The Dissertation Committee for Nishant Panda
certifies that this is the approved version of the following dissertation:

Discontinuous Galerkin methods for resolving non linear and dispersive near shore waves

Committee:

Clint Dawson, Supervisor

L.L. Raja

V. Raman

P. Varghese

Joannes J. Westerink

Discontinuous Galerkin methods for resolving non linear and dispersive near shore waves

by

Nishant Panda, B.Tech.; M.S.E

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2014

To my parents.

Acknowledgments

Everything I have achieved so far has been due to the many people who have offered me their unconditional help. I cannot imagine this dissertation without their contribution.

First and foremost, I would like to thank my advisor Prof. Clint Dawson. Not only did he provide me with the opportunity to work on an exciting and challenging project, but his unwavering support and motivation is what made my dissertation possible. His is the ideal advisor that I could have hoped for.

I would like to thank my parents who allowed me to pursue my dreams and supported every decision I made. My wife, Cynthia Gonzalez, who made my time in Austin extremely enjoyble also holds my deepest gratitude. This would not have been possible without her staying by my side during all the hardships I faced.

I would also like to thank my committe members for their time and insightful input on my research. I enjoyed all the academic courses I took under them which prepared me to tackle my research topic.

Finally I would like to thank all my colleagues and friends that have helped my throughout my graduate studies.

Discontinuous Galerkin methods for resolving non linear and dispersive near shore waves

Nishant Panda, Ph.D.
The University of Texas at Austin, 2014

Supervisor: Clint Dawson

Near shore hydrodynamics has been an important research area dealing with coastal processes. The nearshore coastal region is the region between the shoreline and a fictive offshore limit which usually is defined as the limit where the depth becomes so large that it no longer influences the waves. This spatially limited but highly energetic zone is where water waves shoal, break and transmit energy to the shoreline and are governed by highly dispersive and non-linear effects. An accurate understanding of this phenomena is extremely useful, especially in emergency situations during hurricanes and storms. While the shallow water assumption is valid in regions where the characteristic wavelength exceeds a typical depth by orders of magnitude, Boussinesq-type equations have been used to model near-shore wave motion. Unfortunately these equations are complex system of coupled non-linear and dispersive differential equations that have made the developement of numerical approximations extremely challenging.

In this dissertation, a local discontinuous Galerkin method for Boussinesq-Green Naghdi Equations is presented and validated against experimental results. Currently Green-Naghdi equations have many variants. We develop a numerical method in one horizontal dimension for the Green-Naghdi equations based on rotational characteristics in the velocity field. Stability criterion is also established for the linearized Green-Naghdi equations and a careful proof of linear stability of the numerical method is carried out. Verification is done against a linearized standing wave problem in flat bathymetry and h,p (denoted by K in this thesis) error rates are plotted. The numerical method is validated with experimental data from dispersive and non-linear test cases.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This dissertation is about the developement of numerical techniques for solving extremely non-linear and dispersive near-shore water waves modeled by the Boussinesq-Green-Naghdi equations (Zhang, Kennedy, Panda, Dawson, & Westerink, 2013). Near-shore wave models have gone through a long history of developement and currently there are various models with varying degrees of complexities. The extreme non-linear characteristics of these models along with higher order spatial derivatives has made it cumbersome for the development of highly accurate numerical methods on arbitrary grids. This dissertation work is focused on developing a robust and accurate numerical scheme for such equations.

In modelling the near shore, a particularly spatially limited but highly energetic region is the *surf zone* where waves shoal, break and dissipate energy through to the shoreline. Here, nonlinear surface wave profiles deviate strongly from the linear superposition of sinusoids assumed in deeper waters, with superharmonic phase-locking leading to sharper, higher, crests and flatter troughs, while subharmonic interactions generate low frequency motions that can dominate dynamics in the inner surf and swash (runup) zones (Kennedy, Chen, Kirby, & Dalrymple, 2000)(Mase & Kirby, 1992). Wave setup (which can increase water levels by up to $0.6m$) and wave-driven currents (which may be greater than $2m/s$ in severe storms) are both generated by the transfer of momentum from surf zone waves into larger scale motions (Q. Chen, Kirby, Dalrymple, Shi, & Thornton, 2003)(Ting & Kirby, 1995). Sediment transport and erosion in the surf zone depend strongly on near-bottom wave orbital velocities which, like the nonlinear surface profiles, also deviate strongly from simple sinusoids (Ting & Kirby, 1994)(Ting & Kirby, 1995).

The surf zone becomes especially important in severe storms such as hurricanes where very large wind waves can combine with very fast currents, and water levels may be much higher than normal. The consequences of the wind wave-current interaction during hurricanes can affect inland wind wave propagation, can influence flooding far inland, and can change the sediment dynamics and therefore the shape of the coast. Unfortunately, the ability to model accurately and in detail this highly energetic and important zone has been limited due to requirements for very high levels of mesh resolution, complex governing equations and prohibitive computational costs (Lin, Chang, & Liu, 1999).

The desire to strive a balance between accuracy and complexity of the wave physics has led to the development of many near-shore models. Although the numerical theory for hyperbolic wave equations is well established, numerical approximations for dispersive wave equations have been very challenging to obtain, especially in arbitrary grids. Recently, discontinuous Galerkin (DG) methods have been gaining a lot of popularity in diverse applications. The discontinuous Galerkin methods are locally conservative, stable and high-order methods which can handle complex geometries. This feature has made the method attractive in applications to water wave theories. The ojective of this work is to develop a numerical scheme based on the discontinuous Galerkin framework that is stable, accurate and robust.

Near-shore water waves exhibit complex physics and a fairly accurate understanding of such phenomena is extremely useful. From an engineering perspective, it is important to be able to estimate design loads during the design process of maritime constructions of oil rigs, offshore windmill farms, etc. The ability to predict water levels, current and wave environments near and behind features such as barrier islands, dunes, nearshore breaking zones, inland roads and levees is important in emergency situations like hurricanes and storms. The broader impact of this dissertation work will include the ability to evaluate flood risk behind a barrier or levee, assess the actual degradation of dunes, barrier islands, levees, roads and railroads, compute wave runup behind wave breaking zones which can be very significant on structures such as levees or on deep ocean islands with steep coastal topography, determine nonlinear wave climate around coastal structures such as bridges and buildings and forecast storm surge and waves, plan evacuations, assess coastal risk, design levees and closures, and operate shipping by federal and state agencies including FEMA, NOAA, the USACE, and the U.S. Navy.

1.1 Boussinesq equations

Surface water wave theory has been an evolving research topic where asymptotic models have been used to resolve wave characteristics. Water waves propagating from deep water regions experience significant transformations resulting in a rapid change in height, speed and direction. As depth decreases, waves become skewed about their crest with marked steepening of the forward face until instability sets in resulting in wave breaking. Wave shoaling is described as the transformation of waves from near shore zone until wave breaking.

While shallow water assumptions are valid where the characterstic wavelength (L) exceeds a typical depth (h_0) by orders of magnitude i.e $kh_0 \ll 1$, non-linear near-shore waves (where amplitude a and h_0 are comparable) have mostly been modeled through perturbation techniques based on two non-dimensional parameters $\mu = kh_0$ and $\epsilon = a/h_0$ first formulated by Boussinesq in 1872 and Rayleigh in 1876. The smallness of μ is used to construct a polynomial representation of the velocity field in the vertical co-ordinate which reduces a $3D$ flow model to a $2D$ flow model. Moreover, the non-linear free surface conditions are absorbed in the resulting equations which makes it more tractable.

However, the scalings that are used in the perturbation analysis of Boussinesq models (Madsen & Sørensen, 1992)(Nwogu, 1993)(Peregrine, 1967) can be severly restrictive. Wave shoaling is known to occur when $\mu \approx 1$, while breaking is known to occur when $\epsilon \approx 1$. Hence wave models that have restrictions on μ and ϵ will be inaccurate(Kennedy, Kirby, & Gobbi, 2002) in capturing many shoaling and breaking phenomena. Most of the Bossinesq models also assume an irrotational flow field and are hence valid up to the breaking point. Since, vortices are generated from wave breaking, any model based on irrotational flow will induce large errors in the velocity field.

An alternate approach to the computation of shallow water nonlinear dispersive waves lies in the Green-Naghdi(Green & Naghdi, 1976)(Serre, 1953)(Shields & Webster, 1988) formulation, where a polynomial structure for the velocity field is retained without any irrotational assumptions. Almost all GreenNaghdi based formulations have been developed in the shallow water limit, although researchers(Webster & Kim, 1991) have successfully extended the formulation to deeper waters. Recently, in(Zhang et al., 2013), the authors developed the Green-Naghdi formulation to arbitrary levels of approximation but also retained the Boussinesq scaling. Such a formulation can be naturally extended to model surf-zones.

Henceforth, in this thesis, we will refer to these equations as the **R-GN** equations. There are also water wave theories based on the Green-Naghdi approach that employ irrotational characteristics into the velocity formulation. Such systems have been known to provide accurate linear and non-linear dispersion(Lannes & Bonneton, 2009)(Bonneton, Chazel, Lannes, Marche, & Tissier, 2011), and their irrotational assumption brings it more in line with standard Boussinesq systems. We'll refer to these as **I-GN** equations.

In this thesis, a form of Green-Naghdi equation based on Boussinesq scaling introduced in(Zhang et al., 2013) will be examined and a numerical approximation based on the discontinuous Galerkin method will be investigated. We will also comment on the numerical approximation for the Green-Naghdi equations based on the classical irrotational flow assumption as introduced in(Bonneton et al., 2011). These equations are described in the chapter *Governing equations.*

1.2 Discontinuous Galerkin method

The original discontinuous Galerkin method was introduced in (Reed & Hill, 1973) to solve the neutron transport equation where the angular flux was approximated by piecewise polynomials that were discontinuous across the element boundaries. Because of the linear nature of the equation, the approximate solution was computed element by element when the elements are suitably ordered according to the characteristic direction. The convergence analysis of this method was carried out in(Lesaint & Raviart, 1974) and the order of convergence was shown to be proportional to δx^k where k was the polynomial order of the approximate solution. Later, in(Johnson & Pitkäranta, 1986) a rate of convergence of $\delta x^{k+1/2}$ was proved for general triangulations. The success of this method for linear equations led to the extension of the method to nonlinear hyperbolic conservation laws. A 1D implementation using the discontinuous Galerkin framework of a non-linear hyperbolic differential equation was first carried out in(Chavent & Salzano, 1982). To improve the stability of the scheme, a *slope-limiter* was introduced in(Chavent & Cockburn, 1989). This slope limitter was motivated by the ideas introduced in (Van Leer, 1974). However, the scheme was only first order accurate in time and the use of *slope limiter* to balance the spurious oscillations in smooth regions caused by linear instability adversely affected the quality of the approximation in these regions. This problem was solved by the introduction of the Runge-Kutta discontinuous Galerkin ($RKDG$) scheme in(Cockburn & Shu, 1991b). In(Cockburn & Shu, 1989), this approach was extended to construct (formally) high-order accurate RKDG methods for the scalar conservation law. To derive RKDG methods of order $k+1$, the authors used the DG method with polynomials of degree k for the space discretization, a TVD $(k+1)^{th}$ order accurate explicit time discretization, and a generalized slope limiter. The extension of the RKDG methods to general multidimensional systems was started in(Cockburn & Shu, 1991a) and was completed in(Cockburn & Shu, 1998b). The first extensions of the RKDG method to nonlinear, convection-diffusion systems were proposed in(Z. Chen, Cockburn, Jerome, & Shu, 1995) in the context of semi-conductor devices where approximations of second and third-order derivatives of the discontinuous approximate solution were obtained by using simple projections into suitable finite elements spaces and a *mass lumping* techinque was used to avoid inverting the mass matrices. For higher order polynomial discretization this leads to a substantial degradation of the formal order of accuracy. This issue was resolved in(Bassi & Rebay, 1997) where both the variable and its gradient were treated independently. This idea was generalized in(Cockburn & Shu, 1998a) which led to the developement of the *local discontinuous Galerkin* method. The basic idea to construct the LDG methods is to suitably rewrite the original system as a larger, first-order system and then discretize it by the RKDG method. By a careful choice of this rewriting, nonlinear stability can be achieved even without slope limiters. Another technique to discretize the diffusion terms was proposed by Baumann(Baumann, 1997). The one-dimensional case was studied in(Babuška, Baumann, & Oden, 1999) and the case of convection-diffusion in multidimensions in(Baumann & Oden, 1999). The local discontinuous Galerkin method for convection-diffusion in multidimensions was further analysed in (Cockburn & Dawson, 2000). Discontinuous Galerkin Methods (DG) are locally conservative, stable and high-order methods which can easily handle complex geometries. This feature has made the method attractive in applications to water wave theories(Aizinger & Dawson, 2002) (Dawson et al., 2011)(Yan & Shu, 2002)(Eskilsson & Sherwin, 2006)(Engsig-Karup, Hesthaven, Bingham, & Madsen, 2006). In this section we briefly introduced the discontinuous Galerkin method and the local discontinuous Galerkin method to handle diffusion terms. We will use a similar strategy in devising a numerical scheme for the Boussinesq-Green-Naghdi equations. In the next chapter, the governing equations will be explained in detail.

1.3 Summary of contribution

In this thesis we have investegated a numerical method for solving the Boussinesq-Green-Naghdi equations using the discontinuous Galerkin framework. In particular we have achieved the following:

- We have developed and implemented a local discontinuous Galerkin numerical method to solve the R-GN equations in $1D$. Although the implementation is in $1D$, it can be easily extended to $2D$. At present, there are no higher order numerical methods for Green-Naghdi type equations. Most implementations have been restricted to finite difference schemes. This is largely due to the complexity of Green-Naghdi equations that are extremely non-linear containing higher order spatial derivatives and include mixed space-time derivatives.

- We have verified our method for the linear case where an exact solution is known to exist and observed optimal/sub-optimal convergence rates. Validation of the scheme is done against challenging test cases and results show good agreement with the observational data.

- We have proved the linear stability of the numerical method and derived important constraints that the numerical scheme must satisfy to maintain linear stability. The complete non-linear stability is extremely difficult especially when the equations themselves are not proven to be long-time stable.

1.4 Outline

The remainder of the dissertation is laid out as follows. In Chapter 2 we describe the governing equations in complete detail and extend it to model surf zones in Chapter 3. In Chapter 4 we present the numerical method and give implementation details as well as the proof of its linear stability. Comments on achieving non-linear stability are also outlined. In Chapter 5 we perform the verification and validation of the numerical method and in Chapter 6 we provide concluding remarks together with future work. The Appendix lists various details of the model.

Chapter 2

Governing Equations

2.1 Linearized Water Wave problem

The flow regime under a water wave train can be decomposed into two regions - the bottom boundary layer and the flow outside the boundary layer. Typically for coastal waves whose time periods T_p is around $2 - 30s$, the boundary layer is $\approx 10mm$. For bathymetry that typically ranges from a few meters to a few tens of meters this boundary layer can be neglected and the flow can be treated as irrotational throughout. This assumption leads to the classical small-amplitude linear water wave. A typical figure is shown Figure (2.1).

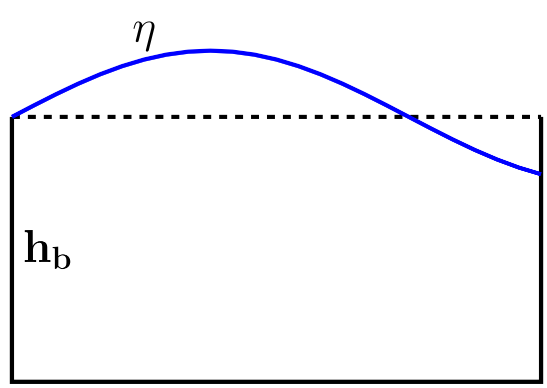

Figure 2.1: Initial set up of the water wave problem

With these assumptions, a fluid potential ϕ exists where

$$\mathbf{v} = -\nabla\phi$$

where $\mathbf{v}$ is the velocity, and so the continuity equation reduces to a laplace equation

$$\nabla \cdot \nabla\phi = 0,$$

that must exist through out the fluid. In order to uniquely solve the above equation we need suitable boundary conditions which are summarized below :

Kinematic boundary conditions : The mathematical expression for the kinematic boundary condition is usually derived from the equation which describes the surface that constitutes the boundary. For a surface given by $F(x, y, z, t) = c$,

$$\frac{D}{Dt}F = 0 = \frac{\partial F}{\partial t} + \mathbf{v}.\nabla F,$$

Let, $\hat{\mathbf{n}}$ be the unit normal to the surface , then

$$\mathbf{v} \cdot \nabla F = \mathbf{v} \cdot \hat{\mathbf{n}} \mid \nabla F \mid,$$

And hence,

$$\mathbf{v} \cdot \hat{\mathbf{n}} = \frac{\frac{-\partial F}{\partial t}}{\mid \nabla F \mid} \quad F(x, y, z, t) = c.$$

For the model problem in $x - z$ co-ordinates, we have two *kinematic* boundary conditions; one at the bottom and other at the free surface. At the **bottom**, the surface is given by $z = -h_b$. However , note that h_b is generally a function of the horizontal dimension (x, y). In this case, $h_b = h_b(x)$. Thus , we have

$F = h_b(x) + z = 0$. Working out the gradient and the normal we get,

$$w = -u\frac{dh_b}{dx} \text{ on } z = -h_b(x).$$

where u, w are the horizontal and vertical components of the velocity $\mathbf{v}$. On the free surface, we note that the surface equation is given by $z = \eta$. However $\eta = \eta(x, y, t)$. Hence, $F = z - \eta = 0$. Working out the gradient and the normal we get,

$$w = \frac{\partial \eta}{\partial t} + \mathbf{v}.\nabla_h \eta \text{ on } z = \eta,$$

where ∇_h is the horizontal gradient.

Dynamic boundary conditions: In contrast to Kinematic Boundary conditions, these conditions are instantaneous conditions expressing that at all times the external stresses on a boundary surface must be balanced by equivalent internal stresses. Hence, these are used to prescribe conditions at the interface between two fluid, fluid and solid etc. In the linearized potential case, Bernoulli's Equation is used to prescribe such conditions and is given by,

$$-\frac{\partial \phi}{\partial t} + \frac{1}{2}(u^2 + v^2 + w^2) + \frac{p}{\rho} + gz = C(t).$$

Lateral boundary conditions: at the *two* ends of the domain. These could be wall, transmissive, radiating, absorbing or periodic boundaries. In the simplest case we enforce periodic boundary conditions.

$$\phi(x, t) = \phi(x + L, t),$$

$$\phi(x, t) = \phi(x, t + T).$$

Solution to the linearized water wave problems can be found in many texts. Here, we follow some basic steps as outlined in (Dean & Dalrymple, 1991). The basic idea is to seek a separation of variables of the form,

$$\phi(x,z,t) = X(x) \cdot Z(z) \cdot \Im(t),$$

where $\Im(t) = sin(\sigma t)$. Here, σ can be thought of as an *angular frequency.* Even though the equation we are solving is linear and periodic, we have a non-linear boundary condition which depends on the solution. Since we have assumed an infinitesimal amplitude, the boundary condition at the free surface is linearized about the mean $z = 0$. Thus we get the following solution,

$$\eta = \frac{a}{2} cos(kx) cos(\sigma t),$$
$$\phi = \frac{a * g * cosh(k(h+z))}{2\sigma cosh(kh_b)} cos(kx) sin(\sigma t).$$

and the dispersion relation $\sigma^2 = gk * tanh(kh_b)$ where $k = 2 * \pi/L$. We briefly summarize the solution to the linearized water wave equation.

- Standing wave : One solution to the problem above is the *Standing Wave* which is,

  $$\phi_1 = \frac{a * g * cosh(k(h_b+z))}{2\sigma cosh(kh_b)} cos(kx) sin(\sigma t),$$
  $$\eta_1 = \frac{1}{g}\frac{\partial \phi_1}{\partial t}|_{z=0} = \frac{a}{2} cos(kx) cos(\sigma t).$$

  This type of wave doesn't propagate. At $kx = \frac{\pi}{2}$, $kx = 3\frac{\pi}{2}$ and so on, *nodes* exist and the free surface elevation is zero. Standing waves occur when incoming waves are completely reflected by walls. Hence a cosine (or sine) wave bounded by walls when left to itself is a standing wave problem much like strings in a guitar.

- Progressive wave : If we consider another *standing wave* problem like above but the *sine* terms replaced by *cosine*, for example :

$$\phi_2 = \frac{a * g * cosh(k(h_b + z))}{2\sigma cosh(kh_b)} sin(kx)cos(\sigma t),$$
$$\eta_2 = \frac{1}{g}\frac{\partial \phi_2}{\partial t}|_{z=0} = -\frac{a}{2} sin(kx)sin(\sigma t)$$

  This standing wave will have different nodes than the previous one. However, note that since we are dealing with a linear Laplace equation, if ϕ_1 and ϕ_2 are solutions to the equation then so is $\phi_1 \pm \phi_2$. Taking $\phi_2 - \phi_1$ we get a new velocity potential and hence new surface elevation given by,

$$\phi = -\frac{a * g * cosh(k(h_b + z))}{2\sigma cosh(kh_b)} sin(kx - \sigma t),$$
$$\eta = \frac{1}{g}\frac{\partial \phi}{\partial t}|_{z=0} = \frac{a}{2} cos(kx - \sigma t).$$

  This wave is a *traveling* wave and propagates in the positive x direction.

We see the presence of the terms kh_b in the solution to the linearized wave problem. The linear theory breaks down when kh_b exceeds π. Three noticeable regimes exists. Even though these limits are defined for small amplitude linearized assumptions, they hold for a general non-linear theory.

- $kh_b < \frac{\pi}{10}$ Shallow water : Long wave theory $\sigma^2 \approx gk^2 h_b$ which gives the wave speed $C = \sqrt{g * h_b}$. The waves are so long that the speed is *independent* of the wavelength. These are *Non-Dispersive* waves.

- $\frac{\pi}{10} < kh_b < \pi$ Intermediate water.

- $kh_b \geq \pi$ Deep water theory : $\sigma^2 \approx gk$ which gives $C = \sqrt{\frac{g}{k}}$.

It is useful to see what happens to the pressure field and the velocity field under a water wave. Below we give a brief explanation of these fields under the influence of a linear water wave. We will see that under the assumption of shallow water theory, the pressure is mainly hydrostatic whereas for large kh_b typically seen in coastal waters there is a significant dynamic component to the pressure. It is the approximation to this component that yields a dispersive water wave models. Also, the water particles under the wave move about in elliptical orbits. Hence, water velocities in the water wave literature are generally referred to as *orbital* velocities.

