A Numerical Method for Extended Boussinesq Shallow-Water Wave Equations

by

Mark Andrew Walkley

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

The University of Leeds
School of Computer Studies

September 1999

The candidate confirms that the work submitted is his own and that the appropriate credit has been given where reference has been made to the work of others.

Abstract

The accurate numerical simulation of wave disturbance within harbours requires consideration of both nonlinear and dispersive wave processes in order to capture such physical effects as wave refraction and diffraction, and nonlinear wave interactions such as the generation of harmonic waves. The Boussinesq equations are the simplest class of mathematical model that contain all these effects in a variable depth, shallow water environment. There are a variety of Boussinesq-type mathematical models and it is necessary to compare and contrast them both for their limitations with respect to the physical parameters of the problem and also for their ease of application as part of a suitable numerical model. It is decided here to consider a set of extended Boussinesq equations which provide an accurate model of the wave processes over a greater range of depths than the classical Boussinesq mathematical model.

A method-of-lines numerical algorithm is proposed for these problems, combining a finite element spatial discretisation with existing, adaptive order, adaptive step size time integration software. Two simpler one-dimensional, nonlinear, dispersive wave models; the Korteweg-de Vries equation and Regularised Long Wave equation, are used in the initial development of the numerical methods. It is shown that within the shallow water framework a linear finite element method is sufficiently accurate for these problems.

This numerical method is then applied to the one-dimensional extended Boussinesq equations. It is shown how the previously developed method can be directly used and that it is of similar accuracy to a previously published finite difference method. Initial conditions and boundary conditions are described in detail taking into account physical, mathematical and computational considerations. A new formulation of internal wave generation is developed which allows reflected waves to pass through the wave generation region. The performance of the numerical model is demonstrated by comparison against theoretical results, a previously published finite difference model and experimental results.

The two-dimensional extended Boussinesq equation system is rewritten in a form suitable for the application of a linear triangular finite element spatial discretisation. The formulation of appropriate initial and boundary conditions in combination with the application of the time integration software to this equation system is considered in detail. The performance of the numerical method is tested by comparison with experimental data and the suitability of the model for harbour design is investigated by simulation of a realistic harbour geometry and wave conditions.

Acknowledgements

I would like to thank my supervisor Prof. Martin Berzins for making this an enjoyable period of study on a very interesting project.

I would also like to thank Dr. Jane Smallman, Dr. Jane Lawson, Dr. Alan Cooper, Dr. Nick Dodd, and Nigel Tozer at HR Wallingford who have supported this work throughout, as well as for the extensive use I have made of their experience in this subject area and for my informative periods of study at HR Wallingford during this work.

This research was funded by the EPSRC and HR Wallingford through an EPSRC CASE award.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

M. Walkley and M. Berzins.
A finite element method for the one-dimensional extended Boussinesq equations.
International Journal for Numerical Methods in Fluids, 29:143-157, 1999.

M. Walkley.
User manual for Bouss2D. A finite element method for the extended Boussinesq equations.
HR Wallingford technical report, 1999.

Contents

List of Figures

Abbreviations

BDF	Backward difference formula
ILU	Incomplete lower-upper
KdV	Korteweg-de Vries
LU	Lower-upper
RLW	Regularised long wave

Chapter 1

Introduction

1.1 Boussinesq wave equations

The design of a harbour will generally involve the testing of a number of possible layouts for a variety of possible wave conditions in order to produce a design that minimises wave disturbance within the harbour. Such problems can be analysed experimentally with scale models in wave tanks, or numerically by the application of a suitable mathematical model and numerical method. The advantage of numerical models is the ease with which different layouts can be constructed and tested compared to rebuilding of the physical model. However the assumptions required in their formulation limit their application and important physical processes may not be modelled correctly if an inappropriate mathematical model is chosen. The numerical models are usually based on shallow-water mathematical models, however there are a variety of further simplifying assumptions that can be made depending on the observed physical situation of the model. The wave processes within and surrounding a harbour can generally be termed to be within the nearshore zone, ie. the region of water where the shape of the free surface is affected by the shape of the sea bed, although it is limited by the exclusion of regions within which waves would be expected to break and hence invalidate the shallow-water mathematical model. In the nearshore zone accurate prediction of wave activity needs to account for both nonlinear and dispersive effects. Important wave processes in this region include diffraction, refraction, shoaling and harmonic interaction. While the incompressible Navier-Stokes equations would provide an accurate mathematical model of these processes the numerical solution of a realistic problem would involve a large three-dimensional spatial domain with a free-surface

boundary. In practice the shallow-water nature of this class of problem can be used to simplify the mathematical model and so reduce the computational expense of its solution. This is achieved by averaging the equations through the depth and reducing the spatial domain of the mathematical model to two horizontal dimensions. The evolution of the free surface is then encapsulated in an equation derived from the three-dimensional continuity equation. This procedure introduces, either explicitly or implicitly, a simplified representation of the vertical velocity which restricts the validity of the mathematical model to a shallow water environment. The simplest such equation systems to capture both nonlinear and dispersive wave processes are the Boussinesq equations. The derivation process for these equations is not unique and in this work all equation systems including both nonlinear and dispersive behaviour are termed as being of Boussinesq-type. All model weakly nonlinear, weakly dispersive water waves in a variable depth environment, and in shallow water their linearised dispersion characteristics approximate Stokes first order wave theory [109].

Historically, nonlinear dispersive wave equations were first developed to explain experimental observations of solitary waves which could travel for relatively large distances without changes in their shape or speed [109]. In the 1950's Ursell [101] unified this work and other long wave theories by defining a parameter based on the wave amplitude, wavelength and water depth which parameterised the relative effects of nonlinearity and dispersion. In the 1960's Ursell's result was used by several researchers [65, 70, 78] to develop Boussinesq-type equations in variable depth environments. Peregrine [79] later presented a set of two-dimensional Boussinesq equations which have been the basis of much of the modern-day work. However this system of equations is limited to very shallow water; their linearised dispersion characteristics rapidly diverge from the true behaviour in deeper water rendering the model invalid in these situations. In recent years there have been several proposals of extended Boussinesq systems for which the dispersion relationship is valid up to the deep water limit, increasing the useful range of these models for many applications. Madsen *et al.* [66] added extra dispersion terms to the original system in order to improve the linear dispersion characteristics. This procedure was extended to a variable depth environment [67], although more recent work by Beji and Nadaoka [11] has suggested that there may be inconsistencies in this approach. Beji and Nadaoka [11] have produced an extended Boussinesq system, valid in a variable depth environment, by a simple algebraic manipulation of Peregrine's original system. The consistency of the various equation systems is still being debated [12, 87] although most recently Schäffer and Madsen [88] have suggested that these equation systems are all equivalent and can be derived from a further set of extended Boussinesq equations involving an additional free parameter [86]. In contrast Nwogu [72] derived an extended system of equations from the full fluid equations by choosing the velocity at an arbitrary depth as one of the variables.

Although the extended systems have improved dispersion characteristics they are all still formally of the same accuracy as the original system and hence are restricted to a shallow water environment.

Most of the numerical schemes for the Boussinesq equations are based on finite difference methods. Peregrine [78] presented probably the first finite difference method for a Boussinesq-type equation system. However Abbot *et al.* [1, 2] pioneered the finite difference solution of the original Boussinesq system for practical engineering problems, and analysed the methods carefully to ensure accurate solutions. Madsen *et al.* [66, 67] and Nwogu [72] applied extensions of these techniques in their finite difference methods for the extended Boussinesq equation systems. More recently Wei and Kirby [106] presented an alternative technique for deriving finite difference schemes for Boussinesq-type systems and demonstrated that accurate solutions could be obtained. These finite difference methods have been used to predict wave elevations inside harbours [2, 94] and wave interactions in the nearshore zone [73].

The finite difference methods are simple to formulate, but difficulties in modelling irregular geometries in two space dimensions with structured grids can lead to a loss of accuracy [94]. Finite element methods are straightforward to apply on complex spatial domains but their application to the Boussinesq equations has until very recently been limited to the original Boussinesq system [8, 56, 57]. Langtangen and Pedersen [58, 59] have developed Galerkin finite element methods based on unstructured grids of linear triangles and biquadratic rectangles within the framework of the DIFFPACK software. Ambrosi and Quartepelle [6] applied a modified linear Taylor-Galerkin method to the Boussinesq equations and applied this to problems on unstructured linear triangular grids.

At the start of this project there were, to the author's knowledge, no applications of finite element methods to extended Boussinesq equation systems. However, very recently, Li et al. [62] have described a bilinear quadrilateral finite element method for Beji and Nadaoka's extended system [11]. This work is compared and contrasted with the numerical model developed here throughout the thesis. However there appear to have been no applications of finite element models of these equations for realistic engineering problems at this time.

1.2 Outline of the thesis

The aim of this work is to design a numerical method for the simulation of the wave disturbance inside and around harbour geometries. The Boussinesq wave equations, accounting for both weakly nonlinear and weakly dispersive wave processes, have been identified as a suitable mathematical model although their form is not unique and the precise form of the system will have to be chosen. In Chapter 2 the derivation of Boussinesq equation

systems is considered and the properties of various alternative systems are compared and contrasted. In particular it is shown that extended Boussinesq equation systems significantly increase the range of depths over which the model is applicable and one such model is selected for use in this work.

The numerical method used must be suitable for arbitrary two-dimensional spatial domains and also must be accurate over the long time periods required for the computations. In this work it is proposed to combine an unstructured triangular finite element method with existing adaptive order, adaptive time stepping software within a method of lines framework. Initial investigations of potential numerical methods use two simpler nonlinear dispersive wave equations; the Korteweg-de Vries equations and the Regularised Long Wave equation. These equations retain many of the important mathematical properties of the Boussinesq equations but are written in only one variable which simplifies the development of the numerical algorithms. The derivation of these equations is considered at the end of Chapter 2 and their numerical approximation is described in Chapter 3. Many different one-dimensional spatial discretisations are possible, although only those that can be extended to two-dimensional unstructured triangular grids are considered in detail. In particular it is shown that by rewriting the Korteweg-de Vries equation, which contains a third spatial derivative, as a system of two equations of lower order differential form a novel linear finite element method can be developed and accurate solutions obtained. The accuracy of the numerical scheme is analysed and numerical experiments are performed with exact solitary wave solutions of the equations.

The numerical modelling of the one-dimensional form of the extended Boussinesq equations is considered in Chapter 4. It is shown here that the numerical methods developed in the previous chapter can be directly applied to this equation system. Particular attention is paid to the initial conditions and boundary conditions for these problems and techniques are described that enable efficient use of the time integration software in combination with them. The accuracy of the numerical method is analysed and compared to a previously published finite difference method. The performance of the numerical method is demonstrated by comparison with theoretical results, the previously published finite difference method and experimental data.

The numerical approximation of the two-dimensional extended Boussinesq equations, representing the most general problem to be considered in this work, is described in Chapter 5. It is necessary to rewrite some parts of the equation system so that the previously developed numerical method can be applied. The equation system is rewritten in a novel lower order differential form by extending the techniques developed in the previous chapter. The approximations necessary are discussed in detail here. The spatial discretisation is then achieved with a linear finite element method based on an unstructured triangular grid. Accurate solution of these problems necessitates the adequate spatial resolution of

the surface waves and this will result in a large sparse differential-algebraic equation system which must be solved efficiently for the method to be fully practical. Sparse direct and iterative solution strategies are considered together with reordering techniques that can further increase efficiency. The performance of the numerical method is evaluated by tests involving simple theoretical solutions, experimental data and the simulation of a realistic harbour configuration.

In Chapter 6 the suitability of the numerical method proposed here for the simulation of the wave disturbance in harbours is reviewed. Areas requiring further work are discussed in detail and possible extensions to the numerical model are outlined.

Chapter 2

The Boussinesq Equations

2.1 Introduction

Boussinesq equation systems model water waves in shallow water, including the first order effects of nonlinearity and dispersion. Various ways of deriving Boussinesq-type systems have been discussed in the literature. These can generally be split into so-called engineering derivations [3, 68, 69, 81] and more rigorous derivations based on small parameter expansions [70, 72, 80]. Here the rigorous derivation of nonlinear, dispersive equations is considered, as the techniques and approximations involved provide important insights into the numerical methods proposed in the following chapters. The mathematical framework for the analysis of nonlinear, dispersive shallow-water waves is outlined in Section 2.2 based on the initial assumptions of an inviscid, incompressible Newtonian fluid. The appropriate non-dimensionalisation and scaling of the equations is described in detail. The derivation procedure for Boussinesq-type shallow water equations is not unique and a variety of different equation systems have been proposed, for example [11, 66, 72, 80]. It is the aim of this chapter to evaluate the possible models and to select the most suitable one for numerical computation. The Boussinesq models can be evaluated theoretically through their linear dispersion characteristics, but it is also important to consider the practicalities of applying a numerical method, bearing in mind the eventual aim of solving three-dimensional problems in variable depth environments. The classical form of the Boussinesq equations is derived in Section 2.3. The work in this section follows the derivation procedure presented by Peregrine [80] but includes the effect of a spatially varying depth to derive the variable depth equation system stated by

Peregrine in that work. The associated dispersion properties of this equation system are described and compared to linear dispersive wave theory.

More recently the original Boussinesq equation system has been extended to include the higher order effects of both nonlinearity [107] and dispersion [11, 66, 72, 95]. Here the detailed derivation of two so-called extended Boussinesq systems, which improve the dispersive properties of the equations, is presented in Section 2.4. The first, due to Beji and Nadaoka [11], is produced by a simple manipulation of Peregrine's original Boussinesq system. The second, due to Nwogu [72], is derived by an alternative procedure. Here these variable depth equations are derived using the non-dimensionalisation and scaling of Peregrine [80], for consistency with the previous work in this chapter. The dispersion properties of these extended systems are compared to the classical equations and to the linear theory. Detailed derivations are given for these dispersive properties with comparison of the results made to those stated in the literature, and to those for alternative extended Boussinesq systems.

The Boussinesq equations have a relatively unusual differential form compared to the more standard fluid flow equations, such as the Euler or Navier-Stokes equations, and may pose special difficulties in their numerical approximation. It is shown in Section 2.5 how two simpler nonlinear dispersive equations are derived from the Boussinesq equations which contain these unusual features but in equations of only one variable. These simpler equations will serve as prototypes for the numerical methods presented in the next chapter. Exact solutions of these equations in the form of solitary waves are derived in this section and the appropriate scale of these waves within the Boussinesq framework is considered.

In Section 2.6 the equation systems derived in this chapter are reviewed and the form of the Boussinesq equations chosen for use in this work is discussed.

2.2 The mathematical model

The starting point for modelling water waves is the incompressible Navier-Stokes equations. However the numerical solution of these equations, a three-dimensional problem with a free-surface boundary, is extremely complex and it is usual to make some simplifying assumptions before the numerical solution is attempted. If the water is initially at rest and waves propagate into the domain from the boundary then the flow can be assumed to be inviscid and irrotational [80]. Vorticity can diffuse in from the boundary once the fluid is in motion but for a finite time the effects will not be felt throughout the fluid.

The Boussinesq equations describing shallow water flow are derived from the incompressible, irrotational Euler equations [80]. For simplicity of presentation only two spatial dimensions are considered; (x,z) being horizontal and vertical respectively. The z coordinate varies between the free surface $\eta(x,t)$ and the sea bed $-h(x)$ with the origin taken

at the still-water depth. The frame of reference is illustrated in Figure 2.1. The governing

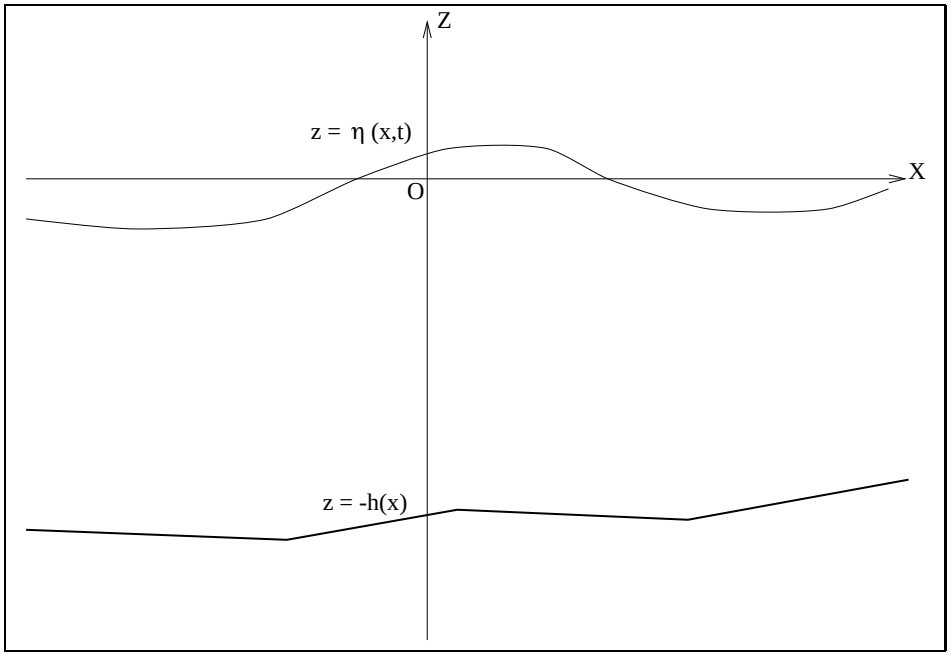


Figure 2.1: The frame of reference.

equation system is comprised of horizontal momentum, vertical momentum, incompressibility and irrotationality equations respectively. These equations are written in terms of the two-dimensional velocity field (U,W), pressure p, constant density ρ and acceleration due to gravity g with respect to the coordinate system (x,z) and time t.

$$\frac{\partial U}{\partial t}+U\frac{\partial U}{\partial x}+W\frac{\partial U}{\partial z}+\frac{1}{\rho}\frac{\partial p}{\partial x} = 0 \tag{2.1}$$

$$\frac{\partial W}{\partial t}+U\frac{\partial W}{\partial x}+W\frac{\partial W}{\partial z}+\frac{1}{\rho}\frac{\partial p}{\partial z}+g = 0 \tag{2.2}$$

$$\frac{\partial U}{\partial x}+\frac{\partial W}{\partial z} = 0 \tag{2.3}$$

$$\frac{\partial U}{\partial z}-\frac{\partial W}{\partial x} = 0 \tag{2.4}$$

At the free surface $z=\eta(x,t)$ the particles are free to move with the fluid velocity, hence there is a kinematic boundary condition,

$$W-U\frac{\partial \eta}{\partial x}-\frac{\partial \eta}{\partial t} = 0 \tag{2.5}$$

At the surface it is also assumed that surface tension is negligible and that there are no applied stresses [80], hence there is a simple constant pressure boundary condition,

$$p = 0 \tag{2.6}$$

where the constant pressure is set to a reference level zero at the free surface.

The bottom boundary $z = -h(x)$ is assumed to be fixed and impermeable and hence the kinematic boundary condition reduces to,

$$W + U\frac{\partial h}{\partial x} = 0 \tag{2.7}$$

2.2.1 Non-dimensionalisation

The parameters ρ, g, and H are chosen to non-dimensionalise the system variables, where H is a typical water depth eg. the average depth. This choice follows the work of Peregrine [80], but is not unique.

$$\tilde{x} = \frac{x}{H}, \quad \tilde{z} = \frac{z}{H}, \quad \tilde{h} = \frac{h}{H}, \quad \tilde{\eta} = \frac{\eta}{H}, \quad \tilde{t} = \sqrt{\frac{g}{H}}t$$
$$\tilde{U} = \frac{U}{\sqrt{gH}}, \quad \tilde{W} = \frac{W}{\sqrt{gH}}, \quad \tilde{p} = \frac{p}{\rho g H}$$

2.2.2 Scaling

To be able to examine the magnitude of individual terms of the equations it is necessary to scale the individual variables so that every aspect of the problem has variation of $\mathcal{O}(1)$. The physical system can be characterised by the typical water depth H, a typical wavelength λ and a typical wave amplitude a. The nonlinearity and dispersion present in the system are parameterised by the ratios ε and σ respectively [80].

$$\varepsilon = \frac{a}{H} \tag{2.8}$$
$$\sigma = \frac{H}{\lambda} \tag{2.9}$$

Ursell [101] discovered a correlation between these two parameters that predicts which wave theory will be applicable. This is known as the Ursell Number, Ur, where.

$$Ur = \frac{\varepsilon}{\sigma^2} = \frac{a\lambda^2}{H^3} \tag{2.10}$$

The Boussinesq wave theory requires $\varepsilon \ll 1$, $\sigma \ll 1$ and Ur to be $\mathcal{O}(1)$ [101]. The equation system can then be consistently derived from the suitably scaled and non-dimensionalised fluid flow equations by integrating through the depth and then expanding in terms of the small parameters σ and ε. Terms up to and including $\mathcal{O}(\varepsilon, \sigma^2)$ are retained.

The coordinates are now scaled appropriately. The horizontal coordinate is scaled such that the scaled coordinate $\hat{x}$ changes $\mathcal{O}(1)$ over one wavelength. Time is also scaled in this manner since the flow will be horizontal to first order and this choice will produce

an $\mathscr{O}(1)$ horizontal velocity. The free surface elevation is scaled to be $\mathscr{O}(1)$ compared to a typical wave amplitude.

$$\hat{x} = \sigma\tilde{x}, \quad \hat{t} = \sigma\tilde{t}, \quad \hat{\eta} = \frac{\tilde{\eta}}{\varepsilon} \tag{2.11}$$

Examining the free surface boundary condition (2.5) and the continuity equation (2.3) leads to the following scaled velocities.

$$\hat{U} = \frac{\tilde{U}}{\varepsilon}, \quad \hat{W} = \frac{\tilde{W}}{\varepsilon\sigma} \tag{2.12}$$

All other variables are assumed to have scaling factors of one.

Inserting these new variables into the equations (2.1)-(2.4) gives,

$$\varepsilon\frac{\partial\hat{U}}{\partial\hat{t}} + \varepsilon^2\hat{U}\frac{\partial\hat{U}}{\partial\hat{x}} + \varepsilon^2\hat{W}\frac{\partial\hat{U}}{\partial\tilde{z}} + \frac{\partial\tilde{p}}{\partial\hat{x}} = 0 \tag{2.13}$$

$$\varepsilon\sigma^2\frac{\partial\hat{W}}{\partial\hat{t}} + \varepsilon^2\sigma^2\hat{U}\frac{\partial\hat{W}}{\partial\hat{x}} + \varepsilon^2\sigma^2\hat{W}\frac{\partial\hat{W}}{\partial\tilde{z}} + \frac{\partial\tilde{p}}{\partial\tilde{z}} + 1 = 0 \tag{2.14}$$

$$\frac{\partial\hat{U}}{\partial\hat{x}} + \frac{\partial\hat{W}}{\partial\tilde{z}} = 0 \tag{2.15}$$

$$\frac{\partial\hat{U}}{\partial\tilde{z}} - \sigma^2\frac{\partial\hat{W}}{\partial\hat{x}} = 0 \tag{2.16}$$

At the free surface $\tilde{z} = \varepsilon\hat{\eta}$ the boundary conditions (2.6) and (2.5) become,

$$\tilde{p} = 0 \tag{2.17}$$

$$\hat{W} - \varepsilon\hat{U}\frac{\partial\hat{\eta}}{\partial\hat{x}} - \frac{\partial\hat{\eta}}{\partial\hat{t}} = 0 \tag{2.18}$$

At the bed $\tilde{z} = -\tilde{h}$ the boundary condition (2.7) becomes,

$$\hat{W} + \hat{U}\frac{\partial\tilde{h}}{\partial\hat{x}} = 0 \tag{2.19}$$

These equations represent a non-dimensionalised, scaled system for small amplitude non-linear water waves and are the basis for the derivation of all the Boussinesq-type shallow water equations considered in this work.

2.3 The classical form of the Boussinesq equations

2.3.1 Derivation

Peregrine [80] provided a precise mathematical framework for the rigorous derivation of water wave equations. In that work a one-dimensional system of Boussinesq equations describing small amplitude nonlinear water waves in a constant depth environment was derived by expanding equations (2.13)-(2.19) in terms of the small parameters ε and σ. Following the procedure outlined by Peregrine the derivation of one-dimensional Boussinesq equations is shown here in detail with the effects of a variable depth included in the derivation. Although the depth is assumed constant initially, terms involving a variation in depth are not neglected in the derivation procedure and the extension to the variable depth equations, stated by Peregrine [80], follows straightforwardly afterwards.

Integrating equation (2.15) with respect to $\tilde{z}$ and applying Liebnitz' Rule [1] gives,

$$\begin{aligned}\int_{-\tilde{h}}^{\tilde{z}} \frac{\partial \hat{W}}{\partial \tilde{z}}\, d\tilde{z} &= -\int_{-\tilde{h}}^{\tilde{z}} \frac{\partial \hat{U}}{\partial \hat{x}}\, d\tilde{z} \\ \hat{W}|_{\tilde{z}} - \hat{W}|_{-\tilde{h}} &= -\frac{\partial}{\partial \hat{x}} \int_{-\tilde{h}}^{\tilde{z}} \hat{U}\, d\tilde{z} - \hat{U}|_{-\tilde{h}} \frac{\partial(-\tilde{h})}{\partial \hat{x}} + \hat{U}|_{\tilde{z}} \frac{\partial \tilde{z}}{\partial \hat{x}}\end{aligned}$$

Using the boundary condition (2.19) at $\tilde{z} = -\tilde{h}$, and the fact that the coordinates $\hat{x}$ and $\tilde{z}$ are independent gives,

$$\hat{W} = -\frac{\partial}{\partial \hat{x}} \int_{-\tilde{h}}^{\tilde{z}} \hat{U}\, d\tilde{z} \tag{2.20}$$

The vertical velocity $\hat{W}|_{\tilde{z}}$ is simply denoted $\hat{W}$. Substituting equation (2.20) in equation (2.16), and integrating with respect to $\tilde{z}$,

$$\begin{aligned}\frac{\partial \hat{U}}{\partial \tilde{z}} &= \sigma^2 \frac{\partial \hat{W}}{\partial \hat{x}} \\ &= -\sigma^2 \frac{\partial^2}{\partial \hat{x}^2} \int_{-\tilde{h}}^{\tilde{z}} \hat{U}\, d\tilde{z} \qquad (2.21) \\ \hat{U} &= \hat{U}_0(\hat{x},\hat{t}) + \mathscr{O}(\sigma^2) \qquad (2.22)\end{aligned}$$

where $\hat{U}_0(\hat{x},\hat{t})$ is an arbitrary function of $\hat{x}$ and $\hat{t}$ introduced by the integration. Equa-

[1] Liebnitz' Rule: Given $f(x,z)$, $a(x)$ and $b(x)$, where f and $\frac{\partial f}{\partial x}$ are continuous in x and z, and a and b are differentiable functions of x,

$$\int_a^b \frac{\partial f}{\partial x}\, dz = \frac{\partial}{\partial x}\left(\int_a^b f\, dz\right) + f(x,a)\frac{\partial a}{\partial x} - f(x,b)\frac{\partial b}{\partial x}$$

tion (2.22) therefore implies that $\hat{U}$ is independent of $\tilde{z}$ to $\mathscr{O}(\sigma^2)$. Substituting equation (2.22) in equation (2.20) and using the independence of $\hat{U}$ from $\tilde{z}$ to evaluate the integral leads to an expression for the vertical velocity $\hat{W}$.

$$\begin{aligned}\hat{W} &= -\frac{\partial}{\partial \hat{x}}\int_{-\tilde{h}}^{\tilde{z}} \hat{U}_0(\hat{x},\hat{t})\, d\tilde{z} + \mathscr{O}(\sigma^2) \\ &= -\frac{\partial}{\partial \hat{x}}\left((\tilde{z}+\tilde{h})\hat{U}_0\right) + \mathscr{O}(\sigma^2) \\ &= -\tilde{z}\frac{\partial \hat{U}_0}{\partial \hat{x}} - \frac{\partial(\tilde{h}\hat{U}_0)}{\partial \hat{x}} + \mathscr{O}(\sigma^2) \end{aligned} \tag{2.23}$$

Substituting equation (2.23) in equation (2.16) and integrating with respect to $\tilde{z}$ obtains an expression for $\hat{U}$.

$$\frac{\partial \hat{U}}{\partial \tilde{z}} = -\sigma^2\left(\tilde{z}\frac{\partial^2 \hat{U}_0}{\partial \hat{x}^2} + \frac{\partial^2(\tilde{h}\hat{U}_0)}{\partial \hat{x}^2}\right) + \mathscr{O}(\sigma^4) \tag{2.24}$$

$$\hat{U} = \hat{U}_0(\hat{x},\hat{t}) - \sigma^2\left(\frac{\tilde{z}^2}{2}\frac{\partial^2 \hat{U}_0}{\partial \hat{x}^2} + \tilde{z}\frac{\partial^2(\tilde{h}\hat{U}_0)}{\partial \hat{x}^2}\right) + \mathscr{O}(\sigma^4) \tag{2.25}$$

Equations (2.14) and (2.23) are now used to obtain an expression for the pressure.

$$\begin{aligned}-\frac{\partial \tilde{p}}{\partial \tilde{z}} &= \varepsilon\sigma^2\frac{\partial \hat{W}}{\partial \hat{t}} + 1 + \mathscr{O}(\varepsilon^2\sigma^2) \\ &= -\varepsilon\sigma^2\left(\tilde{z}\frac{\partial^2 \hat{U}_0}{\partial \hat{x}\partial \hat{t}} + \frac{\partial^2(\tilde{h}\hat{U}_0)}{\partial \hat{x}\partial \hat{t}}\right) + 1 + \mathscr{O}(\varepsilon^2\sigma^2, \varepsilon\sigma^4)\end{aligned}$$

This is integrated with respect to $\tilde{z}$ from an arbitrary depth $\tilde{z}$ to the free surface $\varepsilon\hat{\eta}$.

$$-\int_{\tilde{z}}^{\varepsilon\hat{\eta}} \frac{\partial \tilde{p}}{\partial \tilde{z}}\, d\tilde{z} = -\varepsilon\sigma^2\left[\frac{\tilde{z}^2}{2}\frac{\partial^2 \hat{U}_0}{\partial \hat{x}\partial \hat{t}} + \tilde{z}\frac{\partial^2(\tilde{h}\hat{U}_0)}{\partial \hat{x}\partial \hat{t}}\right]_{\tilde{z}}^{\varepsilon\hat{\eta}} + [\tilde{z}]_{\tilde{z}}^{\varepsilon\hat{\eta}} + \mathscr{O}(\varepsilon^2\sigma^2, \varepsilon\sigma^4) \tag{2.26}$$

The boundary condition (2.17) gives $\tilde{p}|_{\varepsilon\hat{\eta}} = 0$. Denoting $\tilde{p}|_{\tilde{z}}$ simply as $\tilde{p}$ and expanding the right hand side terms, noting that evaluation at $\varepsilon\hat{\eta}$ introduces only $\mathscr{O}(\varepsilon^2\sigma^2)$ terms, gives,

$$\tilde{p} = \varepsilon\sigma^2\left(\frac{\tilde{z}^2}{2}\frac{\partial^2 \hat{U}_0}{\partial \hat{x}\partial \hat{t}} + \tilde{z}\frac{\partial^2(\tilde{h}\hat{U}_0)}{\partial \hat{x}\partial \hat{t}}\right) + \varepsilon\hat{\eta} - \tilde{z} + \mathscr{O}(\varepsilon^2\sigma^2, \varepsilon\sigma^4) \tag{2.27}$$

The expressions for $\hat{W}$ (2.23), $\hat{U}$ (2.25) and $\tilde{p}$ (2.27) are substituted in the horizontal momentum equation (2.13). Rearranging the terms and collecting the high order terms in ε and σ on the right hand side,

$$\frac{\partial \hat{U}_0}{\partial \hat{t}} + \varepsilon\hat{U}_0\frac{\partial \hat{U}_0}{\partial \hat{x}} + \frac{\partial \hat{\eta}}{\partial \hat{x}} = \mathscr{O}(\varepsilon\sigma^2, \sigma^4) \tag{2.28}$$

Now define the depth averaged velocity, $\hat{\bar{u}}(\hat{x},\hat{t})$ [80], in terms of the velocity field $\hat{U}(\hat{x},\tilde{z},\hat{t})$;

$$\hat{\bar{u}} \;=\; \frac{1}{\tilde{h}+\varepsilon\hat{\eta}}\int_{-\tilde{h}}^{\varepsilon\hat{\eta}} \hat{U}\, d\tilde{z} \tag{2.29}$$

The choice of velocity variable is not unique. Mei and Le Méhauté [70] use the bottom velocity $\hat{U}(\hat{x},-\tilde{h},\hat{t})$ to derive an equivalent set of equations. However it will be seen later in this section that the choice of depth averaged velocity allows a particularly simple form for the Boussinesq continuity equation. Substituting from equation (2.25) and rearranging the terms to separate out higher order terms in ε and σ,

$$\begin{aligned}
\hat{\bar{u}} &= \frac{1}{\tilde{h}+\varepsilon\hat{\eta}}\left\{\hat{U}_0(\tilde{h}+\varepsilon\hat{\eta}) - \sigma^2\left(\left(\frac{(\varepsilon\hat{\eta})^3+\tilde{h}^3}{6}\right)\frac{\partial^2\hat{U}_0}{\partial\hat{x}^2}\right.\right.\\
&\qquad\left.\left. +\left(\frac{(\varepsilon\hat{\eta})^2-\tilde{h}^2}{2}\right)\frac{\partial^2(\tilde{h}\hat{U}_0)}{\partial\hat{x}^2}\right)+\mathscr{O}(\sigma^4)\right\}\\
&= \hat{U}_0 - \frac{\sigma^2}{\tilde{h}\left(1+\varepsilon\frac{\hat{\eta}}{\tilde{h}}\right)}\left(\frac{\tilde{h}^3}{6}\frac{\partial^2\hat{U}_0}{\partial\hat{x}^2} - \frac{\tilde{h}^2}{2}\frac{\partial^2(\tilde{h}\hat{U}_0)}{\partial\hat{x}^2}+\mathscr{O}(\varepsilon\sigma^2,\sigma^4)\right)\\
&= \hat{U}_0 - \sigma^2\left(\frac{\tilde{h}^2}{6}\frac{\partial^2\hat{U}_0}{\partial\hat{x}^2} - \frac{\tilde{h}}{2}\frac{\partial^2(\tilde{h}\hat{U}_0)}{\partial\hat{x}^2}\right)+\mathscr{O}(\varepsilon\sigma^2,\sigma^4)
\end{aligned} \tag{2.30}$$

This is rearranged to give an expression for $\hat{U}_0(\hat{x},\hat{t})$ noting that from equation (2.30) $\hat{U}_0 = \hat{\bar{u}}+\mathscr{O}(\sigma^2)$.

$$\hat{U}_0 \;=\; \hat{\bar{u}} + \sigma^2\left(\frac{\tilde{h}^2}{6}\frac{\partial^2\hat{\bar{u}}}{\partial\hat{x}^2} - \frac{\tilde{h}}{2}\frac{\partial^2(\tilde{h}\hat{\bar{u}})}{\partial\hat{x}^2}\right)+\mathscr{O}(\varepsilon\sigma^2,\sigma^4) \tag{2.31}$$

Substituting this in equation (2.28) obtains the momentum equation in the Boussinesq system.

$$\frac{\partial\hat{\bar{u}}}{\partial\hat{t}} + \varepsilon\hat{\bar{u}}\frac{\partial\hat{\bar{u}}}{\partial\hat{x}} + \frac{\partial\hat{\eta}}{\partial\hat{x}} + \sigma^2\left(\frac{\tilde{h}^2}{6}\frac{\partial^2}{\partial\hat{x}^2}\left(\frac{\partial\hat{\bar{u}}}{\partial\hat{t}}\right) - \frac{\tilde{h}}{2}\frac{\partial^2}{\partial\hat{x}^2}\left(\tilde{h}\frac{\partial\hat{\bar{u}}}{\partial\hat{t}}\right)\right) \;=\; \mathscr{O}(\varepsilon\sigma^2,\sigma^4) \tag{2.32}$$

To obtain the second equation of the Boussinesq system equation (2.15) is integrated through the depth.

$$\int_{-\tilde{h}}^{\varepsilon\hat{\eta}}\left(\frac{\partial\hat{U}}{\partial\hat{x}}+\frac{\partial\hat{W}}{\partial\tilde{z}}\right)d\tilde{z} \;=\; 0 \tag{2.33}$$

Using Liebnitz' Rule, the boundary conditions (2.18) and (2.19) and the definition of the depth averaged velocity (2.29),

$$\frac{\partial}{\partial\hat{x}}\int_{-\tilde{h}}^{\varepsilon\hat{\eta}}\hat{U}\,d\tilde{z} - \hat{U}|_{\varepsilon\hat{\eta}}\frac{\partial(\varepsilon\hat{\eta})}{\partial\hat{x}} + \hat{U}|_{-\tilde{h}}\frac{\partial(-\tilde{h})}{\partial\hat{x}} + \hat{W}|_{\varepsilon\hat{\eta}} - \hat{W}|_{-\tilde{h}} \;=\; 0$$

$$\frac{\partial}{\partial \hat{x}}\left((\tilde{h}+\varepsilon\hat{\eta})\hat{\bar{u}}\right)+\frac{\partial \hat{\eta}}{\partial \hat{t}} = 0 \qquad (2.34)$$

Returning the variables to dimensional, unscaled form via the transformations defined in Sections 2.2.1 and 2.2.2, gives the Boussinesq equation system,

$$\frac{\partial \overline{u}}{\partial t}+\overline{u}\frac{\partial \overline{u}}{\partial x}+g\frac{\partial \eta}{\partial x}+\frac{h^2}{6}\frac{\partial^3 \overline{u}}{\partial x^2 \partial t}-\frac{h}{2}\frac{\partial^3 (h\overline{u})}{\partial x^2 \partial t} = 0 \qquad (2.35)$$

$$\frac{\partial \eta}{\partial t}+\frac{\partial}{\partial x}\left((h+\eta)\overline{u}\right) = 0 \qquad (2.36)$$

With a constant depth, H, Peregrine's Boussinesq equation system (2.35)-(2.36) simplifies to,

$$\frac{\partial \overline{u}}{\partial t}+\overline{u}\frac{\partial \overline{u}}{\partial x}+g\frac{\partial \eta}{\partial x}-\frac{H^2}{3}\frac{\partial^3 \overline{u}}{\partial x^2 \partial t} = 0 \qquad (2.37)$$

$$\frac{\partial \eta}{\partial t}+\frac{\partial}{\partial x}\left((H+\eta)\overline{u}\right) = 0 \qquad (2.38)$$

For a variable depth environment an extra scaling parameter characterising the variation in depth must be introduced. Peregrine [80] defines a parameter,

$$\alpha_h = \max\left(\frac{\partial \tilde{h}}{\partial \tilde{x}}\right) \qquad (2.39)$$

over the spatial domain. This scaling primarily affects the bottom boundary condition (2.7). The new scaling leads to the modified form,

$$\sigma\hat{W}+\alpha_h\hat{U}\left(\frac{\partial \tilde{h}}{\partial \tilde{x}}\right) = 0 \qquad (2.40)$$

Comparison with the original scaling in equation (2.19) leads to the conclusion that requiring $\alpha_h = \mathcal{O}(\sigma)$ will lead to the same set of equations as before. This implies that the depth must not vary rapidly over the scale of a typical wavelength, and in particular sudden changes in depth, such as a step, will lead to inconsistencies in this formulation.

A similar procedure in three dimensions leads to the two-dimensional form of Peregrine's Boussinesq equations [80]. Both horizontal velocity components can be non-dimensionalised and scaled in an identical manner to the one-dimensional horizontal velocity.

$$\frac{\partial \overline{\mathbf{u}}}{\partial t}+(\overline{\mathbf{u}}\cdot\nabla)\overline{\mathbf{u}}+g\nabla\eta+\frac{h^2}{6}\nabla\left(\nabla\cdot\frac{\partial \overline{\mathbf{u}}}{\partial t}\right)-\frac{h}{2}\nabla\left(\nabla\cdot\left(h\frac{\partial \overline{\mathbf{u}}}{\partial t}\right)\right) = 0 \qquad (2.41)$$

$$\frac{\partial \eta}{\partial t}+\nabla\cdot\left((h+\eta)\overline{\mathbf{u}}\right) = 0 \qquad (2.42)$$

where $\overline{\mathbf{u}}(x,y,t)$ is the two-dimensional depth averaged velocity field, and ∇ denotes the two-dimensional gradient operator with respect to horizontal coordinates x and y.

These equations have been derived by making assumptions about the physical parameters of the problem, specifically that the wave amplitude is small compared to the depth, and that the depth is small compared to the wavelength. These statements were made precise by the introduction of the parameters ε (2.8) and σ (2.9). It is important to have some idea of the range of application of the Boussinesq model with respect to these parameters so that its usefulness with respect to practical problems can be assessed. In the next section linear dispersive wave theory [109] is introduced and this is then compared to the Boussinesq model in the following section.

2.3.2 Linear dispersive wave theory

It is common to compare the linear dispersion characteristics of the Boussinesq equation systems with those of linear wave theory [11, 67, 72]. Three quantities are generally used; phase velocity, C, which is the velocity of a point on the free surface, group velocity, C_g, which governs the propagation of energy in a wave train, and the shoaling gradient, s, which relates the change in wave amplitude to the change in depth.

The phase velocity of a wave train with frequency ω and wave number k is defined as,

$$C = \frac{\omega}{k} \tag{2.43}$$

For the linear wave theory [109],

$$C^2 = \frac{\omega^2}{k^2} = gh\frac{tanh(kh)}{kh} \tag{2.44}$$

The group velocity of a wave train with frequency ω and wave number k is defined as [109],

$$C_g = \frac{\partial\omega}{\partial k} \tag{2.45}$$

Using the definition of the phase velocity (2.43) to eliminate ω,

$$C_g = \frac{\partial}{\partial k}(kC) = C + k\frac{\partial C}{\partial k} \tag{2.46}$$

Using the phase velocity from linear wave theory (2.44) one arrives at the form stated by Madsen et al. [67],

$$C_g = \frac{C}{2}\left(1 + \frac{2kh}{sinh(2kh)}\right) \tag{2.47}$$

To simplify the following algebra in this section the group velocity is written in a more convenient form. Let,

$$C_g = C\,F(\phi) \tag{2.48}$$

where,

$$\phi = kh \tag{2.49}$$

$$F(\phi) = \frac{1}{2}\left(1+\frac{2\phi}{sinh(2\phi)}\right) \tag{2.50}$$

To derive the shoaling gradient consider a wave train described by a constant frequency, ω, and spatially varying wave number, $k(x)$, and wave amplitude, $a(x)$, over a varying depth, $h(x)$. The statement of the constancy of energy flux is [11],

$$\begin{aligned} \frac{\partial}{\partial x}\left(a^2 C_g\right) &= 0 \\ 2a\frac{\partial a}{\partial x}C_g + a^2\frac{\partial C_g}{\partial x} &= 0 \\ \frac{1}{a}\frac{\partial a}{\partial x} + \frac{1}{2C_g}\frac{\partial C_g}{\partial x} &= 0 \end{aligned} \tag{2.51}$$

From equation (2.48),

$$\frac{1}{C_g}\frac{\partial C_g}{\partial x} = \frac{1}{C}\frac{\partial C}{\partial x} + \frac{1}{F}\frac{\partial F}{\partial x} \tag{2.52}$$

This equation can be simplified by differentiating the phase velocity (2.43) with respect to x,

$$\begin{aligned} \frac{\partial C}{\partial x} &= -\frac{\omega}{k^2}\frac{\partial k}{\partial x} \\ \frac{1}{C}\frac{\partial C}{\partial x} &= -\frac{1}{k}\frac{\partial k}{\partial x} \end{aligned} \tag{2.53}$$

Using this in the group velocity expression (2.52) and substituting this into the shoaling gradient relation(2.51) results in,

$$\frac{1}{a}\frac{\partial a}{\partial x} + \frac{1}{2}\left(\frac{1}{F}\frac{\partial F}{\partial x} - \frac{1}{k}\frac{\partial k}{\partial x}\right) = 0 \tag{2.54}$$

Terms involving F are eliminated from equation (2.54) by differentiating equation (2.50)

with respect to x,

$$\begin{aligned}
\frac{\partial F}{\partial x} &= \frac{\partial \phi}{\partial x}\frac{\partial F}{\partial \phi} \\
&= \frac{\partial \phi}{\partial x}\left(\frac{sinh(2\phi)-2\phi\ cosh(2\phi)}{2\ sinh^2(2\phi)}\right) \\
\frac{1}{F}\frac{\partial F}{\partial x} &= \frac{1}{\phi}\frac{\partial \phi}{\partial x}\left(\frac{2\phi(sinh(2\phi)-2\phi\ cosh(2\phi))}{sinh(2\phi)(sinh(2\phi)+2\phi)}\right)
\end{aligned} \tag{2.55}$$

From the definition of ϕ (2.49),

$$\begin{aligned}
\frac{\partial \phi}{\partial x} &= k\frac{\partial h}{\partial x}+h\frac{\partial k}{\partial x} \\
\frac{1}{\phi}\frac{\partial \phi}{\partial x} &= \frac{1}{h}\frac{\partial h}{\partial x}+\frac{1}{k}\frac{\partial k}{\partial x}
\end{aligned} \tag{2.56}$$

A relationship between the wave number gradient and the depth gradient is derived from the phase velocity. Rearranging equation (2.44),

$$\frac{\omega^2 h}{g} = \phi\ tanh(\phi) \tag{2.57}$$

Differentiating (2.57) with respect to x,

$$\frac{\omega^2}{g}\frac{\partial h}{\partial x} = \frac{\partial \phi}{\partial x}\frac{\partial}{\partial \phi}(\phi\ tanh(\phi)) \tag{2.58}$$

Using (2.57) to eliminate the ω terms,

$$\frac{1}{h}\frac{\partial h}{\partial x}tanh(\phi) = \frac{1}{\phi}\frac{\partial \phi}{\partial x}\left(tanh(\phi)+\phi(1-tanh^2(\phi))\right) \tag{2.59}$$

Equation (2.56) is used in equation (2.59) to derive a relationship between the wave number gradient and the depth gradient. After rearranging the terms,

$$\frac{1}{k}\frac{\partial k}{\partial x} = \left(\frac{-2\phi}{2\phi+sinh(2\phi)}\right)\frac{1}{h}\frac{\partial h}{\partial x} \tag{2.60}$$

Substituting equation (2.55) in the shoaling gradient relation (2.54), and using equation (2.60) to substitute for terms involving k, the shoaling gradient can be written,

$$\frac{1}{a}\frac{\partial a}{\partial x}+s\,\frac{1}{h}\frac{\partial h}{\partial x} = 0 \tag{2.61}$$

where,

$$s = \frac{2\phi\,(sinh(2\phi)+\phi(1-cosh(2\phi)))}{(2\phi+sinh(2\phi))^2} \tag{2.62}$$

Some manipulation of this expression shows that it is identical to the form stated by Madsen et al. [67].

2.3.3 Dispersion characteristics of Peregrine's Boussinesq equation system

The Boussinesq equation system (2.35)-(2.36) is only weakly nonlinear, the nonlinearity present being proportional to the small parameter ε. Linearising these equations by neglecting the nonlinear terms is therefore only a small perturbation to this system and analysis of the linearised equations should provide an adequate model for the full equation system. Considering the constant depth equations (2.37)-(2.38) and neglecting terms proportional to ε, as shown in the scaled nondimensionalised equations (2.32) and (2.34) gives,

$$\frac{\partial\overline{u}}{\partial t}+g\frac{\partial\eta}{\partial x}-\frac{H^2}{3}\frac{\partial^3\overline{u}}{\partial x^2\partial t} = 0 \tag{2.63}$$

$$\frac{\partial\eta}{\partial t}+H\frac{\partial\overline{u}}{\partial x} = 0 \tag{2.64}$$

The dispersion relation for this equation system is derived by assuming a steady periodic wave of amplitude a, period $\tau=2\pi/\omega$ and wavelength $\lambda=2\pi/k$,

$$\eta(x,t) = a\,sin(kx-\omega t) \tag{2.65}$$

$$\overline{u}(x,t) = b\,sin(kx-\omega t) \tag{2.66}$$

Substituting this in the system (2.63)-(2.64),

$$-\omega b+gka-\frac{H^2}{3}\omega k^2 b = 0 \tag{2.67}$$

$$-\omega a+Hkb = 0 \tag{2.68}$$

This leads to the velocity magnitude,

$$b = \frac{\omega a}{kH} \tag{2.69}$$

and the dispersion relation,

$$\frac{\omega^2}{k^2} = \frac{gH}{1+\frac{(kH)^2}{3}} \tag{2.70}$$

Hence, using the notation (2.49) of the previous section and the definition (2.43), the phase velocity for the system is,

$$C = \sqrt{\left(\frac{gH}{1+\frac{\phi^2}{3}}\right)} \tag{2.71}$$

The group velocity is derived from the phase velocity (2.71) and the definition (2.46).

$$C_g = C\left(\frac{3}{3+\phi^2}\right) \tag{2.72}$$

Following a similar argument to that used for the linear wave theory in Section 2.3.2 the shoaling gradient coefficient can be derived for the Boussinesq equations. Key points are the formation of the spatial derivative of the group velocity component $F(\phi)$ (2.55),

$$\frac{1}{F}\frac{\partial F}{\partial x} = -\left(\frac{2\phi^2}{3+\phi^2}\right)\frac{1}{\phi}\frac{\partial \phi}{\partial x} \tag{2.73}$$

and the relationship between the wave number gradient and the depth gradient (2.60),

$$\frac{1}{k}\frac{\partial k}{\partial x} = \frac{1}{2}\left(\frac{\phi^2}{3}-1\right)\frac{1}{h}\frac{\partial h}{\partial x} \tag{2.74}$$

This finally leads to,

$$s = \frac{1}{4}\left(1-\phi^2\right) \tag{2.75}$$

2.3.4 Comparison with linear theory

The comparisons between the Boussinesq and linear theory dispersion characteristics are shown graphically by Figures 2.2(a)-(c). The graphs of phase velocity and group velocity show % error relative to the linear theory, where the error is determined by,

$$100\left(\frac{C_{Boussinesq}-C_{linear}}{C_{linear}}\right) \tag{2.76}$$

Figure 2.2(a) shows that the phase velocity is over 5% in error for depths in excess of 3/10 of the wavelength. Figure 2.2(b) shows that the group velocity is almost 10% in error at a

depth corresponding to 2/10 of the wavelength. The plot of the shoaling gradient, shown in Figure 2.2(c) also rapidly diverges from the linear theory. This restricts the application of the model to very shallow depths compared to the wavelength, or put another way to very long wavelengths compared to the depth. This is a fairly severe restriction on the usefulness of a computational model based on Peregrine's Boussinesq equations.

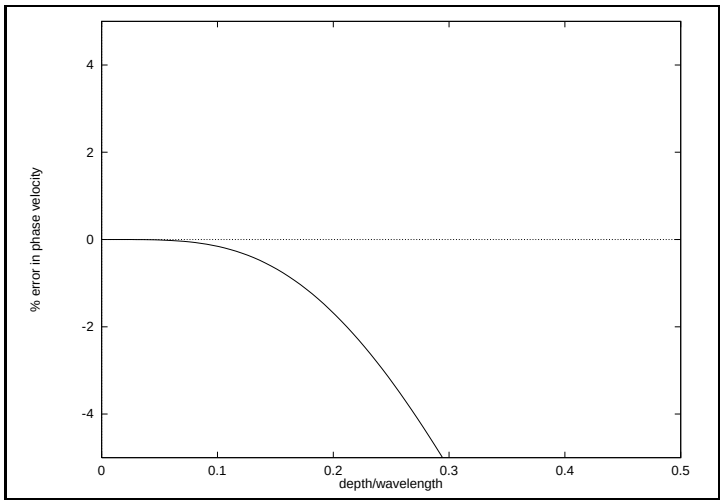
2.4 Extended Boussinesq equation systems

2.4.1 Introduction

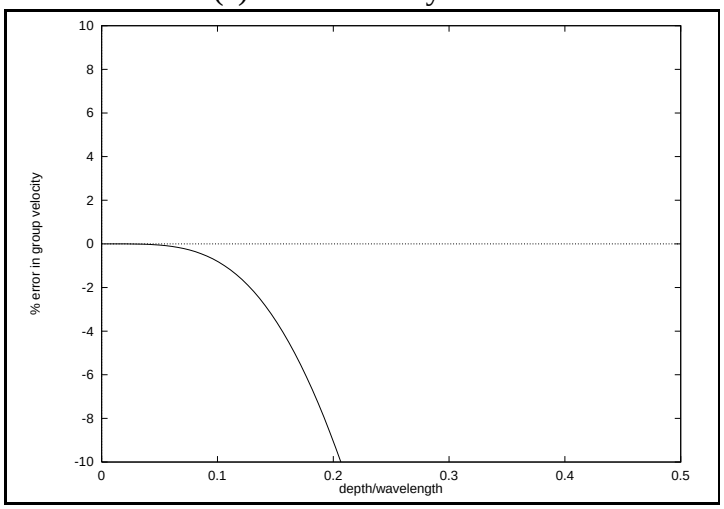
It has been noted in the last section that the dispersion relation of Peregrine's Boussinesq equation system is only an accurate approximation to Stokes first order wave theory for very small values of σ. Much of the recent work in this area has concentrated on extending the validity of the dispersion relation to larger values of σ [11, 66, 72, 111]. Witting [111] presented a one-dimensional constant depth system with improved linear dispersion characteristics. However it appears difficult to generalise this approach to either higher dimensions or a variable depth environment. Madsen et al. [66] improved the dispersion characteristics of Peregrine's system by adding extra dispersive terms to the momentum equation (2.37). The amount of dispersion added was chosen by matching the dispersion characteristics to linear wave theory. This procedure was extended to a variable depth environment [67], although more recent work has suggested that there may be inconsistencies in this approach [11, 12, 87]. Beji and Nadaoka [11] designed a variable depth extended Boussinesq system by a simple algebraic manipulation of Peregrine's system. This introduced a free parameter which could be used to improve the dispersion characteristics. Nwogu [72] consistently derived an extended Boussinesq equation system from the Navier-Stokes equations by using the velocity at an arbitrary depth as one of the dependent variables. The depth of the velocity variable is used as a free parameter to optimise the dispersion characteristics of the equations. A detailed derivation of these equations is given in the following section.

The extended Boussinesq equation systems described so far have all attempted to improve the dispersion characteristics of the basic equation system. These improvements have been taken further [71, 86, 95] to achieve an almost perfect match with linear wave theory but at the expense of increasing complexity in the differential form of the equations. However the equations remain only weakly nonlinear and as such are restricted to small amplitudes relative to the water depth. Fully nonlinear Boussinesq models have been developed [9, 71, 107] to simulate larger waves but again at the expense of increasing complexity in the equations.

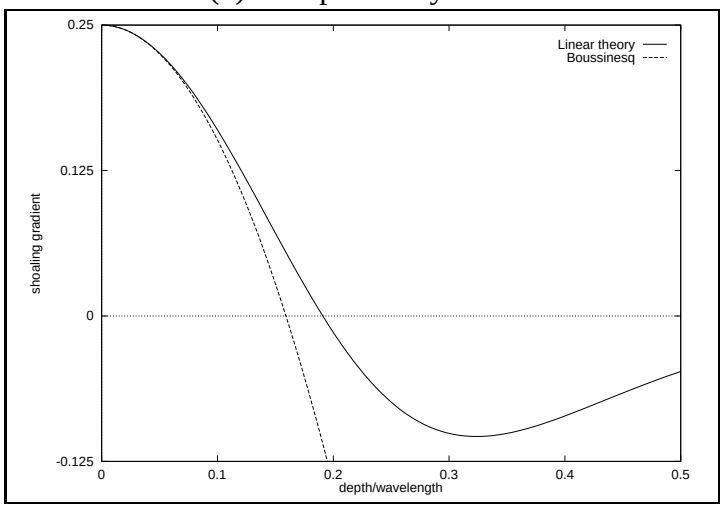
It was decided at this stage that the improved dispersion relations of the extended



(a) Phase velocity error



(b) Group velocity error



(c) Shoaling gradient

Figure 2.2: Comparison of the dispersion characteristics of Peregrine's Boussinesq equations and linear theory

Boussinesq equations would make them far more useful as a practical numerical model, since they are applicable over a far greater range of depths. The extended systems of Beji and Nadaoka [11] and Nwogu [72] are both derived in a mathematically consistent manner and so they are both considered as candidates for the numerical model. The derivation and analysis of these two equation systems is considered in the following sections.

