scheme involves the wetting and drying of cells. The scheme was tested against the analytical solution by Thacker [102] for water sloshing in a parabolic canal. Three different finite difference expressions were used to approximate the advective terms even though Thacker's solution involved no advection terms. With the first advective expression an oscillation of wavelength $2\delta x$ developed rapidly. The first advective formulation was abandoned. With the second advective formulation an initially plane surface in time developed kinks even though the analytical surface is plane for all time. A third advective expression was then used to test against another of Thacker's analytical solutions, with the water surface being parabolic. Smoothing operators were used to dampen all waves components of wavelength $2\delta x$. The surface produced was at times quite different from the analytical parabolic surface. As well, the numerical scheme was tested against a numerical scheme by Carrier and Greenspan [11] for a wave climbing a sloping beach without breaking. The numerical and analytical values for the wave's maximum runup were in agreement.

Pedersen and Gjevik [75] used an implicit finite difference model based on a Lagrangian description to study runup of long water waves. The numerical results were compared with analytical solutions by Spielvogel [97] and experimental data. The analytical and numerical solutions were in close agreement. Simulations of the run-up of solitary waves on relatively steep planes showed surface displacements and runup heights in reasonably good agreement with experiments. The stability of the scheme was investigated using a linear analysis.

Stelling [98] developed a finite difference method to model tidal flow in regions including tidal flats. This model was used to simulate flow in the Eems-Dollard estuary, Netherlands, about half of which is tidal flats. Water elevation as a function of time was set at two open sea boundaries. At other points the water level and velocities were set at zero initially; this caused transients which quickly died out. Velocity vector diagrams at different times showed the wetting and drying of tidal flats.

Zech, Sorel and Vansnick [113] used an explicit finite difference scheme to model the flooding and uncovering of banks and islands in rivers. The model allows for the introduction of new meshes or the taking away of old ones. A new square mesh is introduced when the mean value of the four surrounding water levels exceeds two centimetres. To overcome the problem that the quadratic friction formula implies a very large friction force for very small depths in such cases the water depth is increased by an arbitrary value. The time step was found using the Courant stability condition. Flow was modelled in a rectangular basin with constant bottom slope. On the left boundary the water level varied sinusoidally over time. The motion was given a cold start. A diagram showed the water surface at different times. As well, flow was modelled in a stretch of the Scheldt river, Northern Europe, which includes flood plains. Two diagrams of velocity vectors were shown, one at a time before flooding, another at a time of flooding. Some banks which were uncovered in the first diagram are shown covered in the second diagram.

Peterson et al. [77] developed a numerical scheme for flow involving moving boundaries. It involves a finite element grid with the movement of grid points permitted at the boundary but not in the interior. The scheme is finite difference in time. It was tested against two moving boundary analytical solutions by Thacker [102] for motion in a parabolic canal, one with a surface that remains plane and one with a surface that stays parabolic. Comparison of the modelled water surfaces with the analytical water surfaces at various times showed virtually perfect agreement.

Zelt [114] derived a set of long wave equations in Lagrangian form. A finite element model was used to model the runup of solitary waves on a coastline with variable bottom topography and a curved shoreline.

Falconer and Owens [22] used two different approaches to simulate flooding and drying of tidal reaches in the Humber Estuary, England. The advective terms were expressed at the intermediate time step to ensure stability. If any of the four cross-sections for a wet grid cell were less than some predetermined critical total depth the cell was removed from the computational field and the de Chezy coefficient and velocity components around the cell were set to zero. As a grid cell was allowed to dry, it was assumed that a layer of still water remained over the cell, the depth of this layer corresponding to the value calculated for the cell immediately before it became dry. For the flooding procedure, at multiple time steps, a check was made on the four grid cells surrounding any dry cells to determine which, if any, of the cells were wet. If any of the surrounding cells were found to be wet, the corresponding water elevations were averaged and if the average were found to be larger than the elevation retained on the dry cell when it first became dry, the cell was allowed to flood again. When the flooding and drying technique was applied to the Humber estuary no stable solutions were obtained for any of the parameter variations. The numerical solution was continuously swamped by spurious waves generated at the moving boundaries. The scheme was improved by reducing the discontinuity in the depth mean velocity when a grid cell was removed. Falconer and Owens first tested their original scheme and the modified scheme on an idealised basin with a uniformly sloping bed. At the open boundary a sinusoidal water variation was assumed. While it was found that the first scheme developed substantial spurious waves, in the second scheme the spurious waves were substantially reduced. The modified scheme was then applied to flow in the Humber Estuary. The open sea boundary comprised two adjacent perpendicular straight line segments. On one segment the velocity was assumed to be parallel to the segment which was treated as a free slip wall. On the other segment an elevation was imposed based on tidal data from a nearby point in the estuary. At the landward boundary experimental water elevations and velocities were used. The new technique gave much more realistic predictions for the water elevations and velocity fields than the original technique. Using the new scheme velocity vector diagrams were drawn at low tide and high tide, with the former showing regions of no flow, i.e. dry regions.

Akanbi and Katopedes [2] simulated flood waves propagating on an initially dry bed. They used a finite element scheme that they say is accurate and suppresses the spurious oscillations that are generated in the solution as a result of nonlinear instabilities. Akanbi and Katopedes reformulated the governing equations in a moving coordinate system (ξ, η, t) with respect to the original frame of reference (x, y, t). Then, using the moving coordinate system, they set up finite element equations. Their choice of the weighting function differed from the Galerkin method, where the weighting function is identical to the shape function. A second order difference scheme is employed for the integration in time. The resulting implicit nonlinear equations are solved by the Newton-Raphson method. At the beginning of each time step the boundary node locations are evaluated using the values of nodal velocities from the previous time step. The model was tested against experimental results. The experiments involved a steep increase in discharge in an initially dry permeable rectangular channel. The finite element grid was a rectangular domain with three elements in the y-direction and 10 in the x-direction. Flow was in the x-direction. A plot of the x-coordinate of the front against time showed close agreement between modelled and measured data. The model was also applied to a hypothetical problem involving a flood wave spreading radially on an impervious bed. The flow domain was discretised into quadrilateral finite elements.

Siden and Lynch [95] used the deforming finite element system of Lynch and Gray [70] to solve moving boundary problems involving estuarine length scales. All interior elements deformed. To overcome advective instabilities the advective momentum terms were made explicit. The model was tested against two analytical solutions by Carrier and Greenspan [11], one of which had been expressed in a similar form by Johns [40]. They obtained good agreement between the numerical and analytical solutions.

Gopalakrishnan [28] developed a numerical model for computing two dimensional circulation in regions using the Galerkin finite element method. The model can accommodate a changing domain boundary. This is done by using finite elements that change shape to move with the shoreline. The model was used to represent tidal motion in Kuwait Bay, which contains tidal flats, with widths up to five kilometres. Tidal forcing of period about 24 hours is assumed to take place along an open sea boundary. Given that the position of a node on the shoreline at time t is a certain point, the water surface at time $t + \Delta t$ above that point is projected onto the land to give the node's shoreline position at time $t + \Delta t$. At the shoreline boundary the total depth is taken to be zero and the normal velocity is taken to be zero. To ensure that no element becomes excessively large due to the moving boundary, a maximum elongation of about twice the length perpendicular to the boundary is set as a limit. If further expansion is needed the original discretisation is modified. To ensure that the trajectories of boundary nodes do not cross and hence lead to overlapping elements the lines of movement were predetermined as radially outward normals. At the end of each time step the coefficient matrices of the elements is updated. The mesh of triangular elements was generated using an automatic mesh generation scheme. The generated mesh was was for the water body up to the lower water mark. The time step of 100 seconds was obtained using the Courant condition. Diagrams were shown of velocity vectors in Kuwait Bay. Computations were begun from the low-water mark and continued for 24.75 hours, one cycle of the tide. The model was run for three different values of Manning's constant, n, with a value of 0.05 giving the closest values for amplitude and phase of water level fluctuations to the measured ones at Doha. A comparison of modelled and measured velocity at a station in the middle of the bay showed good agreement.

Roig [86] used a finite element scheme to model flow in tidal flats and tidal marshes adjacent to large estuaries. The scheme used a stationary grid with the fixed boundaries of the fully flooded domain. Roig's scheme allows partially dry elements to remain in the computational grid. The equations describing flow on these elements incorporated a domain coefficient, which was used to ensure that the simulated fluid volume, which is distributed over the whole element, is approximately equal to the actual fluid volume on the partially wet element. Dry elements wetted as soon as at least one node rose above a minimum value. All nodes belonging to an element had to be dry before the element was removed. Roig's scheme, called the marsh element method, was compared against two deforming boundary methods to model flow in a straight river with a shallow sloping rise along one side. A flow of 500 cfs was set at the upstream boundary while a constant water elevation of 2.5 feet was set at the downstream boundary. Although the marsh method with a coarse grid did not track the shoreline as closely as the deforming grid it conserved mass better. Next, the marsh method and a deforming grid method were used to model flow in the Batiquitos Lagoon in southern California. Field data were not available for calibration of either method; therefore only a quantitative comparison was possible. The lagoon has a seaward boundary and a landward boundary, the mouth of a stream, which contributes a discharge of about 500 cfs at the lagoon's entrance. The lagoon consists of broad shallow tidal flats with a central channel. In the model a fluctuating surface elevation of period 12 hours was fixed at the seaward boundary and at the landward boundary a 500 cfs inflow was imposed. Diagrams of volume flow over time across four reference cross-sections indicate that both methods gave similar results.

Leclerc et al. [58] developed an algorithm that takes into account the moving boundary process in free surface flows. The model, which is finite element, has a fixed spatial mesh, with elements becoming dry or wet in a continuous manner. There are three types of elements: dry, partly dry and wet. The algorithm was used to model two different flow situations. They used six node triangular elements with quadratic velocity and linear elevation approximations. If the total water depth at a node is less than some small fixed amount the node is assumed to be dry and the velocity set to zero. On partly wet elements the Reynolds stresses were set to one order of magnitude less than those used on the global flow domain. This range of values is used to ensure is used to ensure a stable behaviour of the model as well as a proper smoothing of the flow field in the vicinity of the moving boundary. Flow in a rectangular basin with a variable slope and with sinusoidal variation of water height over time at one end was modelled. Over time part of the basin was uncovered and later flooded. As well, tidal flow in the Manicougan River estuary, Canada was simulated. At the open sea boundary tidal height as a function of time was imposed. At the upstream boundary an input discharge was imposed. The tide was semi-diurnal. About two thirds of the region undergoes wetting and drying during a tidal cycle. Measured tidal heights and velocities at a number of locations were obtained; the modelled values compared favourably. Diagrams of velocity fields were shown at six different stages of the tidal cycle. For the first three cases there is ebb tide, with gradual uncovering of the tidal flats. In the fourth case the water is almost slack and the tidal flat zones are at their maximum. In the fifth case there is a flood tide and intertidal flats are being recovered. In the sixth case, at lower high tide, flow is slack.

Mader [71] used the SWAN code, which is finite difference, to model two different theoretical tsunamis using the nonlinear shallow water equations. He also used the ZUNI code, which solves the incompressible Navier-Stokes equations, to model one of the tsunamis. The SWAN code was used to model a tsunami wave with 900 second period and a 3 metre half-height travelling 3 kilometres in 12 metres deep water before it interacted with a frictionless one per cent slope. The wave was found to run up the shore over time. Then the periods, slopes and amount of friction were varied. Also studied was a one metre high tsunami, propagating in 4.55 km. deep water. Both the SWAN and ZUNI codes were used. The study showed that the shallow water tsunamis shoal higher, steeper and faster than the Navier-Stokes waves.

Bills [7] modelled the tides in Spencer Gulf, South Australia, using two models. The first model had a fixed shoreline while the second (also reported in Bills and Noye [8]) included the wetting and drying of sandbars and coastal flats. Both models were accurate when compared with experimental data. From now on only the second model is discussed. In the second study tidal flats are defined as regions exposed between ISLW (Indian Spring Low Water) and ISHW (Indian Spring High Water). The wetting and drying scheme is based on that of Flather and Heaps [24]. The tide was modelled as consisting of M_2, S_2, K_1 and O_1 components. In calculating quadratic friction, H, the total depth in the quadratic friction term, is replaced by another term, H_f. If H is less than some minimum value, H_{min} (e.g. 1.0 m.), H_f equals H_{min}; otherwise H_f equals H. Advection and viscosity terms which used values of U and V at the high tide coastal boundary or the open boundary were omitted as their inclusion was found to give unrealistic results for the velocity vector. The ISLW and MSL (mean sea level) values of elements in the flats region were estimated, being a combination of plausible estimates and numerical experiment because no datum level was given on the charts for a lot of the tidal flats. Plots of the predicted elevation signals against time for eight tidal flat elements were shown. Clipping (i.e. horizontal portions of plotted curves) corresponding to drying occurs in all the signals. This occurs when the total depth is zero. There are wiggles in the plots immediately following the wetting. Bills suggested that the wiggles were due to grid-scale oscillations superimposed on the solution because the coastal boundary is caused to move in discrete steps of element size by the wetting/drying scheme. Bills used a grid-scale filter with the advective terms to help suppress these waves. The simulation was run for 32 days. Results for the first three days were discarded because of transients present. A comparison of the predicted and observed tidal constituents at the six available stations showed good agreement except for the O_1 phases, which were about 50 percent above observed values. Comparison of predicted and observed velocity ellipse characteristics at one station showed good agreement.

Mason et al. [72] modelled the bathymetry in the intertidal region of the Wash, England using the waterline method. This method combines a satellite image, which gives the instantaneous location of the shoreline, with numerically modelled sea elevations along the shoreline. The numerical model was a finite difference moving boundary model. A set of the coastlines with their modelled heights was used to produce the bathymetry. Model heights were corrected using local tide gauge information.

The ADCIRC model [16, 63, 64] solves the shallow water wave equations in the form of the GWCE (Generalised Wave Continuity Equation) plus the momentum equations. The GWCE is a second order differential equation obtained from the continuity and momentum equations. The GWCE, introduced by Kinnmark [49], is a modification of the WCE (Wave Continuity equation), introduced by Lynch and Gray [69]. GWCE largely eliminates the large $2\delta x$ modes that often occur when the primitive equations are used. ADCIRC is finite element in space and finite difference in time. ADCIRC can be run using either a Cartesian or a spherical coordinate system. In its original form ADCIRC did not have provision for wetting and drying [63]. In its later versions e.g. Dietrich, Kolar and Luettich [16] and Luettich, Westerink and Scheffner [64], wetting and drying was allowed for. ADCIRC assumes that wetting and drying can be represented by turning areas of the grid on and off element by element. An element is wet if all the nodes are wet and dry if at least one node is dry. The algorithm is located in the middle of the loop, after the solution of the continuity equation, but before the solution of the momentum equation. First, the total water depth at every node is checked against a minimum wetness height, H_0 (e.g. 0.01 m). If the total water depth is larger than or equal to this minimum value, then the node remains active (wet) and is included in the rest of the calculations. However, if the total water depth is below this minimum value then the node is deemed inactive (dry) and removed from the calculations and the velocity set to zero. If the depth falls below $H_0/10$ the depth is reset to $H_0/10$. Secondly, each element that contains only one dry node is tested to determine whether conditions are favourable for wetting that node. The steady state velocity, V_S , that would result from a momentum balance between the water level gradient and the bottom friction between a wet and dry node is checked against a minimum wetting velocity, V_{min} (e.g. 0.05 ms^{-1}). The balance is given by

$$V_S = \frac{g(\zeta_{i-1} - \zeta_i)}{\tau_i \delta s_i}, \tag{2.23}$$

where g is the acceleration due to gravity, ζ_{i-1} is the larger water elevation of the two wet nodes in the element, ζ_i is the water elevation of the previously dry node, τ_i is the bottom friction coefficient at the node and δs_i is the distance between the nodes. The node is wetted if V_S is greater than V_{min}. Dietrich, Kolar and Luettich [16] commented that in this case the V_{min} criterion almost becomes a height restriction, where a node wets if the adjacent node's free surface elevation is sufficiently larger than its own. A landlocked wet node is assumed to be dry. Luettich, Westerink and Scheffner [63] said that the wetting and drying introduce small oscillations into the solution that cause nodes to repeatedly wet and dry in a non-physical manner. To minimise this effect a node must remain wet for a minimum number of time steps before it can dry (e.g. $300/\delta t$). A similar constraint is applied for the wetting of dry nodes. It is not stated in the relevant papers [16, 64] what values of U, V and ζ are given to the newly wetted node. Dietrich, Kolar and Luettich tested their numerical scheme against an analytical solution by Johns [40] (which is a modification of a result by Carrier and Greenspan [11] for frictionless moving boundary flow with periodic forcing over a linear slope). Because the ADCIRC model is unstable without friction the model was run with a small amount of friction. The motion was begun with a cold start. The transients due to the cold start had died out after half a period. After that the numerical results were in good agreement with the analytical results.

Zelt and Raichlen [115] studied inundation by solitary waves over a horizontal bed both experimentally and numerically. They used a Lagrangian finite element model. A variable sized grid was used to increase the resolution of the numerical model near the shoreline to resolve adequately the wave shoaling and inundation flow. Flow both over an initially dry bed and an initially wet bed was examined. For the dry bed the position of the shoreline as a function of time was found to be very similar for both the numerical and experimental results. For the wet bed the numerical solution was unstable.

Westerink et al. [110] used the ADCIRC model [63] to study flow due to tides and hurricane storm surges on the coast between Mississippi Sound and Florida. The open sea boundaries for the tides were forced with K_1, O_1, P_1, M_2 and S_2 components. The Manning coefficient was taken to be 0.030. A minimum bathymetry of 3 m was specified. The grid was generated with the program GREDIT. Initially a fine grid was used. When the standard conservative form of the convective terms in the GWCE was used, convective instabilities occurred in some regions. To ensure stability the GWCE had to be formulated in nonconservative form. Coarser grids did not exhibit any convective instabilities regardless of the way that the convective terms were treated in the GWCE. The simulation was run for 96 days of which the first 6 were discarded. A time step of 90 seconds was used. A time step of 180 seconds led to long-term instabilities. There was good agreement between simulated and field data considering that no tuning was done in the simulation. Simulations with early coarse grids indicate that there needed to be at least about 30 elements per M_2 wavelength. The storm surge due to Hurricane Kate in 1985 was modelled. The shoreline was taken to be a fixed boundary. Tides were forced at the open sea boundary. The wind forcing was computed using a standard hurricane model plus forcing for Hurricane Kate. Predicted and measured water elevations at two stations were in good agreement.

Aramaki et al. [4] modelled tidal flow of period 12 hours in the Ariake Sea, Japan, using the Selective Lumped Mass scheme of Kawahara, Hirano and Tsubota [46], which uses the selective lumping parameter, a weighting parameter. It was found at a given node that, for a given time interval, the lower the selective lumping parameter the lower the tide amplitude. At a given time interval the tide amplitude reaches a certain maximum at a certain selective lumping parameter. If the selective lumping parameter is any higher divergence occurs. It was also found that the smaller the time interval the larger the selective lumping parameter at which divergence occurs. When the modelled tide height as a function of time was plotted against the observed tidal height at a station good agreement was obtained.

Cheng, Casulli and Gartner [13] used a semi-implicit finite difference method to model tides in San Francisco Bay, California. The model involves flooding and drying of computational cells. The model was calibrated and tested against field data from San Francisco Bay, California. The bay includes intertidal flats; the depth data for such areas was sparse and inadequate. If the total depth of a point is computed as negative, it is reset to zero and the velocity to zero. The simulation was begun with a cold start. The open sea boundary conditions for tides were specified using seven tidal constituents, with the values based on data from a nearby shore station. The Manning coefficient was varied, increasing with decreasing depth. Comparison of modelled tides with measured tides at a number of stations showed fairly close agreement. Comparison of speed and direction of tidal current at one station showed moderate agreement. Comparison of speed of tidal current at seven stations showed good agreement.

Kowalik and Murty [52] used a one dimensional moving boundary finite difference scheme to model long wave runup and compared their results with an analytical solution by Carrier and Greenspan [11]. They used an algorithm by Flather and Heaps [24] to identify wet and dry points. Then they used Sielecki and Wurtele's [96] extrapolation of the sea level to the first dry point at the water-land boundary. Spurious short period oscillations in the water level occurred near the boundary. These were removed by a filter. There was good agreement between the numerical and analytical solutions.

Ninomiya and Onishi [73] used the SLM scheme of Kawahara, Hirano and Tsubota [46] to model tidal flow. They wrote a program in BASIC to implement the scheme. First they modelled flow in a rectangular channel of constant depth open to the sea at one end and closed at the other. The water elevation of an incident sine wave function of time of amplitude 0.5 m and period 1 hour was imposed at the open boundary. A plot was shown of the calculated elevation over time at two points. Next the tide in Ariake Bay, Japan, was modelled with an incident wave of amplitude 1.3 m and period 12 hours prescribed along the mouth. The time increment was set to 8 seconds using the stability criterion of Kawahara et al. Manning's coefficient was set to 0.025. With the cold start a periodic tidal current was obtained after 3 periods of the incident wave. Velocity vectors and water level vectors were plotted every 3 hours. The calculated water elevations over time at five locations were shown.

Easton, Singh and Goraya [20] modelled the tidal flow in Lake Wellington, Victoria, Australia, using the SLM method of Kawahara, Hirano and Tsubota [46]. Tides flow into the lake through a narrow entrance at the eastern end. Lake Wellington was discretised into a mesh of nodes and triangles. The positions of the nodes were selected by following the external boundary and internal depth contours on the chart to obtain an appropriate description of the topography of the lake. Once the triangles were drawn depth contours were drawn based on the triangulation. After these depth contours were compared with the depth contours of the nautical chart the positions of the nodes were modified and hence also the triangle boundaries. The final triangulation had 115 nodes and 174 elements. A sinusoidal water level was prescribed at the entrance nodes, with an amplitude of 0.1 m and a period of 12.42 hours. The motion began with a cold start. At the time of maximum velocity at the entrance, the water velocity was found to be significant only close to the entrance. At a node at the western end the amplitude of the water level was found to be 0.07 m and the velocity to be zero.

Titov and Synolakis [103] developed a finite difference model to represent the propagation and runup of one dimensional long waves. The grid spacing changes over time so that the number of gridpoints per wavelength remains constant. The model's results were compared with an analytical solution by Tadepalli and Synolakis [101], other numerical computations and with laboratory data for breaking and nonbreaking solitary waves. The model described the evolution and runup of nonbreaking waves well, but was not so accurate with breaking waves. Titov and Synolakis [104] developed a numerical scheme involving the method of characteristics. The scheme did not involve bottom friction. The model was validated using experimental data for a solitary wave running up a conical island. The scheme was also used to model actual tsunamis in Japan, Russia and Peru. The model was found to reproduce accurately overland flow.

Zhang [116] developed a new finite difference numerical model for computing wave run-up on beaches. The model is based on local Lagrangian coordinates. A transformation is used to transform the computational domain from a moving boundary domain to a fixed boundary domain. The model was tested against the analytical solutions of Carrier and Greenspan [11] for run-up of periodic waves on a beach of constant slope. The numerical solutions were close to the analytical solutions. The model was also tested against an analytical solution by Spielvogel [97] for releasing an initial excess mass of water supported underneath by a sloping sea bed. The time for the waterline to move from its initial position to the origin was 3.94 seconds versus Spielvogel's 4 seconds.

Henry et al. [32] numerically modelled tides and storm surges in eastern Bangladesh waters, with a wave equation replacing the continuity equation [69]. The tidal elevations modelled were close to measured elevations at three coastal sites. However, the model substantially underestimated water level and current speed at an offshore site, south of Sandwip Island, which is close to the coast. It was presumed that the reason that the current was underestimated was due to not including discharge from the Sandwip channel in the model. A storm surge was modelled with the atmospheric pressure being calculated using a model in which the pressure is a function of radial distance from the cyclone centre. The surface wind field was calculated from the pressure distribution using the gradient wind equation. The open sea boundary tidal water levels were estimated from satellite observations. At one location the model overestimated maximum water level substantially. The authors assumed that this was because inundation was not included, although the model can simulate it. At another site the modelled and observed elevations during the surge were quite close.

Balzano [6] reviewed and evaluated seven implicit finite difference shemes for two dimensional flow including wetting and drying. The models were compared for three one dimensional test cases and one two dimensional test case. The first one dimensional case was for a basin 13800 m long and with a uniformly sloping bed. Its depth was 5 m at the open boundary, where a sinusoidal water level variation was imposed, and zero at the other end, where it was closed. Some methods showed smooth water surfaces at various stages while some methods showed wiggles in the free surface profiles at various times. For the next two one dimensional cases some of the methods showed shortcomings. Balzano proposed three new methods to overcome the shortcomings in the three one dimensional test cases. These did not show the unrealistic results of some of the other methods. Balzano then tested the three new methods plus five of the other methods for two dimensional motion with a paraboloidal water surface and a paraboloidal bottom. The results were compared with analytical solutions by Thacker [102]. Balzano's methods gave the best results. One of the methods did not converge. The remaining methods gave moderately accurate results.

Ip, Lynch and Friedrichs [38] developed a finite element model to simulate tidal flooding and dewatering of shallow estuaries. The model involved two dimensional shallow water wave physics, with a porous medium below the open channel to incorporate the realistic drainage of dry elements on a fixed high resolution mesh. Dry areas continued to participate hydraulically in the overall system and the free surface was allowed to fall below the usual bathymetric depth. The Galerkin method was used on linear triangular finite elements and solved implicitly in time. Ip, Lynch and Friedrichs stated that scale analysis and numerical simulation as well as field observation indicate that the primary force balance is between friction and the pressure gradient in shallow tidal embayments; because of this they left out the acceleration terms in the momentum equations. First the model was tested in three hypothetical embayments, each 3 km by 3 km. The first embayment had a uniformly sloping bathymetry, the second a sloping V-shaped bathymetry and the last a sloping W-shaped bathymetry. Each embayment was forced with an M_2 tide at the seaward boundary. For the first embayment the solutions were smooth at all times, with no oscillations. The wet/dry interface migrated over a distance of 1 km during one period. The solutions for the V-shaped embayment also were smooth. The results for the W-shaped embayment were smooth except over a small region, which was said to be due to a resolution issue. Also, the finite element model was applied to model M_2 tides in the Great Bay Estuary system, New Hampshire. The system has tidal flats in the south. The computational grid consisted of 2861 nodes and 5185 elements, with an average resolution of 50 m and a minimum grid spacing of 25 m. Tidal forcing was an 0.9 m amplitude water elevation on the northern inlet boundary. Dynamic equilibrium was established rapidly. While the results seemed realistic no comparison was made between computed and experimental elevations and velocities. This was done in a later study by Erturk [21].

Kashiyama et al. [45] developed an implicit finite element model to simulate frictional flows involving moving boundaries. They used a deforming flood domain. They modelled flow of water in a water tank, the flow being subject to an incident sinusoidal wave. Comparison of measured and modelled runup height at different times showed similar results. There was also good agreement for measured and modelled water elevations.

Vemulakonda et al. [106] used the ADCIRC model [64] to model flooding due to Hurricanes Betsy (1965) and Andrew (1992) in Louisiana. A fixed grid was used with provision for wetting and drying of elements. The grid contained 25,732 nodes and 50,215 elements. There was tidal forcing at the open boundaries, using 5 constituents (M_2, S_2, K_1, O_1 and P_1). Data for the wind forcing was obtained from the National Hurricane Center's hurricane database. Comparison of the observed maximum surges with the modelled maximum surges at selected stations showed good agreement both for Hurricane Betsy and Hurricane Andrew.

Erturk [21] modelled tidal flow in Great Bay, New Hampshire, with inclusion of the effect of the frictional effects of eelgrass on the flow. A finite element model, ADAM, was used. The ADAM model is two dimensional, with a porous medium below the sediment surface to simulate the wetting and drying process in the tidal flats. The ADAM model was developed by Ip, Lynch and Friedrichs [38]. The acceleration terms in the momentum equations were neglected. Tidal flow for the M_2, S_2, and N_2 constituents was modelled. The mesh, which was generated using the TRIANGLE mesh generator, contained 26,455 nodes and 46,740 elements. Generally, the modelled and measured tidal elevations and velocities were found to be close.

Goraya [29] developed a new criterion for the SLM finite element method of Kawahara, Hirano and Tsubota [46], stating that the original citerion, as worked out by Kawahara, Hirano and Tsubota gave stability for a range of values of s_r, the selective lumping parameter, a weighting function, in which the SLM scheme is unstable. Goraya showed that the scheme is stable provided that the time step δt satisfies

$$\delta t \leq \frac{d_m \delta x}{\sqrt{gh}}, \tag{2.24}$$

where δx is the smallest space step, and d_m, which depends on s_r, is obtained from a table by Goraya. The stability condition has been established only for the one dimensional shallow water equations for a constant depth h. Goraya showed, by analysing its complex propagation factor, that the SLM method converges to the correct analytical solution as the number of wavelengths become large and that any $2\delta x$ waves introduced into the solution decrease with decreasing space interval. The complex propagation factor was introduced by Leendertse [59]. Goraya compared analytical solutions for linear flow in a rectangular harbour and a quarter annular harbour [67] with numerical solutions using three finite element models: the SLM model, an harmonic model and the ADCIRC model [63]. There was reasonable agreement of results. Node-to-node oscillations were observed in all models, with the SLM method giving the worst results. Goraya did a convergence study of the SLM method. Linear flow in a rectangular basin of constant depth with forcing at one end and a closed wall at the other end was modelled. Three different meshes were used in the study. The numerical solutions were compared with analytical solutions by Lynch and Gray [67]. It was found that the finer the mesh the closer the results were to analytical solutions. The tides in

Jervis Bay, Australia, were modellled using both the SLM and the ADCIRC models using both a coarse and fine mesh. The ADCIRC model was stable only for the coarse mesh whereas the SLM model was stable for both meshes. Reasonable agreement was obtained with experimental data.

Kowalik et al. [53] developed a new model of global tsunamis. It is finite difference using spherical coordinates. The poles were excluded from the computational domain. A decision is made at each time step whether a wet point will dry and whether a dry point will wet, the latter occurring if the water level at a neighbouring wet point is higher than the land level at the dry point. If the node wets the velocity from the wet point is extrapolated to the previously dry point, while the water level is calculated with the continuity equation. The computational domain involved close to 200 million grid points. The model was applied to the Indian Ocean tsunami in 2004. The tsunami was taken to be generated by an abrupt slip at the India/Burma plate. Computations were made for 50 hours so that the tsunami signal could travel the entire world ocean. Comparison of the observed and modelled travel times of the tsunami showed good agreement at a large proportion of stations.

Luettich [65] used the ADCIRC numerical model [64] to simulate the storm surge due to hurricane Katrina in the USA in 2005. Luettich said that the ADCIRC model has been run for many historical hurricanes and computed surges have compared favorably with measured water levels but that no systematic attempt has been made to verify the surges computed for Hurricane Katrina. Luettich said that in these runs, one of the largest areas of uncertainty in the surge computations is the hurricane wind field.

Chapter 3

Moving boundary analytical solutions of the nonlinear shallow water wave equations

3.1 Introduction

Exact solutions of the nonlinear shallow water wave equations were found by Thacker [102] for frictionless flow involving the Coriolis force in parabolic canals and circular paraboloids and without the Coriolis force for elliptical paraboloids. Thacker's approach was to make assumptions about the nature of the motion and then to solve for the basin in which that motion should be possible. Thacker assumed that the velocity was a function of time only. This implied that the water surface was a plane for all time and that such flow could take place in a circular paraboloid, parabolic canal or elliptical paraboloid. The solutions involve moving shorelines. The motion is oscillatory and continues indefinitely over time. For the circular paraboloid the moving shoreline is a circle in the xy plane, with the centre of the circle orbiting the centre of the basin.

Sachdev, Paliannapan and Sarathy [88] built on Thacker's work, producing periodic solutions for frictionless flow involving the Coriolis force in parabolic canals, circular paraboloids and elliptical paraboloids. Balzano [6], Holdahl, Holden and Lie [35], Lewis and Adams [61], Peterson, Hauser, Thacker and Eppel [77] and Yoon and Cho [112] have compared numerical solutions of the nonlinear shallow water wave equations with some of the analytical solutions in Thacker [102].

The work in this chapter of the thesis builds on the work of Thacker [102]. New exact solutions of the two dimensional nonlinear shallow water wave equations have been found for unforced flow involving linear bottom friction and without the Coriolis force in parabolic canals, circular paraboloids and elliptical paraboloids. The motion decays over time. These solutions involve moving shorelines.

Also exact solutions of the two dimensional nonlinear shallow water wave equations for forced flow and without the Coriolis force in beds with quadratically varying depths have been found for two different types of forcing. For one type of forcing the flow is frictionless while for the other type of forcing the flow is linear frictional. The solutions involve moving shorelines for both types of forcing. For the frictionless flow the motion is oscillatory while for the frictional flow the motion decays over time.

3.2 Thacker's solutions

Thacker [102] considered the case where the motion of water in a basin is governed by two dimensional shallow water equations. The equations that he used are based on equations (2.1), (2.2) and (2.3), with the assumption that wind, bottom friction and horizontal eddy viscosity are negligible. The resulting equations are

$$\frac{\partial U}{\partial t}+U\frac{\partial U}{\partial x}+V\frac{\partial U}{\partial y}-fV+g\frac{\partial \zeta}{\partial x}=0, \tag{3.1}$$

$$\frac{\partial V}{\partial t}+U\frac{\partial V}{\partial x}+V\frac{\partial V}{\partial y}+fU+g\frac{\partial \zeta}{\partial y}=0, \tag{3.2}$$

$$\frac{\partial \zeta}{\partial t}+\frac{\partial (h+\zeta)U}{\partial x}+\frac{\partial (h+\zeta)V}{\partial y}=0. \tag{3.3}$$

Thacker assumed that

$$U=u_0(t), \tag{3.4}$$

$$V=v_0(t). \tag{3.5}$$

It can be shown that equations (3.1) and (3.2) together with equations (3.4) and (3.5) imply that

$$\zeta(x,y,t)=\zeta_0(t)+x\zeta_1(t)+y\zeta_2(t), \tag{3.6}$$

where

$$\zeta_1(t)=-\frac{1}{g}\left(\frac{du_0(t)}{dt}-fv_0(t)\right), \tag{3.7}$$

$$\zeta_2(t)=-\frac{1}{g}\left(\frac{dv_0(t)}{dt}+fu_0(t)\right). \tag{3.8}$$

Thacker assumed that flow takes place in what he termed a parabolic canal, defined by

$$h=h_0\left(1-\frac{x^2}{a^2}\right), \tag{3.9}$$

with h_0 and a constant. Substituting (3.4) (3.5) and (3.9) in (3.3) gives

$$\frac{d\zeta_0\left(t\right)}{dt}+x\frac{d\zeta_1\left(t\right)}{dt}+y\frac{d\zeta_2\left(t\right)}{dt}-\frac{2u_0(t)h_0x}{a^2}+u_0(t)\zeta_1\left(t\right)+v_0(t)\zeta_2\left(t\right)=0. \tag{3.10}$$

Equating the time-varying coefficients of the linearly independent terms 1, x and y

$$\frac{d\zeta_0\left(t\right)}{dt}+u_0(t)\zeta_1\left(t\right)+v_0(t)\zeta_2\left(t\right)=0, \tag{3.11}$$

$$\frac{d\zeta_1\left(t\right)}{dt}-\frac{2u_0(t)h_0}{a^2}=0, \tag{3.12}$$

$$\frac{d\zeta_2\left(t\right)}{dt}=0. \tag{3.13}$$

Substituting (3.7) in (3.12)

$$\frac{d^2u_0\left(t\right)}{dt^2}-f\frac{dv_0}{dt}+\frac{2gh_0u_0\left(t\right)}{a^2}=0. \tag{3.14}$$

Substituting (3.8) in (3.13)

$$\frac{d^2v_0\left(t\right)}{dt^2}+f\frac{du_0}{dt}=0. \tag{3.15}$$

Substituting (3.7) and (3.8) in (3.11)

$$\frac{d\zeta_0\left(t\right)}{dt}-\frac{1}{g}u_0(t)\frac{du_0}{dt}-\frac{1}{g}v_0(t)\frac{dv_0}{dt}=0. \tag{3.16}$$

Thacker gave solutions to equations (3.14), (3.15) and (3.16) without explaining how he obtained his solutions. In the discussion below it is shown how one could obtain Thacker's solutions. If one differentiates (3.14) with respect to t and makes use of (3.15) one obtains a third order differential equation for $u_0(t)$

$$\frac{d^3u_0(t)}{dt^3}+(f^2+\frac{2gh_0}{a^2})\frac{du_0(t)}{dt}=0. \tag{3.17}$$

It can be shown that a solution is

$$u_0(t)=-G\Omega\sin(\Omega t), \tag{3.18}$$

where G is a constant and

$$\Omega = \left(f^2 + \frac{2gh_0}{a^2} \right)^{\frac{1}{2}}. \tag{3.19}$$

Substitution of (3.18) in (3.15) gives a solution

$$v_0(t) = -Gf\cos(\Omega t). \tag{3.20}$$

Substitution of (3.18) and (3.20) in (3.16) and then integration with respect to t gives a solution

$$\zeta_0(t) = -\frac{G^2h_0}{a^2}\cos^2(\Omega t). \tag{3.21}$$

Substitution of (3.18) and (3.20) in (3.7) gives

$$\zeta_1(t) = \frac{2Gh_0}{a^2}\cos(\Omega t). \tag{3.22}$$

Substitution of (3.18) and (3.20) in (3.7) gives

$$\zeta_2 = 0. \tag{3.23}$$

Substitution of (3.21) and (3.22) and (3.23) in (3.6) gives

$$\zeta(x, y, t) = -\frac{G^2h_0}{a^2}\cos^2(\Omega t) + \frac{2Gh_0}{a^2}(\cos(\Omega t))x. \tag{3.24}$$

At the shoreline the total depth is

$$h + \zeta = 0. \tag{3.25}$$

Substitution of (3.9) and (3.24) in (3.25) gives the shorelines

$$x = G\cos(\Omega t) \pm a. \tag{3.26}$$

Thacker derived similar results for flow in circular paraboloidal basins and elliptical paraboloidal basins.

3.3 Model equations

In the following sections of this chapter new analytical moving boundary solutions of the shallow water equations are derived, being modifications of Thacker's solutions. The equations used are based on equations (2.1), (2.2) and (2.3), with the assumption that wind, Coriolis force and horizontal eddy viscosity are negligible. The resulting equations are

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} + \tau U + g\frac{\partial \zeta}{\partial x} = 0, \tag{3.27}$$

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + \tau V + g\frac{\partial \zeta}{\partial y} = 0, \tag{3.28}$$

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (h+\zeta)U}{\partial x} + \frac{\partial (h+\zeta)V}{\partial y} = 0. \tag{3.29}$$

The bottom friction parameter, τ, is considered to be constant.

Equations (3.27), (3.28) and (3.29) differ from Thacker's in that whereas Thacker's equations included Coriolis force terms but did not include friction terms and horizontal eddy viscosity terms, equations (3.27), (3.28) and (3.29) do not include Coriolis force terms or horizontal eddy viscosity, but do include friction terms.

Following Thacker [102] assume solutions for U and V are of the form

$$U = u_0(t), \tag{3.30}$$

$$V = v_0(t). \tag{3.31}$$

Substituting (3.30) in (3.27)

$$\frac{du_0(t)}{dt} + \tau u_0(t) + g\frac{\partial \zeta}{\partial x} = 0, \tag{3.32}$$

which implies that

$$\frac{\partial \zeta}{\partial x} = \zeta_1(t), \tag{3.33}$$

where

$$\zeta_1(t) = -\frac{1}{g}\left(\frac{du_0(t)}{dt} + \tau u_0(t)\right). \tag{3.34}$$

Similarly, substituting (3.31) in (3.28)

$$\frac{\partial \zeta}{\partial y} = \zeta_2(t), \tag{3.35}$$

where

$$\zeta_2(t) = -\frac{1}{g}\left(\frac{dv_0(t)}{dt} + \tau v_0(t)\right). \tag{3.36}$$

Integrating (3.33) with respect to x

$$\zeta(x, y, t) = x\zeta_1(t) + g(y, t). \tag{3.37}$$

Partially differentiating (3.37) with respect to y

$$\frac{\partial \zeta}{\partial y} = \frac{\partial g(y, t)}{\partial y}. \tag{3.38}$$

Hence, using (3.35) and (3.38)

$$\frac{\partial g(y, t)}{\partial y} = \zeta_2(t). \tag{3.39}$$

Integrating (3.39) with respect to y

$$g(y, t) = \zeta_0(t) + y\zeta_2(t). \tag{3.40}$$

Substituting (3.40) in (3.37)

$$\zeta(x, y, t) = \zeta_0(t) + x\zeta_1(t) + y\zeta_2(t). \tag{3.41}$$

It will be shown later how $\zeta_0(t)$ is determined. It can be seen from equation (3.41) that at any time t the water surface is a plane.

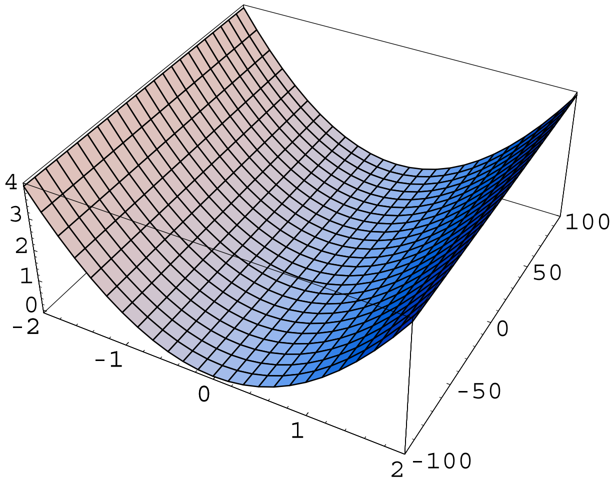

Figure 3.1: Canal of parabolic cross-section

In the discussion that follows flow will be considered in parabolic canals, circular paraboloids, elliptical paraboloids and beds with quadratically varying depths. The discussion is similar to that in Thacker [102], but because the shallow water equations (3.27), (3.28) and (3.29) used in this paper have a slightly different form to Thacker's the discussion leads to different conclusions.

3.4 New solutions for unforced frictional flow in a parabolic canal

3.4.1 Equations

Assume that

$$h = h_0 \left(1 - \frac{x^2}{a^2}\right), \tag{3.42}$$

with h_0 and a constant, so that flow takes place in a canal of parabolic cross section. A diagram of the canal is shown in Figure 3.1.

Substituting (3.30), (3.31), (3.41) and (3.42) in (3.29) gives

$$\frac{d\zeta_0\left(t\right)}{dt}+x\frac{d\zeta_1\left(t\right)}{dt}+y\frac{d\zeta_2\left(t\right)}{dt}-\frac{2u_0(t)h_0x}{a^2}+u_0(t)\zeta_1\left(t\right)+v_0(t)\zeta_2\left(t\right)=0. \quad (3.43)$$

Equating the time-varying coefficients of the linearly independent terms 1, x, and y, respectively

$$\frac{d\zeta_0\left(t\right)}{dt}+u_0(t)\zeta_1\left(t\right)+v_0(t)\zeta_2\left(t\right)=0, \quad (3.44)$$

$$\frac{d\zeta_1\left(t\right)}{dt}-\frac{2u_0(t)h_0}{a^2}=0, \quad (3.45)$$

$$\frac{d\zeta_2\left(t\right)}{dt}=0. \quad (3.46)$$

Substituting (3.34) in (3.45)

$$\frac{d^2u_0\left(t\right)}{dt^2}+\tau\frac{du_0\left(t\right)}{dt}+\frac{2gh_0u_0\left(t\right)}{a^2}=0. \quad (3.47)$$

Substituting (3.36) in (3.46)

$$\frac{d^2v_0\left(t\right)}{dt^2}+\tau\frac{dv_0\left(t\right)}{dt}=0. \quad (3.48)$$

Equations (3.47) and (3.48) have to be solved for $u_0(t)$ and $v_0(t)$.

As equations (3.47) and (3.48) are both second order differential equations, each equation requires two boundary conditions. The solution of (3.47) can be substituted in (3.34) to find $\zeta_1(t)$ and the solution of (3.48) can be substituted in (3.36) to find $\zeta_2(t)$. The solutions of (3.47) and (3.48) plus the solutions for $\zeta_1(t)$ and $\zeta_2(t)$ can be substituted in (3.44), which is first order and hence needs one boundary condition to be solved uniquely for $\zeta_0(t)$.

The auxiliary equation for (3.47) is

$$\lambda^2+\tau\lambda+\frac{2gh_0}{a^2}=0. \quad (3.49)$$

The roots of (3.49) are

$$\lambda = \frac{-\tau \pm \sqrt{\tau^2 - p^2}}{2}, \tag{3.50}$$

where p is defined by

$$p = \sqrt{\frac{8gh_0}{a^2}}. \tag{3.51}$$

Hence, the three possible solutions of (3.49) are for when $\tau < p$, $\tau > p$, and $\tau = p$. Consideration of some typical values of a, h_0 and τ shows that all the possible solutions are realistic solutions of (3.49). Thus there are three realistically possible solutions of (3.47).

The solution of (3.48) is

$$v_0(t) = C + De^{-\tau t}, \tag{3.52}$$

where C and D are constants. If it is assumed that as $t \to \infty$, $v_0(t) \to 0$ then

$$C = 0. \tag{3.53}$$

If it is assumed that $v_0(0) = 0$, then

$$D = 0, \tag{3.54}$$

and hence

$$v_0(t) = 0. \tag{3.55}$$

Substituting (3.55) in (3.36),

$$\zeta_2(t) = 0. \tag{3.56}$$

Hence, making use of the three possible solutions of (3.47) plus the results of (3.55) gives three possible solutions of (3.44). The solutions of (3.41), (3.44) and (3.47) for $\tau < p$, $\tau > p$, and $\tau = p$ are discussed in the subsections below.

3.4.2 Parabolic canal: flow for $\tau < p$

If $\tau < p$, then the solution of (3.47) is

$$u_0(t) = e^{-\frac{\tau t}{2}} \left(A\cos\left(\frac{\sqrt{p^2-\tau^2}}{2}t\right) + B\sin\left(\frac{\sqrt{p^2-\tau^2}}{2}t\right)\right), \qquad (3.57)$$

where A and B are constants, obtained by using given values for $u_0(0)$ and $u_0'(0)$. It can be seen from (3.57) that as $t \to \infty$, $u_0(t) \to 0$.

If A and B are chosen so that $A = 0$ and $B \neq 0$, i.e. $u_0(0) = 0$ and $u_0'(0) = Bs$, then

$$u_0(t) = Be^{-(\tau t)/2}\sin st, \qquad (3.58)$$

where

$$s = \frac{\sqrt{p^2-\tau^2}}{2}. \qquad (3.59)$$

Substituting (3.58) in (3.34)

$$\zeta_1(t) = -\frac{e^{-(\tau t)/2}}{g}\left(Bs\cos st + \frac{\tau B}{2}\sin st\right). \qquad (3.60)$$

Substituting (3.55), (3.56), (3.58) and (3.60) into (3.44) and integrating with respect to t gives

$$\zeta_0(t) = \frac{a^2B^2e^{-\tau t}}{8g^2h_0}\left(-s\tau\sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right) - \frac{B^2e^{-\tau t}}{4g}, \qquad (3.61)$$

with the constant of integration being zero because it is assumed that as $t \to \infty$, $\zeta_0(t) \to 0$.

Substituting (3.56), (3.60) and (3.61) into (3.41)

$$\zeta(x,t) = \frac{a^2B^2e^{-\tau t}}{8g^2h_0}\left(-s\tau\sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right) - \frac{B^2e^{-\tau t}}{4g}$$
$$-\frac{e^{-(\tau t)/2}}{g}\left(Bs\cos st + \frac{\tau B}{2}\sin st\right)x. \quad (3.62)$$

It can be seen that as $t \to \infty$, $\zeta(t) \to 0$, i.e. the displacement of the fluid from equilibrium gradually dies out over time, which is the result that one would expect with a bottom friction force acting on the fluid.

At the shoreline, the total depth

$$h + \zeta = 0. \quad (3.63)$$

Substituting (3.42) and (3.62) in (3.63) gives

$$h_0(1 - \frac{x^2}{a^2}) + \frac{a^2B^2e^{-\tau t}}{8g^2h_0}\left(-s\tau\sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right) - \frac{B^2}{4g}$$
$$-\frac{e^{-(\tau t)/2}}{g}\left(Bs\cos st + \frac{\tau B}{2}\sin st\right)x = 0. \quad (3.64)$$

Multiplying equation (3.64) by $\frac{a^2}{h_0}$ and rearranging gives

$$x^2 + \frac{a^2Be^{-(\tau t)/2}}{gh_0}\left(s\cos st + \frac{\tau}{2}\sin st\right)x = a^2 - \frac{a^4B^2e^{-\tau t}\tau s}{8g^2h_0^2}\sin 2st$$
$$+\frac{a^4B^2\tau^2e^{-\tau t}}{32g^2h_0^2}\cos 2st - \frac{a^4B^2s^2e^{-\tau t}}{8g^2h_0^2}\cos 2st - \frac{a^2B^2e^{-\tau t}}{4gh_0}. \quad (3.65)$$

Completing the square on the left hand side gives

$$\left(x - \frac{a^2Be^{-(\tau t)/2}}{2gh_0}\left(-s\cos st - \frac{\tau}{2}\sin st\right)\right)^2 = a^2 - \frac{a^4B^2e^{-\tau t}\tau s}{8g^2h_0^2}\sin 2st$$
$$+\frac{a^4B^2\tau^2e^{-\tau t}}{32g^2h_0^2}\cos 2st - \frac{a^4B^2s^2e^{-\tau t}}{8g^2h_0^2}\cos 2st - \frac{a^2B^2e^{-\tau t}}{4gh_0}$$
$$+\frac{a^4B^2e^{-\tau t}s^2}{4g^2h_0^2}\cos^2 st + \frac{a^4B^2e^{-\tau t}\tau^2}{16g^2h_0^2}\sin^2 st + \frac{a^4B^2e^{-\tau t}\tau s}{4g^2h_0^2}\sin st\cos st. (3.66)$$

Using (3.59) and trigonometric identities equation (3.66) simplifies to

$$\left(x - \frac{a^2 e^{-(\tau t)/2}}{2gh_0}\left(-Bs\cos st - \frac{\tau B}{2}\sin st\right)\right)^2 = a^2. \tag{3.67}$$

Hence, the projection of the moving shorelines on the xy plane is two parallel straight lines

$$x = \frac{a^2 e^{-(\tau t)/2}}{2gh_0}\left(-Bs\cos st - \frac{\tau B}{2}\sin st\right) \pm a. \tag{3.68}$$

The water moves backwards and forwards across the canal with motion dying out as $t \to \infty$. As $t \to \infty$ the shorelines approach

$$x = \pm a, \tag{3.69}$$

the shorelines for an undisturbed surface, and $\zeta \to 0$, so that friction will cause the initial disturbance to eventually die out. The motion is like that of a damped pendulum.

Consider a parabolic canal for which $a = 3$ km, $h_0 = 10$ m, and $\tau = 0.001$ s $^{-1}$, for motion in which $B = 5$ ms^{-1}. The development of the motion from $t = 0$ s to $t = 3400$ s, in increments of 200 s, is shown in Figure 3.2. The vertical displacement of the water surface from equilibrium as a function of time at $x = 0$ is shown in Figure 3.3. The velocity of the fluid as a function of time is shown in Figure 3.4. The x-coordinate of the left hand shoreline as a function of time is shown in Figure 3.5.

3.4.3 Parabolic canal: flow for $\tau > p$

When $\tau > p$, then the solution of (3.47) is

$$u_0(t) = Ae^{qt} + Be^{rt}, \tag{3.70}$$

where A and B are constants, obtained by using given values for $u_0(0)$ and $u_0'(0)$, and where q and r are defined by

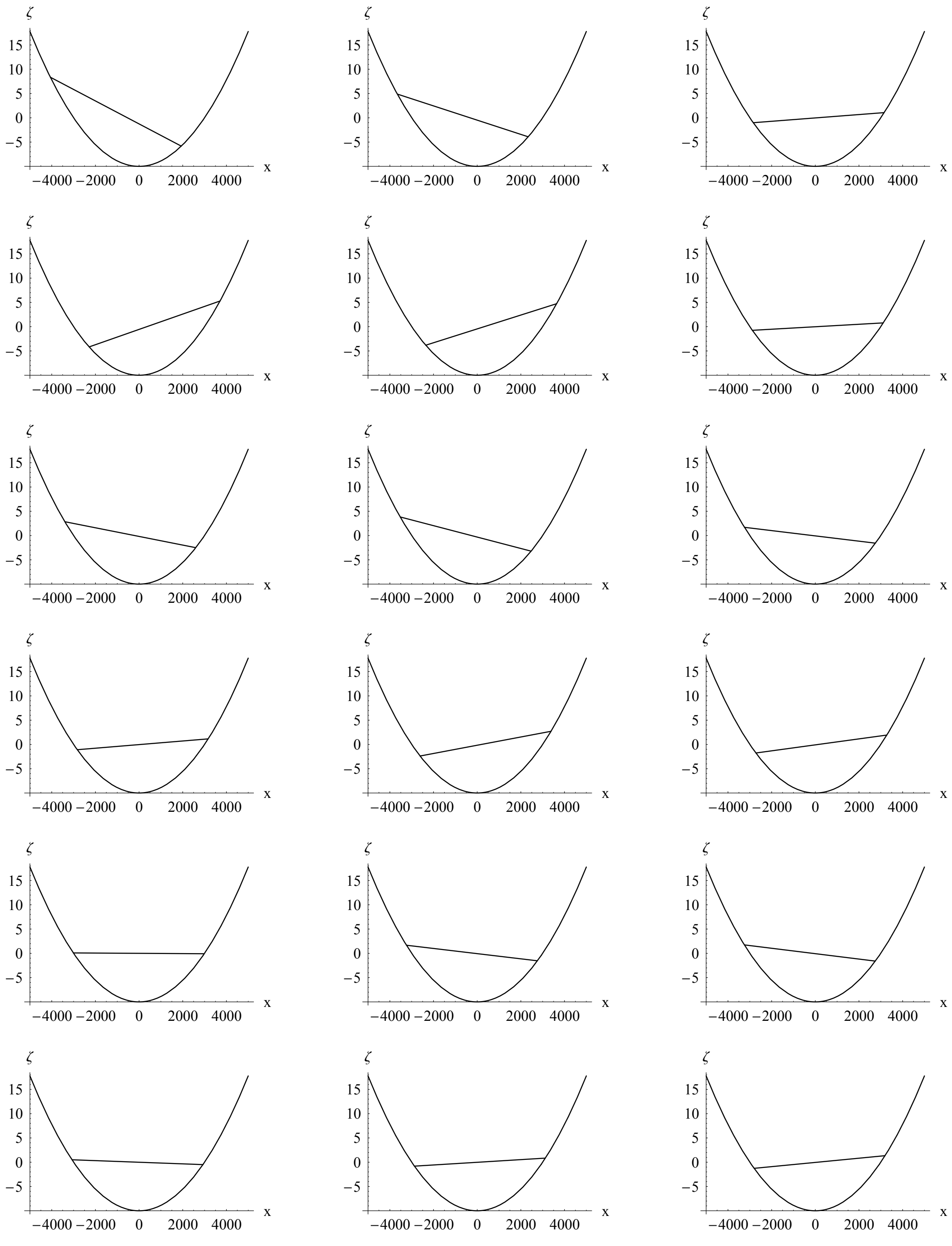

Figure 3.2: The development of the motion of fluid in a parabolic canal with $\tau < p$, $a = 3$ km, $h_0 = 10$ m, and $\tau = 0.001\mathrm{s}^{-1}$, $A = 0$ and $B = 5$ ms^{-1}, from $t = 0$ s to $t = 3400$ s, in increments of 200 s. Dimensions are in metres.

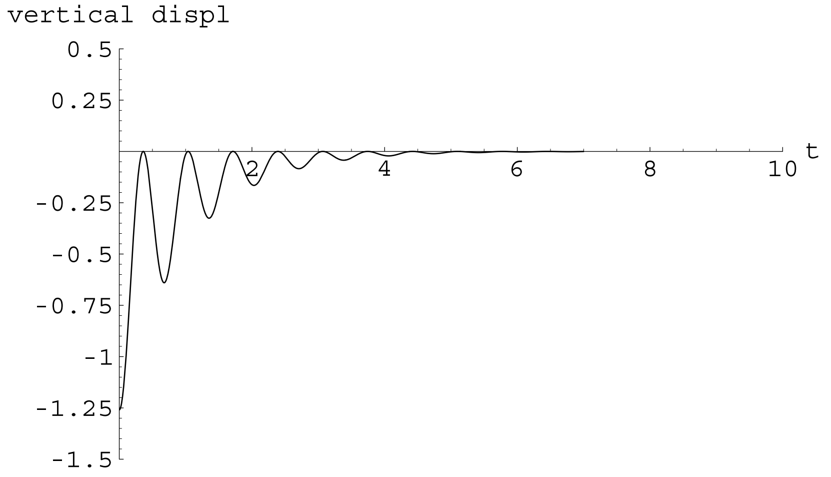

Figure 3.3: The vertical displacement of the water surface from equilibrium in a parabolic canal with $\tau < p$, $a = 3$ km, $h_0 = 10$ m, $\tau = 0.001\text{s}^{-1}$, $A = 0$ and $B = 5\text{ms}^{-1}$, at $x = 0$ km. Dimensions are in metres on the vertical axis and thousands of seconds on the horizontal axis.

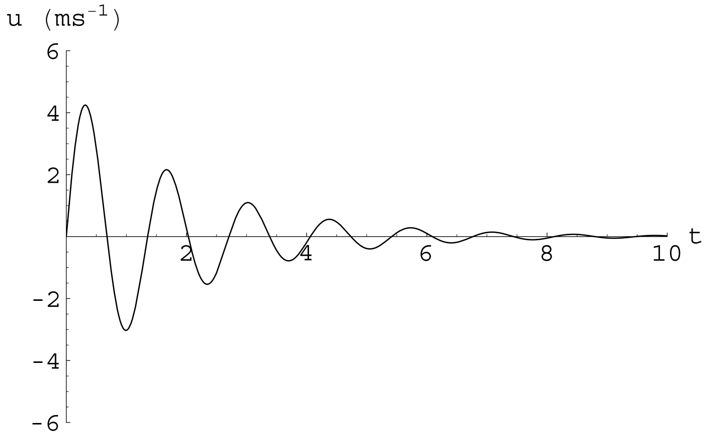

Figure 3.4: The velocity of the fluid in ms $^{-1}$ in a parabolic canal with $\tau < p$, $a = 3$ km, $h_0 = 10$ m, and $\tau = 0.001$ s^{-1}, $A = 0$ and $B = 5$ ms^{-1}. Dimensions are in metres per second on the vertical axis and thousands of seconds on the horizontal axis.

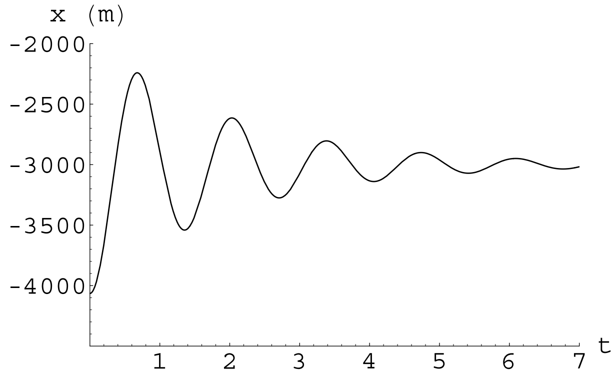

Figure 3.5: The location of the left hand shoreline in a parabolic canal with $\tau < p$, $a = 3$ km, $h_0 = 10$ m, and $\tau = 0.001$ s^{-1} $A = 0$ and $B = 5$ ms^{-1}. Dimensions are in metres per second on the vertical axis and thousands of seconds on the horizontal axis.

$$q = \frac{-\tau + \sqrt{\tau^2 - p^2}}{2} \tag{3.71}$$

and

$$r = \frac{-\tau - \sqrt{\tau^2 - p^2}}{2}. \tag{3.72}$$

If A is chosen to be zero and hence $u_0(0) = B$, $u_0'(0) = Br$, then

$$u_0(t) = Be^{rt}. \tag{3.73}$$

It can be shown that

$$\zeta(x,t) = \frac{1}{g}\left(\left(\frac{r+\tau}{2r}\right)\left(B^2 e^{2rt}\right)\right) - \frac{1}{g}\left(r+\tau\right)Be^{rt}x. \tag{3.74}$$

At the shoreline, the total depth

$$h + \zeta = 0. \tag{3.75}$$

Substituting (3.42) and (3.74) in (3.75)

$$\left(x + \frac{a^2}{2gh_0}\left(B(r+\tau)e^{rt}\right)\right)^2 = a^2. \tag{3.76}$$

Hence, the projection of the moving shoreline on the xy plane is two parallel straight lines

$$x = -\frac{a^2}{2gh_0}\left(B(r+\tau)e^{rt}\right) \pm a. \tag{3.77}$$

As $t \rightarrow \infty$, the shorelines approach

$$x = \pm a, \tag{3.78}$$

the shorelines for an undisturbed surface, and $\zeta \rightarrow 0$ so that friction will cause the initial disturbance to eventually die out.

Consider a parabolic canal for which a = 30 km, h_0 = 10 m, and τ = 0.001s^{-1}, for motion in which B = 0.1 ms^{-1} and hence $u_0(0)$ = 0.1 ms^{-1} and $u_0'(0)$ = -0.000068 ms^{-2} from t = 0 s to t = 7200 s. Figure 3.6 shows the horizontal displacement of the right hand shoreline of this canal from its equilibrium position, while Figure 3.7 shows the vertical displacement of the canal's water surface from its equilibrium position at x = 27 km.

3.4.4 Parabolic canal: flow for $\tau = p$

When $\tau = p$, the solution of (3.47) is

$$u_0(t) = e^{-\frac{\tau t}{2}}(A + Bt), \tag{3.79}$$

where A and B are constants, obtained by using given values for $u_0(0)$ and $u_0'(0)$. It can be seen from equation (3.79) that as $t \rightarrow \infty$, $u_0(t) \rightarrow 0$.

It can be shown that

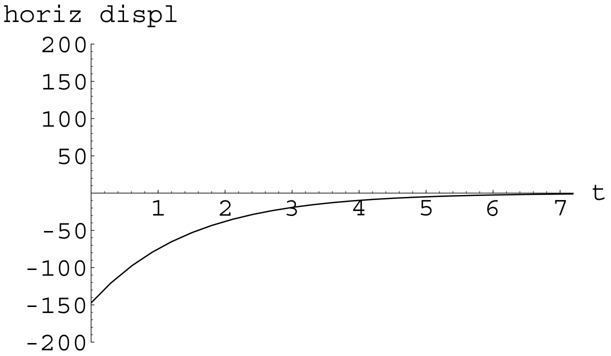

Figure 3.6: The horizontal displacement of the right hand shoreline from equilibrium for a parabolic canal for which $a = 30$ km, $h_0 = 10$ m, and $\tau = 0.001$ s^{-1}, for motion in which $A = 0$, $B = 0.1$ ms^{-1} and $\tau > p$. Dimensions are in metres on the vertical axis and thousands of seconds on the horizontal axis.

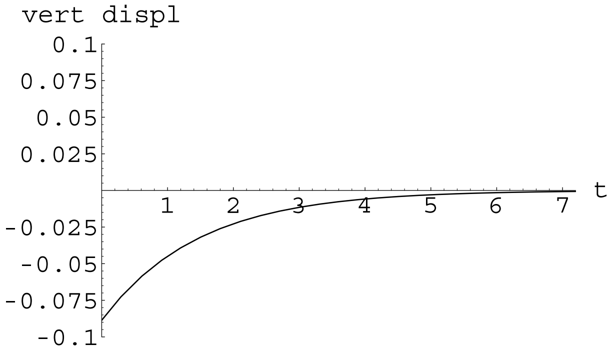

Figure 3.7: The vertical displacement of the water surface from equilibrium at $x = 27$ km for a parabolic canal for which $a = 30$ km, $h_0 = 10$ m, and $\tau = 0.001$ s^{-1}, for motion in which $A = 0$, $B = 0.1$ ms^{-1} and $\tau > p$. Dimensions are in metres on the vertical axis and thousands of seconds on the horizontal axis.

$$\zeta(x,t) = -\frac{e^{-\tau t}}{\tau}\left(K + L\left(t + \frac{1}{\tau}\right)\right) - \frac{e^{-\tau t}}{\tau}\left(M\left(t^2 + \frac{2t}{\tau} + \frac{2}{\tau^2}\right)\right)$$
$$-\frac{x}{g}\left(B + \frac{\tau}{2}(A + Bt)\right)e^{-\tau t/2}, \quad (3.80)$$

where

$$K = \frac{1}{g}\left(AB + \frac{\tau A^2}{2}\right), \quad (3.81)$$

$$L = \frac{1}{g}\left(B^2 + \tau AB\right), \quad (3.82)$$

and

$$M = \frac{\tau B^2}{2g}. \quad (3.83)$$

It can be seen that as $t \to \infty$, $\zeta \to 0$. At the shoreline, the total depth

$$h + \zeta = 0. \quad (3.84)$$

Substituting (3.42) and (3.80) in (3.84)

$$\left(x + \frac{a^2}{2gh_0}e^{-\frac{\tau t}{2}}\left(B + \frac{\tau}{2}(A + Bt)\right)\right)^2 = a^2. \quad (3.85)$$

Hence the projection of the moving shoreline on the xy plane is two parallel straight lines

$$x = -\frac{a^2}{2gh_0}e^{-\frac{\tau t}{2}}\left(B + \frac{\tau}{2}(A + Bt)\right) \pm a. \quad (3.86)$$

As $t \to \infty$ the shorelines approach

$$x = \pm a, \quad (3.87)$$

the shorelines for an undisturbed surface, and $\zeta \to 0$, so that friction will cause the initial disturbance to eventually die out.

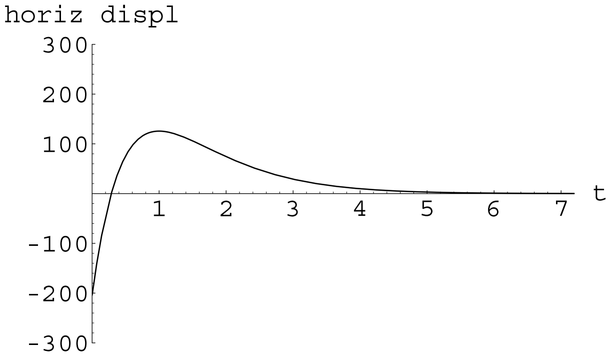

Figure 3.8: The horizontal displacement of the right hand shoreline from equilibrium for a parabolic canal for which $a = 10$ km and $h_0 = 10$m and $\tau = 0.00280143$ s^{-1}, for motion in which $A = 1$ and $B = -0.001$ ms^{-1} and $\tau = p$. Dimensions are in metres on the vertical axis and thousands of seconds on the horizontal axis.

Consider a parabolic canal for which $a = 10$ km and $h_0 = 10$m and $\tau = 0.00280143$ s^{-1}, for motion in which $A = 1$ms $^{-1}$ and $B = -0.001$ ms $^{-1}$ and hence $u_0(0) = 1$ ms $^{-1}$ and $u_0'(0) = -0.00240071$ ms $^{-2}$. Figure 3.8 shows the horizontal displacement of the right hand shoreline of this canal from its equilibrium position from $t = 0$ s to $t = 7200$ s, while Figure 3.9 shows the vertical displacement of the canal's water surface from its equilibrium position at $x = 9$ km from $t = 0$ s to $t = 7200$ s.