From the progressive wave equation above we can get u, w given by

$$u = -\phi_{,x} = cka\frac{cosh(k(h_b + z))}{2sin(kh_b)}cos(kx - \sigma t),$$
$$w = -\phi_{,z} = -cka\frac{sin(k(h_b + z))}{2sin(kh_b)}sin(kx - \sigma t),$$

where $c = \frac{g}{\sigma}tanh(kh_b)$. The particle paths are described by solving :

$$\frac{dx}{dt} = u\left(x(t), z(t), t\right),$$
$$\frac{dz}{dt} = w\left(x(t), z(t), t\right).$$

Since the above equations cannot be solved exactly, we'll have to use some approximations to solve for the particle paths. However, we have already made small amplitude and linear assumptions which suggests that the particles **orbit** around mean paths

$$x(t) = \xi + \Delta x(t)$$

$$z(t) = \zeta + \Delta z(t)$$

Using a Taylor expansion for the velocity field about the mean positions:

$$u(x, z, t) = u(\xi, \zeta, t) + u_{,x}\Delta x + u_{,z}\Delta z + h.o.t,$$

$$w(x, z, t) = w(\xi, \zeta, t) + w_{,x}\Delta x + w_{,z}\Delta z + h.o.t,$$

since $\Delta x(\Delta z)$ and $u_{,x}$ and other derivatives are all $O(a)$, the product terms can be dropped and we get :

$$x(t) - \xi = \int_0^t u(\xi, \zeta, t)dt,$$
$$z(t) - \zeta = \int_0^t w(\xi, \zeta, t)dt,$$

Thus we get :

$$x(t) - \xi = -\frac{a}{2}\frac{cosh(k(\zeta + h_b))}{sinh(kh_b)}sin(k\xi - \sigma t),$$
$$z(t) - \zeta = \frac{a}{2}\frac{sin(k(\zeta + h_b))}{sinh(kh_b)}cos(k\xi - \sigma t).$$

Note that the above equations describe an ellipse with center (ξ, ζ) i.e,

$$\left(\frac{x(t) - \xi}{A}\right)^2 + \left(\frac{z(t) - \zeta}{B}\right)^2 = 1.$$

The **pressure** equation can be determined from the linearized Bernoulli equation by equating values at an arbitrary z to the values at $z = 0$,

$$\frac{P}{\rho} + gz - \phi_{,t} = g\eta - \phi_{,t},$$

since $\eta \approx 0$ and $\eta = \frac{1}{g}\phi_{,t}\mid_{z=0}$ we get

$$P = -\rho gz + \rho\phi_{,t}.$$

We know that $-\rho gz$ is the **hydrostatic pressure**. Hence, the dynamic pressure is given by

$$P_D = \rho\phi_t = \rho g\eta K_p(z),$$

where, $K_p(z)$ is given by,

$$K_p(z) = \frac{cosh(k(h+z))}{cosh(kh)}.$$

Note, for *shallow-water* $K_p(z) = 1$ and hence $P_D = \rho g \eta$ and therefore $P = \rho g(\eta - z)$ which is hydrostatic. In this section we briefly introduced the classical small amplitude linear water wave theory to put the following sections in context. Even though the linear theory cannot be used in practical situations it is very useful in understanding the qualitative behavior of important wave phenomenon. In the next two sections we will describe the non-linear extension of the wave problem first through the classical schemes of Boussinesq, Rayleigh, Serre and others and then via the Boussinesq-Green-Naghdi equations which retains the rotational characteristic in the velocity fields. Even though we will define a set on invscid equations, the benefits of using the full rotational characterstics are two-fold, (1) easier extension to model the surf-zone, where waves due to viscous and turbelent forces and (2) to present a theory that can handle arbitrary levels of approximation to reproduce important wave phenomena like shoaling and dispersion.

2.2 Non-linear Extension : Classical water wave theory

From a historical perspective, and also to gain an understanding of various dispersion water wave models, it is important to understand what quantities are being approximated in the governing equations. An excellent work done in this regard is in the article (Barthélemy, 2004), where the author presents the classical dispersive water wave theory under a single unified approch of depth averaged equations. In the following section, some of these ideas will be presented. To do so, we'll first consider flow in 1 horizontal dimension with a uniform bathymetry and the domain is the same as was considered for the linearized water wave equation as shown in Figure (2.1). In classical theory, one works with non-dimensional Euler equations. The basic non-dimensional scales are given below :

$$\begin{aligned} x^* &= \frac{x}{L} \\ z^* &= \frac{z}{h_0} \\ u^* &= \frac{u}{\epsilon\sqrt{gh_0}} \\ p^* &= \frac{p}{\rho g h_0} \end{aligned}$$

Here a is the characteristic amplitude, L is the characteristic wavelength, and h_0 is the characteristic mean water depth and g is the acceleration due to gravity. Note that the pressure scaling is chosen to be hydrostatic. As defined earlier $\mu = kh_0$ and $\epsilon = a/h_0$. With these scalings the non-dimensional Euler equations in one horizontal dimension x and vertical dimension z, after dropping the $*$, are given as :

$$\begin{aligned} u_x + w_z &= 0, \\ \epsilon u_t + \epsilon(u^2)_x + \epsilon^2(uw)_z &= -p_x, \\ \epsilon\mu^2 w_t + \epsilon\mu^2 u w_x + \epsilon\mu^2 w w_z &= -p_z - 1, \end{aligned} \tag{2.1}$$

with the following boundary condition,

$$\begin{aligned} w = 0 \quad &\text{at } z = 0, \\ w = \eta_t + \epsilon u \eta_x \quad &\text{at } z = H, \end{aligned} \tag{2.2}$$

where $H = 1+\epsilon\eta$ is the non-dimensional total depth. Because in shallow waters the horizontal component of the velocity is quasi-uniform over the depth, the depth averaged velocity is a close approximation. We define depth average values as,

$$\bar{f} = \frac{1}{H}\int_0^H f dz.$$

The depth averaged continuity equation then reduces to,

$$\eta_t + (H\bar{u})_x = 0. \tag{2.3}$$

To derive (2.3), we used the *Leibnitz* integration rule

$$\int_0^H u_x dz = \frac{\partial}{\partial x}\int_0^H u dz - u|_H H_x,$$

and the boundary condition (2.2). The depth averaged momentum equations are a bit more involved. The first three terms in the x momentum equations after depth integration become

$$\begin{aligned}\frac{\epsilon}{H}(H\bar{u})_t - \frac{\epsilon}{H}u|_H H_t,\\ \frac{\epsilon^2}{H}\left(\frac{\partial}{\partial x}\int_0^H u^2 dz\right) - \frac{\epsilon^2}{H}u^2|_H H_x,\\ \frac{\epsilon^2}{H}\left(u|_H w|_H\right).\end{aligned}$$

Inserting these terms in (2.1)and using the boundary conditions (2.2) the depth averaged horizontal momentum equation is given by

$$\epsilon H\bar{u}_t + \epsilon^2\bar{u}\eta_t + \epsilon^2\frac{\partial}{\partial x}\int_0^H u^2 dz = -\int_0^H p_x dz.$$

Using (2.3) in the second term and noting that

$$\frac{\partial}{\partial x}\int_0^H \bar{u}^2 dz = -\int_0 H\bar{u}_x^2 + \bar{u}^2|_H H_x,$$

together with the fact that $\bar{u}$ is not a function of z we have

$$\frac{\partial}{\partial x}\int_0^H \bar{u}^2 dz = \frac{\partial}{\partial x}\left(\bar{u}^2 H\right),$$

and thus the depth averaged x momentum equation can be written as

$$\epsilon H\bar{u}_t + \epsilon^2 H\bar{u}\bar{u}_x + \epsilon^2\frac{\partial}{\partial x}\int_0^H \left(u^2 - \bar{u}^2\right) dz = -\int_0^H p_x dz. \tag{2.4}$$

At this point, we do not know the pressure distribution. If we assume hydrostatic pressure then we get the classic Shallow water equations in non-dimensional form. To get water wave models it will be useful to recast pressure entirely in terms of the velocity field. To do this we have to use the vertical (z) momentum equation. Let us rewrite the the z momentum equation in (2.1) as follows:

$$-p_z = 1 + \epsilon\mu^2\Gamma(x,z,t),$$

$$\Gamma(x,z,t) = w_t + \epsilon u w_x + \epsilon w w_z.$$

Integrating pressure from any z to H we get

$$-p(x,z,t) = (z - H) - \epsilon\mu^2\int_0^H \Gamma(x,\zeta,t)d\zeta.$$

Thus depth averaging the above equation gives us

$$-h\bar{p} = -\frac{1}{2}h^2 - \epsilon\mu^2\int_0^H\int_z^H \Gamma(x,\zeta,t)d\zeta.$$

Now we can use this expression of pressure in the x momentum equation (2.4) by noting that,

$$\int_0^H p_x dz = \frac{\partial}{\partial x}(h\bar{p}) - p|_H H_x.$$

Thus using the fact that pressure at the free surface is 0 and switching the order of integration involving Γ we get

$$\bar{u}_t + \epsilon \bar{u}\bar{u}_x + \frac{\mu^2}{H}\frac{\partial}{\partial x}\int_0^H z\Gamma(x,z,t)dz = -\frac{\epsilon}{H}\frac{\partial}{\partial x}\int_0^H \left(u^2 - \bar{u}^2\right) dz \qquad (2.5)$$

So far these equations have been exact. Note that Γ represents the vertical acceleration. Different approximations to the velocity structure and different scales of μ and ϵ will yield a multitude of water wave models. Most Boussinesq models start with the assumption of irrotationality of the velocity field. In the classic paper (Rayleigh, 1876), the velocity potential is shown to be harmonic for a flat bed and expanded in a Taylor series about $z = 0$ which then gives the following horizontal and vertical velocities

$$u(x,z,t) = u^b(x,0,t) - \frac{1}{2}\mu^2 z^2 \frac{\partial^2 u^b}{\partial x^2} + O(\mu^4)$$
$$w(x,z,t) = -z\frac{\partial u^b}{\partial x} + \frac{1}{3!}\mu^2 z^3 \frac{\partial^3 u^b}{\partial x^3} + O(\mu^4)$$

With this structure for the velocity field, various quantities in (2.5) can be approximated. For example the vertical acceleration Γ can be given as

$$\Gamma = -z\left(\bar{u}_{xt} + \epsilon \bar{u}\bar{u}_{xx} - \epsilon \bar{u}_x^2\right) + O(\mu^2, \epsilon\mu^2).$$

Based on different scales for ϵ, μ we get different models. A few classical ones are outlined below.

1. Airy equation when $\frac{\epsilon}{\mu^2} \ll 1$

$$\eta_t + \left(H\bar{u}\right)_x = 0,$$

$$\bar{u}_t + \epsilon \bar{u}\bar{u}_x + \eta_x = 0.$$

2. Boussinesq equation when $\frac{\epsilon}{\mu^2} \sim 1$

$$\eta_t + (H\bar{u})_x = 0,$$
$$\bar{u}_t + \bar{u}\bar{u}_x + g\eta_x = \frac{h_0^2}{3}\bar{u}_{xxt}.$$

3. Serre equation when $\epsilon \sim 1$

$$\eta_t + (H\bar{u})_x = 0,$$
$$\bar{u}_t + \bar{u}\bar{u}_x + g\eta_x - \frac{1}{3h}\frac{\partial}{\partial x}\left(h^3(\bar{u}_{xt} + \bar{u}\bar{u}_{xx} - (\bar{u}_x)^2\right) = 0.$$

In the next section we'll do away with the irrotational assumption and develop a BoussinesqGreen Naghdi model that works for a general varibale bathymetry.

2.3 Rotational water wave theory: Boussinesq - Green - Naghdi Model

Usually, in Boussinesq theories one works with the *non-dimensional* Euler equations for an incompressible fluid. A typical domain is show in Figure 2.2. Now, we will carry out the non-linear extension of the linear water wave theory for arbitrary bathymetry of the ocean with rotational characteristics. To do so, we will seek an approximation of the velocity field over the depth of the ocean. In particular we will approximate the velocity field as a polynomial over the depth and then solve the integrated non-dimensional momentum equations in a weighted sense. We will impose no irrotational assumption on the velocity field. In this regard, the equations will resemble the classical Green-Naghdi equations.

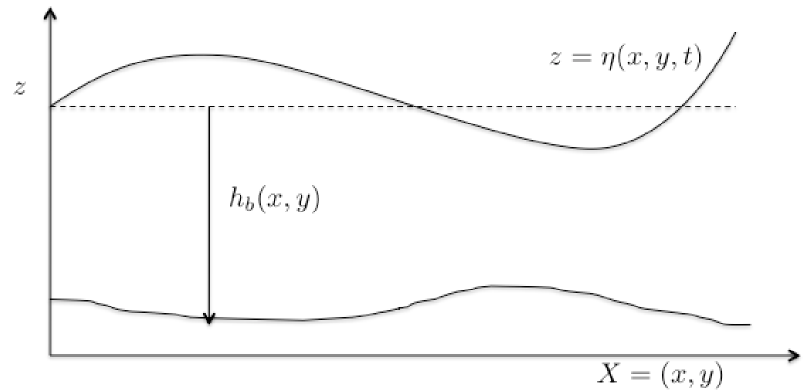

Figure 2.2: Domain showing bathymetry and surface elevation

The continuity equation reduces to the *free surface* equation given by,

$$\frac{\partial \eta}{\partial t} + \nabla \cdot \int_{-h_b}^{\eta} \mathbf{u} dz = 0. \tag{2.6}$$

where $\eta = \eta(x, y, t)$ is the free surface. The non-dimensional momentum equations, in *Cartesian co-ordinates*, are given by

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + w \frac{\partial \mathbf{u}}{\partial z} + \nabla P = 0. \tag{2.7}$$

$$\mu^2 \frac{\partial w}{\partial t} + \mu^2 \mathbf{u} \cdot \nabla w + \mu^2 w \frac{\partial w}{\partial z} + \frac{\partial P}{\partial z} + g = 0. \tag{2.8}$$

To eliminate pressure we integrate (2.8) from z to η, assuming a zero gauge pressure at the free surface to get

$$P(z) = \mu^2 \int_z^{\eta} \frac{\partial w}{\partial t} dz + \mu^2 \int_z^{\eta} \mathbf{u} \cdot \nabla w dz + \mu^2 \int_z^{\eta} w \frac{\partial w}{\partial z} dz + g(\eta - z), \tag{2.9}$$

where, $\nabla = [\partial/\partial x, \partial/\partial y]^T$, $\mathbf{u} = [u, v]^T$ and μ represents a dimensionless wave number. Note that the dynamic pressure is the sum of all the terms in the above equation that are multiplied by μ^2. As expected, when dealing with very small wave numbers the dynamic component of pressure can be neglected as is done in the case of shallow water equations.

In accordance with the classical Boussinesq and Green Naghdi theory, we follow the recipe outlined in (Zhang et al., 2013) where an approximate velocity field given by

$$\mathbf{u} \approx \bar{\mathbf{u}} = \sum_{n=0}^{N} \mu^{\beta_n} \mathbf{u_n}(x, y, t) f_n(z),$$

is inserted into the equations above to get arbitrary levels of approximation. For the sake of completeness we outline the steps in constructing a Rotational Boussinesq - Green - Naghdi approximation of the Euler equations.

1. Define a level of wave approximation $O(\mu^N)$ and choose appropriate basis functions f_n.

2. Insert the approximate velocity field into the free surface equation (2.6), retaining all the terms up to the desired level of approximation.

3. Insert the approximate velocity field into the pressure equation (2.9) to get $\bar{P}$.

4. Insert the approximate velocity field into the horizontal momentum equation (2.7). Integrate in weighted residual sense, using the $N+1$ basis functions used in the approximated velocity field, i.e

$$\int_{-h_b}^{\eta} f_m \left(\frac{\partial \bar{\mathbf{u}}}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \bar{w} \frac{\partial \bar{\mathbf{u}}}{\partial z} + \nabla \bar{P} \right) dz = 0. \qquad m = 0, \ldots, N \quad (2.10)$$

where $\bar{w}$ represents the approximate vertical velocity field which can be determined from the approximate horizontal velocity field (Zhang et al., 2013).

We will focus mainly on the $O(\mu^2)$ equations. As derived in (Zhang et al., 2013), the approximate velocity field is given by

$$\begin{aligned} \bar{\mathbf{u}} &= \mathbf{u_0} + \mu^2 \mathbf{u_1} f_1(q) + \mu^2 \mathbf{u_2} f_2(q), \\ \bar{w} &= -\nabla \cdot \mathbf{u_0} H q - \mathbf{u_0} \cdot \nabla h_b + O(\mu^2), \end{aligned} \tag{2.11}$$

where q is a sigma-type co-ordinate given by $q = \frac{z+h_b}{h_b+\eta}$ and $H = \eta + h_b(x, y)$ is the *total* water depth. Sigma type co-ordinates are very useful in geophysical applications as it allows surfaces to follow model terrain. The convergence properties of such an expansion are discussed in (Zhang et al., 2013).

Of particular importance are the basis functions that are used in the approximation of the velocity field over the depth. Various basis functions $f_m(q)$, for example monomials, shifted Legendre polynomials etc . can be used. Moreover, basis functions can be optimized to give the best linear dispersion or shoaling approximation. This technique is elaborated in (Zhang et al., 2013).

Following the steps above, we end up with the free-surface evolution equation and the momentum equations to solve for η, $\mathbf{u_0}$, $\mathbf{u_1}$ and $\mathbf{u_2}$. The surface elevation equation is given by,

$$\eta_{,t} + \nabla \cdot (\mathbf{u_0} H + \mu^2 \sum_{m=1}^{2} \mathbf{u_m} H c_m) = 0. \tag{2.12}$$

The momentum equations are given by,

$$
\begin{aligned}
&\mathbf{u_{0,t}}Hc_1^m + \mathbf{u_0}\cdot\nabla\mathbf{u_0}Hc_2^m + g\nabla\eta Hc_3^m + \mu^2\sum_{n=1}^{2}(\mathbf{u_{n,t}}Hc_4^m - \mathbf{u_n}\eta_{,t}c_5^m) \\
&- \mu^2\left[\frac{1}{2}\nabla(\nabla\cdot\mathbf{u_{0,t}})H^3c_6^m + \nabla\cdot\mathbf{u_{0,t}}\nabla HH^2c_7^m + \nabla(\mathbf{u_{0,t}}\cdot\nabla h_b)H^2c_8^m\right. \\
&\left. + \mathbf{u_{0,t}}\cdot\nabla h_b\nabla\eta Hc_9^m - (\nabla\cdot\mathbf{u_{0,t}})H^2\nabla h_b c_{10}^m\right] \\
&+ \mu^2\sum_{n=1}^{2}\left[(\mathbf{u_n}\cdot\nabla\mathbf{u_0} + \mathbf{u_0}.\nabla\mathbf{u_n})Hc_{11}^m - \mathbf{u_n}\nabla\cdot(\mathbf{u_0}H)c_{12}^m\right] \\
&+ \mu^2H^2\left[(\nabla\cdot\mathbf{u_0})^2 - \mathbf{u_0}\cdot\nabla(\nabla\cdot\mathbf{u_0})\right](\nabla\eta c_{13}^m + \nabla h_b c_{14}^m) \\
&+ \frac{\mu^2}{2}H^3\nabla\left[(\nabla\cdot\mathbf{u_0})^2 - \mathbf{u_0}\cdot\nabla(\nabla\cdot\mathbf{u_0})\right]c_{15}^m \\
&- \mu^2H\nabla\eta\mathbf{u_0}\cdot\nabla(\mathbf{u_0}\cdot\nabla h_b)c_{16}^m - \mu^2H^2\nabla\left[\mathbf{u_0}\cdot\nabla(\mathbf{u_0}\cdot\nabla h_b)\right]c_{17}^m = 0,
\end{aligned}
\tag{2.13}
$$

$\forall m \in [0, 2]$.

where all the coefficients c_k^m are defined in the appendix. There are many variants of the Green - Naghdi equations based on Boussinesq type scaling (Bonneton et al., 2011). For here on, we will refer to the Boussinesq - Green - Naghdi equations, as discussed above, as the $R - GN$ equations (to emphasize rotational characteristics).

2.4 Dispersion Charactersitics of Boussinesq Models

To understand dispersion in Boussinseq systems it is useful to look at the model equation given by

$$u_t - u_{xxx} = 0.$$

If we carry out the Fourier transform of this equation then we can see that for each wavenumber k the speed is given by $c = k^2$. Thus the speed is a non-linear function of the wave number. This behavior characterizes dispersive equations. In contrast, for a linear advective equation given by

$$u_t - au_x = 0,$$

the speed is a constant and is equal to a. Shallow water equations are governed by hyperbolic partial differential equations like the linearized advection equation and as such are *non-dispersive*. This was verified for the linearized water wave equation where we saw that for long waves ($kh_b \approx 0$), wave speed is independent of the wave number. Boussinesq equations on the other hand exibit dispersive characteristics. The $R-GN$ equations are non-linear dispersive equations and its dispersion characteristics are analyzed by comparing it with the linearized equation discussed in the first section of this chapter. For lower order system like the $O(\mu^2)$ system in this thesis, it is possible to arrive at dispersion results for generalized basis functions $f_m(q)$. If we define

$$f_0 = 1,$$

$$f_1 = a + q,$$

$$f_2 = b + cq + q^2,$$

then the general dispersion relation for the $R-GN$ equations with any choice of (a, b, c) will be

$$\frac{C^2}{gh_b} = \frac{1 + (\frac{1}{6} + \frac{1}{2}(b - ac))(kh_b)^2}{1 + (\frac{1}{2} + \frac{1}{2}(b - ac))(kh_b)^2} \tag{2.14}$$

In the following figures we show the comparison plot of the dispersion relation of $R-GN$ equations and the linear Stokes dispersion for both shifted Legendre

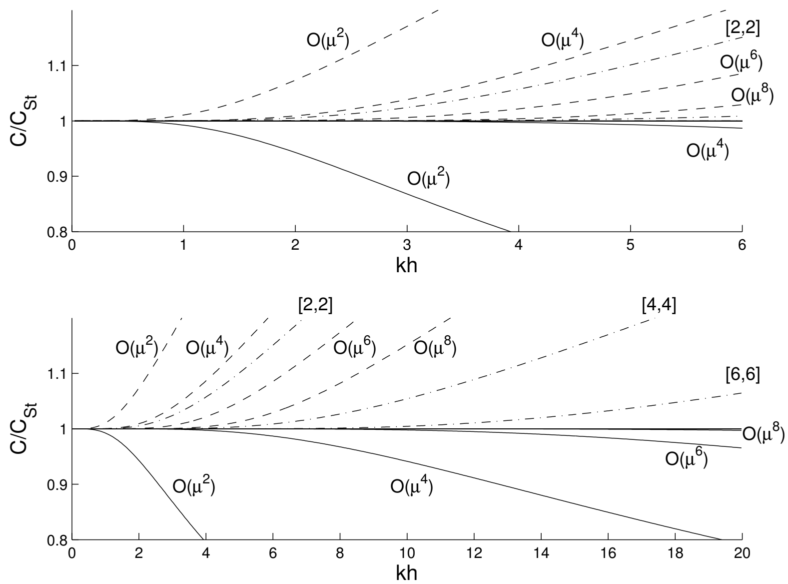

Figure 2.3: Approximate dispersion relations compared to linear dispersion. Top figure is zoomed in on lower kh_b values. Shifted Legendre basis $(-)$ and monomials $(--)$ are used as the basis functions in the $R-GN$ equations.

basis and monomials. For reference we also show Padé$[2,2]$, Padé$[4,4]$ and Padé$[6,6]$.

We see that the $O(\mu^2)$ $R-GN$ equations give about 10% error in linear dispersion when the wave number $kh_b < \pi$ but large errors when $kh_b > 10$. We can get more accuracy in linear dispersion if we use the $O(\mu^4)$ equations which show 10% error all the way up to $kh_b \approx 10$. However, $O(\mu^4)$ are extremely non-linear and from a numerical point almost intractable.

In this chapter we described the inviscid $O(\mu^2)$ $R-GN$ equations which are Green-Naghdi equations based on Boussinesq scaling and allow for a natural extension to model rotational characteristics in the surf zone. These equations are extremely non-linear and dispersive. However, there is limit to linear dispersion in using these equations. In the next chapter we will extend the equations to include viscous terms that will be necessary to model surf zones.

Chapter 3

Building a surf-zone model

In the previous chapter we desicribed the $R-GN$ equations to model coastal water waves. These equations are highly dispersive and non-linear but are invscid. In this chapter we will analyze an important phenomenon known a shoaling and will also extend the invscid equations to account for turbulent stresses that are crucial in modeling wave breaking. We will also describe techniques to generate and absorb waves in the boundaries. This will complete the construction of a true surf-zone model.

3.1 Shoaling

When a wave train propagates towards a gentle plane slope from a normal incidence, the train will gradually slow down since the speed is proportional to the square root of the bathymetric depth. In order to maintain the energy in the water column, the wave will then change its height. This process, during which an approaching wave train will change its wave height based on its offshore condition and local water depth, is known as shoaling. In devising Boussinesq equations it is extremely important to understand the range of applicability of the equations. This is usually done through perturbation analysis of the system to obtain theoretical representations for linear dispersion and shoaling. While the previous chapter included dispersion analysis of the $R-GN$ equations, in this section we'll outline the basic steps for analyzing shoaling errors. The complete details are provided in(Zhang et al., 2013). The following steps are carried out in determining the shoaling error of the model:

- Assume multiple scale expansion in space that has fast and slow spatial derivatives.
- Define the water depth to be slowly varying.
- Insert the multiple scale expansion in the governing equations and gather first order and second order terms.
- Find the (second order) relation of surface elevation and bathymetry. The corresponding co-efficient γ_h is the shoaling gradient which is a function of wave number and can be compared to the linearized equation.