2.4.2 Derivation of Beji and Nadaoka's extended Boussinesq equations

The starting point of the derivation is Peregrine's Boussinesq equation system, written in scaled nondimensional form.

$$\begin{aligned} &\frac{\partial \hat{\bar{u}}}{\partial \hat{t}} + \varepsilon \hat{\bar{u}} \frac{\partial \hat{\bar{u}}}{\partial \hat{x}} + \frac{\partial \hat{\eta}}{\partial \hat{x}} \\ &+\sigma^2 \left(\frac{\tilde{h}^2}{6} \frac{\partial^2}{\partial \hat{x}^2} \left(\frac{\partial \hat{\bar{u}}}{\partial \hat{t}} \right) - \frac{\tilde{h}}{2} \frac{\partial^2}{\partial \hat{x}^2} \left(\frac{\partial (\tilde{h} \hat{\bar{u}})}{\partial \hat{t}} \right) \right) &= \mathcal{O}(\varepsilon\sigma^2, \sigma^4) \qquad (2.77) \\ &\frac{\partial \hat{\eta}}{\partial \hat{t}} + \frac{\partial}{\partial \hat{x}} \left((\tilde{h} + \varepsilon \hat{\eta}) \hat{\bar{u}} \right) &= \mathcal{O}(\varepsilon\sigma^2, \sigma^4) \qquad (2.78) \end{aligned}$$

The dispersive terms from the momentum equation (2.77) are rewritten by adding and subtracting a free parameter α_B which multiplies the dispersive terms.

$$\begin{aligned} &\frac{\partial \hat{\bar{u}}}{\partial \hat{t}} + \varepsilon \hat{\bar{u}} \frac{\partial \hat{\bar{u}}}{\partial \hat{x}} + \frac{\partial \hat{\eta}}{\partial \hat{x}} \\ &\quad +(1+\alpha_B)\sigma^2 \left(\frac{\tilde{h}^2}{6} \frac{\partial^2}{\partial \hat{x}^2} \left(\frac{\partial \hat{\bar{u}}}{\partial \hat{t}} \right) - \frac{\tilde{h}}{2} \frac{\partial^2}{\partial \hat{x}^2} \left(\frac{\partial (\tilde{h} \hat{\bar{u}})}{\partial \hat{t}} \right) \right) \\ &\quad -\alpha_B \sigma^2 \left(\frac{\tilde{h}^2}{6} \frac{\partial^2}{\partial \hat{x}^2} \left(\frac{\partial \hat{\bar{u}}}{\partial \hat{t}} \right) - \frac{\tilde{h}}{2} \frac{\partial^2}{\partial \hat{x}^2} \left(\frac{\partial (\tilde{h} \hat{\bar{u}})}{\partial \hat{t}} \right) \right) = \mathcal{O}(\varepsilon\sigma^2, \sigma^4) \qquad (2.79) \end{aligned}$$

The second group of dispersive terms are rewritten using the first order wave equations,

$$\begin{aligned} \frac{\partial \hat{\bar{u}}}{\partial \hat{t}} + \frac{\partial \hat{\eta}}{\partial \hat{x}} &= \mathcal{O}(\varepsilon, \sigma^2) \qquad (2.80) \\ \frac{\partial \hat{\eta}}{\partial \hat{t}} + \frac{\partial (\tilde{h} \hat{\bar{u}})}{\partial \hat{x}} &= \mathcal{O}(\varepsilon, \sigma^2) \qquad (2.81) \end{aligned}$$

Using these to substitute for first order derivatives within the dispersive terms introduces $\mathcal{O}(\varepsilon\sigma^2, \sigma^4)$ corrections which do not perturb the equations within the order of approximation of the Boussinesq theory [11]. This leads to the extended Boussinesq momentum equation,

$$\frac{\partial \hat{\bar{u}}}{\partial \hat{t}} + \varepsilon \hat{\bar{u}} \frac{\partial \hat{\bar{u}}}{\partial \hat{x}} + \frac{\partial \hat{\eta}}{\partial \hat{x}}$$

$$+(1+\alpha_B)\sigma^2\left(\frac{\tilde{h}^2}{6}\frac{\partial^2}{\partial\hat{x}^2}\left(\frac{\partial\hat{\overline{u}}}{\partial\hat{t}}\right)-\frac{\tilde{h}}{2}\frac{\partial^2}{\partial\hat{x}^2}\left(\frac{\partial(\tilde{h}\hat{\overline{u}})}{\partial\hat{t}}\right)\right)$$
$$+\alpha_B\sigma^2\left(\frac{\tilde{h}^2}{6}\frac{\partial^3\hat{\eta}}{\partial\hat{x}^3}-\frac{\tilde{h}}{2}\frac{\partial^2}{\partial\hat{x}^2}\left(\tilde{h}\frac{\partial\hat{\eta}}{\partial\hat{x}}\right)\right) = \mathscr{O}(\varepsilon\sigma^2,\sigma^4) \quad (2.82)$$

At constant depth H, in unscaled dimensional variables, the equations simplify to,

$$\frac{\partial\overline{u}}{\partial t}+\overline{u}\frac{\partial\overline{u}}{\partial x}+g\frac{\partial\eta}{\partial x}-(1+\alpha_B)\frac{H^2}{3}\frac{\partial^3\overline{u}}{\partial x^2\partial t}-\alpha_B\frac{gH^2}{3}\frac{\partial^3\eta}{\partial x^3} = 0 \quad (2.83)$$
$$\frac{\partial\eta}{\partial t}+\frac{\partial}{\partial x}((H+\eta)\overline{u}) = 0 \quad (2.84)$$

The linear dispersion characteristics of this equation system will be considered in Section 2.4.4

2.4.3 Derivation of Nwogu's extended Boussinesq equations

The following section describes the derivation of Nwogu's one-dimensional extended Boussinesq equations. This derivation expands on that outlined by Nwogu [72] for the two-dimensional equation system, but uses the nondimensionalised, scaled equation system of Peregrine (2.13)-(2.19) as the starting point, for consistency with the previous work in this section, rather than that presented by Nwogu.

Integrating the continuity equation (2.15) through the depth,

$$\int_{-\tilde{h}}^{\tilde{z}}\frac{\partial\hat{W}}{\partial\tilde{z}}\,d\tilde{z} = -\int_{-\tilde{h}}^{\tilde{z}}\frac{\partial\hat{U}}{\partial\hat{x}}\,d\tilde{z}$$
$$\hat{W}|_{\tilde{z}}-\hat{W}|_{-\tilde{h}} = -\frac{\partial}{\partial\hat{x}}\left(\int_{-\tilde{h}}^{\tilde{z}}\hat{U}\,d\tilde{z}\right)+\hat{U}|_{-\tilde{h}}\frac{\partial(-\tilde{h})}{\partial\hat{x}}-\hat{U}|_{\tilde{z}}\frac{\partial\tilde{z}}{\partial\hat{x}} \quad (2.85)$$

Denoting $\hat{W}|_{\tilde{z}}$ simply as $\hat{W}$, and using Liebnitz' Rule and the boundary condition (2.19) at $\tilde{z}=-\tilde{h}$,

$$\hat{W} = -\frac{\partial}{\partial\hat{x}}\left(\int_{-\tilde{h}}^{\tilde{z}}\hat{U}\,d\tilde{z}\right) \quad (2.86)$$

Substituting this in the irrotationality equation (2.16),

$$\frac{\partial\hat{U}}{\partial\tilde{z}} = \sigma^2\frac{\partial\hat{W}}{\partial\hat{x}}$$
$$= -\sigma^2\frac{\partial^2}{\partial\hat{x}^2}\left(\int_{-\tilde{h}}^{\tilde{z}}\hat{U}\,d\tilde{z}\right) \quad (2.87)$$

Consider a Taylor series expansion of $\hat{U}(\hat{x},\tilde{z},\hat{t})$ about $\tilde{z}=\tilde{Z}$. The horizontal velocity $\hat{u}(\hat{x},\hat{t})$ denotes the two-dimensional velocity at depth $\tilde{Z}$; $\hat{U}(\hat{x},\tilde{Z},\hat{t})$. This Taylor expansion

is integrated through the depth from $-\tilde{h}$ to $\tilde{z}$.

$$\hat{U} = \hat{u} + (\tilde{z}-\tilde{Z})\left.\frac{\partial \hat{U}}{\partial \tilde{z}}\right|_{\tilde{z}=\tilde{Z}} + \frac{(\tilde{z}-\tilde{Z})^2}{2}\left.\frac{\partial^2 \hat{U}}{\partial \tilde{z}^2}\right|_{\tilde{z}=\tilde{Z}} + \ldots \tag{2.88}$$

$$\begin{aligned}\int_{-\tilde{h}}^{\tilde{z}} \hat{U}\, d\tilde{z} &= (\tilde{z}+\tilde{h})\hat{u} + \left(\frac{(\tilde{z}-\tilde{Z})^2}{2} - \frac{(\tilde{h}+\tilde{Z})^2}{2}\right)\left.\frac{\partial \hat{U}}{\partial \tilde{z}}\right|_{\tilde{z}=\tilde{Z}} \\ &\quad + \left(\frac{(\tilde{z}-\tilde{Z})^3}{6} + \frac{(\tilde{h}+\tilde{Z})^3}{6}\right)\left.\frac{\partial^2 \hat{U}}{\partial \tilde{z}^2}\right|_{\tilde{z}=\tilde{Z}} + \ldots \end{aligned} \tag{2.89}$$

Substituting this in equation (2.87),

$$\frac{\partial \hat{U}}{\partial \tilde{z}} = -\sigma^2 \frac{\partial^2}{\partial \hat{x}^2}\left((\tilde{z}+\tilde{h})\hat{u} + \left(\frac{(\tilde{z}-\tilde{Z})^2}{2} - \frac{(\tilde{h}+\tilde{Z})^2}{2}\right)\left.\frac{\partial \hat{U}}{\partial \tilde{z}}\right|_{\tilde{z}=\tilde{Z}} + \ldots\right) \tag{2.90}$$

Differentiating equation (2.90) with respect to $\tilde{z}$,

$$\begin{aligned}\frac{\partial^2 \hat{U}}{\partial \tilde{z}^2} &= -\sigma^2 \frac{\partial^2}{\partial \hat{x}^2}\left(\hat{u} + (\tilde{z}-\tilde{Z})\left.\frac{\partial \hat{U}}{\partial \tilde{z}}\right|_{\tilde{z}=\tilde{Z}} + \frac{(\tilde{z}-\tilde{Z})^2}{2}\left.\frac{\partial^2 \hat{U}}{\partial \tilde{z}^2}\right|_{\tilde{z}=\tilde{Z}} + \ldots\right) \\ &= -\sigma^2 \frac{\partial^2 \hat{u}}{\partial \hat{x}^2} + \mathscr{O}(\sigma^4) \end{aligned} \tag{2.91}$$

Differentiating equation (2.91) with respect to z and noting from equations (2.90) and (2.91) that the first and second derivatives are $\mathscr{O}(\sigma^2)$,

$$\begin{aligned}\frac{\partial^3 \hat{U}}{\partial \tilde{z}^3} &= -\sigma^2 \frac{\partial^2}{\partial \hat{x}^2}\left(\left.\frac{\partial \hat{U}}{\partial \tilde{z}}\right|_{\tilde{z}=\tilde{Z}} + (\tilde{z}-\tilde{Z})\left.\frac{\partial^2 \hat{U}}{\partial \tilde{z}^2}\right|_{\tilde{z}=\tilde{Z}} + \ldots\right) \\ &= \mathscr{O}(\sigma^4) \end{aligned} \tag{2.92}$$

Repeated differentiation of this expression will produce expressions for the higher derivatives of $\hat{U}$ with respect to $\tilde{z}$ and show them to be of $\mathscr{O}(\sigma^4)$ or greater. Substituting equations (2.91) and (2.92) back in equation (2.90),

$$\frac{\partial \hat{U}}{\partial \tilde{z}} = -\sigma^2 \frac{\partial^2}{\partial \hat{x}^2}\left((\tilde{h}+\tilde{Z})\hat{u} - \frac{(\tilde{h}+\tilde{Z})^2}{2}\left.\frac{\partial \hat{U}}{\partial \tilde{z}}\right|_{\tilde{z}=\tilde{Z}} + \mathscr{O}(\sigma^2)\right) \tag{2.93}$$

$$= -\sigma^2 \frac{\partial^2}{\partial \hat{x}^2}\left((\tilde{h}+\tilde{Z})\hat{u}\right) + \mathscr{O}(\sigma^4) \tag{2.94}$$

Substituting equations (2.91), (2.92) and (2.94) in the Taylor series expansion (2.88) produces an expression for the horizontal velocity component $\hat{U}$.

$$\hat{U} = \hat{u} - \sigma^2\left((\tilde{z}-\tilde{Z})\frac{\partial^2}{\partial \hat{x}^2}\left((\tilde{h}+\tilde{Z})\hat{u}\right) + \frac{(\tilde{z}-\tilde{Z})^2}{2}\frac{\partial^2 \hat{u}}{\partial \hat{x}^2}\right) + \mathscr{O}(\sigma^4) \tag{2.95}$$

Substituting the horizontal velocity (2.95) in equation (2.86) leads to an expression for

the vertical velocity component $\hat{W}$.

$$\hat{W} = -\frac{\partial}{\partial \hat{x}}\left((\tilde{h}+\tilde{z})\hat{u}+\sigma^2\left(\frac{(\tilde{h}+\tilde{Z})^2}{2}\frac{\partial^2}{\partial \hat{x}^2}\left((\tilde{h}+\tilde{Z})\hat{u}\right) - \frac{(\tilde{h}+\tilde{Z})^3}{6}\frac{\partial^2 \hat{u}}{\partial \hat{x}^2}\right)\right)+\mathscr{O}(\sigma^4) \tag{2.96}$$

Using the velocities $\hat{U}$ (2.95) and $\hat{W}$ (2.96) in the vertical momentum equation (2.14),

$$\varepsilon\sigma^2\left(-\frac{\partial^2}{\partial \hat{x}\partial \hat{t}}((\tilde{z}+\tilde{h})\hat{u})+\mathscr{O}(\sigma^2)\right)+\frac{\partial \tilde{p}}{\partial \tilde{z}}+1+\mathscr{O}(\varepsilon^2\sigma^2) = 0 \tag{2.97}$$

This can be rearranged to give an expression for the pressure $\tilde{p}$.

$$-\frac{\partial \tilde{p}}{\partial \tilde{z}} = 1-\varepsilon\sigma^2\frac{\partial^2}{\partial \hat{x}\partial \hat{t}}((\tilde{z}+\tilde{h})\hat{u})+\mathscr{O}(\varepsilon\sigma^4,\varepsilon^2\sigma^2) \tag{2.98}$$

Integrating through the depth from $\tilde{z}$ to the free surface $\varepsilon\hat{\eta}$,

$$-\tilde{p}|_{\varepsilon\hat{\eta}}+\tilde{p}|_{\tilde{z}} = \varepsilon\hat{\eta}-\tilde{z}+\varepsilon\sigma^2\frac{\partial^2}{\partial \hat{x}\partial \hat{t}}\left(\left(\frac{\tilde{z}^2}{2}+\tilde{h}\tilde{z}\right)U\right)+\mathscr{O}(\varepsilon\sigma^4,\varepsilon^2\sigma^2) \tag{2.99}$$

Using the free surface boundary condition (2.17) for the pressure, and denoting $\tilde{p}|_{\tilde{z}}$ simply as $\tilde{p}$ gives,

$$\tilde{p} = \varepsilon\hat{\eta}-\tilde{z}+\varepsilon\sigma^2\frac{\partial^2}{\partial \hat{x}\partial \hat{t}}\left(\left(\frac{\tilde{z}^2}{2}+\tilde{h}\tilde{z}\right)U\right)+\mathscr{O}(\varepsilon\sigma^4,\varepsilon^2\sigma^2) \tag{2.100}$$

Substituting the equations for the horizontal velocity component (2.95), the vertical velocity component (2.96) and the pressure(2.100) in the horizontal momentum equation (2.13) gives the Boussinesq momentum equation,

$$\begin{aligned}
&\frac{\partial \hat{u}}{\partial \hat{t}}+\varepsilon\hat{u}\frac{\partial \hat{u}}{\partial \hat{x}}+\frac{\partial \hat{\eta}}{\partial \hat{x}} \\
&\quad -\sigma^2\left((\tilde{z}-\tilde{Z})\frac{\partial^2}{\partial \hat{x}^2}\left((\tilde{h}+\tilde{Z})\frac{\partial \hat{u}}{\partial \hat{t}}\right)+\frac{(\tilde{z}-\tilde{Z})^2}{2}\frac{\partial^2}{\partial \hat{x}^2}\left(\frac{\partial \hat{u}}{\partial \hat{t}}\right)\right) \\
&\quad +\varepsilon\hat{u}\frac{\partial \hat{u}}{\partial \hat{x}}+\frac{\partial \hat{\eta}}{\partial \hat{x}}+\sigma^2\frac{\partial^2}{\partial \hat{x}^2}\left(\left(\frac{\tilde{z}^2}{2}+\tilde{h}\tilde{z}\right)\frac{\partial \hat{u}}{\partial \hat{t}}\right)+\mathscr{O}(\varepsilon\sigma^2,\sigma^4) = 0 \\
&\frac{\partial \hat{u}}{\partial \hat{t}}+\varepsilon\hat{u}\frac{\partial \hat{u}}{\partial \hat{x}}+\frac{\partial \hat{\eta}}{\partial \hat{x}}+\sigma^2\left(\tilde{Z}\frac{\partial^2}{\partial \hat{x}^2}\left(\tilde{h}\frac{\partial \hat{u}}{\partial \hat{t}}\right)+\frac{\tilde{Z}^2}{2}\frac{\partial^2}{\partial \hat{x}^2}\left(\frac{\partial \hat{u}}{\partial \hat{t}}\right)\right) = \mathscr{O}(\varepsilon\sigma^2,\sigma^4)
\end{aligned} \tag{2.101}$$

The second equation of the Boussinesq system is developed by first integrating the

continuity equation (2.15) through the depth.

$$\frac{\partial}{\partial \hat{x}}\int_{-\tilde{h}}^{\varepsilon\hat{\eta}} \hat{U}\, d\tilde{z} - \varepsilon\hat{U}|_{\varepsilon\hat{\eta}}\frac{\partial \hat{\eta}}{\partial \hat{x}} + \hat{U}|_{-\tilde{h}}\frac{\partial(-\tilde{h})}{\partial \hat{x}} + \hat{W}|_{\varepsilon\hat{\eta}} - \hat{W}|_{-\tilde{h}} = 0 \qquad (2.102)$$

Using Liebnitz' Rule and the kinematic boundary conditions at the free surface $\tilde{z} = \varepsilon\hat{\eta}$ (2.18) and at the bed $\tilde{z} = -\tilde{h}$ (2.19) gives,

$$\frac{\partial \hat{\eta}}{\partial \hat{t}} + \frac{\partial}{\partial \hat{x}}\int_{-\tilde{h}}^{\varepsilon\hat{\eta}} \hat{U}\, d\tilde{z} = 0 \qquad (2.103)$$

From the expression for the horizontal velocity $\hat{U}$ (2.95),

$$\begin{aligned}
\int_{-\tilde{h}}^{\varepsilon\hat{\eta}} \hat{U}\, d\tilde{z} &= (\tilde{h}+\varepsilon\hat{\eta})\hat{u} - \sigma^2\left(\left[\frac{(\tilde{z}-\tilde{Z})^2}{2}\right]_{-\tilde{h}}^{\varepsilon\hat{\eta}}\frac{\partial^2}{\partial \hat{x}^2}\left((\tilde{h}+\tilde{Z})\hat{u}\right)\right. \\
&\qquad \left. + \left[\frac{(\tilde{z}-\tilde{Z})^3}{6}\right]_{-\tilde{h}}^{\varepsilon\hat{\eta}}\frac{\partial^2 \hat{u}}{\partial \hat{x}^2}\right) + \mathscr{O}(\sigma^4) \\
&= -\sigma^2\left(\left(\frac{\tilde{Z}^2}{2} - \frac{(\tilde{h}+\tilde{Z})^2}{2}\right)\frac{\partial^2}{\partial \hat{x}^2}\left((\tilde{h}+\tilde{Z})\hat{u}\right)\right. \\
&\qquad \left. + \left(-\frac{\tilde{Z}^3}{6} + \frac{(\tilde{h}+\tilde{Z})^3}{6}\right)\frac{\partial^2 \hat{u}}{\partial \hat{x}^2}\right) \\
&\quad +(\tilde{h}+\varepsilon\hat{\eta})\hat{u} + \mathscr{O}(\varepsilon\sigma^2, \sigma^4) \\
&= -\sigma^2\left(\left(-\frac{\tilde{h}^2}{2} + \tilde{h}\tilde{Z}\right)\frac{\partial^2}{\partial \hat{x}^2}\left((\tilde{h}+\tilde{Z})\hat{u}\right)\right. \\
&\qquad \left. + \left(\frac{\tilde{h}^3}{6} + \frac{\tilde{h}^2\tilde{Z}}{2} + \frac{\tilde{h}\tilde{Z}^2}{2}\right)\frac{\partial^2 \hat{u}}{\partial \hat{x}^2}\right) \\
&\quad +(\tilde{h}+\varepsilon\hat{\eta})\hat{u} + \mathscr{O}(\varepsilon\sigma^2, \sigma^4) \\
&= \sigma^2\left(\left(\tilde{Z}+\frac{\tilde{h}}{2}\right)\tilde{h}\frac{\partial^2(\tilde{h}\hat{u})}{\partial \hat{x}^2} + \left(\frac{\tilde{Z}^2}{2} - \frac{\tilde{h}^2}{6}\right)\tilde{h}\frac{\partial^2 \hat{u}}{\partial \hat{x}^2}\right) \\
&\quad +(\tilde{h}+\varepsilon\hat{\eta})\hat{u} + \mathscr{O}(\varepsilon\sigma^2, \sigma^4)
\end{aligned} \qquad (2.104)$$

Substituting equation (2.104) in the free surface equation (2.103) gives the Boussinesq continuity equation,

$$\begin{aligned}
&\frac{\partial \hat{\eta}}{\partial \hat{t}} + \frac{\partial}{\partial \hat{x}}((\tilde{h}+\varepsilon\hat{\eta})\hat{u}) \\
&\qquad +\sigma^2\frac{\partial}{\partial \hat{x}}\left(\left(\tilde{Z}+\frac{\tilde{h}}{2}\right)\tilde{h}\frac{\partial^2(\tilde{h}\hat{u})}{\partial x^2} + \left(\frac{\tilde{Z}^2}{2} - \frac{\tilde{h}^2}{6}\right)\tilde{h}\frac{\partial^2 \hat{u}}{\partial x^2}\right) = 0
\end{aligned} \qquad (2.105)$$

Returning the variables to dimensional, unscaled form via the transformations defined in

Sections 2.2.1 and 2.2.2, Nwogu's Boussinesq equations are,

$$\begin{aligned}
\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+g\frac{\partial \eta}{\partial x}+Z\frac{\partial^2}{\partial x^2}\left(h\frac{\partial u}{\partial t}\right)+\frac{Z^2}{2}\frac{\partial^2}{\partial x^2}\left(\frac{\partial u}{\partial t}\right) &= 0 \qquad (2.106)\\
\frac{\partial \eta}{\partial t}+\frac{\partial}{\partial x}((h+\eta)u)& \\
+\frac{\partial}{\partial x}\left(\left(Z+\frac{h}{2}\right)h\frac{\partial^2(hu)}{\partial x^2}+\left(\frac{Z^2}{2}-\frac{h^2}{6}\right)h\frac{\partial^2 u}{\partial x^2}\right) &= 0 \qquad (2.107)
\end{aligned}$$

Setting the arbitrary depth $Z=\theta h$, where $-1\leq\theta\leq 0$ the system is rewritten as,

$$\begin{aligned}
\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+g\frac{\partial \eta}{\partial x}+A_1h^2\frac{\partial^2}{\partial x^2}\left(\frac{\partial u}{\partial t}\right)+A_2h\frac{\partial^2}{\partial x^2}\left(h\frac{\partial u}{\partial t}\right) &= 0 \qquad (2.108)\\
\frac{\partial \eta}{\partial t}+\frac{\partial}{\partial x}((h+\eta)u)+\frac{\partial}{\partial x}\left(B_1h^3\frac{\partial^2 u}{\partial x^2}+B_2h^2\frac{\partial^2(hu)}{\partial x^2}\right) &= 0 \qquad (2.109)
\end{aligned}$$

where,

$$\begin{aligned}
A_1 &= \frac{\theta^2}{2} \qquad (2.110)\\
A_2 &= \theta \qquad (2.111)\\
B_1 &= \frac{\theta^2}{2}-\frac{1}{6} \qquad (2.112)\\
B_2 &= \theta+\frac{1}{2} \qquad (2.113)
\end{aligned}$$

At constant depth H the equations simplify to,

$$\begin{aligned}
\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+g\frac{\partial \eta}{\partial x}+\alpha H^2\frac{\partial^2}{\partial x^2}\left(\frac{\partial u}{\partial t}\right) &= 0 \qquad (2.114)\\
\frac{\partial \eta}{\partial t}+\frac{\partial}{\partial x}((H+\eta)u)+\beta H^3\frac{\partial^3 u}{\partial x^3} &= 0 \qquad (2.115)
\end{aligned}$$

where,

$$\begin{aligned}
\alpha &= \theta+\frac{\theta^2}{2} \qquad (2.116)\\
\beta &= \alpha+\frac{1}{3} \qquad (2.117)
\end{aligned}$$

A similar procedure in three dimensions will lead to Nwogu's two-dimensional extended Boussinesq equations [72],

$$\frac{\partial \eta}{\partial t}+\nabla\cdot((h+\eta)\,\mathbf{u})+\nabla\cdot\left(B_1h^3\nabla(\nabla\cdot\mathbf{u})+B_2h^2\nabla(\nabla\cdot(h\mathbf{u}))\right) = 0 \qquad (2.118)$$

$$\frac{\partial \mathbf{u}}{\partial t} + g\nabla\eta + (\mathbf{u}\cdot\nabla)\mathbf{u} + A_1 h^2 \nabla\left(\nabla\cdot\frac{\partial \mathbf{u}}{\partial t}\right) + A_2 h\nabla\left(\nabla\cdot\left(h\frac{\partial \mathbf{u}}{\partial t}\right)\right) = 0 \quad (2.119)$$

where the velocity vector $\mathbf{u}(x,y,t) = (u,v)$ represents the two-dimensional velocity field at depth $z = \theta h$ and ∇ represents the two-dimensional gradient operator with respect to the horizontal coordinates x and y.

2.4.4 Dispersion characteristics of the extended Boussinesq equations

The linearised form of Nwogu's constant depth equations (2.114)-(2.115) are,

$$\frac{\partial u}{\partial t} + g\frac{\partial \eta}{\partial x} + \alpha H^2 \frac{\partial^2}{\partial x^2}\left(\frac{\partial u}{\partial t}\right) = 0 \quad (2.120)$$

$$\frac{\partial \eta}{\partial t} + H\frac{\partial u}{\partial x} + \beta H^3 \frac{\partial^3 u}{\partial x^3} = 0 \quad (2.121)$$

Assuming a steady periodic wave form (2.65)-(2.66), and following the procedure described at the start of Section 2.3.3 one can derive a velocity magnitude,

$$b = \frac{\omega a}{kH(1-\beta(kH)^2)} \quad (2.122)$$

and the associated linearised dispersion relation,

$$\frac{\omega^2}{k^2} = gH\left(\frac{1-\beta(kH)^2}{1-\alpha(kH)^2}\right) \quad (2.123)$$

Applying a similar procedure to Beji and Nadaoka's equation (2.83)-(2.84) leads to an identical dispersion relation provided the free parameters satisfy,

$$\alpha = \frac{1+\alpha_B}{3} \quad (2.124)$$

The following analysis is based on Nwogu's equations with the understanding that the results apply equally as well to Beji and Nadaoka's extended equation system. Note also that the linear dispersion characteristics of the extended equation system of Madsen et al. [66] can be made equivalent by a similar relation between the free parameters [11].

The phase velocity is simply derived from the dispersion relation (2.123). Using the notation (2.49) introduced previously.

$$C^2 = gh\frac{1-\beta\phi^2}{1-\alpha\phi^2} \quad (2.125)$$

Some more notation is introduced to simplify the following algebra.

$$P = 1 - \alpha\phi^2 \tag{2.126}$$

$$Q = 1 - \beta\phi^2 \tag{2.127}$$

Hence,

$$C^2 = gh\frac{Q}{P} \tag{2.128}$$

The group velocity is derived from the definition (2.46) and the phase velocity (2.128).

$$C_g = C\left(1 - \frac{\phi^2}{3PQ}\right) \tag{2.129}$$

which is the form stated by Kirby et al. [54].

Following a similar argument to that used for the linear wave theory in Section 2.3.2 the shoaling gradient coefficient can be derived for the extended Boussinesq equations. Key points are the formation of the derivative of the group velocity component $F(\phi)$ with respect to x,

$$\frac{1}{F}\frac{\partial F}{\partial x} = T_1\frac{1}{\phi}\frac{\partial \phi}{\partial x} \tag{2.130}$$

where,

$$T_1 = -\frac{2\phi^2}{PQ}\left(\frac{1-\alpha\beta\phi^4}{3PQ-\phi^2}\right) \tag{2.131}$$

and the relationship between the wave number gradient and the depth gradient,

$$\frac{1}{k}\frac{\partial k}{\partial x} = \left(\frac{1}{T_2} - 1\right)\frac{1}{h}\frac{\partial h}{\partial x} \tag{2.132}$$

where,

$$T_2 = \frac{2}{PQ}\left(1 + \beta\left(\alpha\phi^2 - 2\right)\phi^2\right) \tag{2.133}$$

This finally leads to,

$$s = \frac{1}{2}\left(1 + \frac{T_1 - 1}{T_2}\right) \tag{2.134}$$

which with some algebraic manipulation can be shown to be equivalent to the form derived by Beji and Nadaoka for their extended equation system [11].

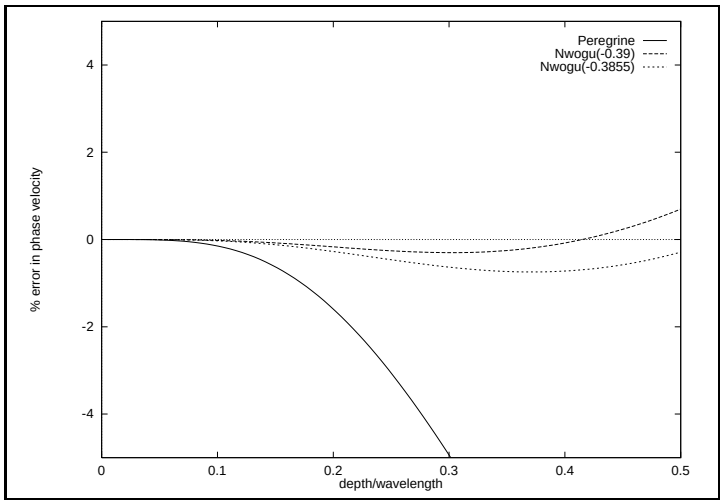
2.4.5 Comparison with linear theory

The free parameter α in the extended Boussinesq equation system is chosen to improve the linear dispersion characteristics of the equations. Nwogu [72] chose $\alpha = -0.39$ from a minimisation of the error in the phase velocity over the range $\sigma(=h/\lambda) \in [0, 1/2]$, where $\sigma = 1/2$ represents the so-called deep water limit. More recently Chen and Liu [27] have derived the value $\alpha = -0.3855$ from a least-squares fit of both the phase velocity and group velocity over the range $\sigma \in [0, 1/2]$.

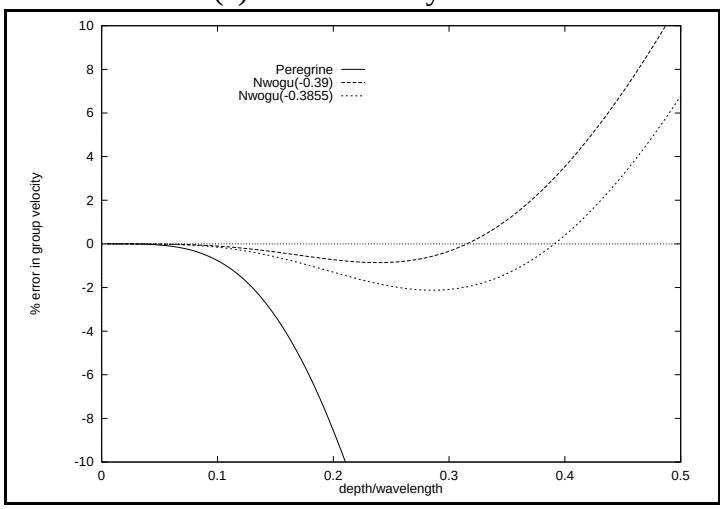
The dispersion characteristics are compared graphically in Figures 2.3(a)-(c). The graphs are plotted of the dispersion characteristics of Peregrine's Boussinesq equations, Nwogu's equations with $\alpha = -0.39$, and Nwogu's equations with $\alpha = -0.3855$, over the range $\sigma = H/\lambda \in [0, 1/2]$

Figure 2.3(a) compares the phase velocity predicted by the linear theory (2.44) with Nwogu's extended Boussinesq equations (2.125). The % error is again calculated using (2.76). The phase velocity of the original Boussinesq equations rapidly diverges from linear theory and the error exceeds 5% for $\sigma = 0.3$. In fact it diverges completely for $\sigma > 0.48$ [66], and hence the model will not capable of reproducing a deep water wave at $\sigma = 1/2$. The extended Boussinesq equations are both accurate and the error is less than 1% over the whole range, although Nwogu's value $\alpha = -0.39$ is clearly more accurate over most of the range. Figure 2.3(b) compares the group velocity predicted by the linear theory (2.47) with Nwogu's extended Boussinesq equations (2.129). The % error is determined in a similar way to the phase velocity. The group velocity of the original Boussinesq equations rapidly diverges again and is over 10% in error for $\sigma > 0.2$. The extended equations are clearly superior, although the errors are greater than with the phase velocity and with Nwogu's value, $\alpha = -0.39$, are over 10% in error at the deep water limit $\sigma = 1/2$. The value $\alpha = -0.3855$ is less accurate than Nwogu's for $\sigma < 0.35$ but over the whole range provides a better model and is only 7% in error at worst. Figure 2.3(c) compares the shoaling coefficient predicted by the linear theory (2.62) with Nwogu's extended Boussinesq equations (2.134). This graph is of the actual values, as the shoaling coefficient passes through zero on the interval $\sigma \in [0, 1/2]$. The shoaling coefficient of the original Boussinesq equations diverges significantly from the linear theory for $\sigma > 0.15$. The extended Boussinesq equations are both accurate for $\sigma < 0.3$ but both become significantly inaccurate at larger values.

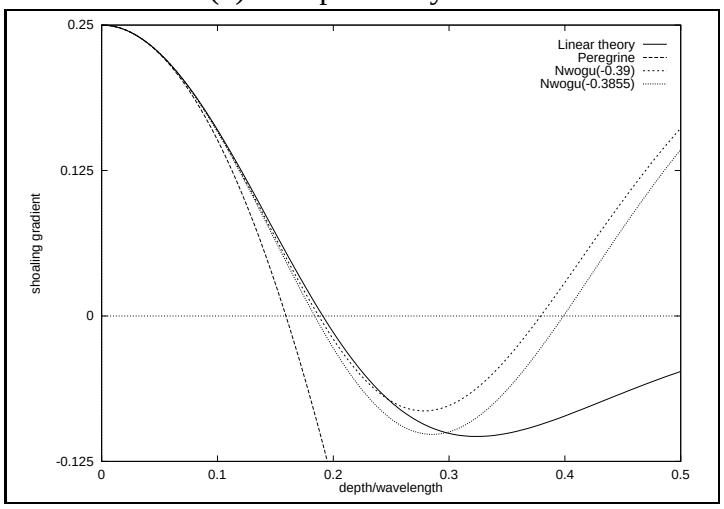
Clearly it is possible to accurately model the phase velocity and group velocity of a wave train over the whole range of depths with the extended Boussinesq equations. The accuracy of the shoaling coefficient is a more discriminating test. The original Boussinesq equation system will be inaccurate for $\sigma > 0.15$, and the extended Boussinesq equation systems, despite being formally valid at the deep water limit, $\sigma = 1/2$, will be signifi-



(a) Phase velocity error



(b) Group velocity error



(c) Shoaling gradient

Figure 2.3: Comparison of the dispersion characteristics of the extended Boussinesq equations and linear theory

cantly inaccurate for $\sigma > 0.3$.

In general the best agreement with linear theory will be obtained by choosing an appropriate value of α for each particular problem. The range of depths, and hence σ, in the problem will determine the region over which an appropriate function of phase velocity, group velocity or shoaling coefficient can be minimised.

2.5 Some simpler nonlinear dispersive equations

The equations described in the previous sections have several unusual features compared to the standard fluid flow models such as the Euler and Navier-Stokes equations. Peregrine's equation system (2.35)-(2.36) contains a third order derivative term of mixed space and time form, and Nwogu's extended system (2.108)-(2.109) additionally contains a third order spatial derivative. In devising numerical methods for these equations it will be important to investigate the numerical approximation of these particular terms. Here it is shown that by making further simplifying assumptions it is possible to derive simpler nonlinear, dispersive wave models which can be used for the initial investigation into possible numerical methods.

If it is assumed that waves travel in only the positive x-direction then it can be shown [80] that to a first approximation the waves have a steady form, ie. the nondimensional transformation,

$$\tilde{X} = \tilde{x} - \tilde{t} \tag{2.135}$$

can be introduced. Assuming that there is a slight variation with time or distance travelled, the appropriate scaling is [80],

$$\hat{X} = \sigma \tilde{X} \tag{2.136}$$

$$\hat{T} = \sigma^3 \tilde{T} \tag{2.137}$$

The free surface and horizontal velocity are scaled as before, according to the definitions (2.11) and (2.12) respectively. Substituting into Peregrine's Boussinesq equation system (2.35)-(2.36) leads to,

$$\sigma^2 \frac{\partial \hat{\bar{u}}}{\partial \hat{T}} - \frac{\partial \hat{\bar{u}}}{\partial \hat{X}} + \varepsilon \hat{\bar{u}} \frac{\partial \hat{\bar{u}}}{\partial \hat{X}} + \frac{\partial \hat{\eta}}{\partial \hat{X}} + \frac{\sigma^2}{3} \frac{\partial^3 \hat{\bar{u}}}{\partial \hat{X}^3} - \frac{\sigma^4}{3} \frac{\partial^3 \hat{\bar{u}}}{\partial \hat{X}^2 \partial \hat{T}} = 0 \tag{2.138}$$

$$\sigma^2 \frac{\partial \hat{\eta}}{\partial \hat{T}} - \frac{\partial \hat{\eta}}{\partial \hat{X}} + \frac{\partial}{\partial \hat{X}} \left((1 + \varepsilon \hat{\eta}) \hat{\bar{u}} \right) = 0 \tag{2.139}$$

To the lowest order of approximation both equations reduce to,

$$\frac{\partial \hat{\eta}}{\partial \hat{X}} \;=\; \frac{\partial \hat{\bar{u}}}{\partial \hat{X}} + \mathscr{O}(\varepsilon, \sigma^2) \tag{2.140}$$

under the assumption that the initial conditions are consistent with this statement. Hence by integrating, assuming that the order stays the same,

$$\hat{\eta} \;=\; \hat{\bar{u}} + \mathscr{O}(\varepsilon, \sigma^2) \tag{2.141}$$

Adding equations (2.138) and (2.139) and using the relation (2.141),

$$2\sigma^2 \frac{\partial \hat{\eta}}{\partial \hat{T}} + 3\varepsilon \hat{\eta} \frac{\partial \hat{\eta}}{\partial \hat{X}} + \frac{\sigma^2}{3} \frac{\partial^3 \hat{\eta}}{\partial \hat{X}^3} \;=\; \mathscr{O}(\sigma^4) \tag{2.142}$$

where it is assumed from the definition of the Ursell Number (2.10) that $\varepsilon = \mathscr{O}(\sigma^2)$. The homogeneous form of this equation is known as the Korteweg-de Vries (KdV) equation [35]. In dimensional variables it reads,

$$\frac{\partial \eta}{\partial t} + \sqrt{gH} \frac{\partial \eta}{\partial x} + \frac{3}{2} \eta \frac{\partial \eta}{\partial x} + \frac{H^2}{6} \sqrt{gH} \frac{\partial^3 \eta}{\partial x^3} \;=\; 0 \tag{2.143}$$

To first order,

$$\frac{\partial \eta}{\partial t} \;=\; -\sqrt{gH} \frac{\partial \eta}{\partial x} \tag{2.144}$$

and this can be used to transform equation (2.143) into another form [78] which has become known as the Regularised Long Wave (RLW) equation [13].

$$\frac{\partial \eta}{\partial t} + \sqrt{gH} \frac{\partial \eta}{\partial x} + \frac{3}{2} \eta \frac{\partial \eta}{\partial x} - \frac{H^2}{6} \frac{\partial^3 \eta}{\partial x^2 \partial t} \;=\; 0 \tag{2.145}$$

Both of the equations (2.143) and (2.145) are nonlinear and dispersive but are of a single variable. These simpler equations will be used in the initial stages of the development of the numerical methods for the extended Boussinesq model.

Both of these equations have exact solutions in the form of a steady translating wave of elevation known as a solitary wave. Such a solution will be useful for validation of any proposed numerical method. The procedure for deriving the KdV solitary wave is described by Drazin [35]. Here the RLW solitary wave is derived by a similar method. Consider the RLW equation,

$$\frac{\partial \eta}{\partial t} + a \frac{\partial \eta}{\partial x} + b \eta \frac{\partial \eta}{\partial x} - \delta \frac{\partial^3 \eta}{\partial x^2 \partial t} \;=\; 0 \tag{2.146}$$

where,

$$a = \sqrt{gH} \tag{2.147}$$

$$b = \frac{3}{2} \tag{2.148}$$

$$\delta = \frac{H^2}{6} \tag{2.149}$$

and assume a solution of steady translating form.

$$\eta(x,t) = F(x-ct) = F(X) \tag{2.150}$$

c is a free parameter relating to the speed of the wave. Changing variables to X and using the chain rule gives,

$$\begin{aligned} \frac{\partial \eta}{\partial t} &= \frac{dF}{dX}\frac{\partial X}{\partial t} \\ &= -c\frac{dF}{dX} \end{aligned} \tag{2.151}$$

$$\begin{aligned} \frac{\partial \eta}{\partial x} &= \frac{dF}{dX}\frac{\partial X}{\partial x} \\ &= \frac{dF}{dX} \end{aligned} \tag{2.152}$$

Substituting this in equation (2.146),

$$\begin{aligned} -c\frac{dF}{dX} + a\frac{dF}{dX} + bF\frac{dF}{dX} + c\delta\frac{d^3F}{dX^3} &= 0 \\ (a-c)\frac{dF}{dX} + \frac{b}{2}\frac{d(F^2)}{dX} + c\delta\frac{d^3F}{dX^3} &= 0 \end{aligned} \tag{2.153}$$

This equation can be integrated. The arbitrary constant introduced by this process can be taken to be zero if F and its first and second derivative vanish as $X \to \infty$.

$$(a-c)F + \frac{b}{2}F^2 + c\delta\frac{d^2F}{dX^2} = 0 \tag{2.154}$$

Multiplying by the first derivative of F and rearranging,

$$\begin{aligned} (a-c)F\frac{dF}{dX} + \frac{b}{2}F^2\frac{dF}{dX} + c\delta\frac{d^2F}{dX^2}\frac{dF}{dX} &= 0 \\ \left(\frac{a-c}{2}\right)\frac{d(F^2)}{dX} + \frac{b}{6}\frac{d(F^3)}{dX} + \frac{c\delta}{2}\frac{d}{dX}\left(\frac{dF}{dX}\right)^2 &= 0 \end{aligned} \tag{2.155}$$

Integrating this, with the arbitrary constant again set to zero,

$$\begin{aligned}\left(\frac{a-c}{2}\right)F^2+\frac{b}{6}F^3+\frac{c\delta}{2}\left(\frac{dF}{dX}\right)^2 &= 0\\ \left(\frac{dF}{dX}\right)^2+F^2\left(\frac{a-c}{c\delta}+\frac{b}{3c\delta}F\right) &= 0 \end{aligned} \tag{2.156}$$

Now try a solution of the form,

$$F(X) = \Psi\, sech^2(\Theta X) \tag{2.157}$$

for arbitrary scalars Ψ and Θ. Substituting this in equation (2.156),

$$\begin{aligned}4\Psi^2\Theta^2\, sech^4(\Theta X)\, tanh^2(\Theta X)&\\ +\Psi^2\, sech^4(\Theta X)\left(\frac{a-c}{c\delta}+\frac{b}{3c\delta}\Psi\, sech^2(\Theta X)\right) &= 0\\ \Psi^2\, sech^4(\Theta X)\left[4\Theta^2\, tanh^2(\Theta X)\right.&\\ \left.+\frac{a-c}{c\delta}+\frac{b}{3c\delta}\Psi\, sech^2(\Theta X)\right] &= 0\\ 4\Theta^2\, tanh^2(\Theta X)+\frac{a-c}{c\delta}+\frac{b}{3c\delta}\Psi\, sech^2(\Theta X) &= 0\\ \frac{c-a}{c\delta}\, cosh^2(\Theta X)-4\Theta^2\, sinh^2(\Theta X) &= \frac{b}{3c\delta}\Psi \end{aligned} \tag{2.158}$$

Comparing this to the trigonometric identity,

$$cosh^2(\theta)-sinh^2(\theta) = 1 \tag{2.159}$$

leads to the relationships,

$$c-a = c\delta\, j \tag{2.160}$$

$$4\Theta^2 = j \tag{2.161}$$

$$\Psi b = 3c\delta\, j \tag{2.162}$$

where j is an arbitrary constant. Rearranging these equations leads to expressions for Ψ and Θ.

$$j = \frac{c-a}{c\delta} \tag{2.163}$$

$$\Theta^2 = \frac{c-a}{4c\delta} \tag{2.164}$$

$$\Psi = 3\left(\frac{c-a}{b}\right) \tag{2.165}$$

Substituting these in the solitary wave solution (2.157),

$$F(X) \;=\; 3\left(\frac{c-a}{b}\right)\; sech^2\left(\frac{1}{2}\sqrt{\frac{c-a}{c\delta}}X\right) \tag{2.166}$$

and, using the definitions (2.147)-(2.149), the solitary wave solution of the RLW equation is,

$$\eta(x,t) \;=\; 2\left(c-\sqrt{gH}\right)\; sech^2\left(\frac{1}{2H}\sqrt{6\left(\frac{c-\sqrt{gH}}{c}\right)}(x-ct)\right) \tag{2.167}$$

Following a similar procedure for the KdV equation (2.143), noting the change to the coefficient δ, leads to a slight modification to equation (2.153),

$$(a-c)\frac{dF}{dX}+\frac{b}{2}\frac{d(F^2)}{dX}+\delta\frac{d^3F}{dX^3} \;=\; 0 \tag{2.168}$$

and to a solitary wave of the form,

$$\eta(x,t) \;=\; 2\left(c-\sqrt{gH}\right)\; sech^2\left(\frac{1}{2H}\sqrt{6\left(\frac{c-\sqrt{gH}}{\sqrt{gH}}\right)}(x-ct)\right) \tag{2.169}$$

The free parameter c determines the speed and amplitude of the wave, and describes a family of solitary waves for a particular depth H. Benjamin et al. [13] remark that in much of the numerical work on the KdV equation the amplitude, and hence velocity, of the solitary wave are chosen to be too large for the shallow-water approximation. They attribute this to the treatment of the equation without considering the physical relevance of the coefficients. Here the value of c is limited by the shallow water condition (2.8), ie. that the amplitude of the wave must be much less than the depth H. For the RLW and KdV equations' solitary wave solutions this can be stated as,

$$\begin{aligned} 2\left(c-\sqrt{gH}\right) \;&\ll\; H \\ c \;&\ll\; \frac{H}{2}+\sqrt{gH} \end{aligned} \tag{2.170}$$

If $c=\sqrt{gH}$ the wave has zero amplitude, and if $c<\sqrt{gH}$ then calculating Θ from equation (2.164) will involve the square root of a negative number. Hence c is restricted to the range,

$$\sqrt{gH} \;<\; c \;\ll\; \sqrt{gH}+\frac{H}{2} \tag{2.171}$$

In practice c is chosen from the formula,

$$c = \sqrt{gH} + \delta_s \frac{H}{2} \tag{2.172}$$

for the positive scalar parameter $\delta_s \ll 1$. This will ensure solitary wave solutions compatible with the shallow water model from which they are derived.

2.6 The chosen form of the equations

In this chapter it has been shown that there are a variety of mathematical models for nonlinear, dispersive, shallow water waves. Peregrine's equation system (2.35)-(2.36) appears to have the simplest differential form but the dispersion characteristics limit its range of application. Extended Boussinesq equations increase the range of depth over which the model is valid but introduce extra complexity into the differential equations. However it is felt that increasing the range of the model is essential if it is to be used for real applications. It will be shown in Chapter 4 that Nwogu's extended Boussinesq equation system is the appropriate choice here. The form of the equations is more amenable to the numerical methods proposed than that of the other extended equation systems. Further extensions to the Boussinesq equations are not considered here as the advantages they provide are offset by the increased complexity in the equations.

In the previous section two simpler nonlinear dispersive equations were derived. The RLW equation (2.145) contains the mixed third order derivative present in all the Boussinesq equation systems. The KdV equation (2.143) contains a third order spatial derivative which is a feature introduced by the extended Boussinesq models. Hence both of these equations will be of use in the following chapter where appropriate numerical methods are discussed and developed. The solitary wave solutions of the RLW and KdV equations, (2.167) and (2.169) respectively, will be used to assess the performance of these methods.

Chapter 3

Numerical Methods

3.1 Introduction

In this chapter the RLW and KdV equations introduced in Section 2.5 are used to investigate numerical solution procedures for dispersive wave equations. Finite difference and finite element spatial discretisation schemes are investigated, coupled with adaptive time integration software. The aim of the work in this chapter is to produce a scheme that will also apply to the one-dimensional extended Boussinesq equations, and eventually to the two-dimensional equations. Possible methods will be evaluated with these considerations in mind.

The method of lines [18] is used as the basic framework for the numerical methods developed in this work. This is a technique for decoupling the spatial and temporal discretisation process. A system of partial differential equations is first semi-discretised in space, for example with a finite difference or finite element method, reducing the problem to that of a system of ordinary differential equations in time. This system can then be solved with an appropriate time integration method. In this work the method of lines approach will allow existing adaptive order, adaptive step size time integration software to be used.

This approach will be applied as follows. Finite difference methods and finite element methods are introduced in Section 3.2 for the spatial approximation of the equations, and several discretisation methods are proposed. These are illustrated with reference to the RLW and KdV equations, (2.145) and (2.143) respectively, introduced in the previous chapter. In particular a finite difference method previously used for the extended Boussi-

nesq equation system [106] and a new finite element method are proposed and analysed. The application of adaptive time integration packages to the spatially discretised equations is described in Section 3.3 and their suitability for this class of problem is assessed.

Numerical experiments with the methods proposed in the preceding sections for the RLW and KdV equations are described in Section 3.4. The exact solitary wave solutions for these equations, derived in the previous chapter in Section 2.5, are used in order to assess the accuracy of the schemes. The results are reviewed in Section 3.5 and the extension of these methods to the one-dimensional extended Boussinesq equation system, and to the solution of the two-dimensional equations is considered.

3.2 Spatial discretisation

3.2.1 Introduction

Within the method-of-lines framework introduced in the previous section the spatial approximation is considered first. Before the spatial approximation the equations are rewritten in a more convenient form. The RLW equation (2.145) is written as,

$$\dot{\eta} - \delta \frac{\partial^2 \dot{\eta}}{\partial x^2} = -\frac{\partial f}{\partial x} \tag{3.1}$$

and the KdV equation (2.143) as,

$$\dot{\eta} = -\frac{\partial f}{\partial x} - \delta \frac{\partial^3 \eta}{\partial x^3} \tag{3.2}$$

where,

$$\dot{\eta} = \frac{\partial \eta}{\partial t} \tag{3.3}$$

$$f = a\eta + \frac{b}{2}\eta^2 \tag{3.4}$$

In order to solve these partial differential equation problems it is necessary to specify appropriate boundary conditions and initial conditions. The problem domain is $x \in [X_1, X_2]$ and $t \in [0, T]$. The examples to be solved in Section 3.4 use the solitary wave solutions (2.167) and (2.169). Here the spatial domain is taken to be large enough so that the solution and its derivatives can be assumed to be zero at the boundaries $x = X_1$ and $x = X_2$. Initial conditions for the time integration are provided by the solitary wave solutions (2.167) and (2.169) evaluated at $t = 0$.

3.2.2 Finite difference methods

3.2.2.1 Outline of the basic method

The finite difference method is developed by first partitioning the spatial domain into a set of non-coincident points, or nodes. Here, for simplicity of presentation, the discussion is limited to a constant spacing between adjacent nodes. The spatial domain of the problem $x \in [X_1, X_2]$ is covered with a set of $N+1$ equally spaced points $\{x_1, x_2, \ldots, x_N, x_{N+1}\}$ where,

$$x_i = X_1 + (i-1)\Delta, \qquad 1 \leq i \leq N+1 \tag{3.5}$$

$$\Delta = \frac{X_2 - X_1}{N} \tag{3.6}$$

and Δ is termed the grid size or grid spacing. A function is approximated over this set of points, or grid, in terms of its value at each node. Spatial derivatives are approximated directly at a given point by using Taylor series expansions in space to relate the derivative at that point to adjacent nodal solution values [113]. This will lead to a global equation system with a banded matrix structure, which can be efficiently solved.

The equations to be considered here, (3.1) and (3.2), contain first, second and third spatial derivatives. Finite difference approximations of second and fourth order accuracy, with respect to the grid size, are shown in equations (3.7)-(3.12). The leading truncation error terms from the Taylor series approximations are shown to make precise the accuracy of these expressions. For the first derivative,

$$\frac{df}{dx}(x_i) = \frac{1}{2\Delta}(f_{i+1} - f_{i-1}) - \frac{1}{6}\Delta^2 \frac{d^3 f}{dx^3}(x_i) + \mathcal{O}(\Delta^4) \tag{3.7}$$

$$\frac{df}{dx}(x_i) = \frac{1}{12\Delta}(f_{i-2} - 8f_{i-1} + 8f_{i+1} - f_{i+2}) - \frac{1}{90}\Delta^4 \frac{d^5 f}{dx^5}(x_i) + \mathcal{O}(\Delta^6) \tag{3.8}$$

For the second spatial derivative,

$$\frac{d^2 f}{dx^2}(x_i) = \frac{1}{\Delta^2}(f_{i-1} - 2f_i + f_{i+1}) + \frac{1}{12}\Delta^2 \frac{d^4 f}{dx^4}(x_i) + \mathcal{O}(\Delta^4) \tag{3.9}$$

$$\frac{d^2 f}{dx^2}(x_i) = \frac{1}{12\Delta^2}(-f_{i-2} + 16f_{i-1} - 30f_i + 16f_{i+1} - f_{i+2}) + \frac{1}{90}\Delta^4 \frac{d^6 f}{dx^6}(x_i) + \mathcal{O}(\Delta^6) \tag{3.10}$$

For the third spatial derivative,

$$\begin{aligned}\frac{d^3f}{dx^3}(x_i) &= \frac{1}{2\Delta^3}(-f_{i-2}+2f_{i-1}-2f_{i+1}+f_{i+2}) \\ &\quad + \frac{7}{60}\Delta^2\frac{d^5f}{dx^5}(x_i) + \mathcal{O}(\Delta^4) \qquad (3.11)\\ \frac{d^3f}{dx^3}(x_i) &= \frac{1}{8\Delta^3}(f_{i-3}-8f_{i-2}+13f_{i-1}-13f_{i+1}+8f_{i+2}-f_{i+3}) \\ &\quad + \frac{7}{120}\Delta^4\frac{d^7f}{dx^7}(x_i) + \mathcal{O}(\Delta^6) \qquad (3.12)\end{aligned}$$

These finite difference formulae are used in the following sections to approximate the model dispersive equations.

All the difference stencils presented above are centred on the node i. When discontinuous or near-discontinuous features are present in a physical system it is possible for such centred schemes to perform badly as they do not take account of the direction of propagation. These problems are commonly remedied by the use of upwind stencils and more sophisticated flux-limiting schemes, generally at the expense of introducing artificial diffusion into the scheme [76]. The diffusion is in the form of second order spatial derivatives and within the Boussinesq framework these artificial terms will be larger than the physical dispersive terms, and so will degrade the solution [1]. For the equations based on the Boussinesq wave theory introduced in Chapter 2 the nonlinearity is assumed to be weak by the definition of the small parameter $\varepsilon \ll 1$, and restrictions to long wavelengths compared to the depth, parameterised by $\sigma \ll 1$, rule out discontinuities. Hence only the centred schemes are considered here.

3.2.2.2 Previous work

Peregrine [78] proposed one of the first finite difference methods for the RLW equation, using second order spatial and temporal approximations. The reported results were dissipative and accuracy was only maintained by choosing a sufficiently small mesh spacing. Vliegenthart [102] constructed a more accurate finite difference scheme for the KdV equation and also introduced the possibility of a dissipative scheme by a Lax-Wendroff type approach. This was used to solve problems with discontinuous initial conditions for which the centred schemes produced poor results. Note however that such discontinuous initial conditions are inconsistent with the mathematical model of shallow water waves described here. Eilbeck and McGuire [36] analysed Peregrine's finite difference scheme for the RLW equation and modified it to improve the accuracy by introducing a more accurate discretisation of the first order derivative terms and a three level approximation to the time derivative. They used this scheme to investigate solitary wave interactions for

the RLW equation [37]. Greig and Morris [45] designed what they term as a hopscotch finite difference method for the KdV equation. By ensuring that this scheme was numerically conservative, ie. non-dissipative, they obtained excellent results for the interaction between two solitary waves.

Here finite difference methods are proposed based on considerations of the physical size of derivative terms in the nondimensionalised scaled equations and of the truncation errors produced by the finite difference approximations.