3.5 New solutions for unforced frictional flow in a circular paraboloid

3.5.1 Equations

For flow in a circular paraboloid, assume that

$$h = h_0 \left(1 - \frac{x^2}{a^2} - \frac{y^2}{a^2} \right), \tag{3.88}$$

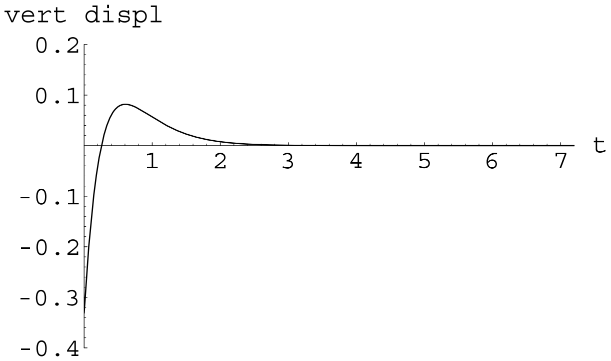

Figure 3.9: The vertical displacement of the right hand shoreline from equilibrium at $x = 9$ km for a parabolic canal for which $a = 10$ km and $h_0 = 10$m and $\tau = 0.00280143$ s^{-1}, for motion in which $A = 1$ and $B = -0.001$ ms^{-1} and $\tau = p$. Dimensions are in kilometres on the vertical axis and thousands of seconds on the horizontal axis.

with h_0 and a constant. The flow represented could be flow in a lake. A diagram of the basin is shown in Figure 3.10.

Substituting (3.30), (3.31), (3.41), and (3.88) in (3.29) gives

$$\frac{d\zeta_0\,(t)}{dt}+x\frac{d\zeta_1\,(t)}{dt}+y\frac{d\zeta_2\,(t)}{dt}-\frac{2u_0(t)h_0x}{a^2}+u_0(t)\zeta_1\,(t)-\frac{2v_0(t)h_0y}{a^2}+v_0(t)\zeta_2\,(t)=0. \tag{3.89}$$

Equating the time-varying coefficients of the linearly independent terms 1, x, and y, respectively

$$\frac{d\zeta_0\,(t)}{dt}+u_0(t)\zeta_1\,(t)+v_0(t)\zeta_2\,(t)=0, \tag{3.90}$$

$$\frac{d\zeta_1\,(t)}{dt}-\frac{2u_0(t)h_0}{a^2}=0, \tag{3.91}$$

$$\frac{d\zeta_2\,(t)}{dt}-\frac{2v_0(t)h_0}{a^2}=0. \tag{3.92}$$

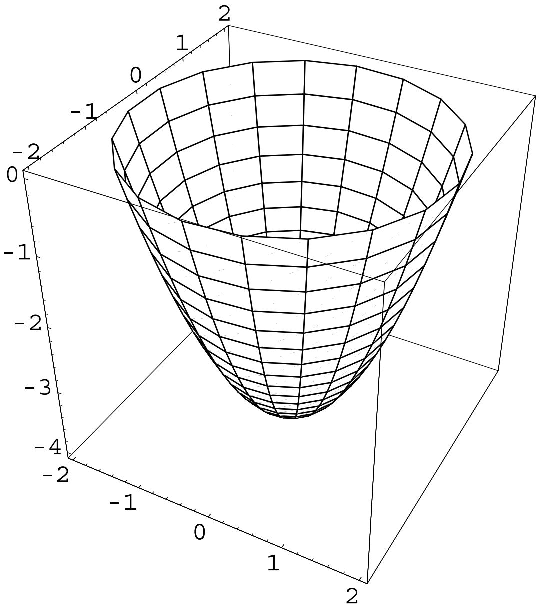

Figure 3.10: Circular paraboloidal basin

Substituting (3.34) in (3.91)

$$\frac{d^2u_0\,(t)}{dt^2}+\tau\frac{du_0\,(t)}{dt}+\frac{2gh_0u_0\,(t)}{a^2}=0. \tag{3.93}$$

Substituting (3.36) in (3.92)

$$\frac{d^2v_0\,(t)}{dt^2}+\tau\frac{dv_0\,(t)}{dt}+\frac{2gh_0v_0\,(t)}{a^2}=0. \tag{3.94}$$

Equations (3.93) and (3.94) have to be solved for $u_0(t)$ and $v_0(t)$.

As equations (3.93) and (3.94) are both second order differential equations, each equation requires two boundary conditions. The solution of (3.93) can be substituted in (3.34) to find $\zeta_1(t)$ and the solution of (3.94) can be substituted in (3.36) to find $\zeta_2(t)$. The solutions of (3.93) and (3.94) plus the solutions for $\zeta_1(t)$ and $\zeta_2(t)$ can be substituted in (3.90), which is first order and hence needs one boundary condition to be solved uniquely for $\zeta_0(t)$.

The auxiliary equation for (3.93) and (3.94) is

$$\lambda^2 + \tau\lambda + \frac{2gh_0}{a^2} = 0. \tag{3.95}$$

The roots of (3.95) are

$$\lambda = \frac{-\tau \pm \sqrt{\tau^2 - p^2}}{2}, \tag{3.96}$$

where p is defined by

$$p = \sqrt{\frac{8gh_0}{a^2}}. \tag{3.97}$$

Hence, the three possible solutions of (3.49) are for when $\tau < p$, $\tau > p$, and $\tau = p$. The solutions of (3.90), (3.93) and (3.94) for $\tau < p$, $\tau > p$, and $\tau = p$, are discussed in the subsections below.

3.5.2 Circular paraboloid: flow for $\tau < p$

If $\tau < p$, then the solution of (3.93) is

$$u_0(t) = e^{-\frac{\tau t}{2}} \left(A \cos\left(\frac{\sqrt{p^2 - \tau^2}}{2} t \right) + B \sin\left(\frac{\sqrt{p^2 - \tau^2}}{2} t \right) \right), \tag{3.98}$$

where A and B are constants, obtained by using given values for $u_0(0)$ and $u_0'(0)$. It can be seen from (3.98) that as $t \to \infty$, $u_0(t) \to 0$.

Similarly, the solution of (3.94) is

$$v_0(t) = e^{-\frac{\tau t}{2}} \left(C \cos\left(\frac{\sqrt{p^2 - \tau^2}}{2} t \right) + D \sin\left(\frac{\sqrt{p^2 - \tau^2}}{2} t \right) \right), \tag{3.99}$$

where C and D are constants, obtained by using given values $v_0(0)$ and $v_0'(0)$. It can be seen from (3.99) that as $t \to \infty$, $v_0(t) \to 0$.

If A, B, C and D are chosen so that $A = D = 0$ and $B = C \neq 0$, i.e. $u_0(0) = 0 = v_0'(0)$ and $u_0'(0) = Bs$ and $v_0(0) = B$, then

$$u_0(t) = Be^{-\frac{\tau t}{2}} \sin st \tag{3.100}$$

and

$$v_0(t) = Be^{-\frac{\tau t}{2}} \sin st, \tag{3.101}$$

where

$$s = \frac{\sqrt{p^2 - \tau^2}}{2}. \tag{3.102}$$

Substituting (3.100) in (3.34)

$$\zeta_1(t) = -\frac{e^{-\frac{\tau t}{2}}}{g} \left(Bs \cos st + \frac{\tau B}{2} \sin st \right). \tag{3.103}$$

Substituting (3.101) in (3.36)

$$\zeta_2(t) = \frac{e^{-\frac{\tau t}{2}}}{g} \left(Bs \sin st - \frac{\tau B}{2} \cos st \right). \tag{3.104}$$

Substituting (3.100), (3.101), (3.103) and (3.104) into (3.90) and integrating with respect to t gives

$$\zeta_0(t) = -\frac{B^2 e^{-\tau t}}{2g}, \tag{3.105}$$

with the constant of integration being zero because it is assumed that as $t \to \infty$, $\zeta_0(t) \to 0$. Substituting (3.103), (3.104) and (3.105) into (3.41) gives

$$\begin{aligned} \zeta(x, y, t) = -\frac{B^2 e^{-\tau t}}{2g} - \frac{e^{-\frac{\tau t}{2}}}{g} \left(Bs \cos st + \frac{\tau B}{2} \sin st \right) x \\ + \frac{e^{-\frac{\tau t}{2}}}{g} \left(Bs \sin st - \frac{\tau B}{2} \cos st \right) y. \end{aligned} \tag{3.106}$$

It can be seen that as $t \to \infty$, $\zeta(t) \to 0$, i.e. the displacement of the fluid from equilibrium gradually dies out over time, which is the result that one would expect with a bottom friction force acting on the fluid.

At the shoreline, the total depth

$$h + \zeta = 0. \tag{3.107}$$

Substituting (3.88) and (3.106) in (3.107) gives

$$\begin{aligned} h_0\left(1 - \frac{x^2}{a^2} - \frac{y^2}{a^2}\right) - \frac{B^2 e^{-\tau t}}{2g} - \frac{e^{-\frac{\tau t}{2}}}{g}\left(Bs\cos st + \frac{\tau B}{2}\sin st\right)x \\ + \frac{e^{-\frac{\tau t}{2}}}{g}\left(Bs\sin st - \frac{\tau B}{2}\cos st\right)y = 0. \end{aligned} \tag{3.108}$$

Equation (3.108) can be shown to simplify to

$$\begin{aligned} \left(x - \frac{a^2 e^{-\frac{\tau t}{2}}}{2h_0 g}\left(-Bs\cos st - \frac{\tau B}{2}\sin st\right)\right)^2 \\ + \left(y - \frac{a^2 e^{-\frac{\tau t}{2}}}{2h_0 g}\left(Bs\sin st - \frac{\tau B}{2}\cos st\right)\right)^2 = a^2. \end{aligned} \tag{3.109}$$

Hence, the projection of the moving shoreline on the xy plane is a circle, which has a moving centre, with its x-coordinate equal to

$$\frac{a^2 e^{-\frac{\tau t}{2}}}{2h_0 g}\left(-Bs\cos st - \frac{\tau B}{2}\sin st\right)$$

and with its y-coordinate equal to

$$\frac{a^2 e^{-\frac{\tau t}{2}}}{2h_0 g}\left(Bs\sin st - \frac{\tau B}{2}\cos st\right),$$

and with constant radius a . A plan view of the moving shoreline at time t is shown in Figure 3.11. As $t \to \infty$ the centre of the circle spirals in towards the origin and the shoreline approaches

$$x^2 + y^2 = a^2, \tag{3.110}$$

the shoreline for an undisturbed surface, and $\zeta \to 0$, so that friction will cause the initial disturbance to eventually die out.

The diagram in Figure 3.12 shows the path of the centre of the projection on the plane of the moving shoreline from $t = 0$ seconds to $t = 7200$ s for

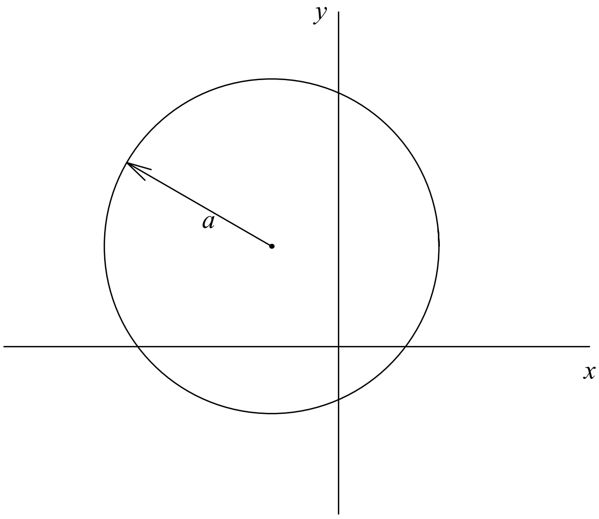

Figure 3.11: A plan view of the moving shoreline at time t for flow in a circular paraboloid. The shoreline is a circle with a moving centre.

motion for which a = 5 km, h_0 = 50 m, τ = 0.001 s^{-1} , $u_0(0)$ = 0 m s^{-1}, u_0' = 0.0062 m s^{-2} , $v_0(0)$ = 0 m s^{-1}, and $v_0'(0)$ = 4 m s^{-1} . The centre gradually spirals clockwise into the origin. The coordinates of the circle's centre in metres are (-159.13,-12.74) at $t = 0$ s, (-10.85,140.32) at $t = 252$ s and (-21.27,-1.70) at $t = 4025$ s.

For the motion in the basin described in the previous paragraph, the vertical displacement of the water surface at the point on the basin, at which $x = 1$ km and $y = 0$ km, is shown in Figure 3.13.

3.5.3 Circular paraboloid: flow for $\tau > p$

When $\tau > p$ then the solution of (3.93) is

$$u_0(t) = Ae^{qt} + Be^{rt}, \tag{3.111}$$

where A and B are constants, obtained by using given values for $u_0(0)$ and $u_0'(0)$, and the solution of (3.94)

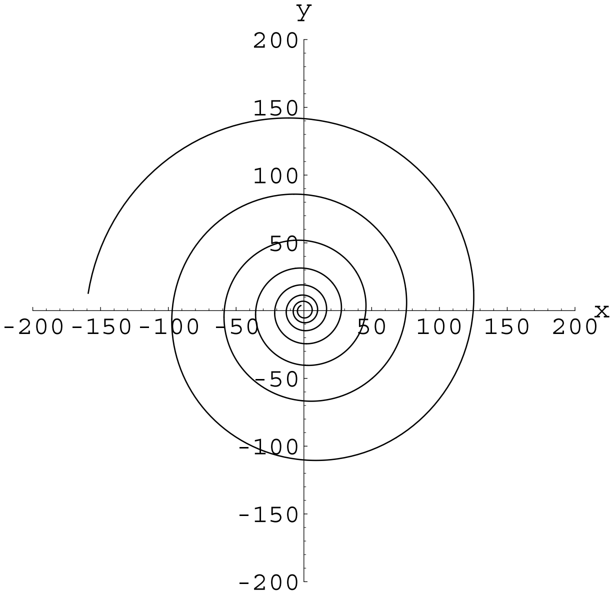

Figure 3.12: The path of the centre of the projection on the xy plane of the moving shoreline. Dimensions are in metres.

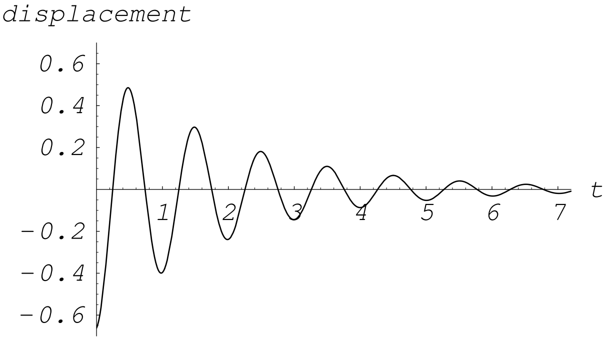

Figure 3.13: The vertical displacement of the water surface at the point on the basin at which $x = 1$ km and $y = 0$ km. Dimensions are in metres on the vertical axis and thousands of seconds on the horizontal axis.

$$v_0(t) = Ce^{qt} + De^{rt}, \tag{3.112}$$

where C and D are constants, obtained by using given values for $v_0(0)$ and $v_0'(0)$, and where q and r are defined by

$$q = \frac{-\tau + \sqrt{\tau^2 - p^2}}{2} \tag{3.113}$$

and

$$r = \frac{-\tau - \sqrt{\tau^2 - p^2}}{2}. \tag{3.114}$$

If A, B, C and D are chosen so that $A = D = 0$ and $B = C$ and hence $u_0(0) = B$, $u_0'(0) = Br$, $v_0(0) = B$, $v_0'(0) = Bq$, then

$$u_0(t) = Be^{rt} \tag{3.115}$$

and

$$v_0(t) = Be^{qt}. \tag{3.116}$$

It can be shown that

$$\begin{aligned}\zeta(x,t) = \frac{1}{g}\left(\left(\frac{r+\tau}{2r}\right)\left(B^2e^{2rt}\right)\right) + \frac{1}{g}\left(\left(\frac{q+\tau}{2q}\right)\left(B^2e^{2rt}\right)\right)\\ -\frac{1}{g}\left(r+\tau\right)Be^{rt}x - \frac{1}{g}\left(q+\tau\right)Be^{qt}y.\end{aligned} \tag{3.117}$$

At the shoreline, the total depth

$$h + \zeta = 0. \tag{3.118}$$

Substituting (3.88) and (3.117) in (3.118)

$$\left(x + \frac{a^2}{2gh_0}\left(B(r+\tau)e^{rt}\right)\right)^2 + \left(y + \frac{a^2}{2gh_0}\left(B(q+\tau)e^{qt}\right)\right)^2 = a^2. \tag{3.119}$$

Hence, the projection of the moving shoreline on the xy plane is a circle of radius a, with a moving centre. As $t \to \infty$, the centre of the circle spirals in towards the origin and the shoreline approaches

$$x^2 + y^2 = a^2, \tag{3.120}$$

the shoreline for an undisturbed surface, and $\zeta \to \infty$, so that friction will cause the initial disturbance to eventually die out.

3.5.4 Circular paraboloid: flow for $\tau = p$

When $\tau = p$, the solution of (3.93) is

$$u_0(t) = e^{-\frac{\tau t}{2}}(A + Bt), \tag{3.121}$$

where A and B are constants, obtained by using given values for $u_0(0)$ and $u_0'(0)$.

When $\tau = p$, the solution of (3.94) is

$$v_0(t) = e^{-\frac{\tau t}{2}}(C + Dt) \tag{3.122}$$

where C and D are constants, obtained by using given values for $v_0(0)$ and $v_0'(0)$. It can be seen from equations (3.121) and (3.122) respectively that as $t \to \infty$, $u_0(t) \to 0$ and $v_0(t) \to 0$. If C and D are chosen so that $C = D = 0$ then

$$v_0(t) = 0. \tag{3.123}$$

It can be shown that

$$\zeta(x,t) = -\frac{e^{-\tau t}}{\tau}\left(K + L\left(t + \frac{1}{\tau}\right)\right) - \frac{e^{-\tau t}}{\tau}\left(M\left(t^2 + \frac{2t}{\tau} + \frac{2}{\tau^2}\right)\right) \\ -\frac{x}{g}\left(B + \frac{\tau}{2}(A + Bt)\right)e^{-\frac{\tau t}{2}}, \tag{3.124}$$

where

$$K = \frac{1}{g}\left(AB + \frac{\tau A^2}{2}\right), \tag{3.125}$$

$$L = \frac{1}{g}\left(B^2 + \tau AB\right), \tag{3.126}$$

and

$$M = \frac{\tau B^2}{2g}. \tag{3.127}$$

It can be seen that as $t \to \infty$, $\zeta \to 0$.

At the shoreline, the total depth

$$h + \zeta = 0. \tag{3.128}$$

Substituting (3.88) and (3.124) in (3.128)

$$\left(x + \frac{a^2}{2gh_0} e^{-\frac{\tau t}{2}}\left(B + \frac{\tau}{2}\left(A + Bt\right)\right)\right)^2 + y^2 = a^2. \tag{3.129}$$

Hence the projection of the moving shoreline on the xy plane is a circle, which has a moving centre and whose radius is a .

As $t \to \infty$, the centre of the circle moves in towards the origin and the shoreline approaches

$$x^2 + y^2 = a^2, \tag{3.130}$$

the shoreline for an undisturbed surface, and $\zeta \to 0$, so that friction will cause the initial disturbance to eventually die out.

3.6 New solutions for unforced frictional flow in an elliptical paraboloid

3.6.1 Basic equations

For flow in an elliptical paraboloid assume that

$$h = h_0 \left(1 - \frac{x^2}{a^2} - \frac{y^2}{b^2} \right), \tag{3.131}$$

with h_0, a and b constants. The flow represented could be flow in a lake.

Substituting (3.30), (3.31), (3.41), and (3.131) in (3.29) gives

$$\frac{d\zeta_0(t)}{dt} + x\frac{d\zeta_1(t)}{dt} + y\frac{d\zeta_2(t)}{dt} - \frac{2u_0(t)h_0 x}{a^2} + u_0(t)\zeta_1(t) - \frac{2v_0(t)h_0 y}{b^2} + v_0(t)\zeta_2(t) = 0. \tag{3.132}$$

Equating the time-varying coefficients of the linearly independent terms 1, x, and y, respectively,

$$\frac{d\zeta_0(t)}{dt} + u_0(t)\zeta_1(t) + v_0(t)\zeta_2(t) = 0, \tag{3.133}$$

$$\frac{d\zeta_1(t)}{dt} - \frac{2u_0(t)h_0}{a^2} = 0, \tag{3.134}$$

$$\frac{d\zeta_2(t)}{dt} - \frac{2v_0(t)h_0}{b^2} = 0. \tag{3.135}$$

Substituting (3.34) in (3.134)

$$\frac{d^2u_0(t)}{dt^2} + \tau\frac{du_0(t)}{dt} + \frac{2gh_0u_0(t)}{a^2} = 0. \tag{3.136}$$

Substituting (3.36) in (3.135)

$$\frac{d^2v_0(t)}{dt^2} + \tau\frac{dv_0(t)}{dt} + \frac{2gh_0v_0(t)}{b^2} = 0. \tag{3.137}$$

Equations (3.136) and (3.137) have to be solved for $u_0(t)$ and $v_0(t)$.

As equations (3.136) and (3.137) are both second order differential equations, each equation requires two boundary conditions. The solution of (3.136) can be substituted in (3.34) to find $\zeta_1(t)$ and the solution of (3.137) can be substituted in (3.36) to find $\zeta_2(t)$. The solutions of (3.136) and (3.137) plus the solutions for $\zeta_1(t)$ and $\zeta_2(t)$ can be substituted in (3.133), which is first order and hence needs one boundary condition to be solved uniquely for $\zeta_0(t)$.

The auxiliary equation for (3.136) is

$$\lambda^2 + \tau\lambda + \frac{2gh_0}{a^2} = 0. \tag{3.138}$$

The roots of (3.138) are

$$\lambda = \frac{-\tau \pm \sqrt{\tau^2 - p^2}}{2}, \tag{3.139}$$

where p is defined by

$$p = \sqrt{\frac{8gh_0}{a^2}}. \tag{3.140}$$

In this thesis, only the case for λ complex (i.e. for $\tau < p$) is discussed. In this case the solution of (3.136) is

$$u_0(t) = e^{-\frac{\tau t}{2}} \left(A \cos\left(\frac{\sqrt{p^2 - \tau^2}}{2} t \right) + B \sin\left(\frac{\sqrt{p^2 - \tau^2}}{2} t \right) \right), \tag{3.141}$$

where A and B are constants, obtained by using given values for $u_0(0)$ and $u_0'(0)$. The constant A will be chosen to be zero and B to be nonzero. Hence

$$u_0(t) = Be^{-\tau t/2} \sin st, \tag{3.142}$$

where

$$s = \frac{\sqrt{p^2 - \tau^2}}{2}. \tag{3.143}$$

Similarly, the auxiliary equation for (3.137) is

$$m^2 + \tau m + \frac{2gh_0}{a^2} = 0. \tag{3.144}$$

The roots of (3.144) are

$$m = \frac{-\tau \pm \sqrt{\tau^2 - n^2}}{2}, \tag{3.145}$$

where n is defined by

$$n = \sqrt{\frac{8gh_0}{b^2}}. \tag{3.146}$$

In this thesis, only the case for m complex (i.e. for $\tau < n$) is discussed. In this case the solution of (3.137) is

$$v_0(t) = e^{-\frac{\tau t}{2}} \left(C \cos\left(\frac{\sqrt{n^2 - \tau^2}}{2} t \right) + D \sin\left(\frac{\sqrt{n^2 - \tau^2}}{2} t \right) \right), \tag{3.147}$$

where C and D are constants, obtained by using given values $v_0(0)$ and $v_0'(0)$. The constant C will be chosen to be zero and D to be equal to B. Hence

$$v_0(t) = Be^{-(\tau t)/2} \sin kt, \tag{3.148}$$

where

$$k = \frac{\sqrt{n^2 - \tau^2}}{2}. \tag{3.149}$$

The solutions of the shallow water equations sought will be for $\tau < p$ and $\tau < n$. Substituting (3.142) in (3.34)

$$\zeta_1(t) = -\frac{e^{-\frac{\tau t}{2}}}{g} \left(Bs \cos st + \frac{\tau B}{2} \sin st \right). \tag{3.150}$$

Substituting (3.148) in (3.36)

$$\zeta_2(t) = -\frac{e^{-\frac{\tau t}{2}}}{g} \left(Bk \cos kt + \frac{\tau B}{2} \sin kt \right). \tag{3.151}$$

Substituting (3.142), (3.148), (3.150) and (3.151) into (3.133) and integrating with respect to t gives

$$\begin{aligned}\zeta_0(t) = {}& \frac{a^2B^2e^{-\tau t}}{8g^2h_0}\left(-s\tau\sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right)\\ &+\frac{b^2B^2e^{-\tau t}}{8g^2h_0}\left(-k\tau\sin 2kt + \left(\frac{\tau^2}{4} - k^2\right)\cos 2kt\right)\\ &\qquad -\frac{B^2e^{-\tau t}}{2g},\end{aligned} \tag{3.152}$$

with the constant of integration being zero because it is assumed that as $t \to \infty$, $\zeta_0(t) \to 0$. Substituting (3.150), (3.151), and(3.152) into (3.41)

$$\begin{aligned}\zeta(x,y,t) = {}& \frac{a^2B^2e^{-\tau t}}{8g^2h_0}\left(-s\tau\sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right)\\ &+\frac{b^2B^2e^{-\tau t}}{8g^2h_0}\left(-k\tau\sin 2kt + \left(\frac{\tau^2}{4} - k^2\right)\cos 2kt\right)\\ &-\frac{B^2e^{-\tau t}}{2g} - \frac{e^{-\frac{\tau t}{2}}}{g}\left(Bs\cos st + \frac{\tau B}{2}\sin st\right)x\\ &-\frac{e^{-\frac{\tau t}{2}}}{g}\left(Bk\cos kt + \frac{\tau B}{2}\sin kt\right)y.\end{aligned} \tag{3.153}$$

It can be seen that as $t \to \infty$ $\zeta(t) \to 0$, i.e. the displacement of the fluid from equilibrium gradually dies out over time, which is the result that one would expect with a bottom friction force acting on the fluid.

At the shoreline, the total depth

$$h + \zeta = 0. \tag{3.154}$$

Substituting (3.131) and (3.153) in (3.154) gives

$$\begin{aligned}&(1/(a^2))\left(x - \frac{a^2e^{-\frac{\tau t}{2}}}{2h_0g}\left(-Bs\cos st - \frac{\tau B}{2}\sin st\right)\right)^2\\ &+(1/(b^2))\left(y - \frac{b^2e^{-\frac{\tau t}{2}}}{2h_0g}\left(-Bk\cos kt - \frac{\tau B}{2}\sin kt\right)\right)^2 = 1.\end{aligned} \tag{3.155}$$

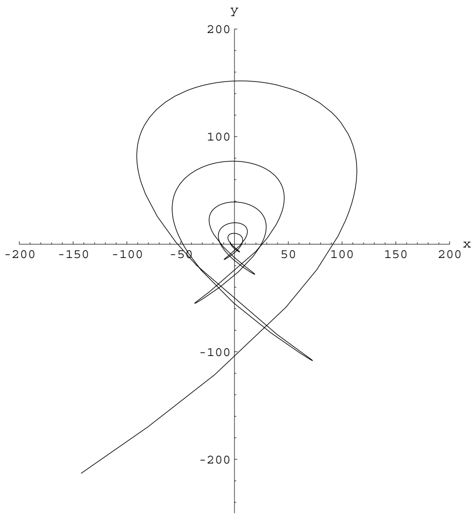

Figure 3.14: The path of the centre of the projection on the xy plane of the moving shoreline from $t = 0$ s to $t = 7200$ s for flow in an elliptical paraboloid with $a = 2$ km, $b = 3$ km, $B = 1$ ms^{-1}, $h_0 = 10$ km, and $\tau = 0.001$ s^{-1}. Dimensions are in metres .

Hence, the projection of the moving shoreline on the xy plane is an ellipse. The path of the centre of the moving ellipse for $t = 0$ to $t = 7200$ s. for $a =$ 2 km., $b = 3$ km., $h_0 = 10$ m, ($\Rightarrow p = 0.0141$ s^{-1}, $n = 0.0094$ s $^{-1}$), $\tau = 0.001$ s $^{-1}$ and $B = 1$ is shown in Figure 3.14.

3.7 New solutions for forced flow above a bed with quadratically varying depth

3.7.1 Introduction

Carrier and Greenspan [11] obtained moving boundary analytical solutions of the one dimensional nonlinear shallow water wave equations for motion caused by periodic forcing in a frictionless fluid above a bed of constant slope. Johns [40] expressed these results in a simpler form. In both the solutions by Carrier and Greenspan and the solutions by Johns ζ and u are implicitly defined.

In this section moving boundary analytical solutions of the two dimensional nonlinear shallow water wave equations for motion caused by periodic forcing in a frictionless fluid above a bed with quadratically varying depth are derived. In addition moving boundary analytical solutions of the two dimensional nonlinear shallow water wave equations for forced motion in a linear frictional fluid above a bed with quadratically varying depth are derived; the forced motion decays over time. These latter solutions are similar to those for unforced moving boundary motion in a parabolic canal as given in Section 3.4. In both of the solutions ζ and u are explicitly defined.

Consider the case of motion of shallow water above a bed with quadratically varying depth defined by

$$h = h_0 \left(1 - \frac{x^2}{a^2} \right), \ x \geq 0. \tag{3.156}$$

It will be assumed that the motion is one dimensional and the velocity is a function of time only and hence

$$U = u_0(t), \tag{3.157}$$

$$V = 0. \tag{3.158}$$

Hence, using equations (3.29), (3.34), (3.36), (3.41), (3.156), (3.157) and (3.158) it follows, using similar reasoning to that in Section 3.3 and Subsection 3.4.1, that

$$\zeta(x, y, t) = \zeta_0(t) + x\zeta_1(t), \tag{3.159}$$

$$\frac{d\zeta_0\,(t)}{dt} + u_0(t)\zeta_1\,(t) = 0, \tag{3.160}$$

and

$$\zeta_1(t) = -\frac{1}{g}\left(\frac{du_0(t)}{dt} + \tau u_0(t)\right), \tag{3.161}$$

and

$$\frac{d^2u_0\left(t\right)}{dt^2} + \tau\frac{du_0\left(t\right)}{dt} + \frac{2gh_0u_0\left(t\right)}{a^2} = 0. \tag{3.162}$$

3.7.2 Frictionless flow

The solution given below is a modification of Thacker's solution for frictionless flow in a parabolic canal as discussed in section 3.2.

Assume that the motion is frictionless, i.e.

$$\tau = 0, \tag{3.163}$$

and subject to forcing

$$\zeta(0,t) = P\cos(\omega t). \tag{3.164}$$

Equations (3.162) and (3.163) imply that

$$\frac{d^2u_0\left(t\right)}{dt^2} + \frac{2gh_0u_0\left(t\right)}{a^2} = 0. \tag{3.165}$$

The general solution of (3.165) is

$$u_0(t) = A\cos\psi t + B\sin\psi t, \tag{3.166}$$

where

$$\psi = \frac{\sqrt{2gh_0}}{a}. \tag{3.167}$$

Substituting (3.163) and (3.166) into (3.161) gives

$$\zeta_1(t) = -\frac{1}{g}\left(-\psi A\sin\psi t + \psi B\cos\psi t\right). \tag{3.168}$$

Substituting (3.166) and (3.168) in (3.160) gives

$$\frac{\partial\zeta_0}{\partial t} = -\frac{\psi(A^2 - B^2)\sin 2\psi t}{2g} + \frac{\psi AB\cos 2\psi t}{g}. \tag{3.169}$$

Integrating (3.169) with respect to t gives

$$\zeta_0(x,t) = \frac{(A^2 - B^2)\cos 2\psi t}{4g} + \frac{AB\sin 2\psi t}{2g} + C, \tag{3.170}$$

where C is a constant. Substituting (3.168) and (3.170) into (3.159) gives

$$\zeta(x,t) = \frac{(A^2 - B^2)\cos 2\psi t}{4g} + \frac{AB\sin 2\psi t}{2g} + C - \frac{1}{g}\left(-\psi A\sin\psi t + \psi B\cos\psi t\right)x. \tag{3.171}$$

Equation (3.171) implies that the boundary condition given in (3.164) will be satisfied if

$$\omega = 2\psi = \sqrt{\frac{8gh_0}{a^2}}, \tag{3.172}$$

and

$$C = 0, \tag{3.173}$$

and

$$AB = 0, \tag{3.174}$$

and

$$P = \frac{A^2 - B^2}{4g}. \tag{3.175}$$

It follows from (3.174) that either $A = 0$ or $B = 0$. If $A = 0$, then

$$\zeta(x,t) = \frac{-B^2\cos 2\psi t}{4g} - \frac{(\psi B\cos\psi t)x}{g} \tag{3.176}$$

and

$$U = B\sin\psi t, \tag{3.177}$$

and the forcing function is

$$\zeta(0,t) = -\frac{B^2}{4g}\cos\left(\frac{2\sqrt{2gh_0}t}{a}\right), \tag{3.178}$$

If $B = 0$, then

$$\zeta(x,t) = \frac{A^2\cos 2\psi t}{4g} + \frac{(\psi A\sin\psi t)x}{g}, \tag{3.179}$$

and

$$U = A\cos\psi t, \tag{3.180}$$

and the forcing function is

$$\zeta(0,t) = \frac{A^2}{4g}\cos\left(\frac{2\sqrt{2gh_0}t}{a}\right). \tag{3.181}$$

At the shoreline, the total depth

$$h + \zeta = 0. \tag{3.182}$$

Consider the shoreline for $A = 0$. Substituting (3.156) and (3.176) into (3.182) gives

$$\left(x - \frac{Ba^2}{2h_0 g}(-\psi\cos\psi t)\right)^2 = a^2(1 + \frac{B^2}{4h_0 g}). \tag{3.183}$$

Hence, the x coordinate of the shoreline is given by

$$x = a\sqrt{1 + \frac{B^2}{4gh_0}} - \frac{Ba}{\sqrt{2gh_0}}\cos\frac{\sqrt{2gh_0}t}{a}. \tag{3.184}$$

Consider a bed with quadratically varying depth for which $a = 3$ km, $h_0 = 10$ m and for motion in which $B = 5$ ms^{-1} and $A = 0$ ms^{-1}. The period of the motion is 1345.71 s. The development of the motion from $t = 0$ to $t = 1270.95$ seconds is shown in Figure 3.15. The periodic forcing at $x = 0$ is shown in Figure 3.16. The vertical displacement of the water surface from equilibrium as a function of time at $x = 0$ is shown in Figure 3.17. The velocity of the fluid as a function of time is shown in Figure 3.18. The x-coordinate of the left hand shoreline as a function of time is shown in Figure 3.19.

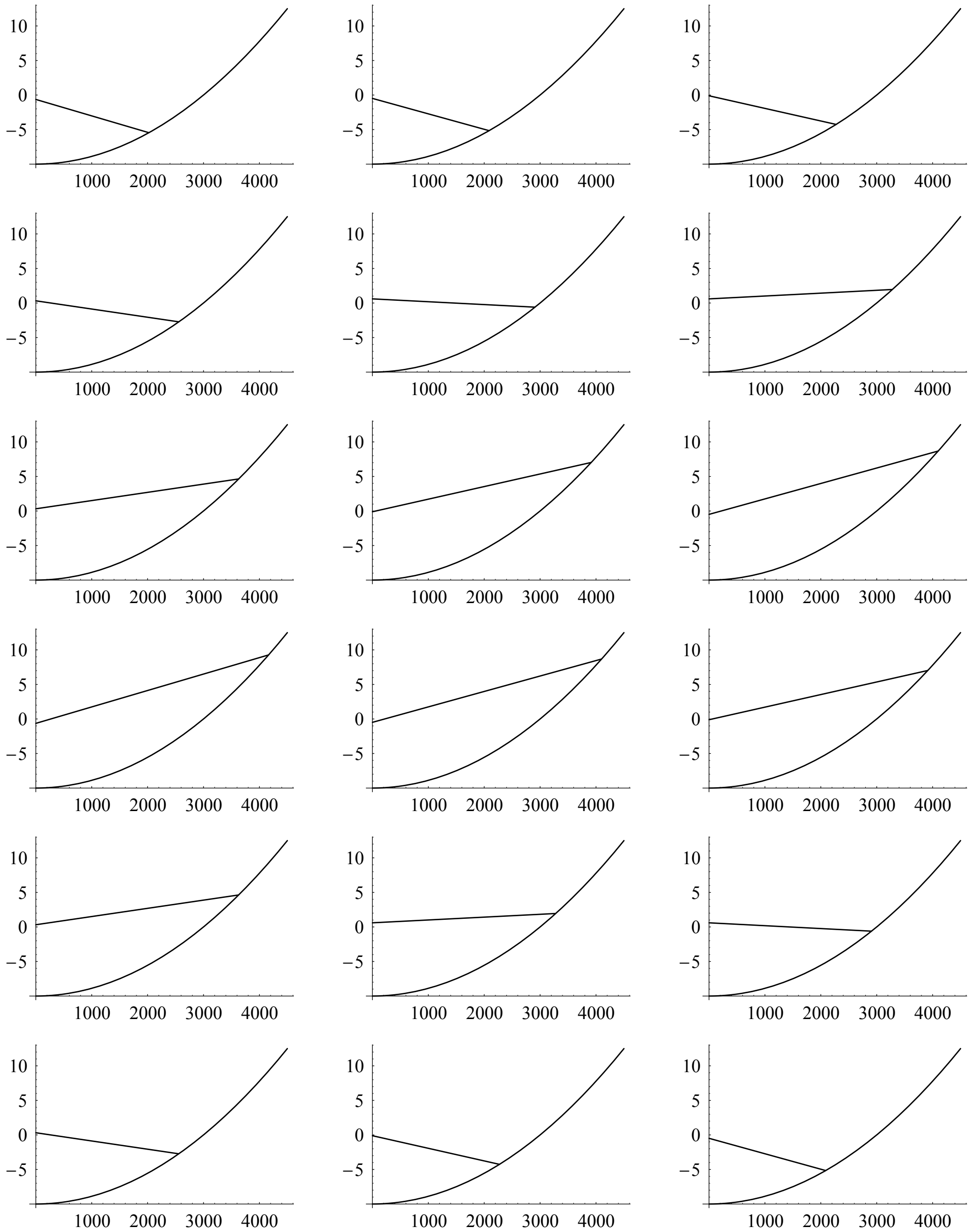

Figure 3.15: The development of the motion of fluid, for a bed with quadratically varying depth with $a = 3$ km, $h_0 = 10$ m, and $B = 5$ ms^{-1}, from $t = 0$ s to $t = 1270.95$ s, in increments of 74.762 s. Dimensions are in metres.

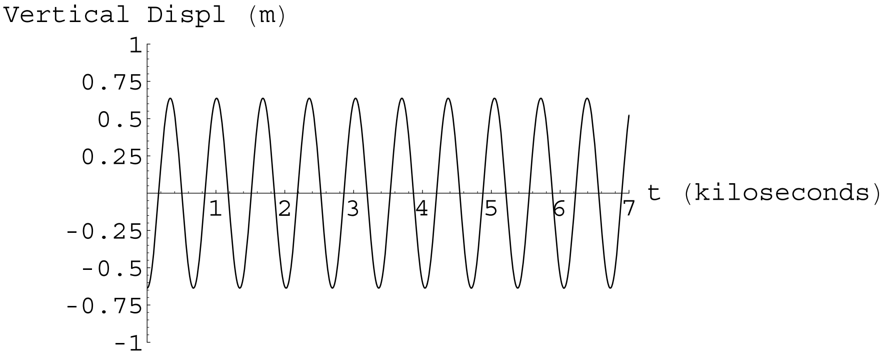

Figure 3.16: The periodic forcing of the water surface at $x = 0$ km above a bed with quadratically varying depth with $a = 3$ km, $h_0 = 10$ m, and $B = 5$ ms^{-1}. Dimensions are in metres on the vertical axis and thousands of seconds on the horizontal axis.

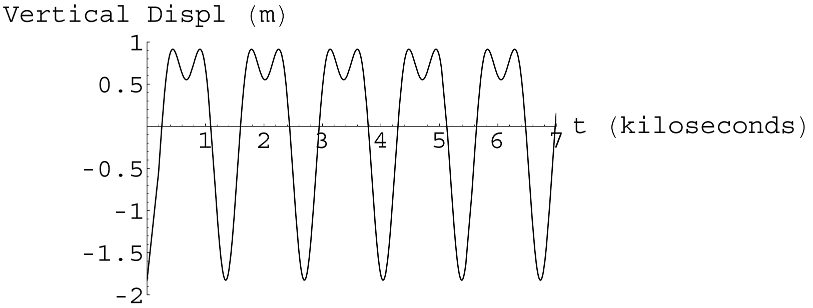

Figure 3.17: The vertical displacement of the water surface at $x = 500$ m for a bed with quadratically varying depth with $a = 3$ km, $h_0 = 10$ m, and $B = 5$ ms $^{-1}$. Dimensions are in metres on the vertical axis and thousands of seconds on the horizontal axis.

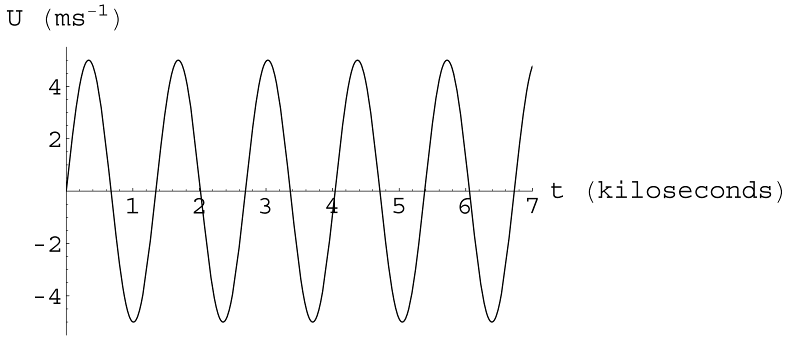

Figure 3.18: The velocity at any time for a bed with quadratically varying depth with $a = 3$ km, $h_0 = 10$ m, and $B = 5$ ms^{-1}. Dimensions are in metres per second on the vertical axis and thousands of seconds on the horizontal axis.

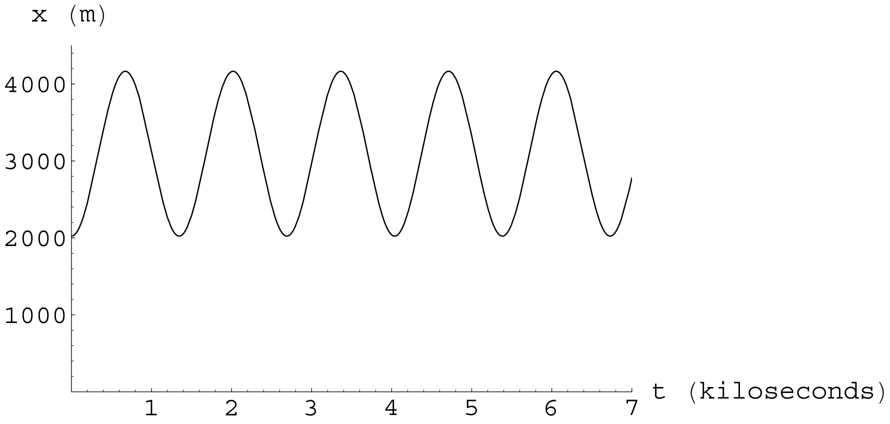

Figure 3.19: The location of the right hand shoreline at any time for a bed with quadratically varying depth with $a = 3$ km, $h_0 = 10$ m, and $B = 5$ ms $^{-1}$. Dimensions are in metres per second on the vertical axis and thousands of seconds on the horizontal axis.

3.7.3 Frictional flow

In this subsection linear frictional flow above a bed with quadratically varying depth will be considered, with $\tau < p$, where

$$p = \sqrt{\frac{8gh_0}{a^2}}, \tag{3.185}$$

subject to a forcing at $x = 0$, given by

$$\zeta(0,t) = e^{-\gamma t}(P\sin\omega t + Q\cos\omega t + R). \tag{3.186}$$

A solution of (3.162) is

$$u_0(t) = Be^{-\frac{\tau t}{2}}\sin st, \tag{3.187}$$

where

$$s = \frac{\sqrt{p^2 - \tau^2}}{2}. \tag{3.188}$$

Substituting (3.187) in (3.161) gives

$$\zeta_1(t) = -\frac{e^{-\tau t}}{g}\left(Bs\cos st + \frac{\tau B}{2}\sin st\right). \tag{3.189}$$

Substituting (3.187) and (3.189) in (3.160) and integrating with respect to t gives

$$\zeta_0(t) = \frac{a^2B^2e^{-\tau t}}{8g^2h_0}\left(-s\tau\sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right) - \frac{B^2e^{-\tau t}}{4g}, \tag{3.190}$$

with the constant of integration being zero because it is assumed that as $t \to \infty$ $\zeta_0(t) \to 0$. Substituting (3.189) and (3.190) in (3.159) gives

$$\begin{aligned}\zeta(x,t) = \frac{a^2B^2e^{-\tau t}}{8g^2h_0}\left(-s\tau\sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right) - \frac{B^2e^{-\tau t}}{4g} \\ -\frac{e^{-\frac{\tau t}{2}}}{g}\left(Bs\cos st + \frac{\tau B}{2}\sin st\right)x.\end{aligned} \tag{3.191}$$

Using equation (3.191) it can be seen that (3.186) will be satisfied if

$$\gamma = \tau, \tag{3.192}$$

$$\omega = \sqrt{\frac{8gh_0}{a^2} - \tau^2}, \tag{3.193}$$

$$P = -\frac{a^2B^2s\tau}{8g^2h_0}, \tag{3.194}$$

$$Q = \frac{a^2B^2}{8g^2h_0}(\frac{\tau^2}{4} - s^2), \tag{3.195}$$

and

$$R = -\frac{B^2}{4g}. \tag{3.196}$$

At the shoreline, the total depth

$$h + \zeta = 0. \tag{3.197}$$

Substituting (3.156) and (3.191) in (3.197) the resulting equation simplifies to

$$x = \frac{a^2e^{-\frac{\tau t}{2}}}{2h_0g}\left(-Bs\cos st - \frac{\tau B}{2}\sin st\right) + a, \tag{3.198}$$

the x-coordinate of the shoreline.

3.8 Conclusions

In this chapter exact solutions of the two dimensional nonlinear shallow water wave equations in the case of unforced flow involving linear bottom friction and without the Coriolis force have been found for flow in parabolic canals, circular paraboloids and elliptical paraboloids. These solutions involve moving shorelines. The motion decays over time, which is what one would expect in a motion involving friction and no input force. In contrast Thacker found exact solutions of the two dimensional nonlinear shallow water wave equations in the case of flow involving the Coriolis force but without bottom friction for flow in a parabolic canal. These solutions also involve moving shorelines. The motion is oscillatory and continues indefinitely over time, which is what one would expect in a motion involving no friction.

In addition analytical moving boundary solutions of the two dimensional nonlinear shallow water wave equations in the case of forced flow above a bed with quadratically varying depth have been found both without and with bottom friction, the latter being linear.

The solutions found in this chapter are useful for testing numerical solutions of the nonlinear shallow water wave equations which include linear bottom friction and whose flow involves moving shorelines. Testing of the analytical solutions for forced frictionless flow above a bed with quadratically varying depth against a numerical model in Chapter 6 of this thesis will show the numerical results to be in close agreement with the analytical solutions; likewise for testing of the analytical solutions for forced linear frictional flow above a bed with quadratically varying depth against a numerical model.

Chapter 4

Fixed boundary analytical solutions of the nonlinear shallow water wave equations

4.1 Introduction

In this chapter new fixed boundary analytical solutions of the one dimensional nonlinear shallow water wave equations will be developed. The testing of these solutions against numerical solutions will be discussed in Chapter 5. The analytical solutions developed are a modification of solutions developed by Airy in 1845 (discussed in [55] and [83]).

4.2 Airy's solutions

In 1845 Airy investigated tidal flow in a channel of constant mean water depth, h, defined for $0 \leq x \leq \infty$, for when ζ, the water level, is not small compared with h. The equations that he used are based on equations (2.1), (2.2) and (2.3) with wind stresses, bottom friction, viscosity and the Coriolis forces omitted, but the nonlinear continuity and advective terms included. The resulting nonlinear shallow water equations are

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + g\frac{\partial \zeta}{\partial x} = 0, \tag{4.1}$$

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (h+\zeta)U}{\partial x} = 0. \tag{4.2}$$

The canal communicates at its mouth, $x = 0$, with the open sea, where the water elevation is given by

$$\zeta(0,t) = a_1 \cos \omega t. \tag{4.3}$$

To solve this problem Airy used the method of successive approximation. The solutions for ζ and U have terms involving ω, the angular frequency of the tidal oscillation plus terms involving twice the frequency, 2ω, the latter terms representing overtides, or tides of the second order. Airy's analysis did not extend to tides of higher order, i.e. whose frequencies are three, four or more times that of that of the forced frequency.

As a first approximation to the nonlinear shallow water equations we have the linear equations

$$\frac{\partial U}{\partial t} + g\frac{\partial \zeta}{\partial x} = 0, \tag{4.4}$$

$$\frac{\partial \zeta}{\partial t} + h\frac{\partial U}{\partial x} = 0. \tag{4.5}$$

The solutions of (4.4) and (4.5) which are consistent with (4.3) are

$$\zeta = a_1 \cos \omega \left(t - \frac{x}{c}\right), \tag{4.6}$$

$$U = \frac{g a_1}{c} \cos \omega \left(t - \frac{x}{c}\right), \tag{4.7}$$

where c, the wave speed, is given by

$$c = \sqrt{gh}. \tag{4.8}$$

The wave is a progressive wave.

As a second approximation these values of ζ and U in (4.6) and (4.7) are substituted in the nonlinear terms of (4.1) and (4.2) to obtain

$$\frac{\partial U}{\partial t} = -g\frac{\partial \zeta}{\partial x} - \frac{\omega g a_1^2}{2ch}\sin 2\omega\left(t - \frac{x}{c}\right), \tag{4.9}$$

$$\frac{\partial \zeta}{\partial t} = -h\frac{\partial U}{\partial x} - \frac{\omega a_1^2}{h}\sin 2\omega\left(t - \frac{x}{c}\right). \tag{4.10}$$

Eliminating U from equations (4.9) and (4.10) one obtains

$$\frac{\partial^2 \zeta}{\partial t^2} = gh\frac{\partial^2 \zeta}{\partial x^2} - \kappa\cos 2\omega\left(t - \frac{x}{c}\right), \tag{4.11}$$

where

$$\kappa = \frac{a_1^2 g\omega^2}{c^2} + \frac{2a_1^2\omega^2}{h}. \tag{4.12}$$

Neither Lamb [55] nor Rahman [83] explained how Airy solved (4.11); it can be solved by assuming that

$$\zeta = a_1\cos\omega\left(t - \frac{x}{c}\right) + Ex\cos 2\omega\left(t - \frac{x}{c}\right) + Fx\sin 2\omega\left(t - \frac{x}{c}\right), \tag{4.13}$$

where E and F are constants. The constants E and F can be found by substituting (4.13) in (4.11) and equating coefficients of $\cos\omega(t - \frac{x}{c})$ and $\sin\omega(t - \frac{x}{c})$, giving

$$\zeta = a_1\cos\omega\left(t - \frac{x}{c}\right) - \left(\frac{3a_1^2 g\omega}{4c^3}\right)x\sin 2\omega\left(t - \frac{x}{c}\right). \tag{4.14}$$

Eliminating ζ from equations (4.9) and (4.10) one obtains

$$\frac{\partial^2 U}{\partial t^2} = gh\frac{\partial^2 U}{\partial x^2} - \frac{3a_1^2 g\omega^2}{ch}\cos 2\omega\left(t - \frac{x}{c}\right). \tag{4.15}$$

If one assumes that

$$\begin{aligned} U = \frac{ga_1}{c}\cos\omega\left(t-\frac{x}{c}\right) + P\cos 2\omega\left(t-\frac{x}{c}\right) + Q\sin 2\omega\left(t-\frac{x}{c}\right) \\ +Rx\cos 2\omega\left(t-\frac{x}{c}\right) + Sx\sin 2\omega\left(t-\frac{x}{c}\right), \end{aligned} \tag{4.16}$$

where P, Q, R and S are constants, then substitutes (4.16) in (4.15) one obtains

$$R = 0, \tag{4.17}$$

$$S = -\frac{3a_1^2\omega}{4h^2}. \tag{4.18}$$

Substituting (4.17) and (4.18) in (4.16) gives

$$\begin{aligned} U = \frac{ga_1}{c}\cos\omega\left(t-\frac{x}{c}\right) + P\cos 2\omega\left(t-\frac{x}{c}\right) + Q\sin 2\omega\left(t-\frac{x}{c}\right) \\ -\frac{3a_1^2\omega}{4h^2}x\sin 2\omega\left(t-\frac{x}{c}\right). \end{aligned} \tag{4.19}$$

Substituting (4.14) and (4.19) in (4.9) gives

$$P = -\frac{a_1^2 g}{8ch}, \tag{4.20}$$

$$Q = 0. \tag{4.21}$$

Substituting (4.20) and (4.21)in (4.19) gives

$$\begin{aligned} U = \frac{ga_1}{c}\cos\omega\left(t-\frac{x}{c}\right) - \frac{a_1^2 g}{8ch}\cos 2\omega\left(t-\frac{x}{c}\right) \\ -\frac{3a_1^2\omega}{4h^2}x\sin 2\omega\left(t-\frac{x}{c}\right). \end{aligned} \tag{4.22}$$

From equation (4.14) it can be seen that the approximate solution will be valid provided that the amplitude of the overtide term (i.e. the second order term) is small compared with the amplitude of the linear term, i.e. $\frac{a_1\omega x}{ch}$, is small.

4.3 Other authors' solutions

As was discussed in the literature review in Chapter 2, a number of authors have developed solutions for tidal flow in estuaries in which ζ, the water level, is not small compared with the undisturbed water depth, h. Kreiss [54] included linear friction in analysing nonlinear oscillations in a tidal channel of finite length. Kreiss had a tidal oscillation of angular frequency, ω, at the open sea boundary. He obtained a solution for the velocity including a second order overtide using the perturbation method. Proudman [82] included quadratic friction in analysing nonlinear oscillations due to tide and surge in a channel of finite length. He had a prescribed incident wave of first order at the mouth plus a reflected wave of first order plus second order terms which reduced to a reflected wave at the mouth. He obtained solutions for ζ and U by in each case integrating twice a second order partial differential equation. Knight [51] extended Proudman's work. Gallagher and Munk [27], Kabbaj and Provost [43] and DiLorenzo [17] all found second order solutions for quadratic frictional tidal flow in channels of finite length using perturbation methods. Gallagher and Munk and DiLorenzo included an overtide in the open sea forcing expression. At the open sea boundary Kabbaj and Provost specified $U + 2\sqrt{g(h+\zeta)}$ to be $2 + 2A\cos\omega t$, with A and ω constant.

4.4 New solutions involving nonlinear continuity and advection

In this section new analytical solutions are developed. The motion considered here is for tidal flow in a channel of length L with h, the undisturbed depth of the water, assumed constant. The water is subject to forcing at $x = L$ and has a barrier at $x = 0$. The shallow water equations used are the same that Airy used (equations (4.1) and (4.2)) but with different boundary conditions

$$\zeta(L, t) = a_1 \cos \omega t, \tag{4.23}$$

$$U(0, t) = 0. \tag{4.24}$$

Solutions will be obtained using the method of successive aproximation, the method that Airy used.

As a first approximation to the nonlinear equations (4.1) and (4.2), the linear equations (4.4) and (4.5) are used. A solution of these equations which satisfies the boundary conditions (4.23) and (4.24) was obtained by Lynch and Gray [67]

$$\zeta(x, t) = \frac{a_1 \cos c_1 x}{\cos c_1 L} \cos \omega t, \tag{4.25}$$

$$U(x, t) = \frac{g a_1 c_1 \sin c_1 x}{\omega \cos c_1 L} \sin \omega t, \tag{4.26}$$

where

$$c_1 = \omega / c. \tag{4.27}$$

As a second approximation these values of ζ and U are substituted in the nonlinear terms of (4.1) and (4.2) to obtain

$$\frac{\partial U}{\partial t} = -g \frac{\partial \zeta}{\partial x} - \frac{a_1^2 c_1^3 g^2}{4\omega^2 \cos^2 c_1 L} \sin 2c_1 x (1 - \cos 2\omega t), \tag{4.28}$$

$$\frac{\partial \zeta}{\partial t} = -h \frac{\partial U}{\partial x} - \frac{a_1^2 c_1^2 g^2}{2\omega \cos^2 c_1 L} \cos 2c_1 x \sin 2\omega t. \tag{4.29}$$

Eliminating U from (4.28) and (4.29) one obtains

$$\begin{aligned} \frac{\partial^2 \zeta}{\partial t^2} = gh \frac{\partial^2 \zeta}{\partial x^2} + \frac{a_1^2 c_1^4 g^2 h}{2\omega^2 cos^2 c_1 L} \cos 2c_1 x (1 - \cos 2\omega t) \\ - \frac{a_1^2 c_1^2 g}{cos^2 c_1 L} \cos 2c_1 x \cos 2\omega t. \end{aligned} \tag{4.30}$$

To solve (4.30) for ζ and then solve for U (using (4.28)) assume, as a second approximation,

$$\zeta(x,t) = \frac{a_1 \cos c_1 x}{\cos c_1 L} \cos \omega t + f_1(x) \sin 2\omega t + g_1(x) \cos 2\omega t \\ + d_2 \cos 2c_1 x + b_2, \qquad (4.31)$$

$$U(x,t) = \frac{g a_1 c_1 \sin c_1 x}{\omega \cos c_1 L} \sin \omega t + p_1(x) \sin 2\omega t + q_1(x) \cos 2\omega t. \qquad (4.32)$$

where $f_1(x)$, $g_1(x)$, $p_1(x)$ and $q_1(x)$ are functions of x which need to be determined and b_2 and d_2 are constants which need to be determined.

Substituting $x = L$ in (4.31) and making use of (4.23) gives

$$a_1 \cos \omega t + f_1(L) \sin 2\omega t + g_1(L) \cos 2\omega t + d_2 \cos 2c_1 L + b_2 = a_1 \cos \omega t. \quad (4.33)$$

Equation (4.33) implies that

$$b_2 = -d_2 \cos 2c_1 L, \qquad (4.34)$$

$$f_1(L) = 0, \qquad (4.35)$$

and

$$g_1(L) = 0. \qquad (4.36)$$

Equations (4.24) and (4.32) imply that

$$p_1(0) \sin 2\omega t + q_1(0) \cos 2\omega t = 0, \qquad (4.37)$$

which implies that

$$p_1(0) = 0, \qquad (4.38)$$

and

$$q_1(0) = 0. \qquad (4.39)$$

Substituting (4.31) in (4.30) one obtains

$$\begin{aligned}
-4\omega^2 f_1(x)\sin 2\omega t - 4\omega^2 g_1(x)\cos 2\omega t \\
= ghf_1^{''}(x)\sin 2\omega t + ghg_1^{''}(x)\cos 2\omega t \\
-4c_1^2 d_2 gh\cos 2c_1 x \\
+\frac{a_1^2 c_1^4 g^2 h}{2\omega^2\cos^2 c_1 L}\cos 2c_1 x \\
-\frac{a_1^2 c_1^4 g^2 h}{2\omega^2 cos^2 c_1 L}\cos 2c_1 x\cos 2\omega t \\
-\frac{a_1^2 c_1^2 g}{\cos^2 c_1 L}\cos 2c_1 x\cos 2\omega t.
\end{aligned} \tag{4.40}$$

Equating the coefficients of the $\cos 2c_1 x$ terms gives

$$d_2 = \frac{a_1^2 c_1^2 g}{8\omega^2\cos^2 c_1 L}. \tag{4.41}$$

Equations (4.34) and (4.41) imply that

$$b_2 = -\frac{a_1^2 c_1^2 g\cos 2c_1 L}{8\omega^2\cos^2 c_1 L}. \tag{4.42}$$

Equating the coefficients of the $\sin 2\omega t$ terms in (4.40) gives

$$ghf_1^{''}(x) + 4\omega^2 f_1(x) = 0. \tag{4.43}$$

The general solution to (4.43) is

$$f_1(x) = E\sin 2c_1 x + F\cos 2c_1 x, \tag{4.44}$$

where E and F are constants.

Substituting (4.44) in (4.35) implies that

$$E\sin 2c_1 L + F\cos 2c_1 L = 0. \tag{4.45}$$

Substituting (4.31) and (4.32) in (4.28) gives

$$\begin{aligned}
2\omega p_1(x)\cos 2\omega t - 2\omega q_1(x)\sin 2\omega t + gf_1^{'}(x)\sin 2\omega t + gg_1^{'}(x)\cos 2\omega t \\
= \frac{a_1^2 c_1^3 g^2}{4\omega^2\cos^2 c_1 L}\sin 2c_1 x\cos 2\omega t.
\end{aligned} \tag{4.46}$$

Substituting $x = 0$ in (4.46) and making use of (4.38) and (4.39) gives

$$f_1'(0) = 0, \tag{4.47}$$

and

$$g_1'(0) = 0. \tag{4.48}$$

Equations (4.44) and (4.47) imply that

$$E = 0. \tag{4.49}$$

Equations (4.45) and (4.49) imply that either

$$F = 0, \tag{4.50}$$

or

$$\cos 2c_1 L = 0. \tag{4.51}$$

Hence, substituting (4.49) and (4.50) or (4.51) in (4.44) gives

$$f_1(x) = 0. \tag{4.52}$$

Equating the coefficients of the $\cos 2\omega t$ terms in (4.40) gives

$$g_1''(x) + \frac{4\omega^2}{gh} g_1(x) = \frac{3a_1^2 c_1^2 \cos 2c_1 x}{2h \cos^2 c_1 L}. \tag{4.53}$$

The solution of (4.53) which satisfies (4.36) and (4.48) is

$$g_1(x) = \frac{3a_1^2 c_1^2}{8h \cos^2 c_1 L}(-L \tan(2c_1 L) \cos 2c_1 x + x \sin 2c_1 x). \tag{4.54}$$

Substituting (4.41), (4.42), (4.52) and (4.54) in (4.31) gives

$$\begin{aligned} \zeta(x,t) = {} & \frac{a_1 \cos c_1 x}{\cos c_1 L} \cos \omega t + \frac{a_1^2 c_1^2 g}{8\omega^2 \cos^2 c_1 L} \cos 2c_1 x - \frac{a_1^2 c_1^2 g \cos 2c_1 L}{8\omega^2 \cos^2 c_1 L} \\ & + \frac{3a_1^2 c_1}{8h \cos^2 c_1 L}(-L \tan(2c_1 L) \cos 2c_1 x + x \sin 2c_1 x) \cos 2\omega t. \end{aligned} \tag{4.55}$$

The second and third terms, which are zero frequency terms, are generated by the advective term $U\frac{\partial U}{\partial x}$. As with Airy's solution, the approximate solution will be valid provided $\frac{a_1 \omega x}{ch}$ is small.

Substituting (4.32) and (4.55) in (4.28) gives

$$2\omega p_1(x)\cos 2\omega t - 2\omega q_1(x)\sin 2\omega t = (p_2(x) + p_3(x) + p_4(x))\cos 2\omega t. \tag{4.56}$$

where

$$p_2(x) = -\left(\frac{3a_1^2 c_1^2 gL\tan 2c_1 L}{4h\cos^2 c_1 L}\right)\sin c_1 x, \tag{4.57}$$

$$p_3(x) = -\frac{3a_1^2 c_1 g}{8h\cos^2 c_1 L}\left(2c_1 x\cos 2c_1 x + \sin 2c_1 x\right), \tag{4.58}$$

and

$$p_4(x) = \frac{a_1^2 c_1^3 g^2}{4\omega^2\cos^2 c_1 L}\sin 2c_1 x. \tag{4.59}$$

Equating the coefficients of $\cos 2\omega t$ in (4.56) gives

$$p_1(x) = \frac{1}{2\omega}(p_2(x) + p_3(x) + p_4(x)) \tag{4.60}$$

which satisfies (4.38). Equating the coefficients of $\sin 2\omega t$ in (4.56) gives

$$q_1(x) = 0. \tag{4.61}$$

Substituting (4.60) and (4.61) in (4.32), and simplifying, gives

$$U(x,t) = u_1(x)\sin\omega t + u_2(x)\sin 2\omega t, \tag{4.62}$$

where

$$u_1(x) = \frac{a_1 c_1 g\sin c_1 x}{\omega\cos c_1 L}, \tag{4.63}$$

and

$$u_2(x) = -\frac{a_1^2 c_1 g}{8wh\cos^2 c_1 L}\left(\left(3c_1 L \tan c_1 L + \frac{1}{2}\right)\sin 2c_1 x + 3c_1 x \cos 2c_1 x\right). \tag{4.64}$$

The analytical solutions found in this chapter are useful for testing numerical solutions of the nonlinear shallow water wave equations which include nonlinear continuity and advection. In Chapter 5 of this thesis the analytical solutions found in this chapter will be compared with numerical solutions.

Chapter 5

Selective lumped mass matrix numerical method

5.1 Introduction

Kawahara, Hirano and Tsubota [46] developed a two dimensional shallow water finite element model. The numerical scheme for the time variation is an explicit two step scheme. As this scheme involves a combination of lumped and unlumped coefficients, the scheme is called a selective lumping scheme. The selective lumped mass matrix model (SLM) has already been discussed in the literature review in Chapter 2. The SLM method has been discussed in some detail and applied by Ninomiya and Onishi [73], Aramaki et al. [4], Easton, Singh and Goraya [20] and Goraya [29]. Further details will be given of the SLM method in this chapter. The SLM method will be used in this thesis to test against the new analytical solutions discussed in Chapter 4 and to model the effect of proposed channel deepening on the tides in Port Phillip Bay, with the domain of flow in both cases having a fixed boundary. In Chapter 6 the wetting and drying algorithm used in the SLM method will be modified from its original form. The resultant modified SLM model will be tested against the analytical solution for moving boundary flow discussed in subsections 3.7.2 and 3.7.3.

5.2 Derivation

5.2.1 Equations

The SLM numerical model is derived in this section. The shallow water equations used are equations (2.1), (2.2) and (2.3), excluding the wind stress terms. The resultant equations are the conservation of momentum equation in the East direction

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} - fV - \nu\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) + \tau U + g\frac{\partial \zeta}{\partial x} = 0, \tag{5.1}$$

the conservation of momentum equation in the North direction

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + fU - \nu\left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2}\right) + \tau V + g\frac{\partial \zeta}{\partial y} = 0, \tag{5.2}$$

and the continuity (conservation of mass) equation

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (HU)}{\partial x} + \frac{\partial (HV)}{\partial y} = 0. \tag{5.3}$$

5.2.2 Finite element formulation

The first step in developing the SLM scheme to approximately solve the shallow water equations is to apply the Galerkin finite element procedure, in which each equation is multiplied by a weight $w(x, y)$ and then integrated with respect to area over Ω, the horizontal region of flow, with $d\Omega = dxdy$.