It should be noted that all errors in shoaling gradient are negative, any cumulative shoaling errors for a wave traveling from deeper waters to shallow waters would be small and is preferred for stability reasons. We plot the shoaling gradient and the cumulative shoaling error as function of wave number in Figure 3.1. Like the dispersion errors discussed in the previous chapter, $O(\mu^2)$ equations show about 10% error for wave numbers up to $kh_b = \pi$, when the basis functions are chosen to be the shifted Legendre Polynomials. Since

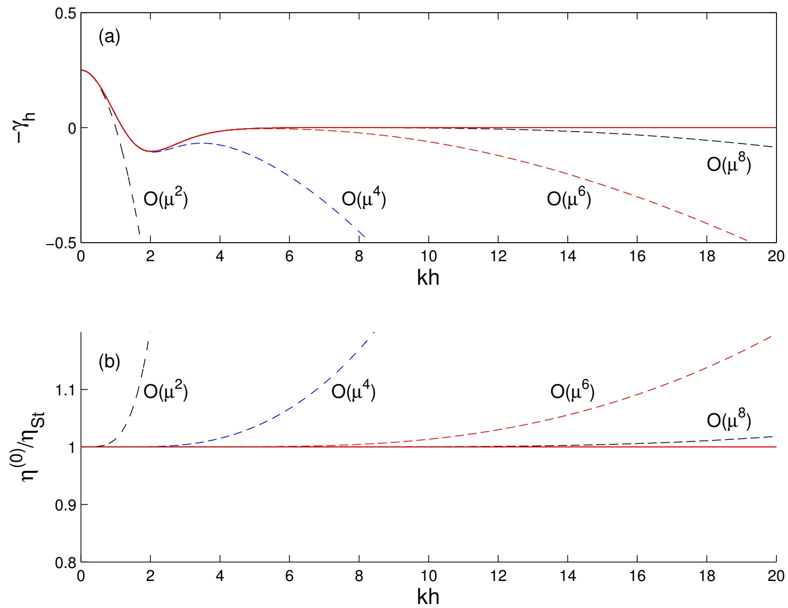

Figure 3.1: Shoaling errors for shifted Legendre polynomials. Top figure is the shoaling gradient while the bottom figure represents the cumulative shoaling error.

there is a flexibility in choosing the basis functions in deriving the $R-GN$ equations, we can construct basis functions that optimize shoaling errors. For example, the following choice

$$f_0 = 1,$$

$$f_1 = a + q,$$

$$f_2 = b + q^2,$$

where $a = -0.432$ and $b = -1/5$ gives low shoaling errors for wave numbers up to 4. Figure 3.2 shows the shoaling error for different choices of a. The optimization was done for wave numbers in the interval $[0\,,\,4]$.

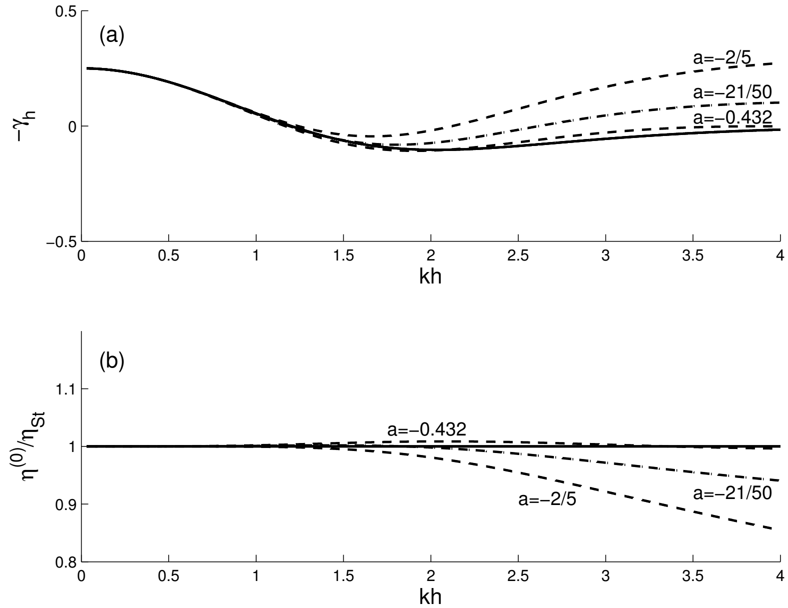

Figure 3.2: Shoaling errors for optimized basis functions. Top figure is the shoaling gradient while the bottom figure represents the cumulative shoaling error.

3.2 Wave breaking: Including viscous stresses

Wave height can be increased due to many reasons, e.g, wave shoaling, continuous wind action, superposition of various wave modes or due to combined wave refraction and diffraction. When the wave height exceeds a certain threshold, the wave system will become unstable and will *break* to release excess energy. This is usually a turbulent process which introduces rotational characteristics in the velocity field. Hence any model based on the irrotational assumption will lead to large errors in the velocity field. In contrast, the $R-GN$ equations can naturally include viscous stress since there is no irrotational assumption used in the derivation. Thus viscous terms in the Navier-Stokes equation respresented as eddy viscosity are added to the inviscid $R-GN$ equations with proper scaling to produce the energy dissipation under the breaking wave crest, while the eddy viscosity is modeled by the depth-integrated turbulent-kinetic-energy equation. This eddy viscosity model is coupled with the wave model to model rotational flow naturally in the surf zone. However, keeping all the dispersive terms we will in principle never be able to simulate the complex free surface found in extreme breaking (plunging breakers) and hence there is an upper limit on the accuracy of the model.

To add viscous terms we modify (2.10) with the following:

$$\int_{-h_b}^{\eta} f_m(\frac{\partial \bar{\mathbf{u}}}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \bar{w}\frac{\partial \bar{\mathbf{u}}}{\partial z} + \nabla \bar{P} - \mu^2 \nabla \cdot \tau_{\mathbf{xx}} - \frac{\partial}{\partial z}\tau_{z\mathbf{x}} dz) = 0. \qquad m = 0 \dots N \tag{3.1}$$

Here $\tau_{\mathbf{xx}}$ is the breaking stress while $\tau_{\mathbf{x}z}$ is the bed-generated bottom stress. Both these terms will act in damping the wave energy and will be treated as separate terms with different evolution equations. This division has a physical basis, as bed generated bottom stresses diffuse upwards while the breaking stresses are surface stresses that diffuse downwards. Note that the pressure equation remains inviscid as given by (2.9). The rotational extension of the $R-GN$ equations will be complete with the definition of both the viscous terms which we will consider in the following paragraph.

Because breaking dissipation and bottom stress effects are separated, they will be modeled with separate eddy viscosities, $\nu_{t1}(\mathbf{x}, t)$ and $\nu_{t2}(\mathbf{x}, z, t)$.

Then the respective turbulent stress terms become,

$$\nabla \cdot \tau_{\mathbf{xx}} = \nabla \cdot \left[\nu_{t1}(\mathbf{x}, t)(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \right], \tag{3.2}$$

$$\frac{\partial}{\partial z} \tau_{\mathbf{x}z} = \frac{\partial}{\partial z} \left[\nu_{t2}(\mathbf{x}, z, t)(\mathbf{u}_z + \mu^2 \nabla w) \right]. \tag{3.3}$$

In order to include the bottom friction we perform integration by parts on (3.3). Thus we obtain the following:

$$\begin{aligned} \int_{-h_b}^{\eta} f_m \frac{\partial}{\partial z} \tau_{\mathbf{x}z} dz &= \int_{h_b}^{\eta} \frac{\partial}{\partial z}(f_m \tau_{\mathbf{x}z}) dz - \int_{-h_b}^{\eta} \frac{\partial f_m}{\partial z} \tau_{\mathbf{x}z} dz \\ &= (f_m \tau_{\mathbf{x}z})|_{-h_b}^{\eta} - \int_{-h_b}^{\eta} \frac{\partial f_m}{\partial z} \tau_{\mathbf{x}z} dz \end{aligned} \tag{3.4}$$

$\tau_{\mathbf{x}z}(\eta)$ is the air-water shear stress and comes from wind forcing. The bottom stress $\tau_{\mathbf{x}z}(-h_b)$ depends on the bed roughness or vegetation type. These are usually placed in a drag framework such as $\tau_{\mathbf{x}z}(-h_b) = C_f \mathbf{u}_b |\mathbf{u_b}|$. To make matters simpler we take the depth averaged eddy viscosity $\nu_{t2}(\mathbf{x}, z, t)$ given by $\epsilon C_f H |\mathbf{u_b}|$.

We still haven't defined the breaking stress eddy viscosity $\nu_{t1}(\mathbf{x}, t)$. In deep water breaking is related to the steepness of the wave whereas in shallow water it is related to the ratio of wave height and the local bathymetric depth. In both cases it is governed by turbulence as the wave builds up excess kinetic energy. In the simplest model $\nu_{t1}(\mathbf{x}, t)$ is related to the $k - l$ model which describes evolution of the turbulent kinetic energy based on a mixing length. The evolution equation of turbulent kinetic energy is given by:

$$\frac{Dk}{Dt} = -\nabla \cdot \mathbf{T}' + \mathcal{P} - \epsilon. \tag{3.5}$$

The turbulent energy flux T' is modeled with a gradient-diffusion hypothesis as given by

$$\mathbf{T}' = -\frac{\nu_{t1}}{\sigma_k}\nabla k, \tag{3.6}$$

where σ_k is the turbulent Prandtl number for kinetic energy and is generally taken to be 1.0. The production term is then given by

$$\mathcal{P} = \nu_{t1}\left[\nabla\mathbf{u}\cdot(\nabla\mathbf{u} + (\nabla u)^T) + 2\mathbf{u}_z\cdot\nabla w + \frac{1}{\mu^2}\mathbf{u}_z\cdot\mathbf{u}_z + 2w_z^2\right], \tag{3.7}$$

where the $O(\mu^2)$ terms are neglected. The turbulent viscosity is defined by

$$\nu_{t1} = ck^{1/2}\bar{l}_m \tag{3.8}$$

where $\bar{l}_m$ is the vertically averaged mixing length, l_m given by $l_m = \kappa q\sqrt{(}1 - q)H$ and $\kappa = 0.412$ is the von Karman constant. At high Reynolds number the dissipation rate is modeled as

$$\epsilon = c^3 k^{3/2}/\bar{l}_m^{\,2}. \tag{3.9}$$

A value of $c = 0.55$ yields the correct behavior for shear flows in the $k - l$ model. Thus the non-dimensional turbulent kinetic energy equation is given by:

$$\frac{Dk}{Dt} = \frac{\mu}{\nu_{t1}\sigma_k}\nabla\cdot(\nu_{t1}^2\nu_{\mathbf{t1}}) + \mu\frac{c^2\bar{l}_m^{\,2}}{2\nu_{t1}}\mathcal{P} - \frac{1}{\mu}\frac{c^2}{2\nu_{t1}}\nu_{t1}^2. \tag{3.10}$$

Integrating (3.10) over the depth we get the depth integrated eddy viscosity equation

$$\frac{\partial\nu_{t1}}{\partial t} + \nabla\nu_{t1}\cdot\sum_{m=0}^{2}\mathbf{u}_m g_m|_0^1 - \frac{\mu}{\nu_{t1}\sigma_k}\nabla\cdot(\nu_{t1}^2\nu_{\mathbf{t1}}) - \mu\frac{c^2\bar{l}_m^{\,2}}{2\nu_{t1}}\int_0^1\mathcal{P}dq + \frac{1}{\mu}\frac{c^2}{2\nu_{t1}}\nu_{t1}^2, \tag{3.11}$$

where g_m is a constant depending on the basis functions $f_m(q)$ used in the $R - GN$ equation and is detailed in the Appendix. This one equation model is coupled with the $R - GN$ equations to account for the turbulent breaking stresses.

3.3 Wave generation and absorption: Sponge Layers

The generation and absorption of waves at the boundary are important for the numerical simulation of Boussinesq and other water wave models. Usually sponge layers have been used to remove unwanted signals at the edge of the domain. In (Zhang, Kennedy, Panda, Dawson, & Westerink, 2014), the authors developed a source function method for the combined wave generation and absorption using modified sponge layers. In this thesis, we'll be using these sponge layers to generate and absorb linear, non-linear, regular and irregular waves.

The main concept of the sponge layer is to include source terms which in general can be written as follows

$$\mathbb{A}_1 \left\{\mathbf{a}_t\right\} + \mathbb{L}_1 \left\{\mathbf{a}_t\right\} + \cdots = \omega_1 \mathbb{A}_1 \left\{\mathbf{a}_{imp} - \mathbf{a}_t\right\} + \omega_2 \mathbb{L}_1 \left\{\mathbf{a}_t\right\}, \tag{3.12}$$

where $\{\mathbf{a}_t\}$ is the vector of variables, ω_1 and ω_2 are damping co-efficients, the matrix $\mathbb{A}_1$ and $\mathbb{L}_1$ represent algebraic multipliers and spatial differential operators of $\{\mathbf{a}_t\}$ respectively. To apply sponge layers to the domain when using $R - GN$ equations we define L_1 and L_3 to be the absorption and generation length while L_{samp} is identified as the domain of interest. Thus we specify forcing functions $\{\mathbf{a}_{imp}\}$ to be non-zero only within the generation zone i.e for $\mathbf{x} \leq L_1$. The damping coefficient ω_1 is described below:

$$\omega_1(\mathbf{x}) = \begin{cases} \frac{\tilde{\omega}}{L_1}(n+1)\left(1 - \frac{\mathbf{x}}{L_1}\right) & \text{if } \mathbf{x} \leq L_1 \\ \frac{\tilde{\omega}}{L_3}(n+1)\left(1 - \frac{\mathbf{x}-(L_1+L_{samp})}{L_3}\right) & \text{if } \mathbf{x} \geq L_3 \end{cases} \tag{3.13}$$

where $\tilde{\omega}$ is the strength of the sponge layer and is taken to be $10 * \sqrt{gh_b}$. In practise ω_2 is taken as zero. For $R - GN$ equations we usually impose the surface elevation η in the generation zone while the velocities evolve as a response to the surface elevation. In the absorption zone too, only the surface elevation is damped to 0 while the velocities evolve naturally. This means that the absorption zone has to be long enough so that the velocities are not reflected back into the domain of consideration. Although this method works well for generating/absorbing linear wave trains, random and non-linear waves can also be generated and the process is detailed in (Zhang et al., 2014). In the Figure (3.3) we generate linear waves of height $H = 0.0001$ and time period $T_p = 1.91s$.

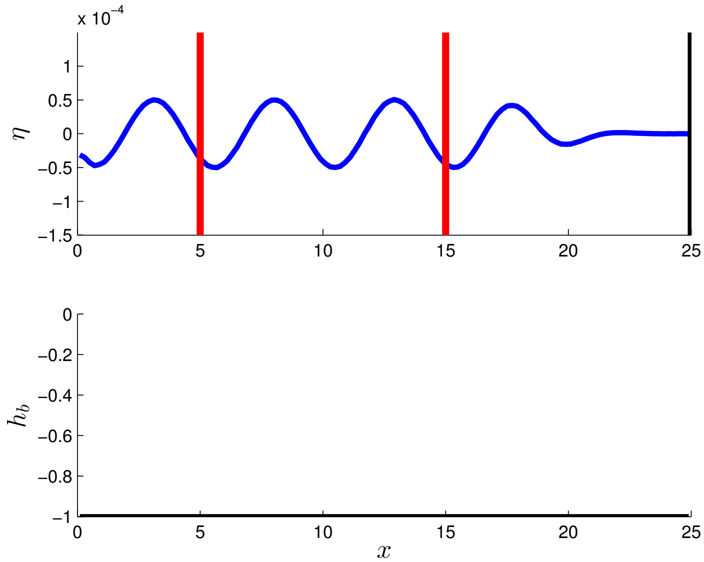

Figure 3.3: Wave generation, propagation and absorption

The time history of the surface elevation is shown in figure (3.4)

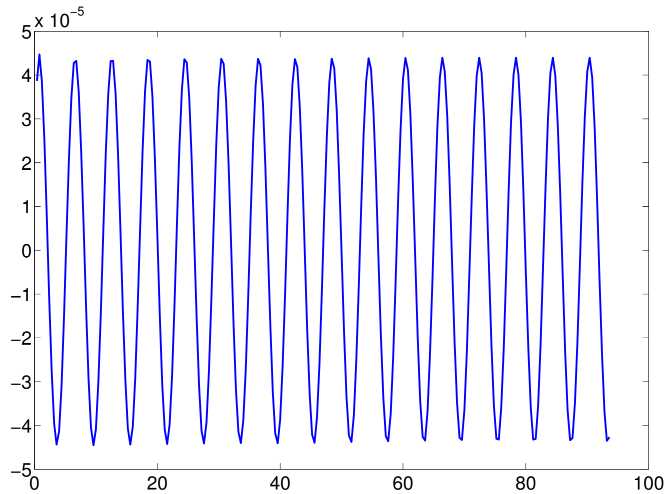

Figure 3.4: Time history of surface elevation at a fixed location in the sample zone.

Chapter 4

Numerical Methods

In the previous chapters we detailed the $O(\mu^2)$ $R-GN$ equations to model complex near-shore wave phenomena. These equations are highly non-linear with dispersive characterstics that include mixed spatio-temporal derivatives. The coupling of velocity coefficients u_0, u_1, u_2 along with the surface elevation equation makes it extremely challenging to develop stable numerical schemes in arbitrary grids. In this thesis we will propose a local discontinuous Galerkin (LDG) method to solve the $R-GN$ equations and perform verification and validation for challenging test cases in $1D$. We will also do a careful $L2$ stability analysis to establish linear stability of our method. Although we will focus only on the $1D$ case, the method will be quite general and can be extended easily to the full $2D$ simulation. Verification, validation and linear stability will give us the confidence to proceed with the development of a numerical method for the $2D$ case in arbitrary grids.

In the following sections we outline the LDG scheme and follow it with the numerical discretization of the $1D$ $R-GN$ equations.

4.1 The Discontinuous Galerkin method

In the following paragraphs we describe some of the basic features of this method as applied to a linear scalar hyperbolic equation and the second order steady heat equation. The linear transport equation can be written as

$$u_t + \nabla \cdot (\mathbf{a}u) = 0 \,\mathrm{in}\, \Omega \times [0, T] \,,$$

$$u(t = 0) = u_0 \,\mathrm{on}\, \partial\Omega.$$

To discretize the transport equation in space by using a DG method, we first triangulate the domain Ω. We then seek a *discontinuous* approximate solution u_h, which, in each element K of the triangulation, belongs to the space of polynomials of degree at most k. We denote this space by $\mathcal{V}(K)$.We then determine the approximate solution on the element by weakly enforcing the transport equation as follows:

$$\int_K (u_h)_t v - \int_K \mathbf{a}u_h \cdot \nabla v + \int_{\partial k} \mathbf{a}\hat{u}_h \cdot nv = 0,$$

for all $v \in \mathcal{V}(K)$. Since u_h is discontinuous across element boundaries, we need to find the right numerical trace or *discrete flux* $\mathbf{a}\hat{u}_h$ to render the scheme stable. Let $\mathbf{x}$ be a point in the set $\overline{\partial K^+} \bigcap \overline{\partial K^-}$ and let $\mathbf{n}^\pm$ denote the outward normal to $\partial K^\pm$. Let $u_h^\pm$ denote the value of u_h as $\mathbf{x}$ approaches the edge from $K^\pm$ and set the following quanatities:

$$\{u_h\} = \frac{1}{2}\left(u_h^+ + u_h^-\right),$$

$$[|u_h|] = u_h^+\mathbf{n}^+ + u_h^-\mathbf{n}^-,$$

as the average and jump of the discrete solution at an element edge. Note that the jump of a scalar is defined as a vector quantity in $2D$ and higher dimensions. With this, the following numerical trace:

$$\mathbf{a}\hat{u}_h = \{u_h\} + C\,[|u_h|]\,,$$

will render the scheme stable. Here C is a positive definite matrix. For example $C = \frac{1}{2}|\mathbf{a}.\mathbf{n}|I_d$ where I_d is the identity matrix yields the calassic upwinding scheme. Similar flux choices have been used in finite volume methods and the local discontinuous Galerkin method can be thought of as a higher order extension of finite volume methods.

Now we describe the LDG method for the discretization of the steady heat equation which is given by:

$$\begin{aligned} -\Delta u &= f \,\text{in}\, \Omega, \\ u &= 0 \,\text{on}\, \partial\Omega. \end{aligned} \tag{4.1}$$

As discussed earlier, the idea of the LDG method is to reduce higher order equations into a system of first order equations which, in the present example become:

$$\begin{aligned} \mathbf{q} &= \nabla u, \\ -\nabla\cdot\mathbf{q} &= f \,\text{in}\, \Omega, \\ u &= 0 \,\text{on}\, \partial\Omega. \end{aligned}$$

The LDG numerical method is obtained as follows. After discretizing the domain Ω into elements J, the approximate solution (q_h, u_h) on the element is taken in the space $(\mathcal{Q}(J), \mathcal{U}(J))$ and is determined by requiring that:

$$\begin{aligned} \int_J \mathbf{q_h}\cdot\mathbf{v} &= -\int_J u_h \nabla\cdot\mathbf{v} + \int_{\partial J} \hat{u}_h \mathbf{v}\cdot\mathbf{n}, \\ \int_J \mathbf{q_h}\cdot\nabla w - \int_{\partial J} w\hat{\mathbf{q}}_h\cdot\mathbf{n} &= \int_J f w, \end{aligned}$$

for all $(\mathbf{v}, w) \in (\mathcal{Q}(J), \mathcal{U}(J))$. Thus we have two numerical traces $\hat{u}_h, \hat{\mathbf{q}}_h$ that needs to be defined correctly to render this scheme stable. The following choice yields a stable scheme (Arnold, Brezzi, Cockburn, & Marini, 2002)

$$\hat{u}_h = \{u_h\} + \mathbf{C_{12}} \cdot [|u_h|]\,,$$

$$\hat{\mathbf{q}}_h = \{\mathbf{q_h}\} - C_{11}\,[|u_h|] - \mathbf{C_{12}}\,[|\mathbf{q_h}|]\,,$$

where the jump in $\mathbf{q_h}$ is defined to be a scalar given by

$$[|\mathbf{q_h}|] = \mathbf{q_h}^+ \cdot \mathbf{n}^+ + \mathbf{q_h}^- \cdot \mathbf{n}^-.$$

In this section we briefly introducted the LDG method as applied to a linear hyperbolic equation and an elliptic equation. The $R-GN$ equations are coupled hyperbolic-elliptic equations and some of these ideas presented here will be elaborated in the context of discretizing the $R-GN$ equations.

4.2 Numerical Disecretization of the R-GN equations

We investigate the LDG method for the spatial discretization of the R-GN equations given by (2.12) - (2.13). The resulting semi-discrete equations are then integrated in time using an explicit Runge-Kutta method to evolve the equations from suitable initial conditions. In this thesis we'll only focus on the $1D$ formulation of the R-GN equations. The full $2D$ equations will be simple extension of the work considered in this thesis.