3.2.2.3 Finite difference methods for the RLW equation

A fully second order finite difference method for this equation using the stencils (3.7) and (3.9) will result in a third spatial derivative in the truncation error, introduced by the first derivative approximation (3.7). In the scaled, nondimensional form introduced in Section 2.5 this third derivative is formally of the same magnitude as the dispersion present and so will produce additional, numerical dispersion at a similar level to the physical dispersion present in the system. Such a scheme is therefore likely to perform badly in numerical tests.

A fully fourth order finite difference method for this equation using the stencils (3.8) and (3.10) produces a scheme with at least fifth order derivatives in the truncation error. This will clearly have sufficient accuracy but at the expense of an enlarged support at a node i; $i-2$ to $i+2$ for the approximation of the first and second derivatives.

Wei and Kirby [106] designed an accurate finite difference scheme for Nwogu's extended Boussinesq equations. Noting that the non-physical dispersion present in a second order finite difference spatial discretisation was due only to the first derivative, they differenced these terms to fourth order, and the dispersive terms to second order. They integrated in time with a fourth order predictor-corrector method to ensure that no more dispersive errors were introduced.

Here it can also be seen that the non-physical dispersion in the second order scheme is produced only by the low order approximation to the first derivative (3.7). By replacing this with the fourth order approximation (3.8) and differencing the other term to second order with (3.9) a scheme is produced that has sufficient accuracy for this equation. The precise form of this difference scheme is stated below as it will be referred to later. For the RLW equation (3.1) this finite difference scheme with the leading truncation error terms at a node i is,

$$\begin{aligned}\dot{\eta}_i - \frac{\delta}{\Delta^2}\left(\dot{\eta}_{i-1} - 2\dot{\eta}_i + \dot{\eta}_{i+1}\right) &= -\frac{1}{12\Delta}\left(f_{i-2} - 8f_{i-1} + 8f_{i+1} - f_{i+2}\right) \\ &\quad -\frac{\delta}{12}\Delta^2\frac{\partial^4\dot{\eta}}{\partial x^4}(x_i) + \frac{1}{90}\Delta^4\frac{\partial^5 f}{\partial x^5}(x_i) \qquad (3.13)\end{aligned}$$

The scheme is only second order accurate in space overall but the high order derivatives in the truncation error will not produce any non-physical dispersion and hence will allow accurate computations for this equation.

3.2.2.4 Finite difference methods for the KdV equation

A fully second order finite difference method for this equation using the stencils (3.7) and (3.11) will introduce a third spatial derivative, from the first derivative (3.7), as the leading truncation error term. In the KdV equation (3.2) this will directly introduce additional numerical dispersion of the same form as that already present in the system. Again this scheme is therefore likely to perform badly in numerical tests.

A fully fourth order finite difference method for this equation using the stencils (3.8) and (3.12) produces a scheme with at least fifth order derivatives in the truncation error. This will have sufficient accuracy for the KdV equation but at the expense of an enlarged support at a node i; $i-3$ to $i+3$ due to the approximation of the third derivative (3.12).

The approach of Wei and Kirby [106] can again be applied to this equation by replacing the first derivative with the fourth order approximation (3.8), and the third derivative with the second order approximation (3.11). This scheme is sufficiently accurate for this problem and has a smaller support than the fully fourth order scheme; $i-2$ to $i+2$ at node i. The form of this difference scheme is stated below as it will be referred to later. For the KdV equation (3.2) this finite difference scheme with the leading truncation error terms at a node i is,

$$\begin{aligned}\dot{\eta}_i &= -\frac{1}{12\Delta}\left(f_{i-2}-8f_{i-1}+8f_{i+1}-f_{i+2}\right)-\frac{\delta}{2\Delta^3}\left(-\eta_{i-2}+2\eta_{i-1}-2\eta_{i+1}+\eta_{i+2}\right)\\ &\quad -\frac{7\delta}{60}\Delta^2\frac{\partial^5\eta}{\partial x^5}(x_i)+\frac{1}{90}\Delta^4\frac{\partial^5 f}{\partial x^5}(x_i) \qquad (3.14)\end{aligned}$$

Similarly to the RLW equation scheme (3.13) this is only second order accurate but should not produce significant numerical dispersion.

3.2.3 Finite element methods

3.2.3.1 Outline of the basic method

As before with the finite difference method, the spatial domain $x \in [X_1, X_2]$ is covered with a set of $N+1$ non-coincident points $\{x_1, x_2, \ldots, x_N, x_{N+1}\}$. Each pair of adjacent points defines a spatial interval, or element, and the set of N non-overlapping elements, or mesh, completely covers the domain. A set of basis functions are defined that form the approximation over the mesh. One of the simplest such approximations is the set of linear

functions $\{\phi_i(x), i = 1,\ldots,N+1\}$ defined by,

$$\phi_i(x_j) \;=\; \delta_{ij} \tag{3.15}$$

where δ_{ij} is the Dirac delta function. They interpolate nodal values of the function. Figure 3.1(a) shows the linear basis function defined by (3.15) at node i. This is termed the global basis function [113]. This basis function is only non-zero on the two elements surrounding node i. This has the advantage, similar to the finite difference method, that spatial approximations at node i involve only nearby nodal values. Considering an element e of the mesh, it can be seen in Figure 3.1(b) that only the two adjacent global basis functions are nonzero on this element. The parts of these two global basis functions defined on the element are termed the local, or element, basis functions [113]. They define a linear interpolation $u^e(x)$ over the element in terms of the nodal values u_i and u_{i+1}.

$$u^e \;=\; \phi_i^e u_i + \phi_{i+1}^e u_{i+1} \tag{3.16}$$

The linear basis function ϕ_i is piecewise continuous over the mesh. Its first derivative will be discontinuous but can be uniquely defined on each element. Other basis functions can be defined with more, or less, continuity than this [113]. More generally one can define the following function spaces, known as Sobolev spaces [51], over the spatial domain Ω. First define $v(x)$ as belonging to the set of square-integrable functions on the domain Ω, $L^2(\Omega)$.

$$v \in L^2(\Omega) \qquad if \int_\Omega v^2 \, d\Omega < \infty \tag{3.17}$$

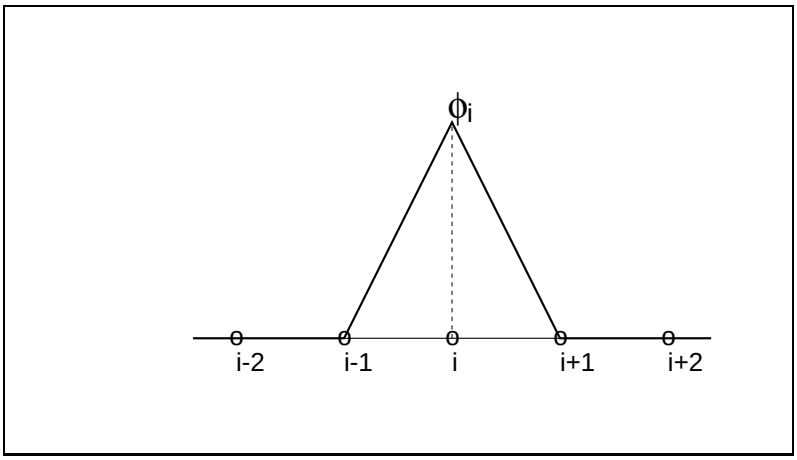
Then the k^{th} Sobolev space $\mathscr{H}^k$ is defined by,

$$v \in \mathscr{H}^k(\Omega) \qquad if\ v,\ and\ \{\frac{d^i v}{dx^i}, i = 1,\ldots,k\} \in L^2(\Omega) \tag{3.18}$$

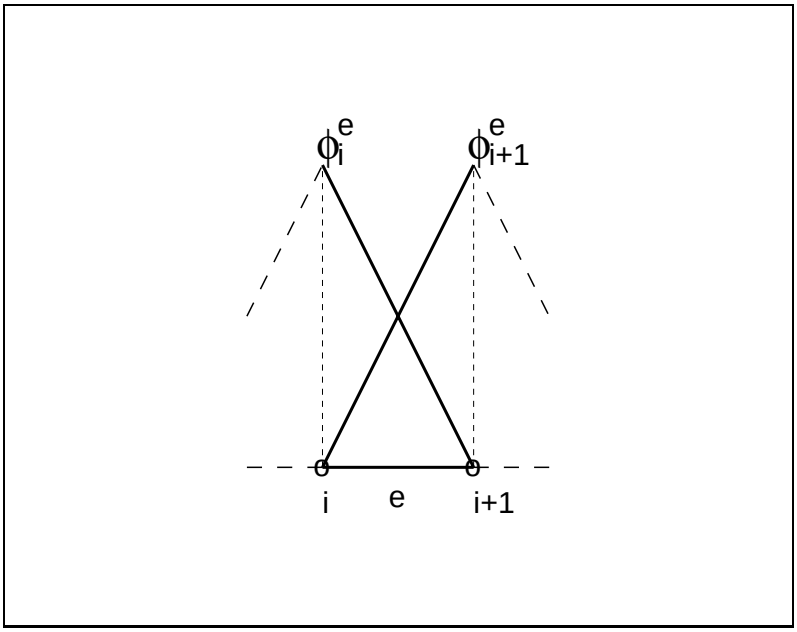
This definition imposes continuity restrictions on the functions v. For example the linear functions $\phi_i \in \mathscr{H}^1$ since ϕ_i and its first derivative are square integrable on the domain. The second derivative is infinite at element boundaries [113] hence $\phi_i \notin \mathscr{H}^2$. The most compact $\mathscr{H}^2$ function is the Hermite cubic [100] where the function and its first derivative are interpolated continuously over the mesh.

The finite element discretisation procedure is illustrated here for a simple heat conduction equation [113], for the unknown function $u(x,t)$ and constant coefficient γ. Using the notation (3.3) for the time derivative,

$$\dot{u} \;=\; \gamma \frac{\partial^2 u}{\partial x^2} \tag{3.19}$$



(a) The global basis function at node *i*



(b) The local basis functions on element *e*

Figure 3.1: The linear finite element basis function

The full specification of this initial-boundary value problem, defined on $x \in [X_1, X_2]$ and $t \in [0, T]$ is completed with appropriate boundary and initial conditions,

$$u(X_1, t) = U_1(t) \tag{3.20}$$

$$u(X_2, t) = U_2(t) \tag{3.21}$$

$$u(x, 0) = U_0(x) \tag{3.22}$$

where $U_1(t)$, $U_2(t)$ and $U_0(x)$ are known functions. To develop the finite element method the partial differential equation (3.19) is multiplied by a test function $\psi_i(x)$ chosen from an appropriate function space, and integrated over the spatial domain.

$$\int_{X_1}^{X_2} \psi_i \left(\dot{u} - \gamma \frac{\partial^2 u}{\partial x^2} \right) dx = 0 \tag{3.23}$$

Integration by parts can be used to reduce the order of the highest spatial derivatives in this expression as this will reduce the continuity requirements on the interpolation of the unknown function [113].

$$\int_{X_1}^{X_2} \left(\psi_i \dot{u} + \gamma \frac{\partial \psi_i}{\partial x} \frac{\partial u}{\partial x} \right) dx = \gamma \left[\psi_i \frac{\partial u}{\partial x} \right]_{X_1}^{X_2} \tag{3.24}$$

Note that if the expression (3.23) were used to develop the finite element method the approximation to u would have to belong to $\mathscr{H}^2$ and ψ_i to L^2. In the form (3.24) the maximum degree of continuity required has been reduced and both ψ_i and the approximation to u can be chosen from $\mathscr{H}^1$.

The boundary term on the right hand side can be neglected in this case as the boundary conditions (3.20) and (3.21) specify values of u at those points. The approximation to the unknown function u is produced by an interpolation over the set of basis functions known as the trial space [113],

$$u(x, t) \approx \sum_{j=1}^{N+1} \phi_j(x) u_j(t) \tag{3.25}$$

$$u_1(t) = U_1(t) \tag{3.26}$$

$$u_{N+1}(t) = U_2(t) \tag{3.27}$$

Within the method of lines framework the time derivative $\dot{u}$ is interpolated in a similar fashion. Inserting the approximations into the integral statement (3.24),

$$\left(\int_{X_1}^{X_2} \psi_i \phi_j \, dx \right) \dot{u}_j + \gamma \left(\int_{X_1}^{X_2} \frac{\partial \psi_i}{\partial x} \frac{\partial \phi_j}{\partial x} \, dx \right) u_j = 0 \tag{3.28}$$

where the summation over the repeated index j is implied. The matrix multiplying the time derivative is formed by the integral of the product of the trial and test functions and is generally termed the mass matrix, M_{ij} [113]. The other integral, involving the product of the derivatives of the two functions is termed the stiffness matrix K_{ij} [113].

$$M_{ij} = \int_{X_1}^{X_2} \psi_i \phi_j \, dx \tag{3.29}$$

$$K_{ij} = \int_{X_1}^{X_2} \frac{\partial \psi_i}{\partial x} \frac{\partial \phi_j}{\partial x} \, dx \tag{3.30}$$

Within the finite element framework this global equation system can be assembled efficiently by considering each element in turn. The integral over the spatial domain can then be split into a sum of element integrals, resulting in element-by-element assembly procedures for the mass and stiffness matrices. Over each element the solution is interpolated by (3.16) and only the local basis functions are non-zero, hence each element integral need only be evaluated for the basis functions local to it. All other basis functions are zero on the element and do not contribute to the integral. This is illustrated for the mass matrix here.

$$M_{ij}\dot{u}_j = \sum_{k=1}^{N} \int_{x_k}^{x_{k+1}} \psi_i \phi_j \dot{u}_j \, dx \tag{3.31}$$

$$= \sum_{k=1}^{N} \left(\int_{x_k}^{x_{k+1}} \psi_i \left(\phi_k^e \dot{u}_k + \phi_{k+1}^e \dot{u}_{k+1} \right) dx, \qquad i = k, k+1 \right) \tag{3.32}$$

This means that the weighted integral form of the partial differential equation (3.23) has been reduced to a sum of element integrals, and since the test function and trial space are known the integrals can be performed for each element. Choosing simple polynomial trial and test functions, such as the linear basis functions discussed previously, ensures that these integrals are trivial to evaluate. Assembling these element integrals into a global equation system produces a linear system of ordinary differential equations in time. This system will be banded due to the localised definition of the basis functions and hence solution algorithms can take advantage of efficient banded matrix methods. The system can be integrated in time with an appropriate method, such as those that will be described in Section 3.3. The approximate solution at any time is then reconstructed by substituting the u_j into (3.25).

If ψ_i and ϕ_j are chosen from the same function space then the method is termed a Galerkin finite element method [113]. This will produce a centred nodal equation analogous to the centred finite difference schemes previously discussed. Upwinding concepts have been introduced in the finite element method by choosing different function spaces for ψ_i and ϕ_j, termed a Petrov-Galerkin finite element method [51]. As with the upwind

finite difference methods described in Section 3.2.2 this will introduce extra artificial diffusion and so these methods are not considered here.

The discretisation procedure and the particular function spaces used for the RLW and KdV equations are discussed in detail in the following sections.

3.2.3.2 Previous work

Alexander and Morris [4] presented dissipative and non-dissipative Galerkin finite element methods for the KdV and RLW equations using $\mathscr{H}^3$ cubic spline basis functions. The spatially discretised equations were integrated in time with adaptive time step software for ordinary differential equations. Results were presented for the RLW equation using the non-dissipative scheme, and for the KdV equation using varying amounts of dissipation to control small scale oscillations produced by the non-dissipative scheme. They concluded that it is much harder to solve the KdV equation than the RLW equation. Sanz-Serna and Christie [85] proposed a Petrov-Galerkin finite element method for the KdV equation using linear test functions and cubic spline trial functions, and a Crank-Nicolson time integration scheme. By analysing the truncation error of the spatially discretised nodal equations produced by this scheme they showed that high accuracy could be obtained and their results were comparable with those of Alexander and Morris [4] at greatly reduced computational cost. Schoombie analysed a $\mathscr{H}^2$ Hermite cubic Galerkin finite element method for the KdV equation [91] and also a Petrov-Galerkin finite element method for the KdV equation with linear test functions and variable order splines as trial functions [92, 93]. This work used a Crank-Nicolson time integration scheme. Dissipative methods were proposed for use when the solution contained discontinuous or near-discontinuous features. Gardner et al. have published work on a cubic spline Galerkin finite element method for the RLW equation [40] and a quadratic spline Galerkin finite element method for the KdV equation [41, 42]. They have used these spatial schemes together with Crank-Nicolson time integration to investigate solitary waves for both of these equations. Katapodes and Wu [55] used a linear Galerkin finite element method for the RLW equation. This was combined with a Lax-Wendroff type approximation in time to produce a fully explicit method. This was used to model solitary wave problems.

Although the methods based on $\mathscr{H}^2$ and $\mathscr{H}^3$ finite elements are useful in one-dimension it would be difficult to extend these methods to two-dimensional problems based on unstructured triangular meshes. The Lax-Wendroff approach has been successfully extended to one-dimensional [5, 56] and two-dimensional [6, 56] forms of the original Boussinesq equations, but the form of the discretisation is not obviously compatible with the adaptive time integration methods considered here. In this chapter the emphasis is on producing a finite element spatial discretisation method that can eventually be applied to the two-

dimensional extended Boussinesq equations.

3.2.3.3 Finite element methods for the RLW equation

The RLW equation (3.1) contains a third order mixed space-time derivative. However, with the method of lines approach the time derivative is treated as a nodal variable at this stage and this term is simply discretised as a second order spatial derivative.

To develop a finite element method a weighted residual statement of equation (3.4) is obtained. Denoting the weight function $\psi_i(x)$, and the spatial domain Ω, and using the notation,

$$(u,v) \;=\; \int_\Omega u\, v \; d\Omega \tag{3.33}$$

the following is obtained.

$$\begin{aligned} (\psi_i,\dot{\eta}) - \delta\left(\psi_i, \frac{\partial^2 \dot{\eta}}{\partial x^2}\right) &= -\left(\psi_i, \frac{\partial f}{\partial x}\right) \\ (\psi_i,\dot{\eta}) + \delta\left(\frac{d\psi_i}{dx}, \frac{\partial \dot{\eta}}{\partial x}\right) &= -\left(\psi_i, \frac{\partial f}{\partial x}\right) \end{aligned} \tag{3.34}$$

The second derivative has been integrated by parts to reduce continuity requirements on the finite element interpolant. At the boundary the values of η are set to zero so the boundary integral does not appear. The functions in equation (3.34) are approximated as follows.

$$\eta(x,t) \;\approx\; \phi_j(x)\eta_j(t) \tag{3.35}$$

$$f(\eta(x,t)) \;\approx\; \phi_j(x) f_j(t) \tag{3.36}$$

where $\psi_i(x)$ and $\phi_j(x)$ are taken from the appropriate sets of finite element basis functions, and the summation over the repeated index j is implied. Note that the nonlinear function $f(\eta)$ (3.4) is interpolated in the same way as the function η. This approximation is known to give a more accurate representation than simply interpolating η [85]. Inserting these approximations into equation (3.34),

$$\left((\psi_i,\phi_j) + \delta\left(\frac{d\psi_i}{dx}, \frac{d\phi_j}{dx}\right)\right)\dot{\eta}_j \;=\; -\left(\psi_i, \frac{d\phi_j}{dx}\right) f_j \tag{3.37}$$

Thus only first order spatial derivatives appear in the formulation and ψ_i and ϕ_j can be chosen from $\mathscr{H}^1$.

Sanz-Serna and Christie [85] analysed their finite element method for the KdV equation by examining the truncation error of the difference scheme produced at a node. This

process is applied here for a linear Galerkin finite element approximation to the expression (3.37). On a uniform mesh with element length Δ equation (3.37) has the following form at an internal node;

$$\frac{\Delta}{6}(\dot{\eta}_{i-1}+4\dot{\eta}_i+\dot{\eta}_{i+1})+\frac{1}{2}(f_{i+1}-f_{i-1})+\frac{\delta}{\Delta}(-\dot{\eta}_{i-1}+2\dot{\eta}_i-\dot{\eta}_{i+1}) \;=\; 0 \qquad (3.38)$$

Note that in contrast to the finite difference method (3.13) a mass matrix term multiplies the time derivative. In the finite difference method these terms are directly replaced by nodal values. Substituting standard Taylor series expansions in space into equation (3.38), and rearranging the terms,

$$\begin{aligned} &\left(\dot{\eta}_i+\frac{\partial f_i}{\partial x}-\delta\frac{\partial^2\dot{\eta}_i}{\partial x^2}\right)\\ &+\frac{\Delta^2}{6}\frac{\partial^2}{\partial x^2}\left(\dot{\eta}_i+\frac{\partial f_i}{\partial x}-\delta\frac{\partial^2\dot{\eta}_i}{\partial x^2}\right)\\ &+\delta\frac{\Delta^2}{12}\frac{\partial^4\dot{\eta}_i}{\partial x^4}\\ &+\Delta^4\left(\frac{1}{72}\frac{\partial^4\dot{\eta}_i}{\partial x^4}+\frac{1}{120}\frac{\partial^5 f_i}{\partial x^5}-\frac{\delta}{360}\frac{\partial^6\dot{\eta}_i}{\partial x^6}\right) \;=\; \mathcal{O}(\Delta^6) \end{aligned} \qquad (3.39)$$

Using the original equation (3.1) it can be seen that the truncation error of the scheme at a node has the form,

$$\delta\frac{\Delta^2}{12}\frac{\partial^4\dot{\eta}_i}{\partial x^4}+\Delta^4\left(\frac{1}{72}\frac{\partial^4\dot{\eta}_i}{\partial x^4}+\frac{1}{120}\frac{\partial^5 f_i}{\partial x^5}-\frac{\delta}{360}\frac{\partial^6\dot{\eta}_i}{\partial x^6}\right)+\mathcal{O}(\Delta^6) \qquad (3.40)$$

The leading term is $\mathcal{O}(\Delta^2)$ due to the linear approximation but multiplies a mixed fifth order derivative. This will not corrupt the physical dispersion present in the system, which depends on third and lower derivatives, and suggests that the linear finite element approximation is acceptable for this equation.

Comparing the finite difference scheme (3.13) and the nodal form of the finite element scheme (3.38) and its truncation error (3.40) it can be seen that truncation errors are of a similar size. Contrasting the two schemes it can be seen that, whereas the finite difference method achieved this accuracy by a fourth order approximation to the first derivative term, the finite element scheme's accuracy is due to the mass matrix multiplying the time derivative, and to the compact approximation of the nonlinear term f defined by (3.36). The beneficial effect of this so-called product approximation of f has been noted previously [28], and is crucial to the truncation error analysis. If the time derivative approximation were replaced by a nodal value as in the finite difference method the resulting finite element scheme would be identical to the fully second order finite difference

method, and hence inaccurate for this equation.

3.2.3.4 Finite element methods for the KdV equation

The KdV equation (3.2) contains a third order spatial derivative. The weighted residual statement is obtained by multipling equation (3.2) by a trial function $\psi_i(x)$ and integrating over the spatial domain Ω. Using the notation of equation (3.33),

$$\begin{aligned}(\psi_i, \dot{\eta}) &= -\left(\psi_i, \frac{\partial f}{\partial x}\right) + \delta\left(\psi_i, \frac{\partial^3 \eta}{\partial x^3}\right) \\ &= -\left(\psi_i, \frac{\partial f}{\partial x}\right) - \delta\left(\frac{d\psi_i}{dx}, \frac{\partial^2 \eta}{\partial x^2}\right)\end{aligned} \tag{3.41}$$

In this case the integration by parts can only reduce the derivatives to second order and hence a higher degree of continuity is required of the basis functions. Interpolation functions from $\mathscr{H}^2$ will be required, the simplest of which is the Hermite cubic element [100]. In this case both the variable and its first derivative are interpolated over the mesh resulting in a solution with continuous first derivative. This doubles the number of unknowns in the discrete system.

Interpolating the variables as in equations (3.35)-(3.36), with the understanding that u_j and f_j are now nodal values of the variable and its first derivative, the finite element equation system can be written in the form,

$$(\psi_i, \phi_j)\dot{\eta}_j = -\left(\psi_i, \frac{d\phi_j}{dx}\right) f_j - \delta\left(\frac{d\psi_i}{dx}, \frac{d^2\phi_j}{dx^2}\right) \eta_j \tag{3.42}$$

As mentioned in the previous section the $\mathscr{H}^2$ Hermite cubic Galerkin finite element method has been used successfully for this equation, as well as methods involving $\mathscr{H}^2$ quadratic and $\mathscr{H}^3$ cubic spline functions. However the aim here is to design a scheme that can be generalised to a two-dimensional finite element method based on unstructured triangular grids. Although some work has been done on extending Hermite-type interpolation onto unstructured grids [83] the additional complexity makes them impractical for calculations with large meshes.

An alternative to using a higher degree of continuity in the finite element basis functions is to rewrite the equation in a lower order form. A new variable, *w*, is introduced to replace either the first or second spatial derivative of η. This is similar in principle to the Keller Box finite difference scheme [76]. There are then no derivatives of order higher than two and an $\mathscr{H}^1$ finite element approximation is possible. After considering the form of the extended Boussinesq free surface equation (2.109) it was decided that the second derivative was the appropriate choice for the auxiliary variable. Rewriting the

KdV equation (3.2),

$$\dot{\eta} = -\frac{\partial f}{\partial x} - \frac{\partial w}{\partial x} \tag{3.43}$$

$$w = \delta \frac{\partial^2 \eta}{\partial x^2} \tag{3.44}$$

A finite element approximation of this equation system results in,

$$(\psi_i, \phi_j)\dot{\eta}_j = -\left(\psi_i, \frac{d\phi_j}{dx}\right) f_j - \left(\psi_i, \frac{d\phi_j}{dx}\right) w_j \tag{3.45}$$

$$(\psi_i, \phi_j) w_j = -\delta \left(\frac{d\psi_i}{dx}, \frac{d\phi_j}{dx}\right) \eta_j \tag{3.46}$$

The auxiliary variable w doubles the number of unknowns in the system, but the continuity restrictions on the test space have been reduced. In particular $\mathscr{H}^1$ functions such as the linear approximation introduced at the start of this section can now be used.

Some simplified schemes for this equation are also investigated to try and reduce the extra work required to solve the auxiliary algebraic system (3.46). In this way the overall computational cost can be reduced and hopefully made competitive with that of the finite difference schemes. These savings will be essential when two-dimensional calculations are considered in Chapter 5. Equation (3.46) can be approximated in a more explicit form. The matrix multiplying w_j, termed the mass matrix in finite element terminology [113], is approximated in diagonal form, or lumped [113]. This approximation replaces equation (3.46) with,

$$\sum_{j=1}^{N+1} (\psi_i, \phi_j) w_i = -\delta \left(\frac{d\psi_i}{dx}, \frac{d\phi_j}{dx}\right) \eta_j \tag{3.47}$$

Solving this coupled system within the time integration algorithm will still result in twice as many unknowns but the solution of the system (3.47) is now trivial. A further approximation is made by decoupling the two equations and solving equation (3.47) explicitly for w. This is a trivial procedure since the matrix multiplying w has been made diagonal. Equation (3.46) is then integrated in time. This reduces the number of degrees of freedom in the time integration system to the same as for the finite difference schemes. Lumping the mass matrix is known to decrease the accuracy of the approximation [113]; however the analysis that follows shows that the spatial truncation error is not significantly affected by this approximation. An alternative approach which regains the full accuracy of the whole mass matrix without matrix inversion would be to use the simple iterative method of Donea and Giuliani [34].

The truncation error of this simplified finite element scheme is now analysed in a

similar manner to the previous section. With a linear finite element discretisation on a regular mesh of element length Δ, equations (3.45) and (3.47) have the following form at an internal node;

$$\frac{\Delta}{6}(\dot{\eta}_{i-1}+4\dot{\eta}_i+\dot{\eta}_{i+1})+\frac{1}{2}(f_{i+1}-f_{i-1})+\frac{1}{2}(w_{i+1}-w_{i-1}) = 0 \qquad (3.48)$$

$$\Delta w_i+\frac{\delta}{\Delta}(-\eta_{i-1}+2\eta_i-\eta_{i+1}) = 0 \qquad (3.49)$$

Again, using Taylor series expansions in space and then differentiating equation (3.49) and using it to replace the w_i terms in equation (3.48),

$$\begin{aligned}
&\left(\dot{\eta}_i+\frac{\partial f_i}{\partial x}+\delta\frac{\partial^3\eta_i}{\partial x^3}\right)\\
&+\frac{\Delta^2}{6}\frac{\partial^2}{\partial x^2}\left(\dot{\eta}_i+\frac{\partial f_i}{\partial x}+\delta\frac{\partial^3\eta_i}{\partial x^3}\right)\\
&+\delta\frac{\Delta^2}{12}\frac{\partial^5\eta_i}{\partial x^5}\\
&+\Delta^4\left(\frac{1}{72}\frac{\partial^4\dot{\eta}_i}{\partial x^4}+\frac{1}{120}\frac{\partial^5 f_i}{\partial x^5}+\frac{\delta}{40}\frac{\partial^7\eta_i}{\partial x^7}\right) = \mathcal{O}(\Delta^6)
\end{aligned} \qquad (3.50)$$

Using the original equation (3.2) the truncation error at a node is,

$$\delta\frac{\Delta^2}{12}\frac{\partial^5\eta_i}{\partial x^5}+\Delta^4\left(\frac{1}{72}\frac{\partial^4\dot{\eta}_i}{\partial x^4}+\frac{1}{120}\frac{\partial^5 f_i}{\partial x^5}+\frac{\delta}{40}\frac{\partial^7\eta_i}{\partial x^7}\right)+\mathcal{O}(\Delta^6) \qquad (3.51)$$

The leading truncation error term indicates that the discretisation is $\mathcal{O}(\Delta^2)$ accurate but that the form of this leading error will not contaminate the physical dispersion. Importantly, the accuracy does not appear to have been significantly reduced by the diagonal approximation of the mass matrix in equation (3.47). Hence by decoupling the algebraic equation and solving it explicitly this finite element scheme can be solved as efficiently as the finite difference scheme and produce results of the same order of accuracy. In Chapter 5 this explicit approximation of the algebraic equations will be used for the two-dimensional equations to reduce a sparse matrix system to diagonal form, resulting in a much larger increase in efficiency than in this one-dimensional case.

3.3 Time integration

The spatial discretisations described in the previous section result in a linear system of differential equations in the form,

$$A_{ij}\dot{y}_j-f_i = 0 \qquad (3.52)$$

where in this case A_{ij} is a constant coefficient matrix, and $f(t,y)$ is a nonlinear function of $y(t)$. The vector $y(t)$ contains all the system unknowns. The localised nature of the one-dimensional finite difference and finite element spatial approximations will mean that A_{ij} is a banded matrix, which can be solved much more efficiently than a full matrix. Together with the appropriate initial and boundary conditions this is generally classed as a differential-algebraic equation system [23]. Assuming that some of the boundary conditions will be applied in algebraic, Dirichlet, form this system (3.52) can be classified as an index one differential-algebraic system [23]. There are various software packages available for solving such problems.

The SPRINT package [19, 20] contains a selection of adaptive order, adaptive step size methods for integrating differential-algebraic systems. It has been designed for the solution of partial differential equations in method of lines form and has efficient sparse iterative solution strategies for large two-dimensional simulations [105]. Direct matrix solution methods are also available, including banded matrix methods which can be used for the one-dimensional cases in this chapter. Two of the time integration methods are considered here; spgear, a fixed coefficient backward difference formulae (BDF) method, and spdasl, a fixed leading coefficient BDF method, which is an implementation of the time integration method DASSL [23]. The fixed coefficient BDF methods are the most computationally efficient for smooth problems, but have to interpolate solution values in cases where the time step changes. They can become inefficient and even unstable for some problems [23, 49]. The fixed leading coefficient BDF methods are genuinely variable step methods but may be less efficient due to the additional work required to account for the changing step sizes. However Jackson and Sacks-Davis [49] point out that this additional work is proportional to the order of time integration used whereas the interpolation used by the fixed coefficient BDF methods to account for a variable step size is proportional to the number of equations in the system. Fully variable coefficient BDF methods are also possible but are generally not efficient compared to the fixed leading coefficient methods [23]. DASPK [25] implements the adaptive order, adaptive step size fixed leading coefficient BDF method DASSL [23]. This method has been combined with the SPARSKIT package [84] for the iterative solution of large sparse equation systems. Banded matrix solution methods are also available. RADAU5 [46] is an order five implicit Runge-Kutta time integration method with adaptive time step control based on user supplied error tolerances. An advantage of this approach over the BDF methods is that, being a single step method, it can integrate over the whole time interval with fifth order accuracy. The BDF methods, which use increasing numbers of time levels for higher order accuracy, must begin time integration at first order accuracy and hence with a relatively small time step to satisfy the user supplied error tolerances. However a drawback with this package is that it does not have any sparse matrix handling implemented which

will limit its usefulness for two-dimensional calculations. For this reason attention has been focussed on the packages SPRINT and DASPK.

The approximation to the time derivative $\dot{y}$ is determined by the time integration method used and to the order of that method, but has the general form,

$$\dot{y}_k = \frac{y_k - z_k}{\alpha \, \Delta t} \tag{3.53}$$

where the exact form of z_k is determined by the time integration algorithm that is being used, and depends on information from previous time steps [19]. In particular z does not depend on the current solution y. The constant α depends on the time integrator used and Δt is the current time step.

The equation system is supplied to the time integration algorithm in the form of a residual vector r [19],

$$r_i = f_i - A_{ik}\dot{y}_k \tag{3.54}$$

The solution is advanced in time by using the approximation to the time derivative (3.53) in the residual (3.54) and then requiring that the residual at the next time level is sufficiently small [23]. The resulting system of nonlinear equations for y is solved by a Newton method [19]. This is the most computationally expensive part of the time integration process and requires the solution of a linear system of equations at each Newton iteration.

$$J_{ij}(y)\Delta y_j = -\alpha \, \Delta t \, r_i(y) \tag{3.55}$$

where Δy is the solution increment at that time step, and J_{ij} is the Jacobian matrix given by,

$$J_{ij} = \alpha \, \Delta t \, \frac{dr_i}{dy_j} \tag{3.56}$$

Substituting the form of the residual (3.54) into the Jacobian expression (3.56), recalling that here A_{ij} has constant coefficients,

$$J_{ij} = \alpha \, \Delta t \left(\frac{df_i}{dy_j} - A_{ik} \frac{d\dot{y}_k}{dy_j} \right) \tag{3.57}$$

The approximation to the time derivative (3.53) gives,

$$\frac{d\dot{y}_k}{dy_j} = \frac{1}{\alpha \, \Delta t} \frac{dy_k}{dy_j} \tag{3.58}$$

$$= \frac{\delta_{kj}}{\alpha\,\Delta t} \tag{3.59}$$

and substituting in the Jacobian expression (3.57) gives,

$$J_{ij} \;=\; \alpha\,\Delta t\frac{d\,f_i}{dy_j} - A_{ij} \tag{3.60}$$

It is possible for the user of the time integration package to provide the exact form of the Jacobian, in which case the form (3.60) is appropriate here. However it is more common to allow the package to numerically approximate the Jacobian using finite differences [20]. These two possibilities are discussed in more detail in the following chapters. Advantage can be taken here of the local support of a nodal equation. This will produce a banded matrix system which can be solved much more efficiently than a full matrix problem [19]. An exact Newton method would require that J_{ij} is recomputed after each Newton iteration but this would require a matrix factorisation step at each iteration which would be too expensive in general. For this reason the Jacobian is kept constant for as long as possible and only recomputed when the error produced by its approximation exceeds a given tolerance [20]. This produces a practical method without an excessive computational expense. Note that for the problems considered here the matrix A_{ij} has constant coefficients, hence for small time steps Δt the Jacobian will remain almost constant. It is found in the computations in Section 3.4 that the Jacobian is only recomputed during the initial stages when the order of the time integration is being raised. Once the order of time integration remains steady the Jacobian is not recomputed again despite changes in the solution values and small changes in the time step.

The choices of time step and order of time integration used are driven by error control methods which attempt to satisfy a user-defined level of accuracy in time. The basic strategy is outlined below.

The exact solution of the partial differential equation is denoted $u(x,t)$. Discretising in space produces the differential-algebraic equation system,

$$F(\dot{U},U,t) \;=\; 0 \tag{3.61}$$

which is a more general form of the system (3.52) stated previously. This is assumed to have an exact solution $U(t)$. At time t^n the numerical solution of the system (3.61) is $V(t^n)$, hence the total error, $E(t)$ at that point is [16],

$$\begin{aligned} E(t^n) &= u(x,t^n) - V(t^n) \\ &= (u(x,t^n) - U(t^n)) + (U(t^n) - V(t^n)) \\ &= e(t) + g(t) \end{aligned} \tag{3.62}$$

$e(t) = u - U$ represents the spatial discretisation error, and $g(t) = U - V$ represents the global error in the time integration.

The local error in the time integration $le(t)$ is defined over one time step $[t^n, t^{n+1}]$ of length $\Delta t = t^{n+1} - t^n$. The differential-algebraic problem,

$$F(\dot{y}^{n+1}, y^{n+1}, t^{n+1}) \;=\; 0 \tag{3.63}$$

with initial condition $y^{n+1}(t^n) = V(t^n)$ is integrated numerically producing a solution $y^{n+1}(t^{n+1})$ at t^{n+1}. The local error is then defined as,

$$le^{n+1} \;=\; V(t^{n+1}) - y^{n+1}(t^{n+1}) \tag{3.64}$$

Controlling the error in this form is known as local error per step control [16]. An alternative is to control the local error per unit step,

$$\frac{le^{n+1}}{\Delta t} \tag{3.65}$$

In this form it can be shown that if the local error per unit step is controlled with a certain tolerance then the global time error is controlled by an amount directly proportional to that tolerance [99].

The choice of time step and order of integration is determined by estimating the local error in y, le, and requiring,

$$\|e\| \;<\; 1 \tag{3.66}$$

where,

$$e_i \;=\; \frac{le_i}{w_i} \tag{3.67}$$

le_i is the ith component of the local error estimate and,

$$w_i \;=\; rtol\,|y_i| + atol \tag{3.68}$$

is a weight factor. The choice of norm $\|.\|$ and the values of $rtol$ and $atol$, which can be vector quantities in general, are supplied by the user. Here the Euclidean norm is used and $rtol$ and $atol$ are scalar values determined by experiment; lowering their values until the solution is independent of them will ensure that the total error in the solution (3.62) is dominated by the spatial discretisation error.

The initial-value problems considered here require a consistent set of discrete initial

conditions $\{y(x,t_0),\dot{y}(x,t_0)\}$ to start the time integration. In order to satisfy the irrotationality condition initially, as specified in Section 2.2, it is common to begin with an undisturbed free surface [80] which trivially satisfies this problem. However the solitary wave cases considered in Section 3.4 will have a non-zero initial free surface distribution. If the solution vector $y(x,t_0)$ is specified, such as in the solitary wave problems considered in this section, then SPRINT and DASPK will calculate consistent initial time derivatives such that the system (3.52) is satisfied [18, 26].

3.4 Numerical examples

3.4.1 Introduction

The numerical examples considered in the following sections use the solitary wave solutions (2.167) and (2.169). The depth $H = 2$, and the solitary wave shape parameter c is chosen according to equation (2.172) as,

$$c \;=\; \sqrt{gH} + 0.1\frac{H}{2} \tag{3.69}$$

which ensures a well behaved solution within the Boussinesq framework. The spatial domain $x \in [-100, 900]$ is large enough so that the solution and its derivatives can be assumed zero at the boundaries. The problem is integrated in time for $t \in [0, 120]$. Hence the initial conditions are of a solitary wave of amplitude 0.2, centred on the origin.

Time integration was performed with the SPRINT modules spgear and spdasl [19]. Initial tests were performed to ensure the correct convergence of the methods with decreasing spatial mesh size, and that the temporal error control tolerances could be reduced until independence of the solution was obtained.

All the examples in the following sections were computed with a nodal spacing of 1, ie. 1001 nodes over the domain, and time integration error tolerances *rtol* and *atol* both equal to 10^{-6}. The results are plotted for the wave at $t = 120$ and only the portion $[350, 650]$ of the spatial domain is shown to make any changes in shape of the wave more apparent. The wave is effectively zero elsewhere in the domain.

In all cases considered in this chapter the adaptive time integration method rapidly increased the order of integration until the fifth order BDF method was selected, reflecting the high order of accuracy required for the computation of these problems. The time steps quoted in the following sections are the constant values that the package selected, once the fifth order BDF method was being used.

3.4.2 The RLW equation

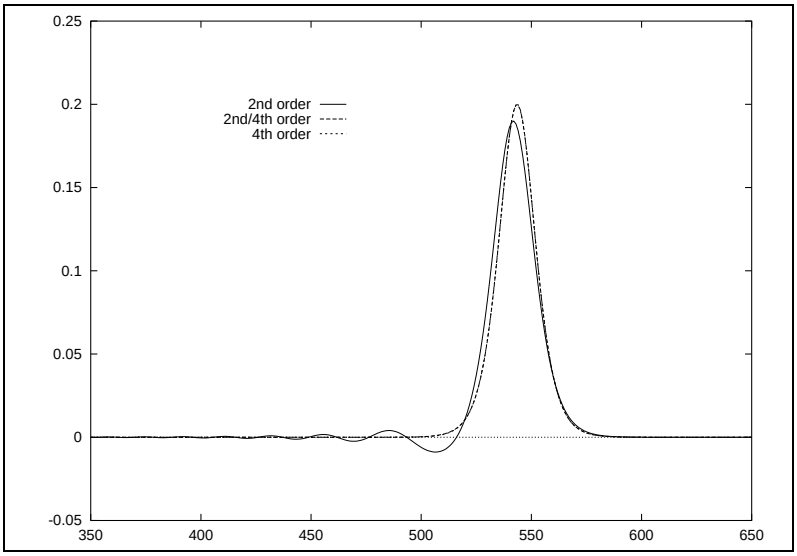
Figure 3.2(a) shows the three finite difference schemes described in Section 3.2.2.3 applied to the propagation of the single solitary wave. The second order scheme has clearly dissipated the wave and an oscillatory tail is present. Examining the nodal amplitudes of the wave reveals that the solution is 10% in error in the worst case. The time integration uses a uniform time step of 0.135 after the initial stages where the order of the BDF method is increased to five. The fourth order scheme appears accurate and a closer examination of the nodal amplitudes reveals a maximum error of 0.1% at a node. The time integration settles to a similar uniform time step of 0.132. The mixed order scheme (3.13) produces results almost coincident with the fourth order scheme with a maximum error also of 0.1%, which reinforces the assumption that the error was solely due to the poor approximation of the first spatial derivative. The time integration again uses a similar uniform time step of 0.134.

Figure 3.2(b) shows two of the finite element schemes described in Section 3.2.3.3 applied to the same problem. Both the linear and the quadratic Galerkin finite element scheme transport the solitary wave solution with no apparent dissipation of the solution. Closer examination reveals the maximum nodal error to be 0.3% and 0.03% for the linear and quadratic schemes respectively. The time integration algorithm settles to time steps of 0.13 for both cases. The accuracy of the linear scheme is comparable with the mixed and fourth order finite difference schemes despite being only formally second order accurate [113]. The apparent accuracy of this scheme confirms the analysis of Section 3.2.3.3 and is discussed in more detail in Section 3.5.

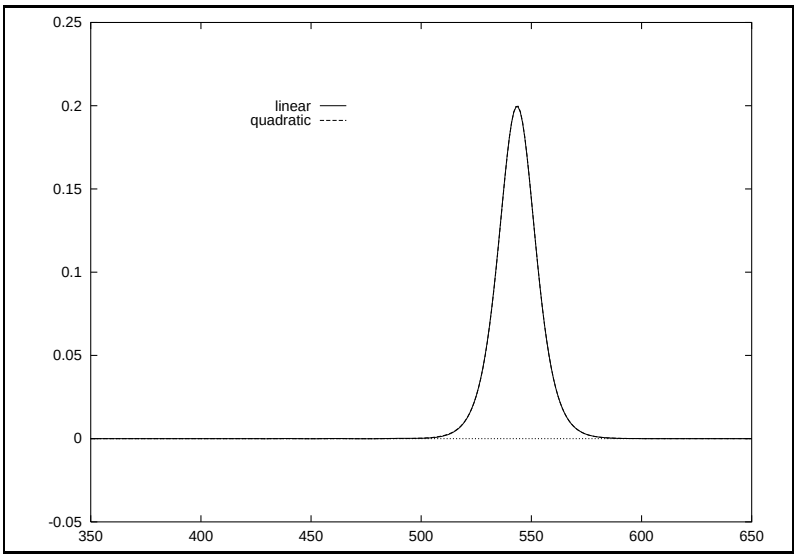
Further tests of the RLW equation models examined the interaction of two solitary waves [37] and the break up of an initial Gaussian profile [41]. The results exhibited the same accuracy characteristics as for the single solitary wave cases.

3.4.3 The KdV equation

Figure 3.3(a) shows the three finite difference schemes described in Section 3.2.2.4 applied to the propagation of the single solitary wave. The second order scheme has clearly dissipated the wave and an oscillatory tail is present. Examining the nodal amplitudes shows a maximum error of 10%. The time integration uses a step size of 0.134. The fourth order scheme appears accurate and a closer examination of the nodal amplitudes reveals a maximum error of 0.1% at a node. In this case the time integration settles to a reduced time step of 0.08, which is probably due to the fourth order accurate stencil of the third spatial derivative. The mixed order scheme (3.14) produces results that appear coincident with the fourth order scheme. Closer examination reveals a maximum nodal error of 0.5%, which is a larger difference than for the RLW equation but still comparable

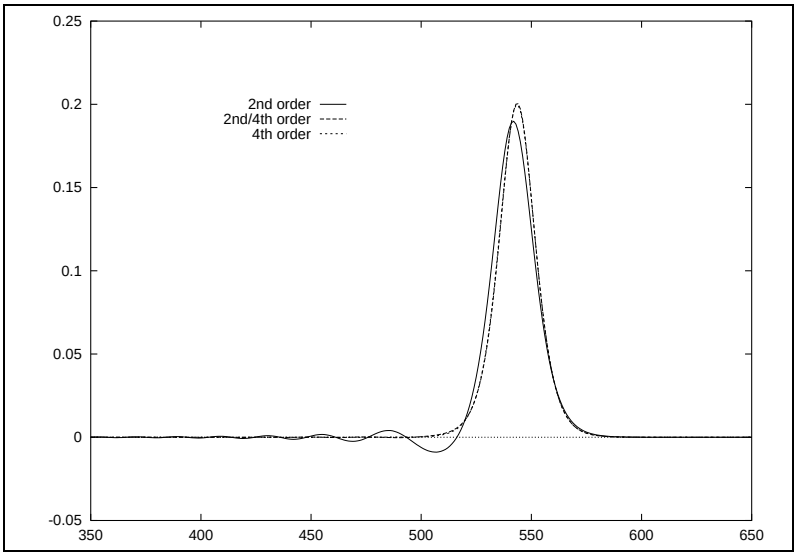


(a) Finite difference schemes

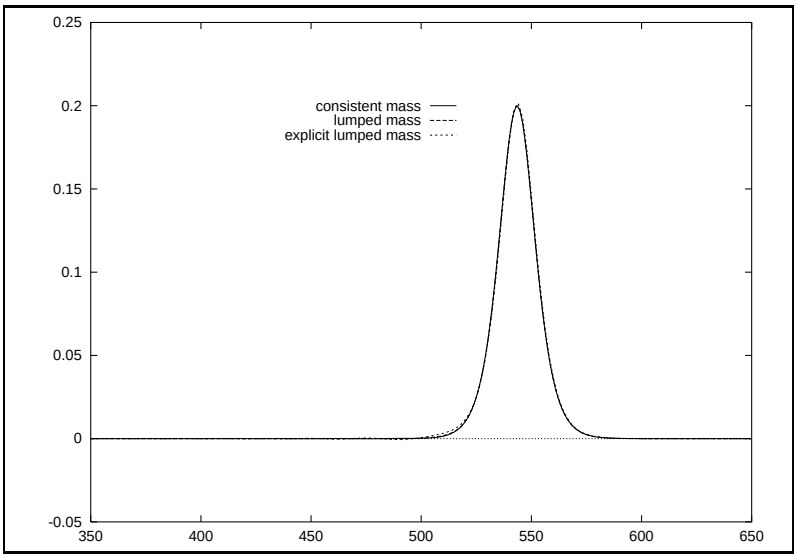


(b) Finite element schemes

Figure 3.2: Numerical methods for the RLW equation solitary wave



(a) Finite difference schemes



(b) Finite element schemes

Figure 3.3: Numerical methods for the KdV equation solitary wave

with the fully fourth order scheme. The time integration algorithm is able to increase the time step to 0.13 and the stencil of the scheme is reduced in this case which are further advantages over the fully fourth order scheme.

Of the finite element schemes described in Section 3.2.3.4 only the methods for the equations rewritten in lower order form are likely to be practical in two dimensions. This is at the expense of doubling the number of unknowns in the system. Figure 3.3(b) shows a linear Galerkin finite element scheme with alternative solution strategies for the auxiliary w equation (3.46). The graph labelled *consistent mass* solves the coupled system (3.45),(3.46) within the time integration algorithm. In this case the time integration module spgear failed to complete the time integration of the problem. This is a fixed coefficient BDF method and it is possible that problems associated with these methods, discussed in Section 3.3, are responsible for the failure, although a precise reason is not known at present. The fixed leading coefficient time integration module spdasl was substituted and found to work for this problem with a maximum nodal error of 0.25%, although at a much reduced time step of 0.02. In this case the time integration system is twice the size of the corresponding finite difference algorithm and the solution of the system takes many more time steps making the finite element method much less efficient. For the graph labelled *lumped mass* the equation (3.46) is approximated in the explicit lumped form (3.47). The coupled system (3.45),(3.47) is then solved within the time integration algorithm producing a maximum nodal error of 0.2% at the end of the time integration. The time step is increased to 0.07 but the system still has double the number of unknowns of the finite difference schemes. For the graph labelled *explicit lumped mass* the equations (3.45) and (3.47) are decoupled. The algebraic equation (3.47) is then solved explicitly for w prior to the assembly of equation (3.45) at each time step. This is a simple procedure as the matrix multiplying w has been made diagonal. Equation (3.46) is then integrated in time. This reduces the number of degrees of freedom in the time integration system to the same as for the finite difference schemes. The numerical results show a very slight oscillatory tail and the maximum error at a node is 2% but the results are still comparable with the high order finite difference schemes confirming the analysis of Section 3.2.3.4. The time step used is increased to 0.13 which is comparable to that used by the finite difference schemes.

Further tests of the KdV equation models examined the interaction of two solitary waves and the break up of an initial Gaussian profile. Again the results exhibited the same accuracy characteristics as for the single solitary wave cases.

3.5 Discussion

In this chapter two new numerical methods for the RLW and KdV equations have been presented. The mixed order finite difference method is based on the method of Wei And Kirby [106] which has been used successfully for the extended Boussinesq equations. The truncation errors of the scheme are in the form of derivatives of order greater than three, and in the non-dimensionalised, scaled Boussinesq framework introduced in Chapter 2 these terms will be small enough to not affect the model. The results for the RLW and KdV equations show that this finite difference method is equally applicable here. The proposed finite element method is based on linear elements. This is much simpler than methods that have been previously used, which were outlined in Section 3.2.3.2. However the goal here is to produce a method that can ultimately be implemented in two dimensions, so the emphasis here has been on maintaining simplicity rather than focusing solely on the accuracy of the scheme. A truncation error analysis of the linear finite element schemes showed a similar behaviour to the mixed order finite difference scheme, and the numerical results reinforced these findings.

It has been demonstrated that these spatial discretisations can be coupled with the adaptive time integration package SPRINT [19]. It was found that it was relatively easy to solve the RLW equation. The finite difference and finite element schemes could compute solutions of similar accuracy with a similar number of time steps for the same error control tolerances. For the KdV equation a similar accuracy between the finite element and finite difference schemes could be obtained if the finite element system was solved in a fully coupled form. This was at the expense of solving a finite element equation system of twice the size compared to the finite difference scheme and with a greater number of time steps. It was also found that the fixed coefficient BDF method spgear was not suitable to solve the finite element system in all cases. The fixed leading coefficient BDF method spdasl was substituted and found to work well for all the problems considered here. Decoupling the finite element system and solving the algebraic equation explicitly reduced the time integration system to the same size as for the finite difference method and allowed a similar time step to be used. This was at the expense of a slight loss in accuracy, although the results were still acceptable.

The greater difficulty in solving the KdV equation has been noted previously by Alexander and Morris [4]. They ascribe this difficulty to the third order spatial derivatives present in the KdV equation compared to second order spatial derivatives in the RLW equation. Benjamin et al. [13] propose that the RLW equation is a preferable model for water waves due to the form of its dispersion relation compared to the KdV equation. Following the procedure described at the start of Section 2.3.3 and assuming a steady

periodic wave form (2.65) the linearised RLW equation dispersion relation is,

$$\frac{\omega}{k} = \frac{\sqrt{gH}}{1+\frac{(kH)^2}{6}} \tag{3.70}$$

and for the linearised KdV equation,

$$\frac{\omega}{k} = \sqrt{gH}\left(1-\frac{(kH)^2}{3}\right) \tag{3.71}$$

For large k the KdV equation phase velocity, defined by equation (2.43), is not bounded. The same is also true of the group velocity, defined by equation (2.45). This implies that small wavelength (large k) features, for example numerical truncation errors, will not be damped away by the model, and may explain the larger errors in the solution of this equation. The RLW equation phase velocity and group velocity are bounded for all k and tend to zero for large k. It is thus a better candidate for numerical modelling as short wavelength numerical errors will not be propagated [13].

In the next chapter it will be shown that the numerical methods developed here for the model nonlinear dispersive equations can be applied to the one-dimensional extended Boussinesq equations. Wei and Kirby's finite difference method and the linear finite element scheme will both be used for the spatial discretisation with the SPRINT module spdasl used to integrate in time. Examination of Nwogu's extended Boussinesq equation system shows that its dispersion relation (2.125) and group velocity (2.129) are both bounded and tend to zero for large k. Hence it may be expected that the numerical methods will be well behaved with respect to any short wavelength numerical errors present in the model.

Chapter 4

An Algorithm for the One-dimensional Extended Boussinesq Equations

4.1 Introduction

The aim of this chapter is to investigate the implementation of finite element spatial discretisation methods and the adaptive time integration software for the one-dimensional extended Boussinesq equations. The published work on the numerical solution of the one-dimensional Boussinesq equations is reviewed in Section 4.2 and it is noted that at the start of this project, to the authors knowledge, there had been no other work published on finite element methods applied to the extended Boussinesq equations. More recently a finite element method for Beji and Nadaoka's extended Boussinesq equations, described in Section 2.4.2, has been reported [62], although this was after the publication of the one-dimensional algorithm developed here [104].

The mathematical model for shallow water flow chosen here; Nwogu's extended Boussinesq equation system, is reviewed in Section 4.3. In the previous chapter a linear Galerkin finite element spatial discretisation was presented that could, in combination with the time integration package SPRINT, be applied directly to the RLW equation and also to a modified form of the KdV equation. In this section it will be shown that Nwogu's extended Boussinesq equations can be rewritten in a form structurally similar to coupled RLW and KdV equations and hence that by using the novel approach adopted for the KdV equation in the previous chapter the linear finite element method can be applied to them. The spatial approximation of the extended Boussinesq equations is considered in

Section 4.4. A previously published finite difference method [106] is reviewed and an alternative linear finite element method is proposed. Appropriate boundary conditions for the equations and their numerical formulations are presented in Section 4.5. Based on the novel form of the differential equations adopted in this work new formulations of the inflow, outflow and wall boundary conditions are derived. In Section 4.5.4 a novel method of generating waves internally to the domain is presented which alleviates problems encountered due to reflected waves approaching the inflow boundaries. The spatial discretisation methods are coupled with the adaptive order, adaptive time stepping software package SPRINT. Issues concerned with the time integration and the initialisation of the problem are discussed in Section 4.6. The spatial accuracy of the finite element scheme is analysed in Section 4.8 by examining the truncation error at a node in a similar manner to that done for the RLW and KdV equations in the previous chapter. It is shown that the linear finite element method should provide sufficient spatial accuracy for these equations and hence that if the error in the time integration can be controlled to be less than the spatial error then accurate solutions will be obtained.

Numerical experiments are presented in Section 4.9 comparing the proposed finite element method with theoretical results, experimental results and the finite difference method. These results are used to validate the mathematical model as well as the numerical model and the boundary conditions, including the new method of internal wave generation. In Section 4.10 the results are reviewed and the issues involved in extending this method to two dimensions are considered.

4.2 Previous work

The first numerical schemes for the Boussinesq equations were based on finite difference methods. Peregrine [78] used second order finite differences and a corrector step for the free surface equation and modelled the development of an undular bore. Abbot et al. [1] applied a finite difference scheme on a staggered spatial grid. Careful analysis of this scheme allowed them to identify and remove sources of non-physical diffusion and dispersion. This scheme was used to model the propagation of periodic waves [2]. Hauguel [47] presented a three stage finite difference scheme coupled with a characteristic type approach for the convection terms. This was used to study solitary wave propagation. Schaper and Zielke [90] applied a finite difference scheme to the equations and used the low order shallow water equations to rewrite all the dispersive truncation errors in a similar form. This term was then used to correct the discretisation. Examples are given for solitary wave and periodic wave propagation. Karambas et al. [53] used a finite difference scheme with corrections for the spatial truncation errors to solve solitary wave and periodic wave propagation problems.