With the above operations applied to the continuity equation (5.3) one obtains

$$\iint_\Omega w\frac{\partial \zeta}{\partial t} d\Omega + \iint_\Omega w\frac{\partial (HU)}{\partial x} d\Omega + \iint_\Omega w\frac{\partial (HV)}{\partial y} d\Omega = 0, \tag{5.4}$$

Using integration by parts, Green's theorem, putting w equal to zero on the sea boundary and putting the normal velocity equal to zero on the land boundary, equation (5.4) becomes

$$\iint_{\Omega} w\frac{\partial \zeta}{\partial t} d\Omega - \iint_{\Omega} \left(\frac{\partial w}{\partial x} HU + \frac{\partial w}{\partial y} HV \right) d\Omega = 0, \tag{5.5}$$

The domain Ω is divided into triangular elements, called finite elements. Computationally, these integrals are calculated separately for each of the elements.

The water elevation ζ, the mean velocity components U and V, and the total water depth H are linearly interpolated using

$$\zeta(x, y, t) = \sum_{\alpha} \phi_{\alpha}(x, y)\zeta_{\alpha}(t), \tag{5.6}$$

$$U(x, y, t) = \sum_{\alpha} \phi_{\alpha}(x, y)U_{\alpha}(t), \tag{5.7}$$

$$V(x, y, t) = \sum_{\alpha} \phi_{\alpha}(x, y)V_{\alpha}(t), \tag{5.8}$$

$$H(x, y, t) = \sum_{\alpha} \phi_{\alpha}(x, y)H_{\alpha}(t), \tag{5.9}$$

where the ϕ_{α} are basis functions, $\alpha = 1, 2, ..., N$, where N is the total number of nodes in the domain Ω, and ζ_{α}, U_{α}, V_{α} and H_{α} are nodal values of the corresponding unknowns. The weighting function w is chosen to be $\phi_1, \phi_2, ..., \phi_N$ in turn. In any given triangular element e containing a node with node number α

$$\phi_{\alpha} = \frac{1}{2\Delta^e}(a_{\alpha} + b_{\alpha}x + c_{\alpha}y), \tag{5.10}$$

with a_{α}, b_{α} and c_{α} constants for that element and Δ^e the area of the element. If node number α is not a node in a given element e, then ϕ_{α} is zero in that element. For any triangle, e, the three triangle vertices, called nodes, are numbered 1, 2 and 3 (the numbering being anticlockwise, it being immaterial which node is used as the starting node).

Hence (5.5) can be rewritten as

$$\sum_e \iint_e w\frac{\partial\zeta}{\partial t}de - \sum_e \iint_e \left(\frac{\partial w}{\partial x}HU + \frac{\partial w}{\partial y}HV\right) de = 0, \tag{5.11}$$

with $e = 1, 2, ..., E$, with E being the number of triangular elements. Substituting (5.6), (5.7), (5.8) and (5.9) into (5.11), with the integration taken over a given triangular element, and using $w = \phi_\alpha$, gives N equations, one for each value of α, of the form

$$\sum_e (\iint_e \phi_\alpha \sum_\beta (\phi_\beta \dot{\zeta}_\beta) de - \iint_e \frac{\partial \phi_\alpha}{\partial x} \sum_\gamma (\phi_\gamma H_\gamma) \sum_\beta (\phi_\beta U_\beta) de$$
$$- \iint_e \frac{\partial \phi_\alpha}{\partial y} \sum_\gamma (\phi_\gamma H_\gamma) \sum_\beta (\phi_\beta V_\beta) de) = 0, \tag{5.12}$$

with $\beta = 1, 2, ..., N$, $\gamma = 1, 2, ..., N$. Equation (5.12) can be written in matrix form as

$$\mathbf{M}\dot{\mathbf{z}} - \mathbf{L}\mathbf{H} = \mathbf{0}, \tag{5.13}$$

where $\mathbf{M}$ and $\mathbf{L}$ are matrices, and $\dot{\mathbf{z}}$ and $\mathbf{H}$ are N-dimensional vectors. The global matrices can be written as $\mathbf{M} = \sum_e \mathbf{M}^e$, $\mathbf{L} = \sum_e \mathbf{L}^e$, where $\mathbf{M}^e$ is the contribution to $\mathbf{M}$ for triangular element e, and $\mathbf{L}^e$ is the contribution to $\mathbf{L}$ for triangular element e, while the vectors are

$$\dot{\mathbf{z}} = \begin{pmatrix} \dot{\zeta}_1 \\ . \\ . \\ . \\ \dot{\zeta}_N \end{pmatrix}$$

and

$$\mathbf{H} = \begin{pmatrix} H_1 \\ . \\ . \\ . \\ H_N \end{pmatrix}.$$

The element matrix $\mathbf{M}^e$ is defined by

$$M^e_{\alpha\beta} = \iint_e \phi_\alpha \phi_\beta de, \tag{5.14}$$

while the element matrix $\mathbf{L}^e$ is defined by

$$L^e_{\alpha\beta} = \iint_e \frac{\partial \phi_\alpha}{\partial x} \phi_\beta \sum_\gamma (\phi_\gamma U_\gamma) de + \iint_e \frac{\partial \phi_\alpha}{\partial y} \phi_\beta \sum_\gamma (\phi_\gamma V_\gamma) de.$$

For convenience of notation, the contribution from each triangular element e can be written as

$$\mathbf{M}^e = \frac{\Delta^e}{12} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix},$$

$$\dot{\mathbf{z}}^e = \begin{pmatrix} \dot{\zeta}_1 \\ \dot{\zeta}_2 \\ \dot{\zeta}_3 \end{pmatrix},$$

$$\mathbf{H}^e = \begin{pmatrix} H_1 \\ H_2 \\ H_3 \end{pmatrix},$$

and

$$\mathbf{L}^e = \mathbf{X}^e + \mathbf{Y}^e, \tag{5.15}$$

where

$$\mathbf{X}^e = \frac{1}{24}\left\{ \begin{pmatrix} 2b_1 & b_1 & b_1 \\ 2b_2 & b_2 & b_2 \\ 2b_3 & b_3 & b_3 \end{pmatrix} U_1 + \begin{pmatrix} b_1 & 2b_1 & b_1 \\ b_2 & 2b_2 & b_2 \\ b_3 & 2b_3 & b_3 \end{pmatrix} U_2 + \begin{pmatrix} b_1 & b_1 & 2b_1 \\ b_2 & b_2 & 2b_2 \\ b_3 & b_3 & 2b_3 \end{pmatrix} U_3 \right\}$$

with $b_1 = y_2 - y_3$ and $b_2 = y_3 - y_1$, and $b_3 = y_1 - y_2$, and

$$\mathbf{Y}^e = \frac{1}{24}\left\{ \begin{pmatrix} 2c_1 & c_1 & c_1 \\ 2c_2 & c_2 & c_2 \\ 2c_3 & c_3 & c_3 \end{pmatrix} V_1 + \begin{pmatrix} c_1 & 2c_1 & c_1 \\ c_2 & 2c_2 & c_2 \\ c_3 & 2c_3 & c_3 \end{pmatrix} V_2 + \begin{pmatrix} c_1 & c_1 & 2c_1 \\ c_2 & c_2 & 2c_2 \\ c_3 & c_3 & 2c_3 \end{pmatrix} V_3 \right\}$$

with $c_1 = x_3 - x_2$ and $c_2 = x_1 - x_3$, and $c_3 = x_2 - x_1$.

The continuity equation can be simplified with equation (5.3) replaced by the equation

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (hU)}{\partial x} + \frac{\partial (hV)}{\partial y} = 0. \tag{5.16}$$

The Galerkin finite element procedure is applied to equation (5.16) with h linearly interpolated using

$$h(x, y, t) = \sum_{\alpha} \phi_\alpha(x, y) h_\alpha(t). \tag{5.17}$$

One obtains an equation similar to equation (5.13), the only change being that $\mathbf{H}$ is replaced by $\mathbf{h}$, where

$$\mathbf{h} = \begin{pmatrix} h_1 \\ . \\ . \\ . \\ h_N \end{pmatrix}.$$

It follows that for the linear case, equation (5.13) is replaced with

$$\mathbf{M\dot{z}} - \mathbf{Lh} = \mathbf{0}. \tag{5.18}$$

The contributions of each triangular element can be written as $\mathbf{M}^e$, $\dot{\mathbf{z}}^e$ and $\mathbf{L}^e$ are defined as for the nonlinear case and

$$\mathbf{h}^e = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix},$$

and $\mathbf{h}^e$ can be replaced by

$$\mathbf{h}^e_{av} = h^e_{av} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

where

$$h^e_{av} = \frac{h_1 + h_2 + h_3}{3},$$

i.e. h_{av} is the average nodal depth for a given triangle.

Using a similar method to that described above for the continuity equation, the momentum equations (5.1) and (5.2) are transformed into matrix terms

$$\mathbf{M}\dot{\mathbf{U}} + \mathbf{AU} - \mathbf{NV} - \mathbf{DU} + \mathbf{BU} + g\mathbf{Pz} = \mathbf{0}, \qquad (5.19)$$

and

$$\mathbf{M}\dot{\mathbf{V}} + \mathbf{AV} + \mathbf{NU} - \mathbf{DV} + \mathbf{BV} + g\mathbf{Qz} = \mathbf{0}. \qquad (5.20)$$

The global matrices can be written as $\mathbf{M} = \sum_e \mathbf{M}^e$, $\mathbf{A} = \sum_e \mathbf{A}^e$, $\mathbf{P} = \sum_e \mathbf{P}^e$, $\mathbf{N} = \sum_e \mathbf{N}^e$, $\mathbf{B} = \sum_e \mathbf{B}^e$, $\mathbf{D} = \sum_e \mathbf{D}^e$, $\mathbf{Q} = \sum_e \mathbf{Q}^e$, while the vectors are

$$\dot{\mathbf{U}} = \begin{pmatrix} \dot{U}_1 \\ . \\ . \\ . \\ \dot{U}_N \end{pmatrix},$$

$$\mathbf{U} = \begin{pmatrix} U_1 \\ . \\ . \\ . \\ U_N \end{pmatrix},$$

$$\dot{\mathbf{V}} = \begin{pmatrix} \dot{V}_1 \\ . \\ . \\ . \\ \dot{V}_N \end{pmatrix},$$

$$\mathbf{V} = \begin{pmatrix} V_1 \\ . \\ . \\ . \\ V_N \end{pmatrix},$$

$$\mathbf{z} = \begin{pmatrix} \zeta_1 \\ . \\ . \\ . \\ \zeta_N \end{pmatrix}.$$

The element matrix $\mathbf{M}^e$ was defined in (5.14) while the other element matrices are defined by

$$A^e_{\alpha\beta} = \sum_\gamma \left\{ \left(\iint_e \phi^e_\alpha \phi^e_\beta \frac{\partial \phi^e_\gamma}{\partial x} de \right) U^e_\gamma + \left(\iint_e \phi^e_\alpha \phi^e_\beta \frac{\partial \phi^e_\gamma}{\partial y} de \right) V^e_\gamma \right\}, \qquad (5.21)$$

$$P^e_{\alpha\beta} = \iint_e \phi^e_\alpha \frac{\partial \phi^e_\beta}{\partial x} de, \qquad (5.22)$$

$$N^e_{\alpha\beta} = f \iint_e \phi^e_\alpha \phi^e_\beta de, \tag{5.23}$$

$$B^e_{\alpha\beta} = \iint_e \tau^e \phi^e_\alpha \phi^e_\beta de, \tag{5.24}$$

$$D^e_{\alpha\beta} = \nu \iint_e \frac{\partial \phi^e_\alpha}{\partial x} \frac{\partial \phi^e_\beta}{\partial x} + \frac{\partial \phi^e_\alpha}{\partial y} \frac{\partial \phi^e_\beta}{\partial y} de, \tag{5.25}$$

$$Q^e_{\alpha\beta} = \iint_e \phi^e_\alpha \frac{\partial \phi^e_\beta}{\partial y} de. \tag{5.26}$$

For convenience of notation the contribution from each triangular element can be written as

$$\dot{\mathbf{U}}^e = \begin{pmatrix} \dot{U}_1 \\ \dot{U}_2 \\ \dot{U}_3 \end{pmatrix},$$

$$\dot{\mathbf{V}}^e = \begin{pmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{V}_3 \end{pmatrix},$$

$$\mathbf{U}^e = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix},$$

$$\mathbf{V}^e = \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix},$$

$$\mathbf{z}^e = \begin{pmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{pmatrix},$$

$$\mathbf{A}^e = \mathbf{S}^e + \mathbf{T}^e,$$

where

$$\mathbf{S}^e = \frac{1}{24}\left\{\begin{pmatrix} 2b_1 & 2b_2 & 2b_3 \\ b_1 & b_2 & b_3 \\ b_1 & b_2 & b_3 \end{pmatrix} U_1 + \begin{pmatrix} b_1 & b_2 & b_3 \\ 2b_1 & 2b_2 & 2b_3 \\ b_1 & b_2 & b_3 \end{pmatrix} U_2 + \begin{pmatrix} b_1 & b_2 & b_3 \\ b_1 & b_2 & b_3 \\ 2b_1 & 2b_2 & 2b_3 \end{pmatrix} U_3 \right\}$$

and

$$\mathbf{T}^e = \frac{1}{24}\left\{\begin{pmatrix} 2c_1 & 2c_2 & 2c_3 \\ c_1 & c_2 & c_3 \\ c_1 & c_2 & c_3 \end{pmatrix} V_1 + \begin{pmatrix} c_1 & c_2 & c_3 \\ 2c_1 & 2c_2 & 2c_3 \\ c_1 & 2c_2 & c_3 \end{pmatrix} V_2 + \begin{pmatrix} c_1 & c_2 & c_3 \\ c_1 & c_2 & c_3 \\ 2c_1 & 2c_2 & 2c_3 \end{pmatrix} V_3 \right\},$$

$$\mathbf{N}^e = \frac{f\Delta^e}{12}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix},$$

$$\mathbf{D}^e = \frac{\nu}{4\Delta^e}\begin{pmatrix} b_1^2 + c_1^2 & b_1b_2 + c_1c_2 & b_1b_3 + c_1c_3 \\ b_2b_1 + c_2c_1 & b_2^2 + c_2^2 & b_2b_3 + c_2c_3 \\ b_3b_1 + c_3c_1 & b_3b_2 + c_3c_2 & b_3^2 + c_3^2 \end{pmatrix},$$

$$\mathbf{B}^e = \frac{\tau^e\Delta^e}{12}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix},$$

(with

$$\tau^e = \frac{n^2 g}{3}\sum_{i=1}^{3}\frac{\sqrt{U_i^2 + V_i^2}}{H_i^{4/3}}$$

for the quadratic friction case, τ^e a constant for the linear friction case and τ^e zero for the frictionless case),

$$\mathbf{P}^e = \frac{1}{6}\begin{pmatrix} b_1 & b_2 & b_3 \\ b_1 & b_2 & b_3 \\ b_1 & b_2 & b_3 \end{pmatrix},$$

and

$$\mathbf{Q}^e = \frac{1}{6}\begin{pmatrix} c_1 & c_2 & c_3 \\ b_1 & b_2 & b_3 \\ b_1 & b_2 & b_3 \end{pmatrix},$$

5.2.3 Time-stepping

The finite element equations (5.13), (5.19) and (5.20) are solved using an explicit modified Euler time stepping scheme. For the first half time step, from time $n\delta t$ to time $(n + \frac{1}{2}\delta t)$, the equations are

$$\bar{\mathbf{M}}\mathbf{z}^{n+\frac{1}{2}} = \widetilde{\mathbf{M}}\mathbf{z}^n + \frac{\delta t}{2}\mathbf{L}^n\mathbf{H}^n, \tag{5.27}$$

$$\bar{\mathbf{M}}\mathbf{U}^{n+\frac{1}{2}} = \widetilde{\mathbf{M}}\mathbf{U}^n + \frac{\delta t}{2}\left(-\mathbf{A}^n\mathbf{U}^n + \mathbf{N}\mathbf{V}^n + \mathbf{D}\mathbf{U}^n - \mathbf{B}\mathbf{U}^n - g\mathbf{P}\mathbf{z}^n\right), \tag{5.28}$$

$$\bar{\mathbf{M}}\mathbf{V}^{n+\frac{1}{2}} = \widetilde{\mathbf{M}}\mathbf{V}^n + \frac{\delta t}{2}\left(-\mathbf{A}^n\mathbf{V}^n - \mathbf{N}\mathbf{U}^n + \mathbf{D}\mathbf{V}^n - \mathbf{B}\mathbf{V}^n - g\mathbf{Q}\mathbf{z}^n\right), \tag{5.29}$$

where $\bar{\mathbf{M}}$ is the lumped mass matrix, with the element lumped mass matrix defined by

$$\bar{\mathbf{M}}^e = \frac{\Delta^e}{12}\begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix},$$

and $\widetilde{\mathbf{M}}$ is the selective lumped mass matrix, with the element selective lumped mass matrix,$\widetilde{\mathbf{M}}^e$,defined by

$$\widetilde{\mathbf{M}}^e = s_r\bar{\mathbf{M}}^e + (1 - s_r)\mathbf{M}^e$$

where s_r is the selective lumping parameter.

The element selective lumped mass matrix is

$$\widetilde{\mathbf{M}}^e = \frac{\Delta^e}{12}\begin{pmatrix} 2+2s_r & 1-s_r & 1-s_r \\ 1-s_r & 2+2s_r & 1-s_r \\ 1-s_r & 1-s_r & 2+2s_r \end{pmatrix},$$

For the second half time step, the equations are

$$\bar{\mathbf{M}}\mathbf{z}^{n+1} = \widetilde{\mathbf{M}}\mathbf{z}^n + \delta t\mathbf{L}^{n+\frac{1}{2}}\mathbf{H}^{n+\frac{1}{2}}, \tag{5.30}$$

$$\bar{\mathbf{M}}\mathbf{U}^{n+1} = \widetilde{\mathbf{M}}\mathbf{U}^n + \delta t(-\mathbf{A}^{n+\frac{1}{2}}\mathbf{U}^{n+\frac{1}{2}} + \mathbf{N}\mathbf{V}^{n+\frac{1}{2}} + \mathbf{D}\mathbf{U}^{n+\frac{1}{2}} - \mathbf{B}\mathbf{U}^{n+\frac{1}{2}} - g\mathbf{P}\mathbf{z}^{n+\frac{1}{2}}), \tag{5.31}$$

$$\bar{\mathbf{M}}\mathbf{V}^{n+1} = \widetilde{\mathbf{M}}\mathbf{V}^n + \delta t(-\mathbf{A}^{n+\frac{1}{2}}\mathbf{V}^{n+\frac{1}{2}} - \mathbf{N}\mathbf{U}^{n+\frac{1}{2}} + \mathbf{D}\mathbf{V}^{n+\frac{1}{2}} - \mathbf{B}\mathbf{V}^{n+\frac{1}{2}} - g\mathbf{Q}\mathbf{z}^{n+\frac{1}{2}}), \tag{5.32}$$

The scheme consisting of the equations (5.27), (5.28) and (5.29) for the first stage and equations (5.30), (5.31) and (5.32) for the second stage is called the selective lumped mass (SLM) scheme. It is relatively simple to apply but is conditionally stable. For the one dimensional linearised scheme, the scheme is stable provided that the time step satisfies

$$\delta t \leq d_m \delta x/\sqrt{gh}, \tag{5.33}$$

where δx is the smallest space step, and d_m is a function of s_r, obtained from a table by Goraya [29]. The stability condition has been established only for the linearised one dimensional shallow water equations for h a constant. For h variable the stability criterion is applied using the maximum value of h. The resulting value is only a guide for two dimensional models with δt given by

$$\delta t \leq d_m \delta x/\sqrt{gh}, \tag{5.34}$$

with d_m defined as for one dimensional scheme, δx is the smallest side of any triangular element used, and h taken to be the maximum water depth.

With the SLM method applied to the two dimensional shallow water equations to model tides initially the sea is assumed to be calm in Ω, the horizontal domain of flow, i.e. $U = 0$ and $V = 0$ at time $t = 0$ and at $t = 0$ $\zeta = 0$ except at the open sea boundary, where the water level, ζ is specified as a function of time, t.

5.3 Convergence study

5.3.1 Introduction

Goraya [29] did a convergence study of the SLM method. He modelled linear flow in a rectangular basin of constant depth with forcing at one end and a closed wall at the other end. Three different meshes were used in the study. The numerical solutions were compared with analytical solutions by Lynch and Gray [67]. It was found that the finer the mesh the closer the results were to analytical solutions. Node-to-node oscillations were not included in Goraya's study.

This thesis discusses a numerical convergence study for a particular rectangular basin of linearly varying depth in which the results for the selective lumped matrix method for various numbers of triangular finite elements were compared to the analytic solution. The analytical solution is for a one dimensional representation of the basin. It was found that the larger the number of finite elements used the closer were the computed amplitudes and phases of the tidal height and velocity at various nodes to the analytical values. It was also found that the larger the number of finite elements used the smaller the node-to-node oscillations.

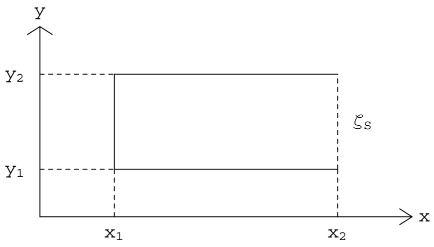

Figure 5.1: Rectangular harbour. Plan view.

5.3.2 The analytical solution used for comparison

The analytical solution used is for one dimensional frictionless linear flow in a rectangular harbour as illustrated in Figure 5.1. The depth varies according to

$$h = \frac{x - x_2}{x_1 - x_2} h_1 + \frac{x - x_1}{x_2 - x_1} h_2, \tag{5.35}$$

as shown in Figure 5.2. The solution was obtained by Lynch and Gray [67]. The flow in the harbour is modelled by linearised shallow water equations. These equations are obtained from the shallow water equations (2.1), (2.2) and (2.3) by leaving out the nonlinear terms in equations (2.1) and (2.3), setting the bottom friction to zero and eliminating (2.2). The equations are

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (hU)}{\partial x} = 0, \tag{5.36}$$

$$\frac{\partial U}{\partial t} + g \frac{\partial \zeta}{\partial x} = 0, \tag{5.37}$$

The land boundary is at $x = x_1$ and the sea boundary at $x = x_2$. On the land boundary the current velocity is set to zero, i.e.

$$U(x_1, t) = 0, \tag{5.38}$$

while on the sea boundary equation the water level, ζ_S, is specified by

$$\zeta_S \equiv \zeta(x_2, t) = A \cos(\frac{2\pi t}{T}), \tag{5.39}$$

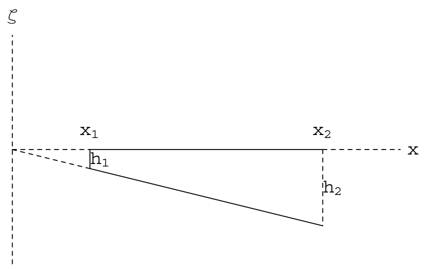

Figure 5.2: Vertical section. Plan view.

where A is the amplitude of the tidal wave component and T is its period.

The analytical solution for $\zeta(x,t)$ is assumed to be of the form

$$\zeta(x,t) = z(x)e^{j\omega t}, \tag{5.40}$$

where

$$\omega = \frac{2\pi}{T}, \tag{5.41}$$

and

$$j = \sqrt{-1}, \tag{5.42}$$

and the analytical solution for U is assumed to be of the form

$$U(x,t) = u(x)e^{j\omega t}. \tag{5.43}$$

Substitution of (5.35), (5.40) and (5.43) into (5.36) and (5.37) gives

$$z(x) = c_1 J_0\left(\frac{2\omega\sqrt{x}}{\sqrt{gH_0}}\right) + c_2 Y_0\left(\frac{2\omega\sqrt{x}}{\sqrt{gH_0}}\right), \tag{5.44}$$

and

$$u(x) = \frac{g}{j\omega}\left(c_1 J_1\left(\frac{2\omega\sqrt{x}}{\sqrt{gH_0}}\right)\frac{\omega}{\sqrt{gH_0 x}} + c_2 Y_1\left(\frac{2\omega\sqrt{x}}{\sqrt{gH_0}}\right)\frac{\omega}{\sqrt{gH_0 x}}\right), \tag{5.45}$$

where

$$H_0 = \frac{h_2}{x_2}, \tag{5.46}$$

and c_1 and c_2 are constants.

Expressions for c_1 and c_2 are obtained by using the boundary conditions (5.38) and (5.39), giving

$$c_2 = \frac{AJ_1\left(\frac{2\omega\sqrt{x_1}}{\sqrt{gH_0}}\right)}{Y_0\left(\frac{2\omega\sqrt{x_2}}{\sqrt{gH_0}}\right)J_1\left(\frac{2\omega\sqrt{x_1}}{\sqrt{gH_0}}\right) - Y_1\left(\frac{2\omega\sqrt{x_1}}{\sqrt{gH_0}}\right)J_0\left(\frac{2\omega\sqrt{x_2}}{\sqrt{gH_0}}\right)}, \tag{5.47}$$

and

$$c_1 = -\frac{c_2 Y_1\left(\frac{2\omega\sqrt{x_1}}{\sqrt{gH_0}}\right)}{J_1\left(\frac{2\omega\sqrt{x_1}}{\sqrt{gH_0}}\right)}. \tag{5.48}$$

5.3.3 Numerical experiments: the SLM solution versus the analytical solution

A computer program using Visual C++, which is listed in Appendix A, has been written to solve the SLM equations corresponding to the linearised shallow water equations (5.36) and (5.37) for finite element meshes of varying size.

The SLM equations for the first half time step are

$$\bar{\mathbf{M}}\mathbf{z}^{n+\frac{1}{2}} = \widetilde{\mathbf{M}}\mathbf{z}^n + \frac{\delta t}{2}\mathbf{L}^n\mathbf{h}^n, \tag{5.49}$$

$$\bar{\mathbf{M}}\mathbf{U}^{n+\frac{1}{2}} = \widetilde{\mathbf{M}}\mathbf{U}^n - \frac{g\delta t}{2}\mathbf{P}\mathbf{z}^n, \tag{5.50}$$

$$\bar{\mathbf{M}}\mathbf{V}^{n+\frac{1}{2}} = \widetilde{\mathbf{M}}\mathbf{V}^n - \frac{g\delta t}{2}\mathbf{Q}\mathbf{z}^n. \tag{5.51}$$

The SLM equations for the second half time step are

$$\bar{\mathbf{M}}\mathbf{z}^{n+1} = \widetilde{\mathbf{M}}\mathbf{z}^n + \delta t\mathbf{L}^{n+\frac{1}{2}}\mathbf{h}^{n+\frac{1}{2}}, \tag{5.52}$$

$$\bar{\mathbf{M}}\mathbf{U}^{n+1} = \widetilde{\mathbf{M}}\mathbf{U}^n - g\delta t\mathbf{P}\mathbf{z}^{n+\frac{1}{2}}, \tag{5.53}$$

$$\bar{\mathbf{M}}\mathbf{V}^{n+1} = \widetilde{\mathbf{M}}\mathbf{V}^{n} - g\delta t\mathbf{Q}\mathbf{z}^{n+\frac{1}{2}}. \tag{5.54}$$

The computer program has been used to simulate the flow for a rectangular basin of width 5000 metres (in the y-direction) and length 10,000 metres (in the $x-$direction) and varying linearly in depth from 2 m at the left end to 10 m at the right end (the open end) with the amplitude of the forcing function $A = 0.5$ m, and the period of the forcing function $T = 1$ hour. The program has been run for five different meshes with s_r set to 0.98 and the time step, δt, set to 30 seconds for the two coarsest meshes and the time step set to 8 seconds for the other three meshes. The results of the program were compared with the 1D analytical solutions. The basin as shown in Figures 5.1 and 5.2, has the following dimensions $x_1 = 2500$ m, $x_2 = 12500$ m, $h_1 = 2$ m and $h_2 = 10$ m.

Five different finite element meshes have been used in order to investigate the convergence of the method of solution. The first mesh has 6 nodes and 4 triangular elements, each triangle being a right-angled isosceles triangle with the equal sides being 5000 metres in length. The mesh is called mesh 1 and is shown in Figure 5.3. The second mesh, called mesh 2, has 15 nodes and 16 triangular elements, each triangle being a right-angled isosceles triangle with the equal sides being 2500 metres in length. Mesh 2 is shown in Figure 5.4. The third mesh, called mesh 3, has 45 nodes and 64 triangular elements, each triangle being a right-angled isosceles triangle with the equal sides being 1250 metres in length. Mesh 3 is shown in Figure 5.5. The fourth mesh, called mesh 4, has 153 nodes and 256 triangular elements, each triangle being a right-angled isosceles triangle with the equal sides being 625 metres in length. Mesh 4 is shown in Figure 5.6. The fifth mesh, called mesh 5, has 561 nodes and 1024 triangular elements, each triangle being a right-angled isosceles triangle with the equal sides being 312.5 metres in length. Mesh 5

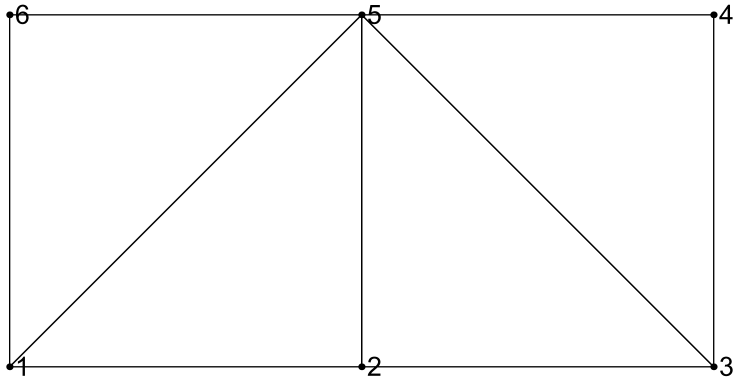

Figure 5.3: Mesh 1

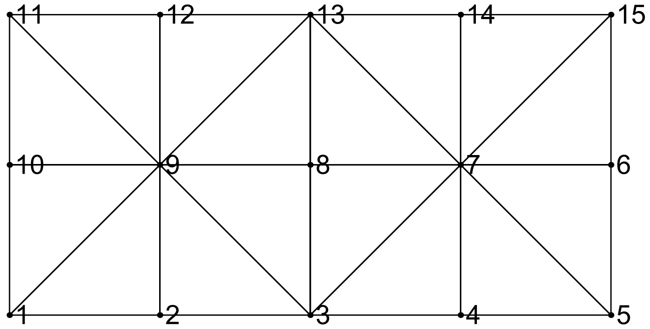

Figure 5.4: Mesh 2

is shown in Figure 5.7. The second mesh has four times as many elements as the first mesh, the third mesh has four times as many elements as the second mesh, etc.

For each mesh the values of U, V and ζ at all the nodes were computed at every time interval for a computed time of several days. The computed period was chosen so that the initial transients have disappeared for a computed period of two days. It was known that the initial transients have disappeared from an analysis of the data using a FFT (Fast Fourier Transform) computer program.

A FFT computer program was used to analyse results at selected nodes.

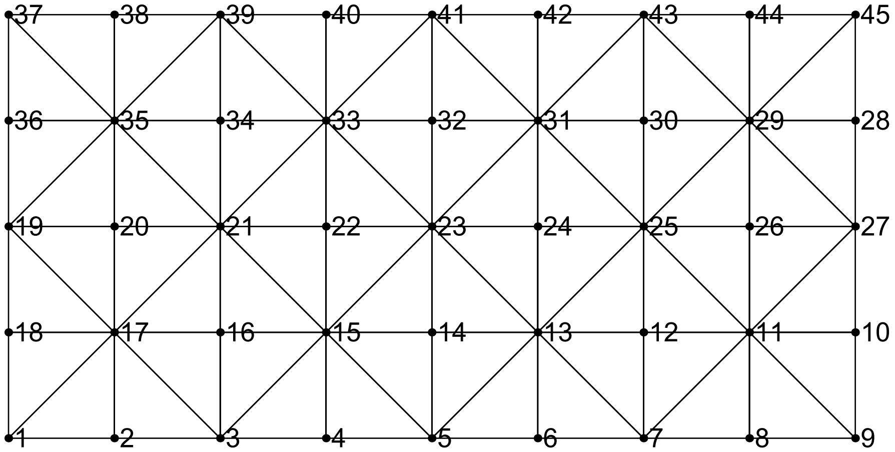

Figure 5.5: Mesh 3

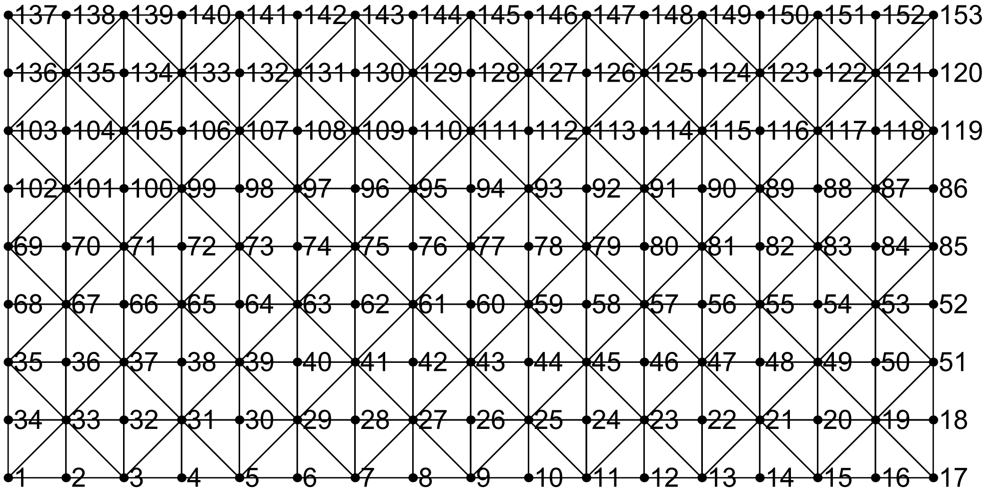

Figure 5.6: Mesh 4

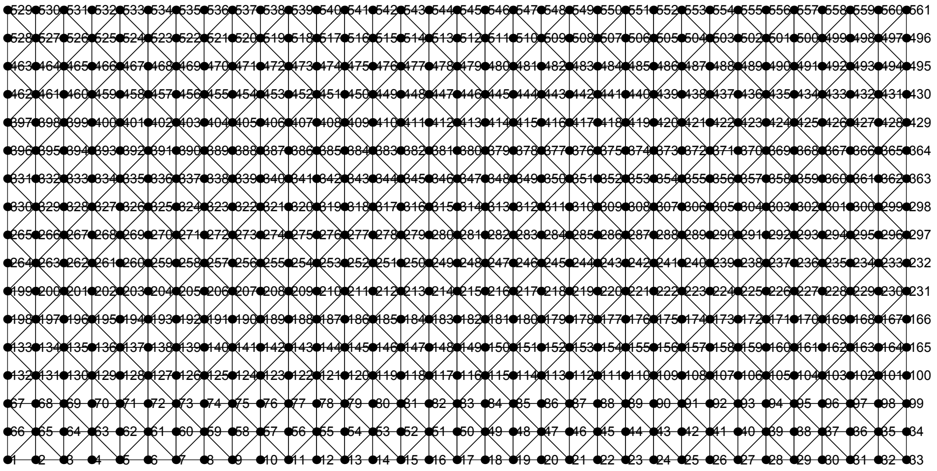

Figure 5.7: Mesh 5

For meshes 1 to 4 these were the nodes along the left boundary, the right boundary and the line parallel to the left boundary and right boundary and midway between them. For mesh 5 the nodes were along the along the left boundary, the right boundary and the line parallel to the left boundary and right boundary and midway between them only up to the line parallel to and midway between the bottom boundary and the top boundary. The FFT was used to find the amplitude and phase for the sinusoidal functions of time of ζ, U and V that remained at these nodes after the transients have died. The FFT was applied at a selection of nodes, it being assumed that the results at a substantial proportion of the nodes would give an indication of the results for all the nodes. The FFT was applied to the data over 2 days. For meshes 1, 2 and 3 the FFT was applied to the data for a period from 4 days to 6 days. For meshes 4 and 5 the FFT was applied to the data for different periods ranging from 2 days to 4 days up to 70 days to 72 days, a typical value for the ζ and U values in mesh 4 being 18 to 20 days and for the V values being 6 to 8 days, and a typical value for the ζ and U values in mesh

5 being 70 to 72 days and for the V values being 2 to 4 days. In some cases the FFT was applied to the data at a given node in a given mesh for more than one period - for example the U values at node 1 in mesh 5 had the FFT applied to them from 30 days to 32 days, from 32 to 34 days, and from 34 to 36 days - and in all cases where this had been done the results for the amplitude and phase were either very close or the same. The results of the FFT analyses are shown in Tables 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14 and 5.15. The analytical phase for ζ at the right boundary is $0°$. The U velocity is out of phase from ζ at the right boundary by $-90°$. All phases of ζ, U and V in the tables are given with reference to the analytical phase for ζ at the right boundary.

The FFT results for the amplitude and phase of ζ, U and V at each selected node were compared with the 1D SWES analytical solutions at the given node. Tables 5.1 to 5.15 show these results, with ζ being in metres and U and V in metres per second. The time steps used in calculating ζ, U and V all satisfied the stability criterion used in equation (5.34). For the two coarsest meshes, time steps of 30 seconds were used. For the finest three meshes time steps of 8 seconds were used.

A comparison of the errors in water level, ζ, U-velocity, and V-velocity at selected points in the rectangular bay is given in Tables 5.17 to 5.21, with the error of any quantity defined as its computed value minus its analytical value. The selected points (labelled O, P, Q, R, S, T, U, V and W) are shown in Figure 5.8. The amplitudes and phase lags at these points for the analytical shallow water model are shown in Table 5.16.

The graph of the analytical solution for ζ when the water is at its maximum height at the right hand boundary is shown in Figure 5.9. The graph of the analytical solution for U when U has its maximum value at the right hand boundary is shown in Figure 5.10.

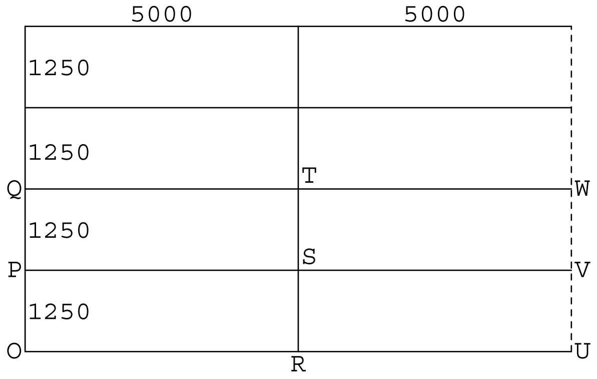

Figure 5.8: Selected points in the rectangular harbour. Dimensions are in metres.

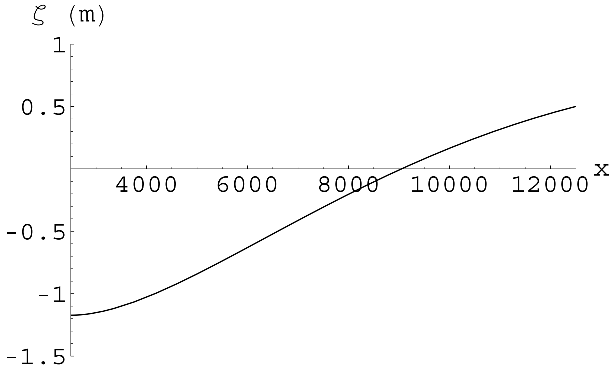

Figure 5.9: The analytical solution for ζ at the time when ζ is a maximum at the right boundary for a basin with dimensions $x_1 = 2500$ m, $x_2 = 12500$ m, $h_1 = 2$ m, $h_2 = 10$ m, $A = 0.5$ m and $t = 3600$ s. Dimensions of the x axis are in metres.

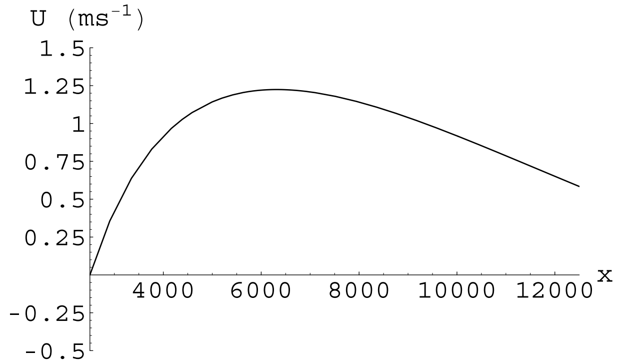

Figure 5.10: The analytical solution for U at the time when U is a maximum at the right boundary for a basin with dimensions $x_1 = 2500$ m, $x_2 = 12500$ m, $h_1 = 2$ m, $h_2 = 10$ m, $A = 0.5$ m, and $t = 3600$ s. Dimensions of the x axis are in metres.

From Tables 5.1, 5.2 and 5.3 it can be seen that at nodes 1, 6, 2 and 5 in Mesh 1, the result for the computed SLM amplitude for the water level ζ is of the right general shape, though significantly different from the analytical value, with the percentage error ranging from 41% to 110%. Similarly, the phase is significantly different, with the percentage error ranging from 174% to 197%. As the analytical U -velocity amplitudes at nodes 1 and 6 are zero, a percentage error cannot be calculated at these nodes. However, it can be said that the error is substantial in absolute terms, being of the order of 0.9 ms^{-1}. At the other nodes the error ranges in value from 40% to 76%. The errors in the phase of the U-velocity range from 15% to 31%. As the analytical V-velocity amplitude at all nodes is zero, a percentage error cannot be calculated at these nodes. However, it can be said that the error in absolute terms is substantial at most nodes, ranging from 0.00 8 ms^{-1} to 0.738 ms^{-1}, the worst result being at the top left hand corner node.

From Tables 5.4, 5.5 and 5.6 it can be seen that the results for ζ in Mesh 2 are an improvement on those in Mesh 1, with the percentage error for

Table 5.1: Comparison of computed SLM and analytical values of the amplitude (m) and phase (degrees) of ζ for mesh 1.

	Amplitude (m)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	0.687	1.173	-132.50	180
6	0.984	1.173	-174.60	180
2	0.646	0.307	143.57	180
5	0.443	0.307	154.19	180

Table 5.2: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of U for mesh 1.

	Amplitude (ms^{-1})		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	0.891	0.000	-29.72	-
6	0.857	0.000	-60.29	-
2	0.645	1.180	-61.69	-90
5	0.706	1.180	-76.29	-90
3	1.029	0.584	-115.23	-90
4	0.939	0.584	-113.00	-90

Table 5.3: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of V for mesh 1.

	Amplitude (m/s)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	0.261	0.000	37.46	-
6	0.738	0.000	52.86	-
2	0.232	0.000	-120.68	-
5	0.082	0.000	18.48	-
3	0.118	0.000	-115.54	-
4	0.008	0.000	9.30	-

the amplitude ranging from 3% to 43%. The error for the phase ranges from 1% to 30%. The errors in the U-velocity amplitude along the right hand boundary and the central line parallel to this boundary are a substantial improvement on the Mesh 1 values, ranging from 4% to 8%. Along the left hand boundary the error is substantial, ranging from about 0.3 to 0.5 ms^{-1}. The error in the phase is small, ranging from 0.07% to 2%. For some nodes the V-velocity is zero to 3 decimal places. The error is still significant at nodes 1 and 11 along the left boundary, being 0.236 ms^{-1}. At the other nodes the error is small, ranging from .012 to .073 (ms^{-1}). The results for the amplitude of water level ζ and the velocities are accurate except along the left hand boundary, where the errors in the velocity values are still quite large. It can be seen that at any two nodes that are an equal distance from the line which runs parallel to the x-axis and which is midway between the top boundary and bottom boundary of the bay that the phases of the V-velocities differ by 180° , whereas the amplitudes and phases of the water level ζ and the U-velocity and the amplitude of the V-velocity are the same.

From Tables 5.7, 5.8 and 5.9 it can be seen that the results for ζ, U

Table 5.4: Comparison of computed SLM and analytical values of the amplitude (m) and phase (degrees) of ζ for mesh 2.

	Amplitude (m)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	1.206	1.173	-179.00	180
10	0.969	1.173	172.43	180
11	1.206	1.173	-179.00	180
3	0.173	0.307	126.45	180
8	0.236	0.307	99.64	180
13	0.173	0.307	126.45	180

Table 5.5: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of U for mesh 2.

	Amplitude (m/s)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	0.529	0.000	-95.94	-
10	0.290	0.000	-146.70	-
11	0.529	0.000	-95.94	-
3	1.236	1.180	-91.59	-90
8	1.272	1.180	-89.94	-90
13	1.236	1.180	-91.59	-90
5	0.561	0.584	-89.56	-90
6	0.559	0.584	-88.95	-90
15	0.561	0.584	-89.56	-90

Table 5.6: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of V for mesh 2.

	Amplitude (m/s)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	0.236	0.000	-79.71	-
10	0.000	0.000	-	-
11	0.236	0.000	100.29	-
3	0.073	0.000	-47.26	-
8	0.000	0.000	-	-
13	0.073	0.000	132.75	-
5	0.012	0.000	154.84	-
6	0.000	0.000	-	-
15	0.012	0.000	-25.17	-

and V in Mesh 3 are an improvement on those in Mesh 2. The errors in the amplitude of ζ range from 0.26% to 7%. The errors in the phase of ζ range from 1% up to 9%. The errors in the U-velocity amplitude are still significant on the left hand boundary nodes, ranging from about .14 to .27 (ms^{-1}). The errors in the phase range up to 7%. The errors in the V-velocity amplitude are significant only at nodes 1 and 37, being .123 (ms^{-1}). At the other nodes the errors are small, ranging from 0 to 0.025 (ms^{-1}). It can be seen that along a line of nodes with the same x-values, inter-nodal oscillation of amplitude and phase values occurs.

The calculations of amplitude and phase of ζ, U and V at various nodes in Meshes 1, 2 and 3 were calculated using the FFT program for a period from 4 days 8 seconds to 6 days. This calculation was repeated at some of the nodes in the meshes for a period from 6 days 8 seconds to 8 days and the same results were obtained for the amplitudes and phases.

Table 5.7: Comparison of computed SLM and analytical values of the amplitude (m) and phase (degrees) of ζ for mesh 3.

	Amplitude (m)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	1.170	1.173	178.32	180.00
18	1.113	1.173	175.97	180.00
19	1.182	1.173	179.07	180.00
36	1.113	1.173	175.97	180.00
37	1.170	1.173	178.32	180.00
5	0.289	0.307	166.37	180.00
14	0.293	0.307	164.62	180.00
23	0.287	0.307	166.54	180.00
32	0.293	0.307	164.62	180.00
41	0.289	0.307	166.37	180.00

Table 5.8: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of U for mesh 3.

	Amplitude (m/s)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	0.269	0.000	-99.96	-
18	0.141	0.000	-139.53	-
19	0.267	0.000	-100.26	-
36	0.141	0.000	-139.27	-
37	0.269	0.000	-99.96	-
5	1.180	1.180	-91.28	-90
14	1.185	1.180	-90.66	-90
23	1.184	1.180	-91.16	-90
32	1.185	1.180	-90.66	-90
41	1.180	1.180	-91.28	-90
9	0.559	0.584	-95.10	-90
10	0.546	0.584	-96.47	-90
27	0.557	0.584	-95.31	-90
28	0.546	0.584	-96.47	-90
45	0.559	0.584	-95.10	-90

Table 5.9: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of V for mesh 3.

	Amplitude (m/s)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	0.123	0.000	-99.27	-
18	0.025	0.000	96.93	-
19	0.000	0.000	-	-
36	0.025	0.000	-83.07	-
37	0.123	0.000	80.73	-
5	0.010	0.000	-32.12	-
14	0.002	0.000	178.61	-
23	0.000	0.000	-	-
32	0.002	0.000	-1.39	-
41	0.010	0.000	147.88	-
9	0.002	0.000	141.91	-
10	0.000	0.000	-	-
27	0.000	0.000	-	-
28	0.000	0.000	-	-
45	0.002	0.000	-38.09	-

From Tables 5.10, 5.11 and 5.12 it can be seen that the results for ζ, U and V in Mesh 4 are an improvement on those in Mesh 3. The errors in the amplitude of ζ range from 0.09% to 2%. The errors in the phase of ζ range from 0.3% to 2%. The errors in U-velocity amplitude are significant only at the left boundary nodes 1, 35, 69, 103 and 137, ranging from 0.083 to 0.095 ms^{-1}. Elsewhere they are small, ranging from 0% to 4%. The errors in U phase are small, ranging up to 2%. The errors in V -velocity amplitude are significant only at nodes 1 and 137 (0.055 ms^{-1}). Elsewhere the errors are small, ranging from 0 to 0.016 ms^{-1}.

From Tables 5.1 to 5.12 it can be seen that the values of ζ, U and V for the nodes in meshes 1 to 4 are symmetric about the line that is parallel to and midway between the bottom and top boundaries. For this reason, with mesh 5 the nodes used for the FFT analysis were along the along the left boundary, the right boundary and the line parallel to the left boundary and right boundary and midway between them only up to the line parallel to and midway between the bottom boundary and the top boundary.

From Tables 5.13, 5.14 and 5.15 it can be seen that the results for Mesh 5 are an improvement on those in Mesh 4, with no significant errors at any points, the most significant being 0.032 ms^{-1} for the U-velocity at node 1 (and, by the symmetry that was shown to exist in previous meshes, at node 529).

The Tables 5.17, 5.18 5.19, 5.20 and 5.21 show how the errors have gradually decreased with an increased number of elements, with the worst results being for the U-velocity along the left boundary. As there is no value for the analytical phase of the U-velocity at the left boundary and no value for the analytical phase of the V-velocity at all nodes, no error values are shown in the tables for these cases. It took a long period of time to get rid of the transients in the FFT analysis of data, ranging from 2 days (48 time

Table 5.10: Comparison of computed SLM and analytical values of the amplitude (m) and phase (degrees) of ζ for mesh 4.

	Amplitude (m)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	1.172	1.173	179.30	180
34	1.155	1.173	179.23	180
35	1.177	1.173	179.41	180
68	1.151	1.173	179.15	180
69	1.179	1.173	179.46	180
102	1.151	1.173	179.15	180
103	1.177	1.173	179.41	180
136	1.155	1.173	179.23	180
137	1.172	1.173	179.30	180
9	0.300	0.307	176.51	180
26	0.301	0.307	176.29	180
43	0.300	0.307	176.57	180
60	0.301	0.307	176.06	180
77	0.300	0.307	176.59	180
94	0.301	0.307	176.06	180
111	0.300	0.307	176.57	180
128	0.301	0.307	176.29	180
145	0.300	0.307	176.51	180

Table 5.11: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of U for mesh 4.

Node No.	Amplitude (m/s)		Phase (degrees)	
	SLM	Analytical	SLM	Analytical
1	0.095	0.000	-101.46	-
34	0.031	0.000	-149.92	-
35	0.086	0.000	-109.25	-
68	0.017	0.000	-154.29	-
69	0.083	0.000	-110.14	-
102	0.017	0.000	-154.29	-
103	0.086	0.000	-109.25	-
136	0.031	0.000	-149.92	-
137	0.095	0.000	-101.46	-
9	1.180	1.180	-90.32	-90
26	1.181	1.180	-90.17	-90
43	1.180	1.180	-90.31	-90
60	1.181	1.180	-90.15	-90
77	1.180	1.180	-90.30	-90
94	1.181	1.180	-90.15	-90
111	1.180	1.180	-90.31	-90
128	1.181	1.180	-90.17	-90
145	1.180	1.180	-90.32	-90
17	0.577	0.584	-91.04	-90
18	0.570	0.584	-91.71	-90
51	0.576	0.584	-91.24	-90
52	0.572	0.584	-91.55	-90
85	0.576	0.584	-91.25	-90
86	0.572	0.584	-91.55	-90
119	0.576	0.584	-91.24	-90
120	0.570	0.584	-91.71	-90
153	0.577	0.584	-91.04	-90

Table 5.12: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of V for mesh 4.

	Amplitude (m/s)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	0.055	0.000	-124.55	-
34	0.016	0.000	72.46	-
35	0.012	0.000	-106.77	-
68	0.006	0.000	85.82	-
69	0.000	0.000	-	-
102	0.006	0.000	-94.18	-
103	0.012	0.000	83.23	-
136	0.016	0.000	-107.54	-
137	0.055	0.000	-65.45	-
9	0.003	0.000	-34.35	-
43	0.002	0.000	-25.90	-
77	0.000	0.000	-	-
17	0.001	0.000	139.42	-
51	0.000	0.000	-	-
85	0.000	0.000	-	-

Table 5.13: Comparison of computed SLM and analytical values of the amplitude (m) and phase (degrees) of ζ for mesh 5.

	Amplitude (m)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	1.172	1.173	179.80	180.00
66	1.169	1.173	179.84	180.00
67	1.173	1.173	179.80	180.00
132	1.168	1.173	179.83	180.00
133	1.174	1.173	179.82	180.00
198	1.167	1.173	179.82	180.00
199	1.174	1.173	179.83	180.00
264	1.167	1.173	179.11	180.00
265	1.174	1.173	179.09	180.00
17	0.305	0.307	179.11	180.00
50	0.305	0.307	179.09	180.00
83	0.305	0.307	179.12	180.00
116	0.305	0.307	179.04	180.00
149	0.305	0.307	179.13	180.00
182	0.305	0.307	179.02	180.00
215	0.305	0.307	179.13	180.00
248	0.305	0.307	179.00	180.00
281	0.305	0.307	179.14	180.00

Table 5.14: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of U for mesh 5.

Node No.	Amplitude (m/s)		Phase (degrees)	
	SLM	Analytical	SLM	Analytical
11	0.032	0.000	-97.54	-
66	0.009	0.000	-84.60	-
67	0.024	0.000	-111.85	-
132	0.003	0.000	-146.73	-
133	0.022	0.000	-114.13	-
198	0.003	0.000	-134.24	-
199	0.022	0.000	-115.45	-
264	0.003	0.000	-131.97	-
265	0.022	0.000	-115.77	-
17	1.180	1.180	-90.08	-90.00
50	1.180	1.180	-90.04	-90.00
83	1.180	1.180	-90.08	-90.00
116	1.180	1.180	-90.04	-90.00
149	1.180	1.180	-90.08	-90.00
182	1.180	1.180	-90.03	-90.00
215	1.180	1.180	-90.07	-90.00
248	1.180	1.180	-90.03	-90.00
281	1.180	1.180	-90.07	-90.00
33	0.583	0.584	-90.16	-90.00
34	0.580	0.584	-90.50	-90.00
99	0.582	0.584	-90.28	-90.00
100	0.580	0.584	-90.41	-90.00
165	0.582	0.584	-90.30	-90.00
166	0.581	0.584	-90.38	-90.00
231	0.582	0.584	-90.30	-90.00
232	0.581	0.584	-90.36	-90.00
297	0.582	0.584	-90.30	-90.00

Table 5.15: Comparison of computed SLM and analytical values of the amplitude (ms^{-1}) and phase (degrees) of V for mesh 5.

	Amplitude (m/s)		Phase (degrees)	
Node No.	SLM	Analytical	SLM	Analytical
1	0.020	0.000	-139.44	-
66	0.006	0.000	61.54	-
67	0.006	0.000	-115.53	-
132	0.005	0.000	79.31	-
133	0.003	0.000	-104.53	-
198	0.003	0.000	90.75	-
199	0.002	0.000	-100.63	-
264	0.000	0.000	-	-
265	0.000	0.000	-	-
17	0.000	0.000	-	-
149	0.000	0.000	-	-
281	0.000	0.000	-	-
33	0.000	0.000	-	-
165	0.000	0.000	-	-
297	0.000	0.000	-	-

Table 5.16: Amplitudes and phase lags for the analytical shallow water model

	Water level (ζ) (m)		U-velocity (ms^{-1})		V-velocity(ms^{-1})	
Node No.	Amp (m)	Phase (°)	Amp (m/s)	Phase (°)	Amp (m/s)	Phase (°)
O	1.173	180.00	0.000	-	0.000	-
P	1.173	180.00	0.000	-	0.000	-
Q	1.173	180.00	0.000	-	0.000	-
R	0.307	180.00	1.180	-90.00	0.000	-
S	0.307	180.00	1.180	-90.00	0.000	-
T	0.307	180.00	1.180	-90.00	0.000	-
U	0.500	0.00	0.584	-90.00	0.000	-
V	0.500	0.00	0.584	-90.00	0.000	-
W	0.500	0.00	0.584	-90.00	0.000	-

periods) to 70 days (1680 time periods).

In addition, the FFT results for the amplitude and phase of ζ at nodes 1 and 9 in Mesh 4 were calculated using different values of s_r (the selective lumping parameter) with a time step of 8 seconds. The results are shown in Tables 5.22 and 5.23. At node 1 the analytical amplitude is 1.173 metres and the analytical phase is 180 degrees, whereas at node 9 the analytical amplitude is 0.307 metres and the analytical phase is 180 degrees.

The best results for the amplitude of ζ at node 1 in mesh 4 were obtained when s_r was 0.97 (to 4 decimal places the amplitude was 1.1718 for s_r equal to 0.97 and 1.1716 for s_r equal to 0.98) and the best results for phase are obtained when s_r is 0.98. The least satisfactory results are obtained for s_r equal to zero where the error is 15%.

The best results for the amplitude of ζ at node 9 in mesh 4 were obtained for s_r equal to 0.9 (to 4 decimal places the error is 0.0007 for s_r equal to 0.9 and 0.0010 for s_r equal to 0.89). The least satisfactory results were obtained

Table 5.17: Error in amplitudes (m) for the water level (ζ)

	Mesh				
Node	6 node	15 node	45 node	153 node	561 node
O	-0.486	-0.033	-0.002	-0.001	-0.001
P	n.a	n.a	-0.060	0.004	0.001
Q	n.a	-0.204	0.010	0.006	0.002
R	0.339	-0.134	-0.018	-0.006	-0.002
S	n.a	n.a	-0.014	-0.007	-0.002
T	n.a	-0.070	-0.019	-0.007	-0.002

Table 5.18: Error in the phase lags (degrees) for water level (ζ)

	Mesh				
Node	6 node	15 node	45 node	153 node	561 node
O	-37.50	1.00	-1.68	-0.70	-0.20
P	n.a.	n.a.	-4.03	-0.59	-0.19
Q	n.a.	-7.57	-0.93	-0.54	-0.17
R	-36.44	-53.55	-13.63	-3.49	-0.89
S	n.a.	n.a.	-15.38	-3.43	-0.87
T	n.a.	-80.36	-13.46	-3.41	-0.86

Table 5.19: Error in the amplitudes (ms^{-1}) for U-velocity

	Mesh				
Node	6 node	15 node	45 node	153 node	561 node
O	0.891	0.529	0.269	0.095	0.031
P	n.a.	n.a.	0.141	0.086	0.022
Q	n.a.	0.290	0.267	0.083	0.022
R	-0.535	0.056	-0.002	-0.001	0.000
S	n.a.	n.a.	0.005	0.000	0.000
T	n.a.	0.092	0.003	0.000	0.000
U	0.445	-0.023	-0.025	-0.007	-0.001
V	n.a.	n.a.	-0.038	-0.008	-0.002
W	n.a.	-0.025	-0.027	-0.009	-0.002

Table 5.20: Error in phase lags (degrees) for U-velocity

	Mesh				
Node	6 node	15 node	45 node	153 node	561 node
R	28.31	-1.59	-1.28	-0.32	-0.08
S	n.a.	n.a.	-0.66	-0.31	-0.08
T	n.a.	0.06	-1.16	-0.30	-0.07
U	-25.23	0.44	-5.10	-1.04	-0.16
V	n.a.	n.a.	-6.47	-1.24	-0.30
W	n.a.	1.05	-5.31	-1.25	-0.30

Table 5.21: Error in the amplitudes (ms^{-1}) for V-velocity

	Mesh				
Node	6 node	15 node	45 node	153 node	561 node
O	0.261	0.236	0.123	0.055	0.020
P	n.a.	n.a.	0.025	0.012	0.003
Q	n.a.	0.003	0.000	0.000	0.000
R	0.232	0.073	0.010	0.003	0.000
S	n.a.	n.a.	0.003	0.002	0.000
T	n.a.	0.000	0.000	0.000	0.000
U	0.118	0.012	0.002	0.001	0.000
V	n.a.	n.a.	0.000	0.000	0.000
W	n.a.	0.000	0.000	0.000	0.000

for s_r equal to 0.0, where the error is 45%.

5.3.4 Conclusions

It was found that when the selective lumped matrix method was used to approximate the shallow water equations for a particular region with particular boundary conditions that the larger the number of elements used to cover the domain of interest the closer the computed results for ζ, U and V and get to the analytical results. This can be seen by comparing the results from Tables 1, 2, 3, 4 and 5 and looking at Tables 7, 8 and 9. It is only with Mesh 5, with 561 nodes and 1024 elements, that the errors in ζ, U and V at all the nodes examined can said to be insignificant. The results support the conclusions of Kawahara, Hirano and Tsubota [46] that about 40 elements per half of a wavelength are required to obtain accurate results using the SLM method. The finer the grid, the larger the number of time steps before transients are eliminated and satisfactory results obtained.

Table 5.22: The values of the computed amplitude and phase of ζ at node 1 in mesh 4 for different values of s_r (time step: $\delta t = 8$ s). The analytical amplitude is 1.173 m. The analytical phase is 180 degrees.

Value of s_r	Computed SLM ζ amplitude (m)	Computed SLM ζ phase (degrees)	Period of days over which FFT was calculated
0.000	0.999	156.01	18-20
0.100	1.023	158.27	18-20
0.200	1.047	160.55	18-20
0.300	1.070	162.84	18-20
0.400	1.092	165.15	18-20
0.500	1.112	167.51	18-20
0.600	1.130	169.93	18-20
0.700	1.144	172.45	18-20
0.800	1.155	175.06	18-20
0.900	1.166	177.61	18-20
0.960	1.171	178.92	18-20
0.970	1.172	179.11	18-20
0.980	1.172	179.30	18-20
0.985	1.171	179.39	38-40
0.990	1.170	179.46	38-40
1.000	Diverges	Diverges	-

Table 5.23: The values of the computed amplitude and phase of ζ at node 9 in Mesh 4 for different values of s_r (time step: $\delta t = 8$ s). The analytical amplitude is 0.307 m. The analytical phase is 180 degrees.

Value of s_r	Computed SLM ζ amplitude (m)	Computed SLM ζ phase (degrees)	Period of days over which FFT was calculated
0.00	0.444	124.07	18-20
0.10	0.438	126.46	18-20
0.20	0.430	129.06	18-20
0.30	0.419	131.95	18-20
0.40	0.406	135.23	18-20
0.50	0.390	139.06	18-20
0.60	0.371	143.71	18-20
0.70	0.349	149.55	18-20
0.80	0.326	157.14	18-20
0.89	0.308	166.03	18-20
0.90	0.306	167.16	18-20
0.98	0.300	176.51	18-20
0.99	0.298	176.63	38-40
1.00	Diverges	Diverges	-

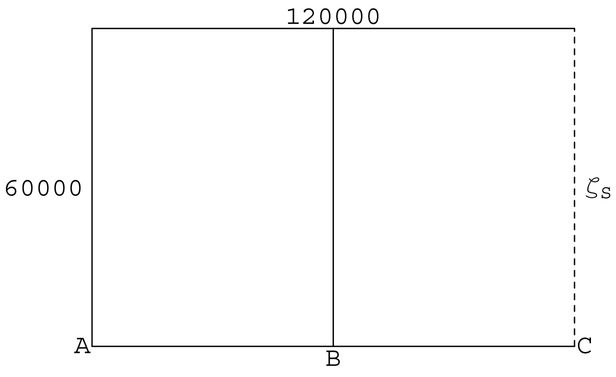

Figure 5.11: Rectangular basin in which nonlinear shallow water flow is forced by a specified water level, ζ_S, at the boundary.

The best results for the lumping mass parameter, s_r, were when s_r was in the range 0.9 to less than 1. In contrast, Kawahara, Hirano and Tsubota [46] obtained the best results for s_r between 0.85 and 0.95.

5.4 Testing the numerical model against a fixed boundary nonlinear analytical model

Testing of the analytical solutions found in Chapter 4 for nonlinear shallow water flow in a rectangular harbour against SLM solutions shows the results to be close.

The flow was in a rectangular basin of width 60,000 m and length 120000 m and of constant depth 40 m (see Figure 5.11). The flow is forced at the right hand side open sea boundary by a specified water level, ζ_S, which is a cosine function of time, with amplitude 0.5 m and period 12 hours.

The finite element mesh used had 1024 nodes and 561 triangles. The numerical parameters used were $s_r = 0.98$ and $\delta t = 50$ seconds.

A comparison of the numerical and analytical values is given in Tables 5.24 and 5.25. It can be seen that the numerical results are in close agreement

Table 5.24: Comparison of the analytical values for ζ, (amplitudes in m, phases in degrees and periods in hours), at two nodes, with the numerical values.

Node		Analytical Solution		Numerical Solution	
	Period	Amp	Phase	Amp	Phase
A	12	0.786	0.00	0.786	0.01
	6	0.026	0.00	0.026	-0.36
	Zero frequency	0.002	-	0.002	-
B	12	0.711	0.00	0.711	0.02
	6	0.019	0.00	0.019	-0.34
	Zero frequency	0.002	-	0.002	-

with the analytical results.

Table 5.25: Comparison of the analytical values for U-velocity, (amplitudes in ms^{-1} and phases, for sine waves, in degrees) at three nodes, with the numerical values.

nodes		Analytical Solution		Numerical Solution	
	Period	Amp	Phase	Amp	Phase
A	12	0.000	-	0.000	-
	6	0.000	-	0.000	-
B	12	0.166	0.00	0.166	0.07
	6	0.009	0.00	0.009	-0.31
C	12	0.300	0.00	0.300	0.02
	6	0.013	0.00	0.013	-0.29

Chapter 6

A new moving boundary numerical scheme

6.1 Introduction

This chapter presents the details of the development of a moving boundary shallow water wave equation numerical model. The model is adapted from the SLM (Selective Lumped Mass) numerical scheme of Kawahara, Hirano and Tsubota [46, 47], which was discussed in detail in Chapter 5. The SLM scheme is finite element in space, using fixed triangular elements. It is finite difference in time and is explicit. The wetting and drying algorithm in the model is different to that of Kawahara, Hirano and Tsubota.

The two-dimensional depth-averaged shallow water wave equations that Kawahara, Hirano and Tsubota solve are equations (2.1) and (2.2), modified to exclude the wind stress terms, plus equation (2.3). The resultant equations are the conservation of momentum equation in the East direction

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} - fV - \nu\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) + \tau U + g\frac{\partial \zeta}{\partial x} = 0, \qquad (6.1)$$

the conservation of momentum equation in the North direction

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + fU - \nu\left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2}\right) + \tau V + g\frac{\partial \zeta}{\partial y} = 0, \qquad (6.2)$$

and the continuity (conservation of mass) equation

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (h+\zeta)U}{\partial x} + \frac{\partial (h+\zeta)V}{\partial y} = 0. \qquad (6.3)$$

In the wetting and drying scheme of Kawahara, Hirano and Tsubota at each node i of a finite element that is in the domain of computation the water elevation, ζ, and velocity components, U and V, are calculated at the end of each half time step. There are three possible outcomes:
(i) for each node in an element the total depth, $H_i > 0$;
(ii) at least one value of H_i is $H_i > 0$ and the rest of H_i have $H_i \leq 0$; and
(iii) all values of H_i are $H_i \leq 0$.
In case (i), the element is taken to be under water. In case (ii), at any nodal point at which $H_i > 0$, the water elevation, ζ_i, and the current velocity are computed and at nodal points at which $H_i \leq 0$ the water elevation is computed but the current velocity is treated as zero. In case (iii) the element is on the exposed sea bed and is omitted from the computation.

In the scheme of Kawahara, Hirano and Tsubota a node which has a negative total water depth can sometimes be retained in the calculation. As a negative total water depth is physically unrealistic, in this thesis if a node is found to have a total water depth $H_i \leq 0$ then the node is regarded as dry.