In this section we will define the numerical method in the abstract setting while all the implementation details are presented in the following subsections. Let

$\Omega = [0, L]$ be the spatial domain. Define a partition

$$0 = x_{1/2} < x_{3/2} < \cdots < x_{J+1/2} = L,$$

and define,

$$\begin{aligned} E_j &= \left[x_{j-1/2}, x_{j+1/2}\right], \\ \mathcal{E} &= \left\{x_{j+1/2}\right\}, \\ h_j &= x_{j+1/2} - x_{j-1/2}, \\ h &= \max_j h_j, \end{aligned} \tag{4.2}$$

to be the finite element, set of boundary points, element size and the maximum element size respectively. Construct a set of test functions V_h^K on the partition, consisting of piecewise polynomials of degree K:

$$V_h^K = \{v : v|_{E_j} \in \mathbb{P}_K(E_j) \quad \forall j = 1, \ldots, J\}. \tag{4.3}$$

Let us denote,

$$\begin{aligned} v(x_{j+1/2}^+) &= \lim_{\epsilon \to 0^+} v(x_{j+1/2} + \epsilon), \\ v(x_{j+1/2}^-) &= \lim_{\epsilon \to 0^+} v(x_{j+1/2} - \epsilon). \end{aligned}$$

The jump and average of v at the endpoints of E_j are:

$$\begin{aligned} \left[|v(x_{j+1/2})|\right] &= v(x_{j+1/2}^-) - v(x_{j+1/2}^+), \\ \left\{v(x_{j+1/2})\right\} &= \frac{1}{2}\left(v(x_{j+1/2}^-) + v(x_{j+1/2}^+)\right). \end{aligned} \tag{4.4}$$

For any $v \in V_h^K$, we can write v as

$$v = \sum_{j=1}^{J} \sum_{i=0}^{K} \tilde{v}_i^j \phi_i(x), \tag{4.5}$$

where $\{\phi_i\}$ is a basis for $\mathbb{P}_K$. We chose $\phi_i = P_i$, where P_i is the normalized Legendre polynomial (Hesthaven & Warburton, 2007). Given $u^h \in V_h^K$, all derivatives of u^h are calculated in an *LDG* sense described below. Define:

$$\lambda^h = u^h_x,$$

$$\mathbb{B}(\lambda^h, w) = \mathbb{L}_{u^h}(w),$$

where $\mathbb{B} : V_h^K \times V_h^K \to \mathbb{R}$ is the *bi-linear* form and $\mathbb{L}_{u^h} : V_h^K \to \mathbb{R}$ is the *linear* form given by

$$\begin{aligned} \mathbb{B}(\lambda^h, w) &= \sum_j \left(\lambda^h, w\right)_{E_j}, \\ \mathbb{L}_{u^h}(w) &= -\sum_j \left(u^h, w_x\right)_{E_j} + \left\langle \hat{u}^h, [|w|]\right\rangle_{\mathcal{E}}, \end{aligned} \tag{4.6}$$

where $w \in V_h^K$ and $(\ ,\ )$ denotes the standard L_2 inner product. In a similar fashion, we compute u^h_{xx}, u^h_{xxx} and so on. Looking ahead, let us define the following *bi-linear* form:

$$\mathbb{B}_\sigma\left(u^h, w\right) = \sum_j \left(u^h, w\right)_{E_j} + \sigma \left\langle \left[|u^h|\right], [|w|]\right\rangle_{\mathcal{E}}, \tag{4.7}$$

Where $\sigma \geq 0$. Note, $\hat{u}^h = F(u^{h-}, u^{h+})$ is the single valued *flux* function evaluated at the edges of E_j. Various flux functions can be found in the DG literature. The simplest flux is the average flux given by:

$$F(u^-_{j+1/2}, u^+_{j-1/2}) = \left\{u(x_{j+1/2})\right\}. \tag{4.8}$$

To calculate the inner products we define an affine mapping given by (Hesthaven & Warburton, 2007):

$$x \in E_j : x(\xi) = x_{j-1/2} + \frac{1+\xi}{2} h_j. \tag{4.9}$$

This maps $x \mapsto [-1, 1]$, where we utilize the Gaussian quadrature formulae so that the integrals are evaluated exactly.

4.2.1 LDG scheme for the R-GN equations

The R-GN equations (2.12) - (2.13) can be written as:

$$\varphi = Rhs_{\eta}, \tag{4.10a}$$

$$\mathcal{L}\left[s_0\right] = Rhs_{u_0}, \tag{4.10b}$$

$$s_1 = Rhs_1, \tag{4.10c}$$

$$s_2 = Rhs_2, \tag{4.10d}$$

where $\varphi = \eta_t$, $s_0 = u_{0,t}$, $s_1 = u_{1,t}$, $s_2 = u_{0,t}$; and $\mathcal{L}$ is an elliptic operator given by $A + B\frac{\partial}{\partial x} - C\frac{\partial^2}{\partial x^2}$, where A,B,C are:

$$\begin{aligned} A &= H\tilde{g}_0 - \mu^2 h_x \eta_x H \tilde{g}_0, \\ B &= -\mu^2 H^2 H_x \tilde{g}_0 - \mu^2 h_{b,x} H^2(\tilde{g}_0 - \tilde{s}_0) + \mu^2 H^2 h_{b,x} \tilde{s}_0, \\ C &= \frac{\mu^2}{2} H^3(\tilde{g}_0 - \tilde{\nu}_0). \end{aligned} \tag{4.11}$$

Rhs_{η}, Rhs_{u_0}, Rhs_1 and Rhs_2 are given in (1.2)(1.3)(1.6) and include non-linear products of derivatives of u_0, u_1, u_2, s_0 and η. $\tilde{g}_0$, $\tilde{s}_0$, $\tilde{\nu}_0$, g_1, g_2 are constants that depend on the type of function $f(q)$ used in (2.11) and g is the non-dimensional gravitational constant. See the appendix for the complete description of these terms. Note that (4.10b) is similar to the dispersive equation in the I-GN equations.

The weak formulation of the R-GN equations (4.10) is then to find:

$$
\begin{aligned}
\varphi^h &\in V_h^K, \\
s_0^h &\in V_h^K, \\
s_1^h &\in V_h^K, \\
s_2^h &\in V_h^K, \\
r^h &\in V_h^K, \\
p^h &\in V_h^K,
\end{aligned}
\tag{4.12}
$$

where r^h, p^h approximate $s_{0,x}$ and $s_{0,xx}$ respectively, such that,

$$\mathbb{B}_\sigma\left(\varphi^h, \chi\right) = \mathbb{L}_1\left(\chi\right), \tag{4.13a}$$

$$\mathbb{B}_s\left(s_0^h, \psi\right) + \mathbb{B}_r\left(r^h, \psi\right) + \mathbb{B}_p\left(-p^h, \psi\right) = \mathbb{L}_2\left(\psi\right), \tag{4.13b}$$

$$\mathbb{B}_\sigma\left(s_1^h, \phi\right) = \mathbb{L}_3\left(\phi\right), \tag{4.13c}$$

$$\mathbb{B}_\sigma\left(s_2^h, \omega\right) = \mathbb{L}_4\left(\omega\right), \tag{4.13d}$$

where $\mathbb{B}_\sigma$ is defined in (4.7). $\mathbb{B}_s$, $\mathbb{B}_r$ and $\mathbb{B}_p$ are given by:

$$
\begin{aligned}
\mathbb{B}_s\left(s_0^h, w\right) &= \sum_j \left(A s_0^h, w\right)_{E_j}, \\
\mathbb{B}_r\left(r^h, w\right) &= \sum_j \left(B r^h, w\right)_{E_j}, \\
\mathbb{B}_p\left(p^h, w\right) &= \sum_j \left(C p^h, w\right)_{E_j},
\end{aligned}
\tag{4.14}
$$

where A, B and C are defined in (4.11). To eliminate r^h and p^h we define the following equations (Arnold et al., 2002):

$$
\begin{aligned}
\sum_j \left(r^h, w\right)_{E_j} &= \sum_j \left(-s_0^h, w_x\right)_{E_j} + \left\langle \hat{s}_0^h, [|w|]\right\rangle_{\mathcal{E}}, \\
\sum_j \left(p^h, w\right)_{E_j} &= \sum_j \left(-r^h, w_x\right)_{E_j} + \left\langle \hat{r}^h, [|w|]\right\rangle_{\mathcal{E}} - \sigma_{11}\left\langle \left[|s_0^h|\right], [|w|]\right\rangle_{\mathcal{E}}.
\end{aligned}
\tag{4.15}
$$

Here σ_{11} is a penalty term and w, χ, ψ, ϕ and $\omega \in V_h^K$. The *linear* forms are given by:

$$\begin{aligned}
\mathbb{L}_1(\chi) &= \sum_j \left(Rhs_\eta^h, \chi\right)_{E_j}, \\
\mathbb{L}_2(\psi) &= \sum_j \left(Rhs_{u_0}^h, \psi\right)_{E_j}, \\
\mathbb{L}_3(\phi) &= \sum_j \left(Rhs_1^h, \phi\right)_{E_j}, \\
\mathbb{L}_4(\omega) &= \sum_j \left(Rhs_2^h, \omega\right)_{E_j}.
\end{aligned} \tag{4.16}$$

The constant σ_{11} is chosen so that linear stability is satisfied. The time stepping algorithm then follows:

$\rightarrow$ Given η^h, u_0^h, u_1^h and u_2^h at t^n

- $\hookrightarrow$ Compute all the spatial derivatives from (4.6).
- $\hookrightarrow$ Determine A, B and C from (4.11), and Rhs_η, Rhs_{u_0}, Rhs_1 and Rhs_2.
- $\hookrightarrow$ Compute $\varphi^h = \eta_t^h$ from (4.13a).
- $\hookrightarrow$ Compute r^h, p^h in terms of s_0^h from (4.15). Then perform the elliptic solve for $s_0^h = u_{0,t}^h$ from (4.13b) and update Rhs_1 and Rhs_2. This will involve the solution of a linear equation.
- $\hookrightarrow$ Compute $s_1^h = u_{1,t}^h$ and $s_2^h = u_{2,t}^h$ from (4.13c) and (4.13d) respectively.

$\rightarrow$ Update η^h, u_0^h, u_1^h and u_2^h from φ^h, s_0^h, s_1^h and s_2^h respectively.

where each update is performed using a fourth order classical Runge-Kutta method. A similar strategy can be followed to solve the dispersive part of the $I - GN$ equations (Bonneton et al., 2011).

4.2.2 Boundary conditions

The boundary conditions in DG methods are generally imposed weakly. The most common boundary conditions that occur when we solve Green-Naghdi equations are wall boundary condition, transmissive boundary condition and periodic boundary conditions.

- Wall: For wall boundary conditions we take $\mathbf{u}^{ext.} = -\mathbf{u}^{int.}$ and $H^{ext.} = H^{int.}$
- Transmissive: We take $\mathbf{u}^{ext.} = \mathbf{u}^{int.}$ and $H^{ext.} = H^{int.}$
- Periodic: The domain can be thought to be *wrapped around* and the exterior edge at L corresponds to the interior edge at 0 of the domain.

here $ext\cdot$ and $int\cdot$ refers to *exterior* and *interior* respectively.

4.2.3 Implementation Details

A typical mesh in $1D$ is shown in the figure (4.1).

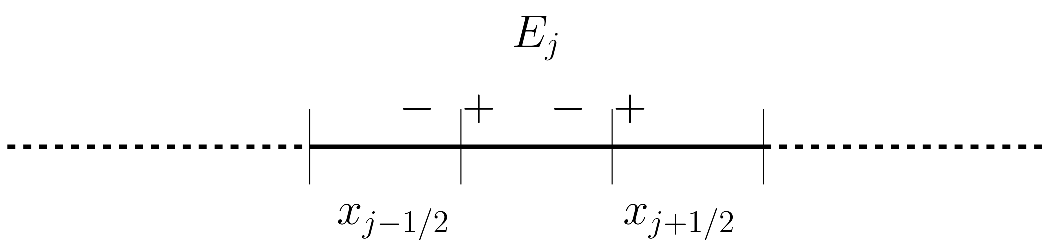

Figure 4.1: $1D$ mesh

In each element E_j, the LDG solution lives in the space of polynomials of degree K. In order to get the initial conditions of a variable u we compute its L_2 projection in each element. Note that the L_2 projection of u is given by

$$\int_{E_j} (u|_{E_j} - \Pi u)v = 0, \tag{4.17}$$

for all $v \in \mathbb{P}_K(E_j)$. Since $u^h|_{E_j} = \Pi u \in \mathbb{P}_K(E_j)$ is our LDG variable restricted to the element E_j, it is given by $u^h|_{E_j} = \sum_{i=0}^{K} \tilde{u}_i^j \phi_i(x)$ where $\tilde{u}_i^j$ are called the *modes* of u^h. For example if $K = 1$ then u^h will have 2 modes. The modes are *hierarchical* in the sense that the first mode represents the *constant* part and the second mode represents the *slope* of the solution. Note that in finite volume we only solve for one mode.

If we choose orthogonal basis functions ϕ_i then it is easy to determine the modes in an element. Below we describe the algorithm to calculate the modes of a variable given an initial fuction.

procedure GETMODES(Ne) ▷ finds the modes
  for $j \leftarrow 1, Ne$ **do** ▷ Loop through elements
    for $i \leftarrow 0, dof$ **do** ▷ Loop through degrees of freedom

$\tilde{u}_i^j = \frac{(u,\phi_i)_{E_j}}{(\phi_i,\phi_i)_{E_j}}$

end for

end for

end procedure

Here $(v, w)_{E_j}$ is the standard L_2 inner product and is equal to $\int_{E_j} vw$. In $1D$ the elemental degrees of freedom are just the number of basis functions and is equal to $K + 1$. Thus our solution variable is of length $Ne \times (K + 1)$. Given any variable u_h, we can *locally* calculate its derivatives as described in (4.6). We detail this procedure in the following paragraph.

Let λ^h be the approximation to u_x. Then

$$(\lambda^h, w)_{E_j} = -\sum_j (u^h, w_x)_{E_j} + \hat{u}^h|_{x_{j+1/2}} w|_{x_{j+1/2}} - \hat{u}^h|_{x_{j-1/2}} w|_{x_{j-1/2}}. \tag{4.18}$$

Here the numerical trace $\hat{u}^h$ at any edge is taken to be the average of the elemental values sharing that edge. Thus, the above equation becomes:

$$\begin{aligned}(\lambda^h, w)_{E_j} = &-\sum_j (u^h, w_x)_{E_j} + 0.5 * u^h|^j_{x_{j+1/2}} w|^j_{x_{j+1/2}} + 0.5 * u^h|^{j+1}_{x_{(j+1)-1/2}} w|^j_{x_{j+1/2}} \\ &- 0.5 * u^h|^j_{x_{j-1/2}} w|^j_{x_{j-1/2}} - 0.5 * u^h|^{j-1}_{x_{(j-1)+1/2}} w|^j_{x_{j-1/2}}.\end{aligned} \tag{4.19}$$

Now we can find the modes of $\lambda^h|_{E_j}$,

$$\mathbb{M}^{(j)} \{\tilde{\lambda}^j\} = \mathbb{K}^{(j)} \{\tilde{u}^j\} + \mathbb{F}^{(j,j)} \{\tilde{u}^j\} + \mathbb{F}^{(j+1,j)} \{\tilde{u}^{j+1}\} + \mathbb{F}^{(j-1,j)} \{\tilde{u}^{j-1}\} \tag{4.20}$$

where the local matrices are defined as follows:

$$\begin{aligned}
&\mathbb{M}^{(j)}\left[l,m\right] = \left(\phi_l, \phi_m\right)_{E_j}\\
&\mathbb{K}^{(j)}\left[l,m\right] = -\left(\phi_{l,x}, \phi_m\right)_{E_j}\\
&\mathbb{F}^{(j,j)}\left[l,m\right] = 0.5 * \phi_l|_{x_{j+1/2}}\phi_m|_{x_{j+1/2}} - 0.5 * \phi_l|_{x_{j-1/2}}\phi_m|_{x_{j-1/2}}\\
&\mathbb{F}^{(j+1,j)}\left[l,m\right] = 0.5 * \phi_l|_{x_{j+1/2}}\phi_m|_{x_{(j+1)-1/2}}\\
&\mathbb{F}^{(j-1,j)}\left[l,m\right] = 0.5 * \phi_l|_{x_{j-1/2}}\phi_m|_{x_{(j-1)+1/2}}
\end{aligned} \tag{4.21}$$

Using this procedure, we compute higher derivatives of u^h. For example, if ω^h represents the approximation to u_{xx}, then we can find the modes of $\omega^h|_{E_j}$ given by

$$\mathbb{M}^{(j)}\left\{\tilde{\omega}^j\right\} = \mathbb{K}^{(j)}\left\{\tilde{\lambda}^j\right\} + \mathbb{F}^{(j,j)}\left\{\tilde{\lambda}^j\right\} + \mathbb{F}^{(j+1,j)}\left\{\tilde{\lambda}^{j+1}\right\} + \mathbb{F}^{(j-1,j)}\left\{\tilde{\lambda}^{j-1}\right\} \tag{4.22}$$

where we can use (4.20) to get $\omega^h|_{E_j}$ in terms of $u^h|_{E_j}$.

The above equation needs to be modified for boundary conditions. The most common boundary condition in $Green - Naghdi$ equations is the wall boundary condition. Here the following are specified,

$$\begin{aligned}
&u_0 = u_1 = u_2 = 0,\\
&u_{0,xx} = u_{1,xx} = u_{2,xx} = 0,\\
&\eta_x = 0.
\end{aligned} \tag{4.23}$$

In general, all the *odd* derivatives of η are zero while all the *even* derivatives of u_0, u_1 and u_2 are zero. Thus, when we calculate the approximation of $u_{0,x}$ or η_x we need to account for the boundary conditions. In DG methods, this is done through the weak form and is easy to implement. Here, we will show

how the the wall boundary conditions is applied at $x=0$ for the numerical approximation of $u_{0,x}, u_{0,xx}$ and η_x. For other equations the strategy will remain the same.

Let us consider the approximation of $u_{0,x}$. The equation as described before is,

$$(\lambda^h, w)_{E_j} = -\sum_j (u_0^h, w_x)_{E_j} + \hat{u}_0^h|_{x_{1+1/2}} w|_{x_{1+1/2}} - \hat{u}_0^h|_{x_{1/2}} w|_{x_{1/2}}. \tag{4.24}$$

Since we have a wall boundary condition at $x_{1/2}$ we impose $\hat{u}_0^h|_{x_{1/2}} = 0$. Thus the matrix equation for the modes become,

$$\mathbb{M}^{(j)} \{\tilde{\lambda}^j\} = \mathbb{K}^{(j)} \{\tilde{u}_0^j\} + {\mathbb{F}_D}^{(j,j)} \{\tilde{u}_0^j\} + {\mathbb{F}_D}^{(j+1,j)} \{\tilde{u}_0^{j+1}\} \tag{4.25}$$

where ${\mathbb{F}_D}^{(j,j)}$ and ${\mathbb{F}_D}^{(j+1,j)}$ have been modified for the boundary and are given by:

$$\begin{aligned} {\mathbb{F}_D}^{(j,j)} [l,m] &= 0.5 * \phi_l|_{x_{3/2}} \phi_m|_{x_{3/2}} \\ {\mathbb{F}_D}^{(j+1,j)} [l,m] &= 0.5 * \phi_l|_{x_{3/2}} \phi_m|_{x_{(2)-1/2}} \end{aligned} \tag{4.26}$$

In a similar fashion let us consider the approximation of $u_{0,xx}$ given by:

$$(\omega^h, w)_{E_j} = -\sum_j (\lambda^h, w_x)_{E_j} + \hat{\lambda}^h|_{x_{3/2}} w|_{x_{3/2}} - \hat{\lambda}^h|_{x_{1/2}} w|_{x_{1/2}}. \tag{4.27}$$

At $x_{1/2}$, we have the wall boundary condition where $u_{0,x}$ is not specified however $u_{0,xx}$ is specified to be zero. Hence, to include this we consider a *Ghost* cell to the left of $x_{1/2}$ and specifiy a *Neumann boundary* condition on λ^h. Thus

$$\hat{\lambda}^h|_{x_{1/2}} = 0.5 * \left((\lambda^h)^{(-)} + (\lambda^h)^{(+)} \right) \tag{4.28}$$

where the Neumann condition means that

$$(\lambda^h)^{(-)} = (\lambda^h)^{(+)}. \tag{4.29}$$

Thus the matrix equation for ω^h is given by:

$$\mathbb{M}^{(j)} \left\{\tilde{\omega}^j\right\} = \mathbb{K}^{(j)} \left\{\tilde{\lambda}^j\right\} + \mathbb{F}_N{}^{(j,j)} \left\{\tilde{\lambda}^j\right\} + \mathbb{F}_N{}^{(j+1,j)} \left\{\tilde{\lambda}^{j+1}\right\} \tag{4.30}$$

where $\mathbb{F}_N{}^{(j)}$ and $\mathbb{F}_N{}^{(j+1,j)}$ have been modified for the boundary and are given by:

$$\begin{aligned}\mathbb{F}_N{}^{(j,j)}\left[l,m\right] &= 0.5 * \phi_l|_{x_{3/2}}\phi_m|_{x_{3/2}} - \phi_l|_{x_{1/2}}\phi_m|_{x_{1/2}} \\ \mathbb{F}_N{}^{(j+1,j)}\left[l,m\right] &= 0.5 * \phi_l|_{x_{3/2}}\phi_m|_{x_{(2)-1/2}}\end{aligned} \tag{4.31}$$

Similarly, in approximating η_x, we'll use equations as above to implement Neumann Boundary conditions.

Now let us describe the bi-linear form described in (4.7) given by:

$$\mathbb{B}_\sigma\left(u^h, w\right) = \sum_j \left(u^h, w\right)_{E_j} + \sigma \left\langle \left[|u^h|\right], [|w|] \right\rangle_{\mathcal{E}},$$

For an element E_j the above equation can be written as:

$$\mathbb{B}_\sigma\big(u^h, w\big)_{E_j} = \big(u^h, w\big)_{E_j} + \sigma \left[|u^h|\right]|_{x_{j+1/2}} w|^j_{x_{j+1/2}} - \sigma \left[|u^h|\right]|_{x_{j-1/2}} w|^j_{x_{j-1/2}}. \tag{4.32}$$

Using the definition of jump in (4.4) we obtain

$$\begin{aligned}\mathbb{B}_\sigma\big(u^h, w\big)_{E_j} =& \big(u^h, w\big)_{E_j} + \sigma u^h|^j_{x_{j+1/2}} w|^j_{x_{j+1/2}} - \sigma u^h|^{j+1}_{x_{(j+1)-1/2}} w|^j_{x_{j+1/2}} \\ &- \sigma u^h|^{j-1}_{x_{(j-1)+1/2}} w|^j_{x_{j-1/2}} + \sigma u^h|^j_{x_{j-1/2}} w|^j_{x_{j-1/2}}.\end{aligned} \tag{4.33}$$

Thus we can write the matrix form as follows:

$$\mathbb{B}^{(j)}_\sigma = \mathbb{M}^{(j)} \left\{\tilde{u}^j\right\} + \mathbb{F}^{(j,j)}_\sigma \left\{\tilde{u}^j\right\} + \mathbb{F}^{(j+1,j)}_\sigma \left\{\tilde{u}^{j+1}\right\} + \mathbb{F}^{(j-1,j)}_\sigma \left\{\tilde{u}^{j-1}\right\} \tag{4.34}$$

where the local matrices are defined below:

$$
\begin{aligned}
\mathbb{F}_{\sigma}^{(j,j)}[l,m] &= \sigma * \phi_l|_{x_{j+1/2}}\phi_m|_{x_{j+1/2}} + \sigma * \phi_l|_{x_{j-1/2}}\phi_m|_{x_{j-1/2}} \\
\mathbb{F}_{\sigma}^{(j+1,j)}[l,m] &= -\sigma * \phi_l|_{x_{j+1/2}}\phi_m|_{x_{(j+1)-1/2}} \\
\mathbb{F}_{\sigma}^{(j-1)}[l,m] &= -\sigma * \phi_l|_{x_{j-1/2}}\phi_m|_{x_{(j-1)+1/2}}
\end{aligned}
\tag{4.35}
$$

Since this bi-linear form is used the time derivatives of the solution variable which are unknown at the given time, we need to assemble this matrix. The global matrix $\mathbb{B}_\sigma$ will be block tri-diagonal and is shown below.

$$
\mathbb{B}_\sigma = \begin{bmatrix}
\mathbb{M}^{(1)} + \mathbb{F}^{(1,1)} & \mathbb{F}^{(2,1)} & \cdots & \cdots \\
\mathbb{F}^{(1,2)} & \mathbb{M}^{(2)} + \mathbb{F}^{(2)} & \mathbb{F}^{(3,2)} & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
\cdots & \cdots & \mathbb{F}^{(Ne-1,Ne)} & \mathbb{M}^{(Ne)} + \mathbb{F}^{(Ne,Ne)}
\end{bmatrix}
\tag{4.36}
$$

Now, we will describe in detail the matrix equation in solving the equation (4.10b) whose discrete form is described in (4.13b). Note that it is an elliptic equation. The bi-linear forms are detailed in (4.14) and (4.15). For completeness let us write the bi-linear form (4.15).