More recently there has been work presented on finite difference methods applied to extended Boussinesq equation systems. Madsen et al. [66] applied the finite difference scheme of Abbot et al. [1] to their extended equation system and showed that waves could be simulated at depths outside the range of the original Boussinesq equations. Beji and Battjes [10] used a predictor-corrector type scheme for these equations to model wave propagation over a bar. Nwogu [72] presented a predictor-corrector scheme for his extended equations with corrections for truncation error terms that would produce non-physical dispersion. He showed that these equations could be used to accurately simulate periodic and irregular wave trains. Wei and Kirby [106] presented an alternative finite difference method for Nwogu's equations that directly removed the dispersive errors by differencing selected terms to higher order and integrating in time with a high order predictor corrector method. This method will be described in more detail in Section 4.4.1. The results presented by Wei and Kirby [106] and other work with the same scheme [112] confirm that this finite difference method is both accurate and efficient. Ohyama et al. [74] presented a staggered grid finite difference method for Nwogu's extended Boussinesq model and applied it to wave propagation over a bar. Schröter et al. [95] applied a finite difference scheme similar in principle to that of Abbot et al. [2] to a different set of extended Boussinesq equations and showed that these equations could model problems outside the range of applicability of the original Boussinesq equations.

Dingemans [32] compares Boussinesq-type models for wave propagation over a bar, including finite difference models for the original and extended equations. The results show the importance of accurately modeling the dispersive effects, but also that the weak nonlinearity inherent in all the models can lead to significant inaccuracies. This bar test case will be considered in more detail in Section 4.9.

Until recently, work with finite element methods appears to have been restricted to the original Boussinesq equations. Katapodes et al. [56] applied a Lax-Wendroff type time discretisation to the equations combined with a linear Galerkin finite element method and used the scheme to solve solitary wave problems. Ambrosi [5] applied a linear modified Taylor-Galerkin finite element method to the equations. By ensuring that the scheme was non-dispersive accurate results were obtained for solitary wave problems. Grasselli et al. [44] and Perotto [82] have developed an a posteriori error estimate for the space-time finite element method applied to the original Boussinesq equations. They have used this to develop an adaptive finite element method which has been used to solve solitary wave problems.

It is only recently that work has been published on finite element methods for extended Boussinesq equations; by the author [104] for Nwogu's equation system, and by Li et al. [62] for Beji and Nadaoka's system. The methods described in the latter paper are compared and contrasted with those developed here.

4.3 The one-dimensional extended Boussinesq equation system

Nwogu's extended Boussinesq equation system (2.108)-(2.109) is written below in terms of free surface elevation $\eta(x,t)$, velocity $u(x,t)$ at depth $z = \theta h$, and spatially varying depth profile $h(x)$.

$$\dot{\eta} + \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left(B_1 h^3 \frac{\partial^2 u}{\partial x^2} + B_2 h^2 \frac{\partial^2}{\partial x^2}(hu) \right) = 0 \tag{4.1}$$

$$\dot{u} + \frac{\partial f}{\partial x} + A_1 h^2 \frac{\partial^2 \dot{u}}{\partial x^2} + A_2 h \frac{\partial^2}{\partial x^2}(h\dot{u}) = 0 \tag{4.2}$$

where the dot notation introduced in equation (3.3) is used for the time derivative,

$$p = (h+\eta)\,u \tag{4.3}$$

$$f = \frac{1}{2}u^2 + g\eta \tag{4.4}$$

and the coefficients A_1, A_2, B_1 and B_2, defined by equations (2.110)-(2.113), depend on the free parameter θ.

With a constant depth H the dispersive terms can be simplified and the equations reduce to,

$$\dot{\eta} + \frac{\partial p}{\partial x} + \beta H^3 \frac{\partial^3 u}{\partial x^3} = 0 \tag{4.5}$$

$$\dot{u} + \frac{\partial f}{\partial x} + \alpha H^2 \frac{\partial^2 \dot{u}}{\partial x^2} = 0 \tag{4.6}$$

where the coefficients α and β are defined by equations (2.116) and (2.117).

Note that in this form the free surface equation (4.5) is in a similar form to the KdV equation (3.2) and the velocity equation (4.6) is in a similar form to the RLW equation (3.1). Hence the numerical methods developed in Chapter 3 can be applied to this system. The spatial approximation of this coupled equation system, and the additional approximations necessary in extending these methods to the more general dispersive terms of the system (4.1)-(4.2) are discussed in the following section.

The full specification of this initial-boundary value problem requires appropriate boundary conditions and initial conditions. These are discussed in detail in Sections 4.5 and 4.6 respectively.

4.4 Spatial discretisation

4.4.1 A finite difference method

The finite difference method of Wei and Kirby [106], previously discussed in Section 3.2.2 for the RLW and KdV equations, was originally designed for Nwogu's extended Boussinesq equations. They showed that if the first spatial derivative is approximated to fourth order accuracy, with the other terms approximated to second order accuracy, then no non-physical dispersion is produced by the numerical scheme. For the analysis of this scheme, and for comparison with the finite element scheme described and analysed in the following sections, it is helpful to state the form of this discretisation with the leading truncation error terms. Using the centred difference stencils (3.8), (3.9) and (3.11) leads to the following difference scheme for the system (4.5)-(4.6) at a node i on a regular grid of size Δ,

$$\begin{aligned}\dot{\eta}_i + \frac{1}{12\Delta}\left(p_{i-2} - 8p_{i-1} + 8p_{i+1} - p_{i+2}\right) \\ + \frac{\beta H^3}{2\Delta^3}\left(-u_{i-2} + 2u_{i-1} - 2u_{i+1} + u_{i+2}\right) \\ = \frac{7\beta H^3}{60}\Delta^2\frac{\partial^5 u}{\partial x^5}(x_i) + \frac{1}{90}\Delta^4\frac{\partial^5 p}{\partial x^5}(x_i) \qquad (4.7)\end{aligned}$$

$$\begin{aligned}\dot{u}_i + \frac{1}{12\Delta}\left(f_{i-2} - 8f_{i-1} + 8f_{i+1} - f_{i+2}\right) + \frac{\alpha H^2}{\Delta^2}\left(\dot{u}_{i-1} - 2\dot{u}_i + \dot{u}_{i+1}\right) \\ = \frac{\alpha H^2}{12}\Delta^2\frac{\partial^4 \dot{u}}{\partial x^4}(x_i) + \frac{1}{90}\Delta^4\frac{\partial^5 f}{\partial x^5}(x_i) \qquad (4.8)\end{aligned}$$

The spatial truncation error terms are expressed in terms of fourth and higher derivatives and will not contaminate the real dispersive terms which are at most third order derivatives in space. In the scaled non-dimensionalised Boussinesq framework these higher derivatives can be neglected as being higher order terms in ε and σ without affecting the model. However this scheme remains formally second order accurate in space.

The variable depth equation system (4.1)-(4.2) differs from the constant depth system (4.5)-(4.6) in the more general form of the dispersive terms. A typical term of the velocity equation (4.2) is differenced at a node i using the second order accurate stencil (3.9) as follows,

$$A_2\, h\frac{\partial^2}{\partial x^2}(h\dot{u}) \;\approx\; \frac{A_2}{\Delta^2}\, h_i\, \left(h_{i-1}\dot{u}_{i-1} - 2h_i\dot{u}_i + h_{i+1}\dot{u}_{i+1}\right) \qquad (4.9)$$

The dispersive terms from the free surface equation (4.1) are differenced by first defining

w as follows,

$$w \;=\; B_1\, h^3 \frac{\partial^2 u}{\partial x^2} + B_2\, h^2 \frac{\partial^2}{\partial x^2}\,(hu) \tag{4.10}$$

The individual terms of w are differenced in a similar fashion to the terms of the velocity equation just discussed, eg. equation (4.9), and then the first derivative of w is approximated by the second order stencil (3.9) to produce the full dispersive term. This produces a second order accurate approximation to the variable depth dispersive terms, which reduces to the scheme (4.7)-(4.8) if the depth is constant.

4.4.2 A finite element method

The linear finite element scheme described in Chapter 3 cannot be applied directly to the extended Boussinesq equation system (4.1)-(4.2) due to the third spatial derivatives present in equation (4.1). This equation is rewritten by introducing w, defined by equation (4.10), as an auxiliary variable producing the system,

$$\dot{\eta} + \frac{\partial p}{\partial x} + \frac{\partial w}{\partial x} \;=\; 0 \tag{4.11}$$

$$\dot{u} + \frac{\partial f}{\partial x} + A_1 h^2 \frac{\partial^2 \dot{u}}{\partial x^2} + A_2 h \frac{\partial^2}{\partial x^2}\,(h\dot{u}) \;=\; 0 \tag{4.12}$$

$$w - B_1\, h^3 \frac{\partial^2 u}{\partial x^2} - B_2\, h^2 \frac{\partial^2}{\partial x^2}\,(hu) \;=\; 0 \tag{4.13}$$

The form of the auxiliary variable w is suggested naturally by the form of the differential equation (4.1) in Nwogu's extended Boussinesq system. This motivated the choice in Section 2.6 of Nwogu's extended equations over those of Beji and Nadaoka [11] for which an equivalent replacement is not so obvious. It also motivated the choice in Section 3.2.3.4 of the second derivative as the auxiliary variable when developing a finite element method for the KdV equation. It will be seen in Chapter 5 that this methodology can be generalised easily to Nwogu's two-dimensional extended Boussinesq equation system.

To simplify what follows, the constant depth form of equations 4.11-(4.13) will be used. The finite element method described will apply equally well to the more general system. The discretisation of a typical variable depth dispersive term is considered at the end of this section.

$$\dot{\eta} + \frac{\partial p}{\partial x} + \frac{\partial w}{\partial x} \;=\; 0 \tag{4.14}$$

$$\dot{u} + \frac{\partial f}{\partial x} + \alpha H^2 \frac{\partial^2 \dot{u}}{\partial x^2} \;=\; 0 \tag{4.15}$$

$$w - \beta H^3 \frac{\partial^2 u}{\partial x^2} = 0 \tag{4.16}$$

where the coefficients α and β are defined by equations (2.116) and (2.117).

The finite element method is applied, as outlined in Section 3.2.3 in the previous chapter. A set of $N+1$ nodes is defined as the intersection points of N non-overlapping elements that completely cover the spatial domain $\Omega = [X_1, X_2]$. A set of $N+1$ H^1 nodal linear basis functions (3.15) is defined on this mesh. The linear Galerkin finite element method is developed by multiplying equations (4.14)-(4.16) by a trial function $\phi_i(x)$ taken from the set of linear basis functions and integrating over the spatial domain Ω. Using the previously introduced notation (3.33),

$$(\phi_i, \dot{\eta}) + \left(\phi_i, \frac{\partial p}{\partial x}\right) + \left(\phi_i, \frac{\partial w}{\partial x}\right) = 0 \tag{4.17}$$

$$(\phi_i, \dot{u}) + \left(\phi_i, \frac{\partial f}{\partial x}\right) + \alpha H^2 \left(\phi_i, \frac{\partial^2 \dot{u}}{\partial x^2}\right) = 0 \tag{4.18}$$

$$(\phi_i, w) - \beta H^3 \left(\phi_i, \frac{\partial^2 u}{\partial x^2}\right) = 0 \tag{4.19}$$

Integration by parts is used to reduce the order of the spatial derivatives appearing in this formulation. In the previous chapter the boundary integrals so introduced were ignored as the solitary wave solutions could be set to zero at the computational boundaries. Here the boundary integrals are retained as they will be required in formulating the appropriate numerical boundary conditions for the equations. These are discussed in detail in the following section. Integrating terms of equations (4.17)-(4.19) by parts,

$$(\phi_i, \dot{\eta}) + \left(\phi_i, \frac{\partial p}{\partial x}\right) + \left(\phi_i, \frac{\partial w}{\partial x}\right) = 0 \tag{4.20}$$

$$(\phi_i, \dot{u}) - \left(\frac{d\phi_i}{dx}, f\right) - \alpha H^2 \left(\frac{d\phi_i}{dx}, \frac{\partial \dot{u}}{\partial x}\right) = -\left[\phi_i \left(f + \alpha H^2 \frac{\partial \dot{u}}{\partial x}\right)\right]_{X_1}^{X_2} \tag{4.21}$$

$$(\phi_i, w) + \beta H^3 \left(\frac{d\phi_i}{dx}, \frac{\partial u}{\partial x}\right) = \beta H^3 \left[\phi_i \frac{\partial u}{\partial x}\right]_{X_1}^{X_2} \tag{4.22}$$

The set of linear basis functions $\phi_j(x)$ is used to interpolate an unknown function $v(x,t)$ over the spatial domain as,

$$v(x,t) \approx \sum_{j=1}^{N+1} \phi_j(x)\, v_j(t) \tag{4.23}$$

The approximation (4.23) is used to replace the continuous functions η, u and w. The non-linear terms p and f are interpolated as nodal quantities in the same manner using (4.23).

This form of approximation is known to be accurate for this class of equations [28], and is found to be crucial for the truncation error analysis in Section 4.8. Introducing these spatial approximations into equations (4.20)-(4.22) produces the discrete equations,

$$M_{ij}\dot{\eta}_j + C_{ji}p_j + C_{ji}w_j = 0 \tag{4.24}$$

$$M_{ij}\dot{u}_j - C_{ij}f_j - \alpha H^2 K_{ij}\dot{u}_j = -\left[\phi_i\left(f + \alpha H^2 \frac{\partial \dot{u}}{\partial x}\right)\right]_{X_1}^{X_2} \tag{4.25}$$

$$M_{ij}w_j + \beta H^3 K_{ij}u_j = \beta H^3\left[\phi_i \frac{\partial u}{\partial x}\right]_{X_1}^{X_2} \tag{4.26}$$

where,

$$M_{ij} = \int_\Omega \phi_i \phi_j \, d\Omega \tag{4.27}$$

$$C_{ij} = \int_\Omega \frac{d\phi_i}{dx}\phi_j \, d\Omega \tag{4.28}$$

$$K_{ij} = \int_\Omega \frac{d\phi_i}{dx}\frac{d\phi_j}{dx} \, d\Omega \tag{4.29}$$

The integrals in equations (4.24)-(4.26) are evaluated by integrating over each element individually and summing the results at the nodes. This procedure is simple to automate using the *isoparametric concept* [113] where the integration on each element is mapped to a unit reference element on which the values (4.27)-(4.29) can be precomputed.

Equation (4.26) represents an additional algebraic equation system to solve at each time step which will increase the required computational time perhaps half as much again compared to the finite difference scheme (4.7)-(4.8), which can directly approximate the third spatial derivative. In two dimensions there will be two auxiliary variables and hence two, in general sparse, matrix systems to be solved. The solution of the system (4.26) requires the inversion of the matrix M_{ij}, generally termed the *mass matrix* [113]. A considerable simplification is achieved if the mass matrix can be replaced by a purely diagonal approximation or in the finite element terminology *lumped* [113], as was shown in Section 3.2.3.4 for the KdV equation. This equation can then be decoupled from the $3N$-dimensional system (4.24)-(4.26) and solved explicitly at each time step. This will make the finite element scheme competitive with existing finite difference spatial discretisations which produce only a $2N$-dimensional system of differential equations in time. The lumping procedure replaces equation (4.26) with,

$$w_i = \frac{1}{M_i^L}\left(-\beta H^3 K_{ij}u_j + \beta H^3 \int_\Gamma \phi_i \frac{\partial u}{\partial x} n_x \, d\Gamma\right) \tag{4.30}$$

where,

$$M_i^L = \sum_{j=1}^{N+1} M_{ij} \tag{4.31}$$

It will be shown in Section 4.8 that this approximation does not significantly affect the accuracy of the scheme.

The variable depth equations (4.11)-(4.13) differ from the constant depth system (4.14)-(4.16) in the extra complexity of the dispersive terms. To illustrate the finite element approximation of these terms, a typical dispersive term from equation (4.13) is discretised. Multiplying the term by a trial function $\phi_i(x)$ and integrating over the domain,

$$\begin{aligned}
\left(\phi_i, B_2 h^2 \frac{\partial^2}{\partial x^2}(hu)\right) &= B_2\left(\phi_i h^2, \frac{\partial^2}{\partial x^2}(hu)\right) \\
&= -B_2\left(\frac{d}{dx}\left(\phi_i h^2\right), \frac{\partial}{\partial x}(hu)\right) \\
&\quad + B_2\left[\phi_i h^2 \frac{\partial}{\partial x}(hu)\right]_{X_1}^{X_2} \\
&= -B_2\left(\frac{d\phi_i}{dx} h^2 + \phi_i \frac{d}{dx}(h^2), \frac{\partial}{\partial x}(hu)\right) \\
&\quad + B_2\left[\phi_i h^2 \frac{\partial}{\partial x}(hu)\right]_{X_1}^{X_2}
\end{aligned} \tag{4.32}$$

The unknown functions are compactly approximated in a similar manner to the approximation of the nonlinear functions p and f according to equation (4.23).

$$h^2 \approx \sum_{j=1}^{N+1} \phi_j \left(h^2\right)_j \tag{4.33}$$

$$hu \approx \sum_{j=1}^{N+1} \phi_j \left(hu\right)_j \tag{4.34}$$

Inserting these approximations into the weak form, the finite element approximation of the dispersive term is,

$$\begin{aligned}
\left(\phi_i, B_2 h^2 \frac{\partial^2}{\partial x^2}(hu)\right) &\approx -B_2\left(\frac{d\phi_i}{dx}\phi_k + \phi_i \frac{d\phi_k}{dx}, \frac{d\phi_j}{dx}\right) h_k^2 h_j u_j \\
&\quad + B_2\left[\phi_i h^2 \frac{\partial}{\partial x}(hu)\right]_{X_1}^{X_2}
\end{aligned} \tag{4.35}$$

where summation over the repeated indices j and k is implied. This procedure can be repeated for all the dispersive terms appearing in the variable depth equation system (4.11)-

(4.13).

Li et al. [62] have recently described a similar scheme in order to apply a linear quadrilateral finite element method to Beji and Nadaoka's two-dimensional extended Boussinesq equation system, the generalisation of the one-dimensional system (2.83)-(2.84) presented in Section 2.4.2. These equations contain third derivatives of the free surface elevation and in their case are discretised by introducing the gradient of the free surface as an additional variable, which is calculated with a lumped mass approximation. However they additionally neglect several of the terms, specifically those that involve second derivatives of the depth $h(x)$ [62], whereas in this case the full system of equations is discretised with no further approximations.

4.5 Boundary conditions

4.5.1 Inflow boundaries

In Section 2.4.4 it was shown that the extended Boussinesq equations could be applied at depths right up to the deep water limit $\sigma = H/\lambda = 1/2$. This allows waves to be introduced into the domain sufficiently far from the region of interest that boundary effects can be considered insignificant. If nonlinearity is small at these boundaries, ie. the amplitude is small compared to the depth, then the linearised equations will be a sufficiently accurate approximation to the problem and a linear wave profile can be introduced at the inflow boundary. For the numerical experiments considered in Section 4.9 a regular periodic wave is input at the boundary. For example a simple periodic wave,

$$\eta(x,t) \;=\; a\,sin(kx-\omega t) \tag{4.36}$$

with amplitude a, wave number k and frequency w, where k and ω satisfy the linear dispersion relation for Nwogu's extended Boussinesq equations (2.123). At these boundaries the velocity profile can be derived from the free surface profile (4.36) by using the linearised equation system.

$$u(x,t) \;= b\,sin(kx-\omega t) \tag{4.37}$$

where b is given by equation (2.122) in Chapter 2. It has been found here that Nwogu's extended Boussinesq equations additionally require the specification of spatial derivative information at the inflow boundary. This is consistent with the specification of inflow boundary conditions described by Madsen et al. [66] for their extended Boussinesq equations. In both the finite difference and finite element schemes this is achieved by additionally specifying the variable w on the inflow boundary. At constant depth w is given

by equation (4.16) and using the inflow velocity profile (4.37) a periodic profile can be derived,

$$\begin{aligned} w(x,t) &= \beta H^3 \frac{\partial^2 u}{\partial x^2} \\ &= -\beta H^3 k^2 b \, sin(kx - \omega t) \end{aligned} \tag{4.38}$$

With a variable depth equation (4.13) can be used to derive a similar expression.

This linear wave approximation will become inaccurate if the amplitude is significant compared to the depth at the inflow boundary. This can be accounted for by introducing a first order correction to the linear wave profile for nonlinearity. Considering a nonlinear perturbation of the free surface and velocity from the linearised solution,

$$\eta = \eta_0 + \varepsilon \eta_1 + \mathscr{O}(\varepsilon^2) \tag{4.39}$$

$$u = u_0 + \varepsilon u_1 + \mathscr{O}(\varepsilon^2) \tag{4.40}$$

where the nonlinearity is parameterised by ε, then the constant depth extended Boussinesq equations (2.114)-(2.115) can be written as a series of problems in increasing orders of ε. At lowest order,

$$\dot{\eta_0} + H \frac{\partial u_0}{\partial x} + \beta H^3 \frac{\partial^3 u_0}{\partial x^3} = 0 \tag{4.41}$$

$$\dot{u_0} + g \frac{\partial \eta_0}{\partial x} + \alpha H^2 \frac{\partial^2 \dot{u_0}}{\partial x^2} = 0 \tag{4.42}$$

This has the linearised solution with η_0 and u_0 given by equations (4.36) and (4.37) respectively. To first order in ε,

$$\dot{\eta_1} + H \frac{\partial u_1}{\partial x} + \beta H^3 \frac{\partial^3 u_1}{\partial x^3} = -\frac{\partial}{\partial x}(\eta_0 u_0) \tag{4.43}$$

$$\dot{u_1} + g \frac{\partial \eta_1}{\partial x} + \alpha H^2 \frac{\partial^2 \dot{u_1}}{\partial x^2} = -\frac{\partial}{\partial x}\left(\frac{u_0^2}{2}\right) \tag{4.44}$$

Substituting the solutions η_0 and u_0 into the right hand sides of these equations gives,

$$\begin{aligned} \dot{\eta_1} + H \frac{\partial u_1}{\partial x} + \beta H^3 \frac{\partial^3 u_1}{\partial x^3} &= -\frac{\partial}{\partial x}\left(ab \, sin^2(kx - \omega t)\right) \\ &= -2abk \, sin(kx - \omega t) \, cos(kx - \omega t) \\ &= -abk \, sin(2(kx - \omega t)) \\ \dot{u_1} + g \frac{\partial \eta_1}{\partial x} + \alpha H^2 \frac{\partial^2 \dot{u_1}}{\partial x^2} &= -\frac{\partial}{\partial x}\left(\frac{a^2}{2} \, sin^2(kx - \omega t)\right) \\ &= -a^2 k \, sin(kx - \omega t) \, cos(kx - \omega t) \end{aligned} \tag{4.45}$$

$$= -\frac{a^2 k}{2} sin(2(kx-\omega t)) \tag{4.46}$$

A solution of the form,

$$\eta_1(x,t) = a_1 cos(2(kx-\omega t)) \tag{4.47}$$
$$u_1(x,t) = b_1 cos(2(kx-\omega t)) \tag{4.48}$$

will satisfy this equation system. Substituting these solutions into equations (4.45) and (4.46) leads to the following expressions for a_1 and b_1.

$$a_1 = \frac{1}{g}\left(\frac{\omega}{k} b_1 \left(1-4\alpha(kH)^2\right) - \frac{b^2}{4}\right) \tag{4.49}$$
$$b_1 = \frac{\frac{ab}{2}g + \frac{b^2}{4}\frac{\omega}{k}}{\frac{\omega^2}{k^2}\left(1-4\alpha(kH)^2\right) - gH\left(1-4\beta(kH)^2\right)} \tag{4.50}$$

The wave profile (4.39) will have a slightly sharper peak and a broader, slightly raised trough, and is the first approximation to a more general periodic wave form known as a cnoidal wave [109]. Note that this modified wave involves a wavelength of half the primary wavelength and hence accurate modelling will require that this is properly resolved, leading in general to twice the resolution of the original wave.

The inflow boundary condition is based on a linearised approximation and hence additional wave components can be introduced at that boundary by simple superposition. However if the input wave is composed of two or more periodic components then it is possible that nonlinear interactions between these components will produce significant harmonic waves. The most important of these waves in coastal engineering applications is the subharmonic, or set-down, wave [21, 22, 98]. Nwogu derived formulae for the harmonic waves produced by two periodic wave components [72], and also extended this analysis to two-dimensional multi-directional waves [73]. Here the derivation is outlined for an initial wave profile consisting of two cosine waves,

$$\eta_0(x,t) = a_1\, cos(k_1 x-\omega_1 t) + a_2\, cos(k_2 x-\omega_2 t) \tag{4.51}$$

The velocity profile can be derived from the linearised equation system,

$$u_0(x,t) = b_1 a_1\, cos(k_1 x-\omega_1 t) + b_2 a_2\, cos(k_2 x-\omega_2 t) \tag{4.52}$$

where b_1 and b_2 are given by,

$$b_1 = \frac{\omega_1}{k_1 H(1-\beta(k_1 H)^2)} \tag{4.53}$$

$$b_2 = \frac{\omega_2}{k_2 H(1-\beta(k_2 H)^2)} \tag{4.54}$$

$$\tag{4.55}$$

The first order nonlinear wave contains the harmonic interactions of these two components in the form [72],

$$\begin{aligned}\eta_1(x,t) &= a_1 a_2 G_{\pm}\, cos(k_{\pm}x - \omega_{\pm}t) + \frac{a_1^2}{2} G_{+}\, cos(2(k_1 x - \omega_1 t)) \\ &\quad + \frac{a_2^2}{2} G_{+}\, cos(2(k_2 x - \omega_2 t))\end{aligned} \tag{4.56}$$

where,

$$k_{\pm} = k_1 \pm k_2 \tag{4.57}$$

$$\omega_{\pm} = \omega_1 \pm \omega_2 \tag{4.58}$$

and $G_{\pm}$, termed the quadratic transfer function [72], relates the amplitudes of the harmonic waves to those of the primary waves. Considering only the subharmonic, or set-down, components of the first order free surface elevation and velocity,

$$\eta_{-}(x,t) = a_1 a_2 G_{-}\, cos(k_{-}x - \omega_{-}t) \tag{4.59}$$

$$u_{-}(x,t) = a_1 a_2 Q_{-}\, cos(k_{-}x - \omega_{-}t) \tag{4.60}$$

and substituting these, and the primary wave components (4.51) and (4.52) into the first order nonlinear equations (4.43)-(4.44) one can, after some algebraic manipulation, derive expressions for G_{-} and Q_{-}.

$$G_{-} = \frac{\frac{b_1+b_2}{2}\frac{\omega_{-}}{k_{-}}\left(1-\alpha(k_{-}H)^2\right) + \frac{b_1 b_2}{2} H\left(1-\beta(k_{-}H)^2\right)}{\frac{\omega_{-}^2}{k_{-}^2}\left(1-\alpha(k_{-}H)^2\right) - gH\left(1-\beta(k_{-}H)^2\right)} \tag{4.61}$$

$$Q_{-} = \frac{\frac{\omega_{-}}{k_{-}H} G_{-} - \frac{b_1+b_2}{2H}}{1-\beta(k_{-}H)^2} \tag{4.62}$$

It has been noted in the literature that if these set-down interactions are neglected then a spurious free long wave will be produced at the inflow boundary with the same period but phase-shifted by 180 degrees [98].

In a practical situation it is more likely that inflow boundary conditions will be in the form of an irregular free surface profile, such as from measured data [72]. This can be handled by a Fourier decomposition into a series of periodic wave components [106]. The individual periodic components and any significant nonlinear wave interactions can

be calculated by the methods previously discussed in this section and this composite wave can then be input to the system.

The specification of all the variables at inflow boundaries does not allow for the waves returning to this boundary, eg. waves reflected from inside the domain. Such waves will then be reflected in a non-physical manner and contaminate the numerical solution. For some of the numerical tests considered in Section 4.9 wave reflection is not significant; however it is important to have a more general purpose inflow boundary condition, in particular if realistic two-dimensional problems are to be attempted. This problem is considered in detail in Section 4.5.4

4.5.2 Solid wall boundaries

At a solid wall the physical boundary condition is that of impermeability, ie. $u \cdot n = 0$, where n is the normal to the wall at that point, which in one dimension reduces to,

$$u = 0 \tag{4.63}$$

at the wall.

A boundary condition on the free surface can be derived from a conservation argument [106]. Integrating the free surface equation (4.11) over the spatial domain, and using the divergence theorem on the spatial derivatives,

$$\begin{aligned} \int_\Omega \left(\dot{\eta} + \frac{\partial p}{\partial x} + \frac{\partial w}{\partial x} \right) d\Omega &= 0 \\ \frac{\partial}{\partial t} \left(\int_\Omega \eta \, d\Omega \right) + \int_\Gamma (p+w) n \, d\Gamma &= 0 \end{aligned} \tag{4.64}$$

where Γ is the domain boundary, and n is the outward normal. At a solid boundary the amount of mass in the system is conserved and hence the time derivative term in expression (4.64) is zero. For the boundary integral to be equal to zero we require $p+w = 0$ on the boundary. This is satisfied by the wall boundary condition on the velocity (4.63), from the definition of p (4.3), and the additional requirement on the auxiliary variable w,

$$w = 0 \tag{4.65}$$

at the wall. The boundary conditions (4.63) and (4.65) together will imply conservation at the wall from equation (4.64) and hence no boundary condition is directly imposed on the free surface.

In the literature it has been common to specify that the normal derivative of the free surface is zero at solid boundaries although this does not follow directly from the equa-

tions. Wei and Kirby [106] derive this condition by considering a first order form of the free surface equation (4.1). Li et al. [62] use this condition as it is compatible with their choice of the gradient of the free surface as an additional variable. However it is felt that the method described here is more consistent with the equation system and the underlying physics.

Another useful boundary condition for practical situations is that of partial reflection or partial absorption, to simulate for example a beach or other non-vertical boundaries, or other boundaries for which perfect reflection is inappropriate. Within the context of this depth-averaged mathematical model all boundaries are assumed to be vertical and hence such boundary conditions cannot be directly formulated. Partial reflection has been implemented in previous work by approximations using linear theory [72] and by a simplified characteristic analysis [47]. The effect of these conditions would seem to be a loss of mass in the system at these boundaries which may be appropriate for porous boundaries but not for other situations. In general the desired effect would be a loss of wave energy at these boundaries rather than mass, although it is unclear exactly how to formulate this boundary condition at present.

4.5.3 Outflow boundaries

An outflow boundary is a non-physical restriction of the spatial domain to a suitably small size in order that practical computations can be performed. Hence at these boundaries it is important to minimise non-physical reflection of information back into the domain. A simple radiation boundary condition [112] is not completely effective as the dispersive waves have no single phase speed. Previous work on this problem has led to higher order radiation boundary conditions [39] and to the addition of source terms and viscous damping terms to the momentum equations [106] to absorb outgoing waves. Here a viscous damping layer, termed a sponge layer [48], is introduced near the outflow boundary in the form of a viscous term which increases in magnitude closer to the outflow boundary. This viscous term is added to the free surface equation in the form illustrated here for the constant depth equation (4.5).

$$\dot{\eta} + \frac{\partial p}{\partial x} + \beta H^3 \frac{\partial^3 u}{\partial x^3} \;=\; c_s \frac{\partial^2 \eta}{\partial x^2} \tag{4.66}$$

where,

$$c_s \;= n_1 \frac{e^{\left(\frac{x-X_s}{X_2-X_s}\right)^{n_2}} - 1}{e^1 - 1} \tag{4.67}$$

for $x \in [X_s, X_2]$ and $c_s = 0$ everywhere else. n_1 and n_2 are constants used to fine tune the amount of viscosity. A dimensional analysis of the equation (4.66) shows that,

$$n_1 = c_v(H\sqrt{gH}) \tag{4.68}$$

is the appropriate choice for an $\mathcal{O}(1)$ constant c_v.

The actual amount of viscosity is problem dependent and has to be determined by experiment so as to effectively damp the outgoing waves without allowing significant reflection. It is found in practice that the width of the sponge layer must be two to three times the wavelength to provide sufficient damping for the cases considered here.

4.5.4 Internal wave generation

For cases where there will be significant reflected waves at the inflow boundaries the methods described in Section 4.5.1 will not be sufficient. The specification of all the variables at this boundary does not allow for waves returning to the inflow region and passing out of the domain. The internal generation of waves is proposed here as a method of allowing reflected waves to propagate through the wave generator and hence leave the computational domain. This is essential if long time simulations of geometrically complex two-dimensional regions are to be performed.

Larsen and Dancy [60] described a method by which waves could be generated internally in a finite difference method for the Boussinesq equations. They added a volume of fluid, equivalent to that which would be displaced by the desired wave at each time step of the method, at a chosen generation point and were able to generate periodic waves in both one and two dimensions. Importantly, these generation points allowed outgoing waves to pass undisturbed and these waves were damped out with sponge layers at all open boundaries. This method was also incorporated into a boundary integral method for the Boussinesq equations by Brorsen and Larsen [24], and further developed by Ohyama and Nadaoka [75] in their implementation of a numerical wave tank. Skotner and Apelt [97] recently reported the application of this method to a finite difference method for Nwogu's extended Boussinesq equations. They found that instabilities were produced due to the introduction of the wave at a single point, although they note that Larsen and Dancy used a staggered grid which would alleviate this problem in their case. Skotner and Apelt [97] modified the spatial difference stencils in the neighbourhood of the wave generation region to prevent this instability. More recently Wei et al. [108] have presented a method for deriving a source function, or alternatively an applied pressure distribution, from the potential form of the Boussinesq equations that can be used to generate accurate wave conditions internally to the domain.

Here, in contrast to that work, an analogy is drawn with a time varying sea bed, and a

term is added to the free surface equation that simulates the lowest order effects of a time varying depth profile. In the free surface equation (4.5) this takes the form,

$$\dot{\eta} + \dot{h_I} + \frac{\partial p}{\partial x} + \beta H^3 \frac{\partial^3 u}{\partial x^3} = 0 \tag{4.69}$$

where the form of the sea bed variation h_I is taken to be,

$$\dot{h_I}(x,t) = c_I \; f_I(x) \; a \; sin(-\omega t) \tag{4.70}$$

where a and ω are the amplitude and frequency of the required wave respectively, and the constant c_I and spatial function $f_I(x)$ are given by,

$$c_I = 2 \; \omega \; c \tag{4.71}$$

$$f_I(x) = \frac{b_I}{\sqrt{\pi}} \, e^{-b_I^2 (x - x_I)^2} \tag{4.72}$$

This simulates an undulating Gaussian hill at the position $x = x_I$. The wave generation region is illustrated in Fig. 4.1. The width of this region $[x_I - a_I, \; x_I + a_I]$ is generally

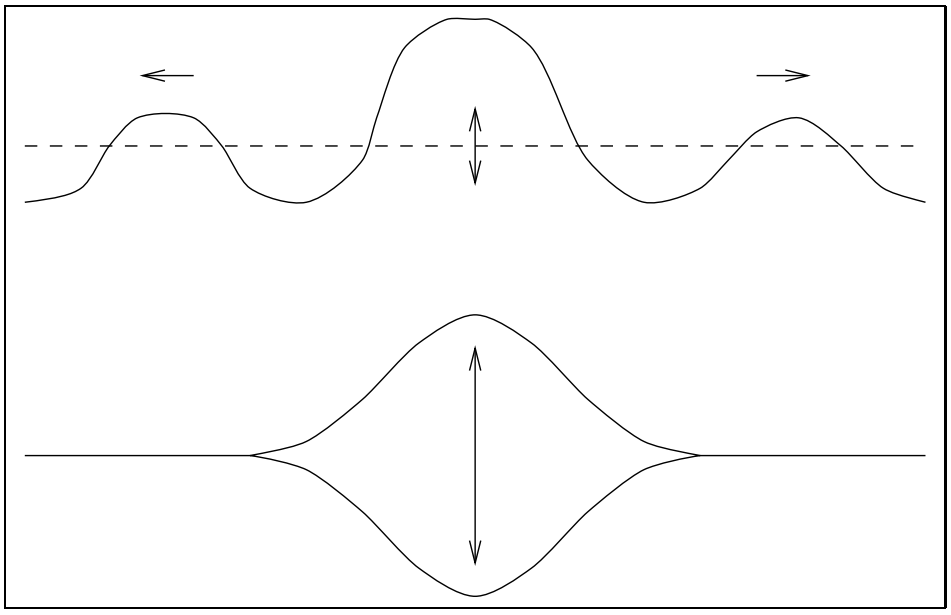


Figure 4.1: Schematic of the internal wave generation process

taken to be one or two wavelengths, which is consistent with the scaling of the variable depth by equation (2.39) in Section 2.3 which requires variations in the depth to be small over distances comparable to the wavelength. The coefficient b_I is chosen such that the function $f_I(x)$ is effectively zero at $x = x_I \pm a_I$ which is essential for the depth variation to appear smooth to the time integration software. The function $f_I(x)$ is normalised such that its integral is one over this region. The constant c_I includes a factor 2 as the wave

generator will produce both a left and right moving wave, the wave speed c to give the expression the correct physical dimensions, and the wave frequency ω to compensate for differentiation with respect to time.

Note that the use of the Gaussian function to distribute the generated wave over several mesh points is similar in principle to the method recently described by Wei et al. [108], although this was published after this part of the work was complete. Their method however is more general, and more rigorously derived and extends straightforwardly to multi-component and irregular waves.

Numerical experiments with the approach described here are presented in Section 4.9.6.

4.6 Time integration

Truncation errors from a low order time integration scheme will also produce non-physical dispersion and contaminate the mathematical model. In previous work the time integration has usually been with a high order accuracy predictor-corrector method [106, 112], or by a lower order method with corrections for the truncation errors [1, 72]. In this work the necessary accuracy is achieved by making use of variable order, variable time step integration as implemented in the SPRINT software [18]. In the previous chapter it was found that the time integration module spdasl was robust enough for all the numerical tests considered and this module is used in all the computations described in Section 4.9.

The spatial discretisations described in Sections 4.4.1 and 4.4.2 produce a $2(N+1)$-dimensional linear system of differential equations in the form,

$$A_{ij}\dot{y}_j - f_i = 0 \tag{4.73}$$

where,

$$y = (\eta_1, u_1, \eta_2, u_2, \ldots, \eta_{N+1}, u_{N+1})^t \tag{4.74}$$

$$f = f(t, y) \tag{4.75}$$

The ordering of the unknowns in y maintains the banded structure of the matrix A_{ij} although the bandwidth is doubled compared to the numerical models of the RLW and KdV equations considered in Chapter 3 due to there being two unknowns at each node of the mesh. The time integration software can use banded direct solution strategies to take advantage of the banded matrix structure to increase the efficiency of the time integration process.

Use of the time integration software and the error control strategy are as described in Section 3.3 in the previous chapter. The software controls the local error in y over

each time step by varying both the order of the BDF method and the time step, so that a uniformly high accuracy in time can be achieved [61]. The user controls the local error by supplying relative and absolute tolerances, *rtol* and *atol* respectively, which are here taken to be scalar values determined by experiment. Their values are lowered until solution independency is established. This will ensure that the error in the solution is dominated by the spatial discretisation error. In this work $rtol = atol = 10^{-6}$ was chosen after repeated experiments, reflecting the high accuracy required in the time integration.

4.7 Initial conditions

Consistent initial conditions are calculated automatically by the software. Given an initial solution, $y(t_0)$, initial time derivatives, $\dot{y}(t_0)$, are computed such that the system (4.73) is satisfied [19]. Except for the solitary wave problems it is common to begin the computation with an undisturbed free surface and then introduce a wave at the inflow boundary as described in the previous section. The initialisation problem is then satisfied trivially with a zero solution. In this case however the wave can appear discontinuous to the time integration software at the point it initially enters the domain and this forces the software to use very small time steps. This is remedied here by the use of a smoothing function,

$$c_t \;=\; m_1\, e^{-m_2 \frac{t}{\tau}} \tag{4.76}$$

where τ is the wave period, and m_1 and m_2 are constants that must be determined by experiment. This term is added to the free surface equation as a viscous coefficient in an identical manner to the sponge layer coefficient c_s in equation (4.66). This term damps the solution throughout the domain for the initial time steps but should not affect the solution obtained after a suitably large time. This form of damping is found to improve the efficiency of the time integration, which can take larger time steps if the solution is smooth. A similar method has subsequently been reported by Li et al. [62] who use a fixed time step predictor-corrector time stepping method. In their case the initial damping of the solution prevents transient oscillations caused by the discontinuous initial conditions.

Another important consideration with respect to the efficiency of the time integration process is that of the choice of initial time step. With an initially undisturbed free surface the time integration algorithm has no information on the time scales in the problem and so will do a considerable amount of work to calculate the initial step size. This involves the numerical approximation and factorisation of the Jacobian matrix. In this one-dimensional case the Jacobian matrix is banded and hence the computational expense is not too significant. In two dimensions the Jacobian matrix will be sparse and its computation and factorisation will be considerably more computationally expensive. The time

integration software provides an option to supply the initial time step and if an appropriate value is known this is the most efficient mode of operation. A starting guess of $\frac{\tau}{100}$ has been used in this work although the value is generally problem dependent. If the software is allowed to select an initial time step size then this value can be noted and supplied to the code in subsequent runs of the same problem. Note that the value will be significantly smaller than that used by the software for the majority of the time interval as the initial step has to be taken with a first order BDF method and the order gradually raised and time step adjusted according to the error control parameters.

4.8 Analysis of the method

In this section the truncation error of the linear finite element scheme defined by equations (4.24), (4.25) and (4.30) will be analysed. The linear finite element scheme is only second order accurate in space and it is therefore possible that error terms will contaminate the physical dispersion in a similar way to a low order finite difference method. The analysis that follows will show that this does not happen here and that the truncation error is of a similar form to that of Wei and Kirby's finite difference scheme (4.7)-(4.8) described in Section 4.4.1.

With a linear finite element discretisation on a regular mesh of element length Δ equation (4.25) has the following form at an internal node;

$$\frac{\Delta}{6}(\dot{u}_{i-1}+4\dot{u}_i+\dot{u}_{i+1})+\frac{1}{2}(f_{i+1}-f_{i-1})-\frac{\alpha H^2}{\Delta}(-\dot{u}_{i-1}+2\dot{u}_i-\dot{u}_{i+1}) \;=\; 0 \qquad (4.77)$$

Substituting standard Taylor series expansions in space into equation (4.77), and rearranging the terms,

$$\begin{aligned} &\left(\dot{u}_i+\frac{\partial f_i}{\partial x}+\alpha H^2\frac{\partial^2\dot{u}_i}{\partial x^2}\right)\\ &+\frac{\Delta^2}{6}\frac{\partial^2}{\partial x^2}\left(\dot{u}_i+\frac{\partial f_i}{\partial x}+\alpha H^2\frac{\partial^2\dot{u}_i}{\partial x^2}\right)\\ &-\alpha H^2\frac{\Delta^2}{12}\frac{\partial^4\dot{u}_i}{\partial x^4}\\ &+\Delta^4\left(\frac{1}{72}\frac{\partial^4\dot{u}_i}{\partial x^4}+\frac{1}{120}\frac{\partial^5 f_i}{\partial x^5}+\frac{\alpha H^2}{360}\frac{\partial^6\dot{u}_i}{\partial x^6}\right) \;=\; \mathscr{O}(\Delta^6) \end{aligned} \qquad (4.78)$$

Using the original equation (4.25) it can be seen that the truncation error of the scheme at a node has the form,

$$-\alpha H^2\frac{\Delta^2}{12}\frac{\partial^4\dot{u}_i}{\partial x^4}+\Delta^4\left(\frac{1}{72}\frac{\partial^4\dot{u}_i}{\partial x^4}+\frac{1}{120}\frac{\partial^5 f_i}{\partial x^5}+\frac{\alpha H^2}{360}\frac{\partial^6\dot{u}_i}{\partial x^6}\right)+\mathscr{O}(\Delta^6) \qquad (4.79)$$

The leading term is $\mathscr{O}(\Delta^2)$ due to the linear approximation but multiplies a mixed fifth order derivative. This will not corrupt the physical dispersion present in the system, which depends on third and lower derivatives, and suggests that the linear finite element approximation is acceptable for this equation.

With a linear finite element discretisation on a regular mesh of element length Δ, equations (4.24) and (4.30) have the following form at an internal node;

$$\frac{\Delta}{6}(\dot{\eta}_{i-1}+4\dot{\eta}_i+\dot{\eta}_{i+1})+\frac{1}{2}(p_{i+1}-p_{i-1})+\frac{1}{2}(w_{i+1}-w_{i-1}) = 0 \tag{4.80}$$

$$\Delta w_i+\frac{\beta H^3}{\Delta}(-u_{i-1}+2u_i-u_{i+1}) = 0 \tag{4.81}$$

Again, using Taylor series expansions in space and then differentiating equation (4.81) and using it to replace the w_i terms in equation (4.80),

$$\begin{aligned}
&\left(\dot{\eta}_i+\frac{\partial p_i}{\partial x}+\beta H^3\frac{\partial^3 u_i}{\partial x^3}\right)\\
&+\frac{\Delta^2}{6}\frac{\partial^2}{\partial x^2}\left(\dot{\eta}_i+\frac{\partial p_i}{\partial x}+\beta H^3\frac{\partial^3 u_i}{\partial x^3}\right)\\
&+\beta H^3\frac{\Delta^2}{12}\frac{\partial^5 u_i}{\partial x^5}\\
&+\Delta^4\left(\frac{1}{72}\frac{\partial^4\dot{\eta}_i}{\partial x^4}+\frac{1}{120}\frac{\partial^5 p_i}{\partial x^5}+\frac{\beta H^3}{40}\frac{\partial^7 u_i}{\partial x^7}\right) = \mathscr{O}(\Delta^6)
\end{aligned} \tag{4.82}$$

Using the original equation (4.24) the truncation error at a node is,

$$\beta H^3\frac{\Delta^2}{12}\frac{\partial^5 u_i}{\partial x^5}+\Delta^4\left(\frac{1}{72}\frac{\partial^4\dot{\eta}_i}{\partial x^4}+\frac{1}{120}\frac{\partial^5 p_i}{\partial x^5}+\frac{\beta H^3}{40}\frac{\partial^7 u_i}{\partial x^7}\right)+\mathscr{O}(\Delta^6) \tag{4.83}$$

The leading truncation error term indicates that the discretisation is $\mathscr{O}(\Delta^2)$ accurate but that the form of this leading error will not contaminate the physical dispersion. Importantly, the accuracy does not appear to have been significantly reduced by the diagonal approximation of the mass matrix in equation (4.81).

It has been shown that a linear finite element discretisation can produce a suitable numerical model for Nwogu's extended Boussinesq equations. The model is only second order accurate but the form of the leading error terms will not contaminate the physical dispersion present in the equations. This scheme can therefore be thought of as an analogue of Wei and Kirby's finite difference scheme which approximates the first order spatial derivatives to higher order to produce a similar effect. Lumping the mass matrix in equation (4.26) considerably improves the efficiency of the scheme as w can now be calculated without the inversion of a matrix. In two dimensions this saving will be even more significant. This truncation error analysis can be directly extended to the Boussinesq

system with non-uniform depth (4.11)-(4.13) discretised on a regular mesh.

The proposed finite element method is simple to extend to a non-uniform mesh. However the truncation error analysis of this section has to be modified to account for a variation in the mesh size. For example, if Δ_i denotes the length of the element $[x_i, x_{i+1}]$, the truncation error of equation (4.25) can, with some manipulation, be shown to be,

$$\left(\frac{\Delta_i - \Delta_{i-1}}{6}\right)\frac{\partial^2 f_i}{\partial x^2} - \frac{1}{12}\left(\frac{\Delta_i^3 + \Delta_{i-1}^3}{\Delta_i + \Delta_{i-1}}\right)\alpha H^2 \frac{\partial^4 \dot{u}_i}{\partial x^4} + \mathscr{O}(\Delta_{i-1}^3, \Delta_i^3) \tag{4.84}$$

The leading term will produce numerical diffusion in the scheme. It may be possible to restrict the amount of diffusion present by limiting the rate of variation in the mesh size, ie. keeping $(\Delta_i - \Delta_{i-1})$ small. An example is included in the following section to illustrate the effect of a non-uniform mesh.

4.9 Numerical experiments

4.9.1 A solitary wave with constant depth

The simulation of a solitary wave requires that both the nonlinearity and dispersion are accurately modelled. This case is used here to compare the accuracy of the finite difference and finite element models and the time integration strategy. Wei and Kirby [106] have derived a solitary wave solution of Nwogu's extended Boussinesq system. This solitary wave is not exact but is derived from the scaled, nondimensionalised equation system by neglecting terms of order greater than that retained in the Boussinesq model itself. The analytical forms of the free surface and velocity derived by Wei and Kirby [106] are stated here, and additional details not mentioned in their work, concerned with the calculation of the free parameter C, are discussed in more depth.

$$\eta(x,t) = a_1\, sech^2\left(b\left(x - \bar{C}t\right)\right) + a_2\, sech^4\left(b\left(x - \bar{C}t\right)\right) \tag{4.85}$$

$$u(x,t) = a\, sech^2\left(b\left(x - \bar{C}t\right)\right) \tag{4.86}$$

where,

$$\bar{C} = \sqrt{gH}\, C \tag{4.87}$$

$$a_1 = \frac{H}{3}\left(\frac{C^2 - 1}{\beta - \alpha C^2}\right) \tag{4.88}$$

$$a_2 = -\frac{H}{2}\left(\frac{C^2 - 1}{C}\right)^2 \left(\frac{\beta + 2\alpha C^2}{\beta - \alpha C^2}\right) \tag{4.89}$$

$$a = \sqrt{gH}\left(\frac{C^2-1}{C}\right) \tag{4.90}$$

$$b = \frac{1}{2H}\sqrt{\left(\frac{C^2-1}{\beta-\alpha C^2}\right)} \tag{4.91}$$

The value of the free parameter C is derived by choosing a wave amplitude δH, noting that the definition of a physically appropriate wave within the Boussinesq framework requires $\delta \ll 1$. From the solitary wave solution (4.85),

$$\delta H = a_1 + a_2 \tag{4.92}$$

Substituting for a_1 and a_2 and rearranging leads to a cubic equation for C^2 [106].

$$2\alpha\left(C^2\right)^3 - \left(3\alpha+\frac{1}{3}+2\alpha\delta\right)\left(C^2\right)^2 + 2\delta\left(\alpha+\frac{1}{3}\right)\left(C^2\right) + \left(\alpha+\frac{1}{3}\right) = 0 \tag{4.93}$$

This equation will generally have three real roots for the ranges of δ and α considered here, two of which are positive and hence consistent with the definition of a right moving wave. An estimate of the value of C can be obtained by choosing $\alpha = -1/3$ which reduces the extended Boussinesq equations to a form equivalent to Peregrine's original system [72]. In this case the expression (4.93) simplifies to,

$$-\frac{2}{3}\left(C^2\right)^2\left(C^2-(1+\delta)\right) = 0 \tag{4.94}$$

and $C^2 = 1+\delta$ is the only positive solution. This will provide a rough estimate of the appropriate value for C, as the value of the extended Boussinesq parameter α used in practice is generally not far from $-1/3$, typically being in the range $[-0.40, -0.38]$ [72]. The correct value of C can be calculated with a simple bisection search of the expression (4.93), with an initial bracket for the root established in a neighbourhood of $C^2 = 1+\delta$.

The spatial domain $x \in [-100, 300]$ is uniformly discretised with 800 linear finite elements and, equivalently, 801 equally spaced finite difference nodes. The spatial domain is infinite theoretically but in practice it is truncated at a suitable distance from the wave and all variables are set to zero at the boundaries. Inflow and outflow boundary conditions are not considered in this example. The wave is initially at the origin and of amplitude 0.045m (δ=0.1) and is propagated for 100s over an undisturbed depth of 0.45m. It was found that the solitary wave exhibited a slight phase error and so the experiment was repeated with 1601 nodes. Fig. 4.2(a) compares the theoretical free surface profile of equation (4.85) at t=100s with the numerical results from both the proposed finite element scheme and the finite difference scheme described in Section 4.4.1 coupled with the SPRINT time integration module spdasl. Only the portion $x \in [200, 230]$ of the full spatial domain is shown

to emphasise the differences in the numerical and analytical solutions. Both methods propagate the solitary wave accurately and there is no apparent phase error in the results, and only a very slight dispersive tail.

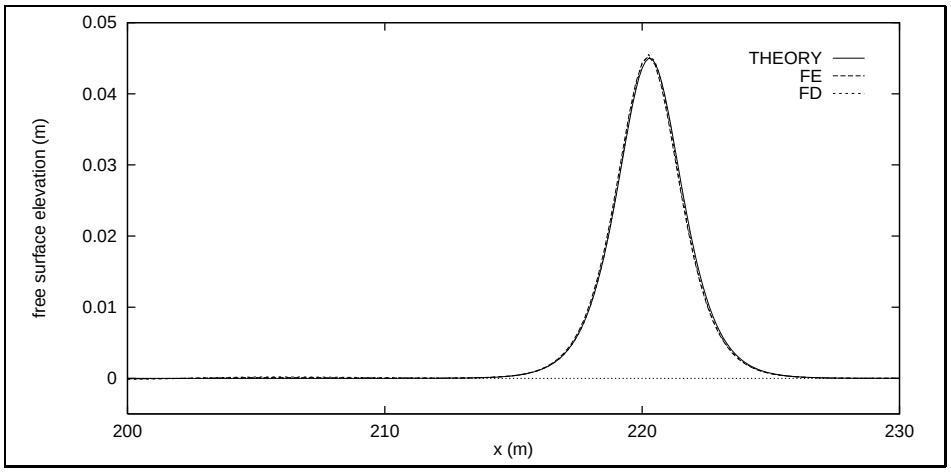
The experiment was repeated with a larger solitary wave of amplitude 0.09m (δ=0.2) propagated for 100s over the same spatial domain. 1601 nodes were used for both the finite element and finite difference methods. Fig. 4.2(b) compares the theoretical and numerical results for this case. In this case a slight phase error is apparent, with the numerical solutions slightly behind the theoretical solution, and the peak amplitude is slightly over-predicted. This is in agreement with the observations of Wei and Kirby [106] for their finite difference numerical solutions. The finite difference and finite element methods give almost identical results.

The experiment was also performed with a solitary wave of amplitude 0.18m (δ=0.4). Fig. 4.2(c) compares the theoretical and numerical spatial profiles at t=100s for this case. Again there is an obvious phase error and both numerical schemes slightly over-predict the peak amplitude. A dispersive wave train is apparent behind this high solitary wave, which is why a larger portion of the spatial domain is shown in this case. This reflects the approximate nature of the solitary wave solution for these equations with the differences becoming more apparent for higher waves, and is in agreement with the results of Wei and Kirby [106] for a δ=0.3 solitary wave.

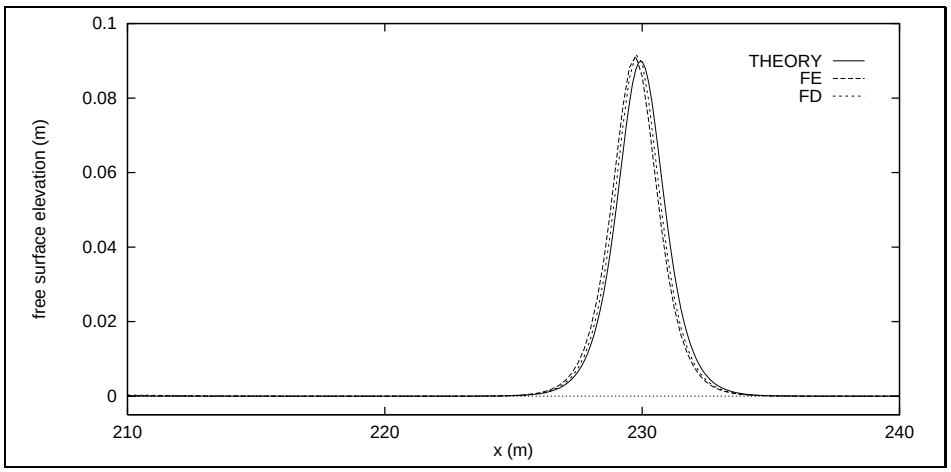
4.9.2 A solitary wave propagating up a slope and reflecting from a vertical wall

The depth profile for this problem is shown in Fig. 4.3, and consists of a constant depth region $x \in [-55m, 0m]$, a $1/50$ gradient slope for $x \in [0m, 20m]$, and a vertical wall at $x = 20m$. Experimental data is available for this problem in the form of time series of the free surface elevation at various locations on the slope [33]. The spatial domain $x \in [-55m, 20m]$ is uniformly discretised with 600 linear finite elements and the equations are integrated in time for $t \in [0s, 30s]$ with the wave initially centred at $x = -30m$ and specified according to the equations (4.85) and (4.86) given in Section 4.9.1. At $x = -55m$ all variables are set to zero and at $x = 20m$ the fully reflecting boundary conditions described in Section 4.5.2 are applied.