Kawahara, Hirano and Tsubota do not discuss certain aspects of their calculations:
(a) whether all the nodes at $t = 0$ are subject to a cold start (e.g. if a node whose depth is above sea level has an initial zero water level this implies that the water is initially underground);
(b) how they decide when a previously dry element becomes wet plus what height the water becomes at a previously dry node that has become wetted;

(c) how they deal with the problem that the quadratic friction (which appears in the momentum equations), which is inversely proportional to H_i , becomes infinite at the shoreline, where H_i equals zero. Section 6.2 discusses how these aspects are dealt with in this thesis.

6.2 The moving boundary numerical model

In the moving boundary model, a critical decision on whether a node is dry or wet must be made at the end of each time step δt. Some nodes will be initially made wet while others will be initially made dry. Some nodes will change from dry to wet or wet to dry at the end of a time step, while some nodes will remain wet or remain dry. There will be wet and dry elements, with an element being wet if all the nodes are wet and dry if at least one node is dry. At any time step or half time step the SLM calculations will be made only for elements that were wet at the end of the last time step. If the total water depth, H, is calculated as less than or equal to 0 at the end of a time step it is reset to 0 and the node is regarded as inactive (i.e. dry) and the velocity set to zero and the triangle that it is in is removed from calculations. As H equals ζ plus h ζ is set to $-h$ at a node which changes from wet to dry. If the total water depth at a previously wet node is calculated as less than or equal to zero at the end of a half time step ζ is set to $-h$ and the velocity is set to zero, but the node is still deemed wet.

As was said in the previous section, Kawahara, Hirano and Tsubota do not discuss certain aspects of their calculations. Aspect (a) is dealt with in this thesis as follows. If the initial values of ζ_i , u_i and v_i are not known then nodes not on the open sea boundary and with positive or zero values of h_i at time t are subject to a cold start. Nodes that are above mean sea level (i.e. with $h_i < 0$) are not given a cold start, because as $H_i = h_i + \zeta_i$, a cold start for nodes above mean sea level would imply that $H_i < 0$, which is unrealistic. Nodes that are at or above mean sea level are regarded as dry at time $t = 0$; other nodes are regarded as wet. On the other hand, if the initial values of ζ_i , U_i and V_i are known they will be used in the numerical model, with a node initially wet only if H_i is positive.

Aspect (b) is dealt with in this thesis as follows. At the end of each time step each element that contains only one inactive (dry) node will be tested to determine whether conditions are favourable for wetting that node. A formula for calculating ζ at the previously dry node at time $t + \delta t$ was developed by the author of this thesis. The formula is based on the approximation $\delta\zeta \approx \frac{\partial\zeta}{\partial x}\delta x + \frac{\partial\zeta}{\partial y}\delta y$ and the momentum equations (with the assumption that the advective, Coriolis and eddy viscosity terms are negligible). The formula is

$$\begin{aligned}\zeta_i^{t+\delta t} = \zeta_{i-1}^{t+\delta t} &- \frac{\delta x}{g}\left(\frac{U_{i-1}^{t+\delta t} - U_{i-1}^{t}}{\delta t} + \tau_{i-1}U_{i-1}^{t+\delta t}\right) \\ &- \frac{\delta y}{g}\left(\frac{V_{i-1}^{t+\delta t} - V_{i-1}^{t}}{\delta t} + \tau_{i-1}V_{i-1}^{t+\delta t}\right) ,\end{aligned} \tag{6.4}$$

where ζ_i is the water elevation of the previously dry node and ζ_{i-1} is the larger water elevation of the two previously wet nodes in the element. If ζ_i is found to be greater than $-h_i$ (i.e. the total depth, H_i , is positive) then the node wets; otherwise it stays dry. If the node wets then the velocity at node i will be set equal to that at node $i - 1$.

Aspect (c) is dealt with in this thesis as follows (following Bills and Noye [8]). In calculating quadratic friction, H, the total depth in the quadratic friction term, is replaced by another term, H_f. If H is less than some minimum value, H_{min} (e.g. 1.0 m), H_f equals H_{min}; otherwise H_f equals H.

6.3 Testing the numerical model

The moving boundary numerical model, which was implemented using a program written in Visual C++ by the author of this thesis, which is listed in Appendix A, was tested against two moving boundary analytical solutions presented earlier in this thesis, in Chapter 3.

6.3.1 The first analytical solution used for comparison

The moving boundary numerical model was first tested against an analytical solution for shallow water frictionless flow with no eddy viscosity and with no Coriolis force and with cosine forcing in a bed with quadratically varying depth. This solution was derived in subsection 3.7.2. The bed has its depth, h, below the x-axis, given by

$$h = h_0\left(1 - \frac{x^2}{a^2}\right),\ x \geq 0. \tag{6.5}$$

The forcing occurs at $x = 0$ and is given by

$$\zeta(0,t) = \frac{-B^2\cos 2\psi t}{4g}, \tag{6.6}$$

where

$$\psi = \frac{\sqrt{2gh_0}}{a} \tag{6.7}$$

and B is a constant. The analytical solution for the water level, ζ, is

$$\zeta(x,t) = \frac{-B^2\cos 2\psi t}{4g} - \frac{(\psi B\cos\psi t)x}{g} \tag{6.8}$$

and for the velocity U is

$$U = B\sin\psi t. \tag{6.9}$$

The x-coordinate of the shoreline is given by

$$x = a\sqrt{1 + \frac{B^2}{4gh_0}} - \frac{Ba}{\sqrt{2gh_0}}\cos\frac{\sqrt{2gh_0}t}{a}. \tag{6.10}$$

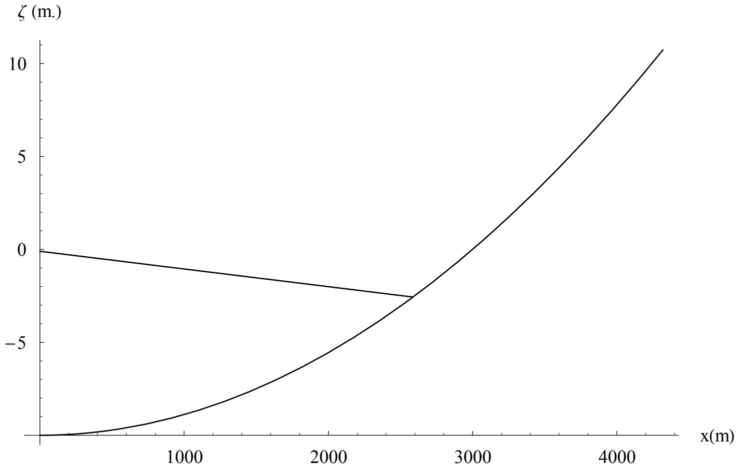

Figure 6.1: A vertical cross-section of the initial position of the water for frictionless flow above a bed with quadratically varying depth for $h_0 = 10$ m, $a = 3000$ m and $B = 2$ ms^{-1}.

6.3.2 The first analytical solution versus the numerical solution

For the numerical model the values chosen were $h_0 = 10$ m, $a = 3000$ m and $B = 2$ ms^{-1} with the initial values of ζ and U set to those of the analytical model. The initial position of the water surface above the bed is shown in Figure 6.1. The initial water velocity was set to the analytical initial water velocity (0 m/s). At the open water boundary, at $x = 0$, the water level was specified as the same function of time as in the analytical model. The calculation was done over one period (1345.71 seconds).

In the numerical model three different triangular meshes were used, each one covering a rectangular region of length 4320 m in the x direction but with different widths, i.e. in the y direction. The width of the rectangular region used in the first and coarsest mesh, Mesh 1, was 2160 metres. The width of the second mesh, Mesh 2, which was finer than Mesh 1, was 720 metres. The width of the finest mesh, Mesh 3, was 240 metres. The coordinates,

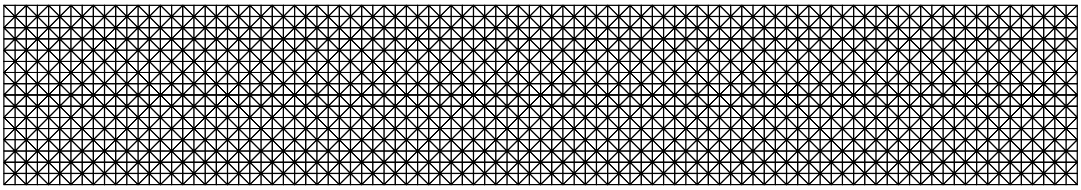

Figure 6.2: The second triangular mesh, Mesh 2, used in the numerical model of the flow.

node numbers and triangle numbers in each mesh were generated using a program written in Visual C++ by the author of this thesis that is listed in Appendix B. The meshes were plotted using a Mathematica package. Each triangle in each mesh is an isosceles right angled triangle. Mesh 1 has 561 nodes and 1024 elements. Mesh 2, shown in Figure 6.2, has 1649 nodes and 3072 elements. Mesh 3 contains 4913 nodes and 9216 elements.

As the velocity in the analytical solution is a function of t only, the advective terms are zero; for this reason with with each mesh the model was initially run without the advective terms, for one period, T, 1345.71 seconds. It was found that the finer the mesh overall the more accurate the results at a number of nodes of varying x-values at different times for water height, ζ, and U-velocity. An indication of the convergence is given by comparing the numerical values of the amplitudes and phases of ζ, U and V (obtained using a Fast Fourier Transform package) at a number of nodes of different x-values, the nodes all lying along a line that runs parallel to the x-axis and is midway between the base of the rectangle and the top of the rectangle. In general, the finer the mesh the better overall the results for amplitude and phase. The best results for mesh 1 were for $\delta t = 3.11$ s and $s_r = 0.875$, for mesh 2 were for $\delta t = 0.195$ s and $s_r = 0.944$, and for mesh 3 were for $\delta t = 0.195$ s and $s_r = 0.7$. Table 6.1 compares the analytical values for amplitude and phase of water height, ζ, at three x-values against the numerical nonadvective flow

Table 6.1: Comparison of the analytical values for water height, ζ (amplitudes in m and phases, for cosine waves, in degrees), at three x-values, with the numerical values for three different meshes for nonadvective flow.

x-values		Analytical Solution		Mesh 1		Mesh 2		Mesh 3	
	Period	Amp	Phase	Amp	Phase	Amp	Phase	Amp	Phase
x=1080 m	1345.71	1.028	0	1.031	-5.7	1.016	-2.0	1.020	-0.7
	678.86	0.102	0	0.119	16.2	0.122	9.5	0.118	4.5
	Zero frequency	0	-	0.032	-	0.011	-	0.004	-
x=2025 m	1345.71	1.928	0	1.912	-5.8	1.897	-2.0	1.905	-0.7
	678.86	0.102	0	0.082	12.7	0.109	8.5	0.111	4.4
	Zero frequency	0	-	0.057	-	0.017	-	0.006	-
x=2565 m	1345.71	2.442	0	2.375	-5.6	2.386	-1.8	2.406	-0.6
	678.86	0.102	0	0.005	212.7	0.063	9.1	0.084	5.2
	Zero frequency	0	-	0.069	-	0.022	-	0.006	-

values for three different meshes, while Table 6.2 compares values of the u-velocity. The last set of values, for x = 2565 m, is for nodes close to the region of wetting and drying.

A comparison of the numerical (without advection) and analytical values of the water surface at times $T/2$ and T for Mesh 3 is shown in the graphs in Figures 6.3 and 6.4. A plot of the numerical (without advection) and analytical values of the x-coordinate of the shoreline as a function of time over one period for Mesh 3 is shown in Figure 6.5. It can be seen that the analytical and numerical solutions were in good agreement.

Table 6.2: Comparison of the analytical values for U-velocity, (amplitudes in ms^{-1} and phases, for sine waves, in degrees), at three x-values, with the numerical values for three different meshes 3 for nonadvective flow.

x-values		Analytical Solution		Mesh 1		Mesh 2		Mesh 3	
	Period	Amp	Phase	Amp	Phase	Amp	Phase	Amp	Phase
x=0 m	1345.71	2.000	0.0	1.912	-5.8	1.940	-2.0	1.969	-0.7
x=1080 m	1345.71	2.000	0.0	1.873	-6.3	1.934	-2.1	1.966	-0.7
x=2025 m	1345.71	2.000	0.0	1.796	-6.4	1.907	-2.0	1.955	-0.7
x=2565 m	1345.71	2.000	0.0	1.528	-5.3	1.803	0.0	1.909	1.0

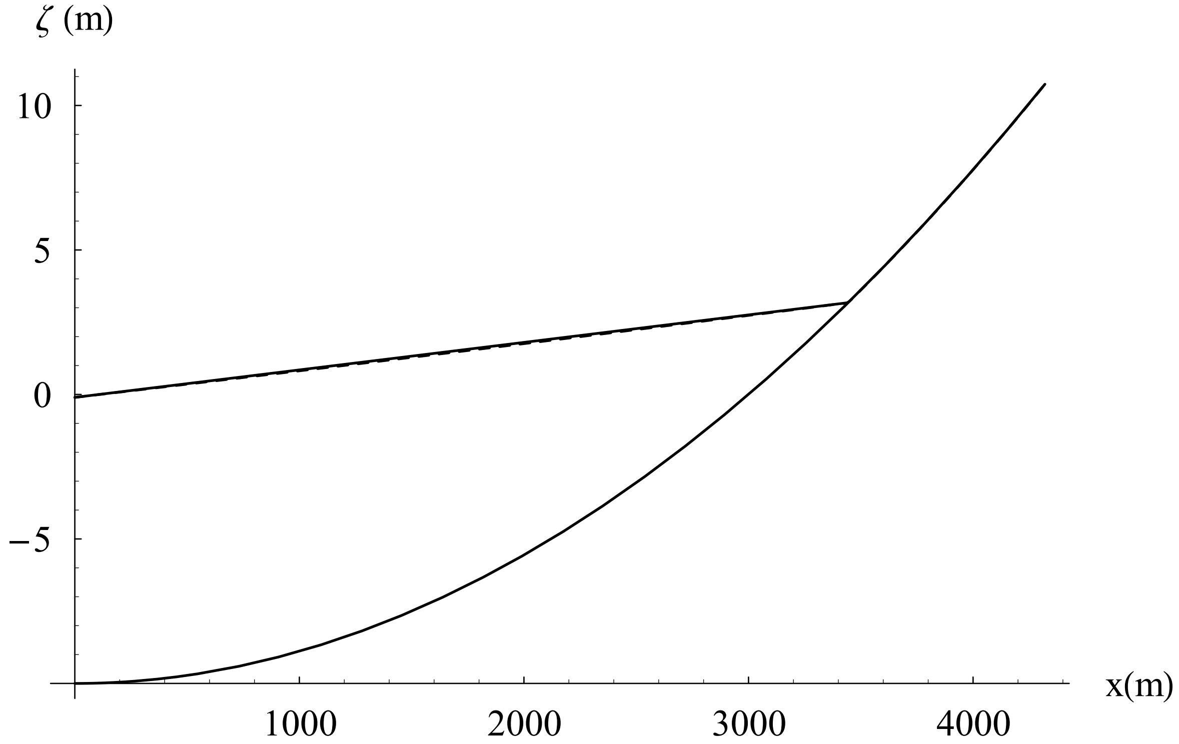

Figure 6.3: A comparison of the numerical (nonadvective flow) and analytical values of the water surface at time $t = T/2$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

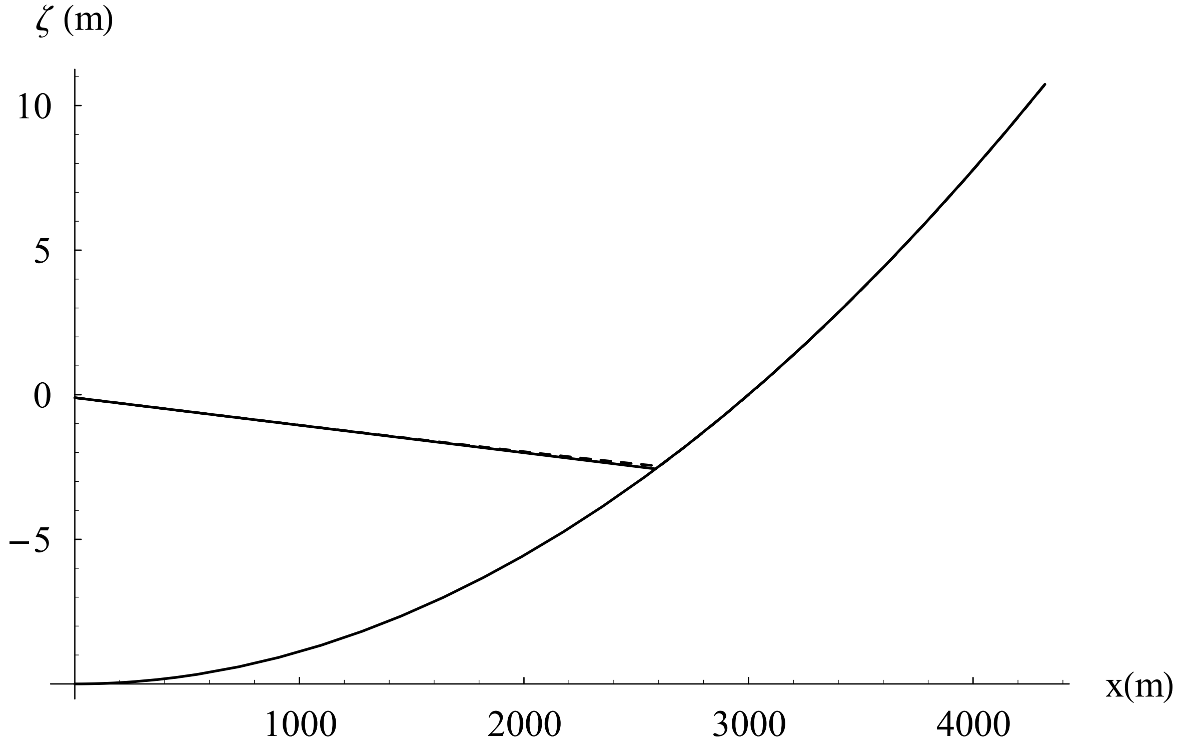

Figure 6.4: A comparison of the numerical (nonadvective flow) and analytical values of the water surface at time $t = T$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

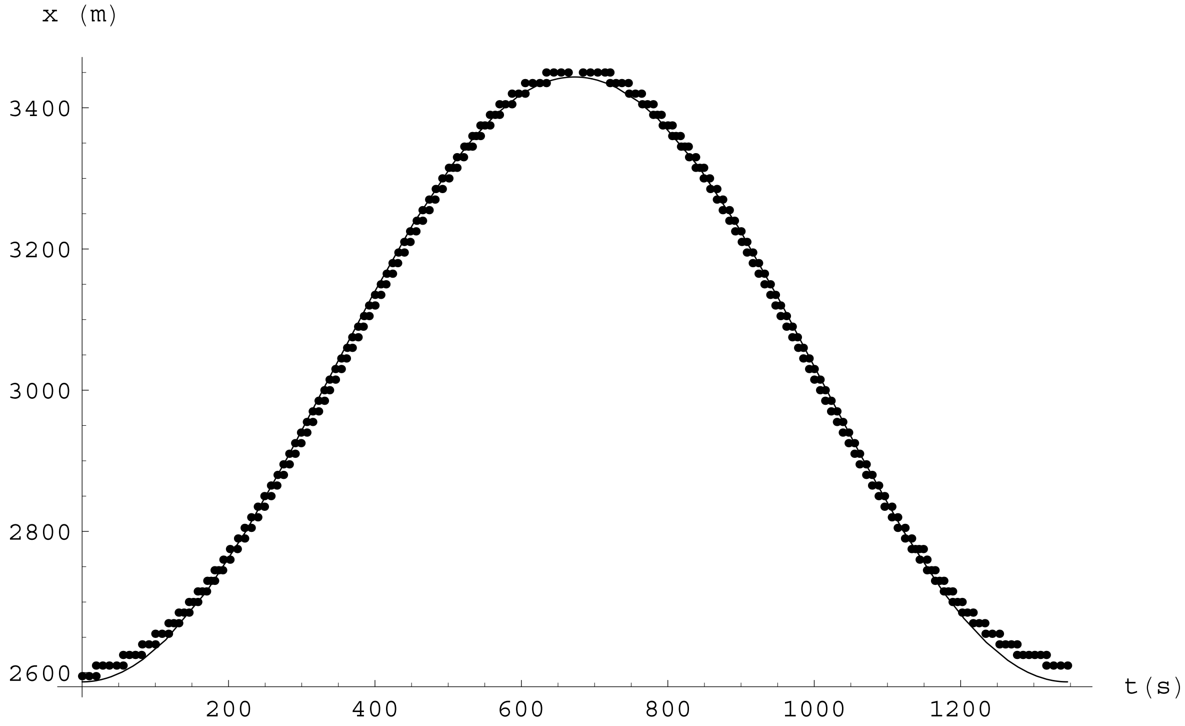

Figure 6.5: A plot of the numerical (nonadvective flow) and analytical values of the x-coordinate of the shoreline as a function of time. The analytical solution is a continuous curve while the numerical solution is a number of black dots.

After the model was run without the advective terms, the model was run with the advective terms with the finest mesh only for one period, T (1345.71 seconds); better results overall were obtained using advective terms than were found without the advective terms. The best results for advective flow in mesh 3 were found for with s_r=.95 and δt=.195 s. The numerical values of the amplitudes and phases of the water level (obtained using a Fast Fourier Transform package) are compared with the analytical values at three different nodes in Table 6.3. It can be seen that the values are in reasonable agreement. The next table, Table 6.4, shows results for U-velocity at four nodes in Mesh 3 for advective flow.

The results for mesh 3 advective flow only are presented in the rest of this section. The values of water elevation and velocity discussed below are for nodes sitting on a line parallel to the base of the rectangular region and half way between the base and top of the region. A comparison of the numerical (with advection) and analytical values of the water level, ζ, at times $t = T/8, T/4, 3T/8, T/2, 5T/8, 3T/4, 7T/8$ and T respectively for mesh 3 is shown in the graphs in Figures 6.6, 6.7, 6.8, 6.9, 6.10, 6.11, 6.12 and 6.13. In each figure the analytical solution is a continuous line whereas the numerical solution is a series of dashed lines. At all the time steps there is good agreement.

A plot of the numerical (with advection) and analytical values of the x-coordinate of the shoreline as a function of time over one period for mesh 3 is shown in Figure 6.14. The analytical solution is a continuous curve while the numerical solution is a series of dots. As can be seen there is good agreement between the analytical and numerical values. Presumably a finer mesh would give even better results. Comparison of Figures 6.5 (for nonadvective flow) and Figure 6.14 (for advective flow) shows that the latter gives slightly better results particularly towards the end of the period.

Table 6.3: Comparison of the analytical values for ζ (amplitudes in metres and phases, for cosine waves, in degrees), at three x-values, with the numerical values for Mesh 3 for advective flow.

x-values		Analytical Solution		Mesh 3	
	Period	Amp	Phase	Amp	Phase
x=1080 m	1345.71	1.028	0	1.038	-0.6
	678.86	0.102	0	0.095	3.5
	Zero frequency	0	-	0.003	-
x=2025 m	1345.71	1.928	0	1.943	-0.4
	678.86	0.102	0	0.098	-2.0
	Zero frequency	0	-	0.000	-
x=2565 m	1345.71	2.442	0	2.453	-0.4
	678.86	0.102	0	0.101	-8.4
	Zero frequency	0	-	0.003	-

Table 6.4: Comparison of the analytical values for U-velocity, (amplitudes in ms^{-1} and phases, for sine waves, in degrees), at three x-values, with the numerical values for Mesh 3 for advective flow.

x-values		Analytical Solution		Mesh 3	
	Period	Amp	Phase	Amp	Phase
x=0 m	1345.71	2.000	0.0	2.024	1.3
x=1080 m	1345.71	2.000	0.0	2.001	-0.3
x=2025 m	1345.71	2.000	0.0	1.992	-0.3
x=2565 m	1345.71	2.000	0.0	1.954	0.1

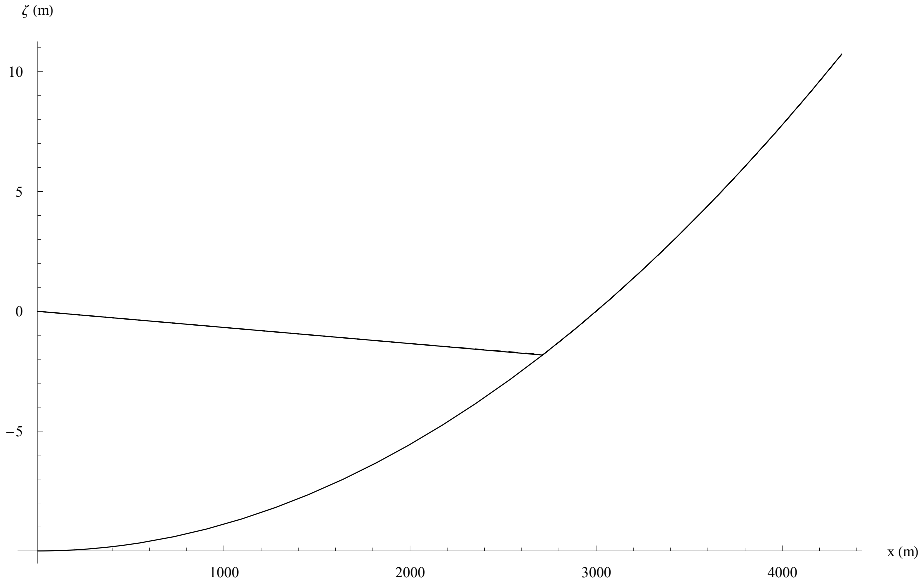

Figure 6.6: A comparison of the numerical (advective flow) and analytical values of the water surface at time $t = T/8$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

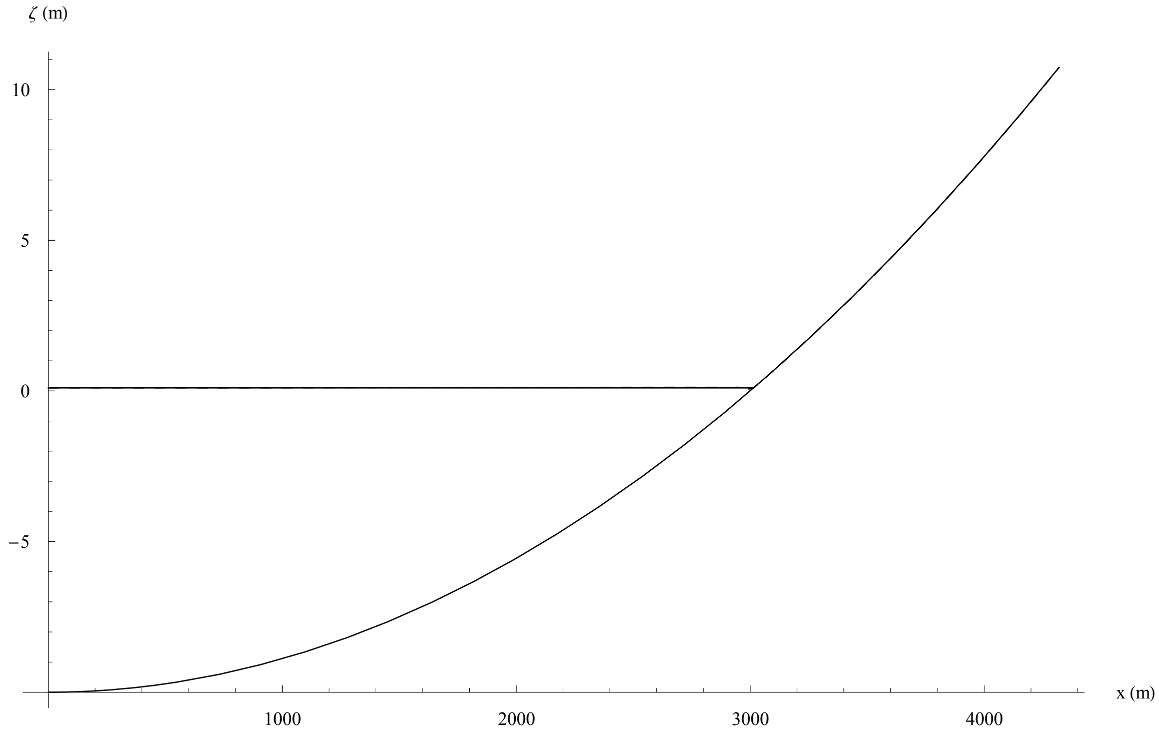

Figure 6.7: A comparison of the numerical (advective flow) and analytical values of the water surface at time $t = T/4$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

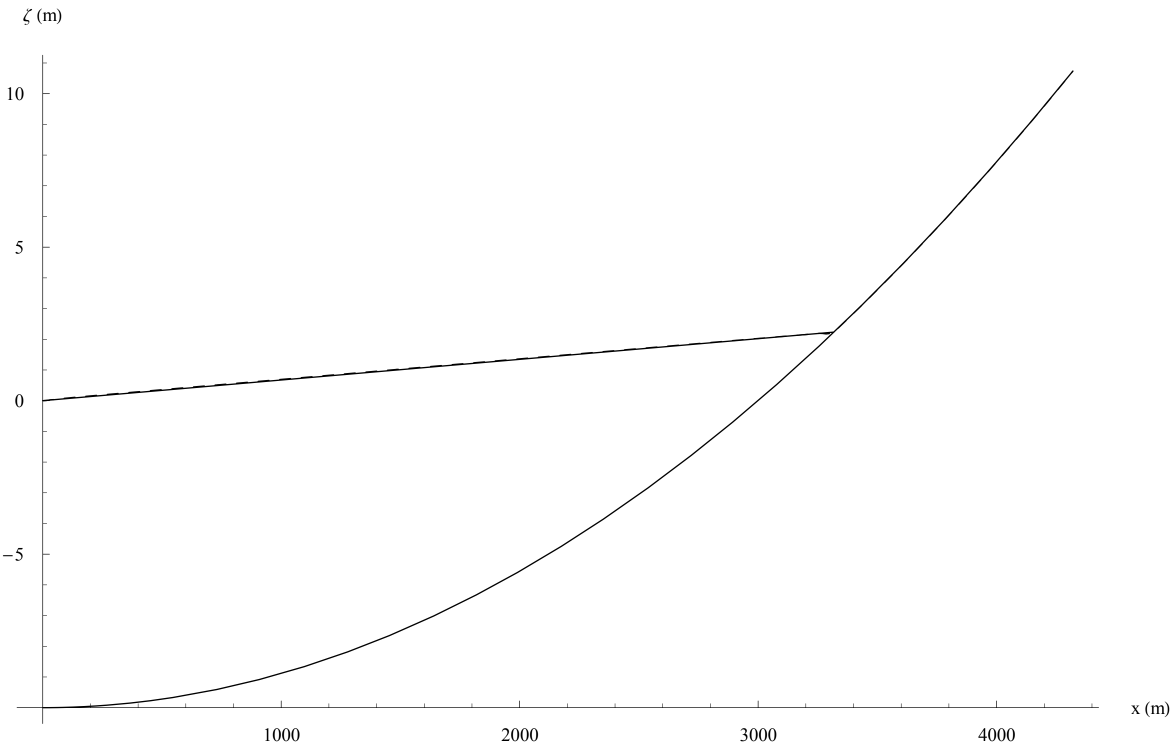

Figure 6.8: A comparison of the numerical (advective flow) and analytical values of the water surface at time $t = 3T/8$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

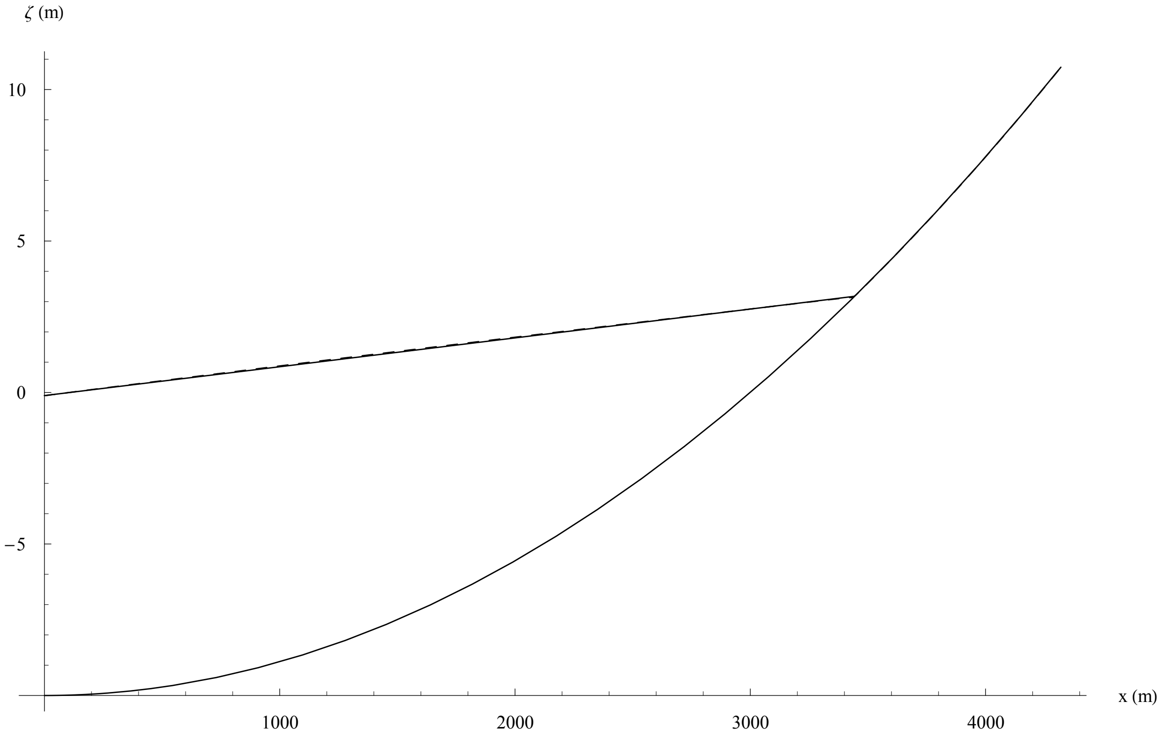

Figure 6.9: A comparison of the numerical (advective flow) and analytical values of the water surface at time $t = T/2$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

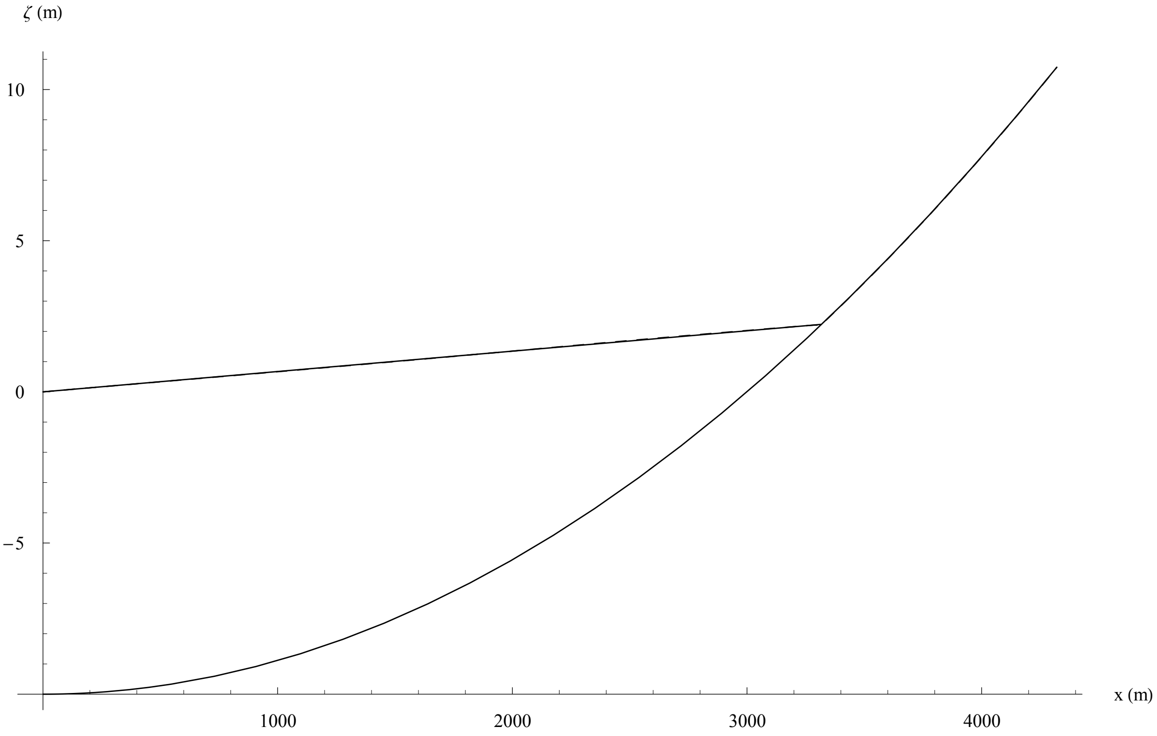

Figure 6.10: A comparison of the numerical (advective flow) and analytical values of the water surface at time $t = 5T/8$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

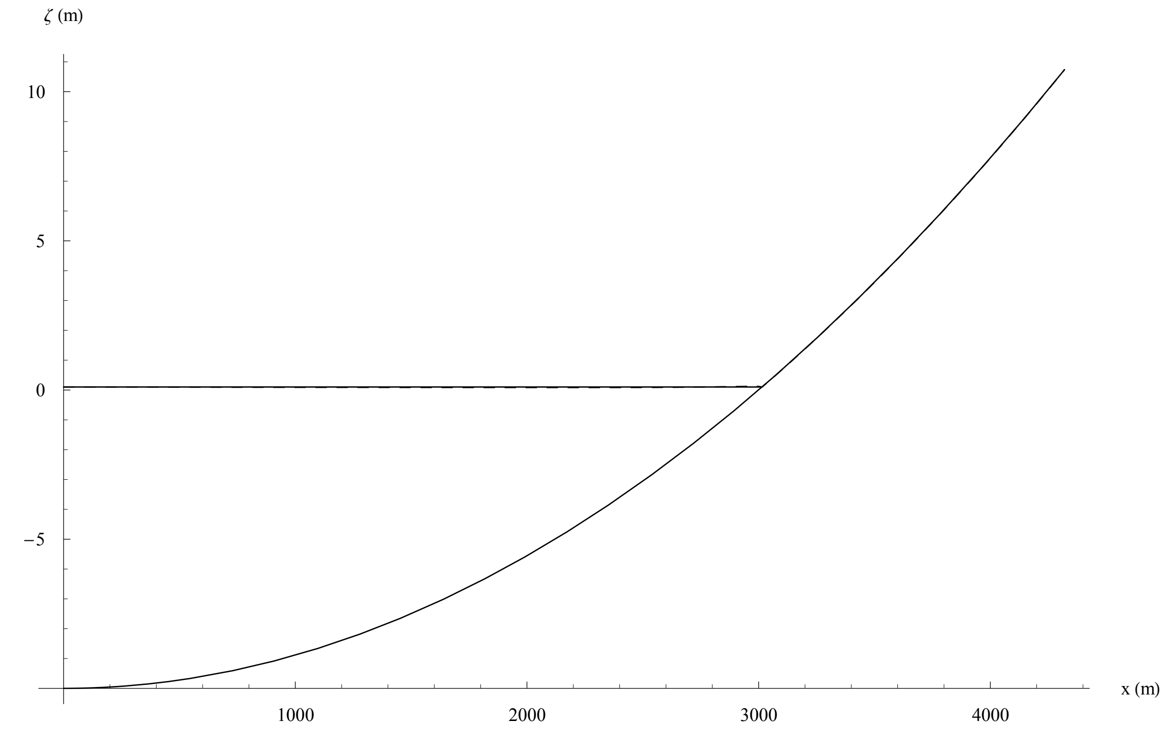

Figure 6.11: A comparison of the numerical (advective flow) and analytical values of the water surface at time $t = 3T/4$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

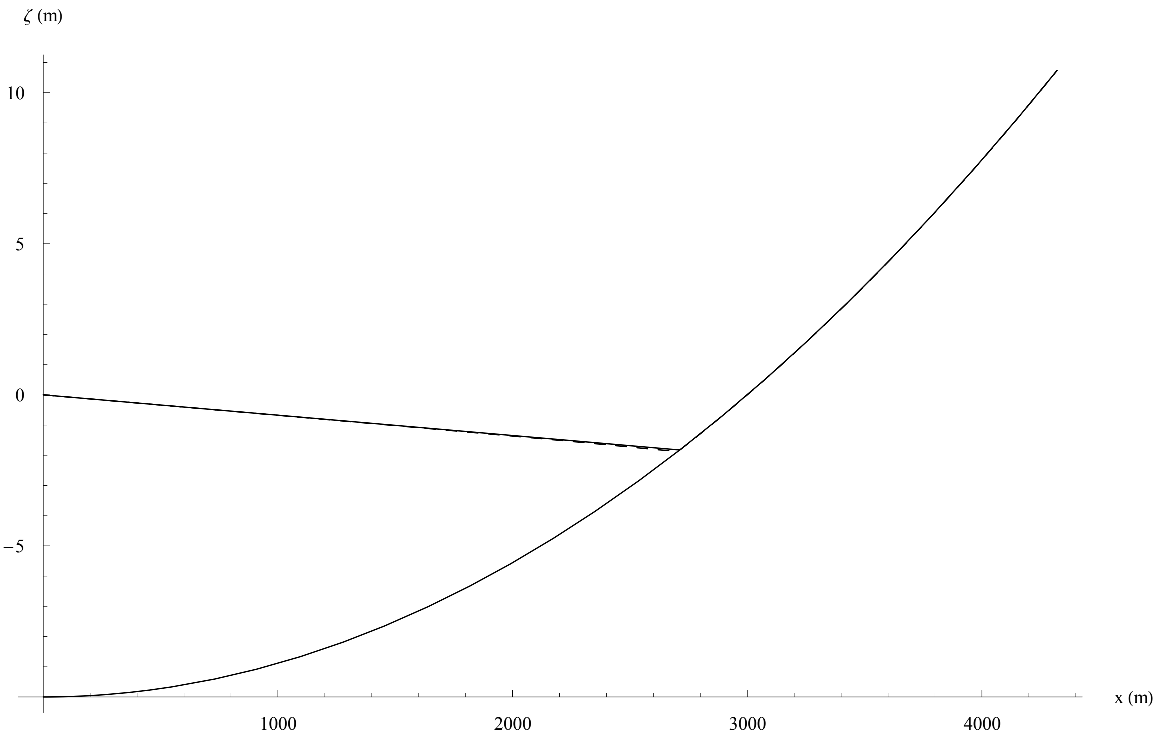

Figure 6.12: A comparison of the numerical (advective flow) and analytical values of the water surface at time $t = 7T/8$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

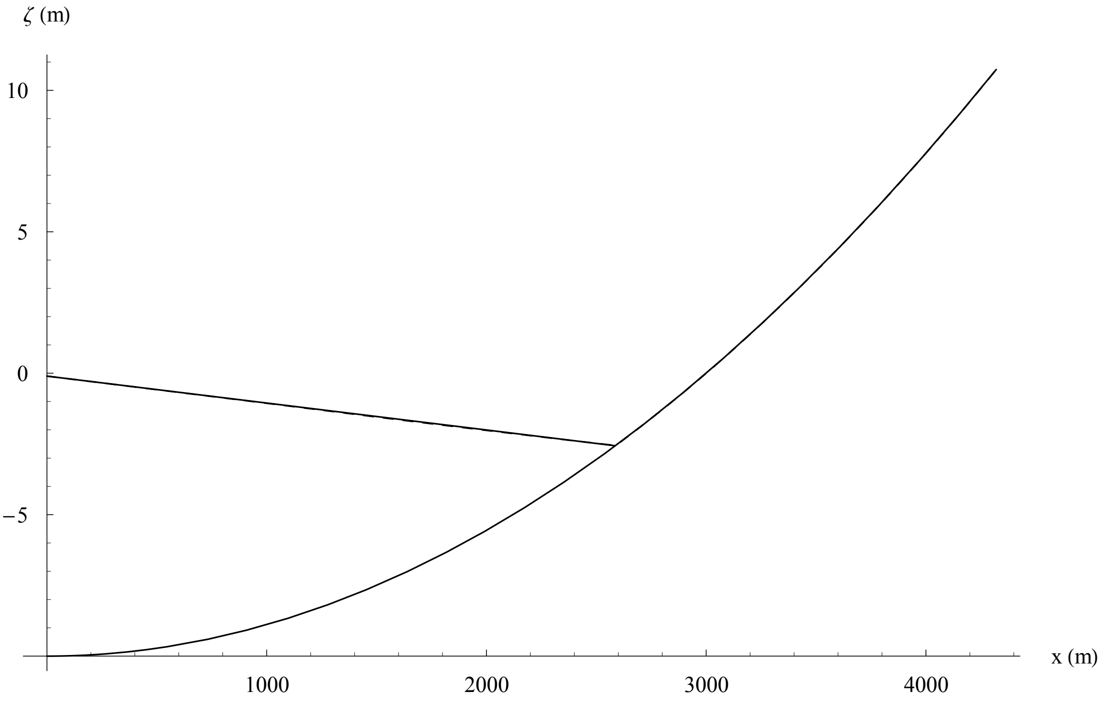

Figure 6.13: A comparison of the numerical (advective flow) and analytical values of the water surface at time $t = T$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

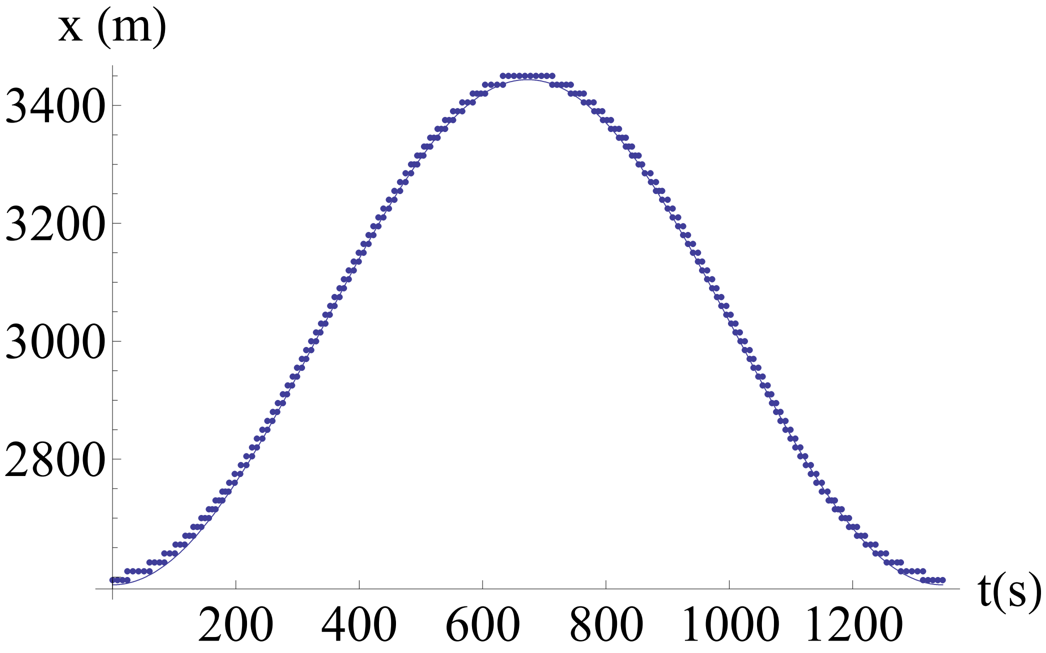

Figure 6.14: The numerical (advective flow) x-coordinate of the shoreline against time plotted together with the analytical x-coordinate of the shoreline against time. The analytical solution is a continuous curve while the numerical solution is a series of black dots.

A graphical comparison of the numerical and analytical values for ζ, the water height above the x-axis, against time over one period at three nodes sitting on a line parallel to the base of the rectangular mesh 3 region and half way between the base and top of the region (for $x = 750$ m, $x = 1500$ m and $x = 2250$ m) are shown respectively in Figures 6.15, 6.16 and 6.17. It can be seen that in each graph that the values are close.

A graphical comparison of the numerical and analytical values for the U-velocity against time at four mesh 3 nodes sitting on a line parallel to the base of the rectangular region and half way between the base and top of the region (for $x = 0$ m, $x = 750$ m, $x = 1500$ m and $x = 2250$ m) are shown respectively in Figures 6.18, 6.19, 1 6.20 and 6.21. It can be seen that the values are close. The closer a node is to the left hand side of the mesh, the closer the numerical values are to the analytical values.

The graph in Figure 6.22 compares the analytical U-velocity against the numerical U-velocity (using the advective scheme) at time $t = T/4$. It shows good agreement.

6.3.3 The second analytical solution used for comparison

The moving boundary numerical model was secondly tested against an analytical solution for shallow water linear frictional flow with no eddy viscosity and with no Coriolis force and with forcing which decays over time in a bed with quadratically varying depth. This solution was derived in subsection 3.7.3. The bed has its depth below the x-axis given by

$$h = h_0 \left(1 - \frac{x^2}{a^2}\right), \; x \geq 0. \tag{6.11}$$

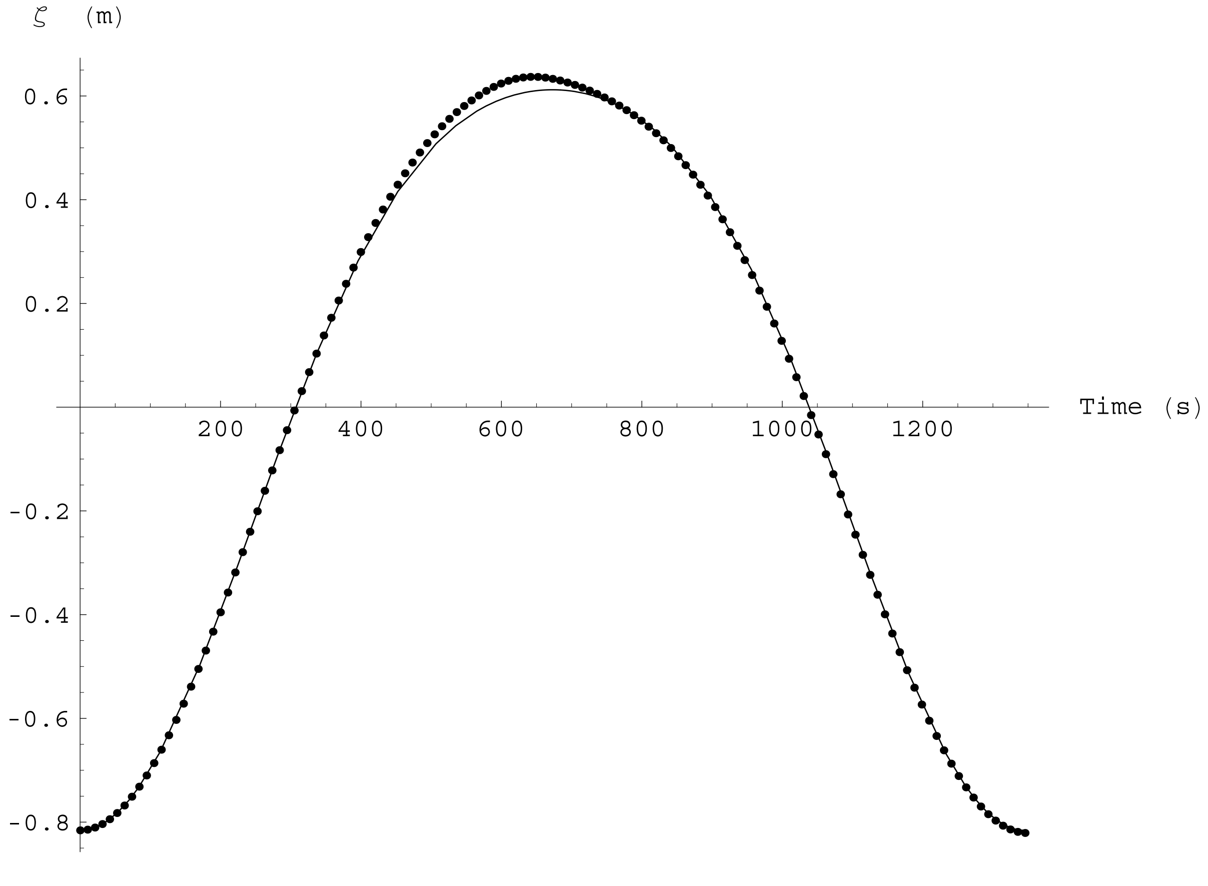

Figure 6.15: The numerical (advective flow) water level against time plotted together with the analytical zeta against time at a node at $x = 750$ m and midway between the top and bottom boundary of Mesh 3, with the numerical plot a series of dots and the analytical plot a continuous line.

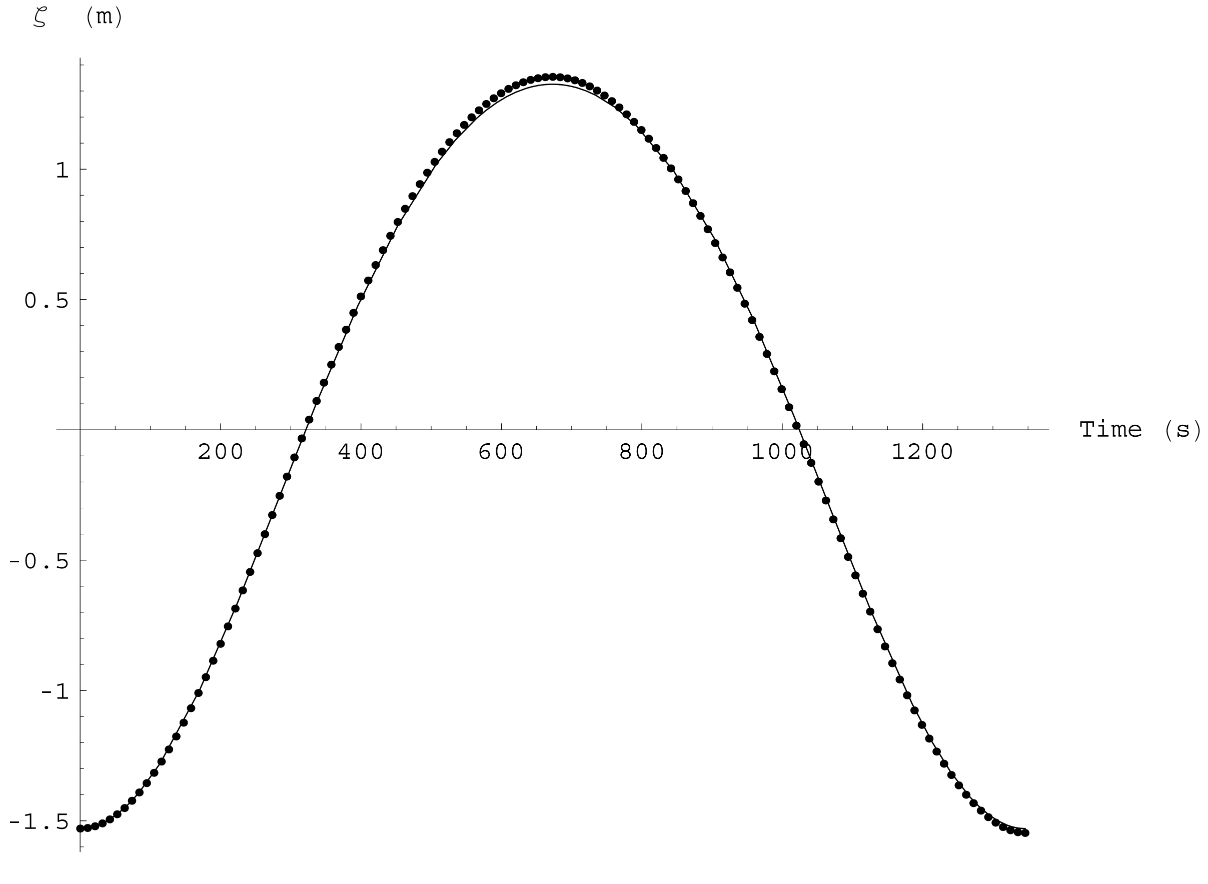

Figure 6.16: The numerical (advective flow) water level against time plotted together with the analytical zeta against time at a node at $x = 1500$ m and midway between the top and bottom boundary of Mesh 3, with the numerical plot a series of dots and the analytical plot a continuous line.

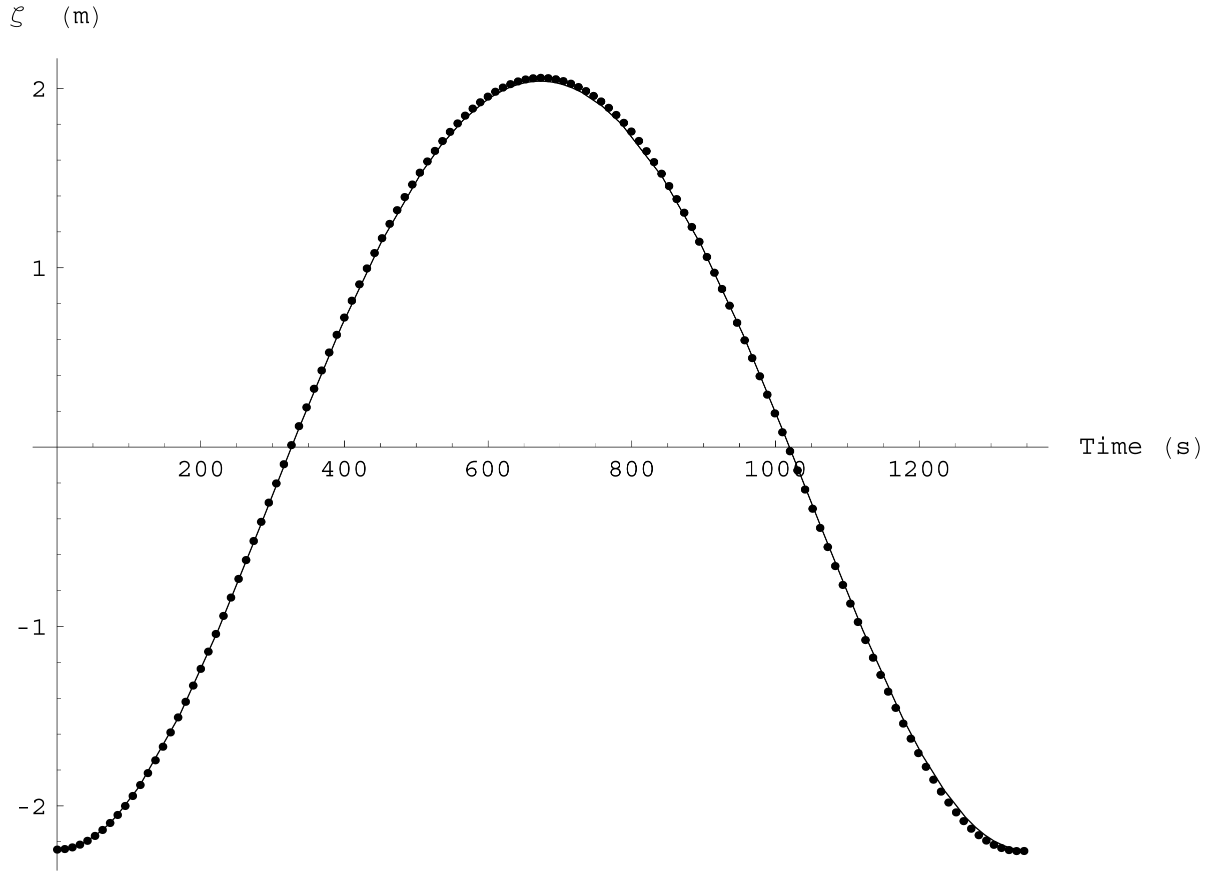

Figure 6.17: The numerical (advective flow) water level against time plotted together with the analytical zeta against time at a node at $x = 2250$ m and midway between the top and bottom boundary of Mesh 3, with the numerical plot a series of dots and the analytical plot a continuous line.

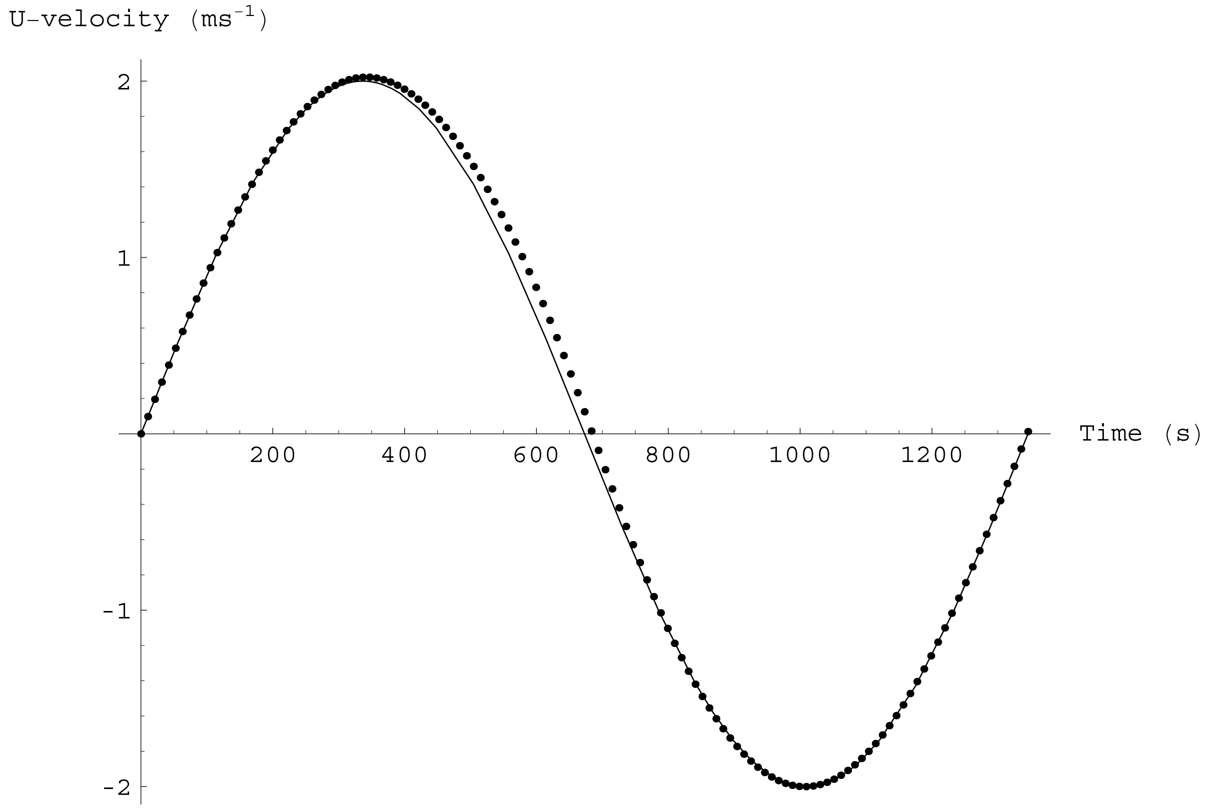

Figure 6.18: The numerical (advective flow) U-velocity against time plotted together with the analytical U-velocity against time at a node at $x = 0$ m and midway between the top and bottom boundary of Mesh 3, with the numerical plot a series of dots and the analytical plot a continuous line.

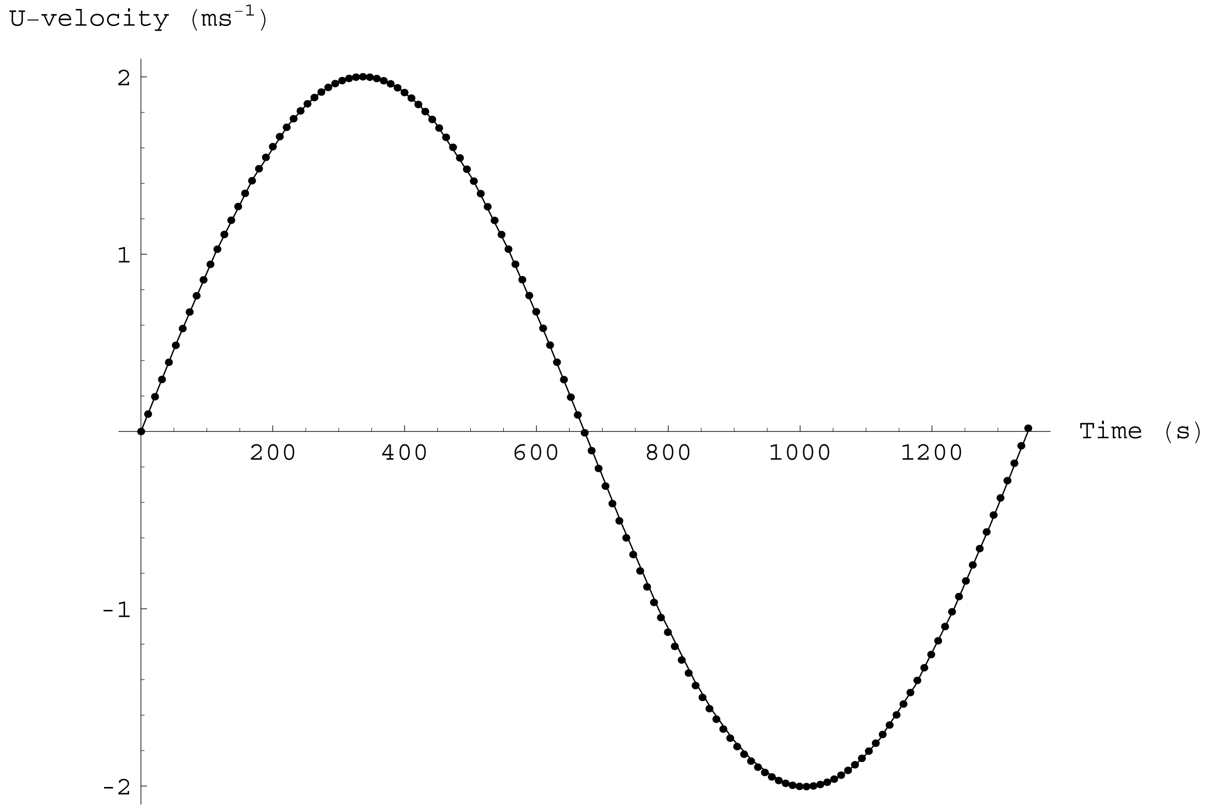

Figure 6.19: The numerical (advective flow) U-velocity against time plotted together with the analytical U-velocity against time at a node at $x = 750$ m and midway between the top and bottom boundary of Mesh 3, with the numerical plot a series of dots and the analytical plot a continuous line.