$$
\begin{aligned}
\sum_j \left(r^h, w\right)_{E_j} &= \sum_j \left(-s_0^h, w_x\right)_{E_j} + \left\langle \hat{s}_0^h, [|w|]\right\rangle_{\mathcal{E}}, \\
\sum_j \left(p^h, w\right)_{E_j} &= \sum_j \left(-r^h, w_x\right)_{E_j} + \left\langle \hat{r}^h, [|w|]\right\rangle_{\mathcal{E}} - \sigma_{11}\left\langle \left[|s_0^h|\right], [|w|]\right\rangle_{\mathcal{E}}.
\end{aligned}
\tag{4.37}
$$

r^h and p^h approximate $s_{0,x} = u_{0,xt}$ and $s_{0,xx} = u_{0,xxt}$ respectively. For an element E_j the above equation becomes:

$$
\left(r^h, w\right)_{E_j} = \left(-s_0^h, w_x\right)_{E_j} + \hat{s}_0^h|_{x_{j+1/2}} w|^j_{x_{j+1/2}} - \hat{s}_0^h|_{x_{j-1/2}} w|^j_{x_{j-1/2}}
\tag{4.38}
$$

$$
\begin{aligned}
\left(p^h, w\right)_{E_j} = \left(-r^h, w_x\right)_{E_j} &+ \hat{r}^h|_{x_{j+1/2}} w|^j_{x_{j+1/2}} - \hat{r}^h|_{x_{j-1/2}} w|^j_{x_{j-1/2}} \\
&- \sigma_{11}\left[|s_0^h|\right]|_{x_{j+1/2}} w|^j_{x_{j+1/2}} + \sigma_{11}\left[|s_0^h|\right]|_{x_{j-1/2}} w|^j_{x_{j-1/2}}.
\end{aligned}
\tag{4.39}
$$

Taking average numerical trace and using the definition of jump in (4.4) the equation above for r^h can be written as:

$$\begin{aligned}\left(r^h, w\right)_{E_j} =& (-s_0^h, w_x)_{E_j} + 0.5 * s_0^h|^j_{x_{j+1/2}} w|^j_{x_{j+1/2}} + 0.5 * s_0^h|^{j+1}_{x_{(j+1)-1/2}} w|^j_{x_{j+1/2}} \\ & - 0.5 * s_0^h|^j_{x_{j-1/2}} w|^j_{x_{j-1/2}} - 0.5 * s_0^h|^{j-1}_{x_{(j-1)+1/2}} w|^j_{x_{j-1/2}}\end{aligned} \tag{4.40}$$

and so the matrix equation relating r^h to s_0^h for an element is

$$\mathbb{M}^{(j)} \left\{\tilde{r}^j\right\} = \mathbb{K}^{(j)} \left\{\tilde{s}_0^j\right\} + \mathbb{F}^{(j,j)} \left\{\tilde{s}_0^j\right\} + \mathbb{F}^{(j+1,j)} \left\{\tilde{s}_0^{j+1}\right\} + \mathbb{F}^{(j-1,j)} \left\{\tilde{s}_0^{j-1}\right\} \tag{4.41}$$

from which we can eliminate $\{\tilde{r}^j\}$ to get

$$\left\{\tilde{r}^j\right\} = \mathbb{M}^{(j)^{-1}} \left(\mathbb{K}^{(j)} + \mathbb{F}^{(j,j)}\right) \left\{\tilde{s}_0^j\right\} + \mathbb{M}^{(j)^{-1}} \mathbb{F}^{(j+1,j)} \left\{\tilde{s}_0^{j+1}\right\} + \mathbb{M}^{(j)^{-1}} \mathbb{F}^{(j-1,j)} \left\{\tilde{s}_0^{j-1}\right\} \tag{4.42}$$

Similarly, the equation for p^h can be written as

$$\begin{aligned}\left(p^h, w\right)_{E_j} =& \left(-r^h, w_x\right)_{E_j} + 0.5 * r^h|^j_{x_{j+1/2}} w|^j_{x_{j+1/2}} + 0.5 * r^h|^{j+1}_{x_{(j+1)-1/2}} w|^j_{x_{j+1/2}} \\ & - 0.5 * r^h|^j_{x_{j-1/2}} w|^j_{x_{j-1/2}} - 0.5 * r^h|^{j-1}_{x_{(j-1)+1/2}} w|^j_{x_{j-1/2}} \\ & - \sigma_{11} s_0^h|^j_{x_{j+1/2}} w|^j_{x_{j+1/2}} + \sigma_{11} s_0^h|^{j+1}_{x_{(j+1)-1/2}} w|^j_{x_{j+1/2}} \\ & + \sigma_{11} s_0^h|^{j-1}_{x_{(j-1)+1/2}} w|^j_{x_{j-1/2}} - \sigma_{11} s_0^h|^j_{x_{j-1/2}} w|^j_{x_{j-1/2}}.\end{aligned} \tag{4.43}$$

and so the matrix equation relating p^h to r^h and s^h for an element can be written as

$$\begin{aligned}\mathbb{M}^{(j)} \left\{\tilde{p}^j\right\} =& \mathbb{K}^{(j)} \left\{\tilde{r}^j\right\} + \mathbb{F}^{(j,j)} \left\{\tilde{r}^j\right\} \\ & + \mathbb{F}^{(j+1,j)} \left\{\tilde{r}^{j+1}\right\} + \mathbb{F}^{(j-1,j)} \left\{\tilde{r}^{j-1}\right\} \\ & - \mathbb{F}^{(j)}_{\sigma_{11}} \left\{\tilde{s}_0^j\right\} - \mathbb{F}^{(j+1,j)}_{\sigma_{11}} \left\{\tilde{s}_0^{j+1}\right\} \\ & - \mathbb{F}^{(j-1,j)}_{\sigma_{11}} \left\{\tilde{s}_0^{j-1}\right\}\end{aligned} \tag{4.44}$$

Thus, using (4.42) we can get $\{\tilde{p}^j\}$ entirely in terms of $\{\tilde{s}_0^j\}$. Now, equation (4.14) can be easily written only in terms of s_0^h. For completness we write the equation (4.14) below:

$$\begin{aligned} \mathbb{B}_s\left(s_0^h, w\right) &= \sum_j \left(A s_0^h, w\right)_{E_j}, \\ \mathbb{B}_r\left(r^h, w\right) &= \sum_j \left(B r^h, w\right)_{E_j}, \\ \mathbb{B}_p\left(p^h, w\right) &= \sum_j \left(C p^h, w\right)_{E_j}, \end{aligned} \tag{4.45}$$

where A, B and C are defined in (4.11). For an element E_j the above equation can be written as

$$\begin{aligned} \mathbb{B}_s\left(s_0^h, w\right)_{E_j} &= \left(A s_0^h, w\right)_{E_j}, \\ \mathbb{B}_r\left(r^h, w\right)_{E_j} &= \left(B r^h, w\right)_{E_j}, \\ \mathbb{B}_p\left(p^h, w\right)_{E_j} &= \left(C p^h, w\right)_{E_j}, \end{aligned} \tag{4.46}$$

The matrix equation for the above then is simply

$$\begin{aligned} \mathbb{B}_s\left(s_0^h, w\right)_{E_j} &= \mathbb{A}_s^{(j)}\ \{\tilde{s}_0^j\} \\ \mathbb{B}_r\left(r^h, w\right)_{E_j} &= \mathbb{B}_r^{(j)}\ \{\tilde{r}^j\} \\ \mathbb{B}_p\left(r^h, w\right)_{E_j} &= \mathbb{C}_p^{(j)}\ \{\tilde{p}^j\} \end{aligned} \tag{4.47}$$

where the local matrices are given by:

$$\begin{aligned} \mathbb{A}_s^{(j)}[l,m] &= (A\phi_l, \phi_m)_{E_j} \\ \mathbb{B}_r^{(j)}[l,m] &= (B\phi_l, \phi_m)_{E_j} \\ \mathbb{C}_p^{(j)}[l,m] &= (C\phi_l, \phi_m)_{E_j} \end{aligned} \tag{4.48}$$

where A, B and C are defined in (4.11). Using (4.42) and (4.44) we can solve for s_0^h. To see this let us define,

$$\bar{\mathbb{B}}^{(m,j)} = \mathbb{B}_r^{(n)}\left[{\mathbb{M}^{(j)}}^{-1}\left(\delta_{mj}\mathbb{K}^{(j)} + \mathbb{F}^{(m,j)}\right)\right] \tag{4.49}$$

and so $\mathbb{B}_r\left(r^h, w\right)_{E_j}$ can be written entirely in terms of s_0^h as given by

$$\mathbb{B}_r\left(r^h, w\right)_{E_j} = \bar{\mathbb{B}}^{(j,j)}\left\{\tilde{s}_0^j\right\} + \bar{\mathbb{B}}^{(j+1,j)}\left\{\tilde{s}_0^{j+1}\right\} + \bar{\mathbb{B}}^{(j-1,j)}\left\{\tilde{s}_0^{j-1}\right\} \tag{4.50}$$

Similarly let us define,

$$\begin{aligned}\bar{\mathbb{C}}^{(m,j)} =& \mathbb{C}_p^{(j)}\left[\mathbb{M}^{(j)^{-1}}\left(\mathbb{K}^{(j)} + \mathbb{F}^{(j,j)}\right)\right]\bar{\mathbb{B}}^{(m,j)} + \mathbb{C}_p^{(j)}\left[\mathbb{M}^{(j)^{-1}}\left(\mathbb{F}^{(j+1,j)}\right)\right]\bar{\mathbb{B}}^{(m,j+1)}\\ &+ \mathbb{C}_p^{(j)}\left[\mathbb{M}^{(j)^{-1}}\left(\mathbb{F}^{(j-1,j)}\right)\right]\bar{\mathbb{B}}^{(m,j-1)}\end{aligned} \tag{4.51}$$

and hence $\mathbb{B}_p\left(r^h, w\right)_{E_j}$ can be completely written in terms of as follows

$$\begin{aligned}\mathbb{B}_p\left(r^h, w\right)_{E_j} =& \left[\bar{\mathbb{C}}^{(j,j)} - \mathbb{M}^{(j)^{-1}}\mathbb{F}_{\sigma_{11}}^{(j,j)}\right]\left\{\tilde{s}_0^j\right\}\\ &+ \left[\bar{\mathbb{C}}^{(j+1,j)} - \mathbb{M}^{(j)^{-1}}\mathbb{F}_{\sigma_{11}}^{(j+1,j)}\right]\left\{\tilde{s}_0^{j+1}\right\}\\ &+ \left[\bar{\mathbb{C}}^{(j-1,j)} - \mathbb{M}^{(j)^{-1}}\mathbb{F}_{\sigma_{11}}^{(j-1,j)}\right]\left\{\tilde{s}_0^{j-1}\right\}\\ &+ \bar{\mathbb{C}}^{(j-2,j)}\left\{\tilde{s}_0^{j-2}\right\}\\ &+ \bar{\mathbb{C}}^{(j+2,j)}\left\{\tilde{s}_0^{j+2}\right\}\end{aligned} \tag{4.52}$$

Now we can solve the elliptic equation (4.13b) by assembling a global matrix $\mathbb{A}_{s_0}$. For any j (with suitable modification in the boundary), let us define the following:

$$\begin{aligned}\mathbb{A}^{(j-2,j)} &= -\bar{\mathbb{C}}^{(j-2,j)}\\ \mathbb{A}^{(j-1,j)} &= \bar{\mathbb{B}}^{(j-1,j)} - \left[\bar{\mathbb{C}}^{(j-1,j)} - \mathbb{M}^{(j)^{-1}}\mathbb{F}_{\sigma_{11}}^{(j-1,j)}\right]\\ \mathbb{A}^{(j,j)} &= \mathbb{A}_s^{(j)} + \bar{\mathbb{B}}^{(j,j)} - \left[\bar{\mathbb{C}}^{(j,j)} - \mathbb{M}^{(j)^{-1}}\mathbb{F}_{\sigma_{11}}^{(j,j)}\right]\\ \mathbb{A}^{(j+1,j)} &= \bar{\mathbb{B}}^{(j+1,j)} - \left[\bar{\mathbb{C}}^{(j+1,j)} - \mathbb{M}^{(j)^{-1}}\mathbb{F}_{\sigma_{11}}^{(j+1,j)}\right]\\ \mathbb{A}^{(j+2,j)} &= -\bar{\mathbb{C}}^{(j+2,j)}\end{aligned} \tag{4.53}$$

and so the global matrix is a block penta-diagonal matrix given by

$$\mathbb{A}_{s_0} = \begin{bmatrix} \mathbb{A}^{(1,1)} & \mathbb{A}^{(2,1)} & \mathbb{A}^{(3,1)} & \dots & \dots & \dots \\ \mathbb{A}^{(1,2)} & \mathbb{A}^{(2,2)} & \mathbb{A}^{(3,2)} & \mathbb{A}^{(4,2)} & \dots & \dots \\ \mathbb{A}^{(1,3)} & \mathbb{A}^{(2,3)} & \mathbb{A}^{(3,3)} & \mathbb{A}^{(4,3)} & \mathbb{A}^{(5,3)} \dots & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ . & \dots & \dots & \mathbb{A}^{(Ne-2,Ne)} & \mathbb{A}^{(Ne-1,Ne)} & \mathbb{A}^{(Ne,Ne)} \end{bmatrix} \tag{4.54}$$

We have detailed the process of obtaining all the *left* hand side solution variables. The implementation details will be complete with the definition of computing the *right* hand side in the equations (4.13) given by the linear forms in (4.16). A typical right hand side term will contain terms which look like the following:

$$(\eta + h_b) * \eta_x * u_0 * u_{0,xx} * u_{0,x} \tag{4.55}$$

Here, we need to project the bathymetry into the discrete space V_h^K. Let η_X^h and λ^h represent the approximations to η_x and $u_{0,x}$ respectively. Then using (4.20) we can get the *modes* of η_X^h and λ^h from the modes of η^h and u_0^h respectively. Similarly by letting ω^h to be the approximation of $u_{0,xx}$ we can get the modes of ω^h from the modes of u_0^h using (4.22) and (4.20). Thus the discrete linear form of the right hand side will be given by

$$\left((\eta^h + \Pi h_b) * \eta_X^h * u_0^h * \omega^h * \lambda^h, w\right)_{E_j} \tag{4.56}$$

for every $w \in \mathbb{P}^k$. Thus we can describe the algorithm for the time step update

1: **procedure** TIMEUPDATE(Ne) ▷ finds the modes
2: solving $\mathbb{B}_\sigma\left(\varphi^h, \chi\right) = \mathbb{L}_1(\chi)$ ▷ $\varphi^h = \eta_t$
3: **for** $j \leftarrow 1, Ne$ **do** ▷ Loop through elements

compute rhs vector $\mathbb{L}_1$ through the procedure detailed in (4.56)

end for

Get η_t^h by solving $[\mathbb{B}_\sigma]\left\{\eta_t^h\right\} = \mathbb{L}_1$

solving $\mathbb{B}_s\left(s_0^h, \psi\right) + \mathbb{B}_r\left(r^h, \psi\right) + \mathbb{B}_p\left(-p^h, \psi\right) = \mathbb{L}_2(\psi)$ ▷ $s_0^h = u_{0,t}$

for $j \leftarrow 1, Ne$ **do** ▷ Loop through elements

compute rhs vector $\mathbb{L}_2$ through the procedure detailed in (4.56)

Assemble the matrix $\mathbb{A}_{s_0}$ detailed in (4.53)

end for

Get $u_{0,t}^h$ by solving $[\mathbb{A}_{s_0}]\left\{s_0\right\} = \mathbb{L}_2$

solving $\mathbb{B}_\sigma(s_1, \chi) = \mathbb{L}_3(\chi)$ ▷ $s_1 = u_{1,t}^h$

solving $\mathbb{B}_\sigma(s_2, \chi) = \mathbb{L}_4(\chi)$ ▷ $s_2 = u_{2,t}^h$

for $j \leftarrow 1, Ne$ **do** ▷ Loop through elements

compute rhs vector $\mathbb{L}_3$ through the procedure detailed in (4.56)

compute rhs vector $\mathbb{L}_4$ through the procedure detailed in (4.56)

end for

Get $u_{1,t}^h$ by solving $[\mathbb{B}_\sigma]\left\{s_1\right\} = \mathbb{L}_3$

Get $u_{2,t}^h$ by solving $[\mathbb{B}_\sigma]\left\{s_2\right\} = \mathbb{L}_4$

end procedure

The basis functions are chosen to be the orthogonal Legendre Poynomials and are normalized. In the following figure (4.2), we plot the first 4 basis functions

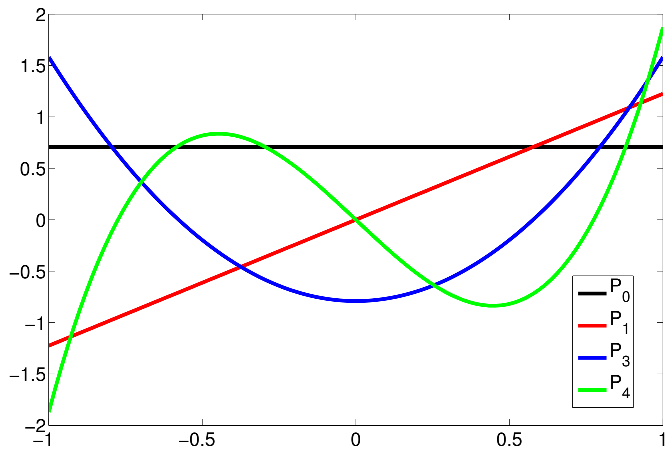

Figure 4.2: The first 4 basis functions

Since the basis functions are orthogonal in $[-1\,1]$ we have to map our element to the master element. As defined earlier the map

$$x \in E_j : x(\xi) = x_{j-1/2} + \frac{1+\xi}{2} h_j. \tag{4.57}$$

sends x to ξ and is shown in the following picture (4.3). The first two basis functions are

$$\begin{aligned} P_0 &= \sqrt{1/2} \\ P_1 &= \sqrt{3/2}\xi \end{aligned} \tag{4.58}$$

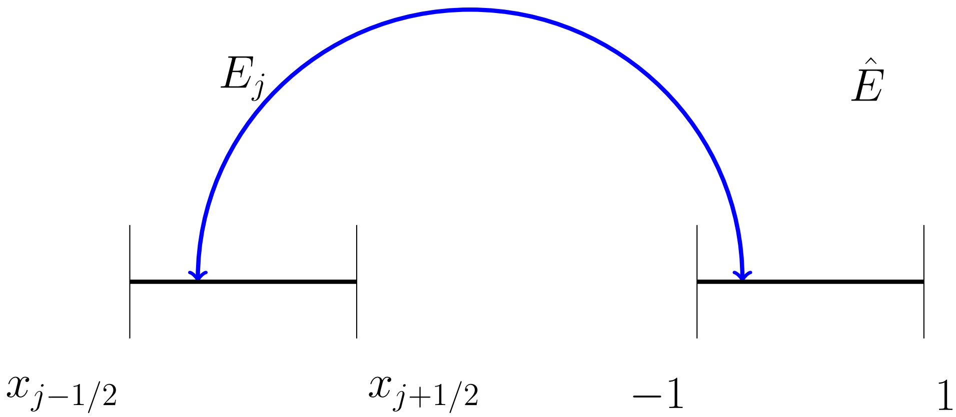

Figure 4.3: Mapping physical domain to reference domain

Using this, the standard inner product $(v(x)\,,\,w(x))_{E_j}$ becomes

$$\frac{h_j}{2}\left(v(x(\xi))\,,\,w(x(\xi))\right)_{\hat{E}}.$$

The derivative $v(x)_x$ is given by $v(x(\xi))_\xi \xi_x$.

In this subsection, we completed the implementation details of the local discontinuous Galerkin method applied to $R-GN$ equations. Before we proceed with the next section where we prove our method's linear stability and comment on the achieving non-linear stability, we close this section with a caution on using the *naive* approach of approximating higher order derivatives with average fluxes. In the following figures we take $u = 0.1\cos\left(2\pi x/10\right)$ and use ploynomial order $K = 1$ to appроimate u, u_x, u_{xx} and u_{xxx}.

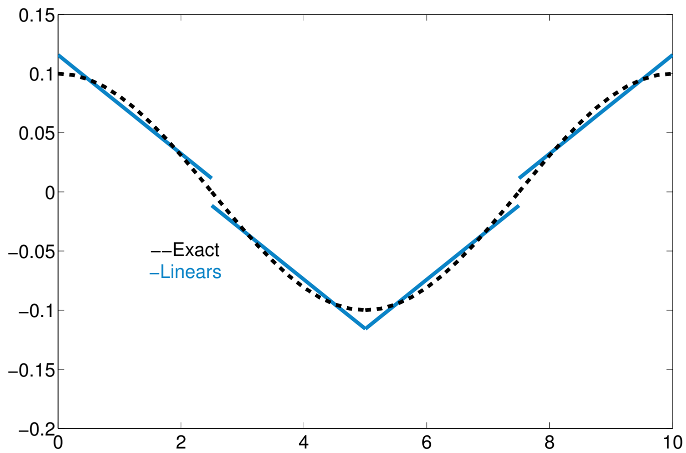

Figure 4.4: Approximation of $u = 0.1\cos(2\pi x/10)$ with $K = 1$.

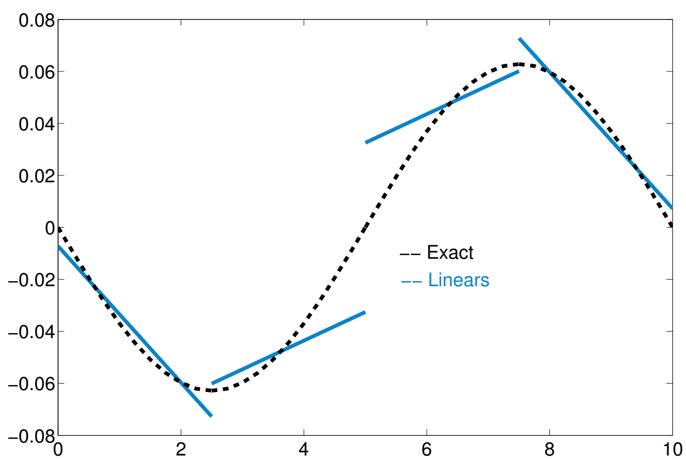

Figure 4.5: Approximation of $u_x = -0.2\pi/10\sin(2\pi x/10)$ with $K = 1$.

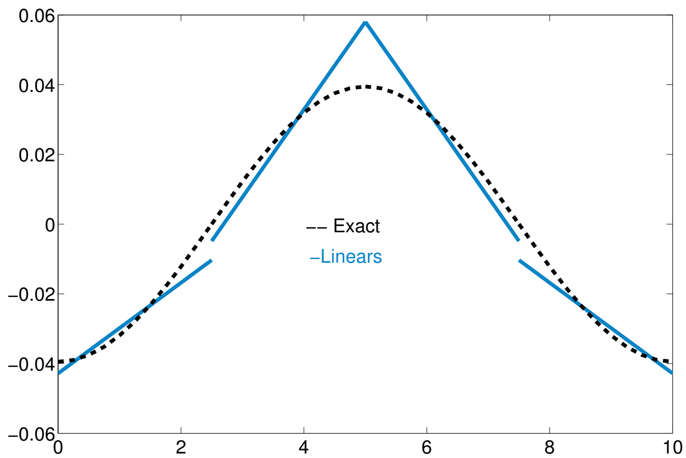

Figure 4.6: Approximation of $u_{xx} = -0.4\pi^2/10^2 \cos(2\pi x/10)$ with $K = 1$.

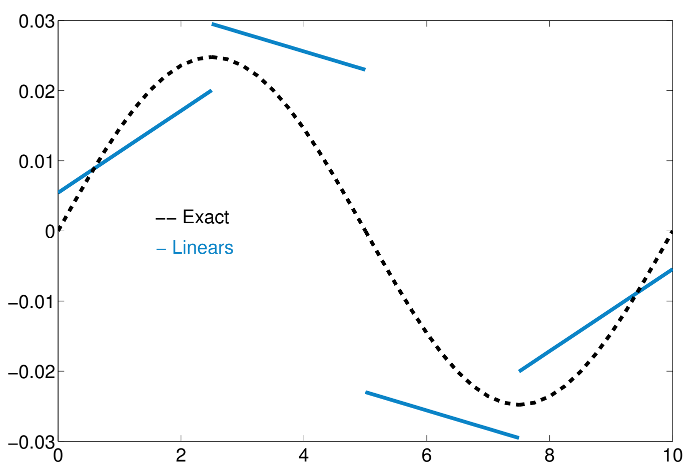

Figure 4.7: Approximation of $u_{xxx} = 0.8\pi^3/10^3 \sin(2\pi x/10)$ with $K = 1$.