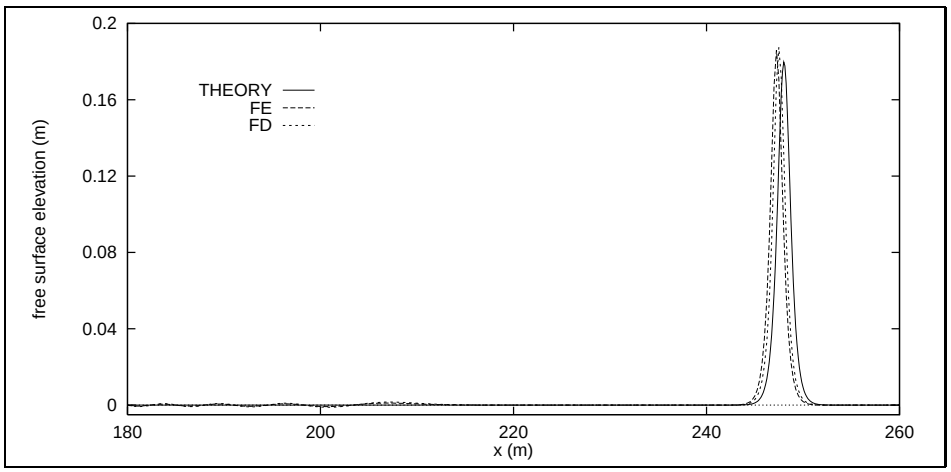
Figs. 4.4(a)-(c) compare the numerical and experimental time series of the free surface elevation at $x = 0.0m$, $x = 16.25m$ and $x = 17.75m$ respectively for a solitary wave of initial amplitude $0.07m$. On each graph the first peak corresponds to the incident wave and the second and subsequent peaks to the reflected wave. It can be seen from Fig. 4.4(c) that the incident wave is modelled with only a slight phase difference apparent by the time the wave nears the top of the slope at $x = 17.75m$. The reflected wave is modelled less



(a) 0.045m solitary wave (δ=0.1)



(b) 0.09m solitary wave (δ=0.2)



(c) 0.18m solitary wave (δ=0.4)

Figure 4.2: Spatial free surface profile of the solitary wave at t=100s

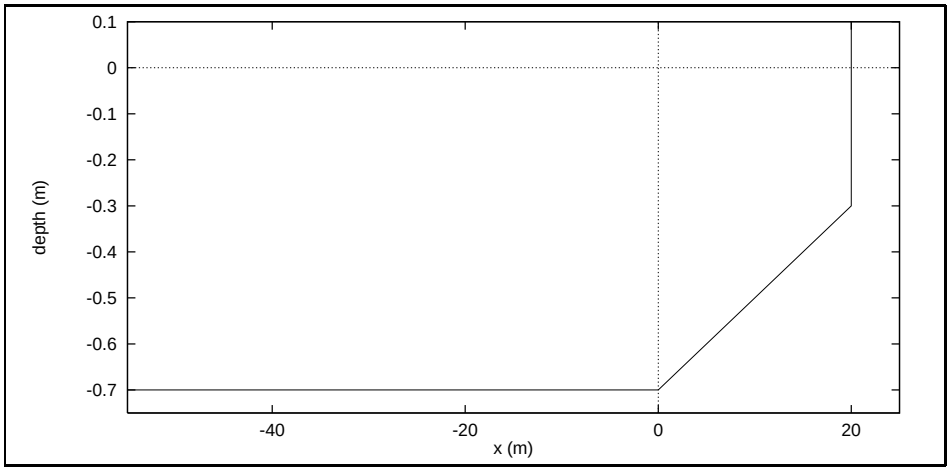
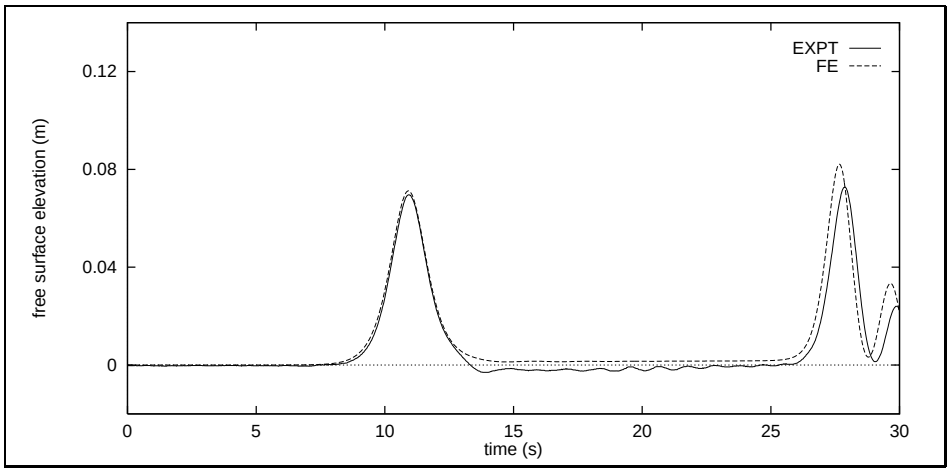


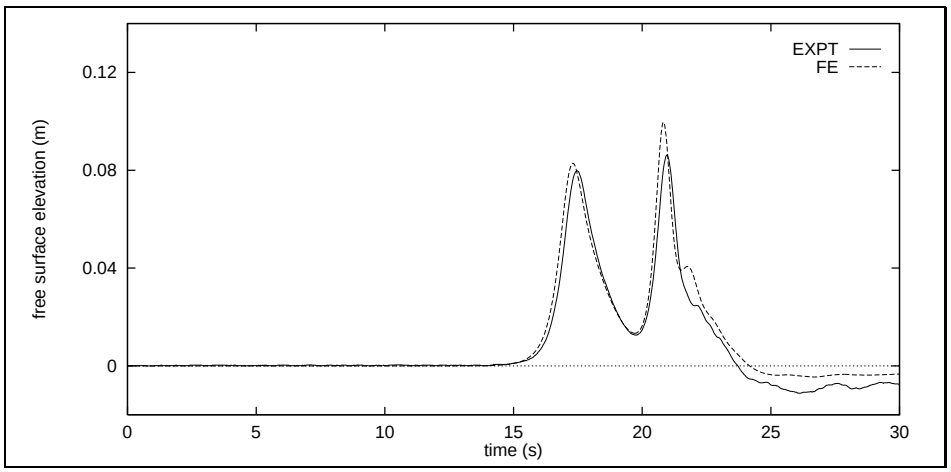
Figure 4.3: Bathymetry for the solitary wave test cases

accurately and is generally over-predicted by the finite element method. Fig. 4.4(c) shows that at $x = 17.75m$, shortly after reflection from the wall, the wave is over-predicted by 8%. It is possible that this over-prediction is produced because the method is based on inviscid equations and also by the assumption of a perfect reflection with no frictional losses at the wall. This is apparent in Figs. 4.4(b)-(c) where the experimental results exhibit a larger trailing wake after reflection. Fig. 4.4(a) shows the return of the wave to the foot of the slope at approximately $t = 26s$ and also the appearance of a second trailing wave which can be seen beginning to appear in the reflected wave at $x = 16.25m$ in Fig. 4.4(b).

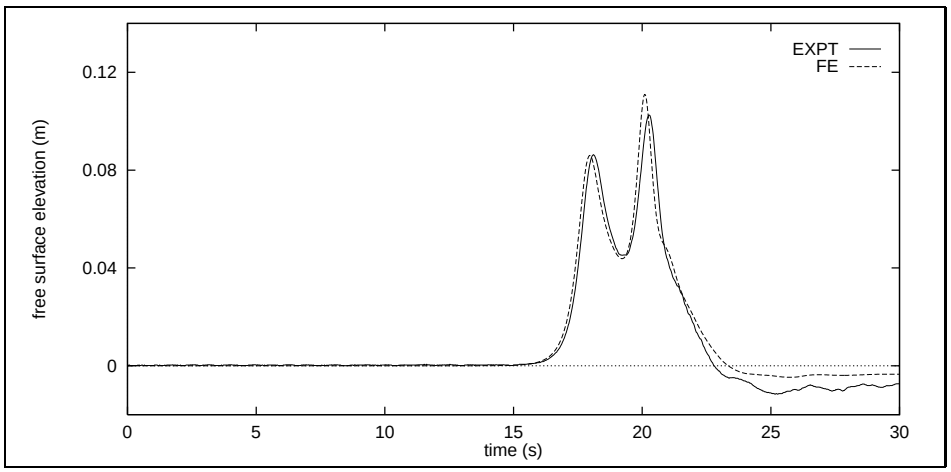
This experiment was repeated with a solitary wave of larger initial amplitude $0.12m$. In this case the experimental data is more noisy and it was found that the numerical code under-predicted the incident wave at $x = 0.0m$. A better numerical model of the incident wave was obtained by using an initial amplitude of $0.1284m$ and this was used to generate the results shown in Figs. 4.5(a)-(c) which compare time series of free surface elevation at the same locations as before. In this case the incident wave is significantly over-predicted at both points on the slope and the reflected wave is over-predicted to an even greater extent, with an error of 38% at $x = 17.75m$. In part this error can be attributed to the inviscid model and assumptions of perfect reflection, and also to the fact that the experimental data appears to exhibit a large degree of noise even before the incident wave arrives at the measuring points. The differences may also be due to the fact that the extended Boussinesq equations are restricted to weakly nonlinear situations whereas in this case the reflected wave attains an amplitude comparable to the depth at the wall, which is clearly outside the range of validity of the Boussinesq theory as stated in Section 4.3. Fig. 4.5(a) shows the wave returning to the foot of the slope at about $t = 25s$ and although the results are not accurate the model is still robust enough to capture the decomposition of the reflected wave into a train of solitary waves in this highly nonlinear



(a) Free surface time history at $x = 0.0m$

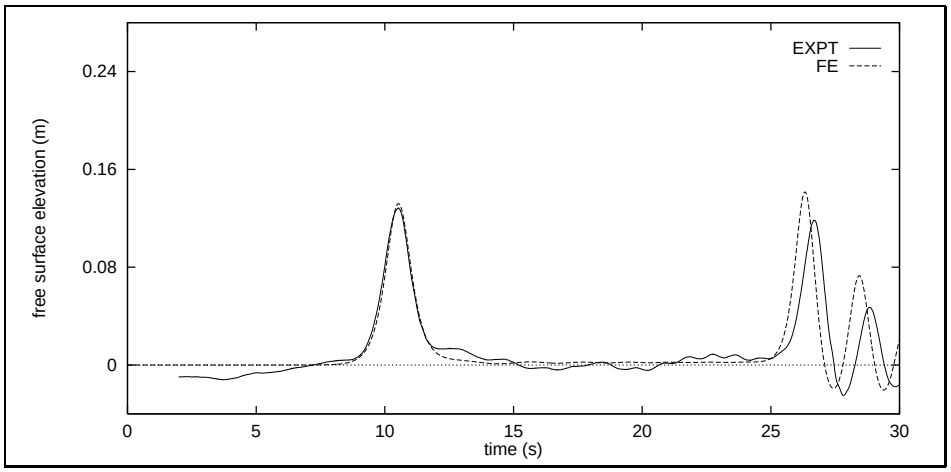


(b) Free surface time history at $x = 16.25m$

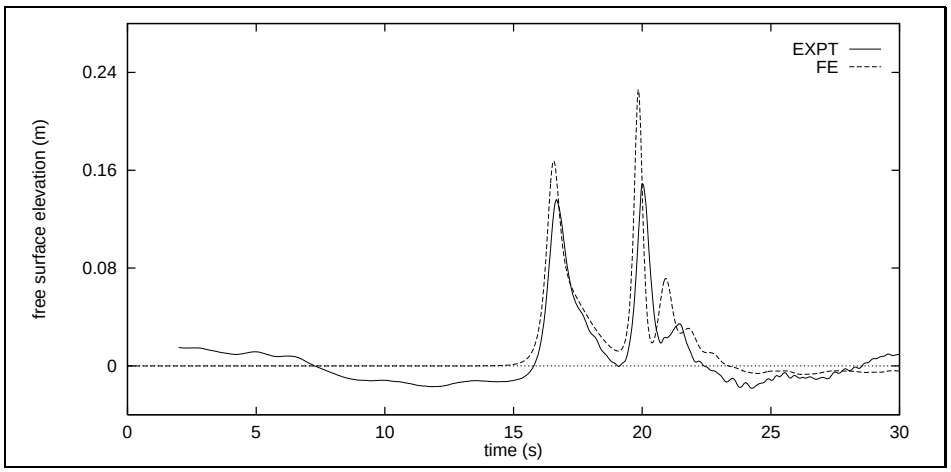


(c) Free surface time history at $x = 17.75m$

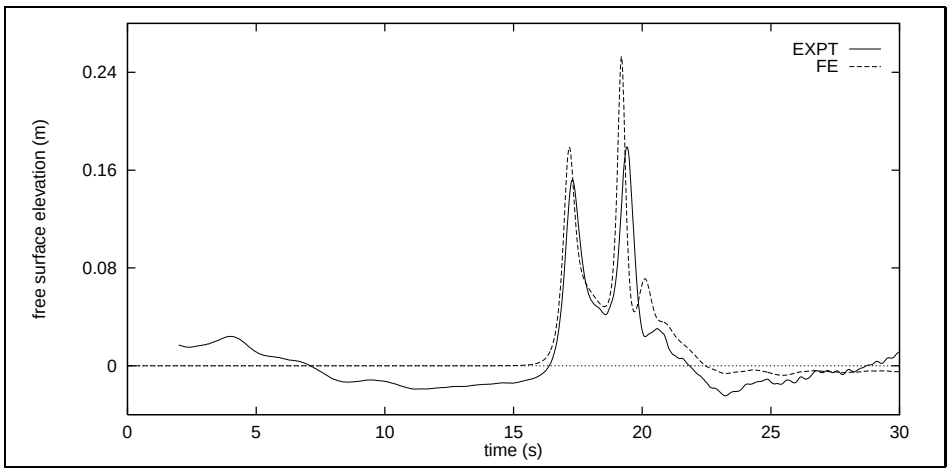
Figure 4.4: Time histories for the $0.07m$ solitary wave



(a) Free surface time history at $x = 0.0m$



(b) Free surface time history at $x = 16.25m$



(c) Free surface time history at $x = 17.75m$

Figure 4.5: Time histories for the $0.12m$ solitary wave

case.

4.9.3 A periodic deep water wave with constant depth

The extended Boussinesq system is able to model wave behaviour right up to the deep water limit $\sigma = \frac{1}{2}$ [72]. The original Boussinesq equations are not applicable at the deep water limit as their dispersion relation diverges for $\sigma > 0.48$ [72]. The test case is a periodic wave of wavelength λ=1.12m and amplitude a=0.025m propagating into an initially undisturbed region of constant depth h=0.56m. The incident wave has the form,

$$\eta(x,t) = a\sin(kx - \omega t) \tag{4.95}$$

where the wave number k is given by,

$$k = \frac{2\pi}{\lambda} \tag{4.96}$$

and the wave period τ is calculated by rearranging the linear dispersion relation (2.123) into,

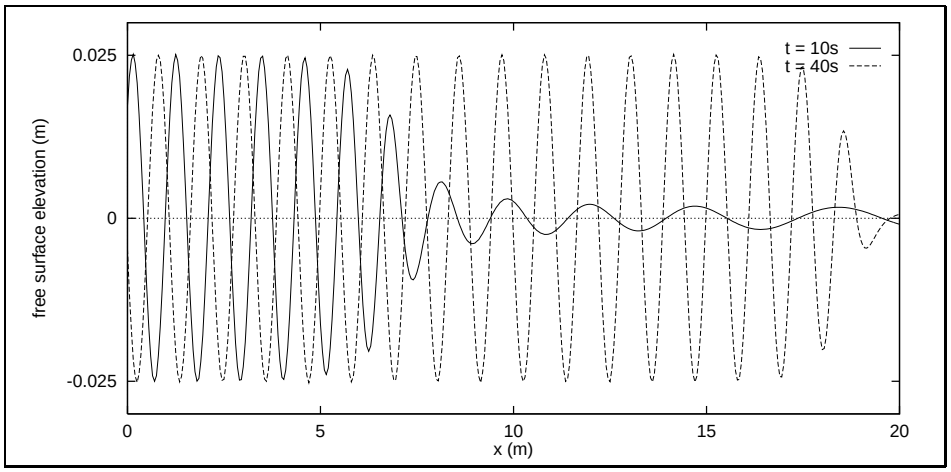
$$\omega = \sqrt{gh}\,k\sqrt{\left(\frac{1-\beta(kh)^2}{1-\alpha(kh)^2}\right)} \tag{4.97}$$

$$\tau = \frac{2\pi}{\omega} \tag{4.98}$$

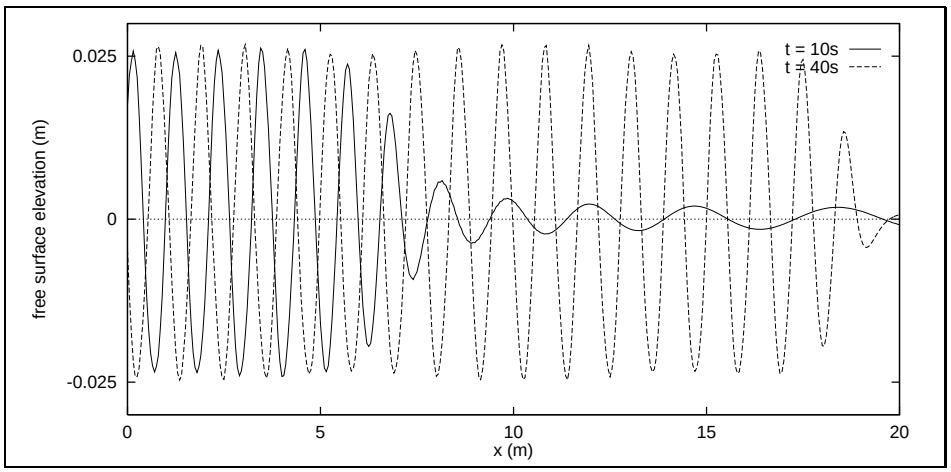
The spatial domain $x \in$ [0m,20m] is uniformly discretised with 400 linear finite elements and the problem is integrated for $t \in$ [0s,40s]. The simple linear inflow boundary condition described in Section 4.5.1 is applied at x=0m and the sponge layer boundary condition described in Section 4.5.3 is used near the outflow boundary at x=20m to prevent reflection back into the domain. The inflow wave is damped for the initial stages of the computation as described at the end of Section 4.7. This ensures that the wave is introduced smoothly into the domain.

For the first numerical experiment the linearised equations are used, in which case the solution (4.95) is exact. Fig. 4.6(a) shows the free surface profiles at t=10s and t=40s. At t=10s the wave has entered the domain at the left but is still being damped. The solution is smooth throughout the domain which suggests that this form of damping is effective here. By t=40s a steady periodic flow has been established and the sponge layer is effectively damping the outgoing wave. No reflection is apparent from the sponge layer as the wave maintains its constant amplitude inside the domain.

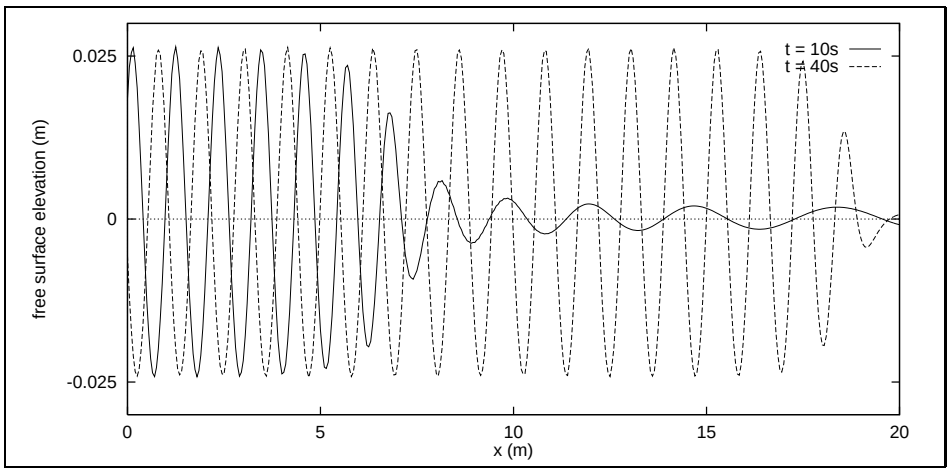
This experiment was repeated with the nonlinear equation system. Fig. 4.6(b) shows the free surface profiles at t=10s and t=40s. The inflow damping again ensures that the so-



(a) Linear equations



(b) Nonlinear equations



(c) Nonlinear equations with cnoidal wave correction

Figure 4.6: Free surface profiles for a periodic wave with constant depth

lution enters the domain smoothly as shown by the solution at t=10s. By t=40s a periodic flow has been established. However a secondary oscillation is apparent in the solution which produces a slight spatial variation in the peak amplitude. It is possible that this is caused by the nonlinearity present in the system so the experiment is repeated with the nonlinear corrections (4.47) and (4.48) added to the inflow wave, as described in Section 4.5.1.

Fig. 4.6(c) shows the free surface profiles at t=10s and t=40s for this experiment. The nonlinear corrections appear to have eliminated the secondary oscillation from the solution and there is now a steady wave profile with constant peak amplitude at t=40s.

4.9.4 A periodic wave propagating over a bar

This experiment was originally proposed by Beji and Battjes [10] as a test for their finite difference model of extended Boussinesq equations based on those of Madsen et al. [67]. A similar experiment with a finite difference model of Nwogu's extended Boussinesq equations was carried out by Ohyama et al. [74]. The geometry and wave conditions used by Beji and Battjes were also used by Dingemans [32] as a benchmark test for dispersive wave models. The experimental data from these tests is available for comparison with the numerical results. The depth profile is shown in Fig. 4.7. This case provides a severe test

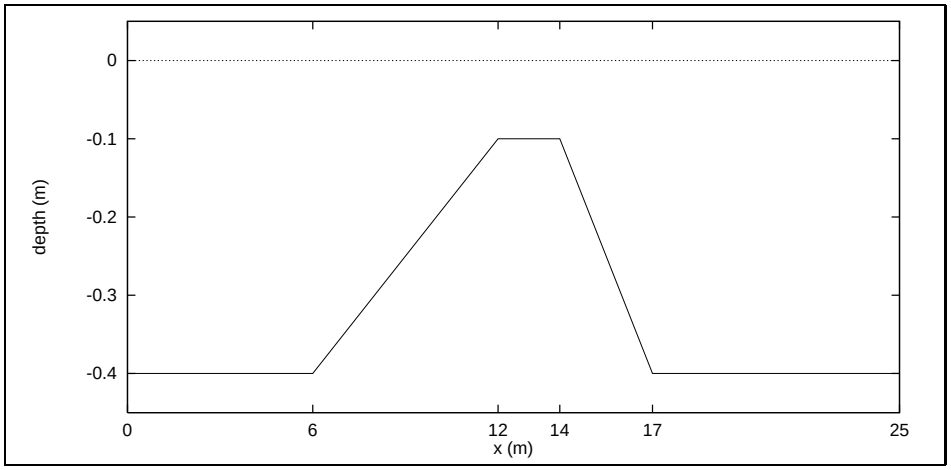


Figure 4.7: Bathymetry for the bar

of the model as nonlinearity will initially steepen the wave on the up-slope and then the increasing depth behind the bar will decompose the wave into short wave components. This will result in a rapidly varying profile behind the bar with the exact form depending crucially on the dispersive characteristics of the numerical model. Experiments with the original Boussinesq equation system [10, 74] have shown that it produces poor results in these cases due to inadequate accuracy in the dispersive terms.

For the numerical tests here the input wave is of the form given in equation (4.95) with wavelength λ=3.73m, wave period τ=2.02s and amplitude a=0.01m and propagates into an initially undisturbed region of depth h=0.40m before reaching the bar. A sponge layer is applied at the right boundary to damp the out-going wave. The spatial domain $x \in$ [0m,25m] is uniformly discretised with 500 linear finite elements and the problem is integrated in time for $t \in$ [0s,40s]. Figs. 4.8(a)-(c) compare the time histories of the free surface profile with the experimental data at x=10.5m,13.5m and 17.3m respectively. A visual comparison of the numerical results with a similar study using a finite difference method [112] shows no significant differences in the graphs.

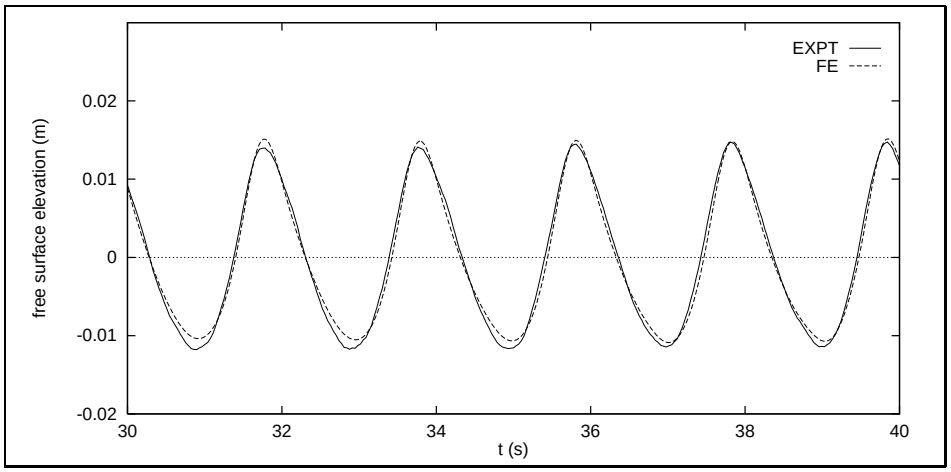
4.9.5 A non-uniform mesh

The solitary wave experiment of Section 4.9.2 is recomputed on a non-uniform grid for the $0.07m$ amplitude wave. The constant depth region $x \in$ [-55m,0m] is uniformly discretised as before with 440 linear elements, producing a mesh spacing of 0.125m. On the slope $x \in$ [0m,20m] the mesh spacing is linearly varied from the constant depth value of 0.125m at the base to 0.036m at the top of the slope. This allows a better resolution of the wave as it steepens and moves up the slope. The numerical results obtained are graphically indistinguishable from the results presented in Figs. 4.4(a)-(c). Closer inspection shows the maximum difference between the solutions to be 1.3×10^{-3}m or roughly 1%. This suggests that a suitably slow variation in the mesh may not affect the accuracy of the scheme.

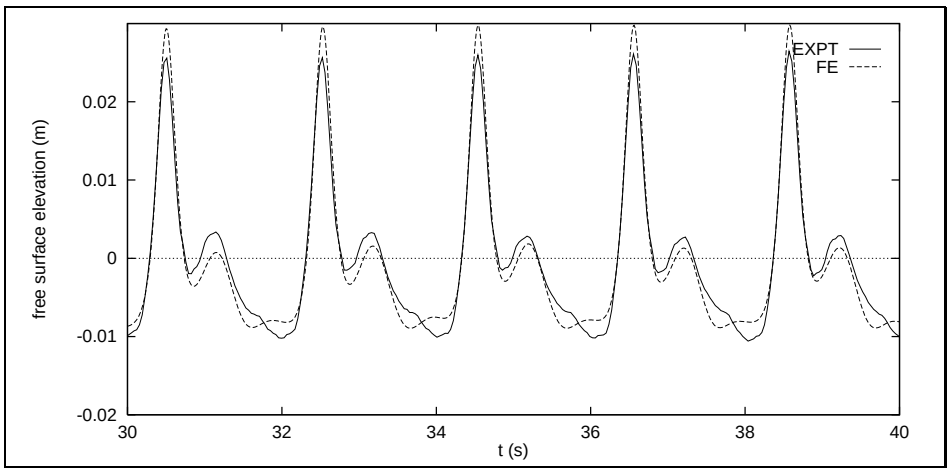
4.9.6 Internal wave generation

In Section 4.5.4 a method of generating waves internally to the domain was proposed. In this way all open boundaries can be augmented with viscous sponge layers and waves reflected from within the computational domain can be absorbed and hence removed from the problem. These numerical tests will first assess the quality of the wave that can be generated, and then a reflection experiment is performed to examine if waves can propagate through the generation region without distortion.

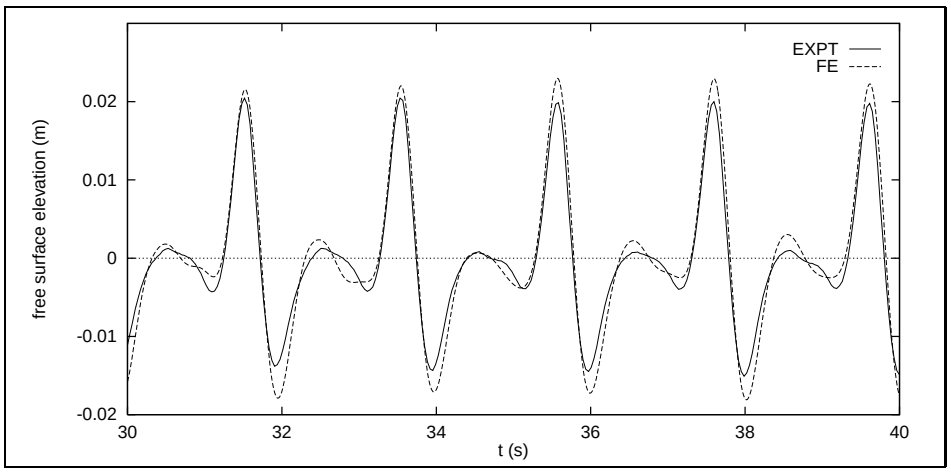
In the first case the problem data for the deep water wave considered in Section 4.9.3 is used to specify the wave. The depth is constant at 0.56m and the wave has amplitude 0.025m and wavelength 1.12m. For this first experiment the linear equation system is used, in which case the sinusoidal wave (4.95) is an exact solution. The spatial domain $x \in$ [0m,20m] is uniformly discretised with 400 linear finite elements and the problem is integrated for $t \in$ [0s,40s]. The wave is generated internally at x=6m. The generation region is $x \in$ [5m,7m] and the parameters used are $a_b = 1$, $b_b = 2$. The required wave period is calculated from equation (4.98). Sponge layers are used for the regions $x \in$ [0m,3m]



(a) Free surface time history at $x = 10.5m$



(b) Free surface time history at $x = 13.5m$



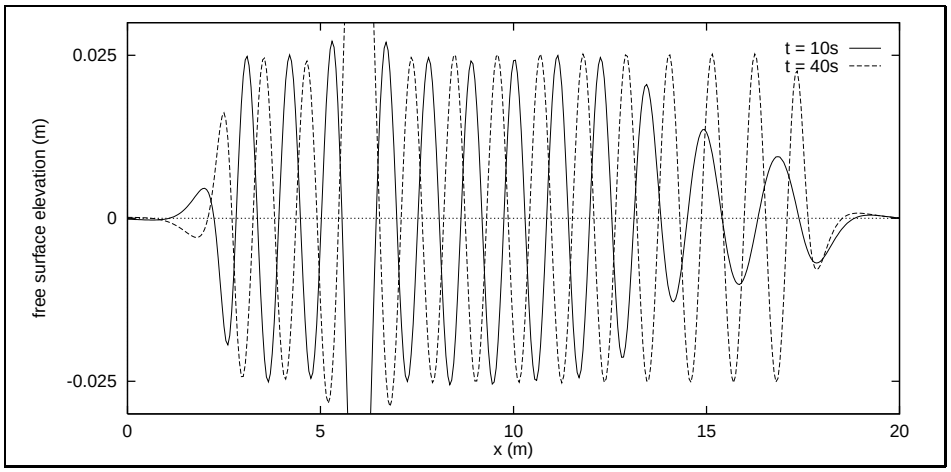
(c) Free surface time history at $x = 17.3m$

Figure 4.8: Time histories for a periodic wave over a bar

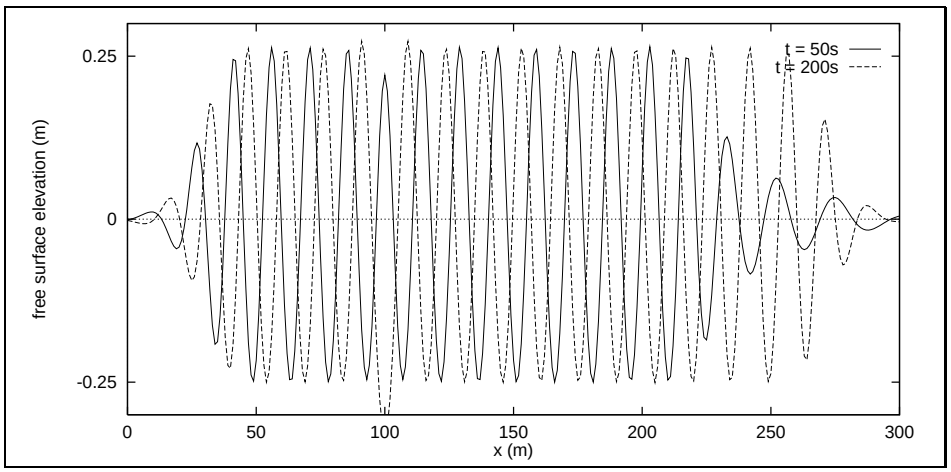
and $x \in$ [17m,20m] to absorb any waves reaching the boundary. The initial time damping of the wave is also used in this case to ensure a smooth start-up of the problem. It is found that the wave generated by this method has a larger amplitude than expected, although the period and wavelength produced are correct. Multiplying the wave generation coefficient c_I, defined by equation (4.71), by a factor of 0.5555 produces the correct amplitude. The origin of this factor is unclear at present and further work is required on this matter. Fig. 4.9(a) shows the spatial free surface profiles obtained at t=10s and t=40s. At t=10s the wave has reached the left hand boundary and is being absorbed. The right moving wave has reached the boundary but has not yet reached its peak amplitude. At t=40s the wave profile is steady and being absorbed at both boundaries. The wave amplitude, period and wavelength are accurately modelled.

The experiment was repeated with different wave parameters to assess any problem dependence of the wave generation method. The spatial domain was chosen as $x \in$ [0m,300m] and was uniformly discretised with 300 elements. A wave was generated at x = 100m with amplitude 0.25m, and wavelength 15m with a constant depth of 7.5m. The problem is integrated for $t \in$ [0s,200s]. The required wave period is calculated from equation (4.98). The wave generation region is $x \in$ [90m,110m] and the parameters used are a_b = 10, b_b = 0.25. Sponge layers are used for the regions $x \in$ [0m,30m] and $x \in$ [270m,300m], and the initial time damping is used. The experiment is first performed with the linearised equation system. It is found that the generated wave must be scaled with the same factor 0.5555 to to produce the correct amplitude. The experiment was then performed with the nonlinear equation system. Fig. 4.9(b) shows the spatial free surface profiles at t=50s and t=200s for this problem. A steady periodic wave train is established by t=200s with the waves being absorbed effectively at both boundaries. This wave does not appear to exhibit the secondary oscillations observed in Section 4.9.3; however examination of the wave parameters reveals that in this case the second order nonlinear corrections have a smaller amplitude. It is also possible that the secondary oscillation is produced by the imposition of the linear wave as a complete set of solution values at the inflow boundary. These enforced values are not an exact solution of the equations and hence will produce some perturbation. In the case of the internally generated wave the method of introduction of the wave is in some sense weaker and the solution values are not forced in any way.

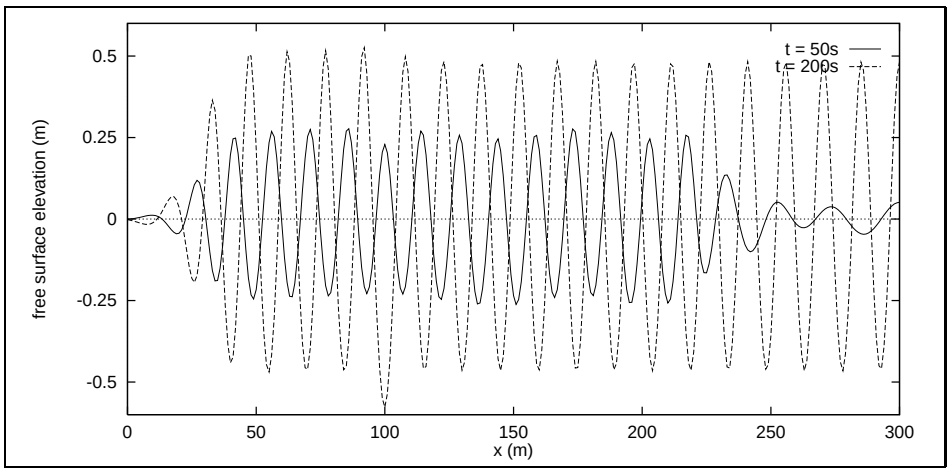
These wave parameters are then used for a further experiment in which the sponge layer boundary at x=300 is replaced with a reflecting wall. The reflection boundary conditions described in Section 4.5.2 are used at the wall. The wave generation parameters and sponge layer at the left boundary are as described in the previous experiment. The time interval $t \in$ [0,200] is long enough for the waves to have reflected from the right hand wall and reached the left hand absorbing boundary. Fig. 4.9(c) shows the spatial



(a) The 0.025m periodic wave



(b) The 0.25m periodic wave



(c) The 0.25m periodic wave reflecting from a wall

Figure 4.9: Spatial free surface profiles of an internally generated periodic wave

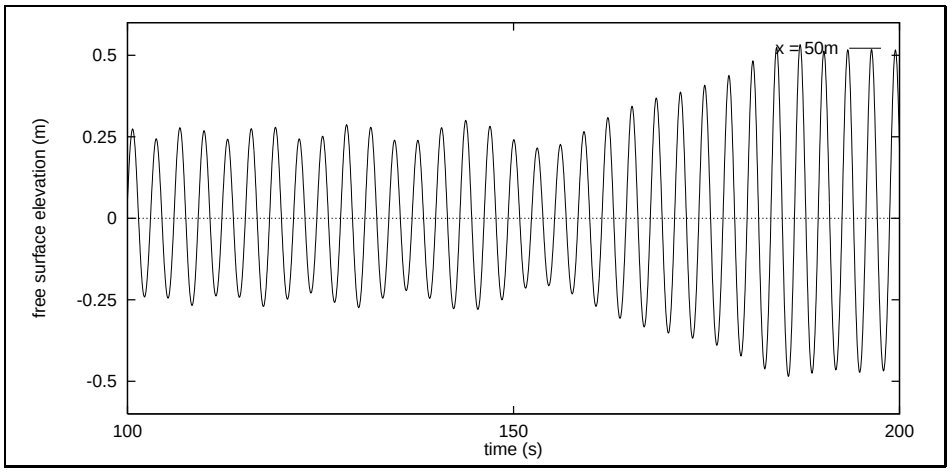
profiles of the wave at t=50s and t=200s. At t=50s the wave has reached the reflecting wall, and at t=200s the reflected wave is being absorbed at the left hand boundary and has established a steady spatial profile. Figs. 4.10(a)-(c) show time series of the free surface elevation at x=50m, 150m and 250m respectively for this problem. At t=100s the initial damped wave front is still apparent in the time histories at x=50m and x=150m and a steady periodic state is not reached until this transient has exited the domain at the left boundary. Fig. 4.10(a) shows the time history at a point to the left of the generation layer. By approximately t=180s the reflected wave has attained a steady amplitude and both it and the left moving wave produced at the wave generator are being absorbed at the left hand boundary. This also suggests that the reflected wave has passed through the generation region without significant distortion. In each case a steady profile is established suggesting that there is no unwanted reflection from the internal generation region, and that the sponge layer is effectively damping the reflected wave.

4.10 Discussion

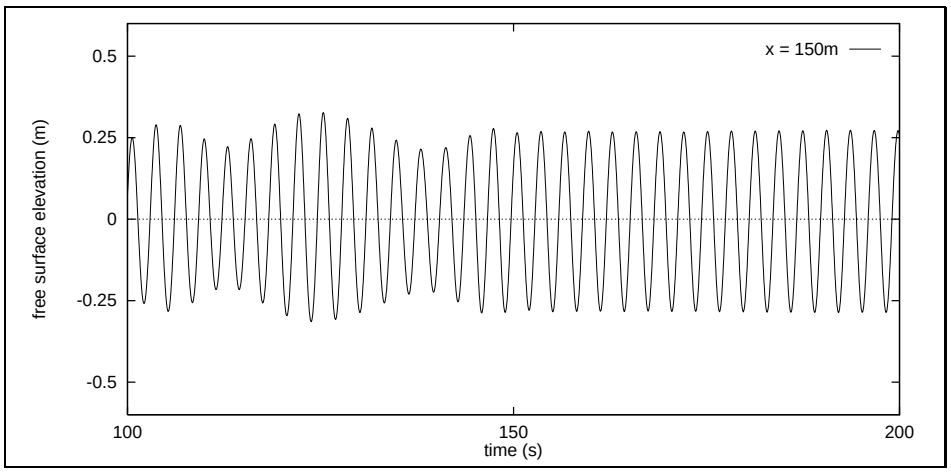
It has been demonstrated in this chapter both by analysis and by numerical experiments that the proposed linear finite element and adaptive time integration scheme can produce an accurate numerical model for Nwogu's extended Boussinesq system. The solitary wave experiment in Section 4.9.1 confirmed the accuracy of the model provided nonlinearity is weak. The higher solitary wave cases considered in Sections 4.9.1 and 4.9.2 showed that the model can become significantly inaccurate as nonlinearity increases, although the model is still robust enough to produce meaningful solutions. The deep water wave case in Section 4.9.3 illustrated the improved dispersion characteristics of the extended Boussinesq equations. In practical situations this will allow problems with a greater range of depths to be simulated than could be done with the original Boussinesq equation system.

In Section 4.5.4 a new method for the internal generation of waves was described, derived from an analogy with a time varying bed profile. It was shown in Section 4.9.6 that periodic waves could be generated internally with this method and that reflected waves could propagate through the wave generation region without distortion. Such a boundary condition is essential if realistic two-dimensional long-time simulations are to be attempted.

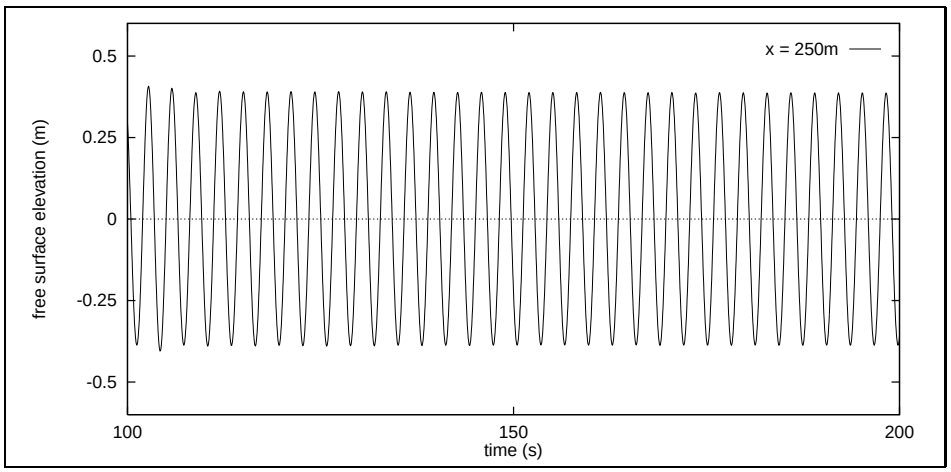
Most importantly this finite element scheme can be extended easily into two dimensions. In this case it will be shown that Nwogu's two-dimensional Boussinesq equations (2.118)-(2.119) can be rewritten in a form similar to equations (4.11)-(4.13) by introducing two auxiliary algebraic equations. In this chapter it was shown that the use of a lumped mass matrix for the auxiliary equation, and its decoupling from the time integration system made the computational cost of the finite element method comparable



(a) Time history at x = 50m



(b) Time history at x = 150m



(c) Time history at x = 250m

Figure 4.10: Time histories for the internally generated wave reflecting from a wall

with existing finite difference schemes. In two dimensions it will be shown that these approximations and the savings in computational cost will be even more significant. The truncation error analysis of the one-dimensional system in Section 4.8 showed that these approximations did not significantly affect the accuracy of the method. This truncation error analysis also depended on the nonlinear terms being written in the compact forms (4.3) and (4.4). An equivalent form for the two-dimensional equations is presented in the following chapter. The truncation error analysis suggests that good results may be expected on regular two-dimensional grids and that if the variation of the mesh is suitably controlled that the method will extend successfully to unstructured triangular grids.

Chapter 5

An Algorithm for the Two-dimensional Extended Boussinesq Equations

5.1 Introduction

In this chapter the finite element method developed for Nwogu's one-dimensional Boussinesq equation system is extended to the two-dimensional form of the equations, representing the most general form of the equations to be considered here. The previously published work on finite difference and finite element methods for the two-dimensional Boussinesq equations is reviewed in Section 5.2, and it is noted that the time-domain solution of two-dimensional extended Boussinesq equations has, until recently, been restricted to finite difference methods. There have been several reported finite element methods for equation systems equivalent to the original Boussinesq equation system and, recently, a method published for Beji and Nadaoka's extended Boussinesq equation system. However, to the author's knowledge, there have been no attempts to apply the finite element method to realistic, geometrically complex problems.

The extended Boussinesq equations as derived by Nwogu are presented in Section 5.3. Some additional work is required to write the two-dimensional equations in a form suitable for the linear finite element approximation developed in this work, and in the rest of the section it is shown how this can be achieved within the order of accuracy of the Boussinesq mathematical model.

The two-dimensional linear triangular finite element method used for the spatial discretisation of these equations is introduced in Section 5.4. The complete spatial discreti-

sation of the equation system and the approximations necessary within this process are presented. In Section 5.5 suitable boundary conditions and their numerical implementation are discussed. As in the case of the one-dimensional algorithm, particular emphasis is placed on the implementation of the boundary conditions such that the time integration may be performed efficiently. The techniques developed for the one-dimensional extended Boussinesq equations in Sections 4.5 are extended to two dimensions with details given of the two-dimensional effects not considered in that section. In Section 5.5.5 the new method for the internal generation of waves, introduced for the one-dimensional equations in Section 4.5.4, is extended to the two-dimensional equations.

The time integration of the spatially discretised system is considered in Section 5.6. In two-dimensions the finite element method will typically produce a sparse matrix system, and the banded matrix solution techniques within the time integration process, described in the previous chapter, will not be efficient. The time integration packages SPRINT and DASPK both contain the facility for sparse matrix handling. Direct solution of the sparse equation system, by techniques such as those contained within the SPRINT package, involves the numerical factorisation of a sparse matrix and can become expensive both in terms of computational storage and cost for large equation systems. It is shown in Section 5.6.2.1 that the equations considered here exaggerate this problem by producing a system with a large bandwidth due to the introduction of the auxiliary variables. This increases the potential size of the matrix factorisation and hence the computational storage and cost required. It is possible to increase the efficiency of the solution methods by reordering the matrix system in various ways. Some of these techniques are reported in Section 5.6.2.2 and their application to the equation system considered here is discussed. An alternative to the direct solution methods used previously is to solve the matrix system iteratively. The SPRINT software has been combined with the iterative methods contained in the WATSIT [105] package which implement the preconditioned iterative method ORTHOMIN. The DASPK software contains routines from the SPARSKIT [25] package which implement the iterative method GMRES. The particular advantages and disadvantages of these methods are discussed in Section 5.6.2.3. The evaluation and factorisation of the Jacobian matrix is a computationally expensive procedure for the sparse matrix systems produced in two dimensions. In Section 5.6.2.4 the options for its calculation and an evaluation of SPRINT and DASPK in this regard is presented. Both of the time integration packages are investigated here for the solution of the spatially discretised problem and a discussion of their overall performance is detailed in Section 5.6.3.

The specification of initial conditions for the problem is considered in Section 5.7. Issues additional to those discussed for the one-dimensional algorithm in Section 4.7 are considered in detail.

Numerical experiments are described in Section 5.8. The algorithm is tested with a

variety of constant depth and variable depth cases, including some standard test cases for dispersive wave models. Results are computed using both structured and fully unstructured triangular meshes and the accuracy and efficiency of the scheme is considered in detail. In Section 5.9 the results are reviewed and the suitability of the proposed method for two-dimensional simulations is discussed.

5.2 Previous work

The first two-dimensional time domain simulations were for finite difference models of the original Boussinesq equation system. Abbot et al. [2] developed a finite difference method for the original Boussinesq equations where spatial terms were differenced to second order accuracy and corrections added in to account for any non-physical dispersion or diffusion produced by the scheme. This scheme was probably the first Boussinesq model to be used as an engineering tool and an application to a real harbour geometry was reported in this work. Hauguel [47] presented a finite difference scheme using a high order characteristic type method for the convection terms coupled with a three stage difference scheme. An example was presented of this model applied to a harbour geometry. Smallman and Cooper [98] used a predictor-corrector time integration scheme and second order spatial differences to investigate set-down in harbours.

More recently there have been several finite difference methods presented for extended Boussinesq equation systems. Madsen et al. [66] presented a finite difference scheme for their constant depth extended Boussinesq equations. They extended the difference scheme of Abbot et al. [1] to these equations and to the variable depth equation system in a subsequent publication [67]. Nwogu [73] applied a similar difference scheme to his extended equations. This was used to investigate nonlinear interactions of irregular multidirectional waves. Wei and Kirby [106] extended their finite difference model of Nwogu's extended equations and compared their results with some standard experimental test cases. These will be discussed in more detail in Section 5.8. The accuracy of this model was confirmed in further work by Zang et al. [112], and more recently in the work of Skotner and Apelt [97] who applied this model to a fictitious harbour geometry. Schröter et al. [95] applied a finite difference method, similar in principle to that of Abbot et al. [2], to another set of extended Boussinesq equations. This model was used to simulate the wave disturbances in a real harbour geometry, reported in a separate publication [94]. Beji and Nadaoka [11] applied a three-level time centred scheme and centred finite differences for their extended equations. This method was validated with some standard test problems.

The application of finite difference methods to the Boussinesq equations on irregular spatial domains has typically been handled by approximating the domain as a series of steps, so that a regularly spaced discretisation is maintained [2, 94]. In practice these

boundaries may produce spurious noise and the application of smoothing to the solution has been reported to alleviate these problems [94]. Modifications to the finite difference stencils are also required at points near the wall and on irregular domains there are a large number of possible cases.

The application of the finite element method has until recently been limited to the original Boussinesq equations, or equivalent systems of the same order of accuracy. Katapodes and Wu [56] developed one of the first finite element methods for the constant depth equations. This was a Galerkin method using bilinear rectangular elements and a Lax-Wendroff type discretisation in time. They presented computations of solitary waves on structured rectangular grids. Antunes Do Carmo et al. [8] developed a Galerkin finite element method with quadrilateral rectangular elements for the variable depth equations and used it to model the flow of a solitary wave past a cylindrical obstruction in a channel. Kawahara and Cheng [57] developed a linear triangular Galerkin finite element method for the variable depth equations coupled with a simple explicit approximation in time. This was applied to a model of a bay using a fully unstructured triangular grid. Langtangen and Pedersen [58] developed Galerkin finite element methods based on linear triangles and biquadratic rectangles within the framework of the DIFFPACK software [29]. They used a staggered time integration scheme which allowed them to solve each equation sequentially rather than in coupled form. In a subsequent paper [59] they included correction terms to increase the time accuracy and to try and eliminate any non-physical dispersion. They showed that unstructured grids adapted to the depth profile could be used to solve problems with far greater efficiency than structured grids, although they found the biquadratic rectangular elements to be generally superior to triangular ones. Ambrosi and Quartepelle [6] applied a modified linear Taylor-Galerkin method to the Boussinesq equations. They used this scheme to model the flow of a solitary wave past a cylindrical obstruction in a channel using unstructured triangular grids. At the time of the development of the numerical model used in this project there were, to the authors knowledge, no published finite element methods for extended Boussinesq equations; however Li et al. [62] have recently presented a bilinear quadrilateral finite element method for Beji and Nadaoka's extended Boussinesq equations. This was applied to several standard test cases for dispersive waves. As mentioned in the previous chapter there are some similarities with the method described here and they are compared and contrasted in this chapter.

Although the finite element method will straightforwardly extend to irregular domains the only work reported to the author's knowledge on such an application is that of Kawahara and Cheng [57] who modelled wave propagation into a real bay.

5.3 The two-dimensional extended Boussinesq equation system

Nwogu's two-dimensional extended Boussinesq equations [72] can be derived from the three-dimensional incompressible, irrotational Euler equations in a similar manner to that presented in Section 2.4.3. Both horizontal coordinates are nondimensionalised and scaled in an identical manner to the one-dimensional coordinate x. Here the equations are written in terms of the free surface elevation $\eta(x,y,t)$, horizontal velocity field $\mathbf{u}(x,y,t) = (u(x,y,t),\, v(x,y,t))$ at depth $z = \theta h$ and spatially varying depth $h(x,y)$ with respect to a horizontal coordinate system (x,y).

$$\frac{\partial \eta}{\partial t} + \nabla\cdot((h+\eta)\,\mathbf{u}) + \nabla\cdot\left(B_1 h^3 \nabla(\nabla\cdot\mathbf{u}) + B_2 h^2 \nabla(\nabla\cdot(h\mathbf{u}))\right) = 0 \qquad (5.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + g\nabla\eta + (\mathbf{u}\cdot\nabla)\,\mathbf{u} + A_1 h^2 \nabla\left(\nabla\cdot\frac{\partial \mathbf{u}}{\partial t}\right) + A_2 h \nabla\left(\nabla\cdot\left(h\frac{\partial \mathbf{u}}{\partial t}\right)\right) = 0 \qquad (5.2)$$

where the coefficients A_1, A_2, B_1 and B_2 are defined by equations (2.110)-(2.113) and depend on the free parameter θ. The two-dimensional vector differential operator ∇ is defined by,

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right) \qquad (5.3)$$

In order to apply the linear finite element method developed in the last chapter the third spatial derivatives must be removed from the system. Due to the structure of the free surface equation (5.1) this can be achieved by defining a vector $\mathbf{w}$ as follows,

$$\mathbf{w} = B_1 h^3 \nabla(\nabla\cdot\mathbf{u}) + B_2 h^2 \nabla(\nabla\cdot(h\mathbf{u})) \qquad (5.4)$$

Using this equation to rewrite equation (5.1) in a lower order form introduces two additional differential equations into the system.

Li et al. [62] rewrite Beji and Nadaoka's extended Boussinesq equation system in lower order form by introducing the gradient of the free surface as an additional variable in a similar manner.

The two-dimensional nonlinear convection terms in the velocity equation (5.2) are expressed in a form that cannot be directly rewritten in a compact manner as in one dimension. However it is possible to approximate these terms within the order of accuracy of the Boussinesq equations to arrive at a compact form. Considering the nonlinear con-

vection term from the first velocity component u of equation (5.2),

$$\begin{aligned} u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y} &= \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right)+v\frac{\partial v}{\partial x}-v\frac{\partial v}{\partial x}+v\frac{\partial u}{\partial y} \\ &= \frac{\partial}{\partial x}\left(\frac{1}{2}\left(u^2+v^2\right)\right)+v\left(\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x}\right) \end{aligned} \tag{5.5}$$

From the assumption of irrotationality made in the derivation of the Boussinesq mathematical model in Chapter 2 the true three-dimensional velocity field,

$$\mathbf{U}(x,y,z)=(U,V,W) \tag{5.6}$$

satisfies the irrotationality condition,

$$\frac{\partial U}{\partial y}-\frac{\partial V}{\partial x} = 0 \tag{5.7}$$

In Section 2.4.3 Nwogu's scaled, nondimensionalised velocity variable $\hat{u}(\hat{x},\hat{t})$ was shown to be an $\mathscr{O}(\sigma^2)$ perturbation of the true velocity field $\hat{U}(\hat{x},\tilde{z},\hat{t})$. Hence in the scaled, nondimensionalised Boussinesq framework,

$$\begin{aligned} \frac{\partial \hat{u}}{\partial \hat{y}}-\frac{\partial \hat{v}}{\partial \hat{x}} &= \frac{\partial \hat{U}}{\partial \hat{y}}-\frac{\partial \hat{V}}{\partial \hat{x}}+\mathscr{O}(\sigma^2) \\ &= \mathscr{O}(\sigma^2) \end{aligned} \tag{5.8}$$

Using this to rewrite the nonlinear convection terms in equation (5.5),

$$\hat{u}\frac{\partial \hat{u}}{\partial \hat{x}}+\hat{v}\frac{\partial \hat{u}}{\partial \hat{y}} = \frac{\partial}{\partial \hat{x}}\left(\frac{1}{2}\left(\hat{u}^2+\hat{v}^2\right)\right)+\mathscr{O}(\sigma^2) \tag{5.9}$$

Similarly the nonlinear convection term from the second velocity component $\hat{v}$ can be rewritten as,

$$\begin{aligned} \hat{u}\frac{\partial \hat{v}}{\partial \hat{x}}+\hat{v}\frac{\partial \hat{v}}{\partial \hat{y}} &= \frac{\partial}{\partial \hat{y}}\left(\frac{1}{2}\left(\hat{u}^2+\hat{v}^2\right)\right)-\hat{u}\left(\frac{\partial \hat{u}}{\partial \hat{y}}-\frac{\partial \hat{v}}{\partial \hat{x}}\right) \\ &= \frac{\partial}{\partial \hat{y}}\left(\frac{1}{2}\left(\hat{u}^2+\hat{v}^2\right)\right)+\mathscr{O}(\sigma^2) \end{aligned} \tag{5.10}$$

Equation (2.101) in Section 2.4.3 shows that within the Boussinesq mathematical model the nonlinear terms are already $\mathscr{O}(\varepsilon)$ hence the approximations (5.9) and (5.10) are $\mathscr{O}(\varepsilon\sigma^2)$ perturbations to the equation system which is within the overall order of accuracy of the Boussinesq equations. This rewriting of the convection terms was used by Beji and Nadaoka [11] in the development of a finite difference method for their extended equation

system, although they stated it without any explanation.