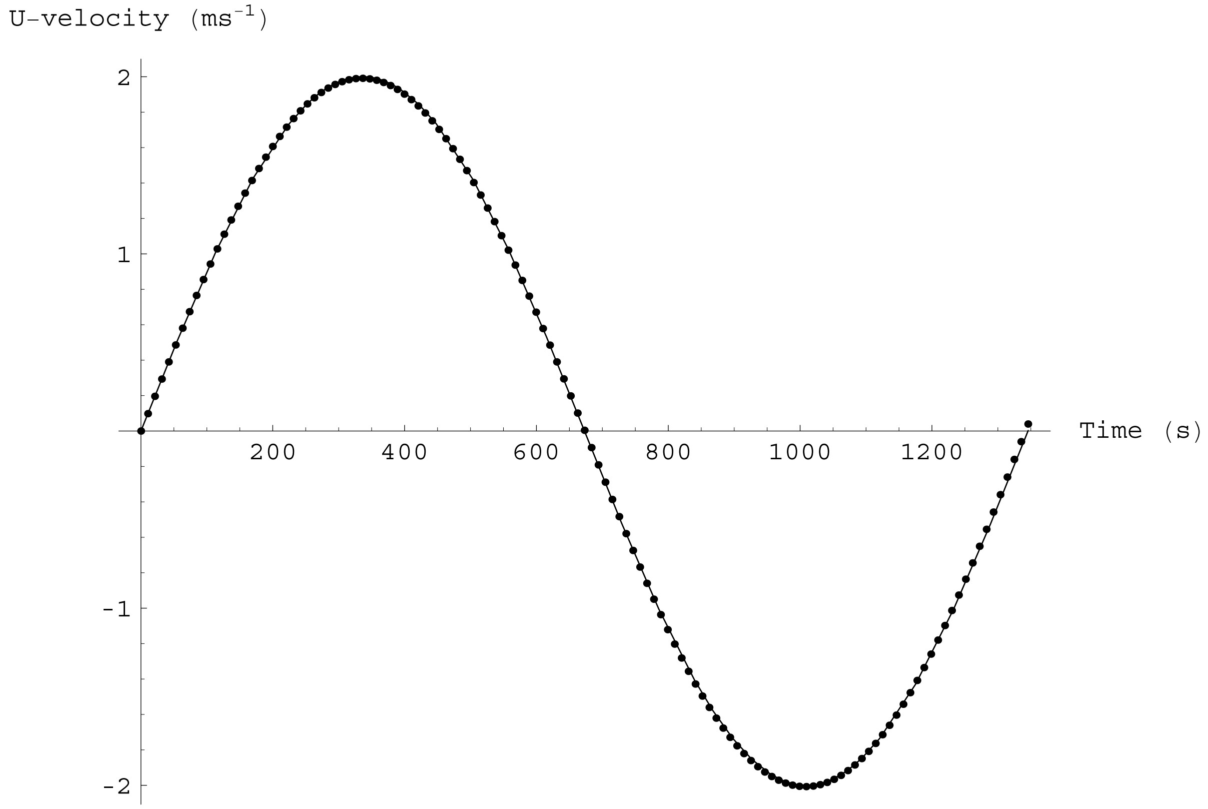

Figure 6.20: The numerical (advective flow) U-velocity against time plotted together with the analytical U-velocity against time at a node at $x = 1500$ m and midway between the top and bottom boundary of Mesh 3, with the numerical plot a series of dots and the analytical plot a continuous line.

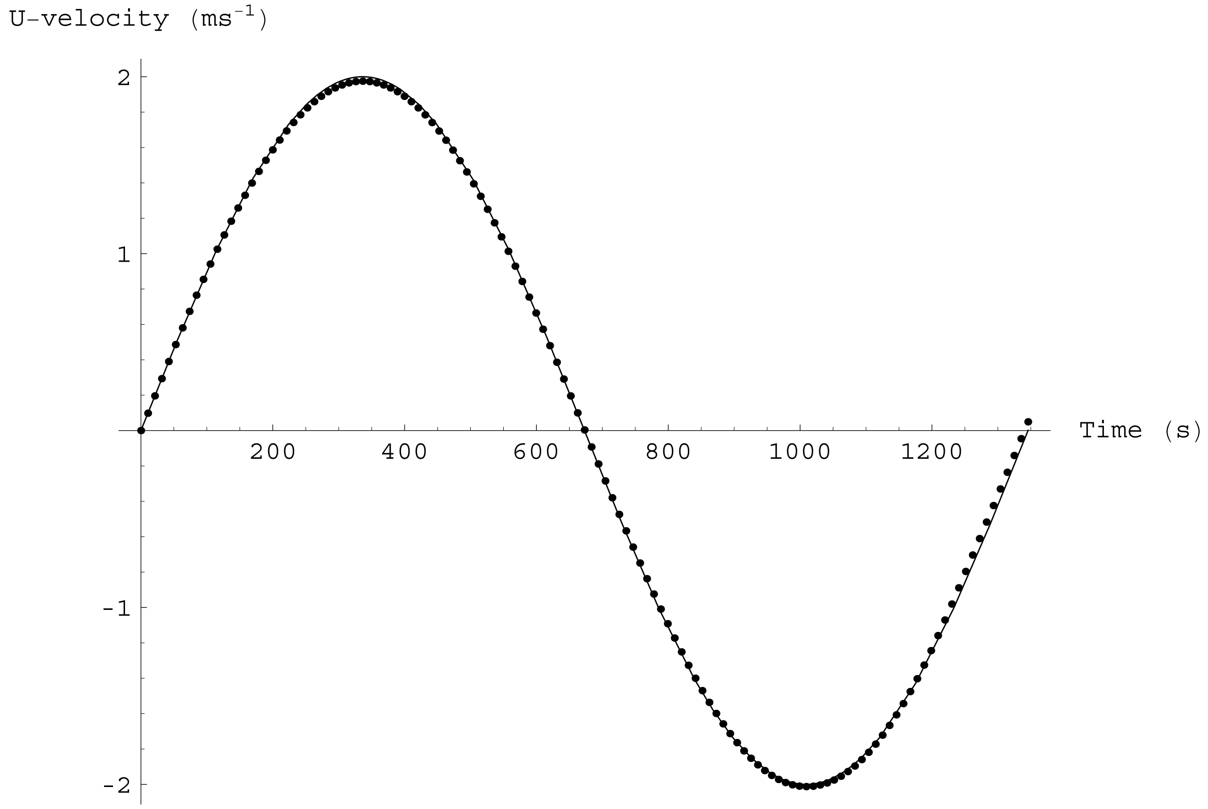

Figure 6.21: The numerical (advective flow) U-velocity against time plotted together with the analytical U-velocity against time at a node at $x = 2250$ m and midway between the top and bottom boundary of Mesh 3, with the numerical plot a series of dots and the analytical plot a continuous line.

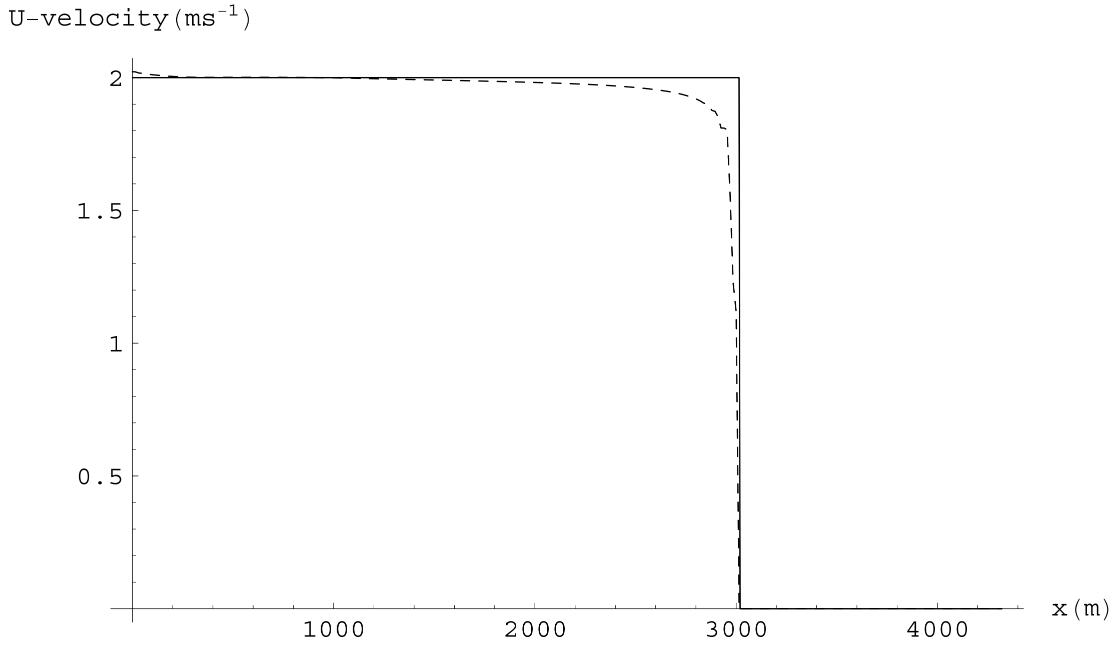

Figure 6.22: The analytical U-velocity against the numerical (advective flow) U-velocity at time $t = T/4$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

The bottom friction parameter, τ, is taken to be a constant, and with $\tau < p$, where

$$p = \sqrt{\frac{8gh_0}{a^2}}. \tag{6.12}$$

The forcing occurs at $x = 0$, and is given by

$$\zeta(0,t) = \frac{a^2B^2e^{-\tau t}}{8g^2h_0}\left(-s\tau \sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right) - \frac{B^2e^{-\tau t}}{4g}, \tag{6.13}$$

where B is a constant, and

$$s = \frac{\sqrt{8gh_0 - \tau^2}}{2a}. \tag{6.14}$$

The analytical solution for the water level, τ, is

$$\begin{aligned}\zeta(x,t) = \frac{a^2B^2e^{-\tau t}}{8g^2h_0}\left(-s\tau \sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right) - \frac{B^2e^{-\tau t}}{4g}\\ -\frac{e^{-\frac{\tau t}{2}}}{g}\left(Bs\cos st + \frac{\tau B}{2}\sin st\right)x,\end{aligned} \tag{6.15}$$

and for the velocity, U, is

$$U = Be^{-(\tau t)/2}\sin st. \tag{6.16}$$

The x-coordinate of the shoreline is given by

$$x = \frac{a^2e^{-(\tau t)/2}}{2gh_0}\left(-Bs\cos st - \frac{\tau B}{2}\sin st\right) + a. \tag{6.17}$$

6.3.4 The second analytical solution versus the numerical solution

For the numerical model the values chosen were $h_0 = 10$ m, $a = 3000$m, $\tau = 0.001$ s^{-1} and $B = 2$ ms^{-1} with the initial values of ζ and U set to those of the analytical model. The period of the trigonometric terms in the motion, T, is 1353.49 s. The initial position of the water surface is shown in Figure 6.23. The initial velocity is 0 ms^{-1}. At the open water boundary, at

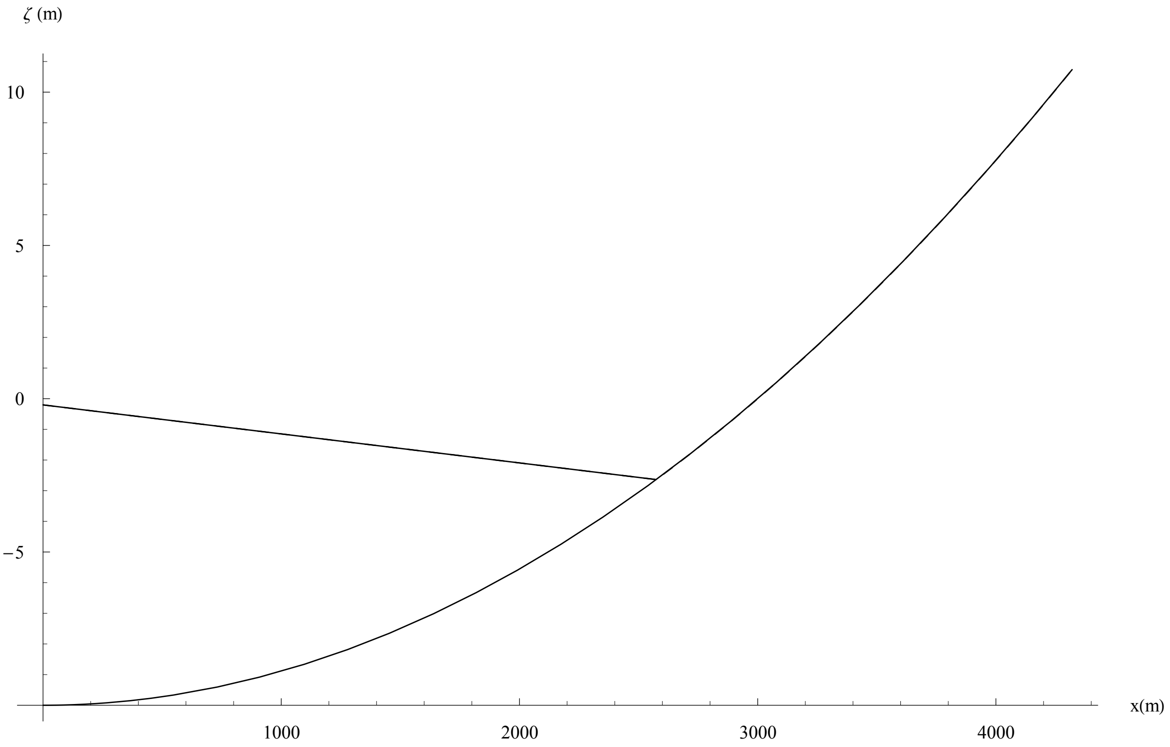

Figure 6.23: A vertical cross-section of the initial position of the water for linear frictional flow above a bed with quadratically varying depth for $h_0 = 10$ m, $a = 3000$ m, $\tau = 0.001$ s^{-1} and $B = 2$ ms^{-1}.

$x = 0$, the water level was specified as the same function of time as in the analytical model. The calculation was done over eight periods (10827.90 s).

A triangular mesh was used, covering a rectangular region of width 4320 m in the x direction and height 240 m in the y direction. Each triangle in the mesh is an isosceles right angled triangle. The mesh contains 4913 nodes and 9216 elements. As the velocity in the analytical solution is a function of t only, the advective terms are zero; for this reason the model was run without advective terms. The numerical model was run initially with $s_r = 0.99$. It was found that δt had to be set to 0.195 s for convergence. It was also found that the larger s_r was the larger the maximum value of δt for convergence. The results presented below are for $s_r = 0.99$ and $\delta t = 0.195$ s. The values of water elevation, ζ, and the U-velocity discussed below are for nodes sitting on a line parallel to the base of the rectangular region and half way between the base and top of the region.

A plot of the numerical and analytical x-coordinates of the shoreline as a function of time over one period is shown in Figure 6.24 and over eight periods is shown in Figure 6.25. The analytical solution is shown in each diagram as a continuous curve while the numerical solution is a number of points; these points are so close together that they appear to be a number of straight lines parallel to the time axis. As the distance between successive nodes is 15 m, the distance between successive apparent straight lines is 15 m, which means that numerically when the shoreline moves it moves 15 m in one time step. As can be seen, there is good agreement between the analytical and numerical values.

A graphical comparison of the numerical and analytical values for ζ, the water height above the x-axis, against time, t, is shown over one period at the node at which $x = 750$ m in Figure 6.26. Figures 6.27, 6.28, 6.29 and 6.30 show the water heights over eight periods at the four nodes for which

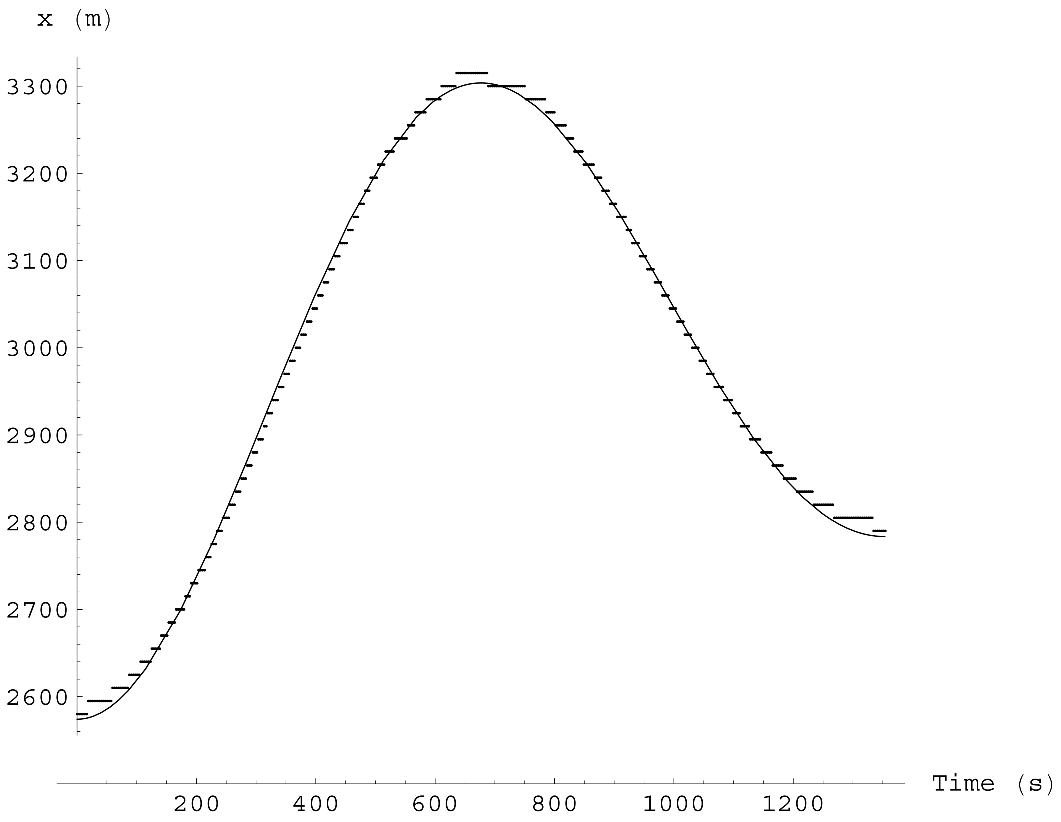

Figure 6.24: A plot of the numerical and analytical values of the x-coordinate of the shoreline as a function of time over one period. The analytical solution is a continuous curve while the numerical solution is a number of points.

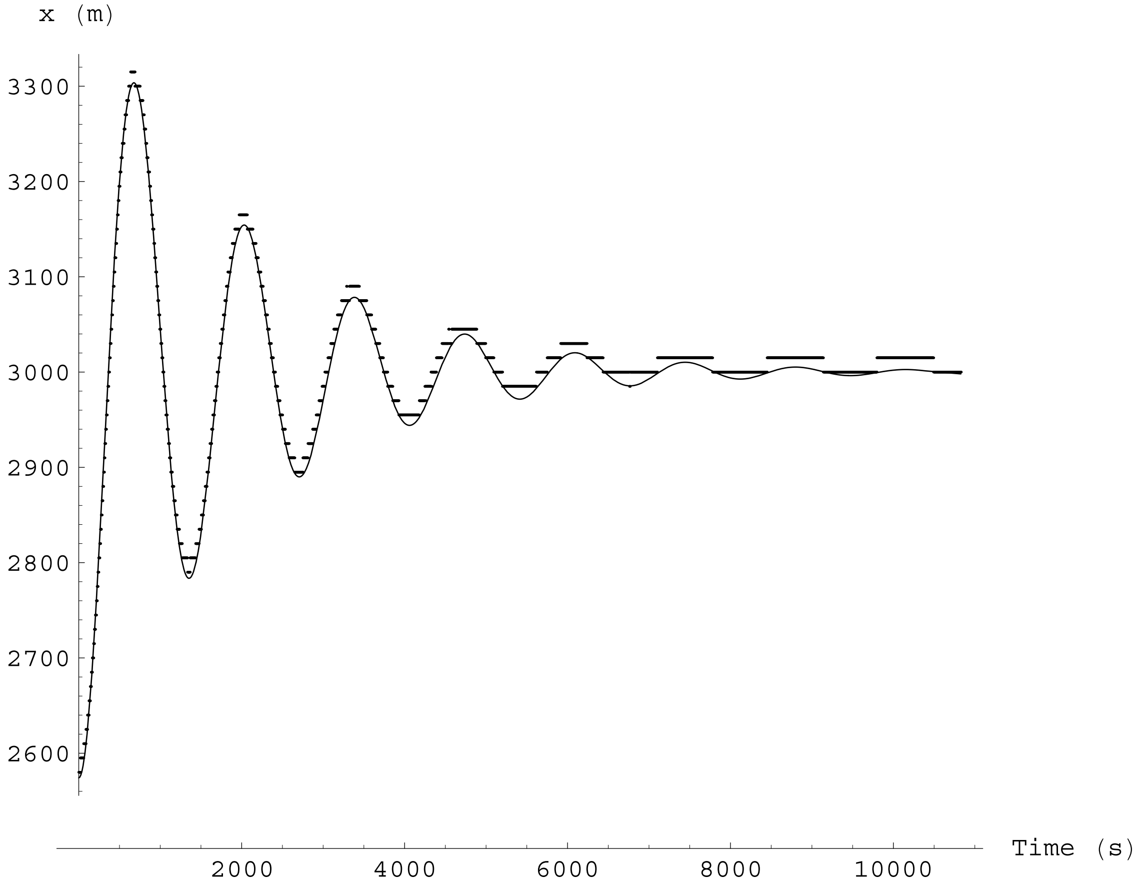

Figure 6.25: A plot of the numerical and analytical values of the x-coordinate of the shoreline as a function of time over eight periods. The analytical solution is a continuous curve while the numerical solution is a number of points.

$x = 750$ m, $x = 1500$ m, $x = 2250$ m and $x = 2565$ m. It can be seen that in each graph that the analytical and numerical values are close.

Figures 6.31, 6.32, 6.33, 6.34 and 6.35 show the U-velocity over eight periods at the four nodes for which $x = 750$ m, $x = 1500$ m, $x = 2250$ m and $x = 2565$ m. It can be seen that in each graph that the analytical and numerical values are close.

Figures 6.36, 6.37, 6.38 and 6.39 show the water level, ζ against x at times $t = T/2$, $t = T$, $t = 3T/2$ and $2T$ respectively. It can be seen that in each graph that the analytical and numerical values are close

The graph in Figure 6.40 compares the analytical U-velocity against the mumerical U-velocity at time $t = T/4$. It shows good agreement.

6.4 Conclusions

A moving boundary shallow water wave numerical model has been developed which is a modification of the Selective Lumped Mass model and is useful for comparing against analytical moving boundary solutions and for modelling tidal flow in bays which contain tidal flats, i.e a region which is wet for part of the day and dry for part of the day. It has been tested against two analytical solutions developed by author. One solution was for the case of frictionless one dimensional moving boundary shallow water wave flow with cosine forcing above a bed with quadratically varying depth. A comparison was made for nonadvective flow for successively finer finite element meshes. There is good agreement between the numerical and analytical solutions with better solutions the finer the mesh. The numerical model was also run with advective flow for the finest mesh. It was found that the advective numerical model gave better results than the nonadvective numerical model. Presumably, a finer mesh than the last one used would give a better result. The second an-

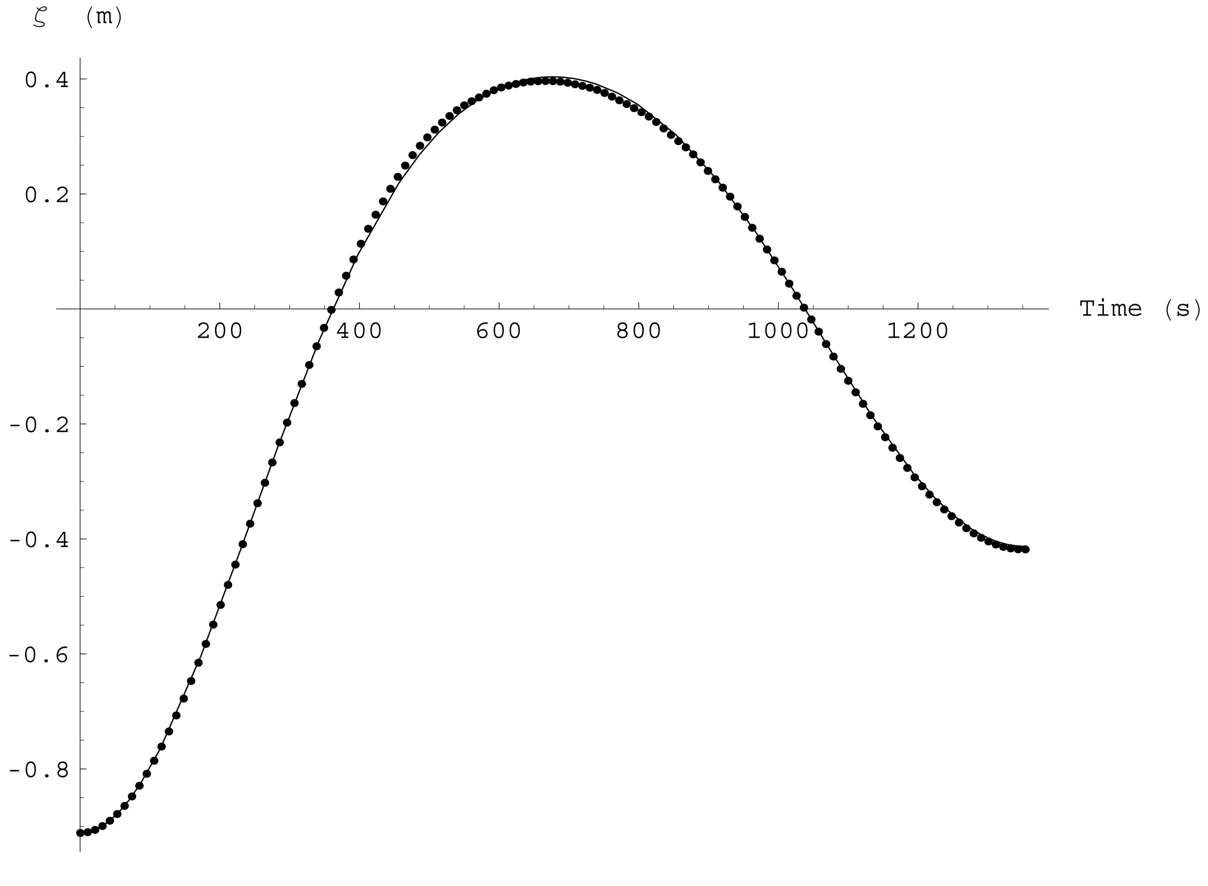

Figure 6.26: The numerical water level against time plotted together with the analytical zeta against time over one period at a node at $x = 750$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

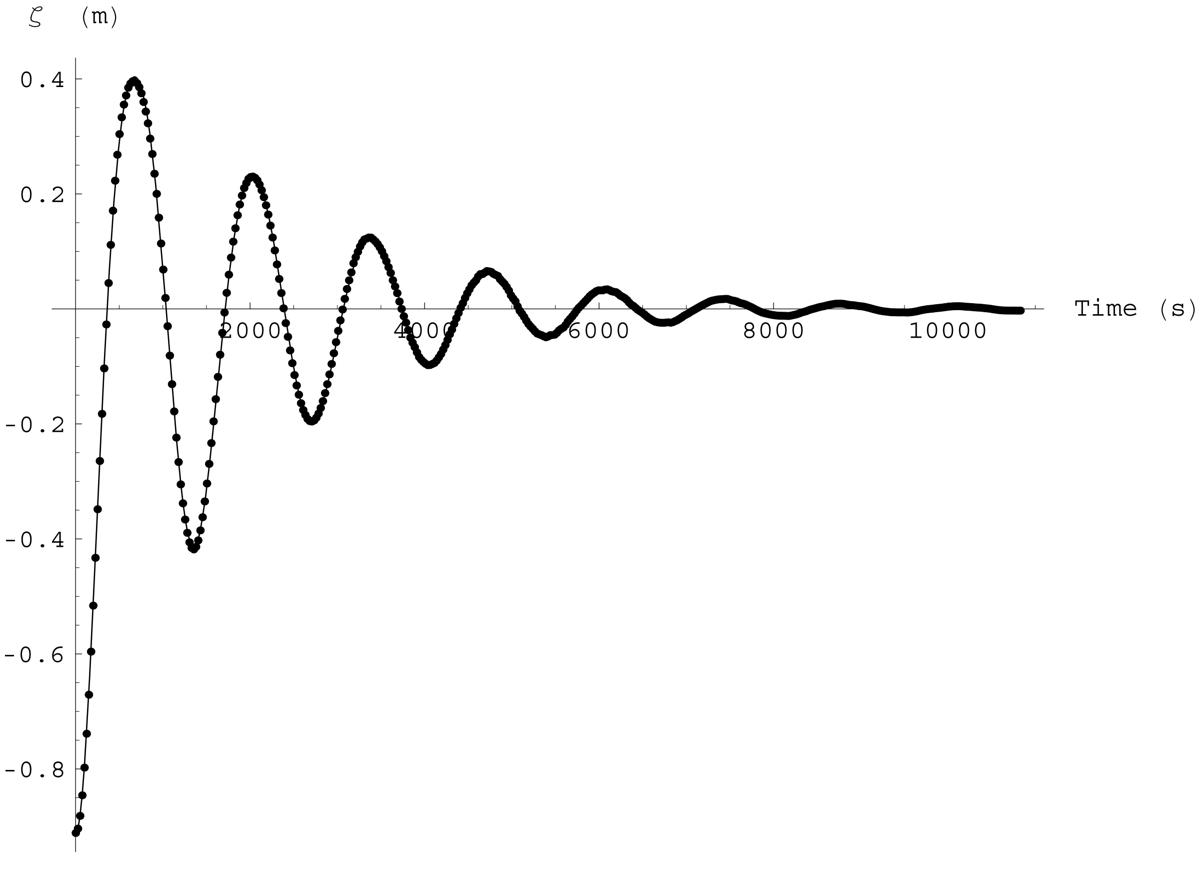

Figure 6.27: The numerical water level against time plotted together with the analytical zeta against time over eight periods at a node at $x = 750$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

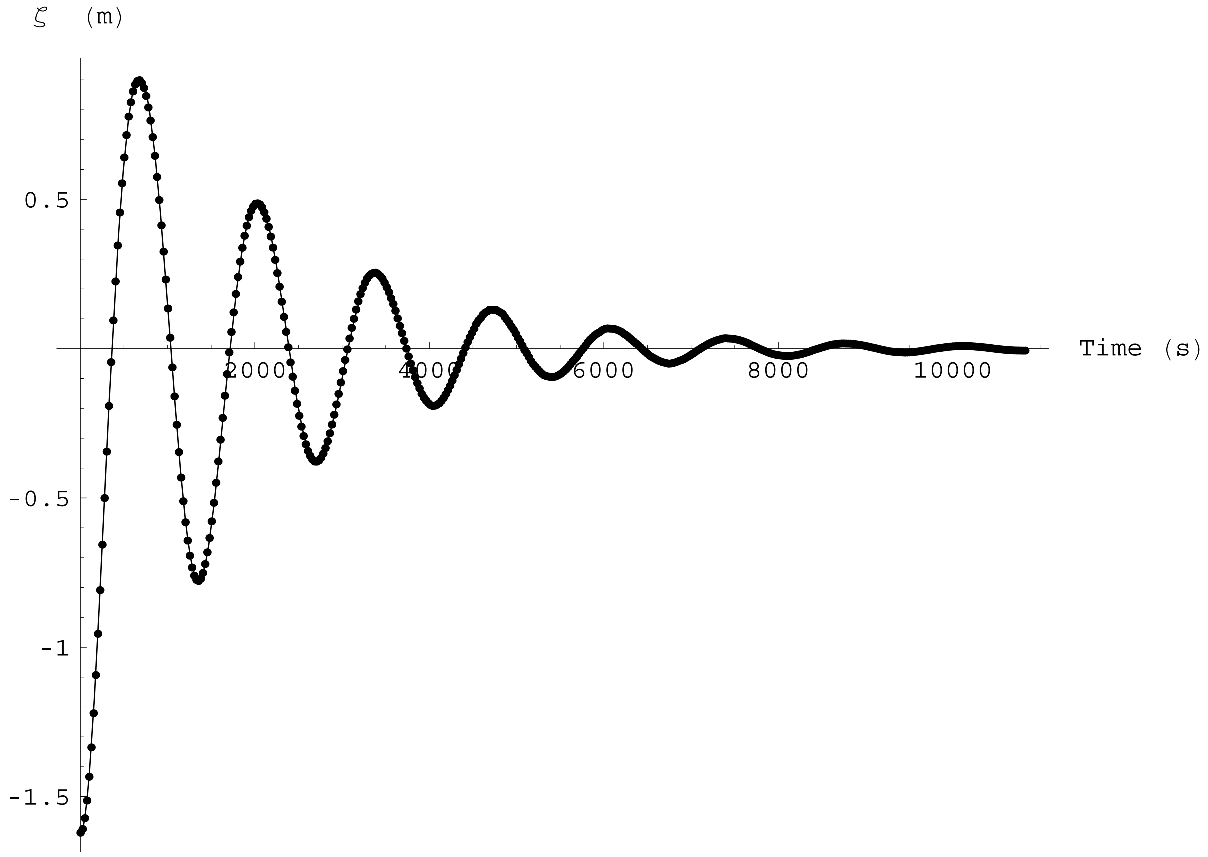

Figure 6.28: The numerical water level against time plotted together with the analytical zeta against time over eight periods at a node at $x = 1500$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

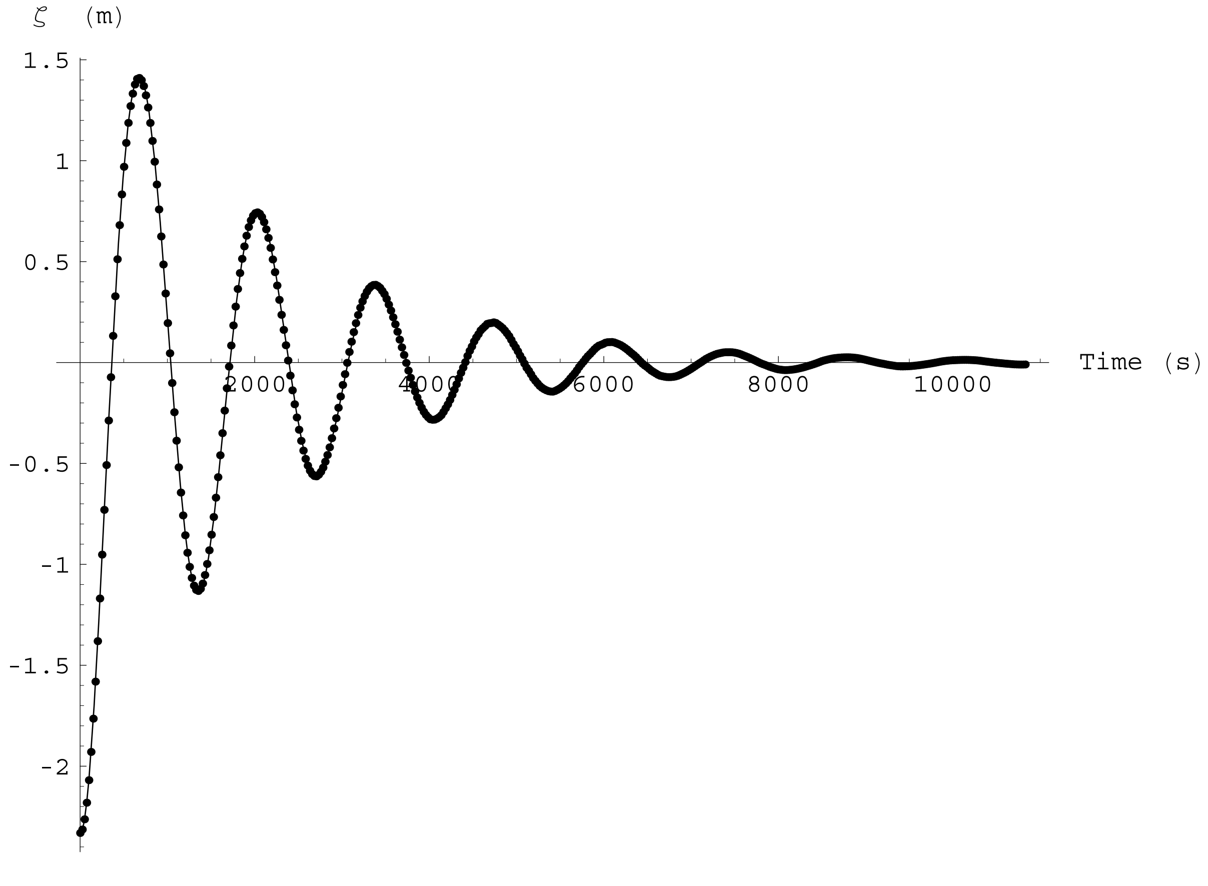

Figure 6.29: The numerical water level against time plotted together with the analytical zeta against time over eight periods at a node at $x = 2250$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

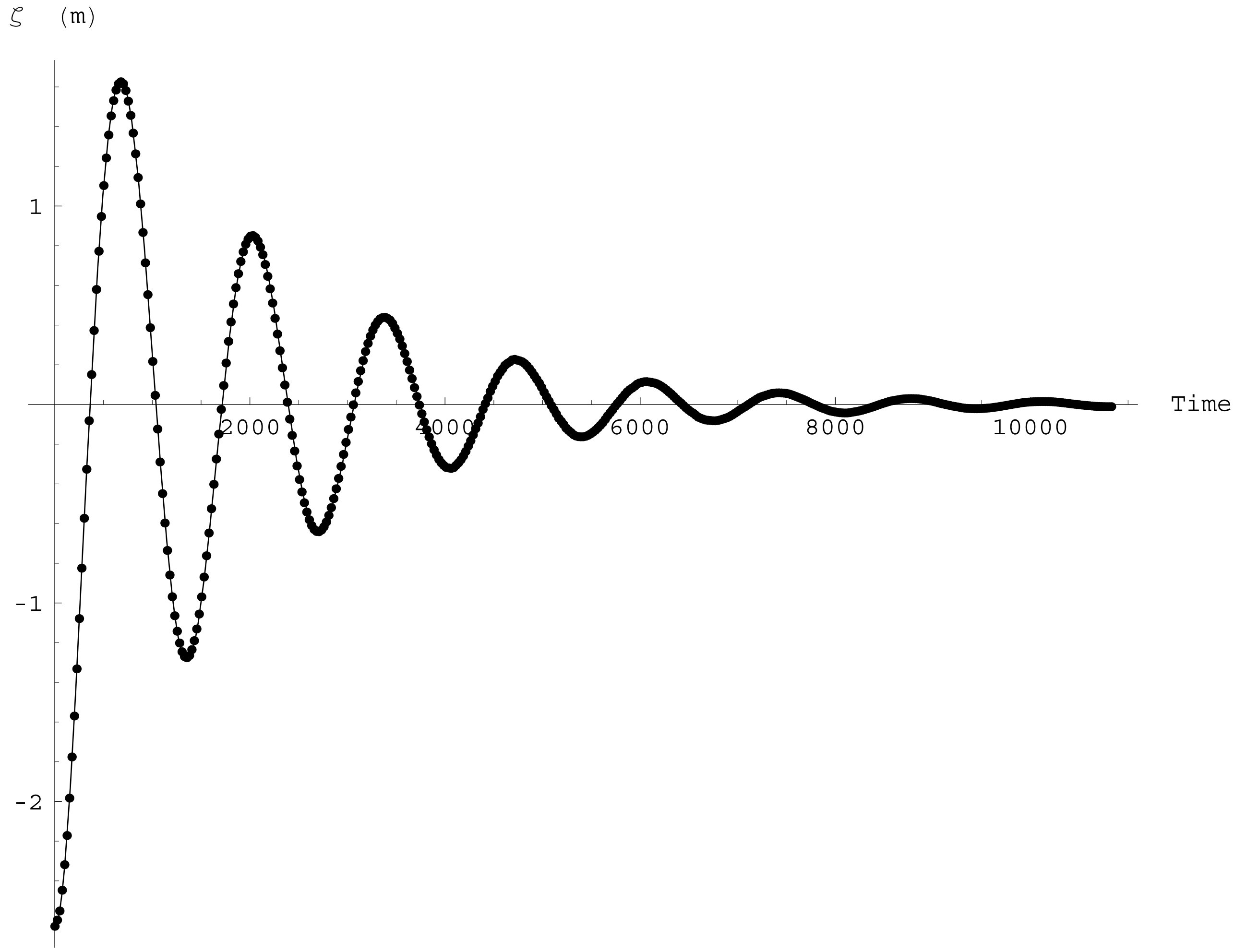

Figure 6.30: The numerical water level against time plotted together with the analytical zeta against time over eight periods at a node at $x = 2565$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

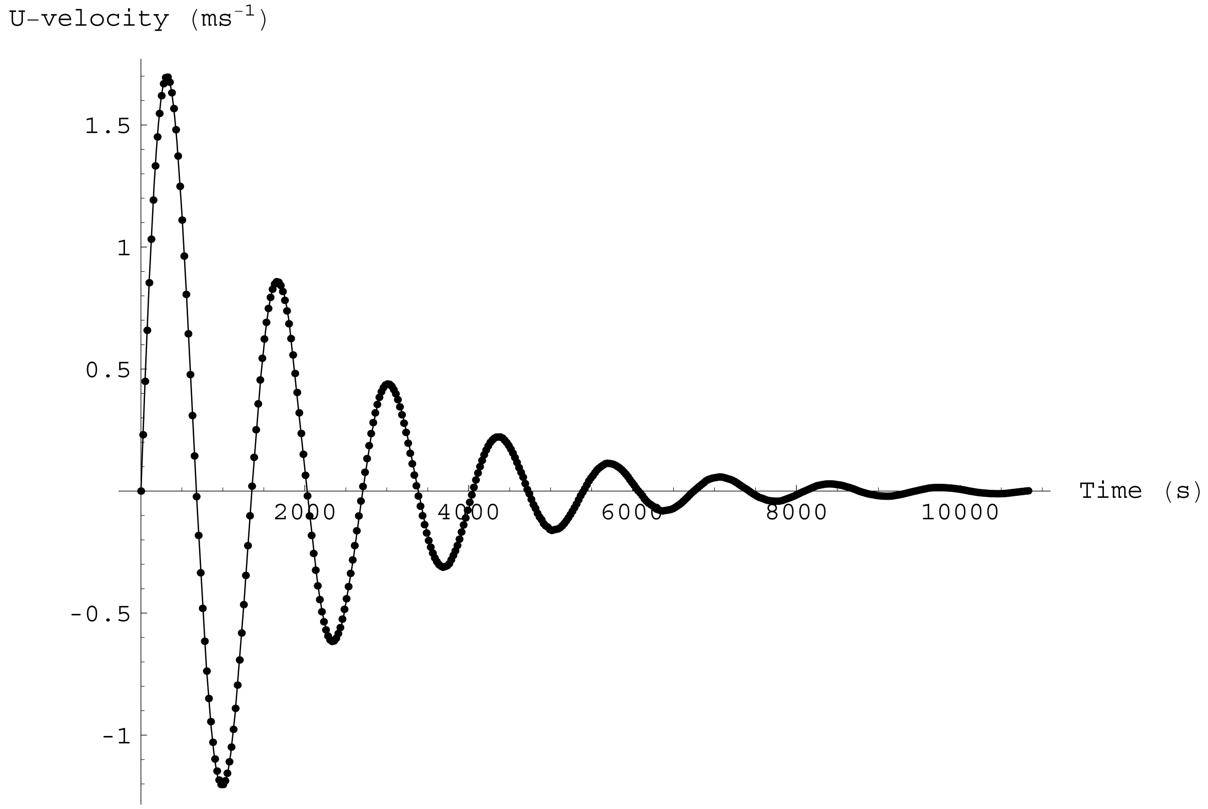

Figure 6.31: The numerical U-velocity against time plotted together with the analytical zeta against time over one period at a node at $x = 0$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

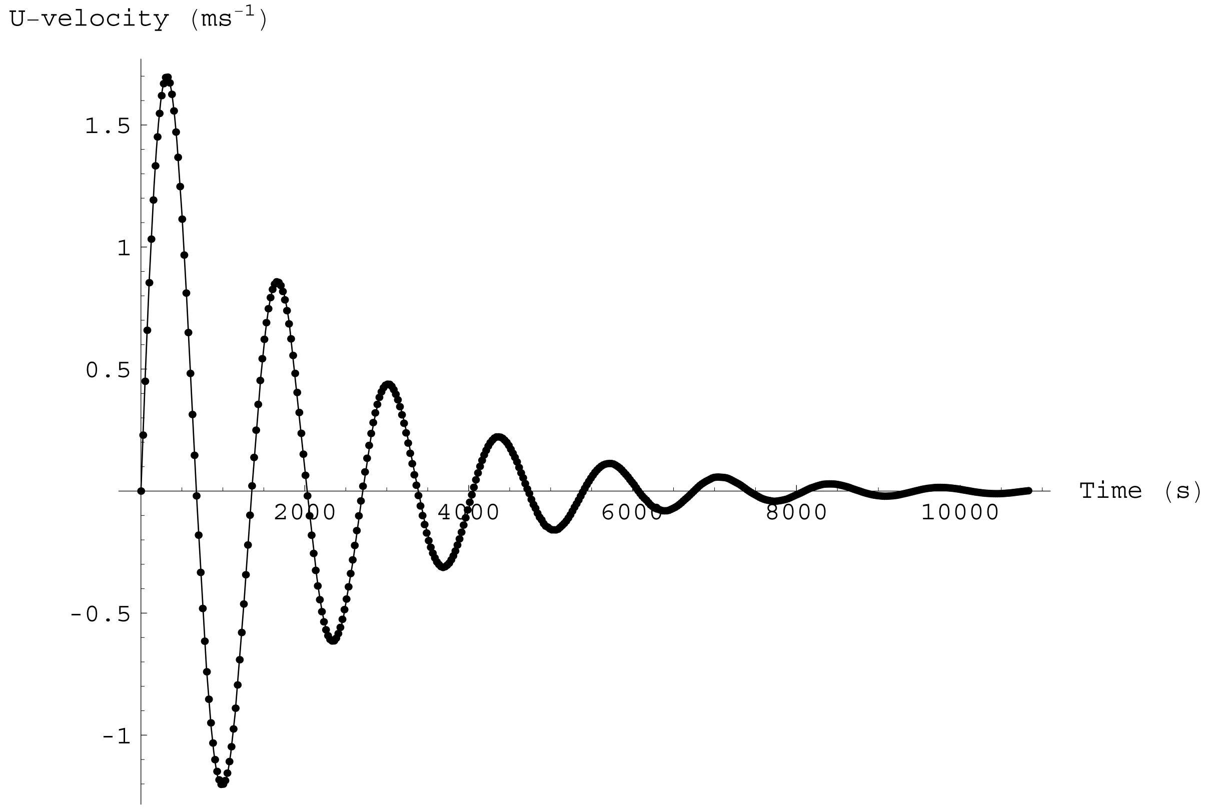

Figure 6.32: The numerical U-velocity against time plotted together with the analytical zeta against time over eight periods at a node at $x = 750$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

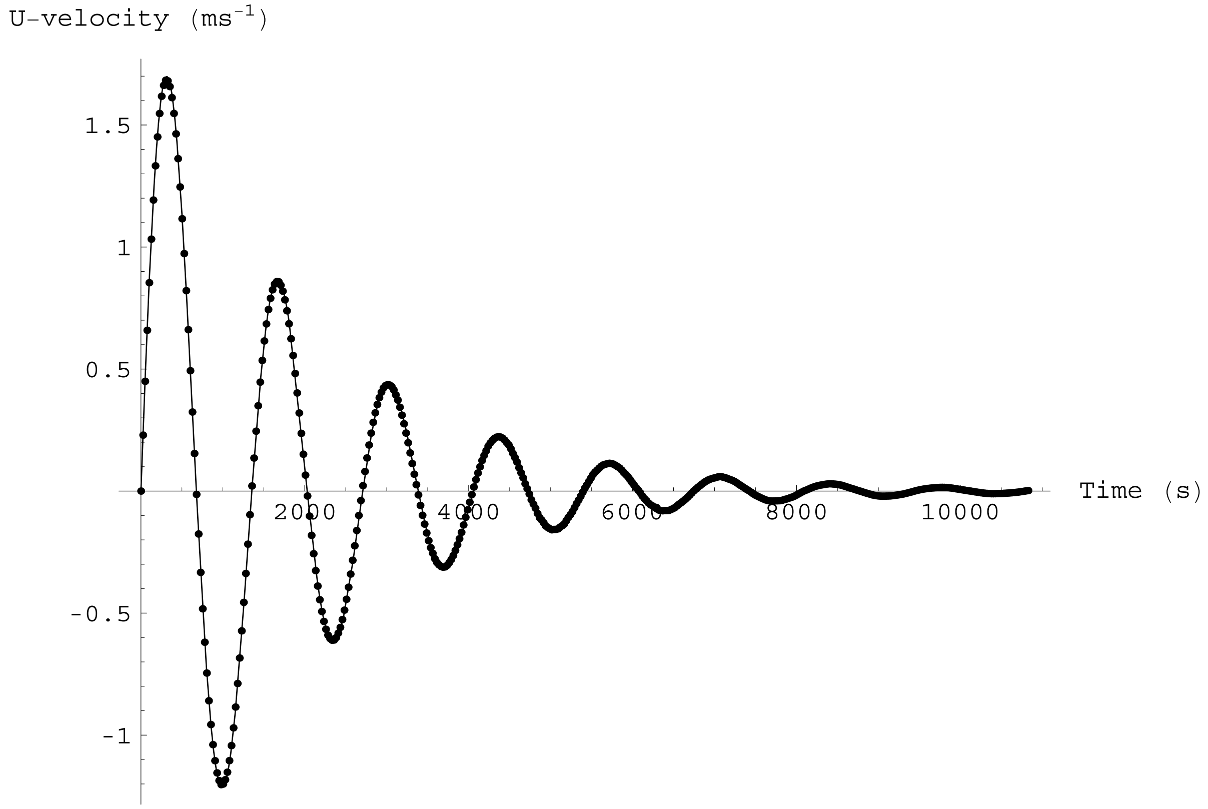

Figure 6.33: The numerical U-velocity against time plotted together with the analytical zeta against time over eight periods at a node at $x = 1500$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

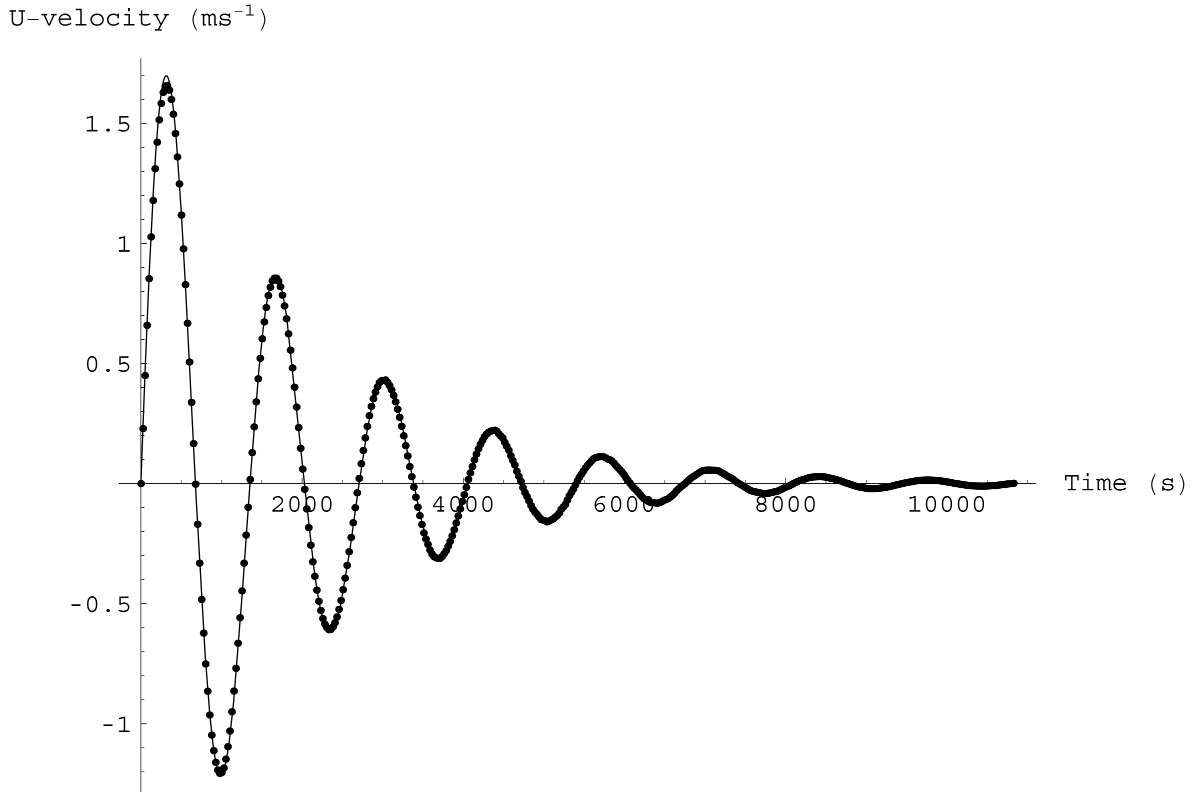

Figure 6.34: The numerical U-velocity against time plotted together with the analytical zeta against time over eight periods at a node at $x = 2250$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

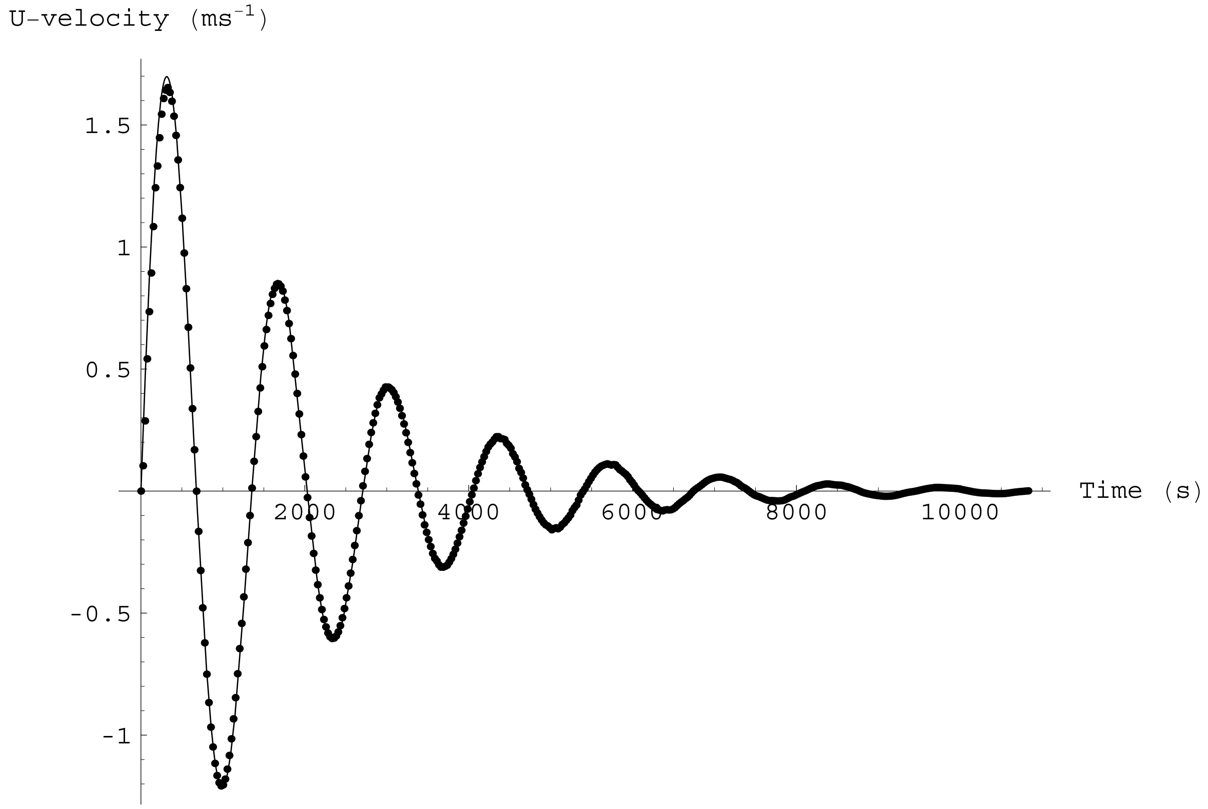

Figure 6.35: The numerical U-velocity against time plotted together with the analytical zeta against time over eight periods at a node at $x = 2565$ m and midway between the top and bottom boundary of the mesh, with the numerical plot a series of dots and the analytical plot a continuous line.

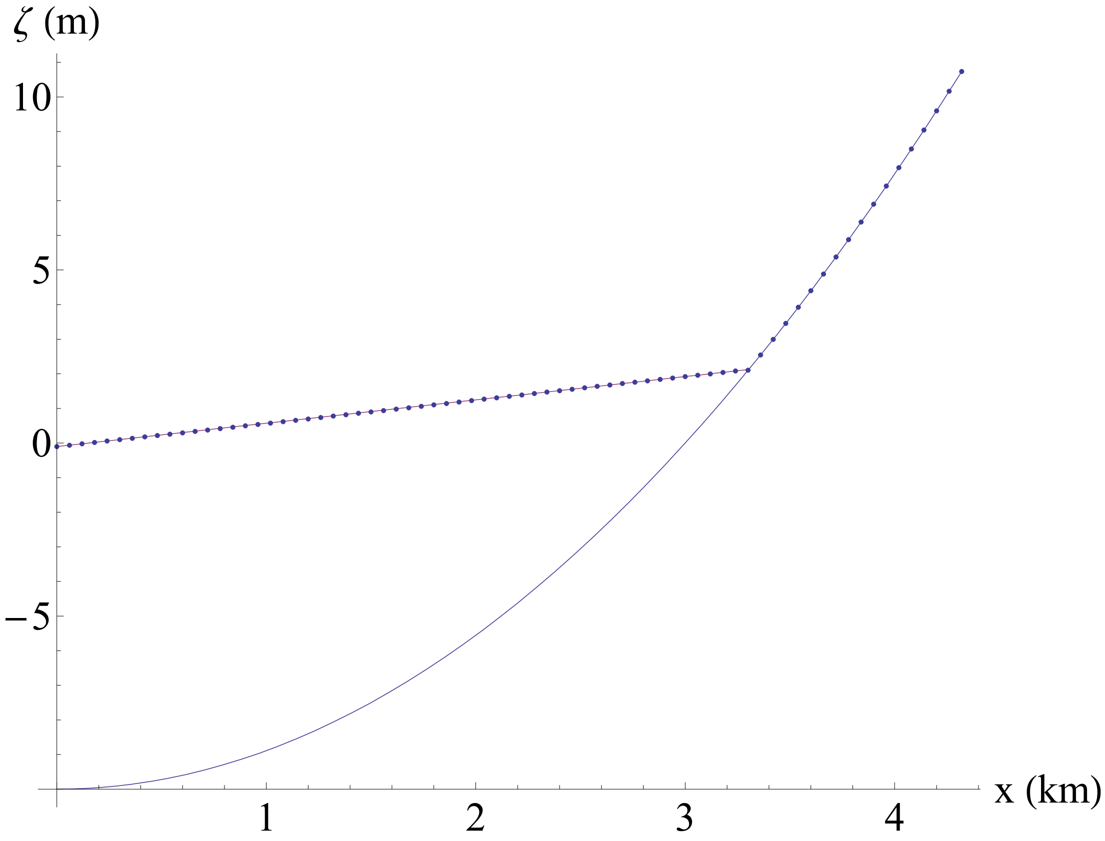

Figure 6.36: A comparison of the numerical and analytical values of the water surface at time $t = T/2$. The analytical solution is a continuous line whereas the numerical solution is a series of dots.

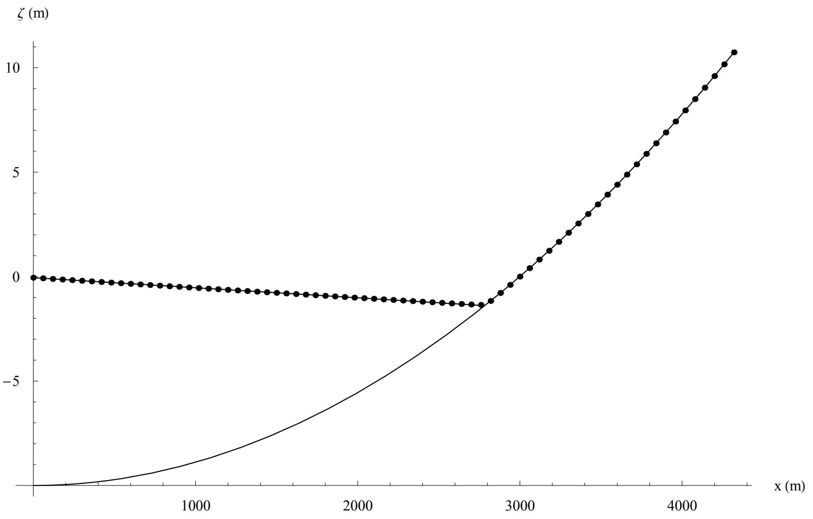

Figure 6.37: A comparison of the numerical and analytical values of the water surface at time $t = T$. The analytical solution is a continuous line whereas the numerical solution is a series of dots.

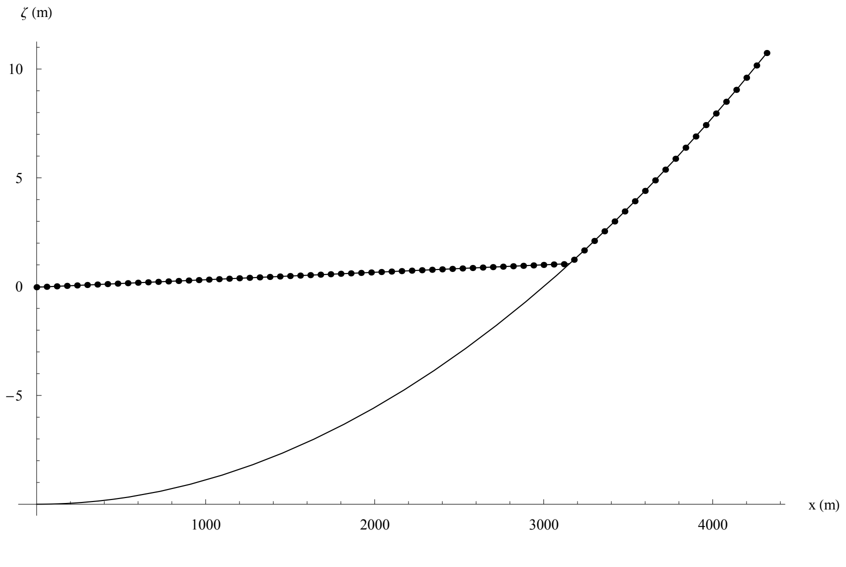

Figure 6.38: A comparison of the numerical and analytical values of the water surface at time $t = 3T/2$. The analytical solution is a continuous line whereas the numerical solution is a series of dots.

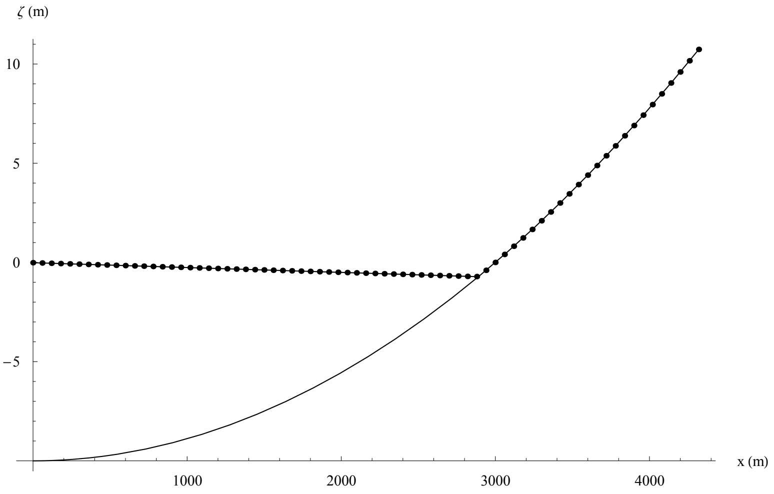

Figure 6.39: A comparison of the numerical and analytical values of the water surface at time $t = 2T$. The analytical solution is a continuous line whereas the numerical solution is a series of dots.

alytical solution was for the case of linear frictional one dimensional moving boundary shallow water wave flow with forcing which decays over time above a bed with quadratically varying depth. There is good agreement between the numerical and analytical solutions.

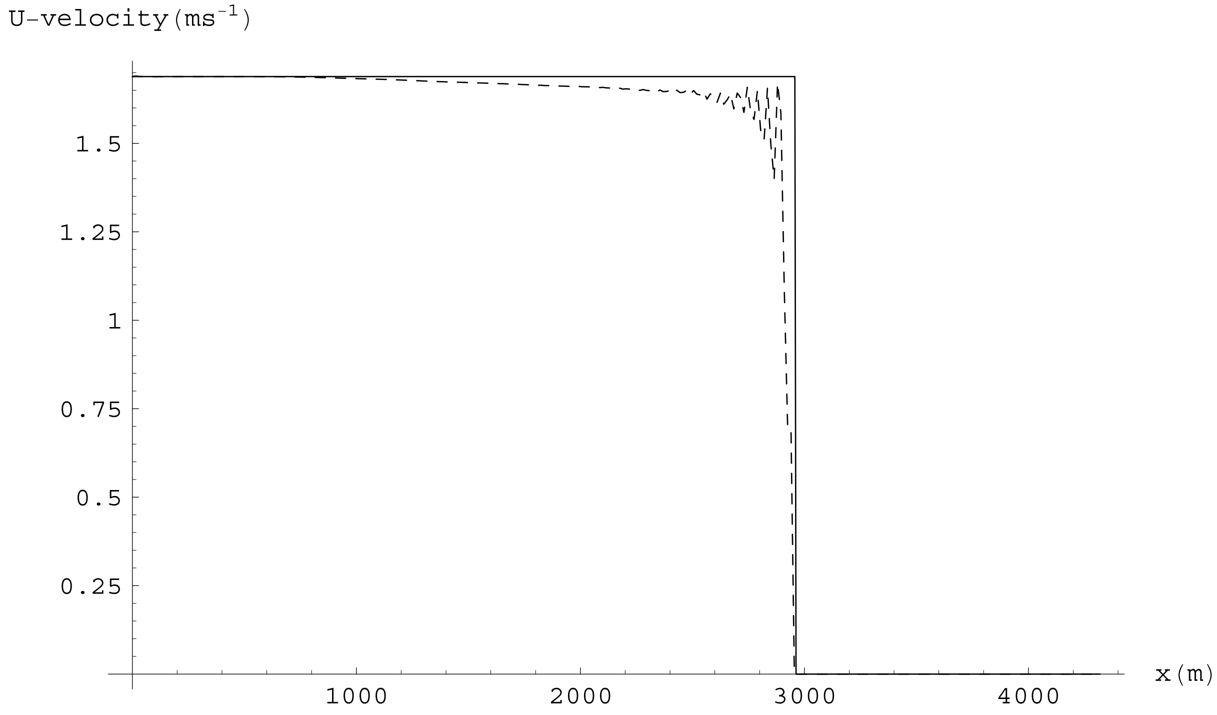

Figure 6.40: The analytical U-velocity against the numerical U-velocity at time $t = T/4$. The analytical solution is a continuous line whereas the numerical solution is a dashed line.