The third order derivative shows large errors. Note that this is not restricted to just $K = 1$. The next plots show the approximation with $K = 3$.

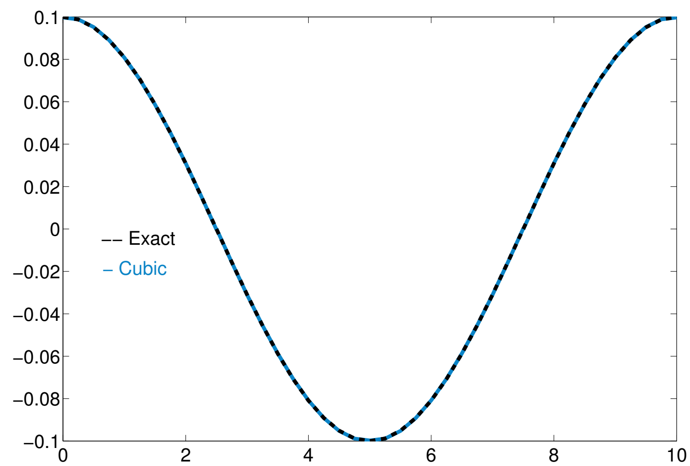

Figure 4.8: Approximation of $u = 0.1\cos(2\pi x/10)$ with $K = 3$.

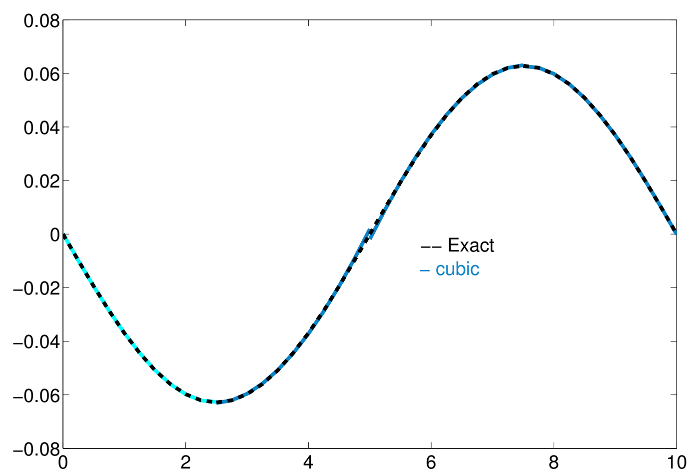

Figure 4.9: Approximation of $u_x = -0.2\pi/10\sin(2\pi x/10)$ with $K = 3$.

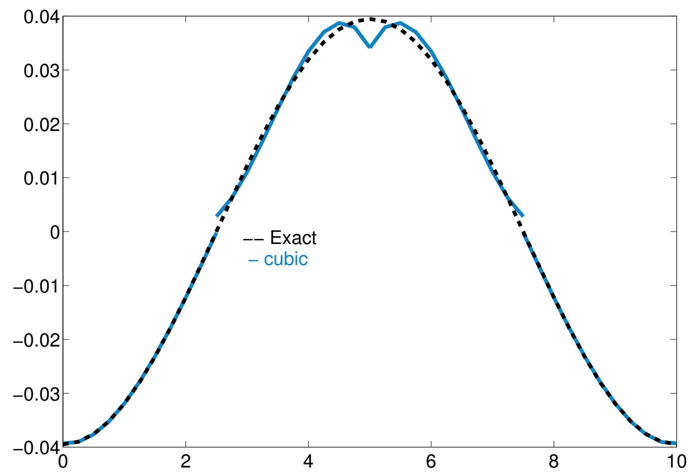

Figure 4.10: Approximation of $u_{xx} = -0.4\pi^2/10^2 \cos(2\pi x/10)$ with $K = 3$.

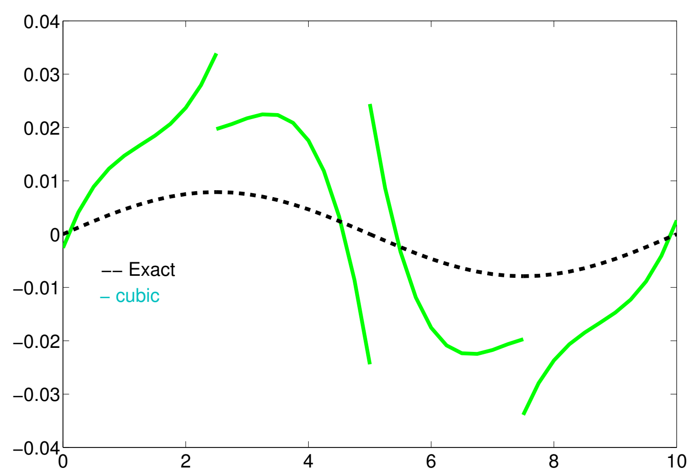

Figure 4.11: Approximation of $u_{xxx} = 0.8\pi^3/10^3 \sin(2\pi x/10)$ with $K = 3$.

Usually the nature of the PDE dictates the choice of fluxes in approximating the derivatives. However, the $R - GN$ equations are extremely non-linear and coupled hyperbolic ellliptic equation. As such it is a challenge to devise stable schemes to approximate derivatives, especially higher-order derivatives. Thus we choose the naive approach but also being mindful of the dangers in doing so. This necessitates the bilinear form (4.7).

4.3 Linear and Non-linear Stability

In this section we will perform a stability analysis of the linearized R-GN equations for a flat bathymetry h_b. For the analytic problem we'll carry out the analysis through Fourier expansion as detailed in (Engsig-Karup et al., 2006) The eigenspectra will be shown to be purely imaginary and bounded. We'll also establish the *flux* criteria for the discrete problem by considering the stability of the numerical solution using the LDG method. The linearized $O\left(\mu^2\right)$ R-GN equations can be written as:

$$\sum_{n=0}^{2} A_{mn}\left(h_b\right) u_{n,t} + \left[B_{m0}\left(h_b\right) u_{0,xt} + C_{m0}\left(h_b\right) u_{0,xxt}\right] + g_m \eta_x = 0, \qquad \forall m = 0 \ldots 2.$$

$$\eta_t + \sum_{n=0}^{2} \left(D_n\left(h_b\right) u_n\right)_{,x} = 0.$$

To keep our analysis simple we choose the shifted Legendre polynomials (Zhang et al., 2013) in (2.11) which decouples u_1 and u_2 above and hence it is sufficient only to look at the following equation:

$$\begin{aligned} &u_{0,t} - c_0 h_b^2 u_{0,xxt} + g\eta_x = 0, \\ &\eta_t + h_b u_{0,x} = 0. \end{aligned} \tag{4.59}$$

Note that by choosing the shifted Legendre polynomials in (2.11) the coefficient of $u_{0,xt}$ becomes 0.

4.3.1 Linear stability of the analytic problem through Fourier analysis

We perform a Fourier stability analysis (Engsig-Karup et al., 2006) assuming a harmonic variation in space, $\eta(x,t) = \hat{\eta}(t)e^{ikx}$, $u_0(x,t) = \hat{u}_0(t)e^{ikx}$. Inserting this into (4.59), we get:

$$\mathbf{U}_t = Q\mathbf{U},$$

where $\mathbf{U} = [\hat{u}_0, \hat{\eta}]^T$ and $Q = A^{-1}B$ where, A and B are given by:

$$A = \begin{bmatrix} 1 + c_0 h_b^2 k^2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & -igk \\ ikh_b & 0 \end{bmatrix}$$

The eigenvalues of Q can be found to be

$$\lambda(Q) = \pm i \sqrt{\frac{g/h_b}{c_0 + \frac{1}{(kh_b)^2}}}$$

To obtain a bound of the magnitude we look at $\lim_{kh_b \to \infty} |\lambda(Q)|$. This gives us $|\lambda_{\max}| = \sqrt{1/c_0}\sqrt{g/h_b}$, where $c_0 = 1/6$.

4.3.2 Linear stability of the numerical method

4.3.2.1 Linear stability of the numerical method

Let us rewrite (4.59) as a system of first order (in space) equations:

$$r - u_{0,xt} = 0, \tag{4.60a}$$

$$u_{0,t} - c_0 h_b^2 r_{,x} + g\eta_{,x} = 0, \tag{4.60b}$$

$$\eta_{,t} + h_b u_{0,x} = 0. \tag{4.60c}$$

For simplicity let us assume $u(0) = u(L) = 0$. Adding (4.60c) and (4.60b) and subtracting (4.60a) after multiplication by $g\eta$, $h_b u_0$ and $c_0 h_b^3 u_{0,x}$ respectively and integrating from 0 to L we get:

$$g\left(\eta_{,t}, \eta\right) + h_b\left(u_{0,t}, u_0\right) + c_0 h_b^3\left(u_{0,xt}, u_{0,x}\right) = 0.$$

Hence, to show stability of the numerical method it is sufficient to show (Cockburn, 2003)

$$g\left(\eta_{,t}^h, \eta^h\right) + h_b\left(u_{0,t}^h, u_0^h\right) + c_0 h_b^3\left(u_{0,xt}^h, u_{0,x}^h\right) + \Theta = 0.$$

where Θ is such that integrating in time we achieve the desired stability. In the following paragraphs we will show the *discrete* time stability of the linearized equations.

For simplicity let us drop all the coefficients and let $u_0 = u$. Then, working with the discrete versions of (4.60a), (4.60b) and (4.60c) our numerical method is given by

$$\left(r^h, v\right)_\Omega = \left(u_{xt}^h, v\right)_\Omega \tag{4.61}$$

$$
\begin{aligned}
\left(u_t^h, w\right)_\Omega = & -\left(r^h, w_x\right)_\Omega + \langle \hat{r}^h, [|w|]\rangle_{\mathcal{E}} \\
& -\sigma_{11}\left\langle \left[|u_t^h|\right], [|w|]\right\rangle_{\mathcal{E}} + \left(\eta^h, w_x\right)_\Omega - \langle \hat{\eta}^h, [|w|]\rangle_{\mathcal{E}}
\end{aligned} \tag{4.62}
$$

$$
\left(\eta_t^h, p\right)_\Omega = \left(u^h, p_x\right)_\Omega - \langle \hat{u}^h, [|p|]\rangle_{\mathcal{E}} \tag{4.63}
$$

where $v, w, p \in V_h^K$. Let

$$
v = u_x^h,
$$

$$
w = u^h,
$$

$$
p = \eta^h.
$$

Thus, for an element E_j, we get,

$$
\left(r^h, u_x^h\right)_{E_j} = \left(u_{xt}^h, u_x^h\right)_{E_j} \tag{4.64}
$$

$$
\begin{aligned}
\left(u_t^h, u^h\right)_{E_j} = & -\left(r^h, u_x^h\right)_{E_j} + \hat{r}^h u^h\Big|_{x_{j-1/2}}^{x_{j+1/2}} \\
& -\sigma_{11}\left[|u_t^h|\right] u^h\Big|_{x_{j-1/2}}^{x_{j+1/2}} + \left(\eta^h, u_x^h\right)_{E_j} - \hat{\eta}^h u^h\Big|_{x_{j-1/2}}^{x_{j+1/2}}
\end{aligned} \tag{4.65}
$$

$$
\left(\eta_t^h, \eta^h\right)_{E_j} = \left(u^h, \eta_x^h\right)_{E_j} - \hat{u}^h \eta^h\Big|_{x_{j-1/2}}^{x_{j+1/2}} \tag{4.66}
$$

Hence, we get the following:

$$
\left(\eta_t^h, \eta^h\right)_{E_j} + \left(u_t^h, u^h\right)_{E_j} + \left(u_{xt}^h, u_x^h\right)_{E_j} + \Theta_{E_j} = \hat{r}^h u^h\Big|_{x_{j-1/2}}^{x_{j+1/2}} \tag{4.67}
$$

where Θ_{E_j} is given by:

$$
-\int_{E_j} d\left(\eta^h u^h\right) + \hat{\eta}^h u^h\Big|_{x_{j-1/2}}^{x_{j+1/2}} + \hat{u}^h \eta^h\Big|_{x_{j-1/2}}^{x_{j+1/2}} + \sigma_{11}\left[|u_t^h|\right] u^h\Big|_{x_{j-1/2}}^{x_{j+1/2}} \tag{4.68}
$$

Adding over the elements we get:

$$
\left(\eta_t^h, \eta^h\right) + \left(u_t^h, u^h\right) + \left(u_{xt}^h, u_x^h\right) + \Theta = \left\langle \hat{r}^h, \left[|u^h|\right]\right\rangle_{\mathcal{E}}, \tag{4.69}
$$

where $\Theta = \mathcal{I} + \mathcal{II} + \mathcal{B}.\mathcal{T}.$ We see that $\mathcal{I}$, $\mathcal{II}$ are given by:

$$
\mathcal{I} = \sum_{\mathcal{E}_i}\left(\left[|u_0^h|\right]\left(\hat{\eta}^h - \left\{\eta^h\right\}\right) + \left[|\eta^h|\right]\left(\hat{u}_0^h - \left\{u_0^h\right\}\right)\right),
$$

$$\mathcal{I}\mathcal{I} = \sigma_{11} \left\langle \left[|u_t^h| \right], \left[|u^h| \right] \right\rangle_{\mathcal{E}}.$$

The boundary terms $\mathcal{B}.\mathcal{T}$ are given by,

$$\begin{aligned} &- \left(\eta^h u_0^h\right)^- |_L + \left(\eta^h u_0^h\right)^+ |_0 \\ &+ \hat{\eta}^h \left(u_0^h\right)^- |_L - \hat{\eta}^h \left(u_0^h\right)^+ |_0 \\ &+ \hat{u}_0^h \left(\eta^h\right)^- |_L - \hat{u}_0^h \left(\eta^h\right)^+ |_0 \end{aligned}$$

Here, $\mathcal{E}_i$ represents the set of interior edges. From the above expressions it is easy to see that if we choose $\hat{u}_0^h = \left\{u_0^h\right\}$, $\hat{\eta}^h = \left\{\eta^h\right\}$ and $\hat{u}_0^h = 0$, $\hat{\eta}^h = \eta^{h^\pm}$ at the boundaries $\mathcal{I}$ and $\mathcal{B}.\mathcal{T}$ become zero. Thus to get the desired stability we have to bound $\left\langle \hat{r}^h, \left[|u^h| \right] \right\rangle_{\mathcal{E}}$. Note that if u^h were continuous in the domain then this term would be zero.

In the following paragraphs we will carry out the discrete time stability. Let us introduce some notation,

$$u_{xt}^h [n] = \frac{u_x^h [n] \; - \; u_x^h [n-1]}{\delta t} \qquad u_t^h [n] = \frac{u^h [n] \; - \; u^h [n-1]}{\delta t} \tag{4.70a}$$

where n is the current *time* level.

We can then find a lower bound for the LHS of the equation (4.69) given by the following:

$$\left(u_{xt}^h [n], u_x^h [n]\right)_\Omega = \frac{1}{2\delta t} \left[||u_x^h [n]||^2_{L^2(\Omega)} - ||u_x^h [n-1]||^2_{L^2(\Omega)} + ||u_x^h [n] - u_x^h [n-1]||^2_{L^2(\Omega)} \right]$$

$$\left(u_t^h[n], u^h[n]\right)_\Omega \geq \frac{1}{2\delta t}\left[\,||u^h[n]||^2_{L^2(\Omega)} - ||u^h[n-1]||^2_{L^2(\Omega)}\right]$$

$$\frac{\sigma_{11}}{\delta t}\left\langle [|u^h[n] - u^h[n-1]|], [|u^h[n]|]\right\rangle_{\mathcal{E}} \geq \frac{\sigma_{11}}{2\delta t}\left[\,||[|u^h[n]|]||^2_{L^2(\mathcal{E})} - ||[|u^h[n-1]|]||^2_{L^2(\mathcal{E})}\right]$$

For the RHS of the equation (4.69) after dropping the index n, we can find an upper bound given by:

$$\begin{aligned}
\langle \hat{r}^h, [|u^h|]\rangle_{\mathcal{E}} &\leq ||\hat{r}^h||_{L^2(\mathcal{E})}\,||[|u^h|]||_{L^2(\mathcal{E})} \\
&= \sigma_{11}^{-1/2}||\hat{r}^h||_{L^2(\mathcal{E})}\,\sigma_{11}^{1/2}||[|u^h|]||_{L^2(\mathcal{E})} \\
&\leq \frac{1}{2}\left(\frac{\sigma_{11}^{-1}}{\epsilon_1}||\hat{r}^h||^2_{L^2(\mathcal{E})} + \epsilon_1\sigma_{11}^{1}||[|u^h|]||^2_{L^2(\mathcal{E})}\right) \\
&\leq \epsilon\sigma_{11}^{-1}||r^h||_{L^2(\Omega)}||r^h||_{H^1(\Omega)} + \frac{1}{2}\epsilon_1\sigma_{11}||[|u^h|]||^2_{L^2(\mathcal{E})} \\
&\leq C_1\sigma_{11}^{-1}h_{\min}^{-1}||r^h||^2_{L^2(\Omega)} + \frac{1}{2}\epsilon_1\sigma_{11}||[|u^h|]||^2_{L^2(\mathcal{E})} \\
&\leq C_1\sigma_{11}^{-1}h_{\min}^{-1}||u^h_{xt}||^2_{L^2(\Omega)} + \frac{1}{2}\epsilon_1\sigma_{11}||[|u^h|]||^2_{L^2(\mathcal{E})} \\
&= C_1\sigma_{11}^{-1}h_{\min}^{-1}\left|\left|\frac{u_x^h[n] - u_x^h[n-1]}{\delta t}\right|\right|^2_{L^2(\Omega)} \\
&\quad + \frac{1}{2}\epsilon_1\sigma_{11}||[|u^h|]||^2_{L^2(\mathcal{E})}
\end{aligned}$$

Here we used the *trace inequality* (Brenner & Scott, 2008) given by:

$$||\hat{r}^h||_{L^2(\mathcal{E})} \leq C_\Omega^t ||r^h||^{1/2}_{L^2(\Omega)} ||r^h||^{1/2}_{H^1(\Omega)} \tag{4.71}$$

and the inverse inequality (Brenner & Scott, 2008)

$$||r^h||_{H^1(E_j)} \leq h_j^{-1} C^i_{E_j} ||r^h||_{L^2(E_j)} \tag{4.72}$$

The trace constant C_Ω^t is known to be finite in regular meshes and the constant from inverse inequality $C^i_{E_j}$ is independent of h_j.

Thus collecting all the terms from above, the equation (4.69) at time level n becomes:

$$\mathcal{L}_1 \leq \frac{1}{2}\epsilon_1 \sigma_{11} || [|u^h\, [n]\, |] ||^2_{L^2(\mathcal{E})}, \tag{4.73}$$

where $\mathcal{L}_1$ is given by

$$\begin{aligned}
&\mathcal{L}_1 = \\
&\quad \frac{1}{2\delta t}\left[\, ||u^h\,[n]||^{\,2}_{L^2(\Omega)} - ||u^h\,[n-1]||^{\,2}_{L^2(\Omega)}\right] \\
&\quad + \frac{\sigma_{11}}{2\delta t}\left[\, || [|u^h\,[n]\,|] ||^{\,2}_{L^2(\mathcal{E})} - || [|u^h\,[n-1]\,|] ||^{\,2}_{L^2(\mathcal{E})}\right] \\
&\quad + \frac{1}{2\delta t}\left[\, ||u^h_x\,[n]||^{\,2}_{L^2(\Omega)} - ||u^h_x\,[n-1]||^{\,2}_{L^2(\Omega)} \right. \\
&\quad \left. \left(\frac{1}{2} - \frac{C_1 \sigma_{11}^{-1} h_{\min}^{-1}}{\delta t}\right) ||u^h_x\,[n] - u^h_x\,[n-1]||^{\,2}_{L^2(\Omega)}\right]
\end{aligned}$$

The above condition imposes the restrictions on σ_{11} for linear stability i.e

$$\sigma_{11} \geq \frac{2C_1}{h_{\min}\delta t}. \tag{4.74}$$

where C_1 contains the constants from inverse inequality and the trace inequality.

Thus summing over *time* from $n = 1$ to $n = N$ and multiplying by δt throughout we get

$$
\begin{aligned}
&||u^h[N]||^2_{L^2(\Omega)} + ||u^h_x[N]||^2_{L^2(\Omega)} + 2\sigma_{11}\,||[|u^h[N]|]||^2_{L^2(\mathcal{E})} + \Theta_N \\
&\leq ||u^h[0]||^2_{L^2(\Omega)} + ||u^h_x[0]||^2_{L^2(\Omega)} + 2\sigma_{11}\,||[|u^h[0]|]||^2_{L^2(\mathcal{E})} \\
&\quad + \delta t\left(\epsilon_1\sigma_{11}\sum_{n=1}^{N}||[|u^h[n]|]||^2_{L^2(\mathcal{E})}\right),
\end{aligned}
$$

where Θ_N is given by

$$
\Theta_N = 2\delta t\left(\frac{1}{2} - \frac{C_1\sigma_{11}^{-1}h_{\min}^{-1}}{\delta t}\right)\sum_{n=1}^{N}\left[||u^h_x[n] - u^h_x[n-1]||^2_{L^2(\Omega)}\right] \tag{4.75}
$$

Thus from *discrete Gronwall's* inequality (Atkinson & Han, 2005) we get the desired stability.

4.3.3 Comments on Non-linear Stability

The stability analysis for the complete non-linear equations is quite complicated and will be considered in future work. However, similar flux choices as derived above can be used in the non-linear equations. Hence, we take the *average* fluxes to calculate derivatives in the complete non-linear equations. The rotational velocity field characteristic of the $Boussinesq - Green - Naghdi$ equations gives a coupled system of u_0, u_1, u_2 and η and hence makes it extremely challenging to construct a stable numerical scheme.

To add additional stability we add *jumps* in the time derivatives of the solution variables which is reflected in the *bi-linear* forms (4.7). To justify this, consider the equation $s_1 = Rhs_1$ where s_1 is the time derivative of u_1. The Rhs_1 terms contain non-linear products of *higher* order derivatives of u_0. If we use first order polynomials to approximate third order derivatives, Rhs_1 will be ill-resolved which in turn will inccur errors in s_1 and will cause instability as we update in time. Thus, instead of solving the weak form of $s_1 = Rhs_1$, we modify it as is given in (4.13) by choosing the bi-linear form described in (4.7). This modified weak form can be thought of as adding penalty to φ, s_1 and s_2 terms which are the time derivatives of η, u_1 and u_2 respectively. Since these variables are unknown at time of update we must solve a linear system for φ, s_1 and s_2 at every time step. Note that as we increase our polynomial order we resolve the right hand terms better but still small errors get amplified when long time integration is performed. The penalty parameter σ is chosen to be a positive number. In-order to remove *aliasing* errors that can arise out of insufficient quadrature (Kirby & Karniadakis, 2003) all our spatial integration involving polynomials are carried out exactly. However, in cases of extreme non-linearity high order polynomial approximation may still become unstable. In those cases additional stability through filtering may be needed. An excellent overview of such filters is given in (Engsig-Karup et al., 2006)(Engsig-Karup, 2006).

Note that (2.12) is a first order hyperbolic equation in η. There are many ways to tackle the spatial derivatives in such an equation. However, it was observed that a standard treatement of the derivatives as is done in the discretization of hyperbolic problems proved to be unstable. In other words, since the momentum and surface elevation equations are coupled, all spatial derivatives must be discretized in a *compatible* way. In our case we found that treating the spatial derivatives of surface elevation equation as the product of standard non-conservative terms yielded the necessary stability. The usual flux scheme like the local Lax-Friedrichs etc, which are used to handle fluxes (in conservative forms) in hyperbolic equations, did not provide the necessary stability. We must point out that in the DG scheme proposed in this paper, polynomial order $K = 0$, i.e approximating solutions using piecewise constants also resulted in an unstable solution.