The two-dimensional equations contain dispersive terms in the form of cross spatial derivatives, as well as mixed space and time derivatives. It is possible to rewrite the dispersion terms by rearranging the order of differentiation, ie.

$$\nabla\left(\nabla\cdot(u,v)\right) \equiv \left(\nabla\cdot\frac{\partial \mathbf{u}}{\partial x}, \nabla\cdot\frac{\partial \mathbf{u}}{\partial y}\right) \tag{5.11}$$

However consideration of the boundary conditions in Section 5.5.3 shows that the form in equations (5.1)-(5.4) is the most appropriate for the numerical methods developed in this work.

Using these approximations and the notation (3.3) for the time derivative, the equation system (5.1)-(5.2) can be written as,

$$\begin{aligned}
\dot{\eta} + \nabla\cdot(p,q) + \nabla\cdot\mathbf{w} &= 0 && (5.12)\\
\dot{u} + \frac{\partial f}{\partial x} + A_1 h^2 \frac{\partial}{\partial x}\left(\nabla\cdot\dot{\mathbf{u}}\right) + A_2 h \frac{\partial}{\partial x}\left(\nabla\cdot(h\dot{\mathbf{u}})\right) &= 0 && (5.13)\\
\dot{v} + \frac{\partial f}{\partial y} + A_1 h^2 \frac{\partial}{\partial y}\left(\nabla\cdot\dot{\mathbf{u}}\right) + A_2 h \frac{\partial}{\partial y}\left(\nabla\cdot(h\dot{\mathbf{u}})\right) &= 0 && (5.14)\\
w_1 - B_1 h^3 \frac{\partial}{\partial x}\left(\nabla\cdot\mathbf{u}\right) - B_2 h^2 \frac{\partial}{\partial x}\left(\nabla\cdot(h\mathbf{u})\right) &= 0 && (5.15)\\
w_2 - B_1 h^3 \frac{\partial}{\partial y}\left(\nabla\cdot\mathbf{u}\right) - B_2 h^2 \frac{\partial}{\partial y}\left(\nabla\cdot(h\mathbf{u})\right) &= 0 && (5.16)
\end{aligned}$$

where,

$$\begin{aligned}
f &= \frac{1}{2}\left(u^2+v^2\right) + g\eta && (5.17)\\
p &= (h+\eta)\,u && (5.18)\\
q &= (h+\eta)\,v && (5.19)
\end{aligned}$$

The assumption of a constant depth, H, simplifies the equations slightly to,

$$\begin{aligned}
\dot{\eta} + \nabla\cdot(p,q) + \nabla\cdot\mathbf{w} &= 0 && (5.20)\\
\dot{u} + \frac{\partial f}{\partial x} + \alpha H^2 \frac{\partial}{\partial x}\left(\nabla\cdot\dot{\mathbf{u}}\right) &= 0 && (5.21)\\
\dot{v} + \frac{\partial f}{\partial y} + \alpha H^2 \frac{\partial}{\partial y}\left(\nabla\cdot\dot{\mathbf{u}}\right) &= 0 && (5.22)\\
w_1 - \beta H^3 \frac{\partial}{\partial x}\left(\nabla\cdot\mathbf{u}\right) &= 0 && (5.23)\\
w_2 - \beta H^3 \frac{\partial}{\partial y}\left(\nabla\cdot\mathbf{u}\right) &= 0 && (5.24)
\end{aligned}$$

where α and β are defined by equations (2.116) and (2.117).

Suitable boundary conditions and initial conditions for these equations and their numerical approximations are considered in detail in Sections 5.5 and 5.7 respectively.

5.4 Spatial discretisation

5.4.1 The two-dimensional finite element method

In two-dimensions the finite element method is developed by first partitioning the spatial domain into a set of *ne* non-overlapping elements that completely cover the domain. In this work triangular elements are used. It is possible to use general polygonal elements [113] but the use of triangular elements allows advantage to be taken of the many software packages available for representing arbitrary spatial domains with triangles. A piecewise linear interpolation can be defined over this set of triangular elements by using the common vertices to define a set of *np* nodes. Higher degrees of polynomial interpolation can be expressed by including nodal points on the common edges of the triangles and interior to the triangular elements [113]. A set of *np* linear global basis functions $\phi_i(x,y)$ are then defined on the set of nodes by,

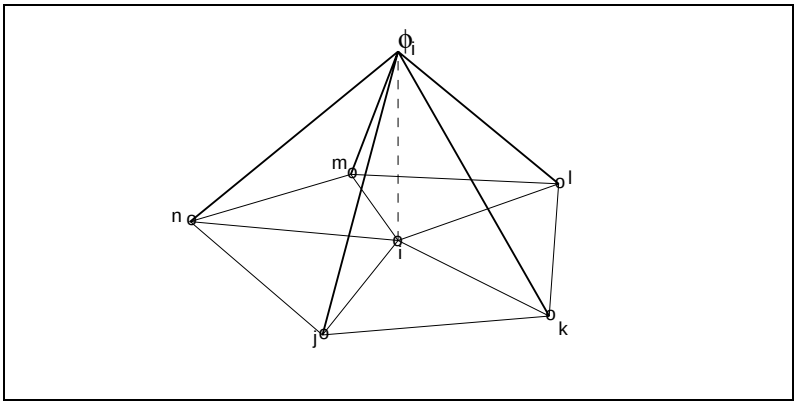
$$\phi_i(x_j,y_j) \;=\; \delta_{ij} \tag{5.25}$$

where δ_{ij} is the Dirac delta function. A typical global basis function is illustrated in Figure 5.1(a), which shows that the interpolation function is non-zero only over the elements immediately surrounding the point. On an element there will be three non-zero local basis functions, with a typical one illustrated by Figure 5.1(b), and these are used to define a linear interpolation over the element. By the definition of the interpolation at the nodes an unknown function is interpolated in a piecewise linear fashion over the whole set of triangles, and clearly $\phi_i(x,y) \in \mathscr{H}^1$, where the definition of $\mathscr{H}^1$ (3.18) is extended to two dimensions by requiring that both first spatial derivatives belong to L^2. Higher degrees of continuity are possible in two dimensions [100] but are not considered in this work.

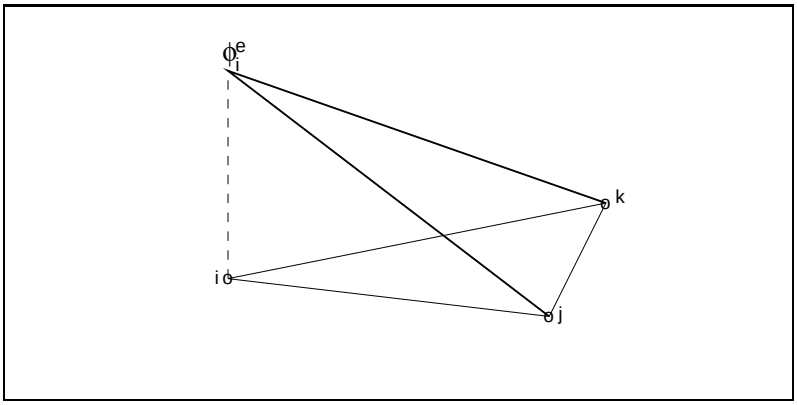
5.4.2 The finite element spatial discretisation

Nwogu's two-dimensional extended Boussinesq equations are discretised in space with a linear triangular Galerkin finite element method. For clarity of presentation the constant depth equation system (5.20)-(5.24) is considered first. Afterwards the discretisation procedure for a typical dispersive term of the variable depth system (5.12)-(5.16) is considered.

The spatial domain Ω is partitioned into a set of *ne* non-overlapping triangular elements that completely cover the domain. The vertices of these triangles define a set of *np*



(a) The global basis function at node *i*



(b) The local basis function at node *i* on element *e*

Figure 5.1: The linear triangular finite element basis function

nodes. A set of linear basis functions $\{\phi_j,\ j=1,\ldots,np\}$ is defined at the nodes of the mesh, as described in the previous section.

To develop the Galerkin finite element method, equations (5.20)-(5.24) are multiplied by a test function $\phi_i(x,y)$ from the set of basis functions and integrated over the spatial domain Ω. The notation (3.33) is used for the spatial integrals.

$$
\begin{aligned}
(\phi_i,\dot{\eta})+(\phi_i,\nabla\cdot(p,q))+(\phi_i,\nabla\cdot\mathbf{w}) &= 0 && (5.26)\\
(\phi_i,\dot{u})+\left(\phi_i,\frac{\partial f}{\partial x}\right)+\alpha H^2\left(\phi_i,\frac{\partial}{\partial x}(\nabla\cdot\dot{\mathbf{u}})\right) &= 0 && (5.27)\\
(\phi_i,\dot{v})+\left(\phi_i,\frac{\partial f}{\partial y}\right)+\alpha H^2\left(\phi_i,\frac{\partial}{\partial y}(\nabla\cdot\dot{\mathbf{u}})\right) &= 0 && (5.28)\\
(\phi_i,w_1)-\beta H^3\left(\phi_i,\frac{\partial}{\partial x}(\nabla\cdot\mathbf{u})\right) &= 0 && (5.29)\\
(\phi_i,w_2)-\beta H^3\left(\phi_i,\frac{\partial}{\partial y}(\nabla\cdot\mathbf{u})\right) &= 0 && (5.30)
\end{aligned}
$$

Integration by parts is used to reduce the order of the second spatial derivatives, introducing integrals on the boundary Γ.

$$
\begin{aligned}
(\phi_i,\dot{\eta})+(\phi_i,\nabla\cdot(p,q))+(\phi_i,\nabla\cdot\mathbf{w}) &= 0 && (5.31)\\
(\phi_i,\dot{u})-\left(\frac{\partial\phi_i}{\partial x},f\right)-\alpha H^2\left(\frac{\partial\phi_i}{\partial x},\nabla\cdot\dot{\mathbf{u}}\right) &= \\
&\quad -\int_\Gamma \phi_i\left(f+\alpha H^2\nabla\cdot\dot{\mathbf{u}}\right)n_x\,d\Gamma && (5.32)\\
(\phi_i,\dot{v})-\left(\frac{\partial\phi_i}{\partial y},f\right)-\alpha H^2\left(\frac{\partial\phi_i}{\partial y},\nabla\cdot\dot{\mathbf{u}}\right) &= \\
&\quad -\int_\Gamma \phi_i\left(f+\alpha H^2\nabla\cdot\dot{\mathbf{u}}\right)n_y\,d\Gamma && (5.33)\\
(\phi_i,w_1)+\beta H^3\left(\frac{\partial\phi_i}{\partial x},\nabla\cdot\mathbf{u}\right) &= \beta H^3\int_\Gamma \phi_i\,(\nabla\cdot\mathbf{u})\,n_x\,d\Gamma && (5.34)\\
(\phi_i,w_2)+\beta H^3\left(\frac{\partial\phi_i}{\partial y},\nabla\cdot\mathbf{u}\right) &= \beta H^3\int_\Gamma \phi_i\,(\nabla\cdot\mathbf{u})\,n_y\,d\Gamma && (5.35)
\end{aligned}
$$

where $\mathbf{n}=(n_x,n_y)$ is the outward unit normal to the boundary Γ. Introducing approximations for the continuous function η, u, v, p, q and f over the mesh in the form,

$$
F(x,y,t)\ \approx\ \sum_{j=1}^{np}\phi_j(x,y)\,F_j(t) \qquad (5.36)
$$

the discrete system can be written,

$$
M_{ij}\dot{\eta}_j+C^1_{ji}p_j+C^2_{ji}q_j+C^1_{ji}(w_1)_j+C^2_{ji}(w_2)_j = 0 \qquad (5.37)
$$

$$M_{ij}\dot{u}_j - C^1_{ij} f_j - \alpha H^2 \left(K^{11}_{ij}\dot{u}_j + K^{12}_{ij}\dot{v}_j\right) = - \int_\Gamma \phi_i \left(f + \alpha H^2 \nabla\cdot\dot{\mathbf{u}}\right) n_x \, d\Gamma \tag{5.38}$$

$$M_{ij}\dot{v}_j - C^2_{ij} f_j - \alpha H^2 \left(K^{21}_{ij}\dot{u}_j + K^{22}_{ij}\dot{v}_j\right) = - \int_\Gamma \phi_i \left(f + \alpha H^2 \nabla\cdot\dot{\mathbf{u}}\right) n_y \, d\Gamma \tag{5.39}$$

$$M_{ij}(w_1)_j + \beta H^3 \left(K^{11}_{ij} u_j + K^{12}_{ij} v_j\right) = \beta H^3 \int_\Gamma \phi_i \left(\nabla\cdot\mathbf{u}\right) n_x \, d\Gamma \tag{5.40}$$

$$M_{ij}(w_2)_j + \beta H^3 \left(K^{21}_{ij} u_j + K^{22}_{ij} v_j\right) = \beta H^3 \int_\Gamma \phi_i \left(\nabla\cdot\mathbf{u}\right) n_y \, d\Gamma \tag{5.41}$$

where summation over the repeated index j is implied, and,

$$M_{ij} = \left(\phi_i, \phi_j\right) \tag{5.42}$$

$$C^m_{ij} = \left(\frac{\partial \phi_i}{\partial x_m}, \phi_j\right) \tag{5.43}$$

$$K^{mn}_{ij} = \left(\frac{\partial \phi_i}{\partial x_m}, \frac{\partial \phi_j}{\partial x_n}\right) \tag{5.44}$$

with $(x_1, x_2) \equiv (x, y)$. In this form, and together with the appropriate boundary conditions and initial conditions, the discrete equations represent a system of $5np$ differential-algebraic equations to be solved within the time integration process. For the one-dimensional equations the computational expense was reduced by approximating the spatially discretised auxiliary equation (4.26) in a more explicit form (4.30) by lumping the mass matrix M. Here a similar approximation is made by approximating the mass matrix M_{ij} in a similar lumped, diagonal form for the calculation of the auxiliary variables w_1 and w_2. This replaces equations (5.40) and (5.41) with,

$$(w_1)_i = \frac{\beta H^3}{M^L_i} \left(- \left(K^{11}_{ij} u_j + K^{12}_{ij} v_j\right) + \int_\Gamma \phi_i \left(\nabla\cdot\mathbf{u}\right) n_x \, d\Gamma \right) \tag{5.45}$$

$$(w_2)_i = \frac{\beta H^3}{M^L_i} \left(- \left(K^{21}_{ij} u_j + K^{22}_{ij} v_j\right) + \int_\Gamma \phi_i \left(\nabla\cdot\mathbf{u}\right) n_y \, d\Gamma \right) \tag{5.46}$$

where,

$$M^L_i = \sum_{j=1}^{np} M_{ij} \tag{5.47}$$

The w_1 and w_2 equations are decoupled from the differential-algebraic system and are calculated explicitly before the differential equations (5.37)-(5.39) are assembled. This

reduces the size of the system solved by the time integration software from $5np$ to $3np$. Note that in this case the original discretised equations (5.40) and (5.41) are sparse matrix systems for the nodal values $(w_1)_i$ and $(w_2)_i$ and hence the reduction to a diagonal form represents a considerable computational saving in itself.

The two-dimensional variable depth equations (5.12)-(5.16) represent the most general equation system to be considered here. The only difference with the constant depth equations (5.20)-(5.24) is the more general form of the dispersive terms, which include the effects of a spatially varying depth $h(x,y)$. The dispersive terms can be discretised in a compact form analogous to that which was described for the one-dimensional equations in Section 4.4.2. This is described for a typical dispersive term from equation (5.15) below.

$$\begin{aligned}
\left(\phi_i, B_2 h^2 \frac{\partial}{\partial x}\left(\nabla\cdot(h\mathbf{u})\right)\right) &= B_2\left(\phi_i h^2, \frac{\partial}{\partial x}\left(\nabla\cdot(h\mathbf{u})\right)\right) \\
&= -B_2\left(\frac{\partial}{\partial x}\left(\phi_i h^2\right), \nabla\cdot(h\mathbf{u})\right) \\
&\qquad + B_2\int_\Gamma \phi_i h^2 \nabla\cdot(h\mathbf{u})\, n_x \, d\Gamma \\
&= -B_2\left(\frac{\partial \phi_i}{\partial x} h^2 + \phi_i \frac{\partial}{\partial x}\left(h^2\right), \nabla\cdot(h\mathbf{u})\right) \\
&\qquad + B_2\int_\Gamma \phi_i h^2 \nabla\cdot(h\mathbf{u})\, n_x \, d\Gamma
\end{aligned} \tag{5.48}$$

The unknown functions are compactly approximated over the mesh in a similar manner to the approximation of the nonlinear functions p, q and f, by equation (5.36).

$$h^2 \approx \sum_{j=1}^{np} \phi_j \left(h^2\right)_j \tag{5.49}$$

$$hu \approx \sum_{j=1}^{np} \phi_j \left(hu\right)_j \tag{5.50}$$

$$hv \approx \sum_{j=1}^{np} \phi_j \left(hv\right)_j \tag{5.51}$$

Inserting these approximations into (5.48), the finite element approximation of the dispersive term is,

$$\begin{aligned}
\left(\phi_i, B_2 h^2 \frac{\partial}{\partial x}\left(\nabla\cdot(h\mathbf{u})\right)\right) &\approx -B_2\left(\left(\frac{\partial \phi_i}{\partial x}\phi_k + \phi_i \frac{\partial \phi_k}{\partial x}\right) h_k^2, \frac{\partial \phi_j}{\partial x} h_j u_j + \frac{\partial \phi_j}{\partial y} h_j v_j\right) \\
&\qquad + B_2\int_\Gamma \phi_i h^2 \nabla\cdot(h\mathbf{u})\, n_x \, d\Gamma
\end{aligned} \tag{5.52}$$

where summation over the repeated indices is implied. A similar procedure can be used

to approximate all of the dispersive terms in equations (5.12)-(5.16).

Li et al. [62] use a similar lumped approximation for their auxiliary variables; however in discretising the variable depth equations they neglect certain terms and the underlying mathematical model will be less accurate at this point.

5.5 Boundary conditions

5.5.1 Introduction

In the last chapter boundary conditions for the one-dimensional extended Boussinesq equations and their numerical implementation were discussed. In this section those techniques are extended to the two-dimensional equations. Two-dimensional effects, not considered in the previous chapter, are discussed in detail. In this work an attempt is made to derive boundary conditions in a mathematically and physically consistent manner and this is compared and contrasted with previously published work.

In two-dimensions the boundary segments completely enclose the domain, and care must be taken to ensure compatibility between conditions imposed at the intersection of boundary segments with different boundary conditions.

5.5.2 Inflow boundaries

In Section 4.5.1 it was shown how periodic waves could be introduced to the domain at inflow boundaries. These techniques can be directly applied to inflow boundaries in two-dimensions if those boundaries are straight and aligned with the x or y coordinate directions and the wave front is parallel to the boundary it crosses.

For more realistic cases involving complicated geometry and wave conditions it is unlikely that this simple boundary condition will be sufficient. In general the incident wave conditions can be multi-directional, multi-component wave forms [73]. This fully general case is not considered here, but some of the issues such as the inclusion of harmonic interactions, which was discussed in Section 4.5.1, and the introduction of waves at an angle to the boundary are considered.

To introduce a wave at an angle to the inflow boundary the one-dimensional plane wave solution is rotated by the appropriate angle. Considering the one-dimensional inflow conditions described in Section 4.5.1,

$$\eta_p = a\,sin(kx-\omega t) \tag{5.53}$$

$$u_p = b\,sin(kx-\omega t) \tag{5.54}$$

$$w_p = -\beta H^3 k^2 b\,sin(kx-\omega t) \tag{5.55}$$

the two-dimensional solution produced by a rotation of the flow direction by an angle θ_R with respect to the x-axis is given by,

$$\eta(x,y,t) = \eta_p \tag{5.56}$$
$$u(x,y,t) = cos(\theta_R)\, u_p \tag{5.57}$$
$$v(x,y,t) = sin(\theta_R)\, u_p \tag{5.58}$$
$$w_1(x,y,t) = cos(\theta_R)\, w_p \tag{5.59}$$
$$w_2(x,y,t) = sin(\theta_R)\, w_p \tag{5.60}$$

As discussed in Section 4.7, for the consistency of the whole problem, all boundary data must be zero at the initial time. The inflow boundary condition described above will not satisfy this condition initially in general. To overcome this problem the wave is gradually introduced into the domain in the following manner. If the inflow condition is assumed to be a plane periodic wave of the form (5.56)-(5.60) then a wave front can be defined as,

$$k z_R - \omega t = 0 \tag{5.61}$$

where,

$$z_R = cos(\theta_R)\, x + sin(\theta_R)\, y \tag{5.62}$$

moving with speed,

$$C = \frac{\omega}{k} \tag{5.63}$$

in a direction normal to the wave front. At each time step the inflow nodes are tested and marked as *active* or *inactive* according to the following rule,

$$(k z_R - \omega t) \begin{cases} < 0 & active\ inflow \\ \geq 0 & inactive\ inflow \end{cases} \tag{5.64}$$

At *active* inflow boundary nodes the solution is multiplied by a factor,

$$c_{in} = c_1 \left(1 - e^{\left(\frac{k z_R}{\omega} - t \right)} \right) \tag{5.65}$$

c_1 is a problem dependent constant to be determined by experiment. The quantity $\frac{k z_R}{\omega}$ represents the time at which the wave front will reach the position z_R, assuming it is at $z_R = 0$ at $t = 0$. After this time the coefficient c_{in} ensures the wave is gradually increased

from zero at that point. *Inactive* boundary nodes are set with zero solution values. This ensures a gentle and continuous start-up of the flow across the inflow boundary which is essential for the efficient use of the adaptive time integration software.

5.5.3 Solid wall boundaries

Since the equation system is based on an inviscid flow model the physical boundary condition is that of impermeability,

$$\mathbf{u}\cdot\mathbf{n} \;=\; 0 \tag{5.66}$$

The application of this condition to the two-dimensional velocity field in the extended Boussinesq equations is consistent with the definition of the true three-dimensional velocity field if all boundaries are considered vertical.

In the continuous case a free surface boundary condition can be derived from the free surface equation (5.12). Integrating this equation over the domain Ω,

$$\int_\Omega (\dot{\eta} + \nabla\cdot(p,q) + \nabla\cdot\mathbf{w})\, d\Omega \;=\; 0 \tag{5.67}$$

Using the divergence theorem, and taking the time derivative outside the spatial integral gives,

$$\frac{\partial}{\partial t}\left(\int_\Omega \eta\, d\Omega\right) + \int_\Gamma ((p,q)+\mathbf{w})\cdot\mathbf{n}\, d\Gamma \;=\; 0 \tag{5.68}$$

Where Γ is the boundary of Ω and $\mathbf{n} = (n_x, n_y)$ is the outward normal on Γ. By conservation of mass the time derivative term must be zero since there is no loss or gain of mass through the wall. Hence,

$$\int_\Gamma ((p,q)\cdot\mathbf{n} + \mathbf{w}\cdot\mathbf{n})\, d\Gamma \;=\; 0 \tag{5.69}$$

Using the definitions of p and q, respectively equations (5.18) and (5.19), this can be written as,

$$\int_\Gamma ((h+\eta)\,\mathbf{u}\cdot\mathbf{n} + \mathbf{w}\cdot\mathbf{n})\, d\Gamma \;=\; 0 \tag{5.70}$$

Equation (5.66) implies that the first part of this expression is zero on Γ and hence additionally requiring,

$$\mathbf{w}\cdot\mathbf{n} \;=\; 0 \tag{5.71}$$

will completely satisfy the boundary condition (5.70). If the boundary conditions (5.66) and (5.71) are imposed it is then inconsistent to apply the original boundary condition (5.70) as this will over-specify conditions at the solid wall. In this work there is no boundary condition applied to the free surface at solid walls, which is equivalent in the discrete case to extrapolating the boundary integral over the element neighbouring the boundary edge. Here, equivalently, the discrete equation (5.37) does not contain a boundary integral as there is no integration by parts, and assembling this equation is equivalent to the extrapolation of the boundary integral.

The boundary condition (5.66) supplies a constraint on the normal velocity at the wall. Tangentially to the wall there is no constraint and a momentum equation is required for the evolution of the tangential velocity at the wall. This problem has been considered previously in the context of fluid flow problems by Engelman et al. [38].

Consider a local coordinate system (n,s) with n aligned with the outward normal and s aligned with the tangent at that point. This corresponds to locally rotating the (x,y) coordinate system by an angle θ_W to the new orientation (n,s). The velocity components (u_N, v_S) with respect to the coordinate system (n,s) can be related to the (x,y) velocities (u,v) by,

$$u_N = \cos\theta_W\, u + \sin\theta_W\, v \tag{5.72}$$

$$v_S = -\sin\theta_W\, u + \cos\theta_W\, v \tag{5.73}$$

In the normal direction the velocity component u_N satisfies the normal boundary condition (5.66),

$$u_N = 0 \tag{5.74}$$

Considering the constant depth equations (5.20)-(5.24) for simplicity of presentation, in the tangential direction the velocity component v_S satisfies the momentum equation,

$$\dot{v_S} + \frac{\partial F}{\partial s} + \alpha H^2 \frac{\partial}{\partial s}\left(\frac{\partial \dot{u_N}}{\partial n} + \frac{\partial \dot{v_S}}{\partial s}\right) = 0 \tag{5.75}$$

where,

$$F = \frac{1}{2}\left(u_N^2 + v_S^2\right) + g\eta \tag{5.76}$$

Note that the surface elevation η is invariant under rotation, and that the quantity $\frac{1}{2}\left(u_N^2 + v_S^2\right)$, corresponding to a velocity magnitude, is similarly invariant. Hence,

$$F(\eta, u_N, v_S) \equiv f(\eta, u, v) \tag{5.77}$$

Using chain rules allows the derivatives with respect to the rotated coordinates (n,s) to be rewritten with respect to the (x,y) coordinate system,

$$\begin{aligned}\frac{\partial}{\partial n} &= \frac{\partial x}{\partial n}\frac{\partial}{\partial x}+\frac{\partial y}{\partial n}\frac{\partial}{\partial y} \\ &= \cos\theta_W\frac{\partial}{\partial x}+\sin\theta_W\frac{\partial}{\partial y} \qquad (5.78)\\ \frac{\partial}{\partial s} &= \frac{\partial x}{\partial t}\frac{\partial}{\partial x}+\frac{\partial y}{\partial t}\frac{\partial}{\partial y} \\ &= -\sin\theta_W\frac{\partial}{\partial x}+\cos\theta_W\frac{\partial}{\partial y} \qquad (5.79)\end{aligned}$$

Now consider the terms of the momentum equation (5.75) in turn. Using (5.73) the time derivative becomes,

$$\dot{v_S} = -\sin\theta_W\ \dot{u}+\cos\theta_W\ \dot{v} \qquad (5.80)$$

Using (5.79) and (5.77) the first derivative becomes,

$$\frac{\partial F}{\partial s} = -\sin\theta_W\frac{\partial f}{\partial x}+\cos\theta_W\frac{\partial f}{\partial y} \qquad (5.81)$$

Considering the dispersive term from equation (5.75),

$$\begin{aligned}\frac{\partial \dot{u_N}}{\partial n}+\frac{\partial \dot{v_S}}{\partial s} &= \left(\cos\theta_W\frac{\partial}{\partial x}+\sin\theta_W\frac{\partial}{\partial y}\right)(\cos\theta_W\ \dot{u}+\sin\theta_W\ \dot{v}) \\ &\quad+\left(-\sin\theta_W\frac{\partial}{\partial x}+\cos\theta_W\frac{\partial}{\partial y}\right)(-\sin\theta_W\ \dot{u}+\cos\theta_W\ \dot{v}) \\ &= \frac{\partial \dot{u}}{\partial x}+\frac{\partial \dot{v}}{\partial y} \qquad (5.82)\end{aligned}$$

ie. the divergence of the velocity is similarly invariant to rotation. Differentiating equation (5.82) with respect to s,

$$\frac{\partial}{\partial s}\left(\frac{\partial \dot{u_N}}{\partial n}+\frac{\partial \dot{v_S}}{\partial s}\right) = \left(-\sin\theta_W\frac{\partial}{\partial x}+\cos\theta_W\frac{\partial}{\partial y}\right)\nabla\cdot(\dot{u},\dot{v}) \qquad (5.83)$$

Substituting equations (5.80), (5.81) and (5.83) into (5.75),

$$\begin{aligned}&-\sin\theta_W\left(\dot{u}+\frac{\partial f}{\partial x}+\alpha H^2\frac{\partial}{\partial x}(\nabla\cdot(\dot{u},\dot{v}))\right) \\ &\qquad+\cos\theta_W\left(\dot{v}+\frac{\partial f}{\partial y}+\alpha H^2\frac{\partial}{\partial y}(\nabla\cdot(\dot{u},\dot{v}))\right) = 0 \qquad (5.84)\end{aligned}$$

Comparison with the momentum equations (5.21) and (5.22) reveals that the tangential

momentum equation (5.84) is obtained from the dot product of the (x,y) momentum equations with the vector $\mathbf{s} = (-\sin\theta_W, \cos\theta_W)$ which is the tangent to the wall at that point.

In the numerical implementation of these boundary conditions it is noted that, when the dot product of the tangent and the discrete momentum equations (5.38) and (5.39) is formed, the boundary integrals disappear and need not be considered at these boundaries.

Note that this formulation of the wall boundary condition depended on the compact form of the nonlinear term f in the equations, in particular that it is invariant to rotation. This is also the reason that the particular ordering of the dispersive derivatives was chosen in equations (5.13) and (5.14) rather than rewriting them with the vector identity (5.11).

At a solid wall boundary the momentum equations (5.38) and (5.39) are replaced by the velocity boundary condition (5.66) and the tangential momentum equation (5.84). Care must be taken in the order of this replacement to ensure that the inviscid boundary condition (5.66) does not leave a zero diagonal entry in the matrix system, eg. if the normal $(n_x, n_y) = (1,0)$ and equation (5.66) replaces the y momentum equation (5.39). To ensure against this, equation (5.66) always replaces the momentum equation corresponding to the component of the normal of largest absolute size.

At a solid wall the boundary condition on the auxiliary variable (5.71) is also applied. In a similar manner to the velocity field the equations (5.12) and (5.13) can be transformed into normal and tangential form, resulting in the disappearance of the boundary integrals in the discrete form of the equations (5.45) and (5.46) in the tangential direction. w_1 and w_2 can then be prescribed as Dirichlet values from,

$$w_1^{wall} = (\mathbf{w}\cdot\mathbf{s})\, s_x \tag{5.85}$$

$$w_2^{wall} = (\mathbf{w}\cdot\mathbf{s})\, s_y \tag{5.86}$$

In previously published work with finite difference methods the wall boundary conditions usually applied are that the normal velocity and the normal derivative of the free surface are zero. This latter condition can be derived from a lower order form of the free surface equation (5.12) [106] although it does not appear to directly enforce conservation as is the case with the condition (5.71) derived here. However it has been frequently used with finite difference methods [97, 106] and also in the finite element method of Li et al. [62]. It has also been noted that with the finite difference schemes an additional non-physical constraint is required at a solid wall to produce a stable numerical scheme [97, 106]. This is that the normal derivative of the tangential velocity component is zero. Assuming a first order difference approximation applied on a square finite difference grid aligned with the wall this condition implies that the tangential velocity at the wall is made equal to the tangential velocity at the adjacent internal point, essentially imposing no shear at the wall. In contrast the boundary conditions derived here require

no further constraints and indeed attempting to apply one will over-specify the solution at the wall.

Corners in the wall boundary segments require special treatment in the numerical method. At concave corners, ie. with internal angles less than 180^o, a non-zero velocity will imply flow through the wall and hence the velocity is set to zero at these points. At convex corners, ie. with internal angles greater than 180^o, the velocity can be non-zero without implying flow through the wall. Skotner and Apelt [97] note that problems are encountered due to non-physical oscillations being introduced at these points in their finite difference method for Nwogu's extended Boussinesq equations. They use upwind differencing at these points, which introduces numerical diffusion, to prevent the oscillations. Here the velocity is allowed to be non-zero and in the direction defined by the normal formed from the average of the two adjacent boundary edges with the boundary conditions (5.66) and (5.71).

Where a boundary segment with inflow boundary conditions meets a wall boundary segment the inflow solution must be consistent with the wall boundary conditions on the velocity (5.66) and the auxiliary variables (5.71).

5.5.4 Outflow boundaries

Outflow boundaries are non-physical restrictions on the size of the domain and hence it is important to minimise the reflection of information back from these boundaries. In Section 4.5.3 viscous damping layers, termed sponge layers [48], were used near these boundaries to absorb outgoing waves and prevent wave reflection. Here, in contrast to previously published work [106], these viscous terms were added to the free surface equation and it was found that they effectively absorbed outgoing waves.

In two dimensions a similar approach is adopted. The viscous damping layer is implemented by adding a viscous term to the free surface equation. Here this is illustrated for the constant depth equation (5.20).

$$\dot{\eta} + \nabla\cdot(p,q) + \nabla\cdot\mathbf{w} \;=\; c_s \nabla\cdot\nabla\eta \tag{5.87}$$

The coefficient of this term c_s decreases exponentially away from the boundary, and is zero in most of the domain. The form of this coefficient is here taken to be a generalisation of the one-dimensional form (4.67),

$$c_s \;= n_1 \frac{e^{\left(\frac{z_n - z_S}{z_{out} - z_S}\right)^{n_2}} - 1}{e^1 - 1} \tag{5.88}$$

where z_n denotes the normal distance from the outflow boundary, and the sponge layer

extends from the boundary $z_n = z_{out}$ to $z_n = z_s$. n_1 and n_2 are constants used to fine tune the amount of viscosity required at the boundary. They will generally be problem dependent. This form appears suitable for the outflow boundaries considered in the examples in this work but will probably not be so in all cases, eg. for more irregular boundary regions.

In two-dimensions these boundary conditions are often found to be much less effective [112]. This is probably due to the fact that the formulation of the viscous layers is based on a one-dimensional analysis involving the wave front arriving parallel to the layer [60]. In two-dimensions it is much harder to prevent reflections from waves that arrive at large angles to the viscous layers. In practice this means that much more work must be expended in fine-tuning the specification of these layers for two-dimensional calculations.

The implementation of the sponge layer in equation (5.87) is in an isotropic form and it is possible that wave direction information could be used to formulate a non-isotropic coefficient and improve its performance for non-parallel waves. However such a form has not been used in this work.

Where a boundary segment with inflow boundary conditions meets an outflow boundary segment the inflow solution must be modified to prevent spurious interactions within the sponge layer. In practice the inflow solution is reduced near sponge layer boundaries and is made zero where the inflow boundary segment is within a sponge layer region.

5.5.5 Internal generation of waves

The simulation of realistic problems requires long time calculations in order to allow the initial transient effects to be removed. If wave generation is implemented by prescribing the waveform at the boundary, as described in Section 5.5.2, then a wave that is reflected back to that boundary will not be able to pass through that region and leave the domain. In geometrically complicated regions interactions with solid boundaries will generally produce a multi-directional wave field and problems at such inflow boundaries will become increasingly likely.

In Section 4.5.4 a new method of generating waves internally to the domain by simulating an oscillation of the sea bed was described. This was applied successfully to the one-dimensional extended Boussinesq equations and allowed reflected waves to pass through the generation region and leave the domain. In this section the application of this method in two dimensions is considered.

Larsen and Dancy [60] were the first to describe such an approach for a two-dimensional finite difference model of the original Boussinesq equation system. They introduced an increment to the free surface along a generation line which varied periodically to produce a wave motion. They showed that this approach could be used on fairly general geometries,

within the restrictions of the cartesian finite difference grid. Skotner and Apelt [97] have described an alternative internal wave generation method for a two-dimensional finite difference model of Nwogu's extended Boussinesq equations. This method is relatively complex, involving a modified stencil in the wave generation region and explicit removal of the wave in the region behind the wave generator to prevent reflection. More recently Wei et al. [108] have rigorously derived a source function that can be added to the free surface equation to generate regular or irregular wave fields inside the domain. This was implemented as part of their finite difference model for Nwogu's extended Boussinesq equations [106].

Here, in two dimensions, it is found that the simple modification for a time-varying depth described by equation (4.69) is not sufficient to produce accurate internal waves. This modification of the equation system disregards the effect of the time variation of the depth on all the other terms in the equation system. Here it is proposed to additionally include these effects in the convective terms of the free surface equation (5.12). The modified free surface equation is then,

$$\dot{\eta} + \dot{h_I} + \nabla\cdot((h + h_I + \eta)\,u) + \nabla\cdot\mathbf{w} \;=\; 0 \tag{5.89}$$

The time variation of the depth will also affect the dispersive terms in the system. However, as discussed in Section 5.5.2, the width of the generating region is proportional to the wavelength and the amplitude of undulation is proportional to the wave amplitude, hence it reasonable to assume that the perturbation to the dispersive terms can be neglected within the order of approximation of the Boussinesq system.

5.6 Time integration

5.6.1 The system of equations

The two-dimensional finite element spatial discretisation described in the previous section produces a differential-algebraic equation system of the form,

$$A_{ik}(y,t)\dot{y}_k(t) \;=\; f_i(y,t) \tag{5.90}$$

where y and $\dot{y}$ contain all the current values of the solution and their time derivatives respectively. The matrix A_{ik} has constant coefficients, and f is a nonlinear function of y. With the explicit approximations (5.45) and (5.46) used for the auxiliary variables, and with these equations decoupled from the differential-algebraic system as described in Section 5.4, equation (5.90) represents a $3np$-dimensional system of nonlinear equations to be solved within the time integration algorithm, where np is the number of nodes in the

finite element mesh. Following the description outlined in Section 3.3 the time integration of the nonlinear system (5.90) will require the solution of the following system of linear equations,

$$J_{ij}(y)\Delta y_j \;=\; -\alpha_T\,\Delta t\; r_i(y) \tag{5.91}$$

where Δt is the current time step, α_T is a constant that depends on the time integration method used, and Δy is the solution increment at that time step. The Jacobian matrix J_{ij} is given by,

$$J_{ij} \;=\; \alpha\,\Delta t\frac{d\,f_i}{dy_j} - A_{ij} \tag{5.92}$$

for the spatial discretisation described by equation (5.90). The vector y contains the free surface and velocity unknowns in the order,

$$y \;=\; (\eta_1, u_1, v_1, \eta_2, u_2, v_2, \ldots, \eta_{np}, u_{np}, v_{np})^t \tag{5.93}$$

The variables (η, u, v) at a node are stored consecutively, however in contrast to the one-dimensional spatial discretisation considered in Section 4.6 the node numbering for a two-dimensional finite element mesh typically produces a sparse matrix system. Within the time integration algorithm this will produce a sparse Jacobian matrix (5.92) and hence a sparse matrix equation system (5.91) to be solved at each Newton iteration. This system will be most efficiently solved using sparse matrix techniques [84], and in particular the banded matrix solution methods used in Chapter 4 for the one-dimensional solution algorithm will be inefficient for these problems.

Within the time integration package SPRINT the most common solution method for such systems is to use a sparse direct solution algorithm via a complete factorisation of the Jacobian matrix [20]. However for large problems the computational time required by the factorisation algorithm and the storage requirements for the factored matrix can become too large for practical computations [105]. An alternative to the direct solution of the sparse matrix system is to use iterative methods. The SPRINT package has been linked with the WATSIT package of iterative methods [50, 105]. The most general method contained in the package is the preconditioned truncated Krylov-subspace based method ORTHOMIN.

The time integration software DASPK has no facility for direct solution of sparse equation systems. However it has been combined with the iterative methods from the SPARSKIT package [84], the most general of which is the preconditioned truncated Krylov-subspace based method GMRES.

The following sections examine the application of direct and iterative solution meth-

ods within SPRINT and DASPK to solve the equation system (5.91). Reordering strategies for the matrix are discussed which can lead to gains in efficiency both in storage requirements and computational time.

5.6.2 Solution methods for the equation system

5.6.2.1 Direct solution methods

SPRINT contains two separate modules for the direct solution of sparse linear equation systems; the Yale Sparse Matrix Package and the NAG sparse matrix routines [20]. An efficient storage method is necessary for the algorithms to take full advantage of the sparse matrix structure. Various methods exist for storage of only the non-zero elements of a sparse matrix [84]. Both the methods within SPRINT use compressed column storage format [20] which stores pointers to every non-zero row entry in a column and pointers to where each new column begins in the list of row entries. These integer lists are an additional computational overhead but this is outweighed by the reduction in the storage for the matrix itself for a sufficiently sparse matrix. The localised nature of the linear finite element basis used in this work will ensure that this method of storage is always efficient for large problems. The direct solution methods both solve the system by first calculating a lower-upper (LU) factorisation of the matrix. The Yale routines are slightly less robust as they do not allow pivoting [20] which can lead to an accumulation of numerical round-off errors in some cases, hence only the NAG routines are considered here.

A disadvantage of direct sparse solution methods is that the factorisation of the Jacobian matrix is a computationally expensive operation both in terms of CPU time and memory requirements. Factorising the matrix generates *fill-in*; non-zero entries in positions that are not included (ie. zero) in the sparse representation of the original matrix, and the size of the factorisation is bounded only by the bandwidth of the sparse matrix and is not directly related to the sparsity pattern. Hence the LU decomposed matrix can have a much larger size than the matrix itself [84].

The two-dimensional finite element model considered here has extra entries on each row of the Jacobian due to the coupling with the auxiliary w_i equations. Fig. 5.2 illustrates the region of the discretisation coupled at a node for a linear triangular mesh. Thus a node is coupled to two layers of elements surrounding it rather than the more usual one layer. A typical linear triangular mesh with six elements surrounding a node will have nineteen non-zero nodal blocks per row, and with three equations per node in the Boussinesq system will correspond to fifty-seven non-zero entries per row. This increases the size of the Jacobian matrix and greatly increases the amount of potential fill-in of the LU factorisation.

Note that solving the equation system in the fully coupled form (5.37)-(5.41) would

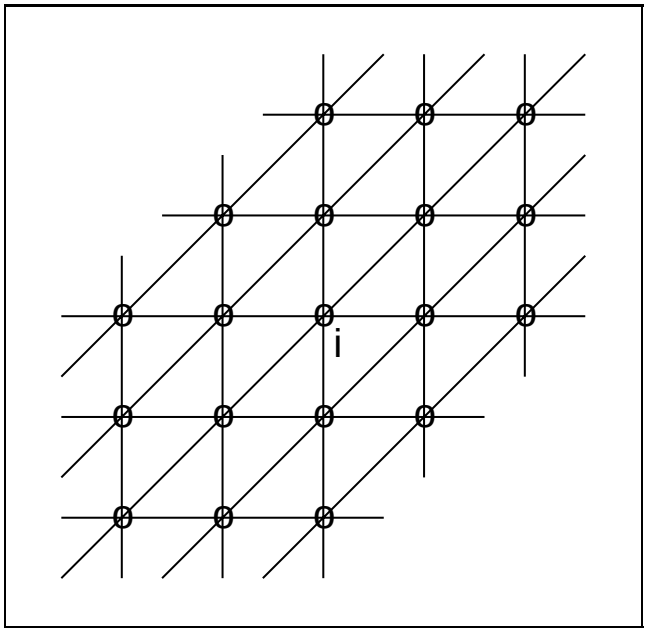


Figure 5.2: The region coupled to node *i* by the finite element discretisation of the extended Boussinesq equations

lead to five equations at each node. However these are coupled only to the first layer of elements surrounding a node, typically seven nodes per row of the matrix, and hence will typically contain thirty-five entries per row. This advantage is reduced by consideration of the increased number of degrees of freedom of the system, and by the increased bandwidth of the matrix, which is discussed in more detail in the following sections.

5.6.2.2 Reorderings of the matrix

The sparsity structure of the LU factorisation is determined by the sparsity pattern of the original matrix [84]. It is therefore possible to increase the efficiency of the process both in storage and computational time by reordering the matrix such that the size of the LU factorisation is minimised. Whilst this problem is too expensive to solve exactly there are a variety of approximate solution strategies that can be used [43]. The strategies considered here are all concerned with renumbering the nodes of the finite element mesh with the ordering so determined leading to a reduced size of the LU factored matrix. These procedures can all be implemented as a preprocessing step on the mesh and do not have any overhead on the cost of the time integration.

One of the most common reordering strategies used in combination with the finite element method is the Reverse Cuthill-McKee ordering [43]. This procedure attempts to minimise the bandwidth of the mesh, where the bandwidth is defined as the maximum distance from the diagonal of a non-zero entry for all rows and columns of the mesh. More precisely if the set of *ns* edges $\{E_{ij}^k, k = 1, \ldots, ns\}$ of the mesh is considered, where *i* and

j denote the numbers of the nodes comprising the edge, then the bandwidth is defined as,

$$\max_{E_{ij}^k} \|i-j\| \qquad \forall k = 1,\ldots,ns \tag{5.94}$$

Using the SPRINT software suitably small two-dimensional problems may be efficiently solved by such a reordering combined with the banded matrix software used for the one-dimensional problems. However for larger two-dimensional problems the increased size of the bandwidth makes such methods inefficient compared to fully sparse methods of solution.

The Reverse Cuthill-McKee ordering also reduces the fill-in produced by an LU factored matrix by constraining it to be within the bandwidth of the original matrix. However this bandwidth will become large as the problem size increases. For example, a square domain discretised with N^2 regularly spaced nodes, and hence $2(N-1)^2$ triangular elements, will have a bandwidth of N+1. For unstructured triangular grids over irregular domains the bandwidth is seen to be related to the number of nodes across the smaller of the width or height of the domain.

For the finite element method considered here a node is coupled to the two layers of elements surrounding it as described in the previous section and the bandwidth is effectively doubled. With three equations at each node the bandwidth of the time integration system is approximately six times the nodal bandwidth of the triangular finite element mesh. The fully coupled differential-algebraic system, with five equations at a node, is only coupled to the first layer of elements surrounding a node, and hence the bandwidth of the time integration system is approximately five times the nodal bandwidth of the triangular finite element mesh.

Simplified versions of the Reverse Cuthill-McKee algorithm have been developed, such as the advancing front renumbering method [64] although these will rarely produce superior orderings.

An alternative ordering strategy is that of nested dissection methods, such as those implemented within the mesh partitioning package METIS [54]. This method is commonly used to partition, or colour, a mesh into disconnected sets of nodes but also has the facility for computing fill-reducing orderings of the nodes. The basic approach is to split the finite element mesh into two disconnected parts with a common boundary. Thus the nodes of one part can produce no fill-in in positions corresponding to the other disconnected part. The procedure is then repeated recursively until the parts are suitably small.

Experiments with METIS have found that the orderings produced are not greatly superior to those from the Reverse Cuthill-McKee algorithm. This may be due to the fact that the finite element discretisation of the extended Boussinesq equations produces extra couplings between the variables due to the w_i equations and this makes the dissection

process more complicated as the nodes are coupled to two layers of elements and the boundaries between disconnected parts have to be thicker for a correct partitioning.

In this work the Reverse Cuthill-McKee algorithm has been used in all the two-dimensional cases considered. This reordering method is available as an option internal to DASPK but here it has been implemented as a stand-alone preprocessing step on the finite element mesh according to the algorithms presented by George and Liu [43].

5.6.2.3 Iterative solution methods

ORTHOMIN is a generalised conjugate residual method [50]. It is implemented within SPRINT through the WATSIT package [105]. It is suitable for use with non-symmetric matrices although they must remain nearly positive definite [105]. At each step of the method a new search direction is computed which is orthogonal to the search directions previously used and the solution is advanced by minimising the residual along this direction. Each step involves a matrix-vector product which can be computed efficiently in sparse form. The process of orthogonalisation becomes expensive if many search directions are stored and so the algorithm terminates after a set number of iterations and restarts with the latest residual and solution. This does not guarantee orthogonality in all the search directions used but in practice is found to be a more efficient method overall [105].

The performance of the basic iterative method is greatly improved by the application of a suitable preconditioning matrix to the system. This preconditioning step should transform the system into one where the eigenvalues are tightly bunched and hence the ORTHOMIN search will converge rapidly [105]. The ideal preconditioner will be a good approximation to the inverse of the matrix but a compromise must be made between the accuracy of the approximation and the computational cost of forming and storing it.

The WATSIT package contains preconditioning strategies based on incomplete LU (ILU) factorisations of the original matrix. These reduce the computational time and storage requirements of the full LU factorisation by disregarding a certain amount of the fill-in entries. The ILU factorisations implemented within WATSIT are based on the graph structure of the sparse matrix and are termed ILU(k) methods [50], the integer k determining the level of fill-in that is allowed in the factorised matrix. The simplest such method, termed ILU(0), produces an LU factorisation with the same sparsity pattern as the original matrix. This is simple to form but may not be as efficient overall as higher level ILU factorisations [84]. An ILU(1) factorisation allows a predetermined amount of fill-in into the decomposition. The new, larger sparsity pattern has to be precomputed which adds to the computational cost as well as the storage. The extra expense is traded off against the faster convergence obtained with the more accurate preconditioner. Higher

level ILU factorisations are also possible and it is generally problem dependent as to which is the most efficient choice. A possible disadvantage of the ILU factorisations described above is that they are based on the sparsity pattern of the original matrix and do not take the numerical values of the entries into account.

The time integration package DASPK has been combined with the iterative solution methods of the SPARSKIT package [25]. This implements the GMRES iterative method together with sophisticated ILU preconditioning techniques, generally termed ILU(tol) [84]. Such methods disregard fill-in entries based on their numerical size and use numerical tolerances to control the amount of fill-in [31]. They can produce more accurate preconditioners for the same amount of storage as ILU(k) methods but are more computationally expensive to compute. DASPK stores the matrix and its factorisation in compressed row format [84], whereby non-zero entries are stored row-by-row together with pointers to indicate the start of each row and the node numbers corresponding to the non-zero entries.

There are many other choices for the preconditioner. Simpler methods such as diagonal scaling or block diagonal scaling may be more efficient for some problems [84]. Other more sophisticated methods include direct approximations to the inverse of the matrix [84]. The preconditioners used in this work are limited to those already implemented within the time integration packages as described previously.

The reordering strategies discussed in Section 5.6.2.2 are also useful for the iterative methods. The Reverse Cuthill-McKee bandwidth reduction method will reduce the potential amount of fill-in in the factorised matrix by constraining it to lie within a narrower region of the matrix, and hence the ILU factorisation will be more accurate for a prespecified amount of storage. Recent work by Benzi et al. [14] on ordering strategies for incomplete matrix factorisation also concluded that the use of the Reverse Cuthill-McKee method was advantageous for non-symmetric matrix problems.

Another technique often used in conjunction with iterative methods is red/black ordering [30, 105]. The equations are partitioned into two groups; the red which are coupled only to black variables, and the black which are all the other equations. Only the black equations are solved iteratively and the red variables are computed by back substitution once the black variables have been computed. If the partition is nearly even, such as would be the case for a rectangular finite difference grid, then a system of equations roughly half the size of the original problem is all that has to be solved. A two-dimensional linear triangular finite element mesh can generally be partitioned into roughly one-third red and two-thirds black nodes. However the extra coupling of variables due to the auxiliary variables used in the finite element method here will reduce the size of the red system significantly. Additionally, when a system of equations is solved there is more than one equation at a node and the coupling of these equations has to be taken into account. In

the case of the finite element method considered here only one of the three equations at a node can be selected as red reducing the overall amount of red equations again. The extra complications involved in solving the system in red/black order mean that the technique is unlikely to be efficient in this case, and it has not been used in this work.

5.6.2.4 Evaluation of the Jacobian matrix

Although iterative methods generally use less storage than direct methods the time integration methods used here still need to compute and store the Jacobian matrix in order to form an efficient preconditioner, and so these methods cannot be considered truly matrix-free [105]. The form of the Jacobian matrix for the proposed finite element method is given by equation (5.92) and the alternatives for its evaluation within the time integration process are that of numerical approximation by the software or of supplying an analytical form of the Jacobian as discussed in Section 3.3.

Supplying an analytical form of the Jacobian is likely to be the most efficient mode of operation [20] if it can be simply constructed. In the case of the finite element method it would be desirable to form this matrix with a finite element assembly loop, with a computational expense equivalent to the spatial discretisation of the equations. However the sparse storage scheme used for the matrix means that it is not straightforward to locate where an element of the Jacobian must be stored. Once the required column is located a simple search could find the position but this will be expensive as it has to be repeated several times for each element matrix. For a simple one variable per node linear triangular finite element model an element of the mesh will have a three-by-three element Jacobian matrix hence an alternative is to store a three-by-three array of precomputed pointers to the locations within the sparse structure. This has been successfully implemented for a simple heat conduction problem within the SPRINT framework. However the extended Boussinesq equations have three equations per node and hence a nine-by-nine element Jacobian matrix. Therefore a significant amount of extra storage would be required for a similar pointer structure. The additional support of a node described in Section 5.6.2.1 due to the auxiliary variables in the extended Boussinesq system further complicates matters as the support of a node extends beyond the immediate neighbourhood of the node and it may be that this pointer system is not suitable for this system.

The computation of this matrix can also be performed numerically within the time integration codes. Within SPRINT this matrix is computed by a series of spatial discretisations which build up a numerical approximation to the Jacobian by perturbing each degree of freedom of the equation system in turn [20]. Advantage can be taken of the sparsity of the matrix to perturb several variables at once if their dependence on the other variables of the system does not intersect. Within SPRINT, non-intersecting groups of

columns are formed from the sparsity pattern and the Jacobian is evaluated numerically with a minimum number of spatial discretisation steps [20]. Within DASPK the Jacobian is evaluated numerically by grouping rows separated by a distance corresponding to the bandwidth of the matrix, which ensures non-intersection of the rows. This can become inefficient for the two-dimensional finite element method developed here as the mesh bandwidth may become large. If the Reverse Cuthill-McKee method described in Section 5.6.2.2 is used this problem is minimised but the cost of the numerical approximation is still significantly greater than that of SPRINT. DASPK uses the compressed row sparse storage format and so the sparse assembly procedure using groups of columns as implemented within SPRINT, which is based on the compressed column sparse format, is not directly applicable.

5.6.3 Application of these methods to the spatially discretised system

Simple test cases for the two-dimensional algorithm, such as the channel flow considered in Section 5.8 showed that the sparse direct solution methods rapidly become inefficient as the problem size increases, and hence attention is focused on the iterative methods.

For quasi-one-dimensional flow, both SPRINT and DASPK compute accurate solutions to the problems. However when considering fully two-dimensional flows it is found that the iterative methods used with SPRINT occasionally fail to converge. This problem has been traced to the form of the Jacobian matrix, which can become non-positive definite due to the mixed second order spatial derivatives present in equations (5.13) and (5.14). These produce significant off-diagonal terms in the Jacobian which are not apparent for one-dimensional flows when the cross derivatives are effectively zero. The ILU(k) preconditioners appear to become inaccurate in these cases as these off-diagonal terms will produce significant fill-in in the factorised matrix in positions that are not considered by these graph based methods unless a high order ILU(k) factorisation is chosen. The ILU(tol) preconditioners used within DASPK are based on numerical tolerances and hence these off-diagonal entries and the fill-in they produce are included in the factorisation making them significantly more accurate.

The numerical examples in Section 5.8 are all produced using the time integration software DASPK with the GMRES iterative method and ILU(tol) preconditioning. The Jacobian matrix is computed numerically within DASPK, despite the inefficiencies noted in Section 5.6.2.4. It is hoped that the Jacobian matrix does not have to be evaluated many times during the time integration process, as discussed in Section 3.3, in which case the effect of this inefficiency may not be too significant. The finite element mesh is preprocessed with the Reverse Cuthill-McKee method to minimise the bandwidth. This reduces

the fill-in of the factorised matrix and also minimises the work done in the numerical approximation of the Jacobian matrix within DASPK.

5.7 Initial conditions

It is common to begin the numerical simulations with an undisturbed free surface. This has the advantage of automatically satisfying the irrotational flow condition. It also avoids the calculation of consistent initial conditions for the time integration process, ie. calculating consistent initial time derivatives given an initial solution, which can require a relatively large amount of additional work [26].

The drawback of imposing zero initial conditions is that they do not give any indication to the time integration software as to the initial behaviour of the system. The time integrator has do a lot of work on the first step in order to calculate a suitable initial step size. It is possible to supply the initial step size to the code and this will greatly reduce the amount of work done in the initial stages; however, if this initial step size is set inappropriately large or small a substantial amount of additional computational work will be performed.

The most efficient way to deal with this is to run the code once for a particular problem for only a few time steps. The initial time step chosen by the code is then supplied for subsequent runs of this problem. This appears to greatly reduce the amount of work required for the first step, although changing the physical parameters of the problem, particularly the wave conditions, will affect the value.

The initial stages of the calculation are damped by the addition of a viscous term to the free surface equation in a similar manner to that done for the sponge layer in equation (5.87). This is achieved by the addition of a second coefficient c_t to the c_s coefficient previously introduced for the sponge layer.