Chapter 7

Modelling the effect of proposed channel deepening on the tides in Port Phillip Bay

7.1 Introduction

At present, the guaranteed minimum depth in the main commercial shipping channels of Port Phillip Bay, Victoria, Australia, is 11.6 metres at all tides. The government of Victoria, in the late 1990s, announced plans to deepen the channels in Port Phillip Bay so that they can accommodate ships of up to 14 metres draft at all tides. This involves deepening sections of the Great Ship Channel at the Bay's entrance, the South Channel and channels going into Port Melbourne and Williamstown. In July 2004 the Port of Melbourne Corporation (which had assumed some of the duties of the former Victorian Channel Authority), the Victorian State government authority responsible for the channel deepening project, released the Environmental Effects Statement on the channel deepening project. Figure 7.1 shows details of the proposed channel deepening. The Victorian Government appointed an inde-

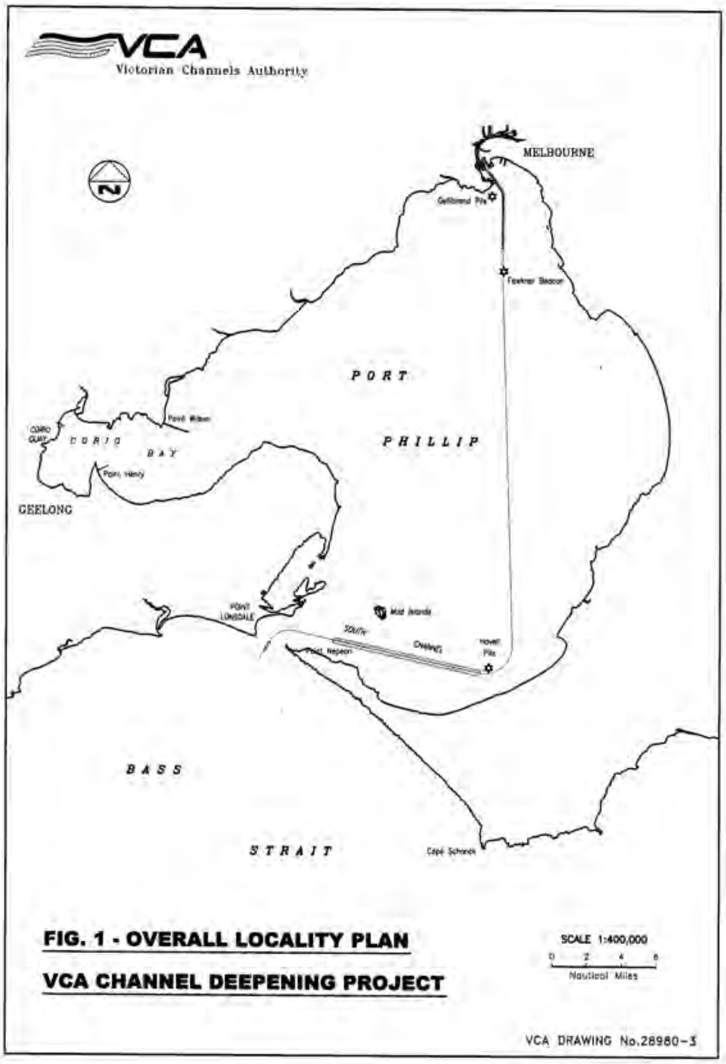

Figure 7.1: Overall Locality Plan Port Phillip Bay Channel Deepening Project

pendent panel to hold an enquiry on the Environmental Effects Statement on the channel deepening project in Port Phillip Bay beginning August 2004. Interested parties were invited to make a written submission to the panel. The author of this thesis, together with his Ph. D. supervisors, made a submission on that section of the Environmental Effects Statement dealing with the effect of the channel deepening on the astronomical tides, the periodic rise and fall of the sea due to the gravitational attraction of the Moon and Sun. This section of the Environmental Effects Statement was written by Lawson and Treloar [56, 57].

Lawson and Treloar modelled the astronomical tides in Phillip Bay. The levels of tides for the existing topography and for the topography including the proposed channel deepening were computed. Their results show that the effect of the proposed channel deepening on astronomical tidal levels in Port Phillip Bay will be to increase the maximum tidal height at most locations, with the greatest increase eight mm and the greatest reduction two mm.

The model used in this thesis gives results very close to those of Lawson and Treloar, with the maximum tidal height increasing at most locations, but with a greatest increase of seven mm and a greatest reduction of two mm.

The results at seven sites show that the model used in this thesis accurately models the existing tides, as does the model of Lawson and Treloar and that of Walker [109]. The results for the post-channel deepening, predicting the largest increase in maximum tidal height of seven mm, are very close to those of Lawson and Treloar, who predict eight mm.

7.2 Literature review

A number of papers have been published on modelling the tides in Port Phillip Bay, Victoria, Australia. These include papers by Easton [19], Black, Hatton and Rosenberg [9], Walker [109], Hubbert and McIntosh [37] and Lawson and Treloar [56, 57]. All of these papers used the shallow water equations except for Walker's, which used a three dimensional model. Port Phillip Bay has a very narrow entrance. Experimental measurements [56] show that the the current is fast in the vicinity of the entrance, of the order of 3.5 ms^{-1}, but very small for most of the bay and that the tidal height diminishes sharply from Point Lonsdale at the entrance to Queenscliff which is about 4 kms away inside the bay. The sea levels at the entrance are about 90 degrees out of phase with those just beyond the Sands region in the South of the bay. However, there is little change in the amplitudes of the constituents at locations North of Queenscliff. There are strong currents in the channels in the Sands region. In the northern part of the bay the tidal currents are negligible.

Easton [19] used a finite difference approach to model the M_2 tides in Corio Bay, situated at the end of the western arm of Port Phillip Bay. Waste is discharged into the bay by various industries on the coastline. A grid length of 375 m was chosen for both the x and y directions. An amplitude of 0.5 m was specified on the open sea boundary, which was some distance East of Corio Bay. A filter was used to minimise $2\delta x$ oscillations. An amplitude of 0.5 m was specified on the open sea boundary. the model showed that there was very little phase lag of the maximum tide. At a certain node the tide height was found to be out of phase with the velocity components by 90 degrees as is the case with a standing wave. The ebb and flood tides were almost identical except for their opposite directions. The maximum flow within Corio Bay is of the order of 0.1 ms^{-1}, which would result in a horizontal movement of about 1 km, much smaller than the length of the bay; hence, normal tidal flow conditions are insufficient to flush the bay.

Black, Hatton and Rosenberg [9] measured amplitudes and phases of the M_2, S_2, K_1 and O_1 tidal constituents at a point about 30 kms south of Pt. Lonsdale. Pt. Lonsdale is at the entrance to the bay. They modelled the tides with a finite difference scheme, using these constituents for forcing at a boundary in Bass Strait, mainly running at 20 km South of Pt. Lonsdale. They found accurate tidal levels at Geelong and Williamstown, at the North of the bay, and at Pt. Lonsdale except for the M_2 amplitude that was 30 % too large. They attributed this to an insufficiently fine grid. The results for the currents were not as good as those for water levels but were nevertheless moderately accurate.

Walker [109] used a three dimensional model to simulate the M_2, S_2, K_1, O_1 and N_2 tidal constituents in Port Phillip Bay. Because of experimental evidence that there are rapid spatial changes in tidal amplitudes in the neighbouring region Walker concluded that the Pt. Lonsdale sea level data was not suitable for forcing his model. Walker used modified values of the Lorne data for forcing. Lorne is about 60 kms South West of Pt. Lonsdale. Comparison of modelled water levels at five locations showed close agreement with experimental data. Similarly, comparison of modelled water currents at five locations showed close agreement with experimental data.

Hubbert and McIntosh [37] produced a moving boundary scheme to model coastal flooding due to storm surges in Port Hedland, Western Australia, and in Port Phillip Bay. In the scheme the wetting and drying depends not only on the sea surface height relative to the adjacent topography but also on the distance travelled by the coastal interface based on the current immediately seaward of the boundary. This means that the inclusion or removal of a grid point may take more than a single time step to accomplish even though the height criterion may be satisfied. The scheme was used to model the storm surge at Port Hedland, Western Australia, 1939, caused by a tropical cyclone. The surface pressure and wind fields during the cyclone, required to drive the storm surge model, were reconstructed by fitting an analytical tropical cyclone model to available weather observations in the region. The modelled peak sea level during the storm surge was close to the measured peak sea level. The scheme was also used to model storm surges that occurred in Port Phillip Bay in 1994. The graph of observed and modelled sea levels at one location over a 60 hour period shows the two curves to be fairly close to each other.

Lawson and Treloar [56, 57] modelled the M_2, S_2, K_1, O_1 and N_2 tidal constituents in Port Phillip Bay. Lawson and Treloar used a curvilinear grid ranging in size from 30-35 m at the entrance to 300-500 m in the North. Lawson and Treloar's scheme is finite difference both in time and space. Lawson and Treloar's model area included all of Port Phillip Bay and extended 15 km offshore into Bass Strait. Lawson and Treloar [57] found that all tidal constituents in the region of their open sea boundary in Bass Strait had similar values to those at Lorne, which is about 60 kms. south west of Pt. Lonsdale, with the exception of the M_2 constituent which was about 10 % larger. Lawson and Treloar used modified values of the Lorne data for forcing. Lawson and Treloar's values for the tidal heights at six locations were found to be close to experimental values. Lawson and Treloar's values for the tidal currents at six locations were found to be moderately close to experimental values. Lawson and Treloar modelled the change in tidal levels at five locations, obtaining a maximum change in level of 8 mm. Lawson and Treloar modelled the change in tidal currents at five locations, obtaining small changes.

7.3 The model

In this thesis the existing astronomical tides and the astronomical tides with the proposed channel deepening in Port Phillip Bay have been modelled, using numerical solutions of the two-dimensional depth-averaged shallow water wave equations [108]. The results of this study are compared in Table 7.4 with the results of Lawson and Treloar [56, 57].

In this thesis a different numerical scheme to that of Lawson and Treloar has been used, with slightly different boundary conditions and a different mesh. The grid used in this thesis contains 2618 triangular elements and 1429 nodes, with a fine grid near the entrance to the bay. Triangle sides range from 300 m in the entrance to 2-2.5 km in the North of the bay. The grid, which was designed by hand and checked using computer programs, is shown in Figure 7.2. The scheme used in this thesis is finite element in space and finite difference in time and explicit. The model area used in this thesis included all of Port Phillip Bay apart from Swan Bay and extended into Bass Strait, a region of water south of Port Phillip Bay, a distance ranging from 10 to 25 km.

The two-dimensional depth-averaged shallow water wave equations used are based on equations (2.1), (2.2) and (2.3) with the assumption that the wind is negligible. The equations used consist of the conservation of momentum equation in the East direction

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} - \nu\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) + \frac{gn^2 U\sqrt{U^2+V^2}}{(h+\zeta)^{\frac{4}{3}}} + g\frac{\partial \zeta}{\partial x} = 0, \quad (7.1)$$

the conservation of momentum equation in the North direction

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} - \nu\left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2}\right) + \frac{gn^2 V\sqrt{U^2+V^2}}{(h+\zeta)^{\frac{4}{3}}} + g\frac{\partial \zeta}{\partial y} = 0, \quad (7.2)$$

and the conservation of mass equation

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (h+\zeta)U}{\partial x} + \frac{\partial (h+\zeta)V}{\partial y} = 0. \quad (7.3)$$

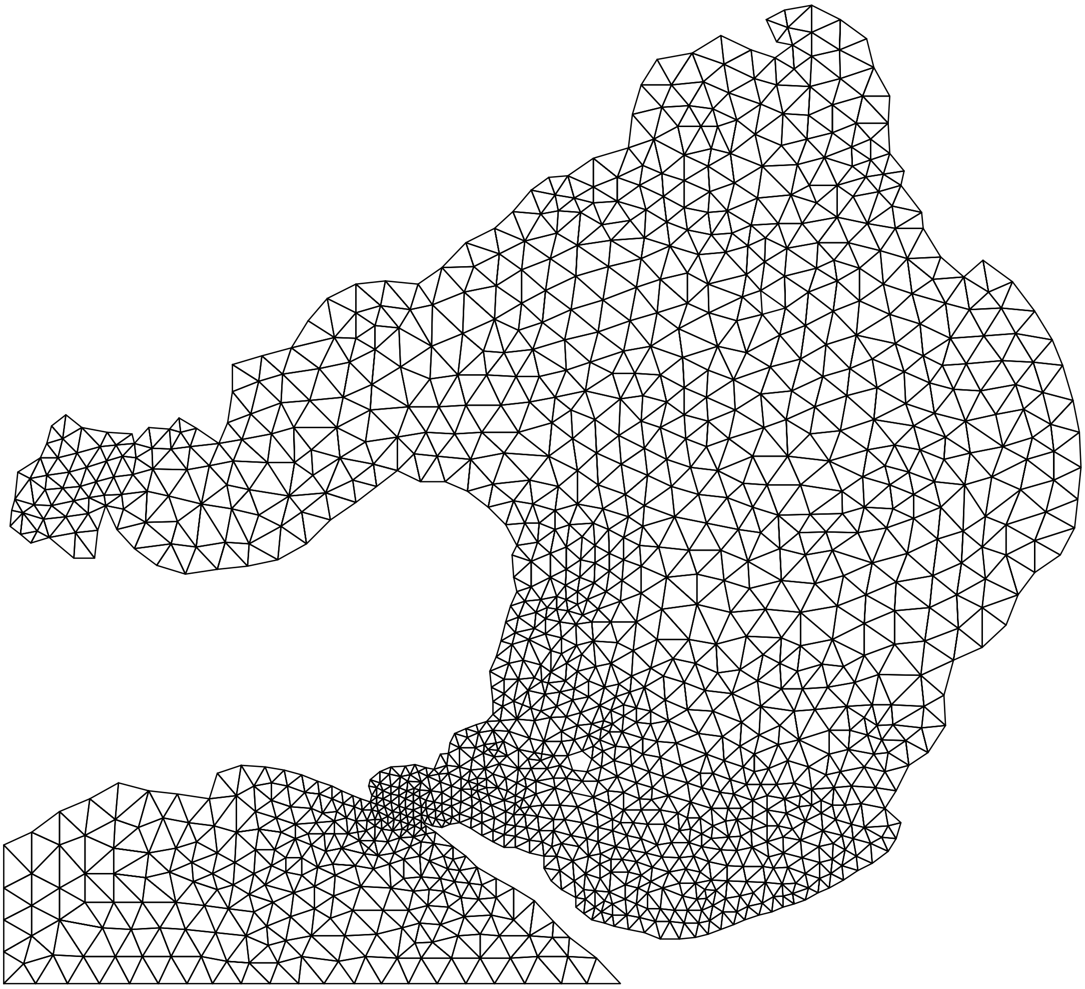

Figure 7.2: Finite element mesh in Port Phillip Bay.

The initial conditions used in the model in this chapter are $\zeta = 0$, $U = 0$ and $V = 0$ at time $t = 0$. At land boundaries, the flow normal to the coastline is zero. At the sea boundary, which consists of two straight line segments,shown in the bottom left hand corner of Figure 7.3, the sea surface elevation, ζ_b, is specified for all time as the sum of five tidal constituents (M_2, S_2, K_1, O_1 and N_2), i.e.

$$\zeta_b = \sum_{j=1}^{5} a_j \cos(\omega_j t - \gamma_j), \tag{7.4}$$

where, for tidal constituent j, a_j is the amplitude, γ_j is the phase, and ω_j is the angular frequency. The tidal constituents are shown in Table 7.1. Using similar forcing data for the same constituents as did Lawson and Treloar gave accurate values for amplitudes and phases of tidal levels at five points inside the bay but gave M_2 amplitudes about 50 % larger than the observed at Queenscliff and Pt. Lonsdale but accurate phases. Perhaps this is because the triangles used at the entrance are not small enough to represent the tidal motion. It was found that if amplitudes and phases of the constituents at Pt. Lonsdale were used for tidal forcing at the boundary in Bass Strait that accurate results were obtained at all seven locations inside the bay for which there are observed values. Then the values of these amplitudes and phases were modified to minimise the errors in the modelled data at the seven locations. Five tidal constituents only were used as these five constituents measure about 75 % of the tidal variation [78, 79], thus giving an accurate estimate of the total change that channel deepening will cause.

The equations were solved numerically using the Selective Lumped Mass scheme [46]. A computer program was written in Visual C++ to implement the scheme. This computer program is listed in Appendix A. The scheme solves for ζ, U and V at each time step, δt. The scheme involves a selective lumping parameter, s_r. The Selective Lumped Mass scheme is restricted by

Table 7.1: Tidal Constituents

Name of constituent	Period (in hours)
M2	12.42
S2	12.00
K1	23.93
O1	25.82
N2	12.66

a stability requirement [29],

$$\delta t \leq d_m \delta x / \sqrt{gh}, \tag{7.5}$$

where δx is the smallest space step, and d_m is a function of s_r. The time step used was 13.5 seconds while s_r was chosen to be 0.98.

The modelled results for the amplitudes and phases of the tidal constituents at seven locations were compared with the observed results. The data for the observed amplitudes and phases of the tide are obtained from the Port of Melbourne Corporation Tables [78, 79]. The most accurate values were found when n was chosen to be 0.020 $\mathrm{m}^{-\frac{1}{3}}\mathrm{s}^{-1}$ and ν to be 6 $\mathrm{m}^2\mathrm{s}^{-1}$. The latter value was used to minimise node-to-node oscillations.

7.4 Existing data

The results in this study for the amplitude and phase of the tide at seven locations (Pt. Lonsdale, Queenscliff, Hovell Pile, West Channel Pile, Pt. Richards Channel No. 1, Geelong and Williamstown) were obtained by running the model for a simulated 32 days and analysing the data over the last 29 days using a FFT (Fast Fourier Transform) package, and compared with the observed data and at five locations compared with those of Lawson and

Treloar [56] and Walker [109]. As there were transients in the data for the first three days the data for this time was not used. It can be shown [18] that 29 days is the minimum period over which data, which is hourly or a divisor of an hour, can be analysed to obtain accurate results for the five tidal constituents. The results of the FFT analysis are shown in Table 7.2 and in three cases compared with those of Lawson and Treloar [56] and of Walker [109]. The data for West Channel Pile, Hovell Pile and Pt. Richards Channel No. 1 are not shown due to lack of space. The amplitude difference is the observed value minus the modelled value. The vector difference is obtained by regarding each constituent as a vector, with its magnitude equal to the amplitude of the constituent and its phase the phase of the constituent. The vector difference is the magnitude of the vector joining the ends of the modelled vector and the observed vector. From the results shown in Table 7.2, it can be seen that the results are close to those of Lawson and Treloar and those of Walker. Perhaps the results would have been better if more constituents were used. As the modelled values for the existing tide are close to those of the observed values, the prediction on the effects of the astronomical tides of post-channel deepening should be accurate. The model is neither inferior nor superior to the other two. The results for modelled tidal currents at three locations (labelled S (in Symonds Channel), A (in South Channel) and T (in Portsea Channel)) in comparison with the modelled values of Black Hatton and Rosenberg [9] and Lawson and Treloar [56] and the observed values of Black, Hatton and Rosenberg [9] are shown in Table 7.3. It can be seen that the modelled results for tidal current are of the same order as Lawson and Treloar's but not as accurate of those of Black, Hatton and Rosenberg.

The bathymetry together with the ten locations are shown in Figure 7.3.

The observed and modelled water level as a function of time at Geelong

Table 7.2: Comparison of tidal constituents between models and measurements

Tidal constituent	Amplitude (m) observed	Amplitude difference (m)			Vector difference (m)		
		This study	LT	W	This study	LT	W
Geelong							
M_2	0.268	0.000	-0.003	0.001	0.022	0.003	0.015
S_2	0.064	0.003	0.001	-0.003	0.008	0.004	0.003
N_2	0.046	-0.003	0.006	0.002	0.009	0.007	0.002
K_1	0.099	-0.007	-0.007	-0.012	0.012	0.012	0.013
O_1	0.070	-0.011	-0.008	-0.006	0.003	0.006	0.006
Queenscliff							
M_2	0.258	-0.006	0.013	-0.033	0.016	0.050	0.047
S_2	0.070	0.000	0.002	-0.015	0.005	0.015	0.028
N_2	0.049	-0.005	0.008	0.000	0.008	0.011	0.006
K_1	0.109	-0.004	0.004	-0.001	0.009	0.005	0.028
O_1	0.082	0.011	-0.007	-0.001	0.004	0.007	0.019
Pt. Lonsdale							
M_2	0.440	0.002			0.039		
S2	0.127	0.011			0.011		
N_2	0.088	-0.003			0.004		
K_1	0.144	0.002			0.004		
O_1	0.104	-0.001			0.002		
Williamstown							
M_2	0.234	-0.006	0.002	0.007	0.016	0.004	0.012
S2	0.054	0.000	0.001	-0.003	0.006	0.003	0.005
N_2	0.041	-0.003	0.006	-0.001	0.005	0.007	0.002
K_1	0.097	-0.006	-0.004	-0.008	0.010	0.004	0.009
O_1	0.068	-0.011	-0.007	-0.006	0.011	0.011	0.006

Table 7.3: Comparison of tidal currents between models and measurements. No data is available for observed N_2 currents.

Tidal constituent	Tidal ellipse, major semi-axes observed (cm s^{-1})	Tidal ellipse, major semi-axes modelled (cm s^{-1})			Direction (degrees) observed	Direction (degrees) modelled	
		This study	LT	BHR		This study	BHR
Symonds Channel							
M_2	83.5	72.1	89.1	89.5	49	47	46
S_2	26.1	17.2	20.6	29.9	49	47	46
N_2	n.a.	13.0	12.9	n.a.	n.a.	47	n.a.
K_1	13.6	16.0	18.9	49.3	48	47	46
O_1	12.9	11.5	12.3	15.3	49	47	44
South Channel							
M_2	66.5	55.2	83.6	78.3	171	165	179
S_2	20.8	13.4	20.6	26.8	171	165	179
N_2	n.a.	10.1	12.6	n.a.	n.a.	165	n.a.
K_1	12.1	14.1	22.7	48.6	174	164	179
O_1	11.5	10.3	17.7	15.7	173	164	175
Portsea Channel							
M_2	53.0	41.8	62.3	52.0	161	179	179
S_2	13.0	10.0	14.2	17.4	162	179	179
N_2	n.a.	7.6	8.4	n.a.	n.a.	179	n.a.
K_1	8.7	9.9	13.0	29.2	163	179	0
O_1	6.9	7.1	10.6	8.4	161	0	1

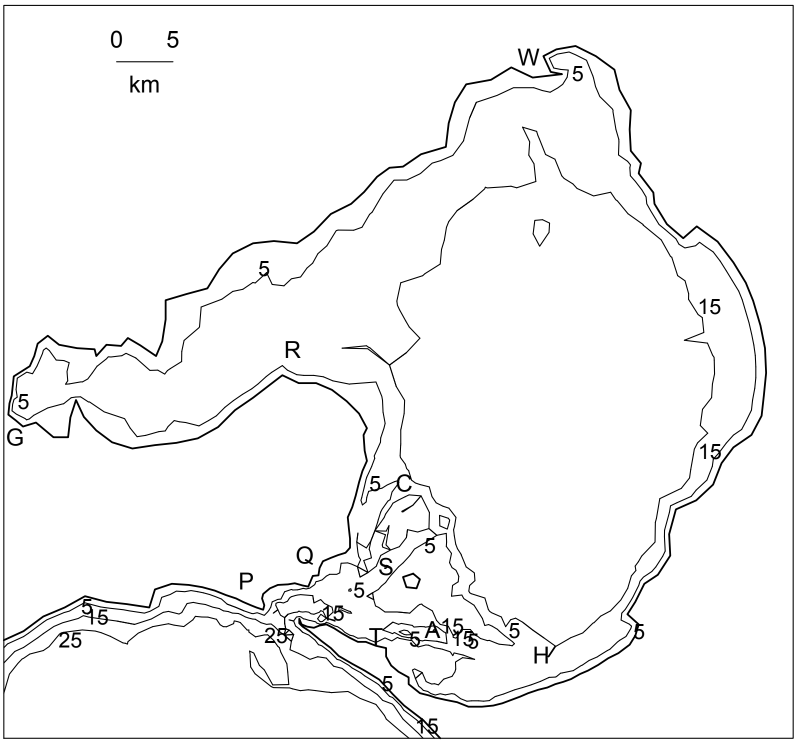

Figure 7.3: Depth contours (m) and locations in Port Phillip Bay. Locations are indicated by letters; P is for Pt. Lonsdale, R for Pt. Richards Channel No. 1, Q for Queenscliff, H for Hovell Pile, C for West Channel Pile, G for Geelong, W for Williamstown, A is in the South Channel, S is in the Symonds Channel and T is in the Portsea Channel.

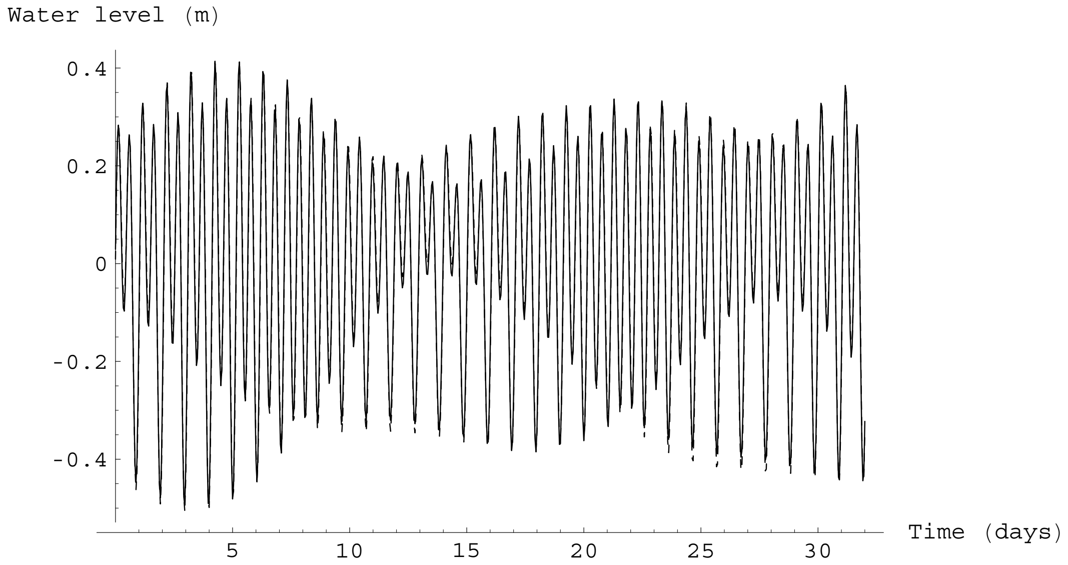

Figure 7.4: Water level (m) as a function of time at Geelong over 32 days. Observed data is a continuous line while modelled data is a dashed line.

are graphed together over 32 days in Figure 7.4, at Hovells Pile in Figure 7.5, at Pt Lonsdale in Figure 7.6, at Pt Richards in Figure 7.7, at Queenscliff in Figure 7.8, at West Channel Pile in Figure 7.9, and at Williamstown in Figure 7.10; it can be seen from the graphs that at all locations that the modelled and observed water levels are very close.

The contour lines for the amplitudes of the modelled M_2 water level, ζ, are shown in Figure 7.11; it can be seen that the amplitude decreases substantially a short distance inside the entrance. The contour lines for the phases of the modelled M_2 water level are shown in Figure 7.12. The contour lines for for K_1 water level amplitudes are shown in Figure 7.13 and phases in Figure 7.14, . Similar results were obtained for the N_2, O_1 and S_2 water levels.

The contour lines for amplitudes of the modelled M_2 U-velocity component are shown in Figure 7.15 and for the phases of the modelled M_2

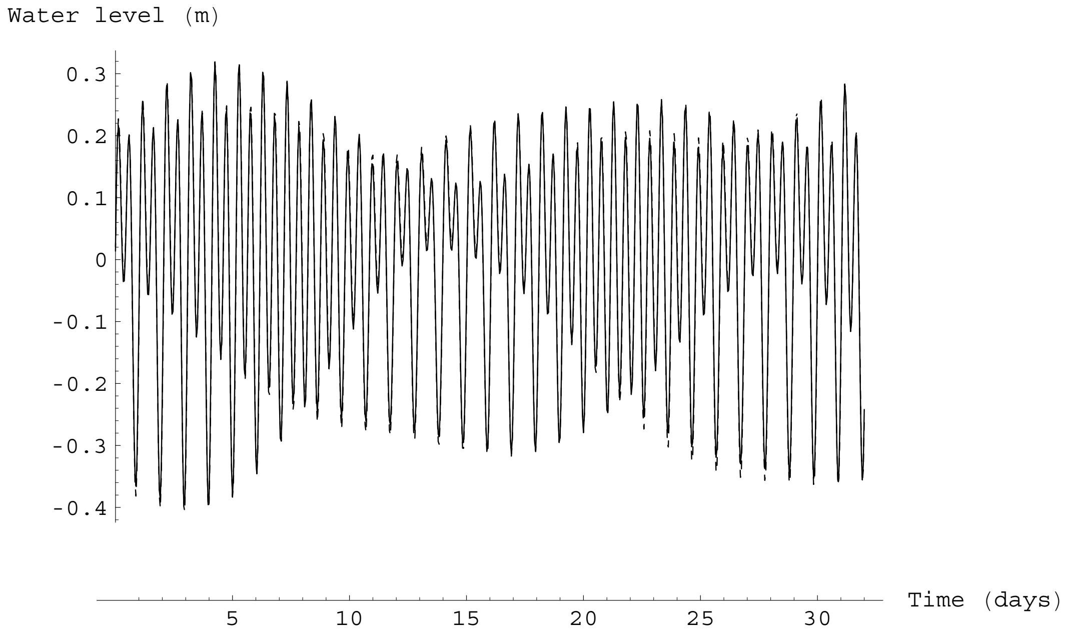

Figure 7.5: Water level (m) as a function of time at Hovell's Pile over 32 days. Observed data is a continuous line while modelled data is a dashed line.

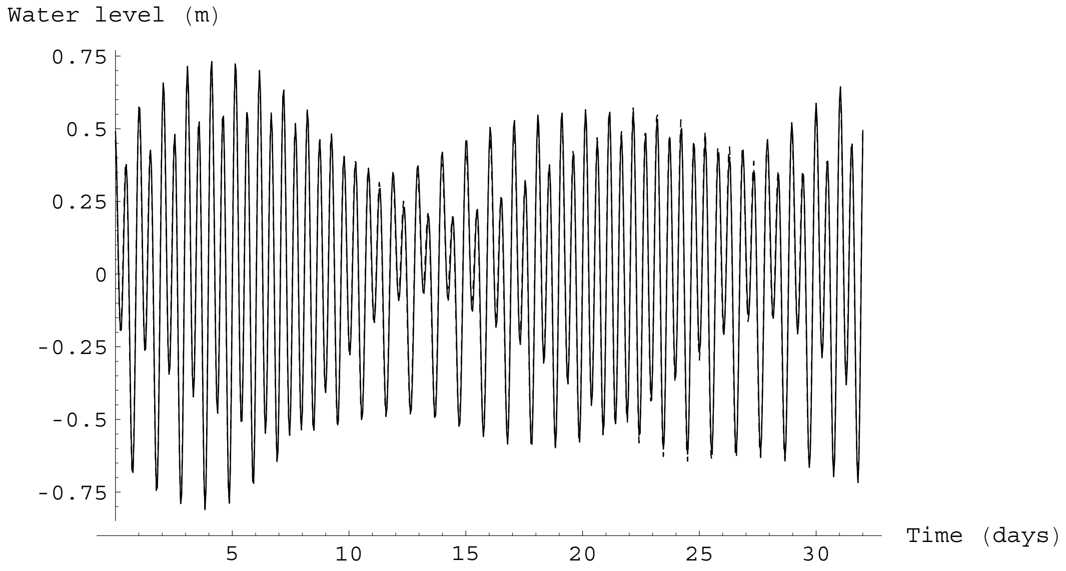

Figure 7.6: Water level (m) as a function of time at Pt. Lonsdale over 32 days. Observed data is a continuous line while modelled data is a dashed line.

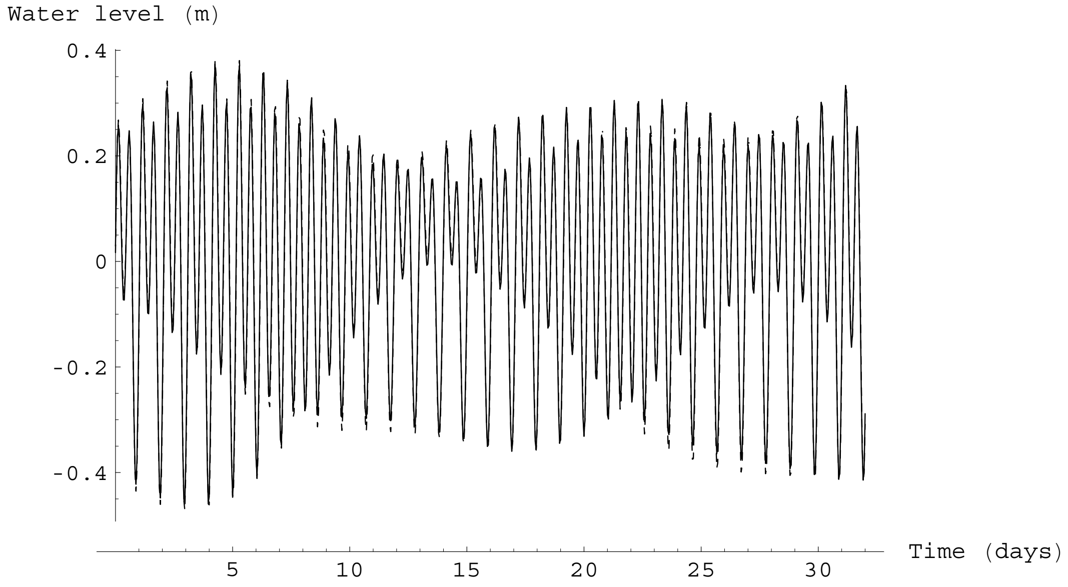

Figure 7.7: Water level (m) as a function of time at Pt. Richards Channel No. 1 over 32 days. Observed data is a continuous line while modelled data is a dashed line.

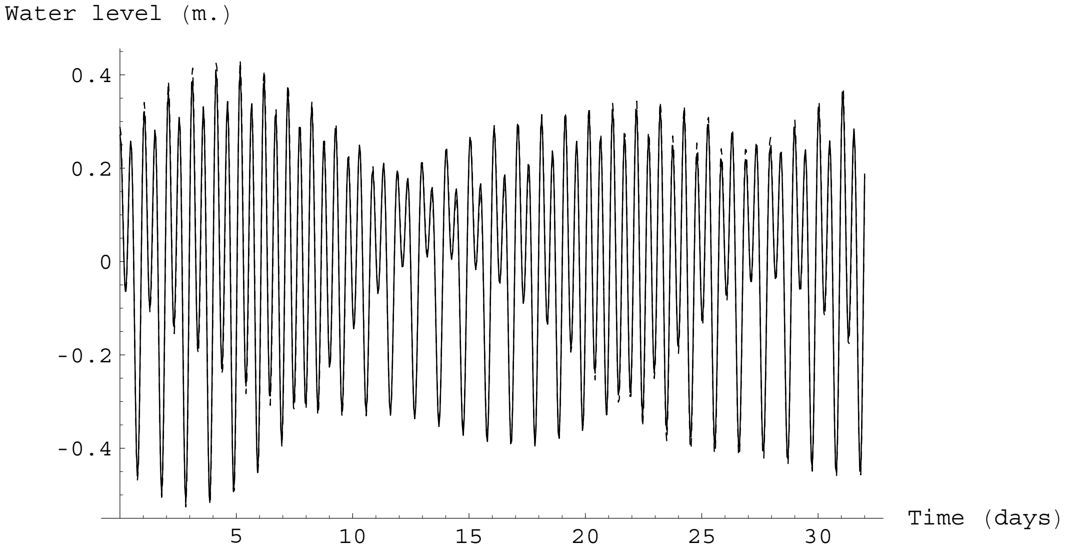

Figure 7.8: Water level (m) as a function of time at Queenscliff over 32 days. Observed data is a continuous line while modelled data is a dashed line.

constituent are shown in Figure 7.16. Similar results were obtained for the U-velocity components of the other tidal constituents, K_1, N_2, O_1 and S_2.

The contour lines for amplitudes of the modelled M_2 V-velocity component are shown in Figure 7.17. Similar contour lines were obtained for the V-velocity components of K_1, N_2, O_1 and S_2. The velocity vectors at a time of high tide at Point Lonsdale are shown in figure 7.18. The velocity vectors at a time of low tide at Point Lonsdale are shown in Figure 7.19. The contour lines for the water level heights at a time of high tide at Point Lonsdale are shown in Figure 7.20, while the water level heights at a time of low tide at Point Lonsdale are shown in Figure 7.21. The M_2 velocity ellipses are shown in Figure 7.22.

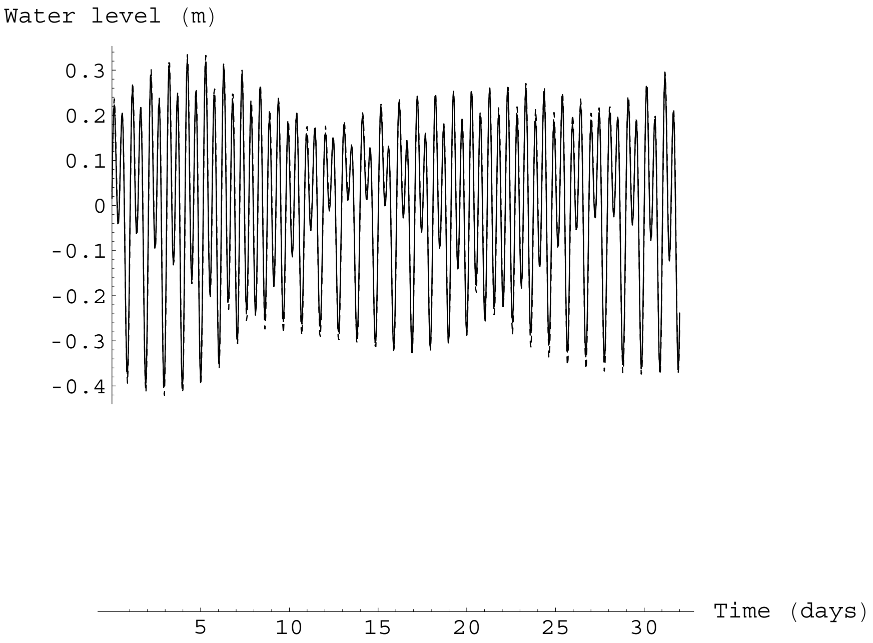

Figure 7.9: Water level (m) as a function of time at West Channel Pile over 32 days. Observed data is a continuous line while modelled data is a dashed line.

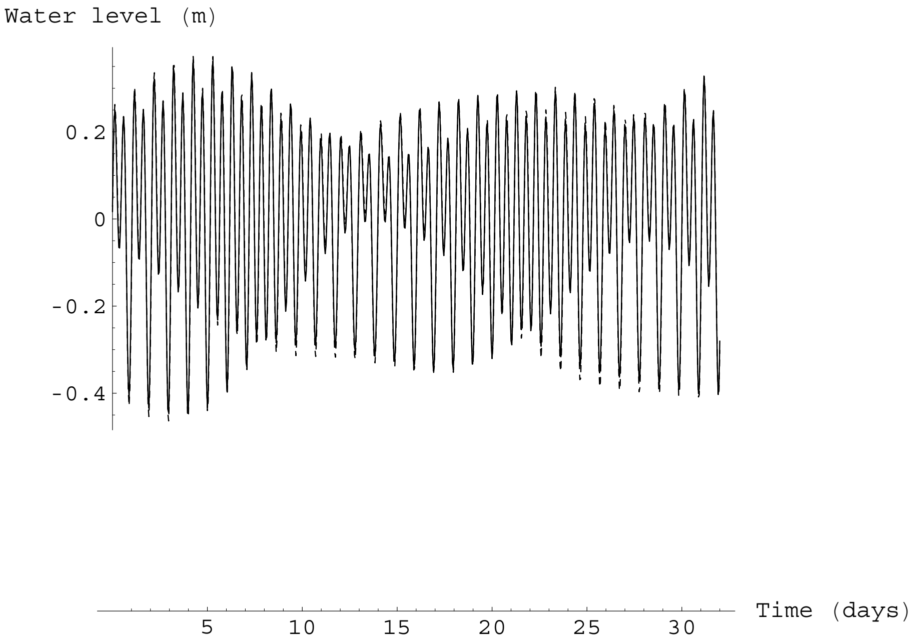

Figure 7.10: Water level (m) as a function of time at Williamstown over 32 days. Observed data is a continuous line while modelled data is a dashed line.

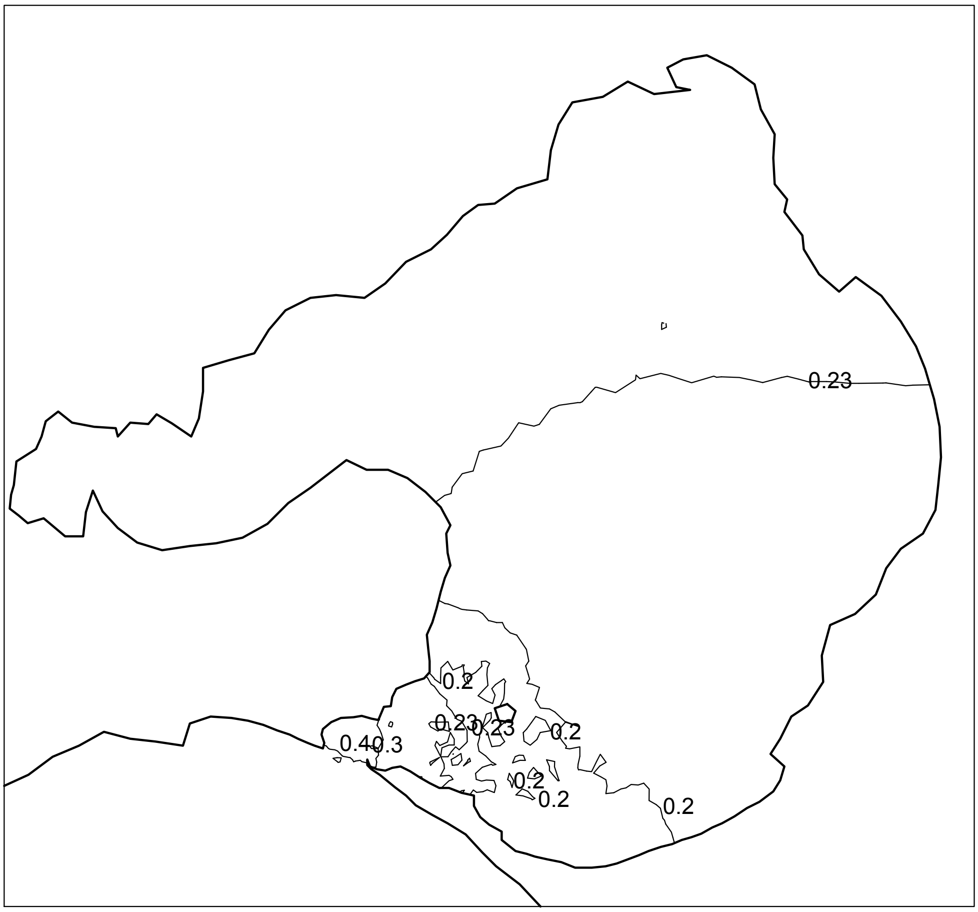

Figure 7.11: The contours of the amplitudes (m) of the modelled M_2 water level component.

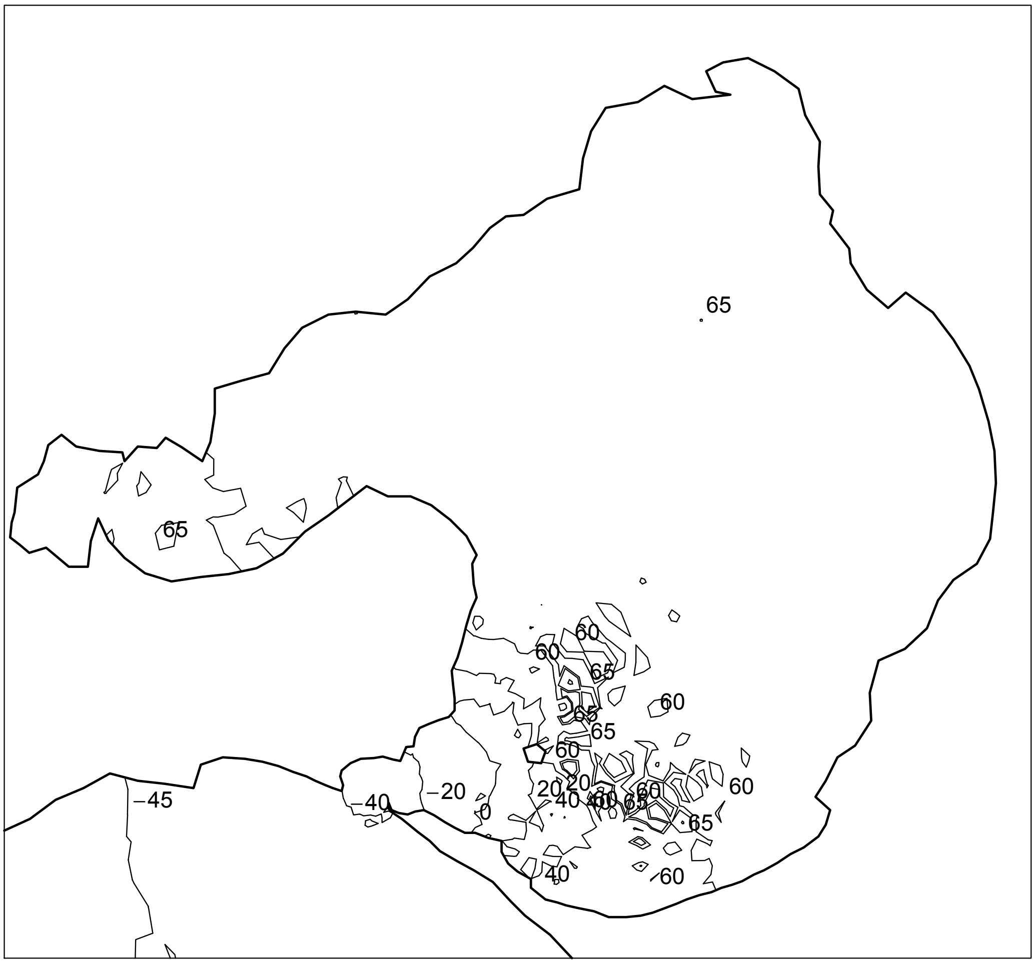

Figure 7.12: The contours of the phases (degrees) of the modelled M_2 water level component.

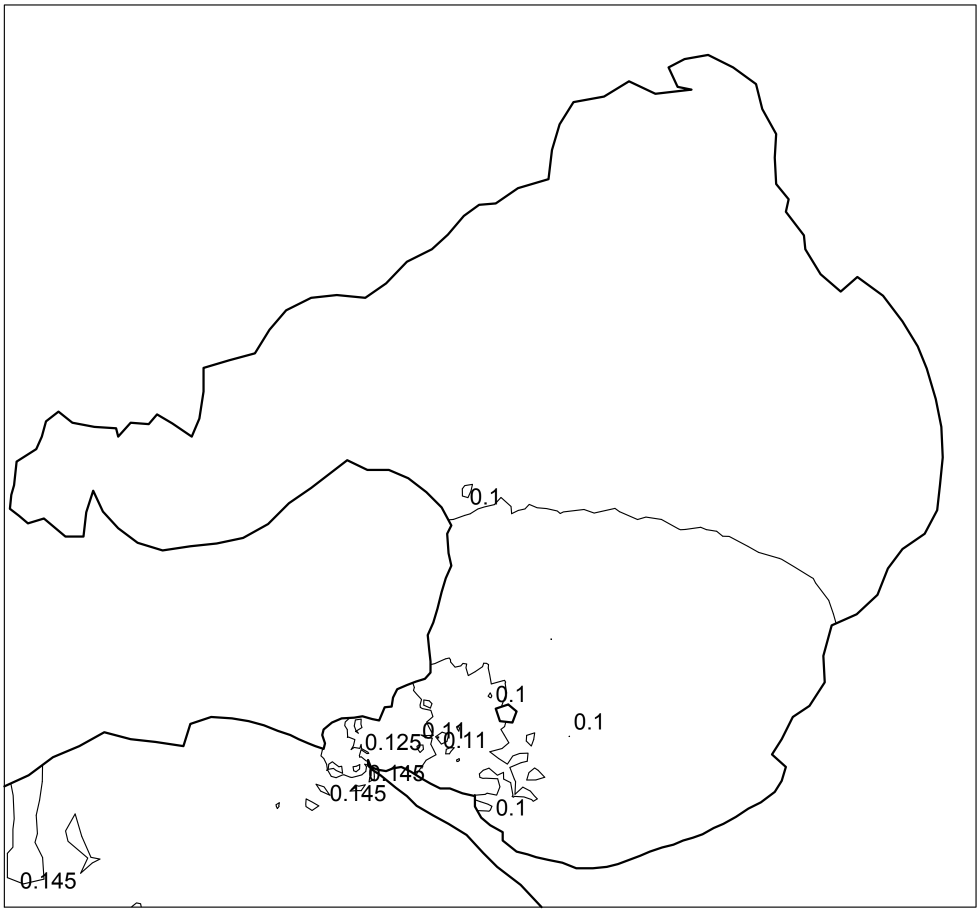

Figure 7.13: The contours of the amplitudes (m) of the modelled K_1 water level component.

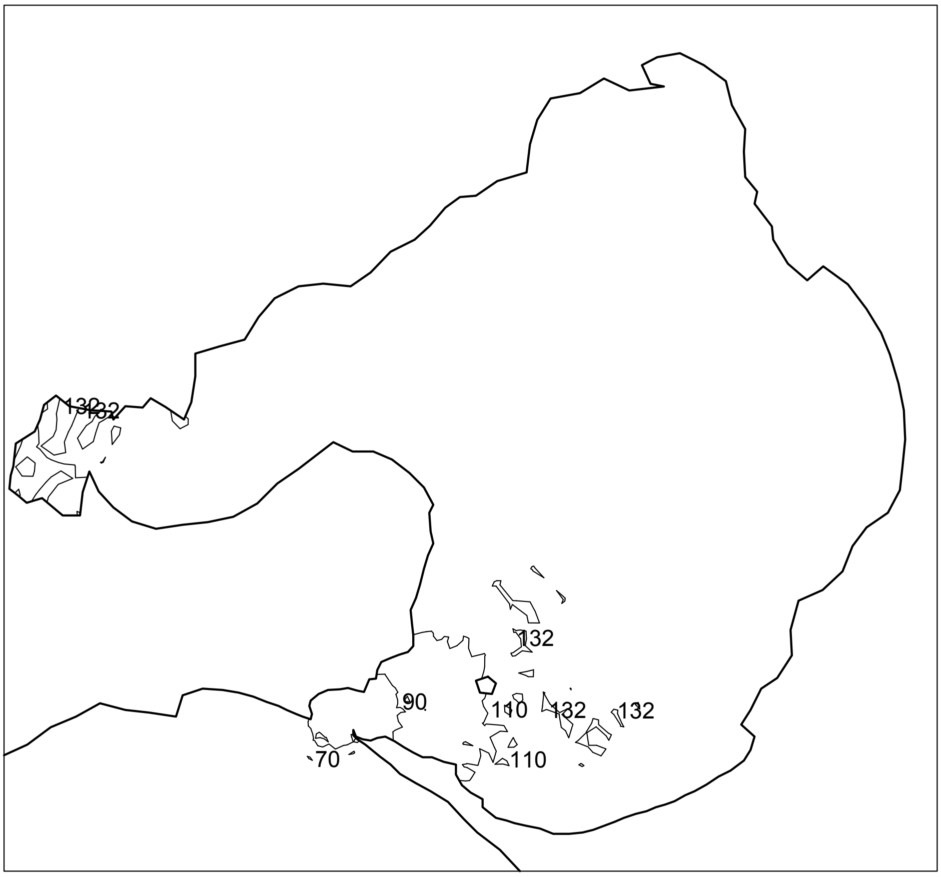

Figure 7.14: The contours of the phases (degrees) of the modelled K_1 water level component.

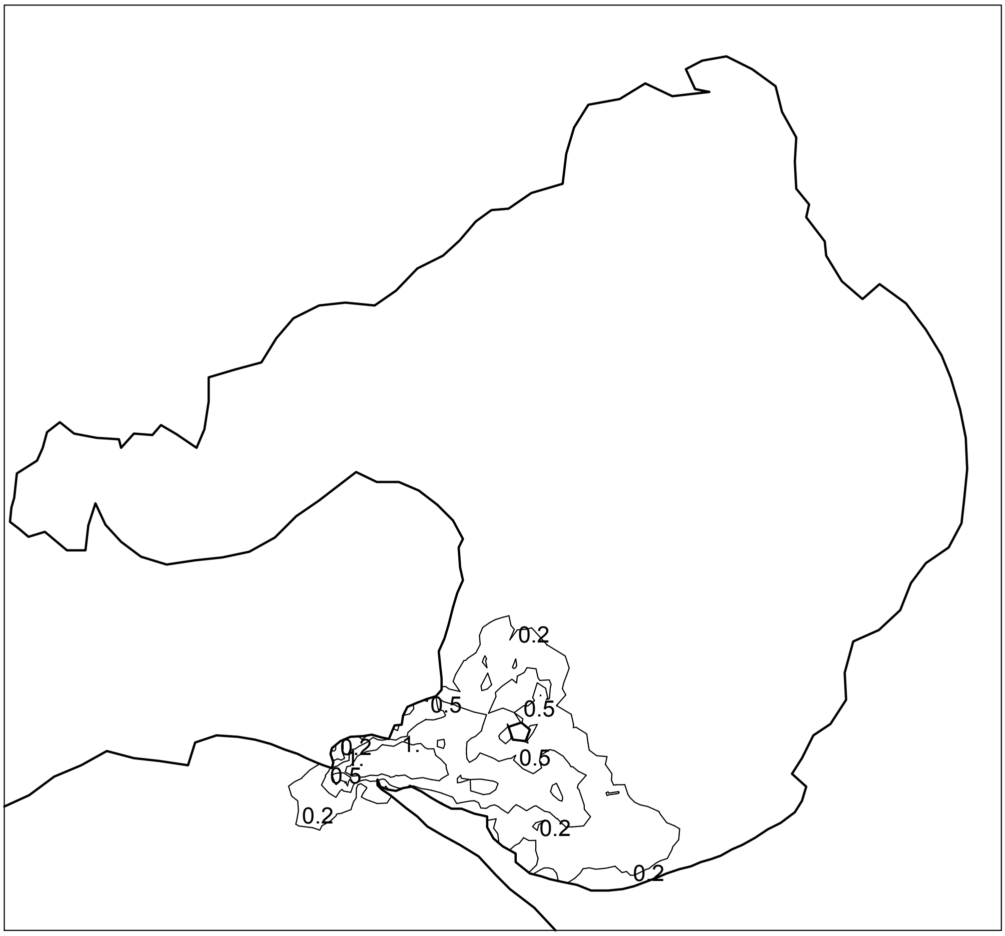

Figure 7.15: The contours of the amplitudes (ms^{-1}) of the modelled M_2 U-velocity.

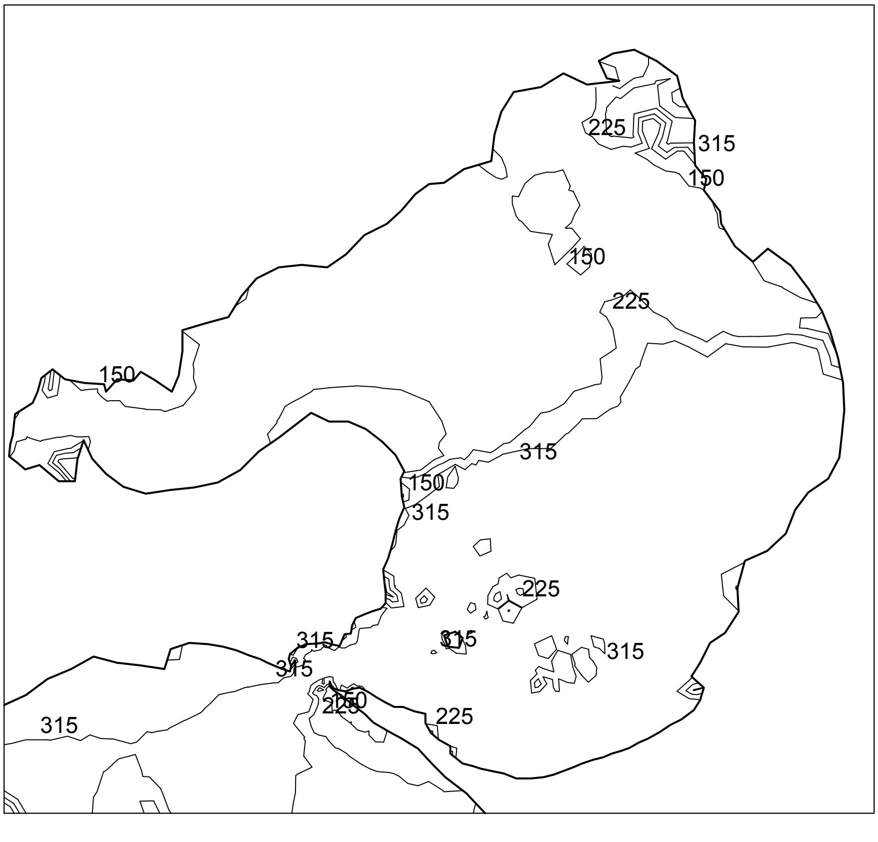

Figure 7.16: The contours of the phases (degrees) of the modelled M_2 U-velocity.

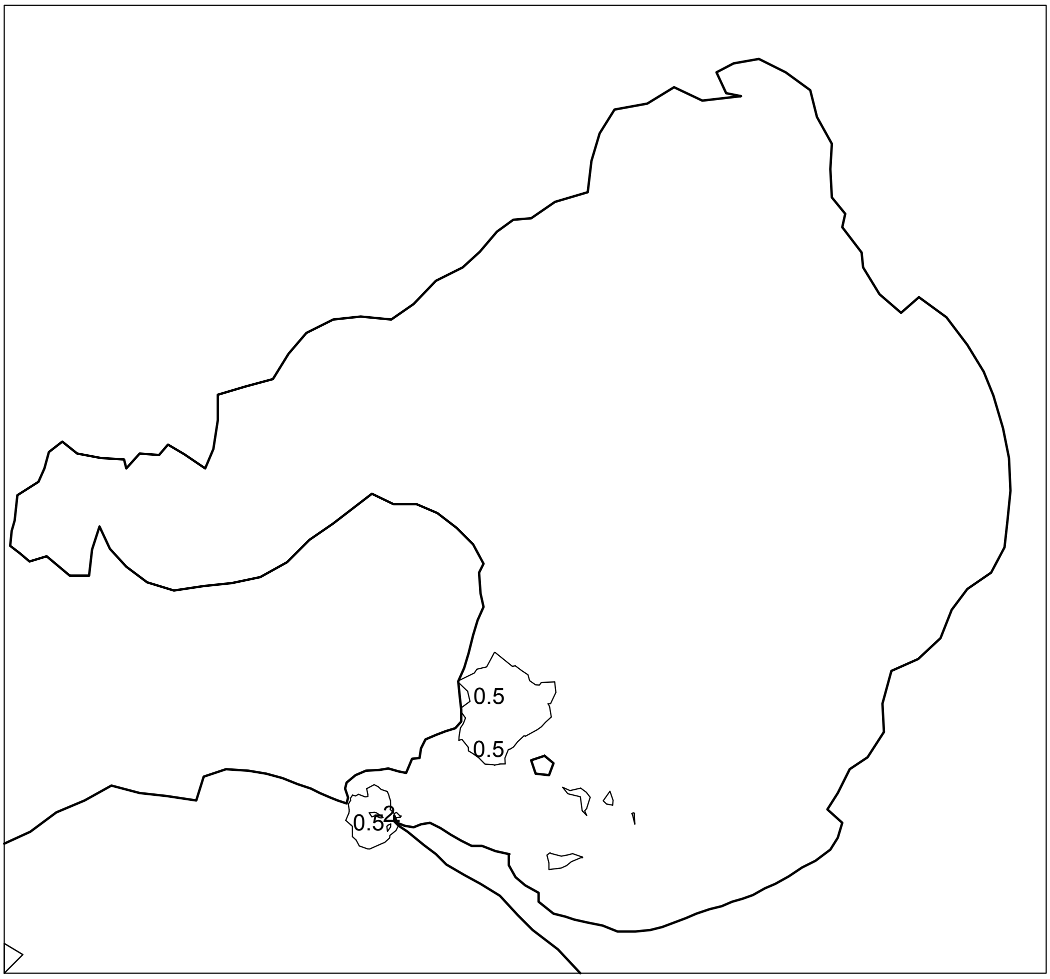

Figure 7.17: The contours of the amplitudes (ms^{-1}) of the modelled M_2 V-velocity.

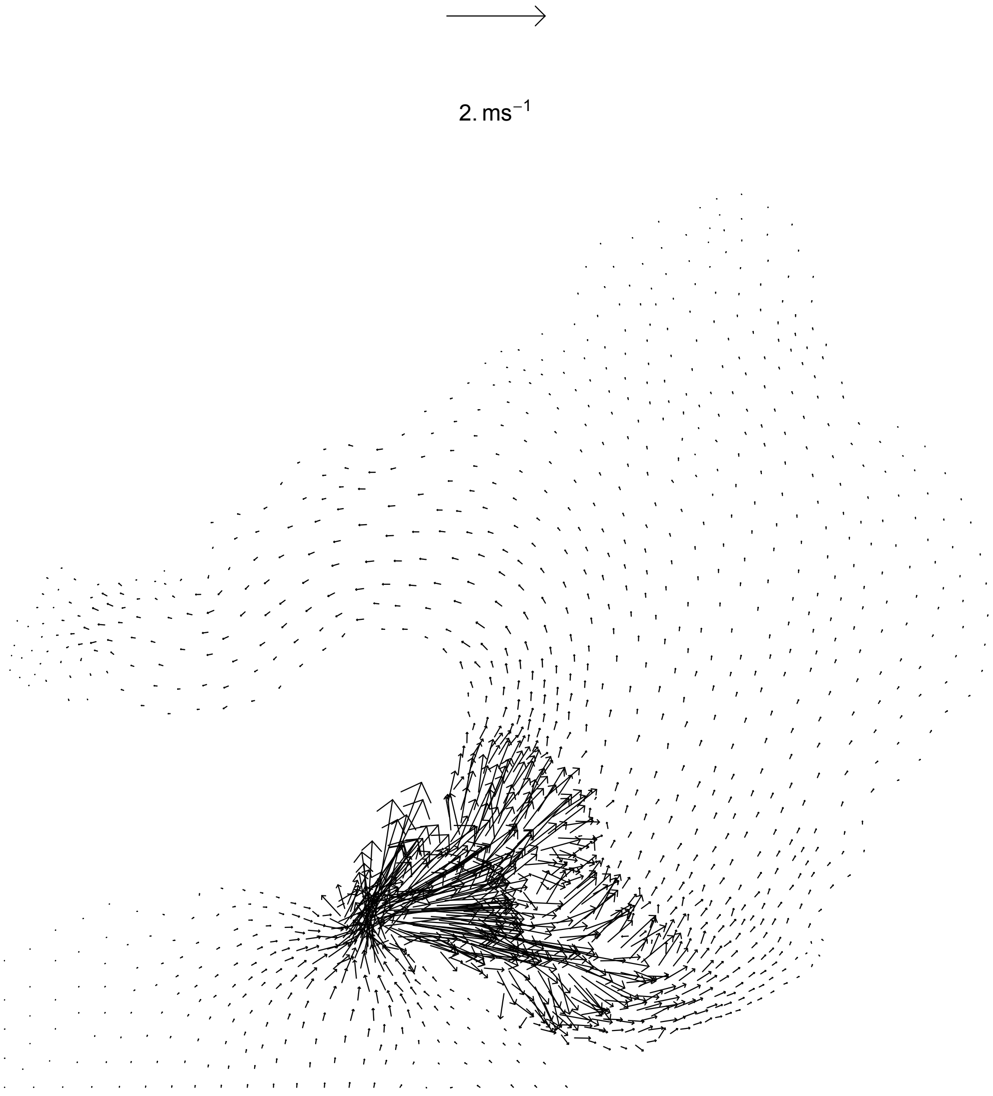

Figure 7.18: The velocity vectors at a time of high tide at Point Lonsdale.

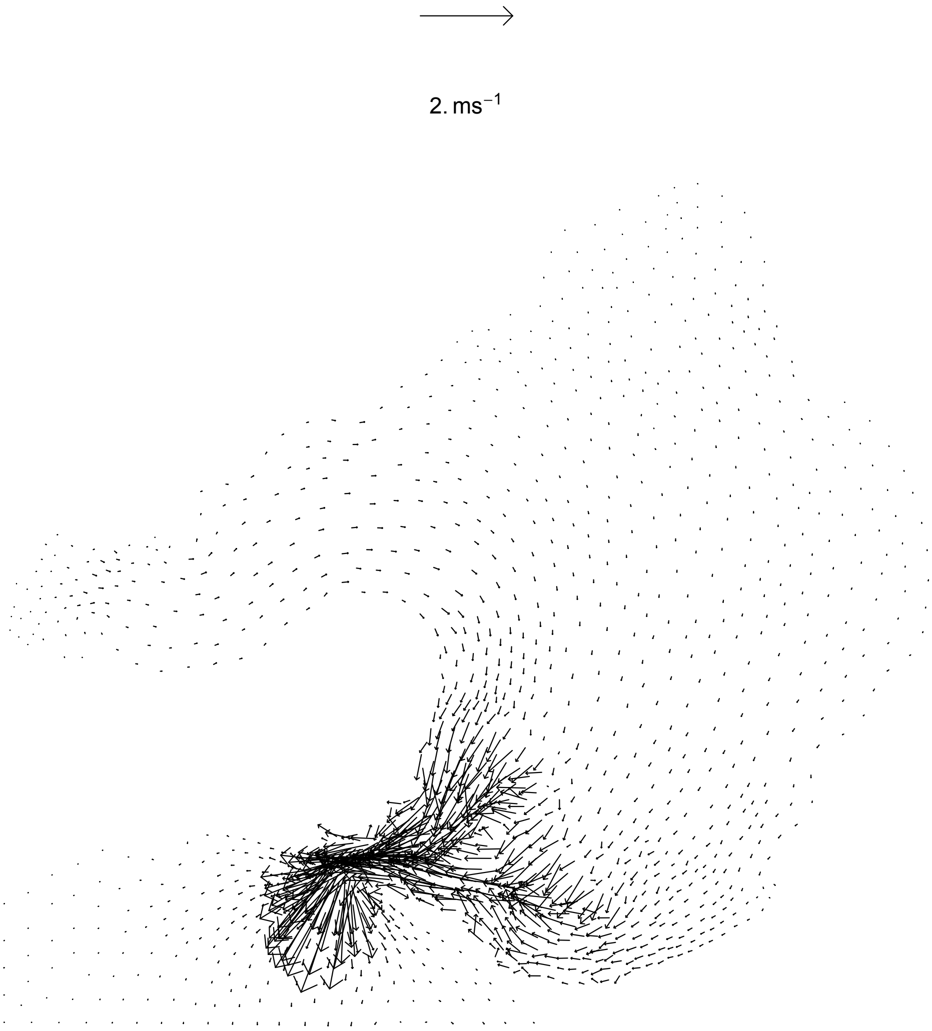

Figure 7.19: The velocity vectors at a time of low tide at Point Lonsdale.