Chapter 5

Verification and Validation

In this chapter, to verify our numerical method we consider a linear standing wave problem, where it is known that the mean water level defined by $\frac{1}{L}\int_0^L \eta dx = 0$ and an exact solution for flat bathymetry exists based on the linearity assumption (Dean & Dalrymple, 1991). We present h and K error convergence rates for our verification. To validate our numerical model, we compare the numerical solution of R-GN equations against experimental results obtained for the transformation of a wave train over a trapezoidal shoal. Here, we use the data reported in (Beji & Battjes, 1993) and (Dingemans, 1994). Such a test has been a standard validation scheme for the numerical models based on Boussinesq and Green-Naghdi type wave models as it tests not only linear dispersion and shoaling but also non-linear shoaling and fissioning. We also validate our numerical method against a non-linear solitary wave reflection problem, with experimental results obtained from (Power & Chwang, 1984). We use a polynomial order $K = 1$ in all our simulations.

5.1 Linear standing wave

The $R-GN$ equations as such don't have any known exact analytic solutions. However it is known that for horizontal bottoms (Dean & Dalrymple, 1991), a linear standing wave solution exists. We choose a linear standing wave given by $a/h_b = 0.02$, and impose wall boundary conditions and the following initial conditions:

$$\begin{aligned} &\eta(x,0) = a\cos\, kx, \\ &u_0(x,0), u_1(x,0), u_2(x,0) = 0. \end{aligned} \tag{5.1}$$

where a and k represent the amplitude and wave-number $(2\pi/L)$ respectively. The domain $L = 5m$ and is shown in Figure (5.1).

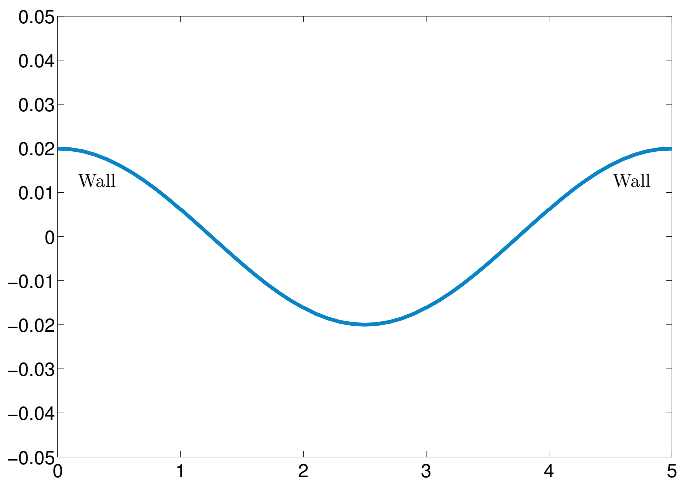

Figure 5.1: Initial domain of the standing wave problem.

The linearized Boussinesq equation for a standing wave admits an exact solution given by

$$\eta = a\cos(kx)\,\cos(\sigma t).$$

In Figure 5.3 we plot the L_2 error of the **linearized** R-GN equations such as (4.59) but with monomial shape functions for the velocity exapansion. We can immedialtely see the optimal $K+1$ convergence for odd polynomial order and suboptimal K convergence for even polynomial order whenever the penalty parameter σ_{11} is chosen to satisfy linear stability.

However, obtaining the convergence rates for the complete non-linear equations is quite cumbersome mainly because there are no known exact solutions for the non-linear R-GN equations and even constructing a manufactured solution is non-trivial. The standing wave problem is a good test of linear stability for the non-linear equations. Here we set the standing wave of amplitude $0.02m$ and plot the solution for large time-steps of the $O(10,000)$. The plot depicting the surface elevation is shown in Figure (5.2).

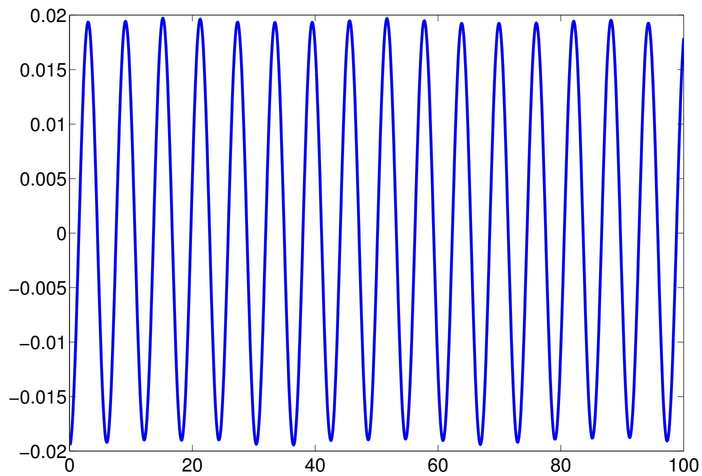

Figure 5.2: Time history of surface elevation at $x = L/2$ for a standing wave.

To study the convergence properties of the non-linear equations we use the initial conditions as used for the linearized equations but for $K = 1$ we consider the *true* solution to be as given by the simulation run on $K = 1, h = 1/8$ and similarly for $K = 2$ we consider the true solution to be as given by $K = 2, h = 1/8$. We then get the h convergence plot by running the simulation for $T = 1$ seconds on grids of $h = 1, 1/2, 1/4$. The time step δt is given by $\delta t = \frac{1}{2*K+1} * \frac{h}{C}$ where C is the linear wave speed (Dean & Dalrymple, 1991). Note that getting error convergence plots for $K \geq 3$ is very tedious due to the elliptic solve required in each time step. Moreover, the condition number increases as h is refined and K is increased and hence getting a suitable CFL criteria for time stepping becomes challenging. In Figure 5.4 we observe similar convergence rates as for the linear case.

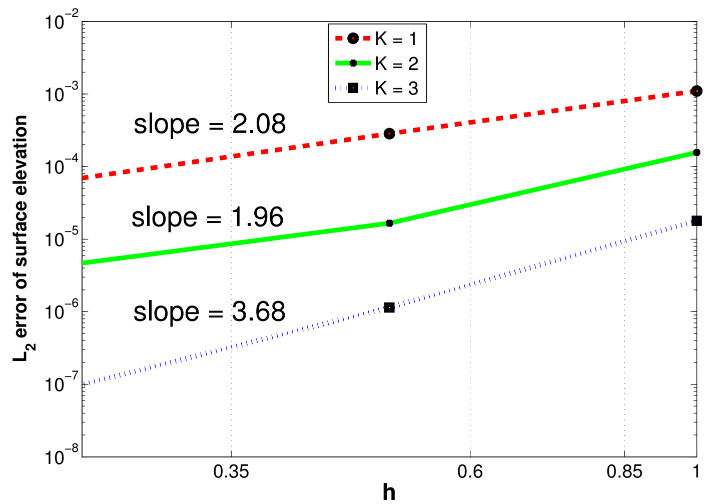

Figure 5.3: L_2 error convergence plots for the linearized equations.

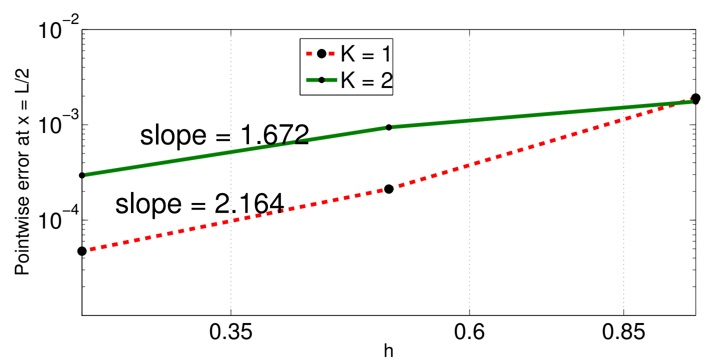

Figure 5.4: pointwise error convergence plots for the non-linearized equations.

5.2 Transformation of a wave train over a submerged shoal

In this experiment first performed in (Beji & Battjes, 1993), a wave train propagates towards a submerged trapezoidal shoal. Linear behavior is exhibited before the bar, while non-linear shoaling causes steepening as the waves interact with the slope. Complex multi-frequency waves are generated after the bar as bound harmonics are released in deeper water at the top of the bar. As the waves propagate onto the front slope of the bar, nonlinear interactions transfer energy from the leading wave component to higher harmonics, causing the wave to become steeper. After the peak of the bar is reached and the bottom slope becomes negative, the nonlinear coupling of the higher harmonics with the fundamental wave becomes progressively weaker, and, from higher to lower harmonics, each of the Fourier components are released as free waves with their own bound higher harmonics. Hence, this experiment tests both the linear dispersion (after the bar) and the non-linear characteristics of the model.

The initial wave train has a period of $T_p = 2.02$s and wave height $2a = 2$cm. The mean water depth is $h_b = 0.4$m. The initial configuration is shown in Figure 5.5. A *non-uniform* grid is used where the grid spacing decreases linearly from $h = 0.3m$ at $x = 0$ to $h = 0.1m$ at $x = 12$ and remains $0.1m$ till $x = 25m$.

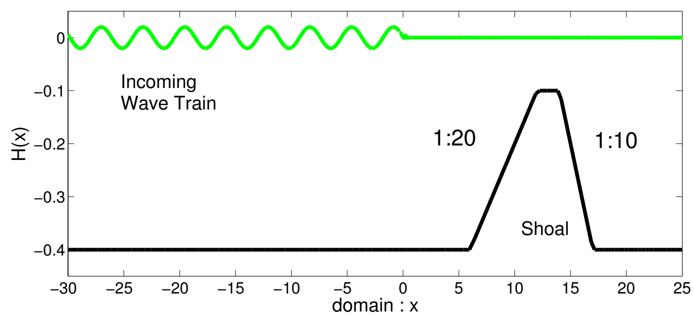

Figure 5.5: Intial configuration for validation case

Since the domain is large we employ a wave generation and absorptio zone. The generation zone is $5m$ long and generates the required wave of $Tp = 2.02$ and $H = 2cm$. The set up is shown in the figure (5.6)

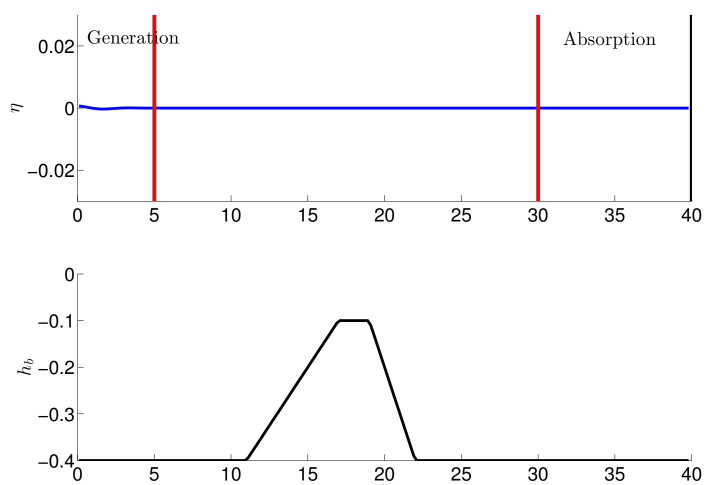

Figure 5.6: Wave generation setup for the shoaling case.

The CFL number is taken as $1/(2K + 1)$, where K is the polynomial order, and δt is calculated using the shallow water speed $c = \sqrt{gh_b}$. The numerical results are validated against the experimental test as shown in the plots in Figure 5.7 - Figure 5.12.

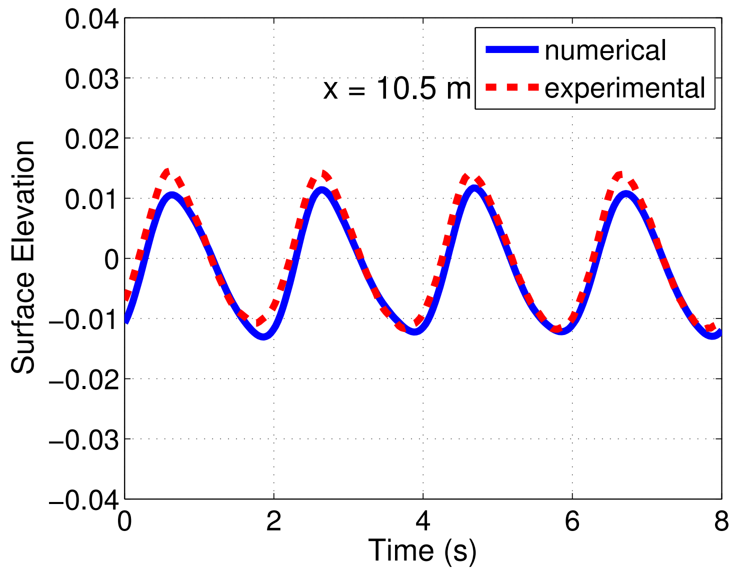

Figure 5.7: Experimental validation of surface elevation at $x = 10.5$

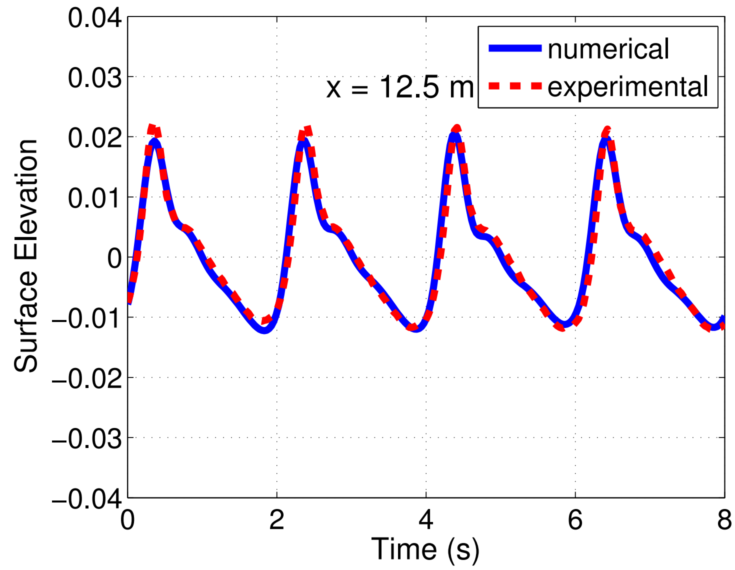

Figure 5.8: Experimental validation of surface elevation at $x = 12.5$

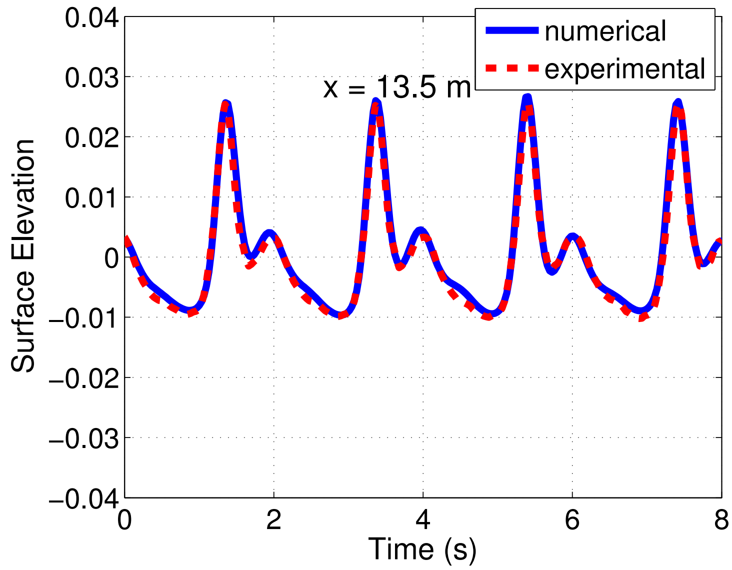

Figure 5.9: Experimental validation of surface elevation at $x = 13.5$

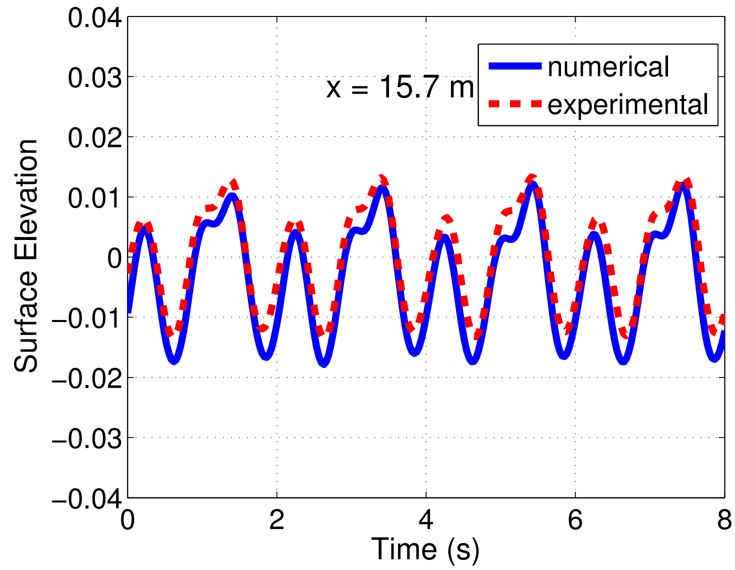

Figure 5.10: Experimental validation of surface elevation at $x = 15.7$

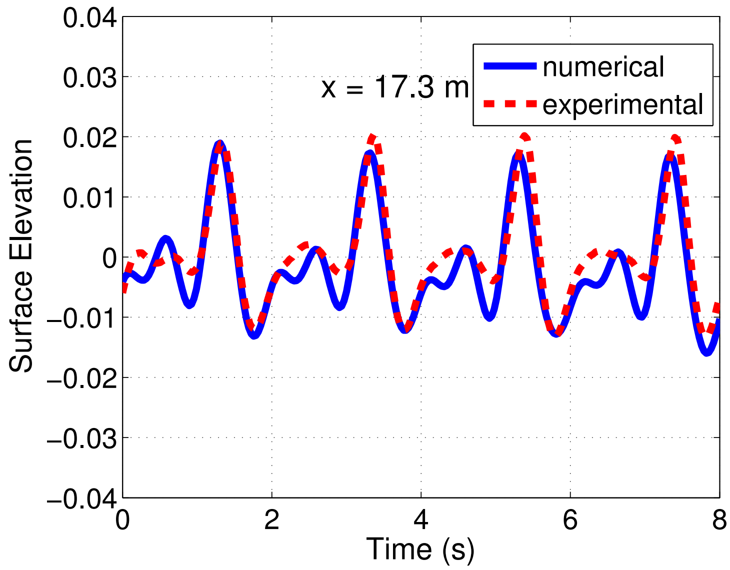

Figure 5.11: Experimental validation of surface elevation at $x = 17.3$

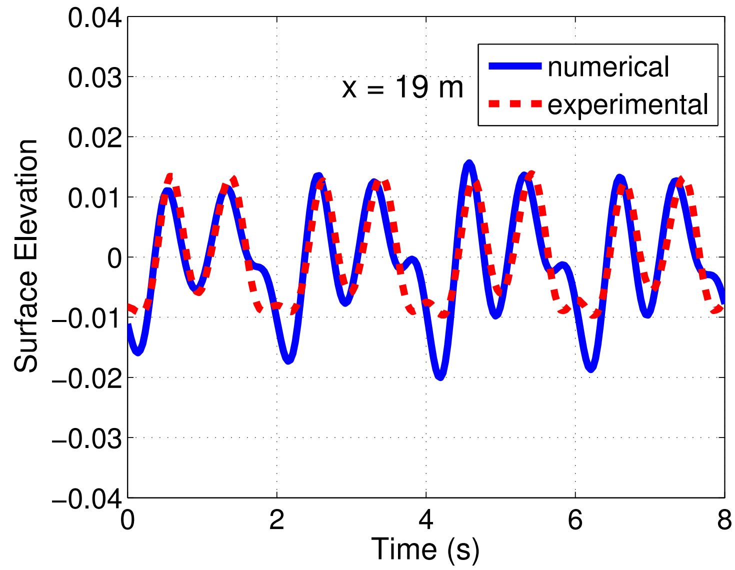

Figure 5.12: Experimental validation of surface elevation at $x = 19.0$

Figure 5.13 depicts the linear dispersion where the non-dimensional wave speed is plotted agianst the non-dimensional frequency (Gobbi & Kirby, 1999). The vertical dotted lines indicate the location of the frequency of the fundamental wave, of which the period is $T_1 = 2.02$ s, and its harmonics with periods $T_2 = T_1/2$, $T_3 = T_1/3$ and so on. As the bound waves are released as free waves, they travel with their own speed which, in the linear limit, are represented by the intersection of the vertical lines T_2, T_3, etc . with the present model's dispersion curve. As inferred from the plot, we don't expect the $O(\mu^2)$ model to give perfect agreement for the higher harmonics after the bar. This is reflected from the surface elevation plot at $x = 19.0m$ in Figure 5.7.

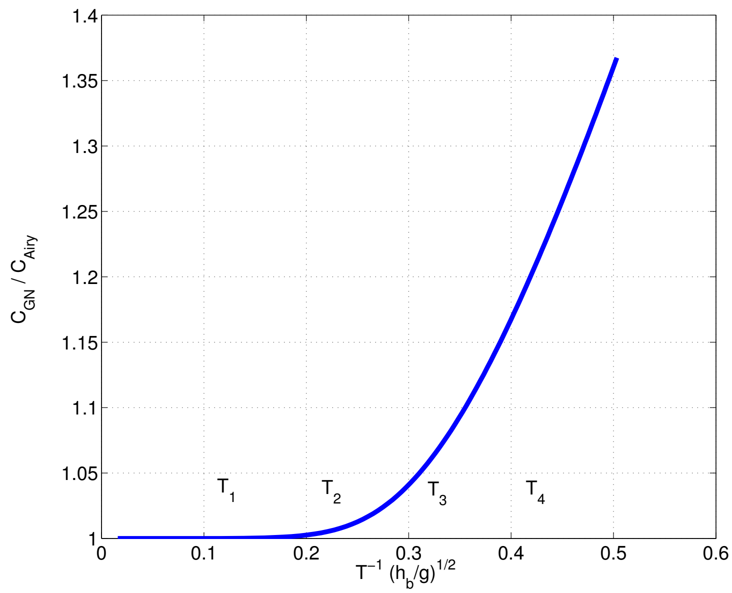

Figure 5.13: Linear dispersion relationship as nondimensional wave speed vs . wave frequency. Vertical lines are waves with periods $T_n = 2.02/n$ s

In Figure 5.14 we compare the results from the RGN model with the results from using a shallow water model at $x = 17.3m$ in the same grid and using the same polynomial order $K = 1$. As we can see we miss the dispersion characteristics when using a shallow water model. Moreover, to account for the sharp change in bathymetry we need to utilize a *slope limitter* (Cockburn, 2003). Here we have used the simplest min-mod limitter. Hence, the shallow water results are a little dissipative.

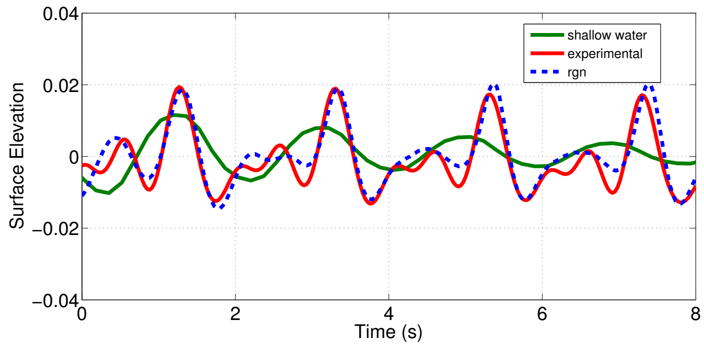

Figure 5.14: Comparison of RGN model, Shallow Water model and experimental result at x = 17.3m

5.3 Wave reflection of solitary wave from a vertical wall

Solitary wave reflection exhibits complex non-linear and dispersive phenomena and has been used as a validation case for numerous numerical models based on Boussinesq - Green - Naghdi equations. Experimental observations in (Su & Mirie, 1980)(Chan & Street, 1970)(Maxworthy, 1976) revealed that solitary waves emerging from a collision, in addition to having experienced changes in their phases, were trailed by a dispersive wave train. Moreover, for large amplitudes, the maximum run-up was observed to be higher than those determined from linear theory.