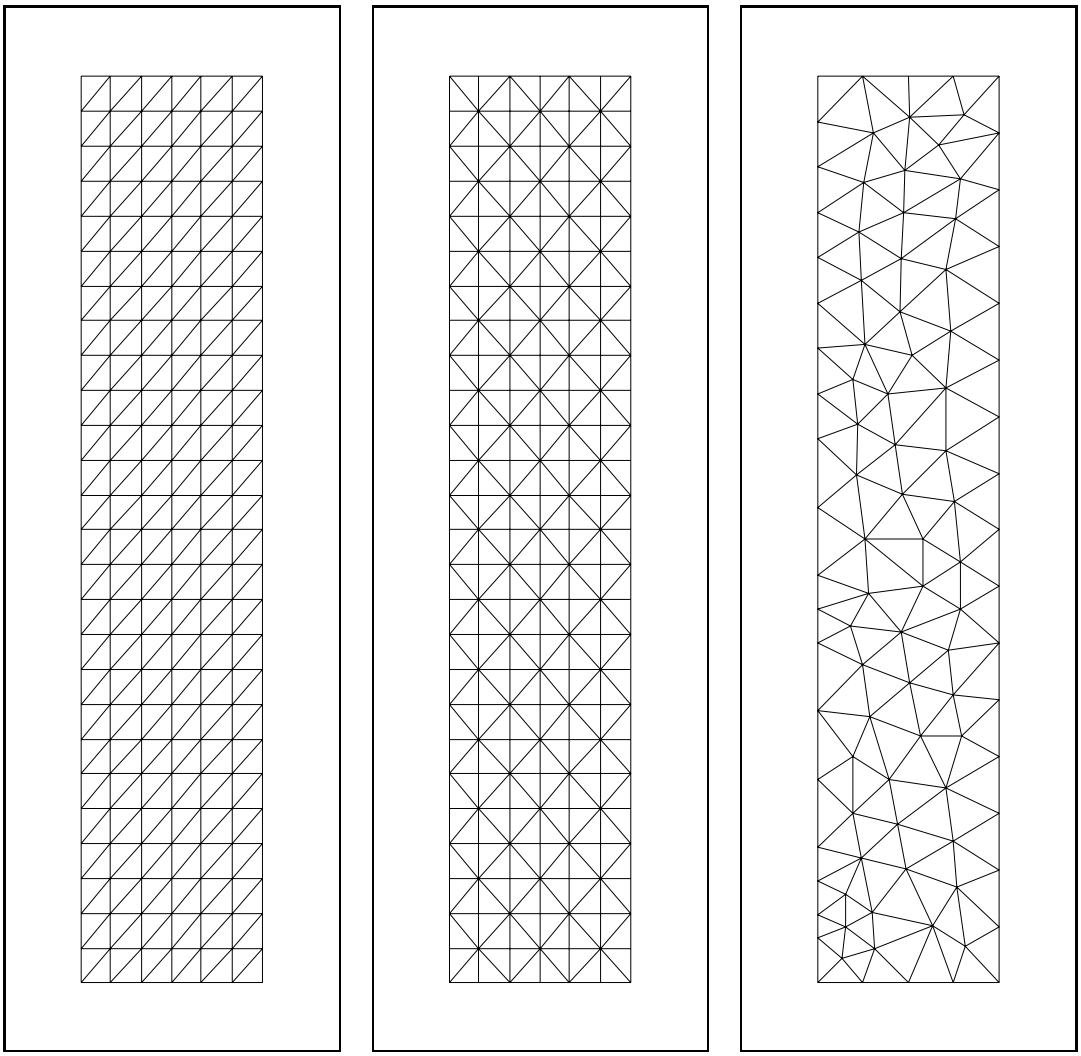
$$c_t \;=\; m_1\, e^{-m_2 t} \tag{5.95}$$

m_1 and m_2 are constants that must be determined by experiment. This term damps the solution throughout the domain for the initial time steps in order that the initial wave front does not appear discontinuous to the time integration software but should not affect the solution obtained after a suitably large amount of time. This damping improves the efficiency of the time integration, which can take larger time steps if the solution is smooth.

5.8 Numerical experiments

5.8.1 One-dimensional periodic waves

Initially the scheme is tested for a monochromatic periodic wave propagating in a constant depth channel parallel to the y-axis. The numerical scheme is tested for three different types of finite element mesh, shown in Figures 5.3(a)-(c), and termed uniform, symmetric



(a) Uniform mesh (b) Symmetric mesh (c) Unstructured mesh

Figure 5.3: Different types of mesh

and unstructured respectively. These tests will allow an assessment of the accuracy of the discretisation and also of any possible mesh dependency effects.

In the first instance the linearised equation system is used. The inflow conditions are then an exact solution of the equation system and are imposed on the inflow boundary as described in Section 5.5.2.

These examples use a monochromatic wave input of the form,

$$\eta(x,t) \;=\; a\,\sin\left(2\pi\left(\frac{x}{\lambda}-\frac{t}{\tau}\right)\right) \tag{5.96}$$

The wavelength λ is calculated such that the linear dispersion relation (2.123) is satisfied for a given period τ and depth h.

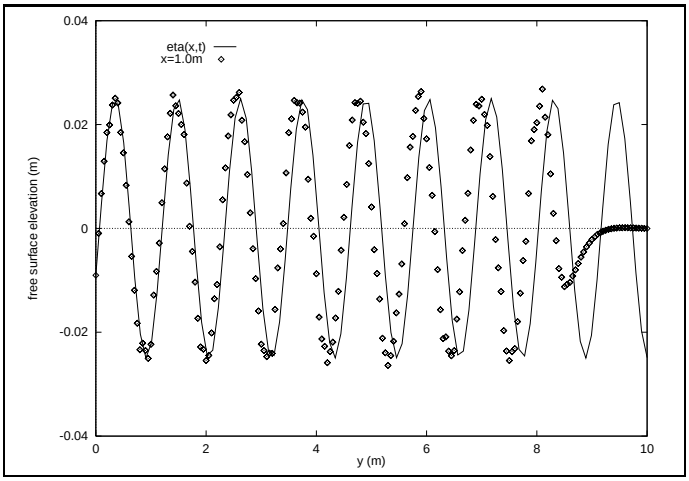
The domain is $(x,y)\in[0,2]\times[0,10]$ with an inflow boundary at $y=0$ and solid walls at $x=0$ and $x=2$. The outflow boundary has an absorbing sponge layer, as described in Section 5.5.4, active for $y>8$.

The flow parameters are,

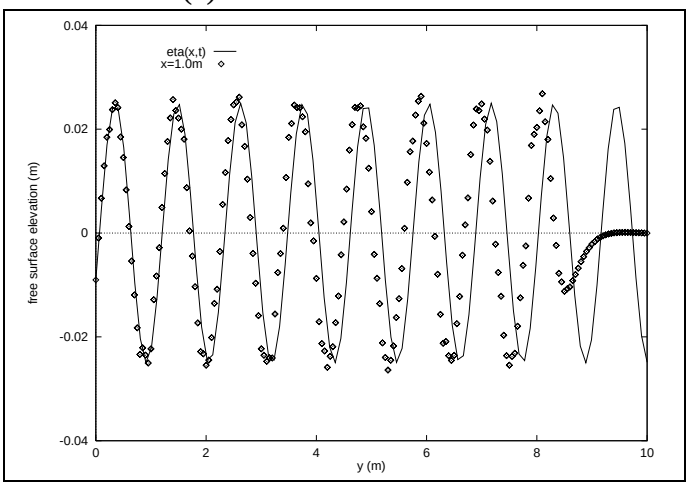
$$h=0.56m \qquad a=0.025m$$
$$\tau=0.85s \qquad \lambda=1.14m$$

Figures 5.4(a)-(b) show the solution at t=40s on 21x101 uniform and symmetric meshes. There are 2121 nodes and 4000 elements, and the wave is resolved with approximately 10 nodes per wavelength in the y direction. The results are very similar for the two meshes and both show a slight phase error. Figure 5.4(c) shows this experiment repeated on a 31x151 symmetric mesh with 4681 nodes and 9000 elements, approximately 15 nodes per wavelength, the phase error is removed and the centreline results agree well with the analytical solution. Figure 5.5 shows however that there are significant differences between the overall solutions obtained on the uniform and symmetric structured meshes. The solution on the uniform grid has become non-symmetric, whereas the solution on the symmetric grid remains symmetric. This mesh dependent effect can be attributed to the non-symmetric stencil at a node on the uniform grid which biases the numerical solution. On the symmetric grid the stencil is symmetric at a node but the form of the stencil alternates from node to node, though this does not appear to significantly affect the solution.

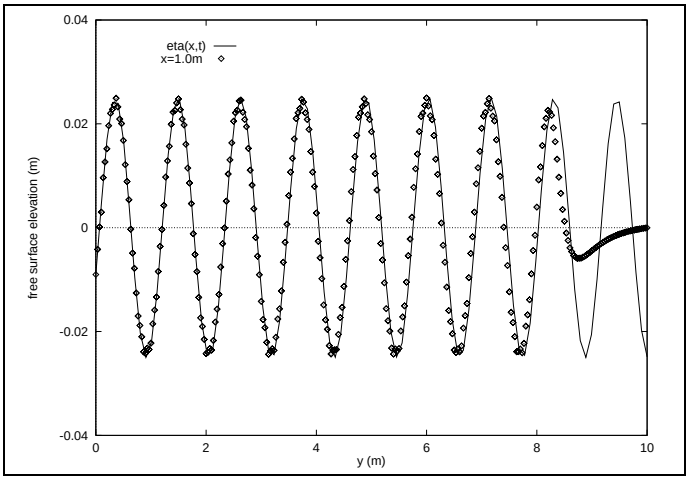
An unstructured grid is produced by moving the nodes of the symmetric mesh slightly, in this case by a random amount up to 5% of the average edge length of the mesh in a random direction. In the one-dimensional case it was shown in Section 4.8 that small perturbations to a uniform grid would not affect the accuracy of the scheme significantly. Repeating the numerical experiment confirms that the solution is not significantly different, although the solution is visibly less regular than on the fully symmetric grid. The mesh connectivity is then altered by swapping edges of the mesh if it reduces the internal angles of the triangles neighbouring that edge, with the nodal positions remaining fixed. Repeating the numerical experiment on this grid shows a much greater difference in the accuracy, with the centreline free surface elevation at t=40s shown in Figure 5.6(a) showing errors in both the phase and amplitude of the wave. This unstructured mesh is



(a) 21x101 uniform mesh

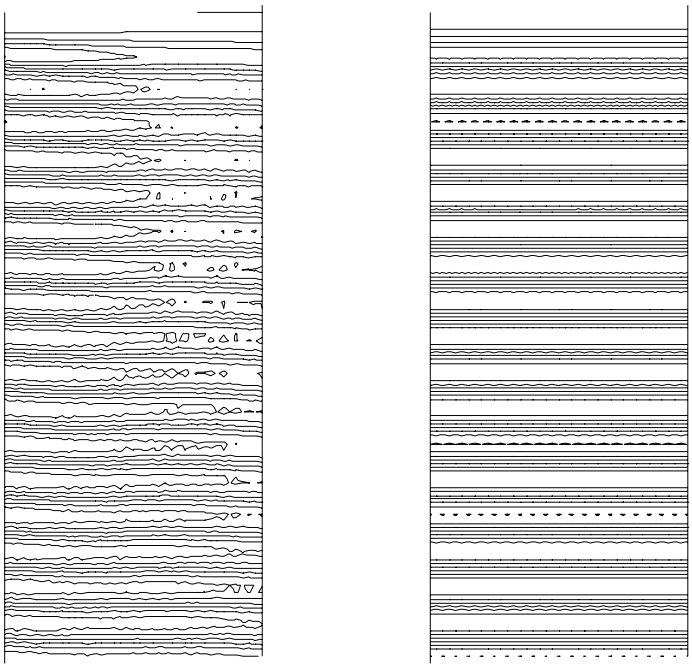


(b) 21x101 symmetric mesh



(c) 31x151 symmetric mesh

Figure 5.4: Centreline free surface elevation at t=40s

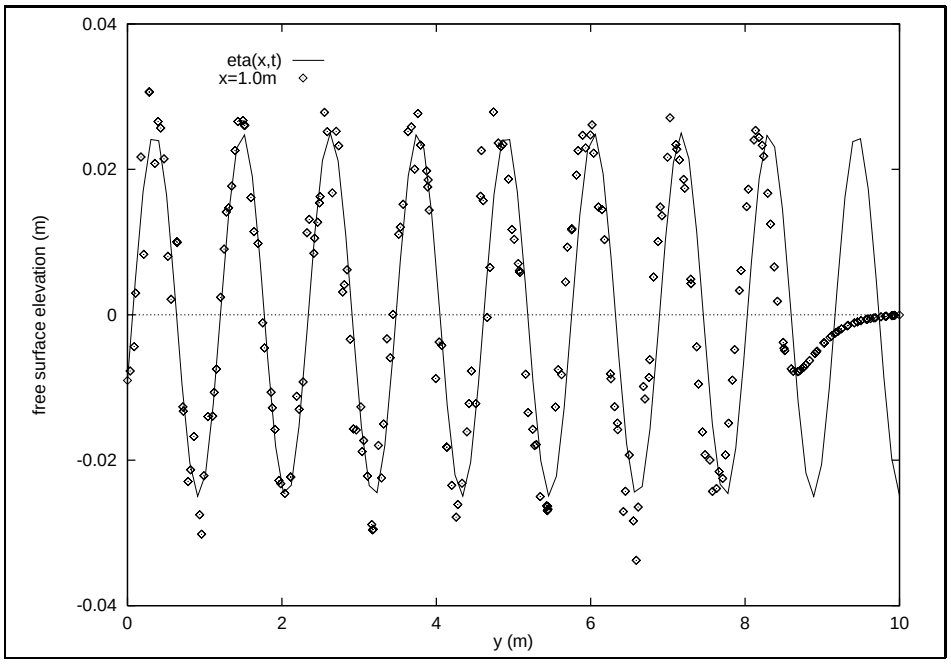


(a) Regular mesh (b) Symmetric mesh

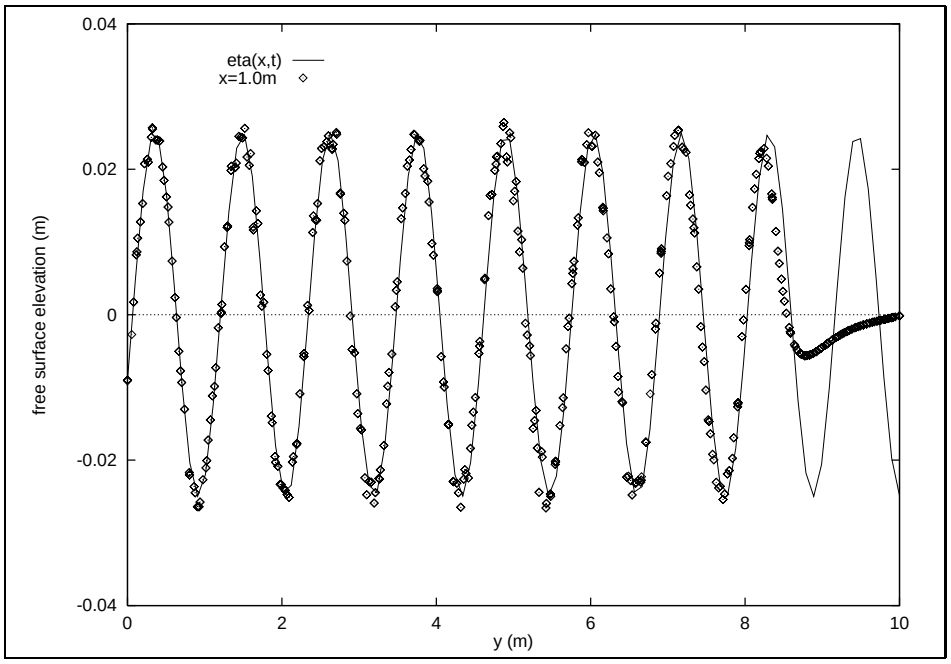
Figure 5.5: Contours of free surface elevation at t=40s

then uniformly refined, doubling the mesh resolution of the wave, and the numerical experiment is repeated. Figure (5.6)(b) shows that the accuracy of the solution is increased and the wave propagation is modelled correctly.

It appears necessary to resolve the domain with a much finer grid if the mesh is unstructured. On a structured grid there will probably be a degree of cancellation of the truncation errors as in the one dimensional case [104]. On an unstructured grid there are lower order truncation errors present and a finer mesh is required to control their size. In one dimension it was shown in Section 4.8 that an irregular grid produced diffusive errors proportional to the change in the mesh size. A similar analysis would be possible, but more complicated, in two dimensions to precisely quantify the suitability of fully unstructured grids for these computations. The numerical experiments also revealed a sensitivity to the connectivity of the mesh, with grids of equal mesh size but differing connectivity producing quite different results. An analysis of the truncation error would also have to take these factors into account.



(a) Coarse unstructured mesh



(b) Fine unstructured mesh

Figure 5.6: Centreline free surface elevation at t=40s

5.8.2 Wave diffraction at a corner

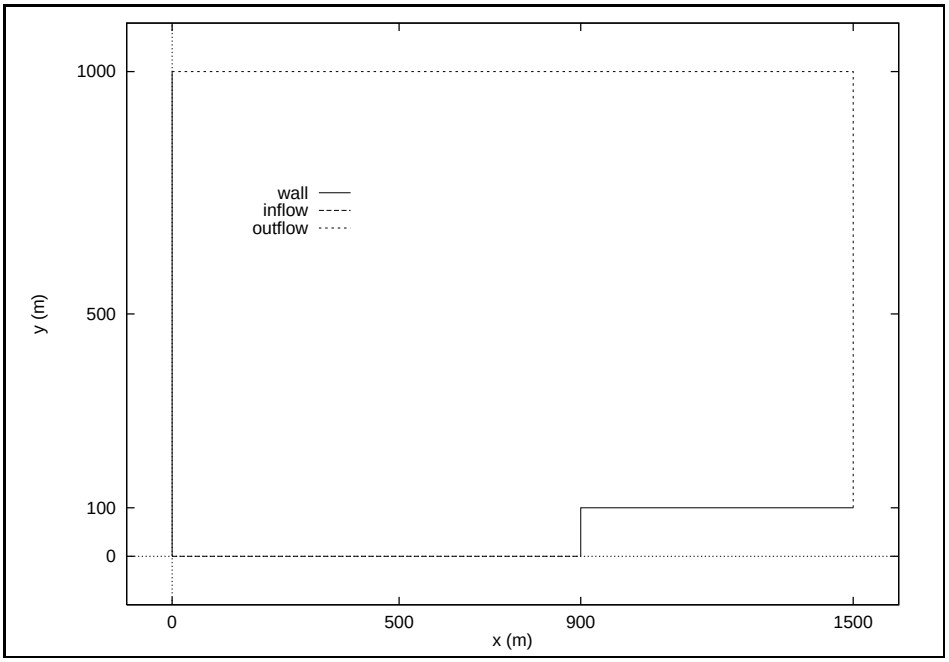
This example considers the diffraction of a wave at a corner. The domain is shown in Figure 5.7(a). There is an inflow boundary at $y = 0$ and solid walls at $x = 0$, $x = 900$, and $y = 100$. The outflow boundaries at $x = 1500$ and $y = 1000$ have an absorbing sponge layer, active for $x > 1300$ and $y > 800$. For an input wave of small amplitude compared to the depth a comparison can be made to the Sommerfeld diffraction solution which is based on linear wave theory [110]. The numerical solution of the extended Boussinesq equations for this problem has been previously reported [66, 97] and the spatial domain and wave conditions used here are the same as those used by Madsen et al. [66].

The inflow parameters used here are those given by Madsen et al. [66],

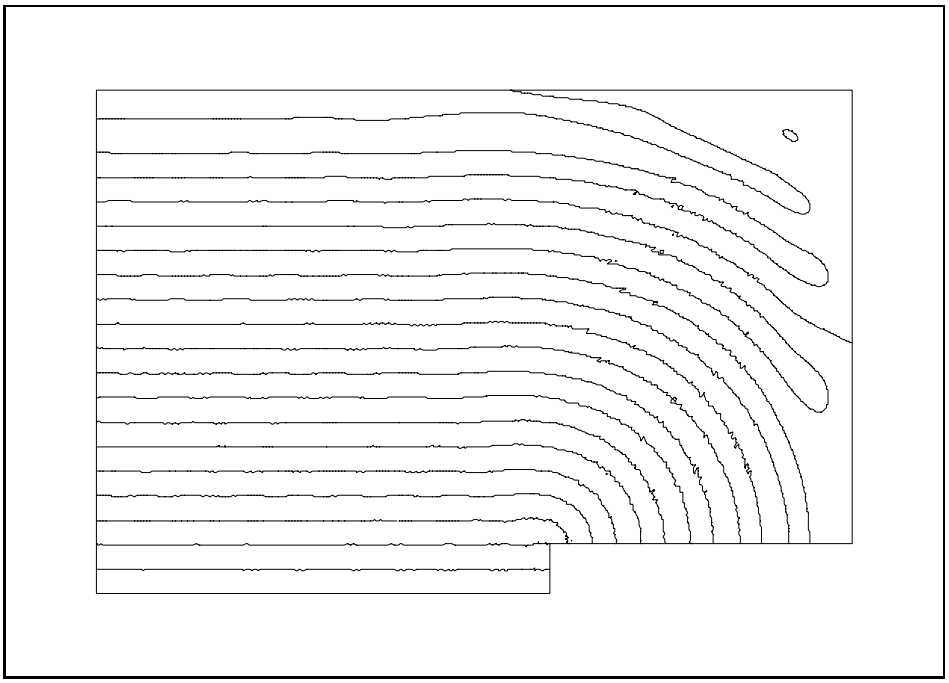
$$\begin{aligned} h &= 40.0m \qquad a = 1.0m \\ \tau &= 8.0s \qquad \lambda = 98.6m \end{aligned}$$

This wave is weakly nonlinear, $\varepsilon = a/h = 0.025$, hence the specification of a linear input wave is sufficient, and the depth is such that $\sigma = h/\lambda \approx 0.4$, implying that the original Boussinesq equation will not be an adequate model.

Initially the domain is discretised with a node spacing of 6.7m giving a resolution of approximately 15 nodes per wavelength in the coordinate directions. Near the corner the mesh is refined by introducing two levels of uniform subdivision of the triangular elements to give additional resolution in this region. The problem is integrated in time for $t \in [0, 200]$ which allows the initial transient wave to have passed through the sponge layer and a steady periodic flow to become established. Figure (5.7)(b) shows the contours of zero free surface elevation at $t = 200s$. It is noted that the results exhibit slight oscillations. These may be due to the corner point, for which the boundary conditions are not consistently specified according to the inviscid theory as discussed in Section 5.5.3. Skotner and Apelt [97] discuss this problem with reference to their finite difference method and use upwind differencing at these points to minimise the oscillations. The finite element analogue of this approach would be to use artificial viscosity at these points. This can be achieved consistently within the finite element framework by using a streamline-upwind Petrov-Galerkin method [51]. Here a simpler approach is adopted and viscosity is added by increasing the value of the viscous coefficient used for the damping layer described in Section 5.5.4. Near a corner the viscous coefficient is specified by a radially decreasing function. The radius of this region, and the strength of the viscosity are user defined and require tuning so as not to significantly damp the flow away from the corner. Note that the implementation of the streamline-upwind Petrov-Galerkin method would also introduce a free parameter that must be tuned for each application. Figure (5.8)(a) shows the contours of zero free surface elevation at $t = 200s$ obtained by this approach, and it is seen that the



(a) Spatial domain for the wave diffraction experiment



(b) Contours of zero free surface elevation at t=200s

Figure 5.7: Wave diffraction at a corner: Spatial domain and initial experiment

oscillations are reduced. Figure (5.8)(b) shows the normalised wave height coefficient, *whc*, calculated at the mesh points. This is defined at a nodal point ip by,

$$whc_{ip} = \frac{\eta_{ip}^{max} - \eta_{ip}^{min}}{2\,a} \tag{5.97}$$

where $2a$ represents the inflow wave height. Comparison with Sommerfeld diffraction theory [110] shows a basic agreement, with the wave height decaying rapidly around the corner, and slightly increasing in height before decreasing to the left of the corner. The Sommerfeld diffraction theory is based on the assumption of a semi-infinite spatial domain whereas the numerical solution includes non-physical inflow and outflow boundaries as indicated on Figure 5.7(a). The oscillations at the left boundary, which is taken to be a solid wall here, and those at the inflow boundary can be attributed to reflections from those boundaries. Comparison with the numerical results of Madsen et al. [66] shows a similar distribution although their results are less oscillatory. They do not show the results for $x < 500$ but note that this boundary must be sufficiently far from the diffraction point. Comparison with the numerical results of Skotner and Apelt [97] shows a very similar distribution of the wave height coefficient away from the inflow boundary.

5.8.3 Internal wave generation

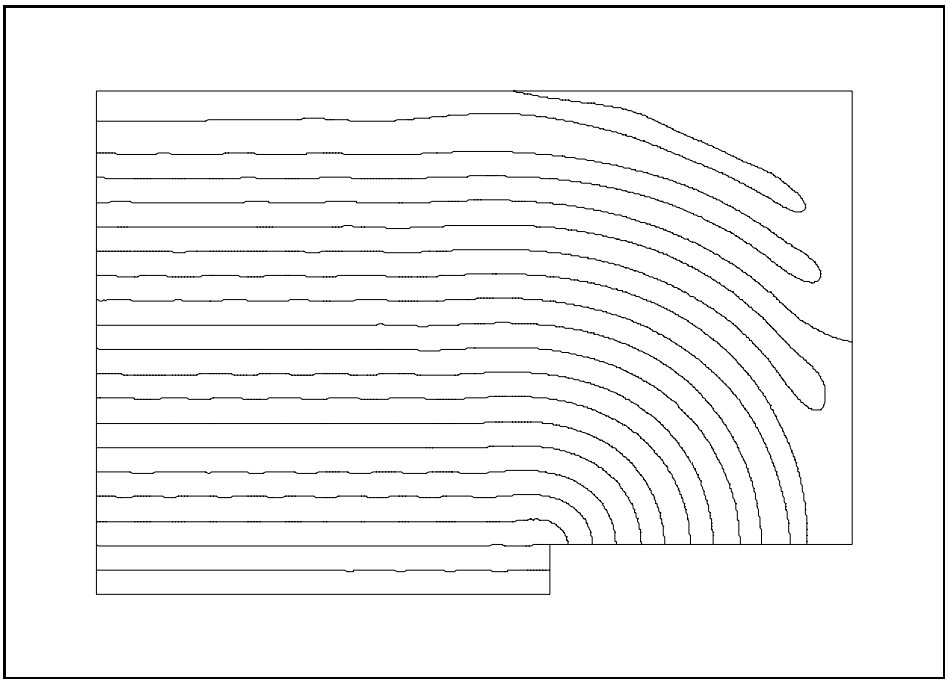
The internal wave generation method described in Section 5.5.5 is tested in a simple constant depth channel. The spatial domain is $(x,y) \in [-2,2] \times [-4,25]$ with the wave generation function centred on $y = 5$ with a width of approximately two wavelengths. The outflow boundaries at $x = -4$ and $x = 25$ have absorbing sponge layers, as described in Section 5.5.4, active for $x < 0$ and $x > 21$ respectively.

The flow parameters are,

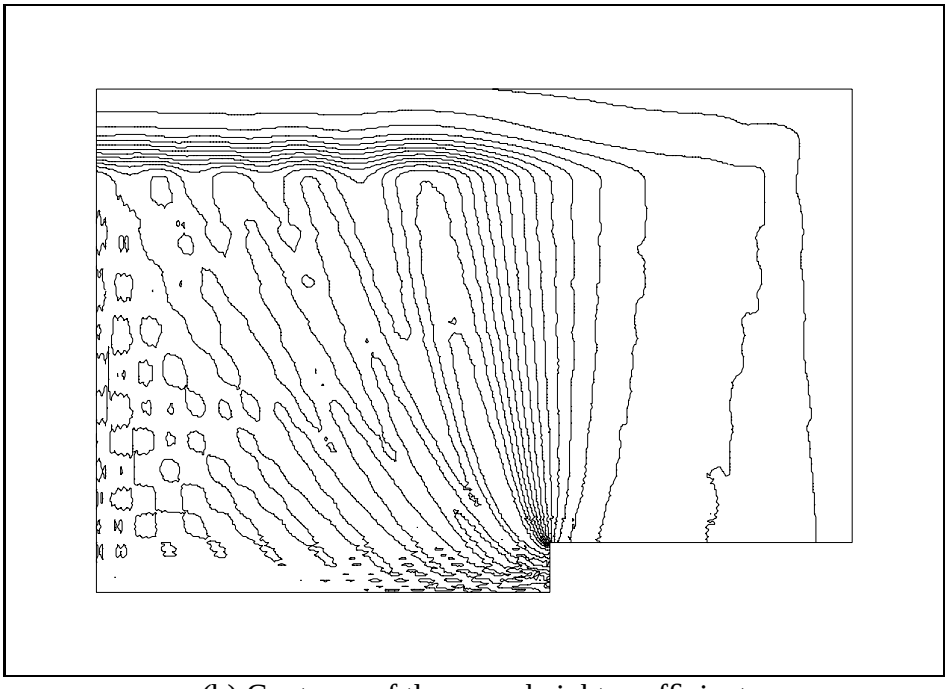
$$\begin{aligned} h &= 0.56m \qquad a = 0.01m \\ \tau &= 0.85s \qquad \lambda = 1.14m \end{aligned}$$

with the wavelength λ calculated to satisfy the linear dispersion relation (2.123). The problem is integrated in time for $t \in [0,40]$.

Figures 5.9(a) and (b) show the free surface elevation along the centreline of the channel at $t = 40s$ computed on 41×291 uniform and symmetric structured meshes respectively. Although these results are very similar the computation on the uniform mesh shows a similar behaviour to that found in the initial wave propagation tests in Section 5.8.1 and the solution becomes non-symmetric as it propagates through the channel. However the computation on the symmetric mesh requires over twice as many linear iterations per timestep, and some linear and nonlinear iteration convergence failures occur during the

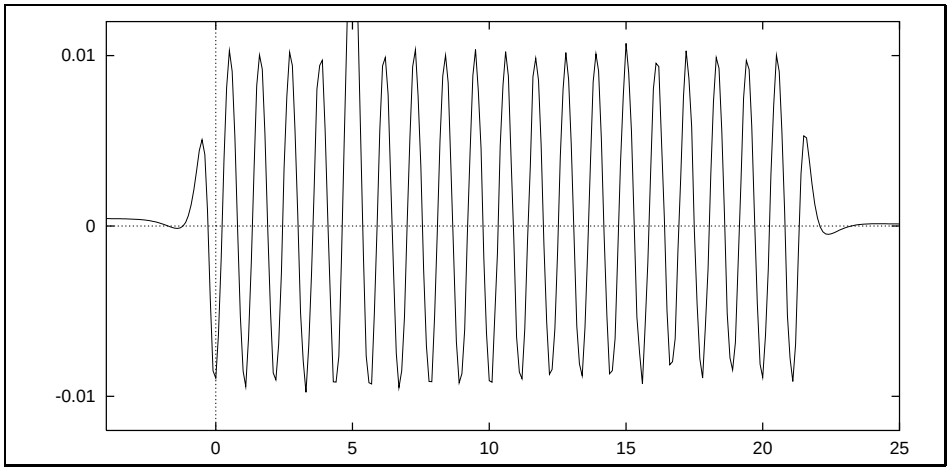


(a) Contours of zero free surface elevation at t=200s

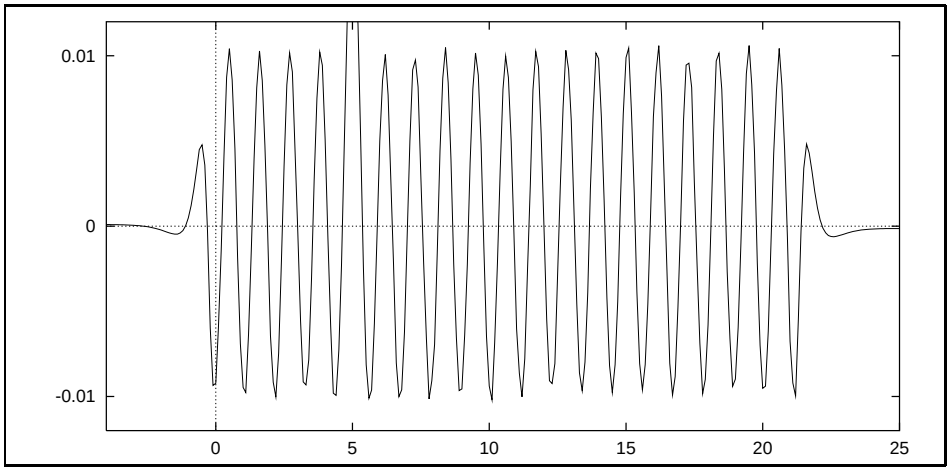


(b) Contours of the wave height coefficient

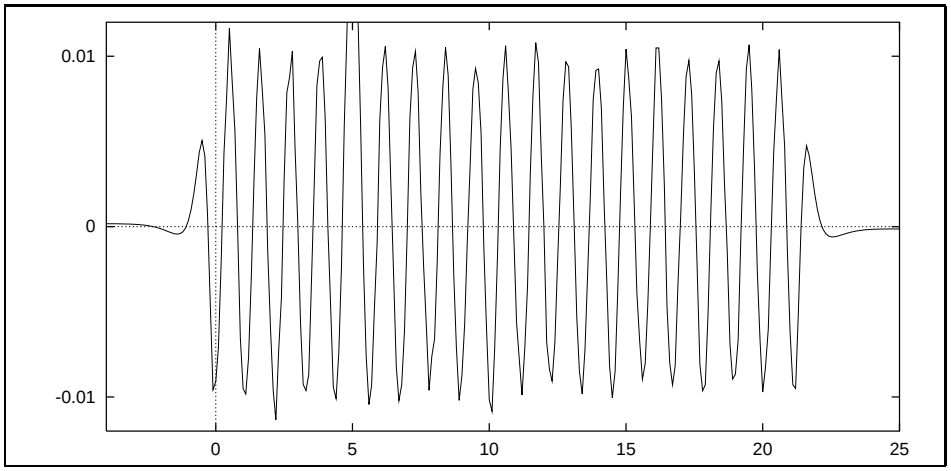
Figure 5.8: Wave diffraction at a corner: Experiments with artificial viscosity



(a) Uniform mesh



(b) Symmetric mesh



(c) Symmetric mesh with uniform wave generation region

Figure 5.9: Centreline free surface elevation for the internal wave generation experiment

time integration process. A possible explanation for this problem is that on the symmetric mesh the support at a node alternates between four and eight surrounding elements and this may produce a slight instability in the scheme. On the uniform mesh the stencil is identical at every internal point. To attempt to remedy this problem the neighbourhood of the wave generation region is discretised with a uniform structured mesh, with the remaining part of the domain discretised with a symmetric mesh. In this case the convergence failures do not occur, and the number of linear and nonlinear iterations required is less than on the symmetric grid, although still greater than on the fully uniform structured mesh. The results for this case are shown in Figure 5.9(c) and although the problems with the convergence of the time integration method are removed the results appear relatively inaccurate compared to those on a fully symmetric mesh.

The use of the wave generation function on a general triangular mesh is an unresolved problem in this project. However it should be possible to use a uniform structured mesh region for the wave generation region and then couple this with a structured or unstructured mesh for the rest of the spatial domain. In general this wave generation region will be required at some distance from the actual region of interest and hence the introduction of a structured mesh here should not compromise the resolution of the physical geometry.

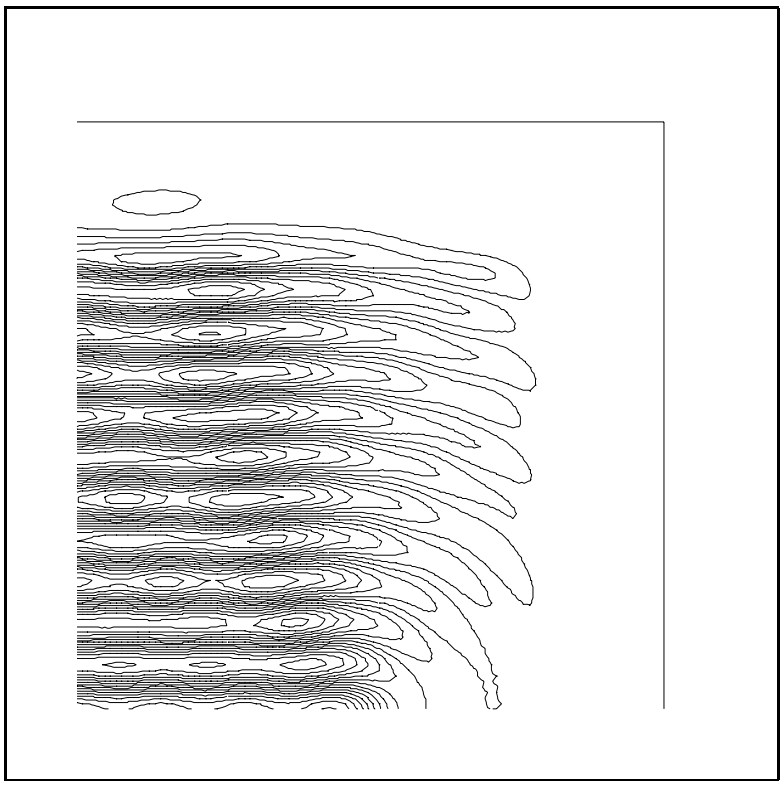
As a first attempt at this a wave generation region is placed inside an open square region, completely surrounded by damping sponge layers. The spatial domain is $(x,y) \in [-14,14] \times [-14,14]$ and is discretised with a 281×281 uniform structured mesh. The wave generation function is centred on $y = 0$ with a width of approximately two wavelengths, and active for $x \in [-6,6]$. All boundaries are considered to be of outflow type and a sponge layer region is specified for $x > 10$, $x < -10$, $y > 10$ and $y < -10$. One of the features of this problem is that the damping region is not terminated by solid walls. Here the generation function is smoothly reduced to zero in the regions $x \in [-8,-6]$ and $x \in [6,8]$. This ensures that the wave generation is not active inside the sponge layer boundary region.

The flow parameters are,

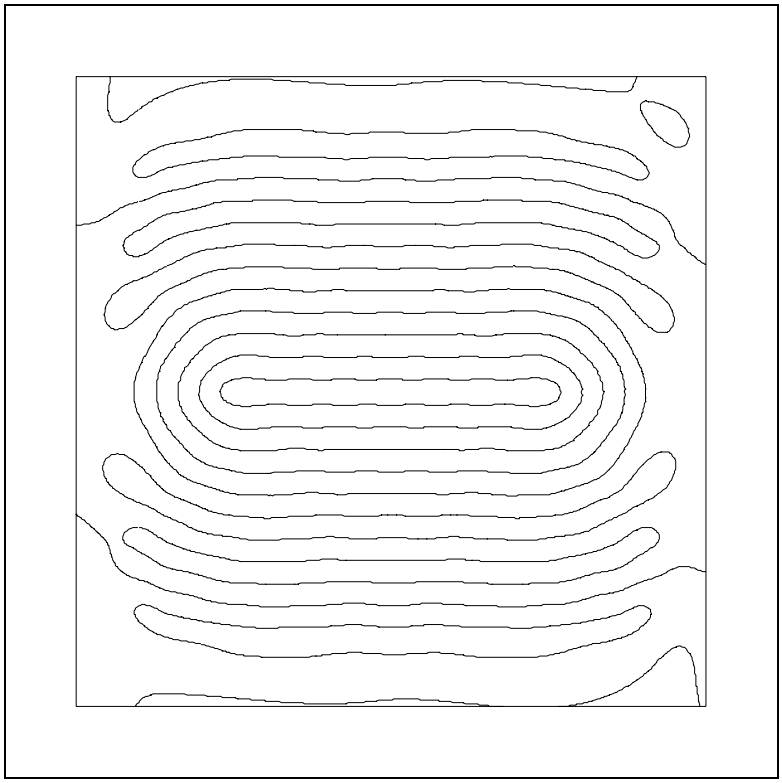
$$h = 0.8m \qquad a = 0.01m$$
$$\tau = 1.14s \qquad \lambda = 2.0m$$

with the period τ calculated to satisfy the linear dispersion relation (2.123). The problem is integrated in time for $t \in [0,20]$.

Figures 5.10(a)-(b) show the computed solution at $t = 20s$. The wave fronts in Figure 5.10(b) show that the generation region produces a periodic wave, however the contours in Figure 5.10(a) show that this primary wave field is contaminated with a secondary field that appears to radiate from the ends of the generation region. Here it is found to



(a) Contours of free surface elevation for one quarter of the domain



(b) Wave fronts over the whole domain

Figure 5.10: Spatial variation of the free surface elevation at t=20s

be impossible to eliminate these two-dimensional end effects from the simulation and the effective, accurate use of this wave generation method is an unresolved problem at present.

5.8.4 Wave focusing by a topographic lens

This variable depth case is a standard test for dispersive wave models [11, 52, 62, 63, 67, 112]. In particular Madsen et al. [67] have shown that the original Boussinesq model is not suitable for the 1s period waves considered here.

The spatial domain is $(x,y) \in [0,6.096] \times [0,35]$ with an inflow boundary at $y = 0$ and solid walls at $x = 0$ and $x = 6.096$. The outflow boundary at $y = 35$ has an absorbing sponge layer, active for $y \in [31,35]$. Figures 5.11(a)-(b) show the bathymetry on the mesh, and along the centre line of the mesh respectively. The depth variation within the domain is given by,

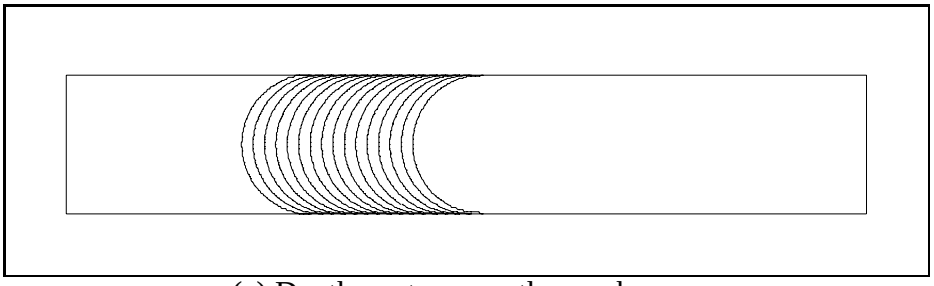
$$
\begin{aligned}
g_x &= \left(6.096\,x - x^2\right)^{\frac{1}{2}} \\
h(x,y) &= \begin{cases} 0.4572 & y \le 10.67 - g_x \\ 0.4572 + 0.04(10.67 - g_x - y) & 10.67 - g_x < y < 18.29d0 - g_x \\ 0.1524 & y \ge 18.29d0 - g_x \end{cases}
\end{aligned}
$$

The inflow parameters are,

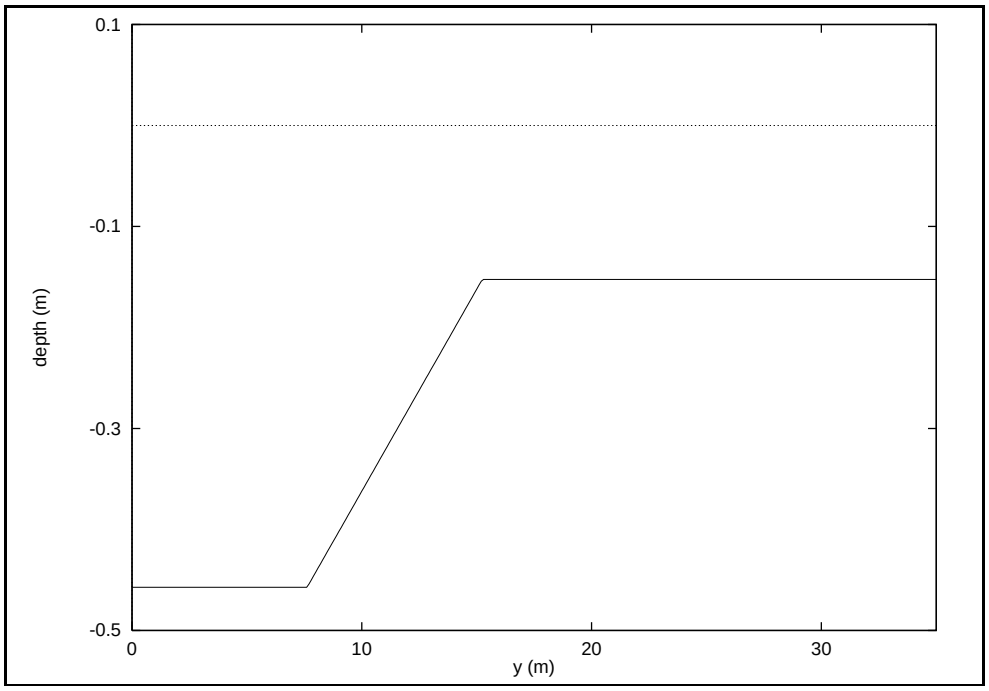
$$
\begin{aligned}
h_{in} &= 0.4572m \qquad & a &= 0.0195m \\
\tau &= 1.0s & \lambda &= 1.488m
\end{aligned}
$$

The nonlinear wave corrections described in Section 4.5.1 are used to correct this wave for errors produced due to nonlinear effects. A 61x351 structured symmetric mesh is used for this problem, giving 21411 nodes and 42000 elements in the mesh. This is a resolution of about 15 nodes per wavelength in the y direction. The problem is integrated in time for $t \in [0,40]$.

Figure 5.12(a) shows the free surface profile along three lines parallel to the y-axis; x=1.016m, 3.048m, 5.080m computed on this mesh. The wave height is amplified along the centre line and decreases near the wall. These results are in good visual agreement with the wave envelope and centreline free surface elevation plots of Madsen et al. [67]. Figure 5.12(b) shows the wave height coefficient distribution over the mesh, calculated according to equation (5.97). This shows oscillations over the mesh which are not apparent in the flow variables themselves. In an attempt to remedy this a small amount of fourth order viscosity is added to the free surface equation (5.12). It is hoped that this fourth order derivative term will not corrupt the physical dispersion present in the

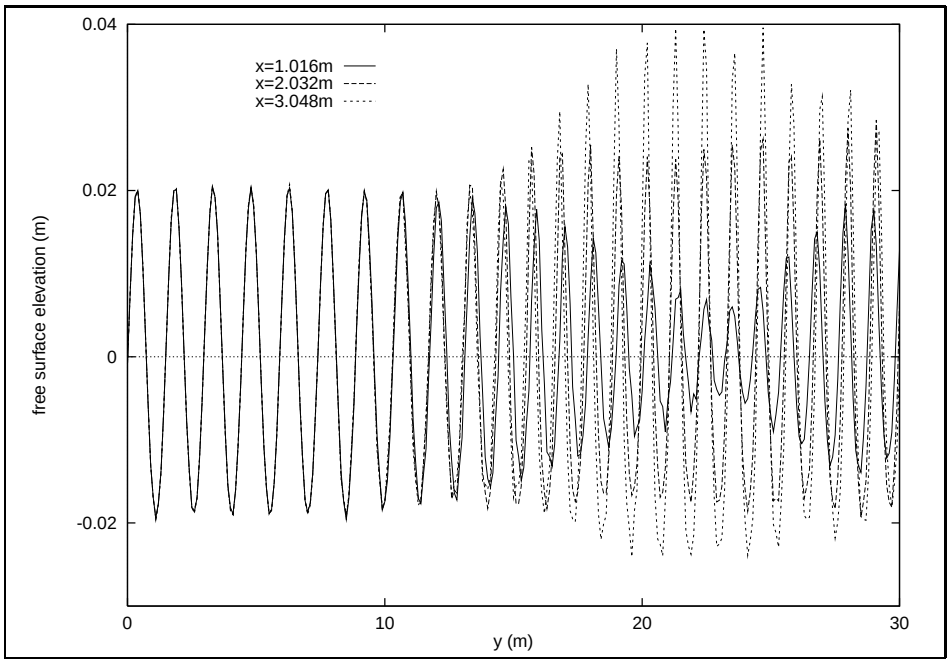


(a) Depth contours on the mesh

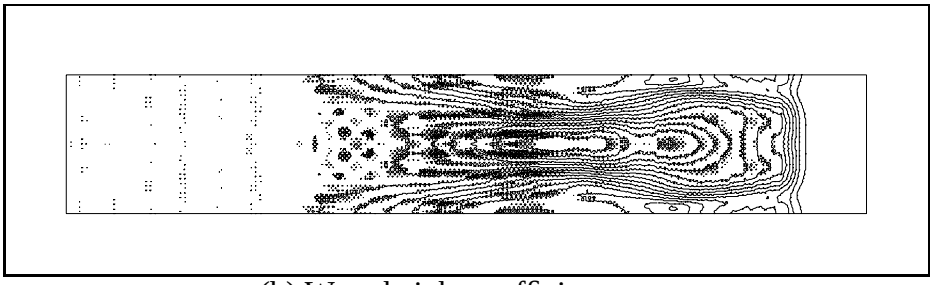


(b) Depth profile on the centreline

Figure 5.11: Bathymetry for the wave focusing experiment



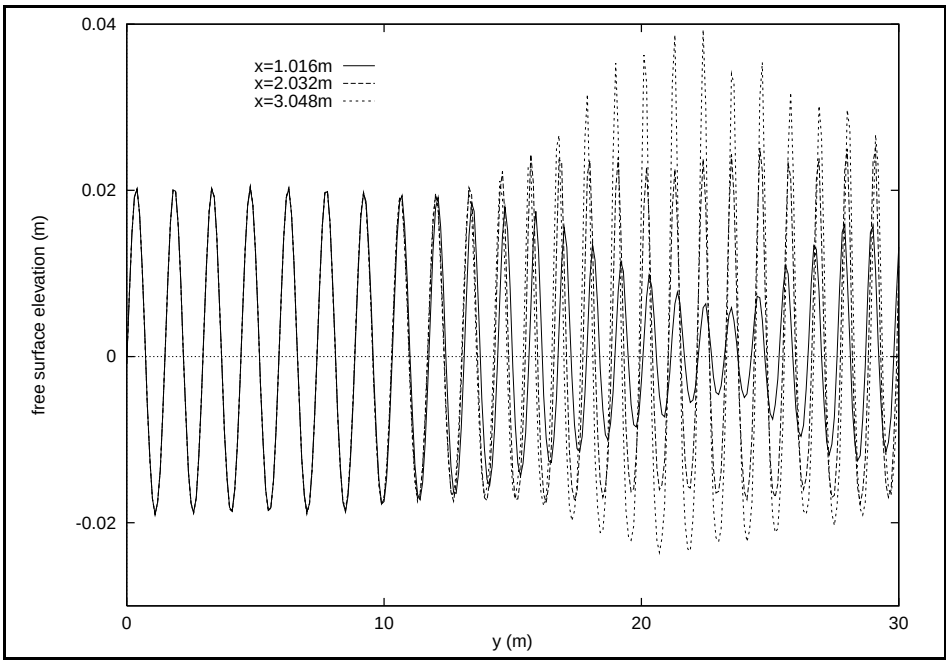
(a) Lengthwise free surface elevations at t=40s



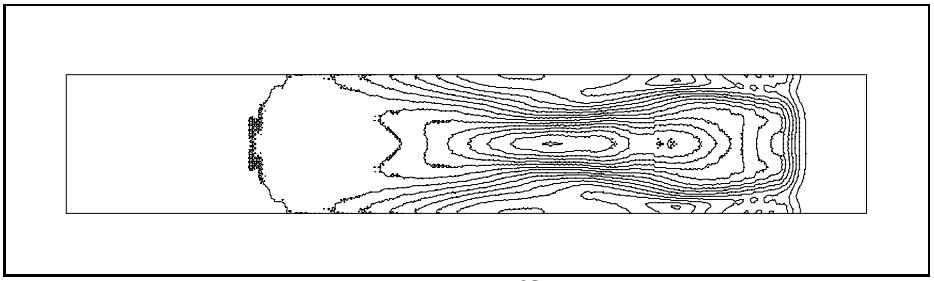
(b) Wave height coefficient

Figure 5.12: Wave focusing experiment on a symmetric mesh

system. The amount of viscosity added is manually tuned and is kept to the minimum possible. The fourth derivative is calculated by first computing a second derivative at the nodes of the mesh and then repeating the process on this value. Alternative procedures are possible [77]; however this procedure is simpler to implement within the existing finite element method. Figure 5.13 shows the same set of results calculated with a small



(a) Lengthwise free surface elevations at t=40s



(b) Wave height coefficient

Figure 5.13: Wave focusing experiment with artificial viscosity

amount of fourth order artificial viscosity. The lengthwise profiles show that the solution has not been significantly damped; however the wave height coefficient distribution now has a much smoother variation over the mesh.

5.8.5 Wave propagation over an elliptical shoal

This problem has been used as a standard variable depth test case for dispersive wave models [52, 106, 108, 112], although the original Boussinesq equations are not appropriate here due to the deep water incident wave conditions.

The domain is $(x,y) \in [-10,10] \times [-10,20]$ with an inflow boundary at $y = -10$ and solid walls at $x = 10$ and $x = -10$. The outflow boundary at $y = 20$ has an absorbing sponge layer, active for $y \in [17,20]$.

The depth variation within the domain is specified by a combination of a 1:50 slope at an angle of 20^o to the y-axis,

$$\begin{aligned} x_r &= \cos(20)\, x - \sin(20)\, y \\ y_r &= \sin(20)\, x + \cos(20)\, y \\ h(x,y) &= \begin{cases} 0.45 & y_r \le -5.82 \\ 0.45 - 0.02(5.82 + y_r) & y_r > -5.82 \end{cases} \end{aligned}$$

and an elliptical bump centred on the origin. For the region,

$$\left(\frac{x_r}{4}\right)^2 + \left(\frac{y_r}{3}\right)^2 < 1$$

the depth is modified by,

$$h(x,y) \Rightarrow h(x,y) + 0.3 - \frac{1}{2}\left(1 - \left(\frac{x_r}{5}\right)^2 - \left(\frac{y_r}{3.75}\right)^2\right)^{\frac{1}{2}}$$

Figures 5.14(a)-(b) show the bathymetry on the mesh, and along the centre line of the mesh respectively. The flow parameters are,

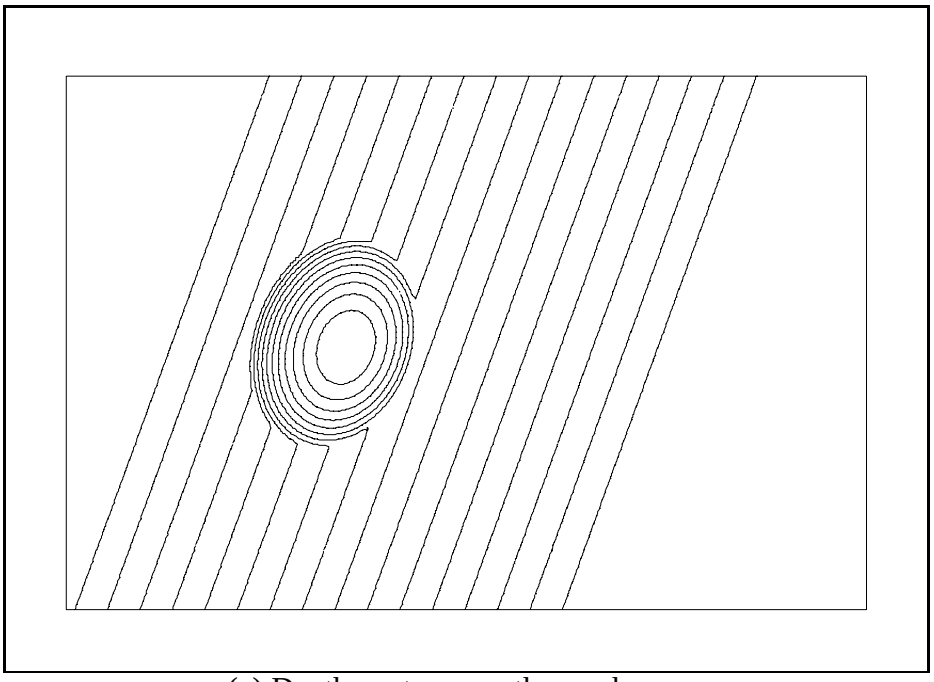
$$\begin{aligned} h_{in} &= 0.45m \qquad & a &= 0.0232m \\ \tau &= 1.0s \qquad & \lambda &= 1.485m \end{aligned}$$

where the wavelength λ has been calculated to be consistent with the linear dispersion relation at the inflow boundary.

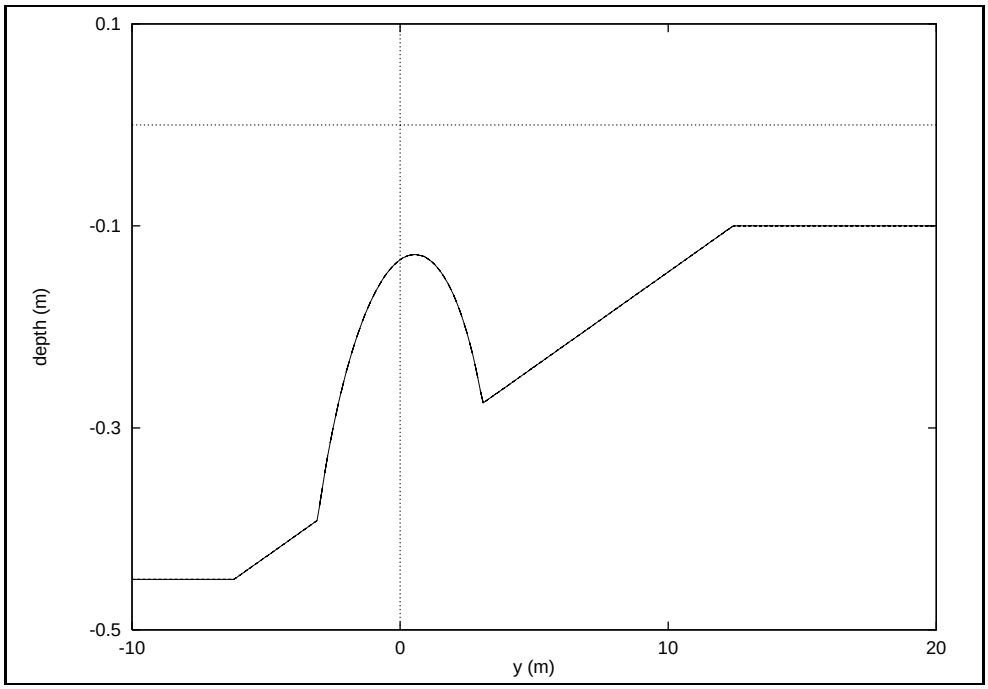
Measurements of wave height coefficient, as defined by equation 5.97, are taken along the following coordinate sections;

$$\{(x,y_m), y_m = 1.0, 3.0, 5.0, 7.0, 9.0\}, \{(x_m,y), x_m = -2.0, 0.0, 2.0\} \tag{5.98}$$

This data can be compared with published experimental data and previous numerical simulations with finite difference methods [106, 112] and a frequency domain parabolic model [52].