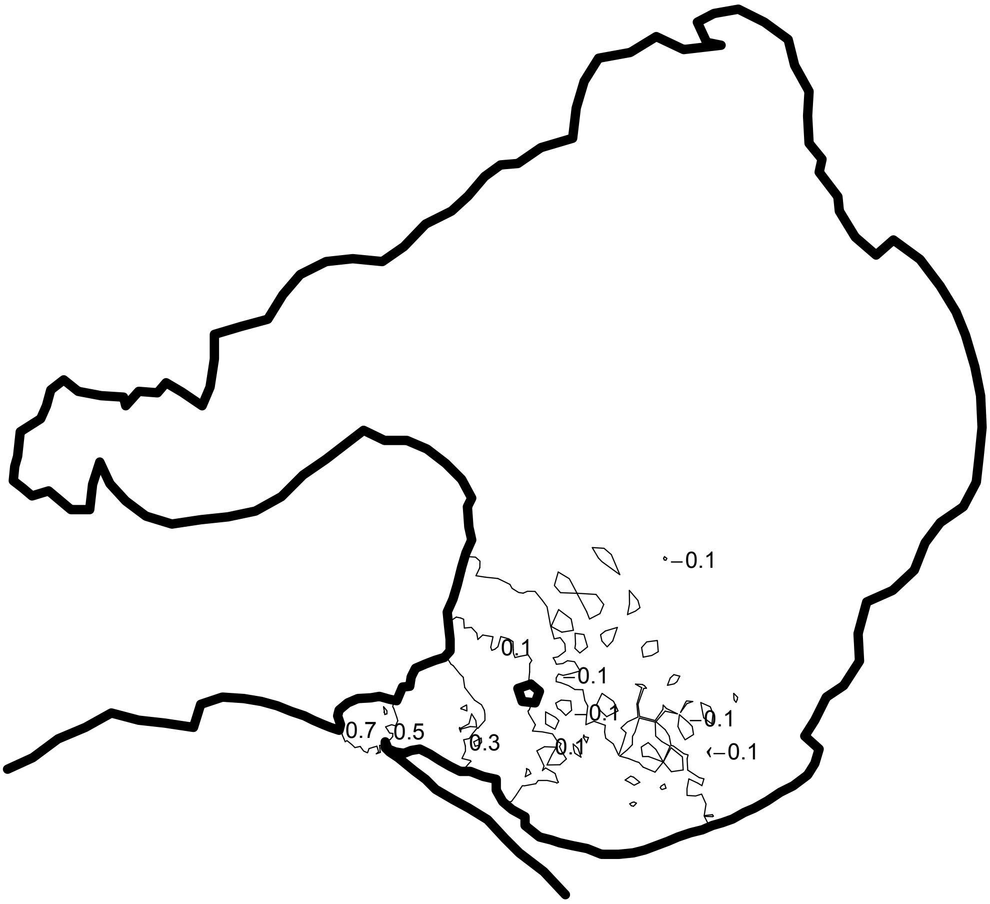

Figure 7.20: The contours of the water level heights (m) at a time of high tide at Point Lonsdale.

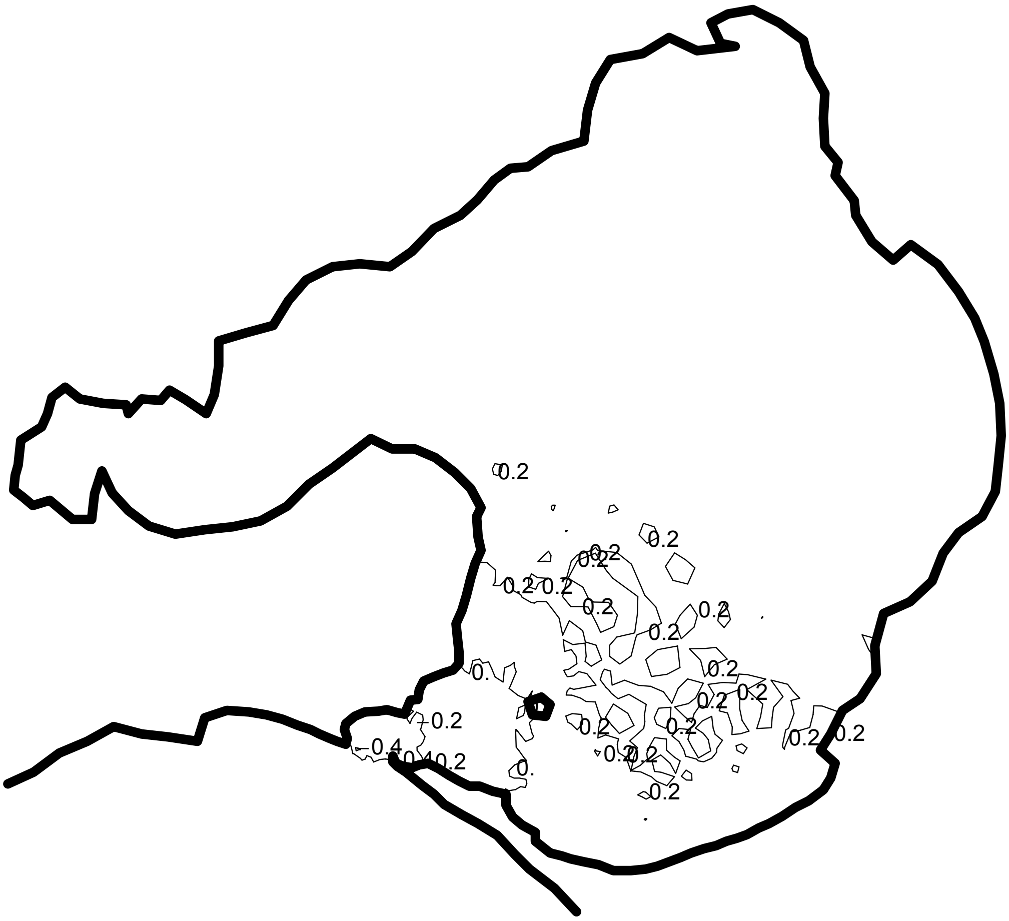

Figure 7.21: The contours of the water level heights (m) at a time of low tide at Point Lonsdale.

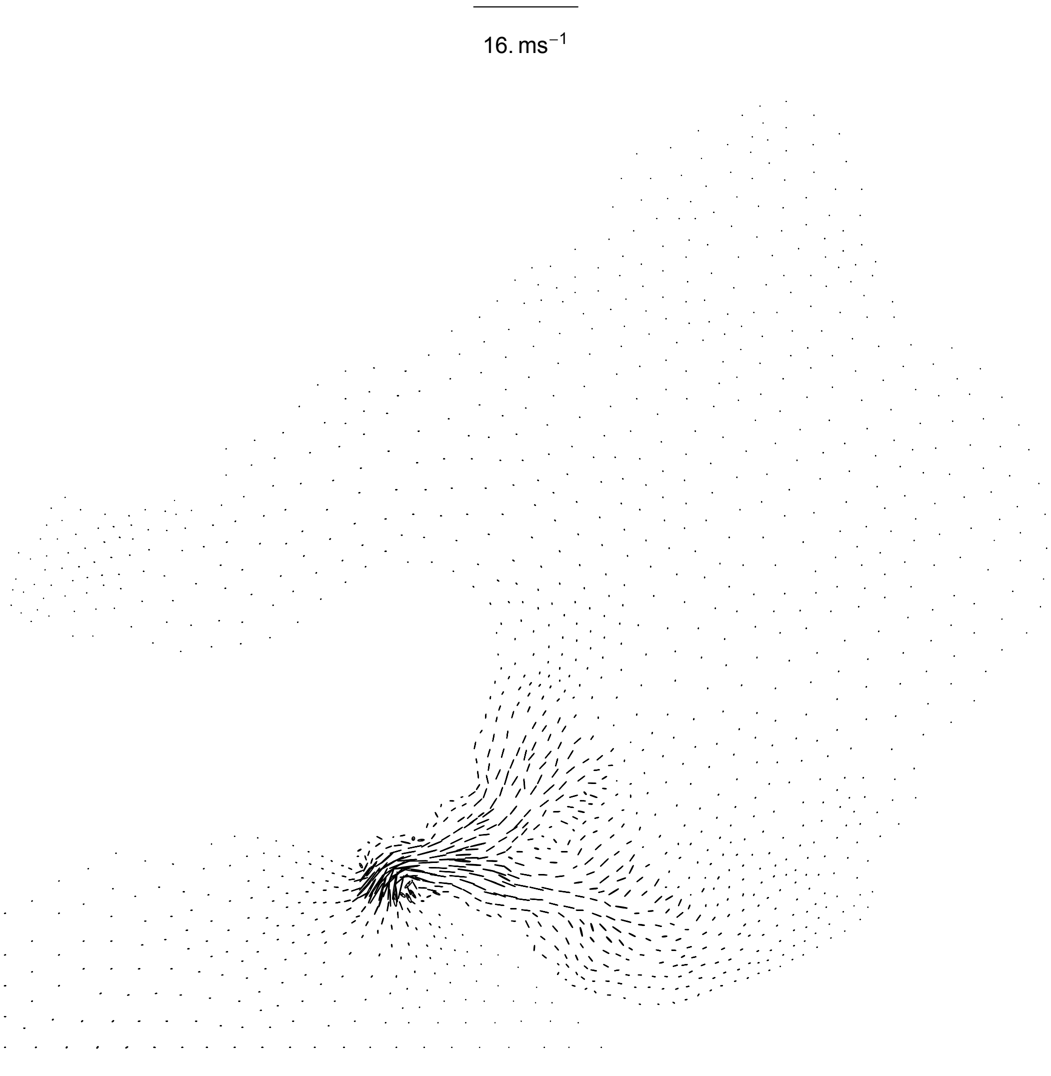

Figure 7.22: The M_2 velocity ellipses in Port Phillip Bay.

7.5 Post channel deepening data

Increasing the depth of the channel at the entrance to the Bay will increase the volume of water flowing into the Bay. The numerical model used in this thesis showed that the tidal levels increased at most locations and decreased at others. The modelled existing tidal levels at seven different locations were compared with the modelled post-channel deepening tidal levels and these results were compared with those of Lawson and Treloar [57]. The results at four locations are shown in Table 7.4. It can be seen that the results are in close agreement with those of Lawson and Treloar for amplitude change. The results for phase change are in good agreement with those of Lawson and Treloar at Geelong and Williamstown. There is some discrepancy in the results for phase change at Queenscliff and Hovell Pile although the differences are small.

The values shown in Table 7.5 were calculated using the experimental data for the observed present tide (called E), the modelled present tide (called M) and the modelled post-channel deepening tide (called P). The terms E, M and P were all calculated using only M_2, S_2, N_2, K_1 and O_1 components. Because the model of the existing tide is slightly inaccurate the change in existing height is estimated by $(E/M)/P - E$.

The results for the tidal currents, given in Table 7.6, show the changes to be small, at most 5%. The largest percentage change was at the South Channel location, where channel deepening is to occur and where current would be expected to have the greatest change. The results are comparable to those of Lawson and Treloar.

Table 7.4: Comparison of tidal constituents between modelled existing and modelled post channel deepening.

Tidal constituent	Amplitude change (m)		Phase change (degrees)		Vector difference (m)	
	This study	LT	This study	LT	This study	LT
Geelong						
M_2	+0.004	+0.005	+0.1	0.0	0.004	0.005
S_2	+0.001	+0.001	+0.1	+0.1	0.001	0.001
N_2	+0.000	+0.001	+0.2	0.0	0.000	0.001
K_1	+0.001	+0.002	−0.4	−0.4	0.001	0.002
O_1	+0.001	+0.001	−0.5	−0.3	0.001	0.001
Queenscliff						
M_2	−0.002	0.000	+0.6	−0.1	0.003	0.001
S_2	0.000	0.000	+0.5	−0.2	0.001	0.000
N_2	0.000	0.000	+0.7	+0.3	0.001	0.000
K_1	0.000	0.000	+0.3	−0.2	0.001	0.000
O_1	0.000	0.000	+0.4	+0.2	0.000	0.001
Williamstown						
M_2	+0.004	+0.004	−0.1	0.0	0.004	0.004
S_2	+0.001	+0.001	+0.1	+0.1	0.001	0.001
N_2	0.000	0.000	+0.2	0.0	0.001	0.000
K_1	+0.001	+0.002	−0.4	−0.4	0.001	0.002
O_1	+0.001	+0.002	−0.5	−0.3	0.000	0.002
Hovell Pile						
M_2	+0.003	+0.004	0.0	−1.4	0.003	0.006
S_2	+0.001	+0.001	+0.2	−1.9	0.001	0.002
N_2	+0.001	+0.001	+0.3	−1.6	0.001	0.001
K_1	+0.002	+0.002	−0.4	−1.2	0.002	0.003
O_1	+0.001	+0.002	−0.5	−1.1	0.001	0.002

Table 7.5: Modelled changes in maximum tidal height at various locations due to the channel deepening.

Location	Change in maximum tidal height (m)	
	This study	Lawson and Treloar
Point Lonsdale	−0.001	
Queenscliff	−0.002	−0.002
Hovell Pile	+0.007	+0.005
West Channel Pile	+0.006	
Point Richards Channel No. 1	+0.007	
Geelong	+0.006	
Williamstown	+0.006	+0.008

7.6 Conclusions

The conclusion of this chapter is that the effect of the proposed channel deepening on astronomical tidal levels in Port Phillip Bay will be to increase the maximum tidal height at most locations, with the largest increase in maximum tidal height predicted to be seven mm. The results are in close agreement with those of Lawson and Treloar, who found that the maximum tidal height will increase at most locations, with the largest increase in maximum tidal height predicted to be eight mm.

Table 7.6: Comparison of modelled existing tidal currents with modelled post-channel deepening currents.

Tidal Constituent	Tidal Ellipse Major semi-axes modelled existing (cm s^{-1})	Tidal Ellipse, Major semi-axes modelled post-channelled deepening (cm s^{-1})	Percentage Change	
			This Study	LT
Symonds Channel				
M_2	72.1	71.9	-0.3	0.5
S_2	17.2	17.1	-0.6	-0.3
N_2	13.0	12.9	-0.8	0.0
K_1	16.0	15.8	-1.3	-1.5
O_1	11.5	11.4	-0.9	1.2
South Channel				
M_2	55.2	57.2	3.6	5.8
S_2	13.4	14.0	4.5	5.6
N_2	10.1	10.5	4.0	8.2
K_1	14.1	14.7	4.3	8.9
O_1	10.3	10.7	3.9	5.0
Portsea Channel				
M_2	41.8	40.7	-2.6	0.9
S_2	10.0	9.8	-2.0	0.8
N_2	7.6	7.6	0.0	1.4
K_1	9.9	9.6	-3.0	0.3
O_1	7.1	6.8	-4.2	0.6

Chapter 8

Conclusions

In this thesis some new solutions of the shallow water wave equations were developed, some analytical and some numerical.

The main achievements of this thesis are:

(i) analytical solutions have been established for moving boundary one dimensional nonlinear shallow water wave equations involving nonlinear continuity and linear friction for unforced flow in a parabolic canal;

(ii) analytical solutions have been established for moving boundary two dimensional nonlinear shallow water wave equations involving nonlinear continuity and linear friction for unforced flow in a circular paraboloidal basin and an elliptical paraboloidal basin;

(iii) analytic solutions have been established for moving boundary one dimensional nonlinear shallow water wave equations involving nonlinear continuity and both no friction and linear friction for forced flow in a bed with quadratically varying depth;

(iv) analytical solutions have been established for one dimensional nonlinear frictionless shallow water wave flow involving advection and nonlinear continuity in a basin of constant depth, with a fixed boundary and with a sinusoidal input at the open sea boundary;

(v) the SLM model of Kawahara, Hirano and Tsubota [46] has been coded in Visual C++ and validated against an analytical solution;
(vi) a convergence study has been carried out for the SLM model;
(vii) a computer program has been written in Visual C++ to generate finite element meshes;
(viii) the fixed boundary analytical solutions that have been developed in (iv) have been compared with numerical solutions obtained by the selective lumped mass model with results in close agreement;
(ix) the SLM model has been applied to moving boundary forced frictionless flow above a bed with quadratically varying depth and moving boundary forced linear frictional flow above a bed with quadratically varying depth; the wetting and drying scheme used was a modification of the scheme of Kawahara, Hirano and Tsubota.The results were compared with the results from the analytical solution developed in this thesis and found to be in close agreement;
(x) the SLM model has been applied to model accurately the existing tidal heights and tidal currents in Port Phillip Bay, Victoria, Australia;
(xi) the SLM model has been applied to model the effects of proposed channel deepening on the tides in Port Phillip Bay. The effect has been found to be very small, affecting the tidal heights by seven millimetres, a result very close to the eight millimetres achieved by consultants to the Victorian government using a different numerical model.

The SLM method that was used in numerical modelling in this thesis has the advantage of giving explicit solutions of the shallow water wave equations. It does however have the disadvantage of being subject to node to node oscillations [29]. However, these oscillations can be minimised by using elements with sides that are small compared with the wavelength as Goraya [29] showed. This also was shown in the convergence study in this thesis. It was found in the modelling of the Port Phillip Bay tides that these oscillations can be further reduced by using turbulent viscosity terms.

The work presented in this thesis can be extended in a number of ways. The analytical moving boundary solutions could be extended to include Coriolis terms. The numerical moving boundary model developed in this thesis was tested against two of the analytical moving boundary solutions developed in the thesis; it could be tested against the other analytical moving boundary solutions developed in the thesis. In addition, the numerical moving boundary model could be used to model tidal flow in regions with significant tidal flats, e.g. Swan Bay, Westernport Bay and Corner Inlet, all in Victoria, Australia.

Also, the numerical moving boundary scheme could be used to model tsunamis with suitable boundary conditions. The boundary conditions could be the displacement of part of the sea bed due to an earthquake. In this case the depth of the sea bottom would vary over time in the vicinity of an earthquake. As well, the numerical moving boundary scheme could be used to model storm surges e.g. those generated from tropical cyclones on the Pacific and Indian Ocean coasts of Australia. The numerical scheme would need to be modified to include pressure gradient and wind stress terms. The surface pressure and wind fields during the cyclone would need to be specified.

In this thesis the numerical moving boundary scheme used fixed nodes and hence fixed elements. The scheme could be modified so that nodes initially on the shoreline stay on the shoreline over time. Hence there would be moving nodes and some elements changing shape over time.

Bibliography

[1] Ages, A. and Woollard, A., *The tides in the Fraser estuary*, Institute of Ocean Sciences, 1976, Unpublished manuscript.

[2] Akanbi, A. A. and Katopodes, N. D., Model for flood propagation on initially dry land, *Journal of Hydraulic Engineering*, **114**, 1988, 689-706.

[3] Apelt, C. J., Gout, J. J. and Szewczyk, A. A., Numerical modelling of pollutant transport and dispersion in bays and estuaries, *Numerical Methods in Fluid Dynamics*, Eds Brebbia, C. A. and Connor, J. J., Pentech Press, London, 1974.

[4] Aramaki, G., Koga, K., Takada, S. and Onishi, K., Tidal current analysis of Ariake Sea by finite elements, *Computational Mechanics*, Eds Valliappan, S., Pulmano, V.A. and Tin-Loi, F., Balkema, Rotterdam, 1993, 1111-1116.

[5] Ball, F. K., An exact theory of simple finite water oscillations on a rotating earth, *Hydraulics and Fluid Mechanics*, Ed Silvester, R., Macmillan, 1964, 293-305.

[6] Balzano, A., Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models, *Coastal Engineering*, **34**, 1998, 83-107.

[7] Bills, P. J., *Barotropic Depth-Averaged and Three-Dimensional Tidal Programs for Shallow Seas*, Ph. D. thesis, University of Adelaide, Adelaide, South Australia, Australia, 1991.

[8] Bills, P. J. and Noye, B. J., Model of a Coastal Sea with Tidal Flats, *Computational Techniques and Applications: CTAC-91*, Eds Noye, B. J., Benjamib, B. R. and Colgan, L. H., 1992, 117-126.

[9] Black, K., Hatton, D. and Rosenberg, M., Locally and Externally-Driven Dynamics of a Large Semi-Enclosed Bay in Southern Australia, *Journal of Coastal Resources*, **9**, 1993, 509-538.

[10] Bokhove, O., Flooding and drying in discontinuous Galerkin finite-element disretizations of shallow-water equations, Part 1: One dimension, *Journal of Scientific Computing*, **22**, 2005, 47-82.

[11] Carrier, G. F. and Greenspan, H. P., Water waves of finite amplitudes on a sloping beach, *Journal of Fluid Mechanics*, **4**, 1958, 97-109.

[12] Charnock, H. and Crease, J., North Sea surges, *Science Progress*, **45**, 1957, 494-511.

[13] Cheng, R. T., Casulli, V., and Gartner, J. W., Tidal, residual, intertidal, mudflat (TRIM) model and its applications to San Francisco Bay, California *Estuarine, Coastal and Shelf Science*, **36**, 1993, 235-280.

[14] Chubarov, L. B. and Fedotova, Z. I., Numerical solution of the long-wave runup on a coast, *Russian Journal of Numerical Analysis and Mathematical Modelling*, **18**, 135-158, 2003.

[15] Daugherty, A. B. and Ingersoll, A. C., *Fluid mechanics with engineering applications*, McGraw-Hill, 1954.

[16] Dietrich, J. C., Kolar, R. L., and Luettich, R. A., *Assessment of ADCIRC's wetting and drying algorithm*, Computational Methods in Water Resources, Proceedings of the 15th International Conference on Computational Methods in Water Resources (CMWR XV), June 13-17, 2004, Chapel Hill, NC, USA, Eds: Miller, C. T., Farthing, M. W., Gray, W. G. and Pinder, G. F., **55**, 2, Elsevier, 2004.

[17] DiLorenzo, J. L., The overtide and filtering response of small inlet/bay systems, *Hydrodynamics and Sediment Dynamics of Tidal Inlets, Coastal and Estuarine Studies*, **29**, Eds Aubrey, D. G. and Weishar, L., Springer-Verlag, 1988, 24-53.

[18] Easton, A., Selected programs for tidal analysis and prediction, *Computing Report No. 9, Flinders Institute for Atmospheric and Marine Sciences, Flinders University of South Australia*, 1977.

[19] Easton, A., A tidal model of Corio Bay, Victoria, *Numerical Simulation of Fluid Motion*, Ed Noye, J., 1978, 357-369.

[20] Easton, A., Singh, M. and Goraya, S., Tidal calculations for Lake Wellington, Victoria, *Ocean and Atmospheric Pacific*, Ed Aung, T.H., National Tidal Facility, Flinders University of South Australia, 1995, 28-32.

[21] Erturk, S. N., *Modelling the friction effects of eelgrass on the tidal flow in Great Bay, New Hampshire*, Ph. D. thesis, University of New Hampshire, 2000.

[22] Falconer, R. A. and Owens, P. H., Numerical simulation of flooding and drying in a depth-averaged tidal flow model, *Proceedings Institution Civil Engineers*, **83**, 1987, 161-180.

[23] Fandry, C. B., Hubbert, G. D. and McIntosh, P. C., Comparison of predictions of a numerical model and observations of tides in Bass Strait, *Australian Journal of Marine and Freshwater Research*, **36**, 1985, 737-52.

[24] Flather, R. A. and Heaps N. S., Tidal computations for Morecambe Bay, *Geophysical Journal of the Royal Astronomical Society*, **42**, 1975, 489-517.

[25] Flather, R. A. and Hubbert, K. P., Tide and surge models for shallow water - Morecambe Bay revisited, *Modelling Marine Systems*, Ed Davies, A.M., 1990, 135-166.

[26] Friedrichs, C. T. and Madsen, O. S., Nonlinear diffusion of the tidal signal in frictionally dominated embayments, *Journal of Geophysical Research*, **97**, 1992, 5637-5650.

[27] Gallagher, B. S. and Munk, W. H., Tides in shallow water: spectroscopy, *Tellus*, **23**, 1971, 346-363.

[28] Gopalakrishnan, T. C., A moving boundary circulation model for regions with large tidal flats, *International Journal for Numerical Methods in Engineering*, **28**, 245-260, 1989.

[29] Goraya, S., *A study of finite element tidal models*, Ph. D. thesis, Swinburne University of Technology, Melbourne, Australia, 2001.

[30] Goto, Ch., Nonlinear equation of long waves in the Lagrangian description, *Coastal Engineering, Japan*, **22**, 1979, 1-9.

[31] Gray, W. G. and Lynch, D. R., Time-stepping schemes for finite element tidal model computations, *Advances in Water Resources*, **1**, 1977, 83-95.

[32] Henry, R. F., Duncalf, D. S., Walters, R. S., Osborne, M. J. and Murty, T. S., A study of tides and storm surges in offshore waters of the Meghna estuary using a finite element model, *MAUSAM*, **48**, 1997, 519-530.

[33] Herrling, B., Finite element model for estuaries with inter-tidal flats, *Coastal Engineering*, **22**, 1976, 3396-3415.

[34] Hibberd, S. and Peregrine, D. H., Surf and run-up on a beach: a uniform bore, *Journal of Fluid Mechanics*, **95**, 1978, 323-345.

[35] Holdahl, R., Holden, H. and Lie, K-A., Unconditionally stable splitting methods for the shallow water equations, *BIT*, **39**, 451-472, 1998.

[36] Holz, K. P., and Nitsche, G., Tidal wave analysis for estuaries with intertidal flats, *Advances in Water Resources*, **5**, 1982, 142-148.

[37] Hubbert, G. D., and McInnes, K. L., A storm surge inundation model for coastal planning and impact studies, *J. Coastal Res.*, **15**, 1999, 168-185.

[38] Ip, J. T. C. , Lynch, D. R. and Friedrichs, C. T., Simulation of estuarine flooding and dewatering with application to Great Bay, New Hampshire, *Estuarine Coastal and Shelf Science*, 119, **47**, 1998, 119-141.

[39] Ippen, A. T., Tidal dynamics in estuaries, Part 1; estuaries of rectangular section in estuary and coastline hydrodynamics, *Estuary and Coastline Dynamics*, Ed. Ippen, A. T., McGraw-Hill Publishing Co., Inc., New York, N.Y., 1966, 493-545.

[40] Johns, B., Numerical integration of the shallow water equations over a sloping shelf, *International Journal for Numerical Methods in Fluids*, **2**, 1982, 253-261.

[41] Johns, B., Dube, S. K., Sinha, P. C., Mohanty, U. C. and Rao, A. D., The simulation of continuously deforming lateral boundaries in problems involving the shallow-water equations, *Computers and Fluids*, **10**, 1982, 105-116.

[42] Johnson, R. S., *A Modern Introduction to the Mathematical Theory of Water Waves*, Cambridge University Press, 1997.

[43] Kabbaj, A. and Le Provost, C., Nonlinear tidal waves in channels; a perturbation method adapted to the importance of quadratic bottom friction, *Tellus*, **32**, 1980, 143-163.

[44] Kanoglu, K., Nonlinear evolution and runup-rundown of long waves over a sloping beach, *Journal of Fluid Mechanics*, **513**, 2004, 363-372.

[45] Kashiyama, K., Sugano, S., Behr, M. and Tezduyar, T. E., Space-time finite element method for shallow water flows considering moving boundaries, *Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference*, 1999, 1-6.

[46] Kawahara M, Hirano H and Tsubota K., Selective lumping finite element method for shallow water flow, *International Journal for Numerical Methods in Fluids*, **2**, 1982, 89-112.

[47] Kawahara, M. and Umetsu, T., Finite element method for moving boundary problems in river flow, *International Journal for Numerical Methods in Fluids*, **6**, 1986, 365-386.

[48] Keller, J. B. and Keller, H. B., Water wave run-up on a beach, *ONR Research Rep. Contract NONR-3828(00)*, Dept. Navy, Washington, D.C., 1964.

[49] Kinnmark, I. P. E., The Shallow Water Wave Equations : Formulation, Analysis and Application, *Lecture Notes in Engineering*, Eds Brebbia, C. A., Orszag, S., Springer-Verlag, Berlin, **15**, 1986.

[50] Kolar, R. L., Gray, W. G., Westerink, J. J. and Luettich, R. A., Shallow water modelling in spherical coordinates: equation formulation, numerical implementation and application, *Journal of Hydraulic Research*, **32**, 1994, 3-24.

[51] Knight, D. W., Long wave propagation in an idealised estuary, *Journal of the Hydraulics Division, American Society of Civil Engineers*, **99**, 1973, 993-1007.

[52] Kowalik, Z. and Murty, T. S., *Numerical modeling of ocean dynamics*, World Scientific, Singapore, 1995.

[53] Kowalik, Z., Knight, W., Logan, T.and Whitmore, P., Numerical modeling of the global tsunami, *Science of Tsunami Hazards*, **23**, 2005, 40-56.

[54] Kreiss, H., Some remarks about nonlinear oscillations in tidal channels, *Tellus*, **9**, 1957, 53-68.

[55] Lamb, H., *Hydrodynamics*, Cambridge University Press, 1945.

[56] Lawson and Treloar Pty Ltd, *POMC Channel Deepening, Sediment Transport and Water Quality Modelling*, Report prepared for the Port of Melbourne Corporation by Lawson and Treloar

Pty Ltd, report Rm2054/J5372 ver 1.0 FINAL June 2004, [Online] http://www.channelproject.com/global/docs/Hydrodynamics%20Report.pdf

[57] Lawson and Treloar Pty Ltd, *POMC Channel Deepening, Sediment Transport and Water Quality Modelling Model Calibration*, Report prepared for the Port of Melbourne Corporation by Lawson and Treloar Pty Ltd, report Rm2074/J5372 ver 1.0 FINAL June 2004, [Online] http://www.channelproject.com/global/docs/Calibration%20Report.pdf

[58] Leclerc, M., Bellemare, J-F., Dumas, G. and Dhatt, G., A finite element model of estuarian and river flows with moving boundaries, *Advances in Water Resources*, **13**, 1990, 158-168.

[59] Leendertse, J. J., *Aspects of a computational model for long period water-wave propagation*, Rand Memorandum Rm-5294-PR, Santa Monica, CA, 1967.

[60] Leendertse, J. J., *A water-quality simulation model for well-mixed estuaries and coastal seas: Vol. I, Principles of Computation*, Rand Memorandum RM-6230-RC, Santa Monica, CA, 1970.

[61] Lewis, C. H. III and Adams, W. M., Development of a tsunami-flooding model having versatile formation of moving boundary conditions, *The Tsunami Society Monograph Series*, No. 1, 1983.

[62] Li, Y. and Raichlen, F., Solitary wave runup on plane slopes, *Journal of Waterway, Port, Coastal, and Ocean Engineering, American Society of Civil engineers*, **127**, 2001, 33-44.

[63] Luettich, R. A., Westerink, J. J and Scheffner, N. W., ADCIRC: an advanced three-dimensional circulation model for shelves, coasts and estuaries, report 1: Theory and methodology of ADCIRC-2DDI and

ADCIRC-3DL, *Technical Report DRP-92-6*, Department of the Army, Washington, D.C., 1991.

[64] Luettich, R. A. and Westerink, J. J., Elemental wetting and drying in the ADCIRC hydrodynamic model for ADCIRC version 34.XX, *Contractors Report*, Department of the Army, US Army Corps of Engineers, Waterways Experiment Station, Vicksburg, MS., 1999.

[65] Luettich, R. A., Hurricane Katrina - storm surge, August 28-29, 2005, [Online] http://www.unc.edu/ims/ccats/surge/katrina/katrina.htm

[66] Lynch, D., *Finite element solution of the shallow water equations*, Ph. D. thesis, Princeton University, 1978.

[67] Lynch, D. R. and Gray, W. G., Analytic solutions for computer flow model testing, *Journal of the Hydraulics Division, American Society of Civil Engineers*, **104**, 1978, 1409-1428.

[68] Lynch, D. R., and Gray, W. G., Finite element simulation of shallow water problems with moving boundaries, *Finite Elements in Water Resources II* Eds Brebbia, C. A., Gray, W. G. and Pinder,G. F., Pentech Press, London, 1978, 2.23-2.24.

[69] Lynch, D. R., and Gray, W. G., A Wave equation model for finite element tidal computations, *Computers and Fluids*, **7**, 1979, 207-228.

[70] Lynch, D. R. and Gray, W. G., Finite element simulation of flowing deforming regions, *Journal of Computational Physics*, **36**, 2, 1980, 135-153.

[71] Mader, C. L., Numerical tsunami flooding study:1, *Science of Tsunami Hazards*, **8**, 1990, 79-96.

[72] Mason, D. C., Davenport, I. J., Flather, R. A. and Gurney, C., A digital elevation model of the inter-tidal areas of the Wash, England, produced by the waterline method, *International Journal of Remote Sensing*, **19**, 1998, 1455-1460.

[73] Ninomiya, H. and Onishi, K., *Flow Analysis Using a PC*, Computational Mechanics, Boston, 1991.

[74] Pearson, C. A., Numerical method for shallow-water motion onto a beach, *Computational Methods in Nonlinear Mechanics*, North Holland Publishing Company, Ed Oden, J. T. 1980, 379-391.

[75] Pedersen, G. and Gjevik, B., Run-up of solitary waves, *Journal of Fluid Mechanics*, **135**, 1983, 283-299.

[76] Pelinovsky, E. and Mazova, R., Exact analytical solutions of nonlinear problems of tsunami wave run-up on slopes with different profiles, *Natural Hazards*, **6**, 1992, 227-249.

[77] Peterson, P., Hauser, J., Thacker, W. C. and Eppel, D., An Error-Minimizing Algorithm for the Non-Linear Shallow-Water Wave Equations with Moving Boundaries, *Numerical Methods for Non-Linear Problems*, Eds C. Taylor, E. Hinton, D. R. J. Owen and E. Onate, **2**, Pineridge Press, 1984, 826-836. [Online] `http://www.cle.de/hpcc/publications/`

[78] Port of Melbourne Corporation, *Secondary Constituents*, Hydrographic Survey Section, 2001.

[79] Port of Melbourne Corporation, *Standard Port Constituents*, Hydrographic Survey Section, 2002.

[80] Prandle, D. and Rahman, M., Responses in estuaries, *Journal of Physical Oceanography*, **10**, 1980, 1552-1573.

[81] Prasad, R. S. and Svensen, I. A., Moving shoreline boundary conditions for nearshore models, *Coastal Engineering*, **49**, 2003, 239-261.

[82] Proudman, J., Oscillations of tide and surge in an estuary of finite length, *Journal of Fluid Mechanics*, **2**, 1957, 371-382.

[83] Rahman, M., *Water waves*, Clarendon Press, 1995.

[84] Ramming, H.-G., Reproduction of physical processes in coastal areas, *Proceedings of the 13th Coastal Engineering Conference, American Society of Civil Engineers*, Vancouver, Canada, 1972.

[85] Reid, R. O. and Bodine, B. R., Numerical model for storm surges in Galveston Bay, *Journal Waterways and Harbors Division, American Society of Civil Engineers*, **9**, 35-57, 1968.

[86] Roig, L. C., *A finite element model for simulating flows in tidal flats*, M.S. Thesis, Department of Civil Engineering, University of California Davis, Davis, CA, 1989.

[87] Runchal, A. K., Numerical model for storm surge and tidal run-up studies, *Symposium on Modeling Techniques Volume II, 2nd Annual Symposium of The Waterways, Harbors and Coastal Engineering Division, American Society of Civil Engineers*, 1975, 1516-1534.

[88] Sachdev, P. L., Paliannapan, D. and Sarathy, R., Regular and chaotic flows in paraboloidal basins and eddies, *Chaos, Solitons and Fractals*, **7**, 383-408, 1996.

[89] Sampson, J., Easton, A. and Singh, M., Moving boundary shallow water flow in circular paraboloidal basins, *Proceedings of the Sixth Engineering Mathematics and Applications Conference, 5th International Congress on Industrial and Applied Mathematics, University of Technology, Sydney, Australia*, Eds May, R. L. and Blyth, W. F., 2003, 223-227.

[90] Sampson, J., Easton, A. and Singh, M., Modelling the effect of proposed channel deepening on the tides in Port Phillip Bay, *Australian and New Zealand Industrial and Applied Mathematics Journal*, **46E**, C888-C901, 2005. [Online] http://anziamj.austms.org.au/V46/CTAC2004/Samp.

[91] Sampson, J., Easton, A. and Singh, M., Moving boundary shallow water flow in parabolic bottom topography, *Australian and New Zealand Industrial and Applied Mathematics Journal*, **47E**, C373-387, 2006, [Online] http://anziamj.austms.org.au/V47EMAC2005/Sampson.

[92] Shuto, N., Run-up of long waves on a sloping beach, *Coastal Engineering, Japan*, **10**, 1967, 23-38.

[93] Shuto, N., Standing waves in front of a sloping dike, *Proceedings of The 13th Conference On Coastal Engineering*, 1971, 1629-1647.

[94] Shuto, N., and Goto, Ch., Numerical simulation of tsunami run-up, *Coastal Engineering, Japan*, **21**, 1978, 13-20.

[95] Siden, G. L. D. and Lynch, D. R., Wave equation hydrodynamics on deforming elements, *International Journal for Numerical Methods in Fluids*, **8**, 1988, 1071-1093.

[96] Sielecki, A. and Wurtele, M. G., The numerical integration of the nonlinear shallow-water equations with sloping boundaries, *Journal of Computational Physics*, **6**, 1970, 219-236.

[97] Spielvogel, L. Q., Single-wave run-up on sloping beaches, *Journal of Fluid Mechanics*, **74**, 4, 1974, 685-694.

[98] Stelling, G. S., *On the construction of computational methods for shallow-water flow problems*, Ph. D. thesis, Delft Univ. of Technology, 1983.

[99] Stoker, J. J., *Water waves*, Interscience, New York, 1957.

[100] Synolakis, C. E., The runup of solitary waves, *Journal of Fluid Mechanics*, **185**, 1987, 532-545.

[101] Tadepalli, S. and Synolakis, C. E., The run-up of N-waves on sloping beaches, *Proc. R. Soc. Lond. A*, **445**, 1994, 99-112.

[102] Thacker, W. C., Some exact solutions to the nonlinear shallow-water wave equations, *Journal of Fluid Mechanics*, **107**, 499-508, 1981.

[103] Titov, V. V. and Synolakis, C. E., Modeling of breaking and nonbreaking long-wave evolution and runup using VTCS-2, *Journal of Waterway, Port, Coastal and Ocean Engineering, American Society of Civil Engineers*, **121**, 1995, 308-316.

[104] Titov, V. V. and Synolakis, C. E., Numerical modeling of tidal wave runup, *Journal of Waterway, Port, Coastal and Ocean Engineering, American Society of Civil Engineers*, **124**, 1998, 157-171.

[105] Tuck, E. O. and Hwang, L. S., Long wave generation on a sloping beach *Journal of Fluid Mechanics*, **51**, 1972, 449-461.

[106] Vemulakonda, S., Scheffner, N., Mark, D. and Brown, M., Water-surface elevation frequency estimates for the Louisiana coast, *The CER-Cular*, **CERC-99-1**, 1999.

[107] Vincent, S., Caltagirone, J-P., and Bonneton, P., Numerical modelling of bore propagation and run-up on sloping beaches using a MacCormack TVD scheme, *Journal of Hydraulic Research*, **39**, 41-49, 2001.

[108] Vreugdenhil, C. B., *Numerical Methods for Shallow-Water Flow*, Kluwer Academic Publishers, 1998.

[109] Walker, S. J., *Hydrodynamic models of Port Phillip Bay*, Technical report No. 38, Port Phillip Bay Environmental Study, CSIRO Environmental Project Office, 1997.

[110] Westerink, J. J., Luettich, R. A., Baptista, A. M., Scheffner, N. W. and Farrar, P., Tide and storm surge predictions using a finite element model, *Journal of Hydraulic Engineering, American Society of Civil Engineers*, **118**, 1992, 1373-1390.

[111] Yeh, G. and Chou, F., Moving boundary numerical surge model, *Journal of Waterway, Port, Coastal and Ocean Engineering, American Society of Civil Engineers*, **105**, 1979, 247-263.

[112] Yoon, S. B., and Cho, J. H., Numerical simulation of Coastal Inundation over Discontinuous Topography, *Water Engineering Research*, **2**, 2001, 75-87.

[113] Zech, Y., Sorel, M. C., and Vansnick, M., Mathematical modelling of floods in river flooding and uncovering of flood plains, *International Conference, Hydraulic Aspects of Floods and Flood Controls*, London, 1983, 217-227.

[114] Zelt, J. A., Tsunamis: The response of harbours with sloping boundaries to long wave excitation, *Report Kh-R-47*, W. M. Keck Laboratory of Hydraulics and Water Resources, California Institute of Technology, 1986.

[115] Zelt, J. A. and Raichlen, F., Overland flow from solitary waves, *Journal of Waterway, Port, Coastal and Ocean Engineering, American Society of Civil Engineers*, **117**, 1991, 247-263.

[116] Zhang, J. E., *Run-up of ocean waves on beaches*, Ph. D. Thesis, California Institute of Technology, 1996.

Appendix A

Visual C++ code for shallow water wave flow

```
/* Recttidalcalc.cpp:a computer program to predict
  tidal flow or other shallow water flow
  in a domain with a fixed boundary
  or a domain with a moving boundary,
  and with linear or quadratic friction,
  and with turbulent viscosity, using the
  selective lumped mass scheme.

  The selective lumped mass scheme,
  developed by Kawahara, Hirano and Tsubota,
  is used to solve the shallow water equations,
  three simultaneous nonlinear partial d.e's, which represent tidal or
  other shallow water flow.

  The selective lumped mass scheme is finite element in space,
  finite difference in time, and is explicit.

  The wetting and drying scheme used is that of the author of this
  computer program.

  The water level at the open sea boundary
  must be specified at each time step.

  At the end of each time step, at each node,
  the U-velocity, V-velocity and water level,
  represented in the program by u, w and zeta, are calculated.
  As well, at the end of each time step, it is calculated whether a
```

```
 node is wet or dry and whether a triangle is wet or dry.

 Joe Sampson*/

 # include <stdio.h>
 # include <math.h>
 # include <cstdlib>

int i,k,l,m2,m3,n,n1,n4;
double x[4920],y[4920]; /* x and y coordinates [m] */
double d[9220];/*water depth [m]*/
double adj;/*amount by which the water depth is adjusted*/
double xbai,ybai,dbai; /* scale factors*/
int ii[9220][3];/* nodenumber(triangle no.,vertex no.)*/
int ml2[5];/*the total number of nodes on the land boundary*/
int ni[5];/*the total number of nodes on the island*/
int ibu[250]; /* the numbers assigned to the nodes on the land boundary*/
int ilu[20]; /*the numbers assigned to the nodes on the island boundary*/
int ibh[20]; /*the numbers assigned to the nodes on the open sea boundary*/
double del[9220]; /*the area of the triangle element [sq. m] */
double ltau;/*the linear friction factor*/
double qtau[9220];/*the quadratic friction factor*/
double tau; /* the friction factor, set equal to ltau for linear friction
(including zero friction) and qtau for quadratic friction */
double u[4920],w[4920]; /* the u-velocity and v-velocity
at the nth time step [m/sec]*/
double zeta[4920];/* the computed tidal height at the nth time step [m]*/
double exptlzeta[4920];/* the actual tidal height at the nth time
step [m] for selected points using the observed M2, S2, K1, O1 and
N2 amplitudes and phases*/
double modelzeta[4920];/* the modelled tidal height
at the nth time step [m] for selected points
using the modelled M2, S2, K1, O1 and N2 amplitudes and phases*/
int idisk, iqfr;
int iprog;
double un[4920], wn[4920], dn[4920]; /* (u, v, zeta) at each half-time step */
double um[4920], wm[4920], em[4920]; /* (u, v, zeta)*am (see next line for
definition of am) at each half-time step */
double am[4920]; /* (sum of triangle areas)/3 for a given node */
double su[3], sw[3], se[3],e0; /* used in calculating u, v, and zeta */
/*double uu[13][6], ww[13][6], dd[13][6]; *//* (u,v,zeta) values
output to a file */
```

```
double a,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15;
double   pai,  delk,  delt, g6;
double dt,ds,t, dtt,dt24, tlimit,tt, tt1, tt2, tt3, tt4, tt5;
double ph1, ph2, ph3, ph4, ph5 ,ph6,ph7, ph8,ph9,ph10,ph11,ph12,ph13,ph14,
ph15;
int  i1, i2, i3,i4, im, ir, iq, id,  nn2, kk1, kk2,j2, nn3;
double  b1, b2, b3, c1, c2, c3,the, biui, ciui, biei,  bdt, de12,
u1,u2,u3,w1, w2, w3;
double biwi, ciwi, ciei, cdt, d1, d2, d3,dav, bihi, cihi, bucw, e1, e2, e3,
twoh1, twoh2, twoh3,twohu,twohw,s0,s1,s2,q1,q2,q3;
int jj, nh, ns, nit,istep;
int nstep;
double eps; /* the lumping parameter epsilon*/
double ae,be;
double nmann; /* nmann is the Manning friction factor*/
double cf; /* cf is the friction coefficient */
double cf1[4920],cf2[4920],cf3[4920];
 /* cf1, cf2, and cf3 are friction coefficients */
double p1;
double p2[4920];
double db[4920];
double d1b,d2b,d3b;
double adv;/*adv is set to 1.0 for advective flow and
0.0 for nonadvective flow */
double hmin;
double visc;
double ed[3][3];
double turb;
int nd1, nd2,nd3;
int i123;
int nwtotal;
int trianglewet[9220],nodewet[4920], twet[4920];
double tauwetnode, depthwetnode, qtauwetnode;
double b, aq, h0;
int incl[4920];
double mindepth, zdiff;

FILE *frectxyddata;
FILE *frectnodedata;
FILE *flanddata;
FILE *fislanddata;
FILE *frectseadata;
FILE *frecttidalcalc;
FILE *frecttidaltrace;
FILE *fxydout;
FILE *fzuvout;
FILE *fexptltidal;
FILE *ftidalday;
FILE *fmodeltidal;

  void vector()
  {/*calculating the velocity components and the tidal height*/

  for(k=0;k<n;k++){
      if (nodewet[k]==1)
 {um[k]=0;
  wm[k]=0;
  em[k]=0;
  am[k]=0;
 };
  };
 dt24=dtt/24.0;
 g6=9.81/6.0;

 /*calculating b1,b2,b3,c1,c2 and c3 for each wet triangle*/
   /*printf("time during vector calc b1 etc  %lf\n",t);*/

  for(k=0;k<n1;k++){
   if (trianglewet[k]==1)
   {delt=del[k]/3.0;
               for(im=0;im<3;im++){
                   ir=ii[k][im];
                   am[ir]=am[ir]+delt;
   /*fprintf(frecttidaltrace," time %lf ir %i am %lf \n",t);*/
                   };
  i1=ii[k][0];
  i2=ii[k][1];
  i3=ii[k][2];
  b1=y[i2]-y[i3];
  b2=y[i3]-y[i1];
  b3=y[i1]-y[i2];
  c1=x[i3]-x[i2];
  c2=x[i1]-x[i3];
  c3=x[i2]-x[i1];
  d1=dn[i1]+d[i1];
  d2=dn[i2]+d[i2];
  d3=dn[i3]+d[i3];
  ed[1][1]=(b1*b1+c1*c1)*(turb/(4*del[k]));
  ed[1][2]=(b1*b2+c1*c2)*(turb/(4*del[k]));
  ed[1][3]=(b1*b3+c1*c3)*(turb/(4*del[k]));
  ed[2][1]=ed[1][2];
  ed[2][2]=(b2*b2+c2*c2)*(turb/(4*del[k]));
  ed[2][3]=(b2*b3+c2*c3)*(turb/(4*del[k]));
  ed[3][1]=ed[1][3];
  ed[3][2]=ed[2][3];
  ed[3][3]=(b3*b3+c3*c3)*(turb/(4*del[k]));

 /* cf is the friction coefficient */
  /*original expression for p1, which I believe is incorrect */
  /*p1=pow(((d[i1]*d[i1]+d[i2]*d[i2]+d[i3]*d[i3])/3.0),(-1.0/3.0));*/
  /*new expression for p1, which I believe is correct */
  /*p1=pow(((d[i1]+d[i2]+d[i3])/3.0),(-1.0/3.0));
  cf=(1.0/3.0)*(nmann*nmann)*9.81*p1;*/
 /* cf1, cf2, and cf3 are friction coefficients */
  /*db[i1]=d[i1];
  if (d[i1]<1.0) db[i1]=1.0;
  db[i2]=d[i2];
  if (d[i2]<1.0) db[i2]=1.0;
  db[i3]=d[i3];
  if (d[i3]<1.0) db[i3]=1.0;
  p2[i1]=pow(db[i1],(-1.0/3.0));
  p2[i2]=pow(db[i2],(-1.0/3.0));
  p2[i3]=pow(db[i3],(-1.0/3.0));
  cf1[i1]=(1.0/3.0)*(nmann*nmann)*9.81*p2[i1];
  cf1[i2]=(1.0/3.0)*(nmann*nmann)*9.81*p2[i2];
  cf1[i3]=(1.0/3.0)*(nmann*nmann)*9.81*p2[i3];*/

  /* calculating qtau[k], the quadratic friction factor, for each triangle */

   d1b=d1;
   if (d1<hmin) d1b=hmin;
   d2b=d2;
   if (d2<hmin) d2b=hmin;
   d3b=d3;
   if (d3<hmin) d3b=hmin;
   /* cf1, cf2, and cf3 are friction coefficients */

  p2[i1]=pow(d1b,(-1.0/3.0));
  p2[i2]=pow(d2b,(-1.0/3.0));
  p2[i3]=pow(d3b,(-1.0/3.0));
  cf1[i1]=(1.0/3.0)*(nmann*nmann)*9.81*p2[i1];
  cf1[i2]=(1.0/3.0)*(nmann*nmann)*9.81*p2[i2];
  cf1[i3]=(1.0/3.0)*(nmann*nmann)*9.81*p2[i3];

 /* qtau[k]=cf*(sqrt(un[i1]*un[i1]+wn[i1]*wn[i1])/d1+sqrt(un[i2]*un[i2]
 +wn[i2]*wn[i2])/d2+sqrt(un[i3]*un[i3]+wn[i3]*wn[i3])/d3);*/
  qtau[k]=cf1[i1]*sqrt(un[i1]*un[i1]+wn[i1]*wn[i1])/d1b
  +cf1[i2]*sqrt(un[i2]*un[i2]+wn[i2]*wn[i2])/d2b
  +cf1[i3]*sqrt(un[i3]*un[i3]+wn[i3]*wn[i3])/d3b;

   /*if (t==5.0*dt) {fprintf(frecttidaltrace,"cf is %lf\n",
  cf);};*/

  /*setting tau, the friction factor; if linear friction (
  including zero friction)is used, tau is set equal to ltau,
  and if quadratic friction is used, tau is set equal to qtau[k];*/
  tau=ltau;/*linear friction is used in the model */
  /*tau=qtau[k];*//*quadratic friction is used in the model */

  /*u component for each triangle */
  biui=b1*un[i1]+b2*un[i2]+b3*un[i3];
  ciui=c1*un[i1]+c2*un[i2]+b3*un[i3];
  biei=(b1*dn[i1]+b2*dn[i2]+b3*dn[i3])*g6;
 /* bie=b1*dn[i1]+b2*dn[i2]+b3*dn[i3];*/
 /* bi1=b1*dn[i1];*/
 /* bi2=b2*dn[i2];*/
 /* bi3=b3*dn[i3];*/
  /* if (t==2.0*dt) {fprintf(frecttidaltrace,"%.9f \n",bie);};*/
  /*if (t==2.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f \n",
  bi1,bi2,bi3);};*/
 /* if (t==2.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f %.9f
 %.9f %.9f\n", b1,b2,b3,dn[i1],dn[i2],dn[i3]);};*/
 /* if (t==2.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f\n",biui, biei);};*/
  /* if (t==2.0*dt) {fprintf(frecttidaltrace,"%.9f \n",g6);};*/

  u1=un[i1]*biui+wn[i1]*ciui;
  u2=un[i2]*biui+wn[i2]*ciui;
```

```
   u3=un[i3]*biui+wn[i3]*ciui;
   /*if (t==1.5*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f \n",
   u1,u2,u3);};*/
   /*if (t==1.5*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f \n",
   un[i1],un[i2],un[i3]);};*/
   bdt=dtt*biei;
    /*if (t==2.0*dt) {fprintf(frecttidaltrace,"%f\n",bdt);};*/
   de12=del[k]/12.0;
   /*if (t==1.5*dt) {fprintf(frecttidaltrace,"%f\n",
   del[k]);};*/
    /* if (t==1.5*dt) {fprintf(frecttidaltrace,"%f\n",
   de12);};*/
    /* s0=(ae*u[i1]+be*u[i2]+be*u[i3])*del[k]/12.0;*/
  /* definition of su[0] for the nonadvective case*/
   /*su[0]=(ae*u[i1]+be*u[i2]+be*u[i3])*del[k]/12.0-bdt
       -dtt*(2.0*un[i1]+un[i2]+un[i3])*tau*del[k]/12.0;*/
   /*definition of su[0] for the nonadvective or advective case*/
  su[0]=(ae*u[i1]+be*u[i2]+be*u[i3])*del[k]/12.0-bdt-adv*dt24*
  (2.0*u1+u2+u3)-dtt*(2.0*un[i1]+un[i2]+un[i3])*tau*del[k]/12.0
   +dtt*(ed[1][1]*un[i1]+ed[1][2]*un[i2]+ed[1][3]*un[i3])*visc*turb;

  /*s0=-(2.0*un[i1]+un[i2]+un[i3])*tau*del[k]/12.0;
   if (t==150.0*dt) {fprintf(frecttidaltrace,"s0 equals %.9lf\n",
   s0);};
if (t==0.9*tt) {fprintf(frecttidaltrace,"tau equals %.9lf\n",
   tau);};*/
   /*s1=(be*u[i1]+ae*u[i2]+be*u[i3])*del[k]/12.0;
    /*definition of su[1] for the nonadvective case*/
   /*su[1]=(be*u[i1]+ae*u[i2]+be*u[i3])*del[k]/12.0-bdt
       -dtt*(un[i1]+2.0*un[i2]+un[i3])*tau*del[k]/12.0;*/
    /*definition of su[1] for the nonadvective or advective case*/
    su[1]=(be*u[i1]+ae*u[i2]+be*u[i3])*del[k]/12.0-bdt-adv*dt24*
    (u1+2.0*u2+u3)-dtt*(un[i1]+2.0*un[i2]+un[i3])*tau*del[k]/12.0
   +dtt*(ed[2][1]*un[i1]+ed[2][2]*un[i2]+ed[2][3]*un[i3])*visc*turb;
    /*s2=(be*u[i1]+be*u[i2]+ae*u[i3])*del[k]/12.0;*/
    /*definition of su[2] for the nonadvective case*/
    /*su[2]=(be*u[i1]+be*u[i2]+ae*u[i3])*del[k]/12.0-bdt
        -dtt*(un[i1]+un[i2]+2.0*un[i3])*tau*del[k]/12.0;*/
    /*definition of su[2] for the nonadvective or advective case*/
    su[2]=(be*u[i1]+be*u[i2]+ae*u[i3])*del[k]/12.0-bdt
    -adv*dt24*(u1+u2+2.0*u3)-dtt*(un[i1]+un[i2]+2.0*un[i3])*tau*del[k]/12.0
    +dtt*(ed[3][1]*un[i1]+ed[3][2]*un[i2]+ed[3][3]*un[i3])*visc*turb;;
   /* if (t==1.5*dt) {fprintf(frecttidaltrace,"%.9f\n",
  s1);};*/
/* if (t==2.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f \n",
 su[0],su[1],su[2]);};*/
 /*if (t==1.5*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f \n",
 u[i1],u[i2],u[i3]);};*/

 /*v component for each triangle, with "v" given the symbol
 "w" in this program*/

 biwi=b1*wn[i1]+b2*wn[i2]+b3*wn[i3];
 ciwi=c1*wn[i1]+c2*wn[i2]+b3*wn[i3];
 ciei=(c1*dn[i1]+c2*dn[i2]+c3*dn[i3])*g6;
 w1=un[i1]*biwi+wn[i1]*ciwi;
 w2=un[i2]*biwi+wn[i2]*ciwi;
 w3=un[i3]*biwi+wn[i3]*ciwi;
 cdt=dtt*ciei;
 /*if (t==2.0*dt) {fprintf(frecttidaltrace,"%f\n",cdt);};*/
 /*if (t==2.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f \n",
 c1,c2,c3);};*/
 /*q0=(ae*w[i1]+be*w[i2]+be*w[i3])*del[k]/12.0;*/
 /* definition of sw[0] for the nonadvective case*/
 /*sw[0]=(ae*w[i1]+be*w[i2]+be*w[i3])*del[k]/12.0-cdt
     -dtt*(2.0*wn[i1]+wn[i2]+wn[i3])*tau*del[k]/12.0;*/
 /* definition of sw[0] for the nonadvective or advective case*/
 sw[0]=(ae*w[i1]+be*w[i2]+be*w[i3])*del[k]/12.0-cdt-adv*dt24*(2.0*w1+w2+w3)
 -dtt*(2.0*wn[i1]+wn[i2]+wn[i3])*tau*del[k]/12.0
 +dtt*(ed[1][1]*wn[i1]+ed[1][2]*wn[i2]+ed[1][3]*wn[i3])*visc*turb;
 /*q1=(be*w[i1]+ae*w[i2]+be*w[i3])*del[k]/12.0;*/
   /* definition of sw[1] for the nonadvective case*/
 /*sw[1]=(be*w[i1]+ae*w[i2]+be*w[i3])*del[k]/12.0-cdt
     -dtt*(wn[i1]+2.0*wn[i2]+wn[i3])*tau*del[k]/12.0;*/
 /* definition of sw[1] for the nonadvective or advective case*/
  sw[1]=(be*w[i1]+ae*w[i2]+be*w[i3])*del[k]/12.0-cdt-adv*dt24*
  (w1+2.0*w2+w3)-dtt*(wn[i1]+2.0*wn[i2]+wn[i3])*tau*del[k]/12.0
 +dtt*(ed[2][1]*wn[i1]+ed[2][2]*wn[i2]+ed[2][3]*wn[i3])*visc*turb;
/* q2=(be*w[i1]+be*w[i2]+ae*w[i3])*del[k]/12.0;*/
  /* definition of sw[2] for the nonadvective case*/
 /*sw[2]=(be*w[i1]+be*w[i2]+ae*w[i3])*del[k]/12.0-cdt
     -dtt*(wn[i1]+wn[i2]+2.0*wn[i3])*tau*del[k]/12.0;*/
  /* definition of sw[2] for the nonadvective or advective case*/
 sw[2]=(be*w[i1]+be*w[i2]+ae*w[i3])*del[k]/12.0-cdt-adv*dt24*(w1+w2+2.0*w3)
 -dtt*(wn[i1]+wn[i2]+2.0*wn[i3])*tau*del[k]/12.0
 +dtt*(ed[3][1]*wn[i1]+ed[3][2]*wn[i2]+ed[3][3]*wn[i3])*visc*turb;

 /*zeta component for each triangle */

/*d1=dn[i1]+d[i1];
d2=dn[i2]+d[i2];
d3=dn[i3]+d[i3];*/
/*if (t==1.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f  \n",
 dn[i1],dn[i2],dn[i3]);};*/
 /*if (t==1.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f  \n",
 d[i1],d[i2],d[i3]);};*/
 /*if (t==1.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f  \n",
 d1,d2,d3);};*/

 /* definitions of twoh1, twoh2, and twoh3 for the
  continuity equation using the divergence of the
depth below mean sea level times velocity i.e. linear continuity*/
/*dav=(d[i1]+d[i2]+d[i3])/3;
twoh1=2*dav+dav+dav;
twoh2=dav+2*dav+dav;
twoh3=dav+dav+2*dav;*/

/* definitions of twoh1, twoh2, and twoh3 for the
  continuity equation using the divergence of the
total depth times velocity i.e. nonlinear continuity*/
twoh1=2*d1+d2+d3;
twoh2=d1+2*d2+d3;
twoh3=d1+d2+2*d3;


/*if (t==1.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f  \n",
 twoh1,twoh2,twoh3);};*/

twohu=twoh1*un[i1]+twoh2*un[i2]+twoh3*un[i3];
twohw=twoh1*wn[i1]+twoh2*wn[i2]+twoh3*wn[i3];
/*if (t==2.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f  \n",
 twohu,twohw);};*/
/*if (t==1.0*dt) {fprintf(frecttidaltrace,"%.9f   \n",
 twohu);};*/

/*e0=(ae*zeta[i1]+be*zeta[i2]+be*zeta[i3])*del[k]/12.0;
```

```
if (t==1.0*dt) {fprintf(frecttidaltrace,"%f\n",e0);};*/
/*    if (t==1.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f \n",
 zeta[i1],zeta[i2],zeta[i3]);};*/
se[0]=(ae*zeta[i1]+be*zeta[i2]+be*zeta[i3])*del[k]/12.0
+dt24*(b1*twohu+c1*twohw);
/*if (t==1.0*dt) {fprintf(frecttidaltrace,"%f\n",se[0]);};*/
/*e1=(be*zeta[i1]+ae*zeta[i2]+be*zeta[i3])*del[k]/12.0;*/
se[1]=(be*zeta[i1]+ae*zeta[i2]+be*zeta[i3])*del[k]/12.0
+dt24*(b2*twohu+c2*twohw);
/*e2=(be*zeta[i1]+be*zeta[i2]+ae*zeta[i3])*del[k]/12.0;*/
se[2]=(be*zeta[i1]+be*zeta[i2]+ae*zeta[i3])*del[k]/12.0
+dt24*(b3*twohu+c3*twohw);
/*if (t==2.0*dt) {fprintf(frecttidaltrace,"%.9f %.9f %.9f \n",
 se[0],se[1],se[2]);};*/

 for(im=0;im<3;im++){
 ir=ii[k][im];
 um[ir]=um[ir]+su[im];
 wm[ir]=wm[ir]+sw[im];
 em[ir]=em[ir]+se[im];
 };

/* if (t==1.5*dt) {fprintf(frecttidaltrace,"%f\n",del[k]);};*/

 };
 };

 /*incl[k] is the number of wet triangles that node k is included in*/
 for (k=0;k<n;k++){
     incl[k]=0;};

  for (k=0;k<n1;k++){
      for (im=0;im<3;im++){
          ir=ii[k][im];
          incl[ir]=incl[ir]+trianglewet[k];};
  };


 for (k=0;k<n;k++){
     if (incl[k]>=1)
     {un[k]=um[k]/am[k];
     wn[k]=wm[k]/am[k];
```

```
      dn[k]=em[k]/am[k];
/*fprintf(fricttidaltrace," vectorend %i time %lf dn %lf am %lf nw %i\n",
       k,t,dn[k],am[k],nodewet[k]);*/
       };

  };
  }

  /*calculating the tidal height on the sea boundary*/

void boundary()
 {
 for(k=0;k<n4;k++){
 l=ibh[k];
 /*fprintf(fricttidaltrace,"node %i time %lf\n",l,t);*/
 dn[l]=a1*cos(pai/tt1*t-ph1*pai/360)+a2*cos(pai/tt2*t-ph2*pai/360)
     +a3*cos(pai/tt3*t-ph3*pai/360)+a4*cos(pai/tt4*t-ph4*pai/360)
     +a5*cos(pai/tt5*t-ph5*pai/360);
 /*set wn[l] to zero for quadratic bed; do not set a value of wn[l] */
 /*otherwise wn[l]=0.0;*/
/*fprintf(fricttidaltrace," boundary node %i time %lf dn %lf\n",l,t,dn[l]);*/
 };
 }

 void semidryhalftimestep()
{/* at the end of a half time step, if the water is calculated to be
  under the sea bed, reset the total depth to 0 m and the velocity
  to zero */

     for (k=0;k<n;k++){
     if (nodewet[k]==1)
     {ds=d[k]+dn[k];
       if(ds<=0.0)
       {un[k]=0.0;
       wn[k]=0.0;
       dn[k]=-d[k];
       };
     };
     };
     /*If a node k is wet and not in a wet triangle
     the total water depth is made zero */
   for (k=0;k<n;k++){
        if ((nodewet[k]==1) && (incl[k]==0))
        {un[k]=0.0;
        wn[k]=0.0;
        dn[k]=-d[k];};
    };
 }

void dryfulltimestep()
 {/* at the end of a full time step, if the water is calculated to be
   under the sea bed, reset the total depth to 0 m and the velocity
   to zero and dry the node, i.e. set nodewet to zero*/
      for (k=0;k<n;k++){
      if (nodewet[k]==1)
      {ds=d[k]+dn[k];
        if(ds<=0.0)
        {un[k]=0.0;
        wn[k]=0.0;
        dn[k]=-d[k];
        nodewet[k]=0;
        };
      };
        };

      /*If a node k is wet and not in a wet triangle it is made dry */
    for (k=0;k<n;k++){
        if ((nodewet[k]==1) && (incl[k]==0))
        {un[k]=0.0;
        wn[k]=0.0;
        dn[k]=-d[k];
        nodewet[k]=0;};
    };


      /*fprintf(frecttidaltrace,"dryfulltimestep t %lf i %i %i\n",t,k,
      nodewet[k]);*/
    /* for (k=264;k<265;k++) {fprintf(frecttidalcalc,"dryft % lf %lf",
    t,dn[k]);};
     for (k=265;k<296;k++) {fprintf(frecttidalcalc," %lf",dn[k]);};
     for (k=296;k<297;k++) {fprintf(frecttidalcalc," %lf\n",dn[k]);};*/
      /*for (k=264;k<265;k++) {fprintf(frecttidaltrace,
      "dryfulltimestep t %lf  i %i %i",t,k,nodewet[k]);};
     for (k=265;k<296;k++) {fprintf(frecttidaltrace,"
    %i  %i",k,nodewet[k]);};
    for (k=296;k<297;k++)
    {fprintf(frecttidaltrace,"  %i  %i\n",k,nodewet[k]);};*/
}

void wettingcheck ()

  {
      /* Calculating nwtotal in a triangle at the end of each time step.
      if nwtotal=2, this is an interface triangle, i.e. with two wet nodes
      and one dry node.  Zeta, the water height above mean water level,
      will be calculated at the dry node using a momentum equation formula.
      If zeta is found to be greater than -h,
      where h is the depth of the bed
      from mean water level, then the node wets.  The velocity is set equal
      to that of the other nodes which has the higher water elevation.
      twet[i] makes node i wet but this is not activated till later in
      principalloop() in which if twet[i]=1 nodewet[i]=1*/

      for (k=0;k<n1;k++)
      {
          i1=ii[k][0];
          i2=ii[k][1];
          i3=ii[k][2];
          nwtotal=nodewet[i1]+nodewet[i2]+nodewet[i3];
          /*fprintf(frecttidaltrace,"triangle %i  nwtotal %i time %lf  \n",
       k,nwtotal,t);*/

          /*Case 1: See whether node i3, which was
          previously  dry, wets */

          if ((nwtotal==2)&&(nodewet[i1]==1)&&(nodewet[i2]==1))
          {i123=i1;
          if (dn[i2]>dn[i1]) i123=i2;

          /*Calculating qtauwetnode, the quadratic friction factor
          for node i123*/
          /*depthwetnode=dn[i123]+d[i123];
          if (depthwetnode<hmin)
              depthwetnode=hmin;
          qtauwetnode=(1.0)/(3.0)*(nmann*nmann)*9.81
          *pow(depthwetnode,(-4.0,3.0));*/
```

```
   /*fprintf(frecttidaltrace,"triangle %i  qtauwetnode %lf time %lf
   \n",
k,qtauwetnode,t);*/
   /*tauwetnode=qtauwetnode;*/
   /*quadratic friction is used in the model*/
   tauwetnode=ltau;/*linear friction is used in the model*/
   /*Initial wetting scheme:
   calculating a value for the water elevation at the dry node, i3.
   If this value is greater than -h the node wets;
   otherwise it remains dry. */
   dn[i3]=dn[i123]-((x[i3]-x[i123])/9.81)*((un[i123]-u[i123])/dt+
       tauwetnode*un[i123])
       -((y[i3]-y[i123])/9.81)*((wn[i123]-w[i123])/dt+
       tauwetnode*wn[i123]);
   un[i3]=un[i123];
   wn[i3]=wn[i123];
   twet[i3]=0;
   if (dn[i3]>-d[i3]){twet[i3]=1;};

   /*2nd wetting scheme:Calculating whether node i123
   is sufficiently high above node i3
   for the latter node to dry*/

   /*twet[i3]=0;
   if (dn[i123]-dn[i3]>zdiff)
   {dn[i3]=-d[i3]+mindepth;
   un[i3]=un[i123];
   wn[i3]=wn[i123];
   twet[i3]=1;};*/

   /*fprintf(frecttidaltrace,"t %lf i3 %i i123 %i dni123 %lf dni3
    %lf di3 %lf xi3 %lf xi123 %lf\n",
    t,i3,i123,dn[i123],dn[i3],d[i3],x[i3],x[i123]);
    fprintf(frecttidaltrace,"t %lf uni123 %lf ui123 %lf dt %lf\n",
    t,un[i123],u[i123],dt);
    fprintf(frecttidaltrace,"t %lf tri %i i3 %i wet %i\n",t,k,i3,
    nodewet[i3]);*/

   /*Initial wetting scheme*/
    if (dn[i3]<=-d[i3])
   {dn[i3]=-d[i3];
   un[i3]=0.0;
```

```
 wn[i3]=0.0;
 nodewet[i3]=0;};

  /*2nd wetting scheme*/
/* if (dn[i3]<=-d[i3]+mindepth)
 {dn[i3]=-d[i3];
 un[i3]=0.0;
 wn[i3]=0.0;
 nodewet[i3]=0;};*/
  /*fprintf(frecttidaltrace," wcheck t %lf tri %i i3 %i wet %i\n",
  t,k,i3,nodewet[i3]);*/
 };

  /*Case 2: See whether node i2, which was
 previously  dry, wets */
 if ((nwtotal==2)&&(nodewet[i1]==1)&&(nodewet[i3]==1))

 {
  i123=i1;
  if (dn[i3]>dn[i1]) i123=i3;
  /*Calculating qtauwetnode, the quadratic friction factor
 for node i123*/
 /*depthwetnode=dn[i123]+d[i123];*/
 /*fprintf(frecttidaltrace," triangle %i depthwetnode %lf time
 %lf\n",k,depthwetnode,t);*/
 /*if (depthwetnode<hmin)
     depthwetnode=hmin;
 qtauwetnode=(1.0)/(3.0)*(nmann*nmann)*9.81
 *pow(depthwetnode,(-4.0,3.0));*/
 /*fprintf(frecttidaltrace,"triangle %i  qtauwetnode %lf time %lf
 \n",k,qtauwetnode,t);*/
 /*tauwetnode=qtauwetnode;*/
 /*quadratic friction is used in the model*/
 tauwetnode=ltau;/*linear friction is used in the model*/
 /*Initial wetting scheme:
 calculating a value for the water elevation at the dry node, i2.
 If this value is greater than -h the node wets;
 otherwise it remains dry. */
  dn[i2]=dn[i123]-((x[i2]-x[i123])/9.81)*((un[i123]-u[i123])/dt+
     tauwetnode*un[i123])
     -((y[i2]-y[i123])/9.81)*((wn[i123]-w[i123])/dt+
     tauwetnode*wn[i123]);
```

```
   un[i2]=un[i123];
   wn[i2]=wn[i123];
   twet[i2]=0;
   if (dn[i2]>-d[i2]){twet[i2]=1;};

   /*2nd wetting scheme:Calculating whether node i123 is
   sufficiently high above node i2 for the latter node to dry*/
   /*twet[i2]=0;
   if (dn[i123]-dn[i2]>zdiff)
   {dn[i2]=-d[i2]+mindepth;
   un[i2]=un[i123];
   wn[i2]=wn[i123];
   twet[i2]=1;};*/

   /*fprintf(frecttidaltrace,"t %lf i2 %i i123 %i dni123 %lf
   dni2 %lf  di2 %lf xi2 %lf xi123 %lf\n",
    t,i2,i123,dn[i123],dn[i2],d[i2],x[i2],x[i123]);
    fprintf(frecttidaltrace,"t %lf uni123 %lf ui123 %lf dt %lf\n",
    t,un[i123],u[i123],dt);
    fprintf(frecttidaltrace,"t %lf tri %i i2 %i wet %i\n",t,
    k,i2,nodewet[i2]);*/
  /*Initial wetting scheme*/
   if (dn[i2]<=-d[i2])
  {dn[i2]=-d[i2];
   un[i2]=0.0;
   wn[i2]=0.0;
   nodewet[i2]=0;};

   /*2nd wetting scheme*/
  /*if (dn[i2]<=-d[i2]+mindepth)
  {dn[i2]=-d[i2];
   un[i2]=0.0;
   wn[i2]=0.0;
   nodewet[i2]=0;};*/
   /*fprintf(frecttidaltrace,"wcheck t %lf tri %i i2 %i wet %i\n",
   t,k,i2,nodewet[i2]);*/
   };

/*Case 3: See whether node i1, which was
   previously  dry, wets */

 if ((nwtotal==2)&&(nodewet[i2]==1)&&(nodewet[i3]==1))
```

```
{
 i123=i2;
 if (dn[i3]>dn[i2]) i123=i3;
 /*Calculating qtauwetnode, the quadratic friction factor
for node i123*/
/*depthwetnode=dn[i123]+d[i123];*/
/*if (depthwetnode<hmin)
    depthwetnode=hmin;
qtauwetnode=(1.0)/(3.0)*(nmann*nmann)*9.81*pow(depthwetnode,(
-4.0,3.0));*/
/*fprintf(frecttidaltrace,"triangle %i  qtauwetnode %lf time %lf
 \n",k,qtauwetnode,t);*/
/*tauwetnode=qtauwetnode;*/
/*quadratic friction is used in the model*/
tauwetnode=ltau;/*linear friction is used in the model*/
/*Initial wetting scheme:
calculating a value for the water elevation at the dry node, i1.
If this value is greater than -h the node wets;
otherwise it remains dry. */

dn[i1]=dn[i123]-((x[i1]-x[i123])/9.81)*((un[i123]-u[i123])/dt+
    tauwetnode*un[i123])
    -((y[i1]-y[i123])/9.81)*((wn[i123]-w[i123])/dt+
    tauwetnode*wn[i123]);

un[i1]=un[i123];
wn[i1]=wn[i123];
twet[i1]=0;
if (dn[i1]>-d[i1]){twet[i1]=1;};

/*2nd wetting scheme:Calculating whether node i123 is
sufficiently high above node i1 for the latter node to dry*/
/*twet[i1]=0;
if (dn[i123]-dn[i1]>zdiff)
{dn[i1]=-d[i1]+mindepth;
un[i1]=un[i123];
wn[i1]=wn[i123];
twet[i1]=1;};*/

/*fprintf(frecttidaltrace,"t %lf i1 %i i123 %i dni123 %lf  dni1
% lf di1 %lf xi1 %lf xi123 %lf\n",
 t,i1,i123,dn[i123],dn[i1],d[i1],x[i1],x[i123]);
```

```
            fprintf(frecttidaltrace,"t %lf uni123 %lf ui123 %lf dt %lf\n",
            t,un[i123],u[i123],dt);
            fprintf(frecttidaltrace,"t %lf tri %i i1 %i wet %i\n",t,k,i1,
            nodewet[i1]);*/
            /*Initial  wetting scheme*/
           if (dn[i1]<=-d[i1])
           {dn[i1]=-d[i1];
           un[i1]=0.0;
           wn[i1]=0.0;
           nodewet[i1]=0;};

           /*2nd wetting scheme*/
           /*if (dn[i1]<=-d[i1]+mindepth)
           {dn[i1]=-d[i1];
           un[i1]=0.0;
           wn[i1]=0.0;
           nodewet[i1]=0;};*/
           /*fprintf(frecttidaltrace,"wcheck t %lf tri %i i1 %i wet %i\n",
           t,k,i1,nodewet[i1]);*/
           };
       };
   }
/* void shallow()*/
   /*set the total depth and velocity to zero if the water is
   calculated to be under the sea bed*/

/* {
 for(k=0;k<n;k++){
 ds=d[k]+dn[k];
 if(ds<=0)
 {un[k]=0;
 wn[k]=0;
 dn[k]=-d[k];
 };
 if (t==0.5*dt) {fprintf(frecttidaltrace,"%lf %i %lf %lf %lf \n",
  0.5,k,dn[k],un[k],wn[k]);};
 if (t==1.0*dt) {fprintf(frecttidaltrace,"%lf %i %lf %lf %lf \n",
  1.0,k,dn[k],un[k],wn[k]);};
 if (t==1.5*dt) {fprintf(frecttidaltrace,"%lf %i %lf %lf %lf \n",
  1.5,k,dn[k],un[k],wn[k]);};
 if (t==2.0*dt) {fprintf(frecttidaltrace,"%lf %i %lf %lf %lf \n",
  2.0,k,dn[k],un[k],wn[k]);};
```

```
if (t==2.5*dt) {fprintf(frecttidaltrace,"%lf %i %lf %lf %lf \n",
 2.5,k,dn[k],un[k],wn[k]);};
if (t==10.0*dt) {fprintf(frecttidaltrace,"%lf %lf %lf %lf \n",
 10.0,dn[k],un[k],wn[k]);};
if (t==60.0*dt) {fprintf(frecttidaltrace," %lf %.9lf %.9lf %.9lf \n",
 60.0,dn[k],un[k],wn[k]);};
if (t==70.0*dt) {fprintf(frecttidaltrace,"%lf %.9lf %.9lf %.9lf \n",
 70.0,dn[k],un[k],wn[k]);};
if (t==80.0*dt) {fprintf(frecttidaltrace,"%lf %.9lf %.9lf %.9lf \n",
 80.0,dn[k],un[k],wn[k]);};
if (t==90.0*dt) {fprintf(frecttidaltrace,"%lf %.9lf %.9lf %.9lf \n",
 90.0,dn[k],un[k],wn[k]);};
if (t==100.0*dt) {fprintf(frecttidaltrace,"%lf %.9lf %.9lf %.9lf \n",
 100.0,dn[k],un[k],wn[k]);};
if (t==110.0*dt) {fprintf(frecttidaltrace,"%lf %.9lf %.9lf %.9lf \n",
 110.0,dn[k],un[k],wn[k]);};
if (t==120.0*dt) {fprintf(frecttidaltrace,"%lf %.9lf %.9lf %.9lf \n",
 120.0,dn[k],un[k],wn[k]);};
};

}*/

 void answer()

 {for (k=0;k<n;k++)
 {u[k]=un[k];
 w[k]=wn[k];
 zeta[k]=dn[k];
 };
 }


void disko()

{
if(iq==nstep+1) goto jdi1;
if (iprog!=0) goto jk1;
fprintf(frecttidalcalc,"\nThe Solution");
fprintf(frecttidalcalc," after %lf secs, after time step %i\n",t,iq);
fprintf(frecttidalcalc,"Node Height   U vel   V vel\n\n");
/*if (t==3.5*12.4*3600.0)
{fprintf(fzuvout,"\nThe Solution for tidal flow in ");
```

```
fprintf(fzuvout,"Port Phillip Bay\n");
fprintf(fzuvout,"using n=0.045\n");
fprintf(fzuvout,"after %lf secs, after time step %i\n\n",t,iq);
fprintf(fzuvout,"Node Height   U vel   V vel\n\n");};*/
for(k=0;k<n;k++){
    fprintf(frecttidalcalc,"%i %lf %lf %lf\n",k,zeta[k],u[k],w[k]);};
 /*if (iq==27524)
 for(k=0;k<n;k++){
  {fprintf(fzuvout,"%i %lf %lf %lf\n",k,zeta[k],u[k],w[k]);};};*/

if (iprog==0) goto jdi1;

jk1:;

    if (iprog==2) goto kl0;
    if (iprog==3) goto kl1;
    if (iprog==4) goto kl2;
    if (iprog==5) goto kl3;
    if (iprog==6) goto k14;
    if (iprog==7) goto k15;
    if (iprog==8) goto k16;
    if (iprog==9) goto k17;
    if (iprog==10) goto k18;

    /* calculating u,v or zeta at a single node; this command
    needs to be changed depending on whether u, v or
    zeta is required and
    depending on the node it is required at;
    note that u, v or zeta also has to be set for t=0:
    this command is in the main function at the bottom of p.10 */

    exptlzeta[174]=a6*cos(pai/tt1*t-ph6*pai/360)+a7*cos(pai/tt2*t
    -ph7*pai/360)+a8*cos(pai/tt3*t-ph8*pai/360)+a9*cos(pai/tt4*t
    -ph9*pai/360)+a10*cos(pai/tt5*t-ph10*pai/360);
    modelzeta[174]=a11*cos(pai/tt1*t-ph11*pai/360)+
        a12*cos(pai/tt2*t-ph12*pai/360)
    +a13*cos(pai/tt3*t-ph13*pai/360)+a14*cos(pai/tt4*t-ph14*pai/360)
    +a15*cos(pai/tt5*t-ph15*pai/360);
    for(k=2462;k<2463;k++){fprintf(frecttidalcalc,"%lf %lf\n",t,u[k]);};
    for(k=2565;k<2566;k++){fprintf(ftidalday,"%lf %lf\n",t/(24*3600),
    zeta[k]);};
    for(k=174;k<175;k++){fprintf(fexptltidal,"%lf %lf\n",t/(24*3600),
    exptlzeta[k]);};
    for(k=174;k<175;k++){fprintf(fmodeltidal,"%lf %lf\n",t/(24*3600),
    modelzeta[k]);};
    /*for(k=156;k<157;k++)
    {fprintf(frecttidalcalc,"%lf %lf\n",t,zeta[k]);};*/
    /*for(k=156;k<157;k++){fprintf(frecttidalcalc,"%lf %lf\n",t,u[k]);};*/
    /*for(k=156;k<157;k++){fprintf(frecttidalcalc,"%lf %lf\n",t,w[k]);};*/

    if (iprog==1) goto jdi1;

    /* calculating u,v and zeta at a number of nodes */

    kl0:;

    for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,u[k]);};
    for (k=33;k<34;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=34;k<35;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=67;k<68;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=68;k<69;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=8;k<9;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=25;k<26;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=42;k<43;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=59;k<60;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=76;k<77;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=16;k<17;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=17;k<18;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=50;k<51;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=51;k<52;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
    for (k=84;k<85;k++) {fprintf(frecttidalcalc," %lf\n",u[k]);};

    if (iprog==2) goto jdi1;

    kl1:;

    for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,w[k]);};
    for (k=33;k<34;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
    for (k=34;k<35;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
    for (k=67;k<68;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
    for (k=68;k<69;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
    for (k=8;k<9;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
    for (k=25;k<26;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
    for (k=42;k<43;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
```

```
 for (k=59;k<60;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
 for (k=76;k<77;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
 for (k=16;k<17;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
 for (k=17;k<18;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
 for (k=50;k<51;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
 for (k=51;k<52;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
 for (k=84;k<85;k++) {fprintf(frecttidalcalc," %lf\n",w[k]);};
 if (iprog==3) goto jdi1;

 kl2:;

 for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
 for (k=33;k<34;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=34;k<35;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=67;k<68;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=68;k<69;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=8;k<9;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=25;k<26;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=42;k<43;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=59;k<60;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=76;k<77;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};

 if (iprog==4) goto jdi1;

kl3:;

for (k=174;k<175;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
for (k=331;k<332;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=552;k<553;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=578;k<579;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=637;k<638;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=647;k<648;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=1059;k<1060;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=1166;k<1167;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=1268;k<1269;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};

if (iprog==5) goto jdi1;

k14:;

/*for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
for (k=1;k<8;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
```

```
 for (k=8;k<9;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};*/

 /* For Port Phillip Bay Version 8:
  dt=12.5 secs and istep=288 if print output every hour */

 /*for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
 for (k=1;k<n-1;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=n-1;k<n;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};*/

 /* For Quadratic Bed:
   istep=1 if print output every time step; print
   output for nodes 264 to 296 for 1st mesh;
   776 to 872 for 2nd mesh;
   2312 to 2600 for 3rd mesh*/

/*for (k=264;k<265;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
 for (k=265;k<296;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=296;k<297;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};


for (k=264;k<265;k++) {fprintf(frecttidalcalc,"% lf %lf",t,un[k]);};
 for (k=265;k<296;k++) {fprintf(frecttidalcalc," %lf",un[k]);};
 for (k=296;k<297;k++) {fprintf(frecttidalcalc," %lf\n",un[k]);};

 for (k=264;k<265;k++) {fprintf(frecttidalcalc,"% lf %lf",t,wn[k]);};
 for (k=265;k<296;k++) {fprintf(frecttidalcalc," %lf",wn[k]);};
 for (k=296;k<297;k++) {fprintf(frecttidalcalc," %lf\n",wn[k]);};


for (k=264;k<265;k++) {fprintf(frecttidaltrace,"disko t %lf  i %i %i",
t,k,nodewet[k]);};
 for (k=265;k<296;k++) {fprintf(frecttidaltrace," %i  %i",
 k,nodewet[k]);};
 for (k=296;k<297;k++) {fprintf(frecttidaltrace,"  %i  %i\n",k,nodewet[k]);
 */

 /*for (k=776;k<777;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
 for (k=777;k<872;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=872;k<873;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};


for (k=776;k<777;k++) {fprintf(frecttidalcalc,"% lf %lf",t,un[k]);};
```

```
 for (k=777;k<872;k++) {fprintf(frecttidalcalc," %lf",un[k]);};
 for (k=872;k<873;k++) {fprintf(frecttidalcalc," %lf\n",un[k]);};

 for (k=776;k<777;k++) {fprintf(frecttidalcalc,"% lf %lf",t,wn[k]);};
 for (k=777;k<872;k++) {fprintf(frecttidalcalc," %lf",wn[k]);};
 for (k=872;k<873;k++) {fprintf(frecttidalcalc," %lf\n",wn[k]);};

for (k=776;k<777;k++) {fprintf(frecttidaltrace,"disko t %lf  i %i %i",
t,k,nodewet[k]);};
 for (k=777;k<872;k++) {fprintf(frecttidaltrace," %i  %i",k,
 nodewet[k]);};
 for (k=872;k<873;k++) {fprintf(frecttidaltrace,"  %i  %i\n",k,
 nodewet[k]);};*/

 for (k=2312;k<2313;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
 for (k=2313;k<2600;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=2600;k<2601;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};

for (k=2312;k<2313;k++) {fprintf(frecttidalcalc,"% lf %lf",t,un[k]);};
 for (k=2313;k<2600;k++) {fprintf(frecttidalcalc," %lf",un[k]);};
 for (k=2600;k<2601;k++) {fprintf(frecttidalcalc," %lf\n",un[k]);};

 for (k=2312;k<2313;k++) {fprintf(frecttidalcalc,"% lf %lf",t,wn[k]);};
 for (k=2313;k<2600;k++) {fprintf(frecttidalcalc," %lf",wn[k]);};
 for (k=2600;k<2601;k++) {fprintf(frecttidalcalc," %lf\n",wn[k]);};

for (k=2312;k<2313;k++) {fprintf(frecttidaltrace,
"disko t %lf  i %i %i",t,k,nodewet[k]);};
 for (k=2313;k<2600;k++) {fprintf(frecttidaltrace," %i  %i",k,
 nodewet[k]);};
 for (k=2600;k<2601;k++) {fprintf(frecttidaltrace,"  %i  %i\n",k,
 nodewet[k]);};

 if (iprog==6) goto jdi1;

 k15:;

/* istep=288 if print output every hour */
  istep=288;

 for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,u[k]);};
 for (k=1;k<n-1;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
```

```
   for (k=n-1;k<n;k++) {fprintf(frecttidalcalc," %lf\n",u[k]);};

    if (iprog==7) goto jdi1;

   k16:;

  /* istep=288 if print output every hour */
    istep=288;

   for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,w[k]);};
   for (k=1;k<n-1;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
   for (k=n-1;k<n;k++) {fprintf(frecttidalcalc," %lf\n",w[k]);};

    if (iprog==8) goto jdi1;

  k17:;

  /*the below command prints tidal level, u-velocity and v-velocity
    at a given time at all nodes to the file zuvout;
    iq = 26240 for a High Tide at Pt.Lonsdale
    iq = 27840 for a Low Tide at Pt.Lonsdale
    iq = 27254 for a Zero Tide at Pt.Lonsdale
    iq = 27130 for a Zero Velocity at Pt. Lonsdale*/

  if (iq==27130)
  for(k=0;k<n;k++){
  {fprintf(fzuvout,"%i %lf %lf %lf\n",k,zeta[k],u[k],w[k]);};};
   if (iprog==9) goto jdi1;

  k18:;

  for(k=174;k<175;k++){
    fprintf(frecttidalcalc,"%lf %lf %lf %lf\n",t/(24*3600),zeta[k],u[k],
    w[k]);};
  if (iprog==10) goto jdi1;


  jdi1: ;
}

void principalloop()
```

```
/*counting the number of iterations (full time step)*/

{iq=iq+1;

/*increasing the time by half a time step*/

t=t+dt/2.0;

/*dtt is set to half a time step*/

dtt=dt/2.0;
printf("time in seconds=%f\n",t);

vector();/*calls the function vector
 which does most of the calculations*/
/*printf("time after 1st vector %lf\n",t);*/

for (k=0;k<n;k++)
{
    /*if (t==1.5*dt){fprintf(frecttidaltrace," node %i time %lf dn %lf
      \n",k,t,dn[k]);};*/
};

boundary();

/*for (k=0;k<n;k++)
{
  if (t==1.5*dt){fprintf(frecttidaltrace," node %i time %lf dn %lf   \n",
      k,t,dn[k]);};
};*/

 semidryhalftimestep();

 for (k=0;k<n;k++)
{
  /*if (t==0.5*dt) {fprintf(frecttidaltrace,"%lf  %i %i %lf  \n",
      0.5,k,nodewet[k],dn[k]);};*/
};

/*shallow();*/

t=t+dt/2.0;/*increase the time by half a time interval*/
```

```
 dtt=dt;
 printf("time in seconds=%f\n",t);
 /*printf("zeta at node 176 =%f\n",zeta[176]);*/

 vector();/*calls the function vector
  for the second half of the time interval*/

 boundary();

dryfulltimestep();

/*Calculating which triangles are wet and which are dry;
 a triangle is wet if all nodes are wet - otherwise it is dry */
 for (k=0; k<n1;k++){
 i1=ii[k][0];
 i2=ii[k][1];
 i3=ii[k][2];
 nd1=nodewet[i1];
 nd2=nodewet[i2];
 nd3=nodewet[i3];
 trianglewet[k]=nd1*nd2*nd3;};

 for (i=0; i<n;i++)
 {  (twet[i]=0)
 ;};

 wettingcheck();

/* shallow();*/

for (k=0;k<n;k++)
{

   /*if (t==1.0*dt) {fprintf(frecttidaltrace,"%lf  %i %i %lf  \n",
       1.0,k,nodewet[k],dn[k]);};
 if (t==1000.0*dt) {fprintf(frecttidaltrace,"%lf %i %i %lf %lf \n",
       1000.0,k,nodewet[k],dn[k],zeta[k]);};*/};

 /*Any node i that had twet[i] set to 1 in wettingcheck function
 has nodewet[i] set to 1;*/
 /*for (k=264;k<265;k++) {fprintf(frecttidaltrace,
 "beforetemp t %lf  i %i %i",t,k,nodewet[k]);};
```

```
for (k=265;k<296;k++) {fprintf(frecttidaltrace,"  %i  %i",k,nodewet[k]);};
for (k=296;k<297;k++) {fprintf(frecttidaltrace," %i  %i\n",k,nodewet[k]);};

 for (k=264;k<265;k++) {fprintf(frecttidaltrace,
 "beforetemp t %lf  i %i %i",t,k,twet[k]);};
for (k=265;k<296;k++) {fprintf(frecttidaltrace,"  %i  %i",k,twet[k]);};
for (k=296;k<297;k++) {fprintf(frecttidaltrace," %i  %i\n",k,twet[k]);};
*/

/*for (i=0; i<n;i++)
{ fprintf(frecttidaltrace,"beforeiftemp t %lf i %i %i\n",t,i,nodewet[i]);
if (twet[i]==1){nodewet[i]=1;};
    fprintf(frecttidaltrace,"afteriftemp t %lf i %i %i\n",t,i,nodewet[i]);
};*/

  for (i=0;i<n;i++)
  {if (twet[i]==1)
  nodewet[i]=1;};
   /*for (k=264;k<265;k++) {fprintf(frecttidaltrace,
   "aftertwetconversion t %lf  i %i %i",t,k,nodewet[k]);};
   for (k=265;k<296;k++) {fprintf(frecttidaltrace," %i  %i",k,
   nodewet[k]);};
   for (k=296;k<297;k++) {fprintf(frecttidaltrace,"  %i  %i\n",k,
   nodewet[k]);};
   */

/*for (k=264;k<265;k++) {fprintf(frecttidaltrace,
"aftertemp t %lf  i %i %i",t,k,nodewet[k]);};
for (k=265;k<296;k++) {fprintf(frecttidaltrace,"  %i  %i",k,nodewet[k]);};
for (k=296;k<297;k++) {fprintf(frecttidaltrace," %i  %i\n",k,nodewet[k]);};*/

/*Calculating which triangles are wet and which are dry;
a triangle is wet if all nodes are wet - otherwise it is dry */
for (k=0; k<n1;k++){
i1=ii[k][0];
i2=ii[k][1];
i3=ii[k][2];
nd1=nodewet[i1];
nd2=nodewet[i2];
nd3=nodewet[i3];
trianglewet[k]=nd1*nd2*nd3;};
```

```
answer();

iqfr=iq/istep;
idisk=iq-iqfr*istep;
if (idisk==0)
{disko();};

}

  void main()

{/* the data for tidal flow in a sample bay*/

n=4913;/*n is the number of nodes;
401 for Version 1 Port Phillip Bay
and 598 for Version 2 Port Phillip Bay
and 1506 for Versions 3 and 4 Port Phillip Bay
and 1475 for Version 5 Port Phillip Bay
and 1429 for Version 6 Port Phillip Bay
and 1504 for Versions 8 9 and 10 Port Phillip Bay
and 1505 for Version 11 Port Phillip Bay
and 561 for rectangular basin no. 5
and 1649 for rectangular basin nos. 6 and 7
and 4913 for rectangular basin no. 8*/
n1=9216;/*n1 is the number of triangular elements;
672 for version 1 Port Phillip Bay
and 1056 for Version 2 Port Phillip Bay
and 2769 for Versions 3 and 4 Port Phillip Bay
and 2710 for Version 5 Port Phillip Bay
and 2618 for Version 5 Port Phillip Bay
and 2739 for Versions 8 9 and 10 Port Phillip Bay
and 2741 for Version 11 Port Phillip Bay
and 1024 for for rectangular basin no. 5
and 3072 for rectangular basin no.6 and 7
and 9216 for rectangular basin no. 8*/
n4=17;/*n4 is the number of nodes on the open sea boundary
9 for Port Phillip Bay Versions 1 and 2
and 26 for Port Phillip Bay Versions 3 and 4 and 5 and 6 and 8
and 17 for rectangular basins no. 5 6 and 8*/
ml2[1]=81;/*ml2[1] is the number of nodes on the land boundary;
119 for Version 1 Port Phillip Bay
and 129 for Version 2 Port Phillip Bay
and 210 for Versions 3 and 4 and 5  and 6 Port Phillip Bay
and 225 for Version 8 Port Phillip Bay
and 81 for for rectangular basin no. 5*/
m2=1;/*m2 is the number of boundaries for land*/
m3=2;/*m3 is the number of islands*/
xbai=0.432;/*xbai is the scale factor for the x-axis, equal
to 1000.0 for Port phillip Bay and
12.0 for rectangular basins for period of 12 hours
and 0.432 for forced flow in a quadratic bed*/
ybai=0.432;/*ybai is the scale factor for the y-axis,  equal
to 1000.0 for Port Phillip Bay and
12.0 for rectangular basins for period of 12 hours
and 0.432 for forced flow in a quadratic bed*/
dbai=1.0;//*dbai is the scale factor for sea depth*/
ni[1]=5;/*ni[1] is the number of nodes on the island boundary;
4 for Versions 1 and 2 Port Phillip Bay and
5 for Versions 3 and 4 and 5Port Phillip Bay */
frecttidaltrace=fopen("recttidaltrace","w");

a=0.44;/*a is the tidal amplitude in metres */
aq=3000.0;/*aq is the radius in metres of the quadratic bed*/
b=2.0;/*b is related to the amplitude of the forcing
    function for a quadratic bed*/
h0=10.0;/*h0 is the maximum depth of the quadratic bed*/
pai=6.283185307;/*pai equals twice pi*/
/*a1=0.467;*/
/*a1 is the tidal amplitude of M2 in metres
        at the open sea boundary:
        0.467 in WS (Written Submission),
        0.7 in Black's model;
        or alternatively the amplitude of the
        forcing function for a quadratic bed*/
a1=-(b*b)/(4*9.81);/*a1 is the amplitude of the
        forcing function for a quadratic bed*/
a2=0.0;/*a2 is the tidal amplitude of S2 in metres
         at the open sea boundary;
         0.135 in WS;
         0.195 at Lorne;
         */
a3=0.0;/*a3 is the tidal amplitude of K1 in metres
         at the open sea boundary;
         0.145 in WS;
         0.210 at Lorne*/
a4=0.0;/*a4 is the tidal amplitude of O1 in metres
         at the open sea boundary;
         0.102 in WS;
         0.144 at Lorne*/
a5=0.0;/*a5 is the tidal amplitude of N2 in metres
         at the open sea boundary;
         0.075 in WS;
         0.116 at Lorne*/
a6=0.44;/*a6 is the observed tidal amplitude of M2 in metres
        at a selected point*/
a7=0.127;/*a7 is the observed tidal amplitude of S2 in metres
         at a selected point*/
a8=0.145;/*a8 is the observed tidal amplitude of K1 in metres
         at a selected point*/
a9=0.102;/*a9 is the observed tidal amplitude of O1 in metres
         at a selected point*/
a10=0.085;/*a10 is the observed tidal amplitude of N2 in metres
          at a selected point*/
a11=0.440;/*a11 is the modelled tidal amplitude of M2 in metres
        at a selected point*/
a12=0.124;/*a12 is the modelled tidal amplitude of S2 in metres
         at a selected point*/
a13=0.144;/*a13 is the modelled tidal amplitude of K1 in metres
         at a selected point*/
a14=0.104;/*a14 is the modelled tidal amplitude of O1 in metres
         at a selected point*/
a15=0.088;/*a15 is the modelled tidal amplitude of N2 in metres
          at a selected point*
tt=12.42;/*tt is the cycle of the tide (in hours)*/
tt=tt*3600.0;/*tt in seconds*/
/*tt1=12.42;*/
tt1=(pai*aq/7200)/sqrt(2*9.81*h0);
/*fprintf(frecttidaltrace," tt1  %lf   \n",
      tt1);*/
    /*tt1 is the period of M2 in hours
          or alternatively the period of the
        forcing function in hours (and half the period of the motion)
        for a quadratic bed*/
tt1=tt1*3600.0;/*tt1 in seconds*/
/*fprintf(frecttidaltrace," tt1  %lf   \n",
      tt1);*/
```

```
tt2=12.0;/*tt2 is the tperiod of S2 in hours */
tt2=tt2*3600.0;/*tt2 in seconds*/
tt3=23.93;/*tt3 is the periode of K1 in hours */
tt3=tt3*3600.0;/*tt3 in seconds*/
tt4=25.82;/*tt4 is the period of O1 in hours */
tt4=tt4*3600.0;/*tt4 in seconds*/
tt5=12.66;/*tt5 is the periode of N2 in hours */
tt5=tt5*3600.0;/*tt5 in seconds*/
ph1=0.0;/*ph1 is the tidal phase of M2 in degrees at open sea boundary;
          322.0 in WS;
          322.6 in Black's model*/
ph2=0.0;/*ph2 is the tidal phase of S2 in degrees at open sea boundary;
         90.0 in WS;
         87.6 at Lorne*/
ph3=0.0;/*ph3 is the tidal phase of K1 in degrees at open sea boundary;
        85.0 in WS;
        56.5 at Lorne*/
ph4=0.0;/*ph4 is the tidal phase of O1 in degrees at open sea boundary;
         30.0 in WS;
         26.1 at Lorne*/
ph5=0.0;/*ph5 is the tidal phase of N2 in degrees at open sea boundary;
          270.0 in WS;
          271.3 at Lorne*/
ph6=326.0;/*ph6 is the observed tidal phase of M2 in degrees at
a selected point*/
ph7=97.4;/*ph7 is the observed tidal phase of S2 in degrees at
a selected point*/
ph8=70.8;/*ph8 is the observed tidal phase of K1 in degrees at
a selected point*/
ph9=38.8;/*ph9 is the observed tidal phase of O1 in degrees at
a selected point*/
ph10=280.3;/*ph10 is the observed tidal phase of N2 in degrees at
 a selected point*/
ph11=319.9;/*ph11 is the modelled tidal phase of M2 in degrees at
 a selected point*/
ph12=98.8;/*ph12 is the modelled tidal phase of S2 in degrees at
 a selected point*/
ph13=70.9;/*ph13 is the modelled tidal phase of K1 in degrees at
 a selected point*/
ph14=40.8;/*ph14 is the modelled tidal phase of O1 in degrees at
 a selected point*/
ph15=281.2;/*ph15 is the modelled tidal phase of N2 in degrees at
```

```
  a selected point*/
/*dt=12.5;*//*dt is the time interval (in secs) for calculations*/
 dt=tt1/3456.0;/*dt for a quadratic bed*/
 /*fprintf(frecttidaltrace," dt %lf\n",dt);*/

/*set the value of tlimit*/
 /*tlimit=10.0;*//*tlimit is the no. of days
 (u,v,zeta) are calculated for */
 /*tlimit=12.4*10.0;*//* tlimit is the no. of hours (u,v,zeta) are
  calculated for */
 /*tlimit=32*24*3600;*/
 tlimit=2*tt1;/*tlimit is the no. of secs. (u,v,zeta) are calculated
 for */
 fprintf(frecttidaltrace," period %lf\n",tlimit);
 /*printf("period %lf\n",tlimit);*/

 nstep=floor(tlimit/dt);/* the number of time steps in tlimit */
 /*set the value of eps*/
 eps=0.95; /* the lumping parameter epsilon*/
 ae=2+2*eps;
 be=1-eps;
 ltau=0.000000;/* the linear friction factor, in units of (sec)^(-1) */
 nmann=0.020; /* nmann is the Manning friction factor;
               0.02 in Port Phillip Bay for dt=13.5 s or 12.5 s
               for version 6 elements for the best results*/
 adv=1.0;/*adv is set to 1.0 for advective flow
            and 0.0 for nonadvective flow;
            0.0 in Port Phillip Bay*/
 visc=0.0;/*visc is set to  1.0 for turbulent  flow and
 0.0 for nonturbulent  flow;
 1.0 in Port Phillip Bay*/
 turb=6.0;/*turb is the turbulent viscosity coefficient in
          square metres per second;
          6.0 in Port Phillip Bay*/
 hmin=0.85;/*hmin is the minimum depth in metres used
 in the quadratic friction formula;
 0.85 in Port Phillip Bay*/

 zdiff=0.1;/*zdiff is the minimum height (in metres)
 that a wet node can be above a neighbouring dry node before the dry node
  wets*/
 mindepth=0.05;/*mindepth is the minimum depth of water(in metres)
```

```
for a wet node*/

 for (i=0; i<n;i++)
{  (twet[i]=0)
;};

/*set the value of iprog;
iprog=0 if (u,v,zeta) printed for all
nodes at every istep time intervals to recttidalcalc and
(u,v,zeta) printed to zuvout at one time interval;
iprog=1 if only one of
(u,v,zeta) printed at one node for every time interval;
iprog=2 if only u printed at 15 given nodes for every time interval;
iprog=3 if only v printed at 15 given nodes for every time interval;
iprog=4 if only zeta printed at 15 given nodes for every time interval;
iprog=5 if only zeta printed at 9 given nodes for every istep
time intervals;
iprog=6 if only zeta printed at all given nodes for every istep
time intervals (for Port Phillip Bay) or if  zeta, u and v printed
at all given nodes for every istep
time intervals for moving boundary flow above quadratic bed;
iprog=7 if only u printed at all given nodes for every istep
time intervals;
iprog=8 if only v  printed at all given nodes for every istep
time intervals;
iprog=9 if zeta, u, and v  printed for all nodes at one time step;
iprog=10 if zeta, u, and v  printed for one node at all time steps;
*/
/*setting the value of iprog*/
iprog=6;
/*printf("iprog %i\n,iprog);*/
istep=320;/*for iprog =0 : output printed out for every node
 every istep time steps*/
if (iprog==0) goto jk1;

istep=1;/* for iprog =1, 2, 3, 4 , 5, 6, 7, 8, 9, 10:
        set istep =1 initially in a lot of cases;
        for prog =5, 6, 7 or 8  istep reset later*/

jk1:;

/*frecttidaltrace=fopen("recttidaltrace","w");*/
```

```
 frecttidalcalc=fopen("recttidalcalc","w");
 ftidalday=fopen("tidalday","w");
 fexptltidal=fopen("exptltidal","w");
 fmodeltidal=fopen("modeltidal","w");
 if (iprog!=0) goto jk2;
  fprintf(frecttidaltrace,"Checking program is outputting okay\n");
  fprintf(frecttidalcalc,"Solution of the  2D shallow water wave equations
  \n");

 jk2:;

 if (iprog!=0) goto jk3;
 fprintf(frecttidalcalc,"solved by the finite element method\n\n");
 fprintf(frecttidalcalc,"The number of time steps is %i\n",nstep);
 fprintf(frecttidalcalc,"Each time step is %lf seconds\n",dt);
 fprintf(frecttidalcalc,"Output every %i steps\n",istep);

 jk3:;

 /*inputting the x,y and d coordinates from a file*/
 frectxyddata=fopen("rectxyddata","r");
 fxydout=fopen("xydout","w");
 fzuvout=fopen("zuvout","w");
 if (iprog!=0) goto jk4;
 fprintf(frecttidalcalc,"The number of nodes is %i\n",n);

 jk4:;

 for(i=0;i<n;i++){
 fscanf(frectxyddata,"%lf%lf%lf",&x[i],&y[i],&d[i]);
 /*adjusting the depths in Port Phillip Bay */
 /*original adjustment formula*/
 /*adj[i]=(1271.4165+9.39*x[i]-14.355*y[i])/1484.565;*/
 /*amended adjustment formula*/
 /*adj[i]=(1264.62+5.55*x[i]-12.279*y[i])/1484.565;*/
 /* adjustment terms for versions 1 and 2 */
/*if (i<=10) adj[i]=0.9;
 if ((114<=i) && (i<=156)) adj[i]=0.9;*/
 /* adjustment terms for version 2 only*/
 /*if ((401<=i) && (i<=418)) adj[i]=0.9;
 if (i==435) adj[i]=0.9;
 if (i==587) adj[i]=0.9;
 if (i==590) adj[i]=0.9;*/

 /*adjustment formula from 16 June onwards,
 with adj, in metres, constant; 0.85 in Port Phillip Bay,
 0.0 m for comparison with analytical solutions*/
 /*adj=0.85;*/
 adj=0.0;

 d[i]=d[i]+adj;
 x[i]=x[i]*xbai;
 y[i]=y[i]*ybai;
 d[i]=d[i]*dbai;
/* d[i]=10.0;*/
 d[i]=h0*(1-(x[i]*x[i])/(aq*aq));/*d[i] for a quadratic bed*/
 /*fprintf(frecttidaltrace," i %i d %lf\n",i,d[i]);*/

 if (iprog!=0) goto jk5;
 fprintf(frecttidalcalc,"The node %i is at (%.2lf,%.2lf) with
 depth %.2lf\n",i,x[i],y[i],d[i]);

 jk5:;

 fprintf(fxydout,"%i %.6lf %.6lf %.6lf\n",i,x[i],y[i],d[i]);
 };
 fclose(frectxyddata);
 fclose(fxydout);

 /*inputting the node numbers for each of the triangular elements*/

 frectnodedata=fopen("rectnodedata","r");
 if (iprog!=0) goto jk6;
 fprintf(frecttidalcalc,"\n");
 fprintf(frecttidalcalc,"The number of elements is %i\n",n1);

 jk6:;

 for(k=0;k<n1;k++){
 fscanf(frectnodedata,"%i%i%i",&ii[k][0],&ii[k][1],&ii[k][2]);
 /*ii[k][0]=ii[k][0]-1;
 ii[k][1]=ii[k][1]-1;
 ii[k][2]=ii[k][2]-1;*/
 if (iprog!=0) goto jk7;
```

```
fprintf(frecttidalcalc,"The element %i consists of nodes %i, %i and %i\n",
    k,ii[k][0],ii[k][1],ii[k][2]);

jk7:;

};

fclose(frectnodedata);

/*"inputting the node numbers for the land boundary*/
flanddata=fopen("landdata","r");
for(l=0;l<ml2[1];l++){
fscanf(flanddata,"%i",&ibu[l]);
/*ibu[l]=ibu[l]-1;*/
};
fclose(flanddata);

 /*"inputting the node numbers for the island boundary*/
fislanddata=fopen("islanddata","r");
for(l=0;l<ni[1];l++){
fscanf(fislanddata,"%i",&ilu[l]);
ilu[l]=ilu[l]-1;
};
fclose(fislanddata);

 /*"inputting the node numbers for the sea boundary*/
frectseadata=fopen("rectseadata","r");
for(k=0;k<n4;k++){
fscanf(frectseadata,"%i",&ibh[k]);
/*ibh[k]=ibh[k]-1;*/
};
fclose(frectseadata);

/*initialising values*/
for (k=0;k<n;k++){
am[k]=0;/*am[k] is (the total area of the triangles that node
 number k belongs to)/3*/
 un[k]=0;/*un[k] is u at each half-time step or
         time step*/
 wn[k]=0;/*vn[k] is v at each half-time step or
         time step*/
/* if (d[k]>0)
```

```
  dn[k]=0;
  else
  {dn[k]=-d[k];};*//* setting initial dn[k] for Port Phillip Bay
                tidal flow, where dn[k] is zeta at each half-time step or
          time step*/
 if (x[k]<2023.21)
  dn[k]=-(b*b)/(4*9.81)-(0.5*pai/tt1)*(b/9.81)*x[k];
  else
  {dn[k]=-d[k];};/* setting initial dn[k] for quadratic bed
                 flow, where dn[k] is zeta at each half-time step or
          time step*/
/* fprintf(frecttidaltrace,"k %i x %lf dn %lf\n",k,x[k],dn[k]);*/
  u[k]=0;/*u[k] is u at each time step*/
  w[k]=0;/*w[k] is v at each time step*/
  /*if (d[k]>0)
  {zeta[k]=0;
  nodewet[k]=1;}
  else
  { zeta[k]=-d[k];
  nodewet[k]=0;}*//*setting initial zeta[k] for Port Phillip Bay
                tidal flow, where zeta[k] is zeta at each time step*/
  /* setting initial zeta
  (i) for cosine forced quadratic bed for B=5 aq=3000 h0=10,
   use if (x[k]<2023.21)
  (ii) for cosine forced quadratic bed for B=2 aq=3000 h0=10,
   use if x[k]<2586.9*/
  if (x[k]<2586.9)
  {zeta[k]=-(b*b)/(4*9.81)-0.5*(pai/tt1)*(b/9.81)*x[k];

   nodewet[k]=1;}
  else
  {zeta[k]=-d[k];
  nodewet[k]=0;}/* setting initial zeta[k] for quadratic bed
   flow, where zeta[k] is zeta at each
   time step*/
/* fprintf(frecttidaltrace,"k %i x %lf zeta %lf\n",k,x[k],zeta[k]);*/
 /*fprintf(frecttidaltrace," tt1  %lf   \n",
      tt1);*/};

 printf("time in seconds=%f\n",t);
  if (iprog!=0) goto jk8;
   fprintf(frecttidalcalc,"\n");
 jk8:;

 /*Calculating which triangles are wet and which are dry;
a triangle is wet if all nodes are wet - otherwise it is dry */
for (k=0; k<n1;k++){
i1=ii[k][0];
i2=ii[k][1];
i3=ii[k][2];
nd1=nodewet[i1];
nd2=nodewet[i2];
nd3=nodewet[i3];
trianglewet[k]=nd1*nd2*nd3;};
/*setting the initial height on the sea boundary*/
if (iprog!=0) goto jk9;
fprintf(frecttidalcalc,"The number of sea boundary conditions is  %i \n",n4);

jk9:;

  for(k=0;k<n4;k++){
l=ibh[k];
dn[l]=a1*cos(-ph1*pai/360)+a2*cos(-ph2*pai/360)
    +a3*cos(-ph3*pai/360)+a4*cos(-ph4*pai/360)
    +a5*cos(-ph5*pai/360);
zeta[l]=a1*cos(-ph1*pai/360)+a2*cos(-ph2*pai/360)
    +a3*cos(-ph3*pai/360)+a4*cos(-ph4*pai/360)
    +a5*cos(-ph5*pai/360);
if (iprog!=0) goto jk10;
fprintf(frecttidalcalc,"The bc at node %i is z=%lf cos (2*pi)*(%lf^(-1))\n",
l,a,tt);

jk10:;

};
if (iprog!=0) goto jk11;
 fprintf(frecttidalcalc,"\n");

 fprintf(frecttidalcalc,"\nThe number of element boundary conditions
 is %i\n", ml2[1]-1);

 jk11:;
 for (k=0;k<ml2[1]-1;k++){
 if (iprog!=0) goto jk12;
     fprintf(frecttidalcalc,"The boundary condition from");
     fprintf(frecttidalcalc," node %i to node %i is nvel=zero\n",ibu[k],
     ibu[k+1]);

 jk12:;};

  if (iprog!=0) goto jk16;
  fprintf(frecttidalcalc,"\n");

 jk16:;

   /*calculating del(k), the area of each triangle, and am(ir), (the sum
   of the triangle areas a node forms part of)/3),for each node*/

  for(k=0;k<n1;k++){
  i1=ii[k][0];
  i2=ii[k][1];
  i3=ii[k][2];
  delk=x[i2]*y[i3]+x[i1]*y[i2]+x[i3]*y[i1];
  del[k]=(delk-(x[i3]*y[i2]+x[i1]*y[i3]+x[i2]*y[i1]))*.5;
 /* fprintf(frecttidaltrace,"%f  \n",
 del[k]);*/
  /*delt=del[k]/3.0;
              for(im=0;im<3;im++){
                  ir=ii[k][im];
                  am[ir]=am[ir]+delt;
                  };*/
  };
  if (iprog!=0) goto jk13;
  fprintf(frecttidalcalc,"\nThe selective lumped mass parameter = %lf\n",
  eps);
  /*fprintf(frecttidalcalc,"\nThe linear friction factor = %lf\n",ltau);*/
  fprintf(frecttidalcalc,"\nThe Manning friction factor = %lf\n",nmann);

 jk13:;

/*initialising values*/

iq=0; /*iq is the number of iterations*/
t=0;  /*t is the time in seconds*/
```

```
  if (iprog!=0) goto jk14;
  fprintf(frecttidalcalc,"\nThe Initial Values");
  fprintf(frecttidalcalc," after %lf secs, after time step %i\n",t,iq);
  fprintf(frecttidalcalc,"Node Height   U vel   V vel\n");

 for(k=0;k<n;k++){
  fprintf(frecttidalcalc,"%i %lf %lf %lf \n",k,zeta[k],u[k],w[k]);
  /*fprintf(fzuvout,"%i %lf %lf %lf\n",k,zeta[k],u[k],w[k]);*/
 };

  jk14:;

  if (iprog==0) goto jk15;
      if (iprog==2) goto mn0;
      if (iprog==3) goto mn1;
      if (iprog==4) goto mn2;
      if (iprog==5) goto mn3;
      if (iprog==6) goto mn4;
      if (iprog==7) goto mn5;
      if (iprog==8) goto mn6;
      if (iprog==9) goto mn7;
      if (iprog==10) goto mn8;

/* calculating u,v or zeta at a single node; this command
      needs to be changed depending on whether u, v or
      zeta is required and
      depending on the node it is required at
      */

     exptlzeta[174]=a6*cos(-ph6*pai/360)+a7*cos(-ph7*pai/360)
      +a8*cos(-ph8*pai/360)+a9*cos(-ph9*pai/360)
      +a10*cos(-ph10*pai/360);
     modelzeta[174]=a11*cos(-ph11*pai/360)+
          a12*cos(-ph12*pai/360)
      +a13*cos(-ph13*pai/360)+a14*cos(pai/tt4*t-ph14*pai/360)
      +a15*cos(-ph15*pai/360);
      for(k=2462;k<2463;k++){fprintf(frecttidalcalc,"%lf %lf\n",t,u[k]);};
      for(k=2447;k<2448;k++){fprintf(ftidalday,"%lf %lf\n",t/(24*3600),
      zeta[k]);};
      for(k=174;k<175;k++){fprintf(fexptltidal,"%lf %lf\n",t/(24*3600),
      exptlzeta[k]);};
      for(k=174;k<175;k++){fprintf(fmodeltidal,"%lf %lf\n",t/(24*3600),
modelzeta[k]);};
/*for(k=156;k<157;k++)
{fprintf(frecttidalcalc,"%lf %lf\n",t,zeta[k]);};*/
/*for(k=156;k<157;k++){fprintf(frecttidalcalc,"%lf %lf\n",t,u[k]);};*/
/*for(k=156;k<157;k++){fprintf(frecttidalcalc,"%lf %lf\n",t,w[k]);};*/

if (iprog==1) goto jk17;

 mn0:;

for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,u[k]);};
for (k=33;k<34;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=34;k<35;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=67;k<68;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=68;k<69;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=8;k<9;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=25;k<26;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=42;k<43;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=59;k<60;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=76;k<77;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=16;k<17;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=17;k<18;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=50;k<51;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=51;k<52;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
for (k=84;k<85;k++) {fprintf(frecttidalcalc," %lf\n",u[k]);};

if (iprog==2) goto jk17;

mn1:;

for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,w[k]);};
for (k=33;k<34;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=34;k<35;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=67;k<68;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=68;k<69;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=8;k<9;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=25;k<26;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=42;k<43;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=59;k<60;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=76;k<77;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=16;k<17;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=17;k<18;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=50;k<51;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=51;k<52;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
for (k=84;k<85;k++) {fprintf(frecttidalcalc," %lf\n",w[k]);};

if (iprog==3) goto jk17;

mn2:;

for(k=68;k<69;k++){fprintf(frecttidalcalc,"%lf %lf\n",t,zeta[k]);};
for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
for (k=33;k<34;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=34;k<35;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=67;k<68;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=68;k<69;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=8;k<9;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=25;k<26;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=42;k<43;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=59;k<60;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=76;k<77;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};

if (iprog==4) goto jk17;

mn3:;

/* dt=12.5 secs and istep=288 if print output every hour */

istep=288;
for (k=174;k<175;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
for (k=331;k<332;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=552;k<553;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=578;k<579;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=637;k<638;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=647;k<648;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=1059;k<1060;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=1166;k<1167;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
for (k=1268;k<1269;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};

if (iprog==5) goto jk17;

mn4:;

/* For Port Phillip Bay Version 8:
  dt=12.5 secs and istep=288 if print output every hour */

  /*istep=288;


 for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
 for (k=1;k<n-1;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=n-1;k<n;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};*/

/* For Quadratic Bed:
   istep=1 if print output every time step; print
   output for nodes 264 to 296 for 1st mesh;
   776 to 872 for 2nd mesh;
   2312 to 2600 for 3rd mesh*/

 istep=1;

 /*for (k=264;k<265;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
 for (k=265;k<296;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=296;k<297;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};


 for (k=264;k<265;k++) {fprintf(frecttidalcalc,"% lf %lf",t,un[k]);};
 for (k=265;k<296;k++) {fprintf(frecttidalcalc," %lf",un[k]);};
 for (k=296;k<297;k++) {fprintf(frecttidalcalc," %lf\n",un[k]);};

 for (k=264;k<265;k++) {fprintf(frecttidalcalc,"% lf %lf",t,wn[k]);};
 for (k=265;k<296;k++) {fprintf(frecttidalcalc," %lf",wn[k]);};
 for (k=296;k<297;k++) {fprintf(frecttidalcalc," %lf\n",wn[k]);};

 for (k=264;k<265;k++) {fprintf(frecttidaltrace,"init t %lf  i %i %i",
 t,k,nodewet[k]);};
 for (k=265;k<296;k++) {fprintf(frecttidaltrace,"  %i  %i",k,nodewet[k]);};
 for (k=296;k<297;k++) {fprintf(frecttidaltrace," %i  %i\n",k,
 nodewet[k]);};*/

 /*for (k=776;k<777;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
 for (k=777;k<872;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=872;k<873;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};


for (k=776;k<777;k++) {fprintf(frecttidalcalc,"% lf %lf",t,un[k]);};
```

```
 for (k=777;k<872;k++) {fprintf(frecttidalcalc," %lf",un[k]);};
 for (k=872;k<873;k++) {fprintf(frecttidalcalc," %lf\n",un[k]);};

 for (k=776;k<777;k++) {fprintf(frecttidalcalc,"% lf %lf",t,wn[k]);};
 for (k=777;k<872;k++) {fprintf(frecttidalcalc," %lf",wn[k]);};
 for (k=872;k<873;k++) {fprintf(frecttidalcalc," %lf\n",wn[k]);};


for (k=776;k<777;k++) {fprintf(frecttidaltrace,"disko t %lf  i %i %i",t,
k,nodewet[k]);};
 for (k=777;k<872;k++) {fprintf(frecttidaltrace," %i  %i",k,
 nodewet[k]);};
 for (k=872;k<873;k++) {fprintf(frecttidaltrace,"  %i  %i\n",k,
 nodewet[k]);};
 */

 for (k=2312;k<2313;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
 for (k=2313;k<2600;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
 for (k=2600;k<2601;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};


 for (k=2312;k<2313;k++) {fprintf(frecttidalcalc,"% lf %lf",t,
 un[k]);};
 for (k=2313;k<2600;k++) {fprintf(frecttidalcalc," %lf",un[k]);};
 for (k=2600;k<2601;k++) {fprintf(frecttidalcalc," %lf\n",un[k]);};

 for (k=2312;k<2313;k++) {fprintf(frecttidalcalc,"% lf %lf",t,wn[k]);};
 for (k=2313;k<2600;k++) {fprintf(frecttidalcalc," %lf",wn[k]);};
 for (k=2600;k<2601;k++) {fprintf(frecttidalcalc," %lf\n",wn[k]);};


for (k=2312;k<2313;k++) {fprintf(frecttidaltrace,"disko t %lf  i %i %i",
t,k,nodewet[k]);};
 for (k=2313;k<2600;k++) {fprintf(frecttidaltrace," %i  %i",k,
 nodewet[k]);};
 for (k=2600;k<2601;k++) {fprintf(frecttidaltrace,"  %i  %i\n",k,
 nodewet[k]);
 };
 /*for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,zeta[k]);};
  for (k=1;k<8;k++) {fprintf(frecttidalcalc," %lf",zeta[k]);};
  for (k=8;k<9;k++) {fprintf(frecttidalcalc," %lf\n",zeta[k]);};*/

  if (iprog==6) goto jk17;



 mn5:;

/* dt=12.5 secs and istep=288 if print output every hour */
  istep=288;

 for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,u[k]);};
 for (k=1;k<n-1;k++) {fprintf(frecttidalcalc," %lf",u[k]);};
 for (k=n-1;k<n;k++) {fprintf(frecttidalcalc," %lf\n",u[k]);};

  if (iprog==7) goto jk17;

 mn6:;

/* dt=12.5 secs and istep=288 if print output every hour */

  istep=288;

 for (k=0;k<1;k++) {fprintf(frecttidalcalc,"% lf %lf",t,w[k]);};
 for (k=1;k<n-1;k++) {fprintf(frecttidalcalc," %lf",w[k]);};
 for (k=n-1;k<n;k++) {fprintf(frecttidalcalc," %lf\n",w[k]);};

 if (iprog==8) goto jk17;

 mn7:;

if (iq==27130)
for(k=0;k<n;k++){
{fprintf(fzuvout,"%i %lf %lf %lf\n",k,zeta[k],u[k],w[k]);};}

if (iprog==9) goto jk17;

mn8:;

for(k=174;k<175;k++){
    fprintf(frecttidalcalc,"%lf %lf %lf %lf\n",t/(24*3600),zeta[k],
    u[k],w[k]);};

    if (iprog==10) goto jk17;


    jk15:;

    for (k=0;k<n;k++){fprintf(frecttidaltrace,"%lf %i %lf %lf %lf \n",
    0.0,k,dn[k],un[k],wn[k]);};

    jk17:;

    for (i4=0;i4<nstep;i4++){

    principalloop();

    };

    printf ("end of computation\n");
    printf("press return key");
    fclose(frecttidalcalc);
    fclose(fzuvout);
    fclose(ftidalday);
    fclose(frecttidaltrace);
    fclose(fexptltidal);
    fclose(fmodeltidal);

}
```

Appendix B

Visual C++ code for generating finite elements

```
/* Finite element mesh.cpp: a program to generate a finite
 element mesh of right angled isosceles triangles
 for a rectangular region of  length lh and width bh.
 The rectangle has p+1 nodes on its base ,which is in the x-direction,
and q+1 nodes along its left side,which is in the  y-direction.
Note: p and q have to be even. The triangles generated have their
isoceles sides parallel to the x- or y-directions. The program
generates (p+1)(q+1) nodes and 2pq triangles. The depth at the left
hand end is dleft and the depth at the right hand end is dright. The
values of dleft, dright, lh, p and q must be input. The value of bh
is calculated to ensure that the triangles are isosceles.

                Joe Sampson                               */

# include <stdio.h>
# include <math.h>
# include <cstdlib>

 double bh,dleft,dright,lh; int j,k,l,n,p,q;
 double x[4920],y[4920]; /* x and y coordinates [m] */
 double d[4920]; /* water depth [m] */
 int nodeno[9220][3]; /* nodenumber(triangleno.,vertex no.) */

 FILE *fddata;
 FILE *finodedata;
 FILE *fixyddata;
 FILE *flhdata;
 FILE *fnodedata;
 FILE *fpqdata;
 FILE *ftrace;
 FILE *fxyddata;

void type_alpha()
{for (l=0;l<q/2-1;l++){
        nodeno[j+4*p*(l+1)][0]=nodeno[j+4*p*l][0]+2*p+2;
        nodeno[j+4*p*(l+1)][1]=nodeno[j+4*p*l][1]+2*p+2;
        nodeno[j+4*p*(l+1)][2]=nodeno[j+4*p*l][2]+2*p+2;
}; }
void type_beta()
{for (l=0;l<q/2-1;l++){
        nodeno[j+4*(k+1)+4*p*(l+1)][0]=nodeno[j+4*(k+1)+4*p*l][0]+2*p+2;
        nodeno[j+4*(k+1)+4*p*(l+1)][1]=nodeno[j+4*(k+1)+4*p*l][1]+2*p+2;
        nodeno[j+4*(k+1)+4*p*(l+1)][2]=nodeno[j+4*(k+1)+4*p*l][2]+2*p+2;
                            };
}

void type_gamma()
{for (l=0;l<q/2-1;l++){
        nodeno[j-4*(k+1)+4*p*(l+1)][0]=nodeno[j-4*(k+1)+4*p*l][0]+2*p+2;
        nodeno[j-4*(k+1)+4*p*(l+1)][1]=nodeno[j-4*(k+1)+4*p*l][1]+2*p+2;
        nodeno[j-4*(k+1)+4*p*(l+1)][2]=nodeno[j-4*(k+1)+4*p*l][2]+2*p+2;
                            };
}

void type_a()

{type_alpha();

    for (k=0;k<p/2-1;k++){
    nodeno[j+4*(k+1)][0]=nodeno[j+4*k][0]-2;
    nodeno[j+4*(k+1)][1]=nodeno[j+4*k][1]+2;
    nodeno[j+4*(k+1)][2]=nodeno[j+4*k][2]-2;
        type_beta();
}; }

void type_b()

    {type_alpha();
```

```
  for (k=0;k<p/2-1;k++){
  nodeno[j+4*(k+1)][0]=nodeno[j+4*k][0]+2;
  nodeno[j+4*(k+1)][1]=nodeno[j+4*k][1]-2;
  nodeno[j+4*(k+1)][2]=nodeno[j+4*k][2]+2;
  type_beta();
}; }

void type_c()

  {type_alpha();

  for (k=0;k<p/2-1;k++){
  nodeno[j-4*(k+1)][0]=nodeno[j-4*k][0]-2;
  nodeno[j-4*(k+1)][1]=nodeno[j-4*k][1]+2;
  nodeno[j-4*(k+1)][2]=nodeno[j-4*k][2]-2;
    type_gamma();
}; }

void type_d()

  {type_alpha();

  for (k=0;k<p/2-1;k++){
  nodeno[j-4*(k+1)][0]=nodeno[j-4*k][0]+2;
  nodeno[j-4*(k+1)][1]=nodeno[j-4*k][1]-2;
  nodeno[j-4*(k+1)][2]=nodeno[j-4*k][2]+2;
  type_gamma();
}; }

/* given the x, y and d coordinates of 2 points in the mesh, the x,
y and d coordinates of some of the other points are generated using
the type_e function and for some of the other
 points by using the type_f function */

void type_e() /*type_e() generates the x, y and d coordinates on the
bottom row, on the row that is 2 rows above the bottom row, on the
row that is 4 rows above the bottom row, etc.*/ {   j=0;
  x[j]=0.0;
  y[j]=0.0;
  d[j]=dleft;
  for (n=0;n<q/2+1;n++){
    x[j+n*(2*p+2)]=x[j];
```

```
        y[j+n*(2*p+2)]=y[j]+n*2*lh/p;
        d[j+n*(2*p+2)]=d[j];
        for (k=0;k<p+1;k++){
            x[j+n*(2*p+2)+k]=x[j]+k*lh/p;
            y[j+n*(2*p+2)+k]=y[j+n*(2*p+2)];
            d[j+n*(2*p+2)+k]=d[j]+(dright-dleft)*k/p;};
                       };
}

void type_f() /*type_f() generates the x, y and d coordinates on the
bottom row, on the row that is 1 row above the bottom row, on the
row that is 3 rows above the bottom row, etc.*/ {      j=p+1;
    x[j]=lh;
    y[j]=lh/p;
    d[j]=dright;
    for (n=0;n<q/2;n++){
        x[j+n*(2*p+2)]=x[j];
        y[j+n*(2*p+2)]=y[j]+n*2*lh/p;
        d[j+n*(2*p+2)]=d[j];
        for (k=0;k<p+1;k++){
            x[j+n*(2*p+2)+k]=x[j]-k*lh/p;
            y[j+n*(2*p+2)+k]=y[j+n*(2*p+2)];
            d[j+n*(2*p+2)+k]=d[j]-(dright-dleft)*k/p;};
                       };
}

void main()

{       ftrace=fopen("trace","w");
        fddata=fopen("fddata","r");
        flhdata=fopen("flhdata","r");
        fpqdata=fopen("pqdata","r");


        fscanf(fddata,"%lf %lf",&dleft,&dright);
        fscanf(flhdata,"%lf",&lh);
        fscanf(fpqdata,"%i %i",&p,&q);

        bh=q*lh/p;

        j=0;
        nodeno[j][0]=2*p+1;
```

```
nodeno[j][1]=0;
nodeno[j][2]=2*p;
/*fprintf(ftrace,"before type a \n");*/
type_a();

j=1;
nodeno[j][0]=1;
nodeno[j][1]=2*p;
nodeno[j][2]=0;
type_b();

j=2;
nodeno[j][0]=2;
nodeno[j][1]=2*p;
nodeno[j][2]=1;
type_b();

j=3;
nodeno[j][0]=2*p;
nodeno[j][1]=2;
nodeno[j][2]=2*p-1;
type_a();

j=4*p-4;
nodeno[j][0]=2*p-1;
nodeno[j][1]=2*p+4;
nodeno[j][2]=2*p;
type_c();

j=4*p-3;
nodeno[j][0]=2*p+3;
nodeno[j][1]=2*p;
nodeno[j][2]=2*p+4;
type_d();

j=4*p-2;
nodeno[j][0]=2*p+2;
nodeno[j][1]=2*p;
nodeno[j][2]=2*p+3;
type_d();

j=4*p-1;
```

```
        nodeno[j][0]=2*p;
        nodeno[j][1]=2*p+2;
        nodeno[j][2]=2*p+1;
        type_c();

        type_e();

        type_f();

/* writing the nodedata and the x, y and d values to files */

        fxyddata=fopen("xyddata","w");
        fixyddata=fopen("ixyddata","w");
        fnodedata=fopen("nodedata","w");
        finodedata=fopen("inodedata","w");

        for (k=0;k<(p+1)*(q+1);k++){
            printf("nodeno=%i\n",k);
            fprintf(fxyddata,"%.9lf %.9lf %.9lf\n",x[k],y[k],d[k]);};

        for (k=0;k<(p+1)*(q+1);k++){
            fprintf(fixyddata,"%i %.9lf %.9lf %.9lf\n",k,x[k],y[k],d[k]);};

        for(k=0;k<2*p*q;k++){
            fprintf(fnodedata,"%i %i %i\n",nodeno[k][0],nodeno[k][1],
                nodeno[k][2]);};

        for(k=0;k<2*p*q;k++){
            fprintf(finodedata,"%i %i %i %i\n",k,nodeno[k][0],nodeno[k][1],
                nodeno[k][2]);};

        fclose(fddata);
        fclose(finodedata);
        fclose(fixyddata);
        fclose(flhdata);
        fclose(fnodedata);
        fclose(fpqdata);
        fclose(ftrace);
        fclose(fxyddata);

}
```

List of refereed publications

1. Sampson, J., Easton, A. and Singh, M., Moving Boundary Shallow Water Flow in Circular Paraboloidal Basins, *Proceedings of the Sixth Engineering Mathematics and Applications Conference, 5th International Congress on Industrial and Applied Mathematics, at the University of Technology, Sydney, Australia*, editors R. L.May and W. F. Blyth, 223-227, 2003.

2. Sampson, Joe, Easton, Alan and Singh, Manmohan, Modelling the effect of proposed channel deepening on the tides in Port Phillip Bay, *Australian and New Zealand Industrial and Applied Mathematics Journal*, **46** (E), C888-C901, 2005.
[Online] http://anziamj.austms.org.au/V46/CTAC2004/Samp.

3. Sampson, Joe, Easton, Alan and Singh, Manmohan, Moving Boundary Shallow Water Flow in Parabolic Bottom Topography, *Australian and New Zealand Industrial and Applied Mathematics Journal*, **47** (EMAC), C373-387, 2006.
[Online] http://anziamj.austms.org.au/V47EMAC2005/Sampson.

4. Sampson, Joe, Easton, Alan and Singh, Manmohan, A New Moving Boundary Shallow Water Wave Numerical Model, *Australian and New Zealand Industrial and Applied Mathematics Journal*, 48 (CTAC2006), pp. C605–C617, 2007.
[Online] http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/78

Papers accepted for publication

1. Zoppou, C., Roberts, S., Delis, A., and Sampson, J. Shock-Capturing Unsteady and Steady Flow Over Variable Topography without Riemann Solvers, 22 p. *Paper accepted for publication in the Journal of Hydraulic Research.*

2. Sampson, Joe, Easton, Alan and Singh, Manmohan, Moving Boundary Shallow Water Flow in a Region with Quadratic Bathymetry *Paper accepted for publication in Australian and New Zealand Industrial and Applied Mathematics Journal.*

List of written submissions

Sampson, Joe, Easton, Alan and Singh, Manmohan (2004), Port Phillip Bay Channel Deepening Project, *A Submission to the Planning Panels Victoria, Department of Sustainability and Environment, State of Victoria, an Independent Panel on the Proposed Channel Deepening in Port Phillip Bay (Sep-Dec 2004) (this panel reported to the Minister of Planning, Government of Victoria, Australia).*