(a) Depth contours on the mesh

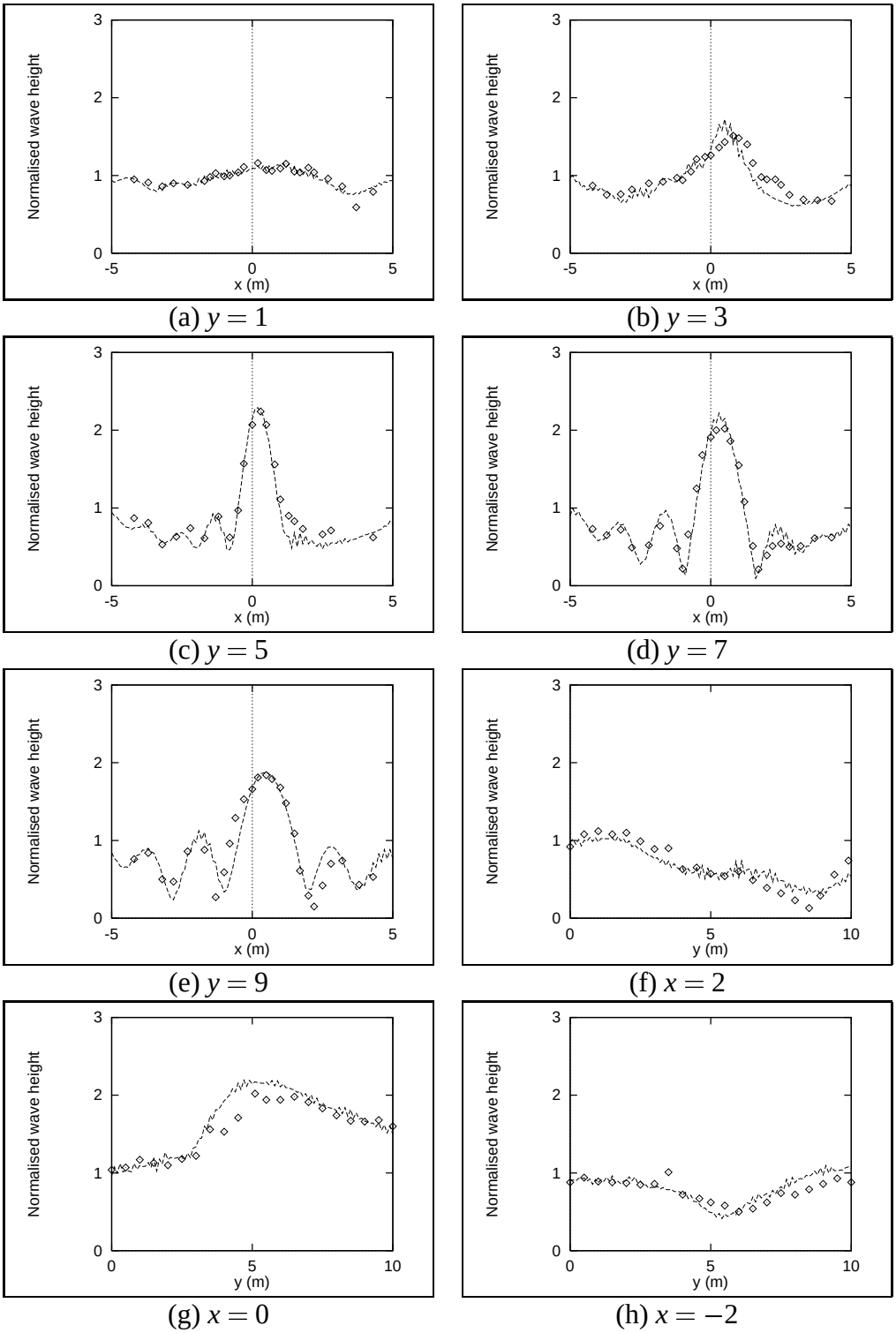


(b) Depth profile on the centreline

Figure 5.14: Bathymetry for the elliptical shoal experiment

The spatial domain is partitioned with a symmetric mesh with a uniform nodal spacing of 0.1m giving a resolution of approximately 15 points per wavelength at the inflow boundary. This produces a mesh with 60501 points and 120000 elements. The computation is performed for $t \in [0, 50]$ and the wave height data is extracted for $t \in [42, 44]$ by which time the initial transient has passed out of the domain and a periodic wave motion has been established.

Figure 5.15(a)-(h) shows the wave height coefficients on each of the coordinate sections indicated previously. It is noted that while the main features are accurately represented there is evidence of noise in the numerical data in the form of node to node oscillations. This instability is not apparent in the basic flow variables but only in the wave height coefficient which is derived from observations of the free surface elevation at a point over time. This quantity is often used in the analysis of wave problems hence it is important to investigate the cause and possible remedies for this problem. It is found that these oscillations can be reduced by the addition of a small amount of viscosity. For this case a second order viscosity term is added to the free surface equation (5.12) with coefficient $1.0e^{-4}$. The implementation of this term is identical to that of the sponge layer described in Section 5.5.4 and adds a uniform amount of viscosity across the whole spatial domain for all time. Figures 5.16(a)-(h) show the wave height coefficients on the same coordinate sections as Figure 5.15 and it is seen that the oscillations are reduced without a significant reduction in the overall values. Comparison of these results with those previously published shows a much better agreement with the experimental data. Wei and Kirby's finite difference method [106] predicts a wider central focusing region in Figures 5.16(c)-(e) and does not capture the secondary peaks very accurately. Their results for the sections shown in Figures 5.16(f) and (h) show large differences near $y = 10$ and also a large deviation from the experimental data on the section shown in Figure 5.16(g) in a region just behind the elliptical bump. The results of Zang et al. [112] produced with a similar finite difference scheme show an improved model of the central focusing regions in Figures 5.16(c)-(e) but a relatively poor representation of the secondary peaks on those sections. Their results for the section shown in Figure 5.16(g) appear more accurate than Wei and Kirby's although the two other lengthwise sections are not shown in that work. In a recent publication Wei et al. [108] have solved this problem again using the same finite difference scheme with an internal wave generation method for the incident wave. These results are very similar to those presented here although they do not appear to have any oscillations in their data. However they note that the results are calculated by averaging the results over four wave periods which may remove such high frequency noise. This problem has also been modelled with a frequency domain parabolic model [52] although the authors used linearised Boussinesq equations and the results show relatively poor agreement for the secondary peaks in Figures 5.16(c)-(e) compared to those given



(a) $y = 1$ (b) $y = 3$

(c) $y = 5$ (d) $y = 7$

(e) $y = 9$ (f) $x = 2$

(g) $x = 0$ (h) $x = -2$

Figure 5.15: Wave height coefficient sections for the elliptical shoal experiment. (Line - numerical. Data points - experimental)

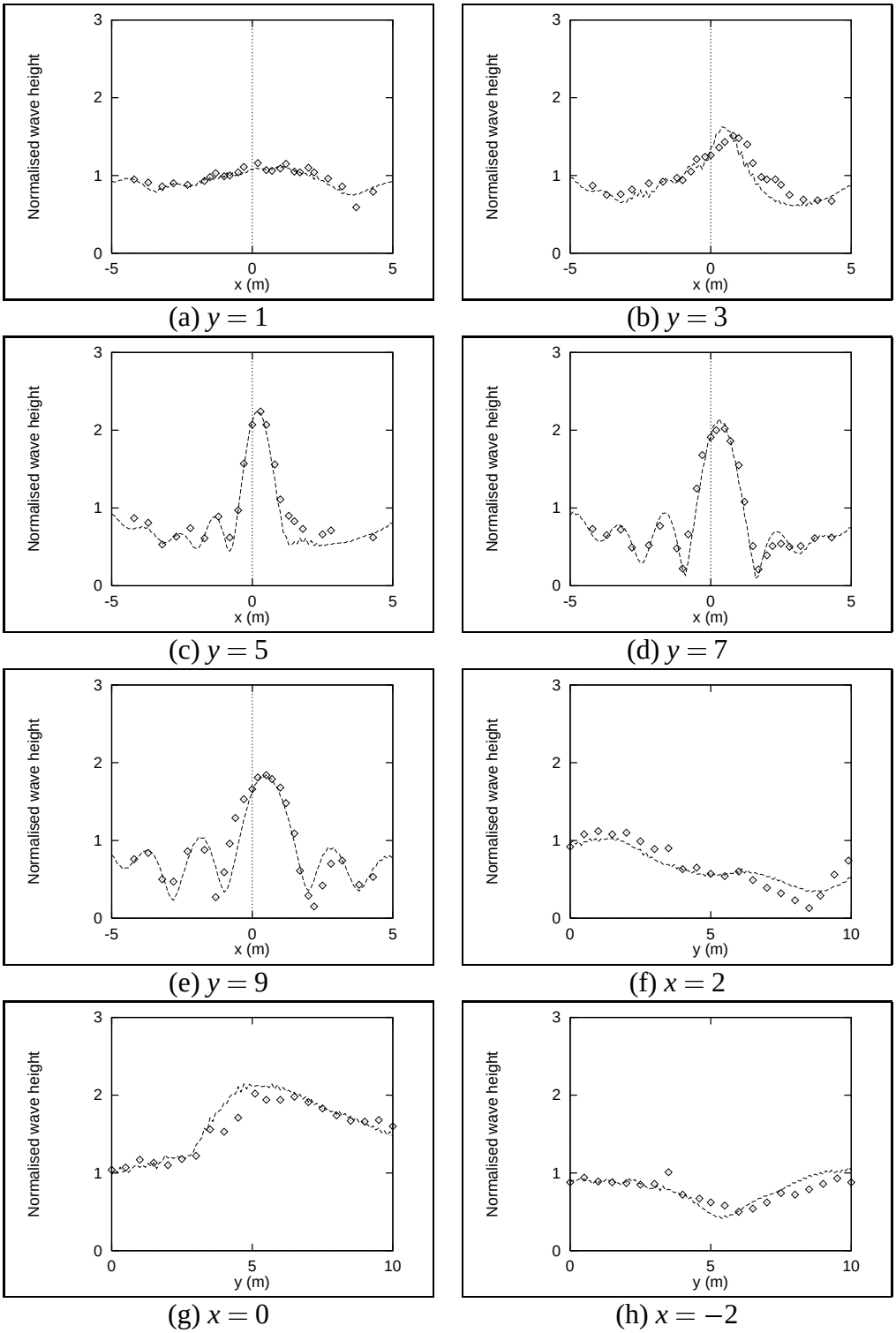


Figure 5.16: Wave height coefficient sections for the elliptical shoal experiment with artificial viscosity. (Line - numerical. Data points - experimental)

here.

5.8.6 A prototype harbour geometry

The motivation for this research project was to model the wave disturbance inside and around harbours. Previously published studies using finite difference models for such applications were reviewed in Section 5.2. To the authors knowledge there have been no finite element Boussinesq models applied to such problems.

At many stages of the definition of the computational problem compromises must be made between the real situation and what can be represented within the numerical model. This process is outlined below with respect to the specific case considered here which is based on a HR Wallingford case study of a real harbour [15].

The exact geometry data supplied with the case study is shown as the solid boundary on Figure 5.17. The computational domain used is fully specified by introducing addi-

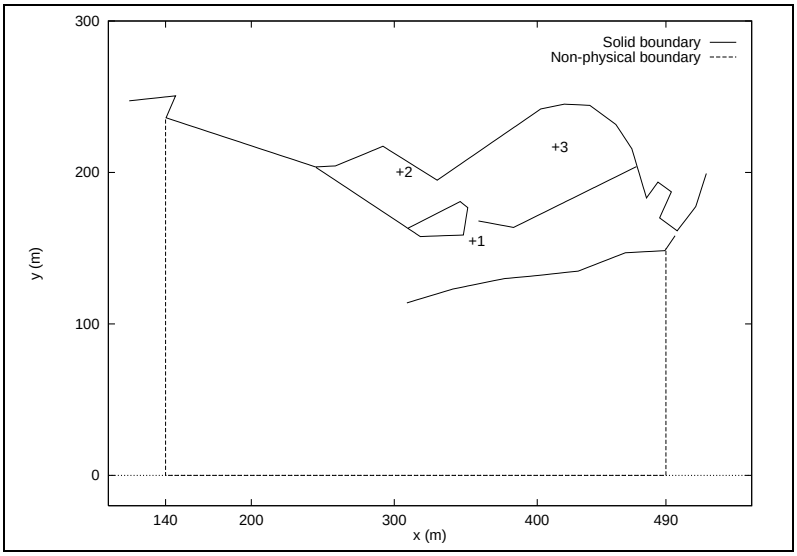
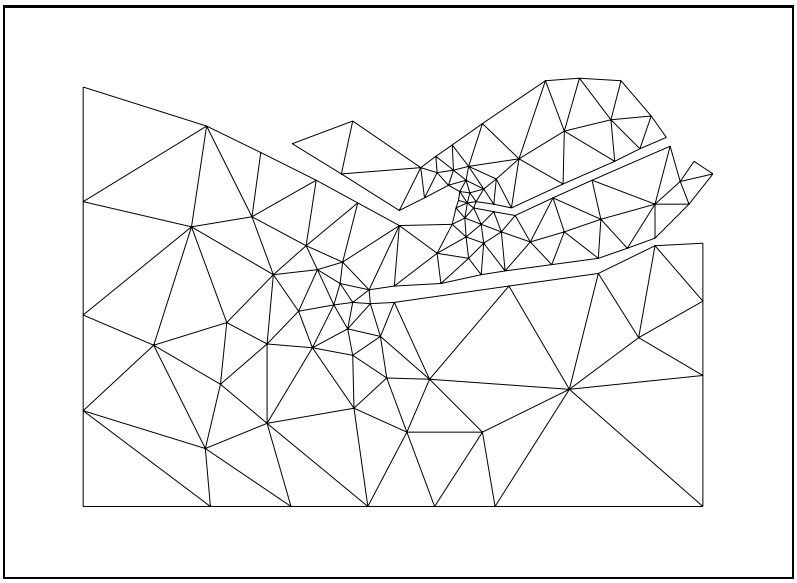


Figure 5.17: Spatial domain for the harbour geometry

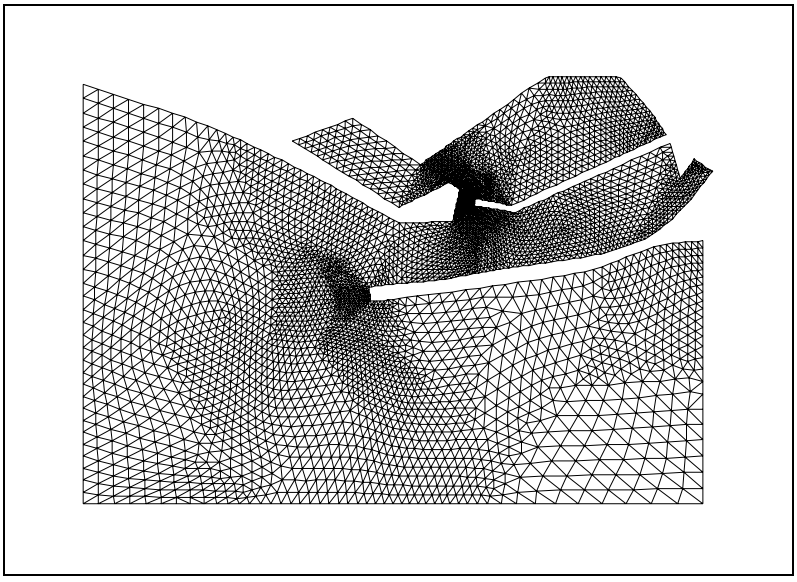
tional seaward boundaries as shown on the figure. These are chosen to be suitably far from the region of interest, but are constrained by the expected size of the finite element grid which must not be too large for practical computations.

The previous experiments with unstructured triangular grids showed that the numerical method employed here can be sensitive to irregularities in the mesh. For the irregular domain considered in this case a grid must be generated automatically. Here this is achieved with the Delaunay mesh generation software Triangle [96]. However it is found that the mesh produced is fairly irregular and as such unsuitable for use with this numerical method. Here the mesh generation procedure adopted is to use Triangle to generate a coarse unstructured grid, suitable for representing the geometry, and then to post-process

the mesh by successively applying coordinate smoothing and edge swapping to "regularise" the mesh. The mesh is then uniformly subdivided to produce a finer grid and that grid is then post-processed with the same procedures. This is repeated until a suitable mesh size is reached, in this case a resolution of the expected inflow wave of approximately 20 nodes per wavelength. Figures 5.18(a)-(b) show the initial mesh generated by



(a) Initial mesh generated by Triangle



(a) Intermediate mesh generated by the post-processing

Figure 5.18: Unstructured triangular meshes for the harbour geometry

Triangle and an intermediate mesh produced by the post-processing after three levels of uniform refinement. Note that the boundary definition shown in Figure 5.17, taken from

the report, is modified slightly by prescribing a finite thickness to the boundary regions. A similar procedure is carried out for the HR Wallingford case study although the exact dimensions will not be identical. In the computational model used here all physical boundaries are assumed to be vertical and perfectly reflecting, although this was not the case in reality [15]. The final mesh produced contained 167936 elements and 85057 nodes with a resolution at the inflow boundary of approximately 24 points per wavelength. The mesh bandwidth was reduced to 350 by the Reverse Cuthill-McKee algorithm

For the initial experiment a constant depth is assumed over the whole region and the simple monochromatic waveform (5.96) is input at the lower boundary. The flow parameters are,

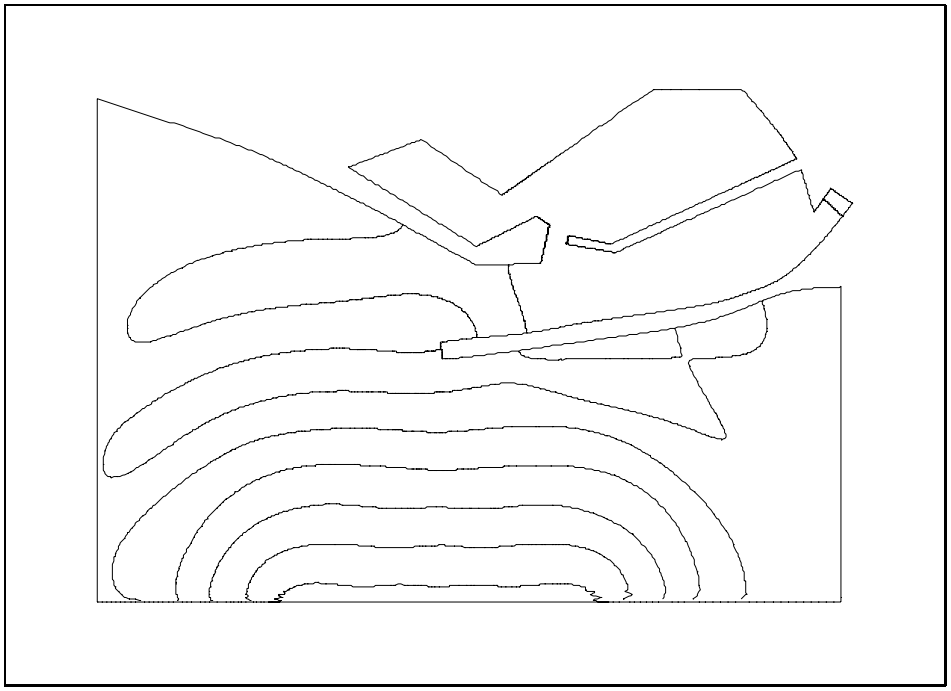
$$\begin{aligned} h_{in} &= 7.0m \qquad a = 1.0m \\ \tau &= 5.3s \qquad \lambda = 36.5m \end{aligned}$$

where the wavelength λ has been calculated to be consistent with the linear dispersion relation at the inflow boundary. The inflow wave is specified at $y = 0$ for $x \in [220, 420]$. Sponge layers are used to remove any outgoing waves at the sides of the domain. They are specified as described in Section 5.5.4 for the regions $x \in [140, 220]$ and $x \in [420, 490]$. The inflow wave is specified so as not to be active inside the sponge layer regions by smoothly decreasing the wave amplitude to zero as it approaches either layer. Artificial viscosity is used in the neighbourhood of sharp corners to prevent any non-physical oscillations developing in the solution.

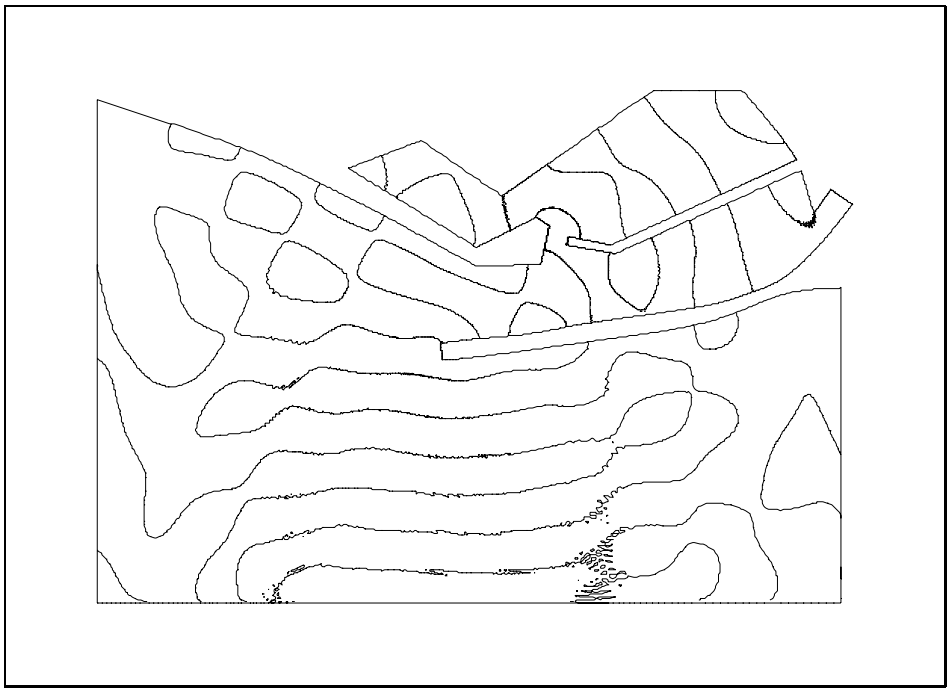
Figure 5.19(a) shows the solution wave fronts at $t = 25s$, by which point the input waves have reached the harbour but the reflected waves are not significant. At $t = 50s$, shown in Figure 5.19(b), wave reflection from the solid boundaries has propagated back to the inflow boundary and is contaminating the solution.

Time histories of the free surface elevation at three points within the harbour, $(x,y) = \{(352,155),(301,200),(410,217)\}$ indicated on Figure 5.17, are shown in Figure 5.20, A periodic flow is developing but the simulation cannot be run far enough for a steady state to be established due to the restrictions of the inflow boundary conditions.

The obvious remedy to this problem is to remove the inflow boundary and instead use the internal wave generation method described previously, with damping layers at all the non-physical seaward boundaries to absorb the outgoing waves. However the lack of robustness in the internal wave generation method meant that here it was not possible to generate useful results.



(a) Wave fronts at $t = 25s$



(b) Wave fronts at $t = 50s$

Figure 5.19: Contours of zero free surface elevation

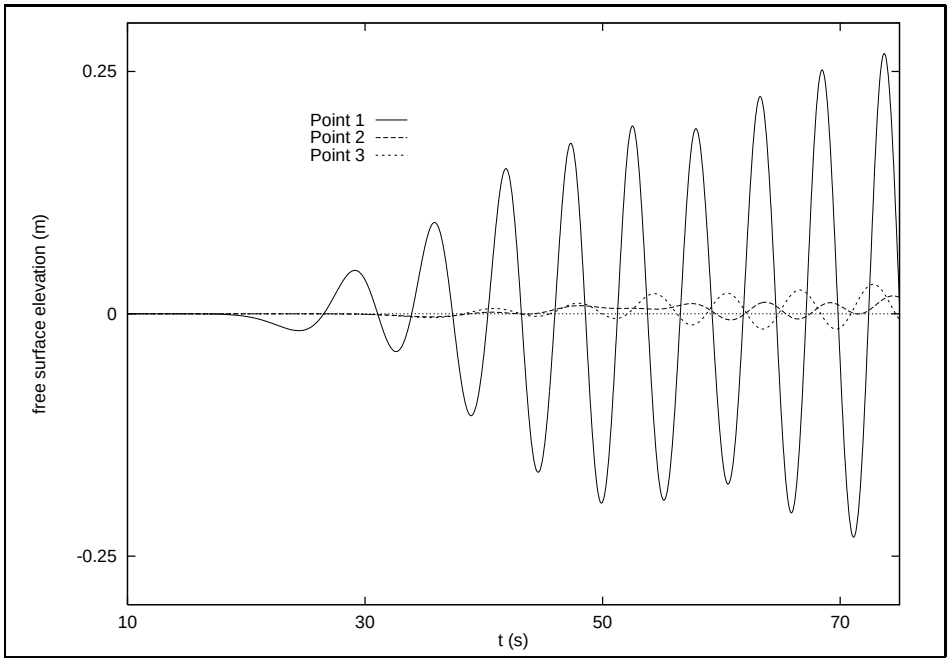


Figure 5.20: Time histories of the free surface elevation inside the harbour

5.9 Discussion

A new finite element element method for Nwogu's two-dimensional extended Boussinesq equation system has been described and implemented in this chapter, extending the algorithm described for the one-dimensional equation system in Chapter 4. Boundary condition are described in Section 5.5 including a novel formulation for total reflection at a solid wall and a new technique for the generation of waves internally to the domain. The implementation of the time integration algorithm is considered in Section 5.6. For greater efficiency, sparse preconditioned iterative techniques are employed to solve the linear equation system within each time step. Mesh renumbering algorithms are discussed and the Reverse-Cuthill McKee algorithm is implemented to increase the efficiency of the iterative method.

The initial numerical experiments described in Section 5.8.1 showed that there was significant mesh dependence in the results when structured triangular grids are used. In particular the method is sensitive to the orientation of the diagonals and if this is done in a non-symmetric way the solution will also become non-symmetric. Experiments using unstructured triangular grids showed that greater spatial resolution is required in order to achieve a comparable accuracy and that the mesh must be made as smoothly varying as possible if the time integration algorithm is to work efficiently. The wave diffraction experiment described in Section 5.8.2 showed that sharp corners in the domain could produce non-physical oscillations in the solution. Artificial viscosity was used in the neighbourhood of the corner to control the oscillations. A small amount of artificial viscosity

was also introduced into the scheme to prevent high frequency oscillations such as those observed in the elliptical bump experiment in Section 5.8.5. At present these amounts of viscosity have to be chosen by repeated experiments, so as to remove the non-physical behaviour without affecting the global solution, and it will require more research to find a reliable way of choosing the appropriate coefficients. It is possible that a more sophisticated finite element method may be required, such as the streamline-upwind finite element method [51] which adds viscosity selectively to the discretisation. However this also introduces a free parameter which requires tuning and standard values will almost certainly be too dissipative for this numerical scheme. When the previously described techniques are used on symmetric structured triangular grids the method produces accurate results for the standard dispersive wave test cases. However if internal wave generation is required, as an alternative to specifying a wave form at inflow boundaries, the generation region must be discretised with a uniform structured mesh for greatest efficiency. This method does not appear as reliable in two dimensions and the performance of this boundary condition appears to be highly mesh dependent.

Several physical and numerical approximations are required in order to simulate a realistic practical problem. The spatial domain must be approximated; here the use of unstructured triangular grids allows an accurate representation of this. However it has been found necessary to pre-process the unstructured mesh to ensure that the mesh size is smoothly varying and control the mesh dependent effects. This is achieved here with stages of standard coordinate smoothing and edge swapping combined with uniform refinement of an initial coarse unstructured grid. The simulation of the harbour showed that while the basic flow algorithm could transport the wave over the unstructured mesh and into the harbour the computation was limited by the interaction of the inflow boundary and waves reflected from the solid boundaries within the domain. The lack of robustness in the internal wave generation method considered here meant that it was not possible run the model for a long enough time to extract meaningful results inside the harbour.

It has been shown in this chapter that the proposed finite element scheme can provide an accurate model for the extended Boussinesq equations. However the mesh dependent effects, both for the basic scheme and for the internal wave generation, and the artificial viscosity coefficients that must be added in to stabilise the scheme, require further research if the method is to be used on practical problems with confidence.

Chapter 6

Conclusions

6.1 Summary

In this thesis the numerical approximation of nonlinear, dispersive Boussinesq wave equations was considered. In this field there is still active research on finding the best mathematical model for this class of problem, as well as on their numerical approximation, and in Chapter 2 the Boussinesq wave theory was reviewed in detail. It was shown that there are a variety of equation systems termed Boussinesq-type and that they can be compared by considering their linear dispersion characteristics. A compromise must be reached between the theoretical accuracy of the mathematical model and the practicalities of solving the equation system numerically. In this work a system of extended Boussinesq equations due to Nwogu was selected. Their mathematical properties were compared to the original Boussinesq equation system and it was shown that they were significantly more accurate. While further enhancements to the mathematical model are possible, and many have been proposed in the literature, it was decided here that the extra complications to the differential form of the equations would overly complicate the numerical approximation.

Numerical methods for nonlinear, dispersive wave equations were considered in Chapter 3, in particular finite difference and finite element spatial discretisations coupled with adaptive order, adaptive step size time integration software. These methods were implemented for two model equations; the RLW equation and the KdV equation, which represent the simplest possible nonlinear, dispersive equations. A finite difference method was developed based on a previously published scheme for the one-dimensional extended Boussinesq equations. It was shown that this method was accurate for both equations. A

new linear finite element scheme was then proposed and it was shown, both theoretically and by experiment, that this scheme was also sufficiently accurate for these equations. The KdV equation contains third order spatial derivatives and it was shown here that, by rewriting the equation in a lower order differential form, the linear finite element method could be applied. The efficiency of these spatial discretisation techniques in combination with the time integration software was considered in detail and it was shown that, by introducing further approximations into the finite element method for the KdV equation, an accuracy comparable to the finite difference methods could be obtained for a similar computational expense.

The numerical solution of Nwogu's one-dimensional extended Boussinesq equations was described in Chapter 4. It was shown that the constant depth form of this system was structurally similar to coupled RLW and KdV equations and hence that the methods described and analysed in the previous chapter could be directly applied. In particular the third space derivatives could be rewritten in lower order differential form by introducing an auxiliary equation. Based on this a novel linear finite element method for Nwogu's one-dimensional system was designed. To the author's knowledge this is the first published finite element method for the extended Boussinesq equations. In this work an attempt has been made to derive boundary conditions in a mathematically and physically consistent manner. This led to a new formulation of the solid wall boundary condition for these equations and, by drawing an analogy with a time varying sea bed, a new formulation of internal wave generation. The numerical method was tested for theoretical and experimental problems and excellent results were obtained.

Chapter 5 described the numerical approximation of Nwogu's two-dimensional extended Boussinesq equations. It was shown that the equations could be rewritten in a lower order form, in a similar manner to that described for the one-dimensional system, by introducing two auxiliary equations. Hence, by extending the techniques described in the previous chapter, an unstructured linear triangular finite element spatial discretisation could be applied to the equations. To the author's knowledge this is the first attempt at such a method for the extended Boussinesq equations. The boundary conditions developed for the one-dimensional equation system were extended to the two-dimensional system, including the novel formulations of the solid wall condition and the method for internal wave generation. The spatial approximation was coupled with the adaptive time integration software, with a sparse preconditioned GMRES iterative method used to solve the linear equation system at each time step. The efficient implementation of the software in combination with the finite element method was discussed in detail. The results show that this method can produce accurate results for nonlinear, dispersive wave problems. In particular excellent results were obtained for the elliptical bump experiment, for which experimental results were available for comparison. The numerical experiments show

that when applying the method to geometrically complex problems it is important to use a suitably smoothly varying mesh. For the case of the harbour geometry considered at the end of the chapter it was found to be necessary to generate the unstructured mesh very carefully to ensure this. This experiment showed that while it was possible to simulate the propagation of waves into and around the harbour structure the interaction of the reflected waves with the non-physical seaward boundaries made it impossible to run the model for large enough times to extract useful results. The internal wave generation method developed in this work, which in principle would be suitable for this problem, was found to be highly mesh dependent and it proved impossible to generate useful results with this model.

6.2 Further work

The Boussinesq wave model developed in this work has limitations both in the mathematical formulation of the problem and in the computational algorithm. These are discussed in the following paragraphs and some possible extensions to the model developed here are outlined.

The mathematical model is limited to weakly nonlinear, weakly dispersive shallow water wave conditions as described in Chapter 2. By employing extended Boussinesq equations the restrictions on weak dispersion were improved but the equations remain only weakly nonlinear, and it was seen in the numerical experiments in Chapter 4 that the model will become inaccurate as the waves become higher. Fully nonlinear Boussinesq models have been developed [107] but the mathematical form of the equations is more complex. The equations considered here are also subject to a lower limit of the depth, since as the wave amplitude becomes large compared to the depth the waves will tend to break and the depth averaged formulation, which implies a relatively small vertical velocity, will become invalid. The inclusion of wave breaking in Boussinesq models has been reported in the literature [7, 54, 89], the simplest such method being an eddy viscosity approach with a dissipative term used to diffuse wave energy in the wave breaking region. Such an approach would be relatively simple to include in this model, although these models generally include free parameters that require calibration [89].

The boundary conditions implemented in this work were limited to those necessary for the experiments that were performed. Further boundary condition types that could be implemented include partial reflection, to represent solid boundaries of different materials, and a method to introduce multidirectional, multicomponent waves into the model. Previous formulations of the former condition were discussed in Section 4.5.2. The one-dimensional implementation of the latter was discussed in Section 4.5.1, involving the approximation of the wave with a sum of periodic wave components. These waves, and

any harmonic waves produced by nonlinear interaction, can then be introduced by superimposing the individual components at the inflow boundary [106]. A similar procedure for multidirectional wave in two dimensions has been described by Nwogu [73]. The method derived in this work for internal wave generation appears to be robust and accurate for the one-dimensional equation system, but the implementation of this method in two-dimensions was not completely successful. A more precise approach to this problem would be to rederive the equation system, following the procedure described in Chapter 2, with the assumption of a scaled, nondimensionalised time varying depth. However this would significantly complicate the derivation process. It is also possible that the problems encountered here are purely mesh dependent effects in which case a more detailed analysis of the interaction of the numerical scheme and the wave generation region is required.

At present one of the major computational overheads in the numerical algorithm is the calculation, factorisation and storage of the Jacobian matrix during the time integration process. The incomplete factorisation of this matrix is required to precondition the GMRES iterative method. Major gains in efficiency could be obtained if an alternative, less expensive, preconditioning technique could be employed, or if the current preconditioning matrix could be obtained in a more efficient manner. At present the DASPK software is used to compute the Jacobian matrix numerically. However, as described in Section 5.6.2.4, the implementation is relatively inefficient with a computational expense proportional to the matrix bandwidth, which can become large in the cases considered here. The SPRINT time integration software implements a different procedure, based on the matrix sparsity pattern, which is far more efficient in general and large increases in efficiency would be expected if a similar procedure were used here. The computational expense of the incomplete factorisation of this matrix could be reduced by using a simpler preconditioning strategy although the non-positive definite, non-symmetric form of the matrix system produced here requires an accurate method if the iterative method is to converge and be robust. It is possible that alternative reordering strategies will improve the performance of the iterative method but this is unlikely to offer significant gains over the Reverse Cuthill-McKee method already employed.

More work is required to investigate the mesh dependent effects observed in the two-dimensional simulations in Chapter 5. An analysis of the algorithm, similar to that done in the one-dimensional case, would provide information about the relationship between the truncation errors and the mesh spacing although this will be significantly more complicated in the two-dimensional case. The one-dimensional analysis in Section 4.8 suggested a limit on the local variation of the mesh size but this is difficult to apply to two-dimensional unstructured triangular grids for geometrically complicated domains. The mesh generation procedure is complicated by the requirements of a smoothly varying

grid and reliable procedures are required to produce meshes for practical geometries.

Mesh adaptivity has not been considered in this work, primarily because the emphasis here has been on designing the basic numerical algorithm for the equations. The mesh resolution required is generally predetermined by the expected wavelength and at present the mesh is designed to resolve that wavelength across the whole spatial domain. In the initial stages of the computation the wave is gradually introduced at the inflow boundary and computational savings could be made if the domain far from this region was not fully resolved at this point. However the combination of the adaptivity process with the restrictions on the mesh uniformity required by this model may complicate this process. Reliable error estimation procedures would be required to drive the adaptivity process and while results have been published and applied for the original form of the one-dimensional Boussinesq equations [44, 82] there is at present no published work on the extended Boussinesq equations in one or two dimensions. A further complication is provided by the time integration software which requires a consistent set of solution values and time derivatives on the adapted mesh if uniform accuracy of the time integration process is to be maintained [17]. This places restrictions on the interpolation of data to the adapted grid.

The code developed as part of this project has been fully documented and a user manual has been written [103] including detailed descriptions of the implementation and application of this method. This has been supplied to HR Wallingford for further investigation and development.

Bibliography

[1] M.B. Abbot, A.D. McCowan, and I.R. Warren. Accuracy of short-wave numerical models. *ASCE Journal of Hydraulic Engineering*, 110(10):1287–1301, 1984.

[2] M.B. Abbot, H.M. Petersen, and O. Skovgaard. On the numerical modelling of short waves in shallow water. *Journal of Hydraulic Research*, 16(3):173–203, 1978.

[3] M.B. Abbot and G.S. Rodenhuis. A numerical simulation of the undular hydraulic jump. *Journal of Hydraulic Research*, 10(3):239–257, 1972.

[4] M.E. Alexander and J.L. Morris. Galerkin methods applied to some model equations for non-linear dispersive waves. *Journal of Computational Physics*, 30:428–451, 1979.

[5] D. Ambrosi. A new finite element scheme for the Boussinesq equations. *Mathematical Models and Methods in Applied Sciences*, 7(2):193–209, 1997.

[6] D. Ambrosi and L. Quartapelle. A Taylor-Galerkin method for simulating nonlinear dispersive water waves. *Journal of Computational Physics*, 146:546–569, 1998.

[7] J.S. Antunes do Carmo and F.J. Seabra Santos. On breaking waves and wave-current interaction in shallow water: A 2D*H* finite element model. *International Journal for Numerical Methods in Fluids*, 22:429–444, 1996.

[8] J.S. Antunes do Carmo, F.J. Seabra Santos, and E. Barthélemy. Surface waves propagation in shallow water: A finite element model. *International Journal for Numerical Methods in Fluids*, 16:447–459, 1993.

[9] S. Beji. Note on conservation equations for nonlinear surface waves. *Ocean Engineering*, 25(7):607–613, 1998.

[10] S. Beji and J.A. Battjes. Numerical simulation of nonlinear-wave propagation over a bar. *Coastal Engineering*, 23(1-2):1–16, 1994.

[11] S. Beji and K. Nadaoka. A formal derivation and numerical modelling of the improved Boussinesq equations for varying depth. *Ocean Engineering*, 23(8):691–704, 1996.

[12] S. Beji and K. Nadaoka. Authors' reply to 'Discussion of Schäffer and Madsen on "A formal derivation and numerical modelling of the improved Boussinesq equations for varying depth" (Ocean Engineering, 25 (1998) 497-500)'. *Ocean Engineering*, 25(7):615–618, 1998.

[13] T.B. Benjamin, J.L. Bona, and J.J. Mahony. Model equations for long waves in nonlinear dispersive systems. *Philosophical Transactions of the Royal Society of London. Series A*, 272:47–78, 1972.

[14] M. Benzi, D.B. Szyld, and A. Van Duin. Orderings for incomplete factorization preconditioning of nonsymmetric problems. *SIAM Journal on Scientific Computing*, 20(5):1652–1670, 1999.

[15] P.J. Beresford. Pittenweem Harbour, Fife, Scotland. Random wave disturbance studies of a proposed fishing harbour development. Technical Report EX 1505, HR Wallingford, 1996.

[16] M. Berzins. Temporal error control for convection-dominated equations in two space dimensions. *SIAM Journal on Scientific and Statistical Computing*, 16(3):558–580, 1995.

[17] M. Berzins, P.J. Capon, and P.K. Jimack. On spatial adaptivity and interpolation when using the method of lines. Technical Report 96.31, School of Computer Studies, University of Leeds, 1996.

[18] M. Berzins, P.M. Dew, and R.M. Furzeland. Developing software for time-dependent problems using the method of lines and differential-algebraic integrators. *Applied Numerical Mathematics*, 5:375–397, 1989.

[19] M. Berzins and R.M. Furzeland. A user's manual for SPRINT - a versatile software package for solving systems of algebraic, ordinary and partial differential equations: Part 1 - algebraic and ordinary differential equations. Technical Report TNER.85.058, Thornton Research Centre, Chester, 1985.

[20] M. Berzins, R.M. Furzeland, and L.E. Scales. A user's manual for SPRINT - a versatile software package for solving systems of algebraic, ordinary and partial differential equations: Part 3 - advanced use of SPRINT. Technical Report TNER.85.058, Thornton Research Centre, Chester, 1985.

[21] E.C. Bowers and P.J. Beresford. Set-down beneath wave groups: Laboratory work using long crested random waves. Technical Report IT 243, HR Wallingford, 1983.

[22] E.C. Bowers and P.J. Beresford. Diffraction of set-down waves in harbours: Laboratory work using regular wave groups. Technical Report IT 283, HR Wallingford, 1984.

[23] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.

[24] M. Brorsen and J. Larsen. Source generation of nonlinear gravity waves with the boundary integral equation method. *Coastal Engineering*, 11:93–113, 1987.

[25] P.N. Brown, A.C. Hindmarsh, and L.R. Petzold. Using Krylov methods in the solution of large-scale differential-algebraic systems. *SIAM Journal on Scientific Computing*, 15(6):1467–1488, 1989.

[26] P.N. Brown, A.C. Hindmarsh, and L.R. Petzold. Consistent initial condition calculation for differential- algebraic systems. *SIAM Journal on Scientific Computing*, 19(5):1495–1512, 1998.

[27] Y. Chen and P.L.-F. Liu. Modified Boussinesq equations and associated parabolic models for water wave propagation. *Journal of Fluid Mechanics*, 288:351–381, 1995.

[28] I. Christie, D.F. Griffiths, A.R. Mitchell, and J.M. Sanz-Serna. Product approximation for non-linear problems in the finite element method. *IMA Journal of Numerical Analysis*, 1:253–266, 1981.

[29] M. Dæhlen and A. Tveito, editors. *Numerical Methods and Software Tools in Industrial Mathematics*. Birkhäuser Boston, 1997.

[30] E.F. D'Azevedo, P.A. Forsyth, and W.-P. Tang. Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems. *SIAM Journal on Matrix Analysis and Applications*, 13(3):944–961, 1992.

[31] E.F. D'Azevedo, P.A. Forsyth, and W.-P. Tang. Towards a cost-effective high order ILU preconditioner with higher level fill. *BIT*, 32(3):442–463, 1992.

[32] M.W. Dingemans. Comparison of computations with Boussinesq-like models and laboratory measurements. Technical Report H1684.12, MAST G8M Coastal Morphodynamics Research Programme, 1994.

[33] N. Dodd. A numerical model of wave run-up, overtopping and regeneration. *ASCE Journal of Waterway, Port, Coastal, and Ocean Engineering*, 124(2):73–81, 1998.

[34] J. Donea and S. Giuliani. A simple method to generate high-order accurate convection operators for explicit schemes based on linear finite elements. *International Journal for Numerical Methods in Fluids*, 1:63–79, 1981.

[35] P.G. Drazin and R.S. Johnson. *Solitons: an Introduction*. Cambridge University Press, 1989.

[36] J.C. Eilbeck and G.R. McGuire. Numerical studies of the Regularised Long-Wave equation. I: Numerical methods. *Journal of Computational Physics*, 19:43–57, 1975.

[37] J.C. Eilbeck and G.R. McGuire. Numerical studies of the Regularised Long-Wave equation. II: Interaction of solitary waves. *Journal of Computational Physics*, 23:63–73, 1977.

[38] M.S. Engelman, R.L. Sani, and P.M. Gresho. The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible fluid flow. *International Journal for Numerical Methods in Fluids*, 2:225–238, 1982.

[39] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Mathematics of Computation*, 31(139):629–651, 1977.

[40] L.R.T. Gardner and G.A. Gardner. Solitary waves of the Regularised Long-Wave equation. *Journal of Computational Physics*, 91:441–459, 1990.

[41] L.R.T. Gardner, G.A. Gardner, and A.H.A. Ali. Simulations of solitons using quadratic spline finite elements. *Computer Methods in Applied Mechanics and Engineering*, 92:231–243, 1991.

[42] L.R.T. Gardner, G.A. Gardner, and I. Dag. Hermite infinite elements and graded quadratic B-spline finite elements. *International Journal for Numerical Methods in Engineering*, 36:3317–3332, 1993.

[43] A. George and J.W.-H. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, 1981.

[44] M. Grasselli, S. Perotto, and F. Saleri. Space-time finite elements for Boussinesq equations. Technical Report 294/P, Department of Mathematics. Politecnico di Milano, 1997.

[45] I.S. Greig and J.L. Morris. A hopscotch method for the Korteweg-de-Vries equation. *Journal of Computational Physics*, 20:64–80, 1976.

[46] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer-Verlag, 1991.

[47] A. Hauguel. A numerical model of storm waves in shallow water. In *17th International Conference on Coastal Engineering, Sydney, Australia*, pages 746–762, 1980.

[48] M. Israeli and S.A. Orszag. Approximation of radiation boundary conditions. *Journal of Computational Physics*, 42:115–135, 1981.

[49] K.R. Jackson and R. Sacks-Davis. An alternative implementation of variable step-size multistep formulas for stiff ODEs. *ACM Transactions on Mathematical Software*, 6(3):295–318, 1980.

[50] K.R. Jackson and W.L. Seward. Adaptive linear equation solvers in codes for large stiff systems of ODEs. *SIAM Journal on Scientific Computing*, 14(4):800–823, 1993.

[51] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, 1987.

[52] J.M. Kaihatu and J.T. Kirby. Two-dimensional parabolic modeling of extended Boussinesq equations. *ASCE Journal of Waterway, Port, Coastal, and Ocean Engineering*, 124(2):57–67, 1998.

[53] Th. Karambas, Y. Krestenitis, and C. Koutitas. A numerical solution of Boussinesq equations in the inshore zone. *Hydrosoft*, 3(1):34–37, 1990.

[54] G. Karypis and V. Kumar. Metis. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Technical Report Version 3.0, Dept. of Computer Science, University of Minnesota, 1997.

[55] N.D. Katopodes and C. Wu. A model for unidirectional water waves. *ASCE Journal of Engineering Mechanics*, 112(7):697–717, 1986.

[56] N.D. Katopodes and C. Wu. Computation of finite-amplitude dispersive waves. *ASCE Journal of Waterway, Port, Coastal, and Ocean Engineering*, 113(4):327–346, 1987.

[57] M. Kawahara and J.Y. Cheng. Finite element method for Boussinesq wave analysis. *International Journal of Computational Fluid Dynamics*, 2:1–17, 1994.

[58] H.P. Langtangen and G. Pedersen. Finite elements for the Boussinesq wave equations. In J. Grue, B. Gjeviki, and J.E. Weber, editors, *Waves and Nonlinear Processes in Hydrodynamics*, pages 1–10. Kluwer Academic, 1996.

[59] H.P. Langtangen and G. Pedersen. Computational methods for weakly dispersive and nonlinear water waves. *Computer Methods in Applied Mechanics and Engineering*, 160:337–371, 1998.

[60] J. Larsen and H. Dancy. Open boundaries in short wave simulations - a new approach. *Coastal Engineering*, 7:285–297, 1983.

[61] J. Lawson, M. Berzins, and P.M. Dew. Balancing space and time errors in the method of lines for parabolic equations. *SIAM Journal on Scientific and Statistical Computing*, 12(3):573–594, 1991.

[62] Y.S. Li, S.-X. Liu, Y.-X. Yu, and G.-Z. Lai. Numerical modeling of Boussinesq equations by a finite element method. *Coastal Engineering*, 37:97–122, 1999.

[63] P.L.-F. Liu, S.B. Yoon, and J.T. Kirby. Nonlinear refraction-diffraction of waves in shallow water. *Journal of Fluid Mechanics*, 153:185–201, 1985.

[64] R. Löhner. Some useful renumbering strategies for unstructured grids. *International Journal for Numerical Methods in Engineering*, 36:3259–3270, 1993.

[65] O.S. Madsen and C.C. Mei. The transformation of a solitary wave over an uneven bottom. *Journal of Fluid Mechanics*, 39(4):781–791, 1969.

[66] P.A. Madsen, R. Murray, and O.R. Sørensen. A new form of the Boussinesq equations with improved linear dispersion characteristics. *Coastal Engineering*, 15:371–388, 1991.

[67] P.A. Madsen and O.R. Sørensen. A new form of the Boussinesq equations with improved linear dispersion characteristics. Part 2. A slowly varying bathymetry. *Coastal Engineering*, 18:183–204, 1992.

[68] A.D. McCowan. Equation systems for modelling dispersive flow in shallow water. In *21st IAHR Congress*, pages 50–57, 1985.

[69] A.D. McCowan. The range of application of Boussinesq type numerical short wave models. In *23rd IAHR Congress*, pages 378–384, 1987.

[70] C.C. Mei and B. Le Méhauté. Note on the equations of long waves over an uneven bottom. *Journal of Geophysical Research*, 71(2):393–400, 1966.

[71] K. Nadaoka, S. Beji, and Y. Nakagawa. A fully dispersive weakly nonlinear model for water waves. *Proceedings of the Royal Society of London. Series A*, 453(1957):303–318, 1997.

[72] O. Nwogu. Alternative form of Boussinesq equations for nearshore wave propagation. *ASCE Journal of Waterway, Port, Coastal, and Ocean Engineering*, 119(6):618–638, 1993.

[73] O. Nwogu. Nonlinear transformation of multi-directional waves in water of variable depth. Draft manuscript, 1993.

[74] T. Ohyama, W. Kioka, and A. Tada. Applicability of numerical models to nonlinear dispersive waves. *Coastal Engineering*, 24:297–313, 1995.

[75] T. Ohyama and K. Nadaoka. Development of a numerical wave tank for analysis of nonlinear and irregular wave field. *Fluid Dynamics Research*, 8(5-6):231–251, 1991.

[76] S.V. Pennington and M. Berzins. New NAG Library software for first-order partial differential equations. *ACM Transactions on Mathematical Software*, 20(1):63–99, 1994.

[77] J. Peraire, J. Peiro, and K. Morgan. A 3D finite element multigrid solver for the Euler equations. Technical Report 92-0449, AIAA, 1992.

[78] D.H. Peregrine. Calculations of the development of an undular bore. *Journal of Fluid Mechanics*, 25(2):321–330, 1966.

[79] D.H. Peregrine. Long waves on a beach. *Journal of Fluid Mechanics*, 27(4):815–827, 1967.

[80] D.H. Peregrine. Equations for water waves and the approximations behind them. In R. Meyer, editor, *Waves on Beaches and Resulting Sediment Transport*, pages 95–122. Academic Press, 1972.

[81] D.H. Peregrine. Discussion of: A numerical simulation of the undular hydraulic jump, by M.B. Abbot and G.S. Rodenhuis. *Journal of Hydraulic Research*, 12(1):141–157, 1974.

[82] S. Perotto. A posteriori error estimates for Boussinesq equations. In *Proceedings of the ICFD Conference on Numerical Methods for Fluid Dynamics VI*, pages 451–457, 1998.

[83] J. Petera and J.F.T. Pittman. Isoparametric Hermite elements. *International Journal for Numerical Methods in Engineering*, 37:3489–3519, 1994.

[84] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS, 1996.

[85] J.M. Sanz-Serna and I. Christie. Petrov-Galerkin methods for nonlinear dispersive waves. *Journal of Computational Physics*, 39:94–102, 1981.

[86] H.A. Schäffer and P.A. Madsen. Further enhancements of Boussinesq-type equations. *Coastal Engineering*, 26:1–14, 1995.

[87] H.A. Schäffer and P.A. Madsen. Discussion of "A formal derivation and numerical modelling of the improved Boussinesq equations for varying depth". *Ocean Engineering*, 25(6):497–500, 1998.

[88] H.A. Schäffer and P.A. Madsen. Further discussion related to 'A formal derivation and numerical modelling of the improved Boussinesq equations for varying depth' by Beji and Nadaoka. *Ocean Engineering*, 26(11):1057–1062, 1999.

[89] H.A. Schäffer, P.A. Madsen, and R. Deigaard. A Boussinesq model for waves breaking in shallow water. *Coastal Engineering*, 20:185–202, 1993.

[90] H. Schaper and W. Zielke. A numerical solution of Boussinesq type wave equations. In *19th International Coastal Engineering Conference*, pages 1057–1069, 1984.

[91] S.W. Schoombie. Finite element methods for the Korteweg-de Vries equation I. Galerkin methods with Hermite cubics. Technical Report NA/43, Department of Mathematics. University of Dundee, 1980.

[92] S.W. Schoombie. Finite element methods for the Korteweg-de Vries equation II. Petrov-Galerkin methods with splines. Technical Report NA/44, Department of Mathematics. University of Dundee, 1980.

[93] S.W. Schoombie. Spline Petrov-Galerkin methods for the numerical solution of the Korteweg-de Vries equation. *IMA Journal of Numerical Analysis*, 2:95–109, 1982.

[94] A. Schröter, R. Mayerle, A. Kahlfeld, and W. Zielke. Assessment of a Boussinesq wave model for the design of a harbour. In *International Conference on Coastal and Port Engineering in Developing Countries*, pages 741–753, 1995.

[95] A. Schröter, R. Mayerle, and W. Zielke. Optimized dispersion characteristics of the Boussinesq wave equations. Technical report, Institute of Fluid Mechanics and Computer Applications in Civil Engineering. University of Hannover, 1994.

[96] J.R. Shewchuck. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *First Workshop on Applied Computational Geometry*, pages 124–133, 1996.

[97] C. Skotner and C.J. Apelt. Internal wave generation in an improved two-dimensional Boussinesq model. *Ocean Engineering*, 26(4):287–324, 1999.

[98] J.V. Smallman and A.J. Cooper. A mathematical model for set down in harbours. *Coastal Engineering*, 13:247–261, 1989.

[99] H.J. Stetter. Tolerance proportionality in ODE codes. In R. März, editor, *Proceedings of the Second Conference on Numerical Treatment of Ordinary Differential Equations*, pages 109–123. Humboldt University, Berlin, 1980.

[100] G. Strang and G.J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, 1973.

[101] F. Ursell. The long-wave paradox in the theory of gravity waves. *Proceedings of the Cambridge Philosophical Society*, 49:685–694, 1953.

[102] A.C. Vliegenthart. On finite-difference methods for the Korteweg-de Vries equation. *Journal of Engineering Mathematics*, 5(2):137–155, 1971.

[103] M. Walkley. User manual for Bouss2D. A finite element method for the extended Boussinesq equations. Technical report, HR Wallingford, 1999.

[104] M. Walkley and M. Berzins. A finite element method for the one-dimensional extended Boussinesq equations. *International Journal for Numerical Methods in Fluids*, 29(2):143–157, 1999.

[105] J. Ware and M. Berzins. Towards SPRINT+. Part 1: Reduced storage and parallel linear algebra routines. 1990.

[106] G. Wei and J.T. Kirby. Time-dependent numerical code for extended Boussinesq equations. *ASCE Journal of Waterway, Port, Coastal, and Ocean Engineering*, 121(5):251–261, 1995.

[107] G. Wei, J.T. Kirby, S.T. Grilli, and R. Subramanya. A fully nonlinear Boussinesq model for surface waves. Part 1. Highly nonlinear unsteady waves. *Journal of Fluid Mechanics*, 294:71–92, 1995.

[108] G. Wei, J.T. Kirby, and A. Sinha. Generation of waves in Boussinesq models using a source function approach. *Coastal Engineering*, 36:271–299, 1999.

[109] G.B. Whitham. *Linear and Nonlinear Waves*. Wiley-Interscience, 1974.

[110] R.L. Wiegel. *Oceanographical Engineering*. Prentice-Hall, 1964.

[111] J.M. Witting. A unified model for the evolution of nonlinear water waves. *Journal of Computational Physics*, 56:203–236, 1984.

[112] J. Zang, J. Lawson, N. Tozer, and N. Dodd. The development of a computational wave model based on extended Boussinesq equations. Technical report, HR Wallingford, 1996.

[113] O.C. Zienkiewicz and K. Morgan. *Finite Elements and Approximation*. Wiley-Interscience, 1983.