In this numerical study we follow the numerical setup of (Power &

Chwang, 1984). The initial conditions are (Bonneton et al., 2011):

$$\begin{aligned}
&\eta(x,0) = asech^2(\kappa(x - x_0 - ct)),\\
&u_0(x,0) = c\left(1 - \frac{h_b}{\eta + h_b}\right),\\
&u_1(x,0) = u_2(x,0) = 0,\\
&\kappa = \frac{\sqrt{3a}}{2h_b\sqrt{h_b + a}},\\
&c = \sqrt{g(h_b + a)}.
\end{aligned}$$

The initial velocity is such that continuity is satisfied at $t = 0$ and the initial configuration is shown in Figure 5.15. As the solitary wave moves closer to the wall where the reflection takes place, its amplitude as well as its phase velocity increases quite rapidly. When the wave crest reaches the wall, it doesen't immediately reflect back. There is phase lag during which the amplitude increases to more than double the initial amplitude. This maximum run-up against a vertical wall is compared against experimental results of (Maxworthy, 1976)(Chan & Street, 1970) reported in (Power & Chwang, 1984) in Figure 5.16. A non-uniform grid of $h_{\max} = 0.5$ toward the left of the domain and $h_{\min} = 0.2m$ near the wall is used and a polynomial order of $K = 3$ is taken. The numerical results agree well with the experimental data.

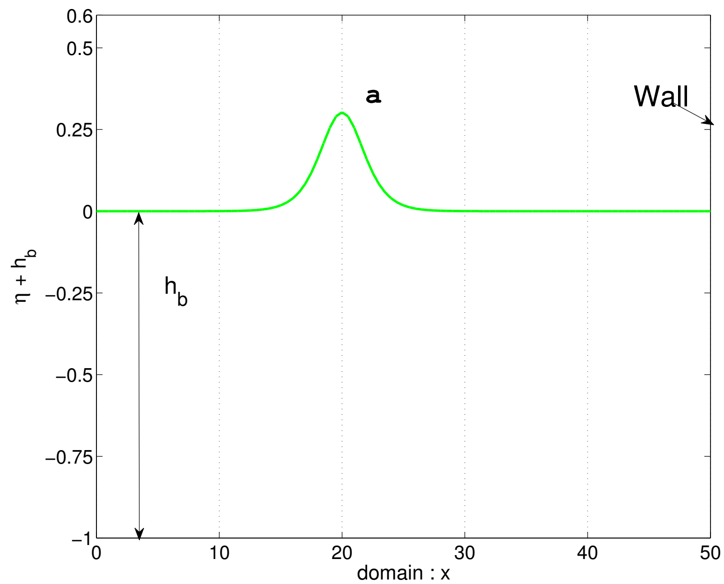

Figure 5.15: Intial configuration for the validation case

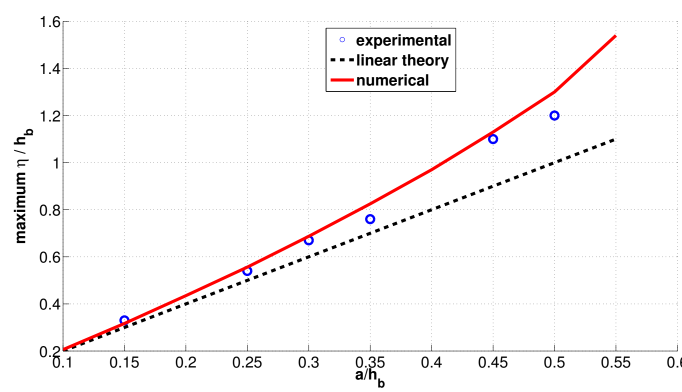

Figure 5.16: Maximum Surface elevation Vs the initial amplitude

5.4 Solitary wave progation over sloping beach

In this test we perform the numerical validation of the propagation of a non-linear and non-breaking solitary wave over a sloped beach and its reflection from the wall. This test case captures both non-linear and dispersive effects.

The domain is shown in Figure 5.17. The beach slopes at 1 : 50 and is terminated in the end by a wall. The recording location is at $x = 17.75m$ and the surface elevation is recorded of the propagating and reflecting wave. The polynomial order is taken to be $K = 1$ and a uniform grid of $h = 0.2m$ is taken in the numerical simulation.

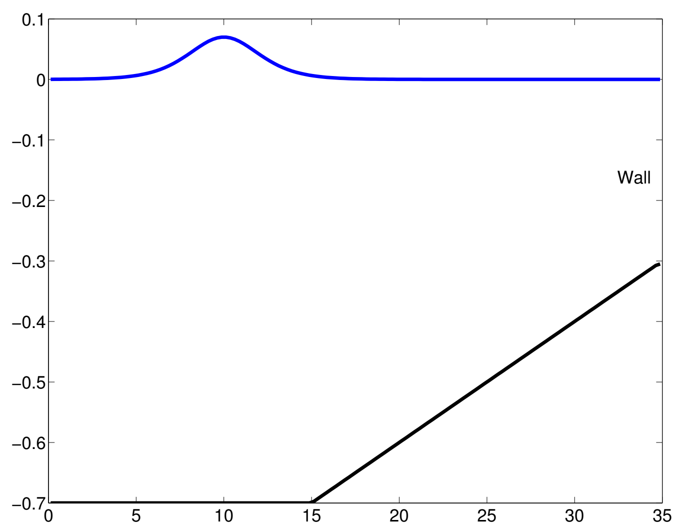

Figure 5.17: Intial configuration for the validation case.

The solitary wave is generated using the wave generation technique and reflected using wall boundary conditions. The setup is shown in Figure 5.18.

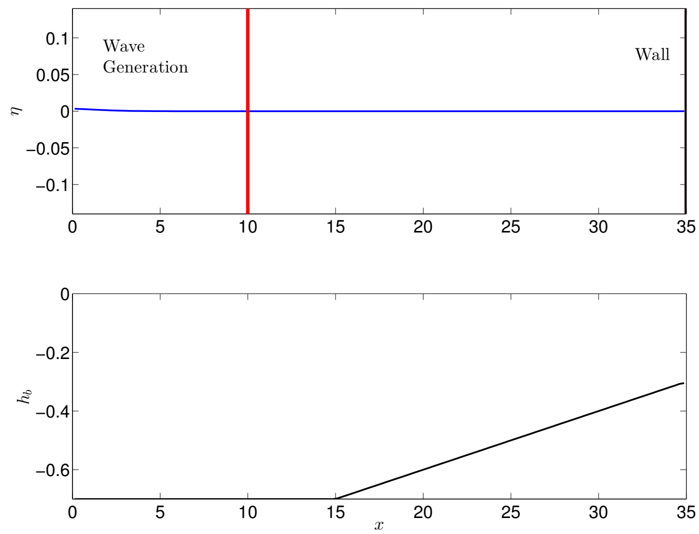

Figure 5.18: Wave generation set up of the validation case.

The surface elevation recording at $x = 17.75m$ is shown in the Figure 5.19. The first peak corresponds to the incident wave while the second peak corresponds to the reflected wave and is higher in amplitude. We also plot the surface elevation at various locations along the sloped beach in Figure 5.20. We can observe that the incident solitary wave gains amplitude as it progresses over the sloped beach while the reflected wave is of higher amplitude and leaves a dispersive wave train. The peak non-dimensional value as reported in (Bonneton et al., 2011) is around 1.3 and matches well with the numerical value shown in the following figure 5.19.

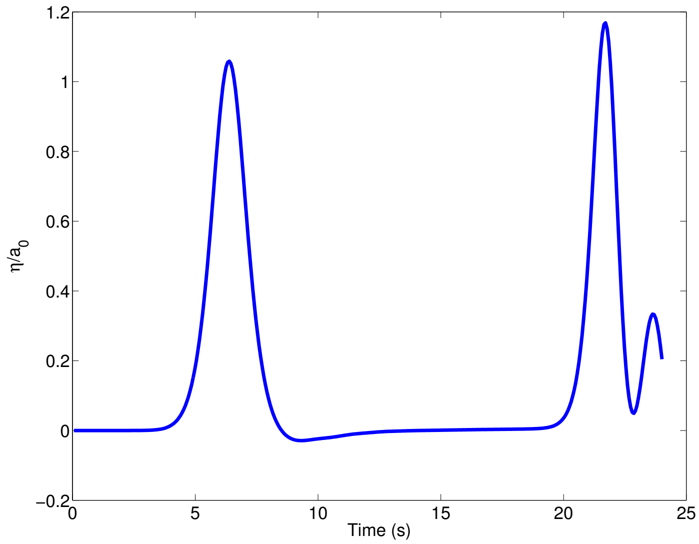

Figure 5.19: Non-dimensional surface elevation at $x = 17.75m$.

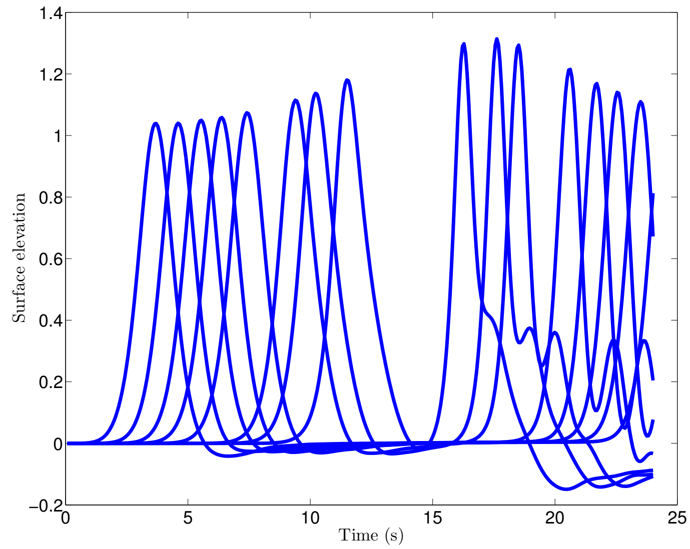

Figure 5.20: Time history of non-dimensional surface elevation at locations along the beach.

In this section we performed extensive validation of the numerical method with experimental results. The numerical results agree well with the experimental data. By using the discontinuous Galerkin framework we were able to use non-uniform grid in our numerical simulation and this feature is extremely advantageous when we extend our method to solve $2D$ problems. All our validation so far has been for the inviscid cases and wave breaking due to turbulence will be treated in future work.

Chapter 6

Conclusion

In this work we developed a new local discontinuous Galerkin finite element method to solve Green-Naghdi Equations in modeling non-linear and dispersive water waves. Two broad classes of Green-Naghdi Equations namely the **R-GN** and **I-GN** models were considered and a numerical discretization scheme was outlined for both.

A careful stability analysis based on the Fourier transformation was then carried out for the linearized R-GN equations. The eigenspectra was found to be complex and the magnitude was bounded. *Flux criterion* for the numerical method was then established from the stability analysis of the method based on the discontinuous Galerkin framework. A general non-linear stability analysis has been left for future work, however, a few comments on achieving long time stability were also presented. In general, high order approximation for extremely non-linear cases need additional stability which may render the scheme *inconsistent*. However lower order approximation have been observed to be stable provided the correct bi-linear forms are used as defined in (4.7).

The final part consisted of verification and validation of the R-GN model. A linear standing wave in a flat bathymetry with known exact solution was used for the verification of the linearized equations. Pointwise error at $x = L/2$ was used to compare solutions with different mesh refinement and polynomial order for the complete non-linearRGN equations. Error plots were shown to give optimal/sub-optimal h, K convergence rates. For validation, three challenging test cases were considered. Wave transformation over a submerged shoal, solitary wave reflection from a vertical wall and solitary wave propogation over a sloping beach were chosen and the numerical results show good agreement with the experimental values. Such validation schemes have been standard benchmarks to test not only linear dispersion properties but also complex non-linear transformations.

Although Green-Naghdi equations have been used to model complex non-linear and dispersive water wave characteristics, the inclusion of non-linear products of higher order derivatives in *non-conservative* form has made it cumbersome for the development of numerical schemes in non-uniform grids. The present numerical method hopes to remove this difficulty in using Green-Naghdi based models for modeling near-shore phenomenon. Future work will consider the following:

- Include viscous terms and validate wave breaking.
- Carry out extensive tests on arbitrary non-uniform grids in $1D$.
- Extend the $1D$ implementation to solve the full $R - GN$ equations in $2D$.

- Numerically couple near shore wave model with general circulation and shallow water model.
- Add uncertainty quantification to the near shore models.

Acknowledgments

This work was funded under the National Science Foundation grant OCE 1025561 and 1025519. Their support is gratefully acknowledged. Digital data files for wave propagation over a shoal were provided by Maarten Dingemans.

Appendix

Appendix 1

Appendix

In this section, we'll complete the description of the R-GN equations. As described in (2.11), the approximate velocity field is expanded in the *shape functions* $f_n(q)$, where q is a non-dimensional parameter that varies from 0 at the bottom to 1 at the surface elevation. Based on a given shape function $f_n(q)$, the Table 1.1 below gives some useful integral definitions (Zhang et al., 2013).

$g_n = \int f_n dq$	$r_n = \int f_n' q dq$	$G_n = \int g_n dq$
$R_n = \int r_n dq$	$\phi_{mn} = \int f_m f_n dq$	$\gamma_{mn} = \int f_m g_n dq$
$\rho_{mn} = \int f_m r_n dq$	$\Gamma_{mn} = \int f_m G_n dq$	$\Theta_{mn} = \int f_m R_n dq$
$\theta_{mn} = \int f_m g_n q dq$	$\nu_m = \int q^2 f_m dq$	$S_m = \int q f_m dq$
$\epsilon_{mn} = \int f_m f_n' q dq$	$\Psi_{mn} = \int f_m f_n q dq$	$F_{mn} = \int f_m r_n q dq$

Table 1.1: Integrals based on the shape function

Using these, we can define the constants introduced in (2.12) and (2.13).

$$
\begin{aligned}
&c_1 = g_1\\
&c_2 = g_2\\
&c_1^m = c_2^m = c_3^m = g_m\\
&c_4^m = \phi_{mn};\ c_5^m = \epsilon_{mn};\ c_6^m = g_m - \nu_m\\
&c_7^m = g_m;\ c_8^m = g_m - S_m;\ c_9^m = g_m;\ c_{10}^m = S_m\\
&c_{11}^m = \phi_{mn};\ c_{12}^m = \epsilon_{mn};\ c_{13} = g_m\\
&c_{14}^m = g_m - S_m;\ c_{15}^m = g_m - \nu_m;\ c_{16}^m = g_m;\ c_{17}^m = g_m - S_m
\end{aligned}
\tag{1.1}
$$

where all the integrals defined in the table above are evaluated at $q = 1$. For the 1D R-GN equations introduced in (4.10), we get the following terms:

Rhs_η is given by:

$$
\begin{aligned}
&-\big(u_0H_{,x} + u_{0,x}H + \mu^2 g_1 u_1 H_{,x} + \mu^2 g_1 u_{1,x} H + \mu^2 g_2 u_2 H_{,x}\\
&+ \quad \mu^2 g_2 u_{2,x} H\big)
\end{aligned}
\tag{1.2}
$$

Rhs_{u_0} is given by:

$$
\begin{aligned}
&-\big(d_0u_0u_{0,x} + e_0u_{0,x}^2 + f_0u_0u_{0,xx} + h_0u_{0,x}u_{0,xx} + i_0u_0u_{0,xxx}\\
&+ \quad j_0u_0^2 + k_0u_1u_{0,x} + l_0u_0u_{1,x} + n_0u_2u_{0,x} + o_0u_0u_{2,x}\\
&+ \quad p_0u_1u_0 + q_0u_2u_0 + r_0u_1 + t_0u_2 + v_0g\eta_{,x})
\end{aligned}
\tag{1.3}
$$

Rhs_{u_1} is given by:

$$
\begin{aligned}
&-\big(a_1s_0 + b_1s_{0,x} + c_1s_{0,xx} + d_1u_0u_{0,x} + e_1u_{0,x}^2 + f_1u_0u_{0,xx}\\
&+ \quad h_1u_{0,x}u_{0,xx} + i_1u_0u_{0,xxx} + j_1u_0^2 + k_1u_1u_{0,x} + l_1u_0u_{1,x}\\
&+ \quad n_1u_2u_{0,x} + o_1u_0u_{2,x} + p_1u_1u_0 + q_1u_2u_0 + r_1u_1\\
&+ \quad t_1u_2 + v_1g\eta_{,x})
\end{aligned}
\tag{1.4}
$$

Rhs_{u_2} is given by:

$$
\begin{aligned}
& -\big(a_2 s_0 + b_2 s_{0,x} + c_2 s_{0,xx} + d_2 u_0 u_{0,x} + e_2 u_{0,x}^2 + f_2 u_0 u_{0,xx} \\
+ \quad & h_2 u_{0,x} u_{0,xx} + i_2 u_0 u_{0,xxx} + j_2 u_0^2 + k_2 u_1 u_{0,x} + l_2 u_0 u_{1,x} \\
+ \quad & n_2 u_2 u_{0,x} + o_2 u_0 u_{2,x} + p_2 u_1 u_0 + q_2 u_2 u_0 + r_2 u_1 \\
+ \quad & t_2 u_2 + v_2 g \eta_{,x}\big)
\end{aligned} \tag{1.5}
$$

Rhs_1, and Rhs_2 are given by:

$$
\begin{aligned}
Rhs_1 &= \frac{\phi_{12} Rhs_{u_2} - \phi_{22} Rhs_{u_1}}{\phi_{12}\phi_{21} - \phi_{22}\phi_{11}} \\
Rhs_2 &= \frac{\phi_{21} Rhs_{u_1} - \phi_{11} Rhs_{u_2}}{\phi_{12}\phi_{21} - \phi_{22}\phi_{11}}
\end{aligned} \tag{1.6}
$$

For $m = 0$, the coefficients are given as:

$$
\begin{aligned}
d_m &= Hg_m - \mu^2 H\eta_{,x}h_{b,x}\tilde{g}_m + (3h_{b,xx})(-\mu^2 H^2(\tilde{g}_m - \tilde{S}_m)) \\
e_m &= \mu^2 H^2 \eta_{,x}\tilde{g}_m \\
f_m &= -\mu^2 H^2(\eta_{,x}\tilde{g}_m + 2(\tilde{g}_m - \tilde{s}_m)h_{b,x}) \\
h_m &= +\frac{\mu^2}{2}H^3(\tilde{g}_m - \tilde{\nu}_m) \\
i_m &= \frac{-\mu^2}{2}H^3(\tilde{g}_m - \tilde{\nu}_m) \\
j_m &= -\mu^2 H\eta_{,x}h_{b,xx}\tilde{g}_m - h_{b,xxx}\mu^2 H^2(\tilde{g}_m - \tilde{S}_m) \\
k_m &= \mu^2 H(-\tilde{\epsilon}_{m1}) \\
l_m &= 0 \\
n_m &= \mu^2 H(-\tilde{\epsilon}_{m2}) \\
o_m &= 0 \\
p_m &= -\mu^2 H_{,x}\tilde{\epsilon}_{m1} \\
q_m &= -\mu^2 H_{,x}\tilde{\epsilon}_{m2} \\
r_m &= -\mu^2 \eta_{,t}\tilde{\epsilon}_{m1} \\
t_m &= -\mu^2 \eta_{,t}\tilde{\epsilon}_{m2} \\
v_m &= H\tilde{g}_m
\end{aligned}
\tag{1.7}
$$

For $m = 1, 2$, the coefficients are given as:

$$
\begin{aligned}
a_m &= Hg_m - \mu^2 h_{b,x}\eta_{,x}Hg_m - \mu^2 h_{b,xx}H^2(g_m - S_m)\\
b_m &= -\mu^2 H^2 H_{,x}g_m - \mu^2 h_{b,x}H(g_m - S_m) + \mu^2 H^2 h_{b,x}S_m\\
c_m &= \frac{-\mu^2}{2}H^3(g_m - \nu_m)\\
d_m &= Hg_m - \mu^2 H\eta_{,x}h_{b,x}g_m + (3h_{b,xx})(-\mu^2 H^2(g_m - S_m))\\
e_m &= \mu^2 H^2 \eta_{,x}g_m\\
f_m &= -\mu^2 H^2(\eta_{,x}g_m + 2(g_m - S_m)h_{b,x})\\
h_m &= +\frac{\mu^2}{2}H^3(g_m - \nu_m)\\
i_m &= \frac{-\mu^2}{2}H^3(g_m - \nu_m)\\
j_m &= -\mu^2 H\eta_{,x}h_{b,xx}g_m - h_{b,xxx}\mu^2 H^2(g_m - S_m)\\
k_m &= \mu^2 H(\phi_{m1} - \epsilon_{m1})\\
l_m &= \mu^2 H\phi_{m1}\\
n_m &= \mu^2 H(\phi_{m2} - \epsilon_{m2})\\
o_m &= \mu^2 H\phi_{m2}\\
p_m &= -\mu^2 H_{,x}\epsilon_{m1}\\
q_m &= -\mu^2 H_{,x}\epsilon_{m2}\\
r_m &= -\mu^2 \eta_{,t}\epsilon_{m1}\\
t_m &= -\mu^2 \eta_{,t}\epsilon_{m2}\\
v_m &= Hg_m
\end{aligned}
\tag{1.8}
$$

See (Zhang et al., 2013) for obtaining $\tilde{}$ quantities of the integrals. We take $\mu = 1$ in all our computations.

References

Aizinger, V., & Dawson, C. (2002). A discontinuous galerkin method for two-dimensional flow and transport in shallow water. *Advances in Water Resources*, *25*(1), 67–84.

Arnold, D. N., Brezzi, F., Cockburn, B., & Marini, L. D. (2002). Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM journal on numerical analysis*, *39*(5), 1749–1779.

Atkinson, K., & Han, W. (2005). *Theoretical numerical analysis* (Vol. 39). Springer.

Babuška, I., Baumann, C. E., & Oden, J. T. (1999). A discontinuous hp finite element method for diffusion problems: 1-d analysis. *Computers & Mathematics with Applications*, *37*(9), 103–122.

Barthélemy, E. (2004). Nonlinear shallow water theories for coastal waves. *Surveys in Geophysics*, *25*(3-4), 315–337.

Bassi, F., & Rebay, S. (1997). A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations. *Journal of computational physics*, *131*(2), 267–279.

Baumann, C. E. (1997). *An hp-adaptive discontinuous finite element method for computational fluid dynamics* (Unpublished doctoral dissertation). University of Texas at Austin.

Baumann, C. E., & Oden, J. T. (1999). A discontinuous hp finite element method for convection diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, *175*(3), 311–341.

Beji, S., & Battjes, J. (1993). Experimental investigation of wave propagation over a bar. *Coastal Engineering*, *19*(1), 151–162.

Bonneton, P., Chazel, F., Lannes, D., Marche, F., & Tissier, M. (2011). A splitting approach for the fully nonlinear and weakly dispersive green–naghdi model. *Journal of Computational Physics*, *230*(4), 1479–1498.

Brenner, S. C., & Scott, R. (2008). *The mathematical theory of finite element methods* (Vol. 15). Springer.

Chan, R. K.-C., & Street, R. L. (1970). A computer study of finite-amplitude water waves. *Journal of Computational Physics*, *6*(1), 68–94.

Chavent, G., & Cockburn, B. (1989). The local projection ppl-discontinuous-galerkin finite element method for scalar conservation laws. *RAIRO Modél. Math. Anal. Numér*, *23*, 565–592.

Chavent, G., & Salzano, G. (1982). A finite-element method for the 1-d water flooding problem with gravity. *Journal of Computational Physics*, *45*(3), 307–344.

Chen, Q., Kirby, J. T., Dalrymple, R. A., Shi, F., & Thornton, E. B. (2003). Boussinesq modeling of longshore currents. *Journal of Geophysical Research: Oceans (1978–2012)*, *108*(C11).

Chen, Z., Cockburn, B., Jerome, J. W., & Shu, C.-W. (1995). Mixed-rkdg finite element methods for the 2-d hydrodynamic model for semiconductor device simulation. *VLSI Design*, *3*(2), 145–158.

Cockburn, B. (2003). Discontinuous galerkin methods. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, *83*(11), 731–754.

Cockburn, B., & Dawson, C. (2000). Some extensions of the local discontinuous galerkin method for convection-diffusion equations in multidimensions. In *The proceedings of the conference on the mathematics of finite elements and applications: Mafelap x (jr whiteman, ed.), elsevier* (pp. 225–238).

Cockburn, B., & Shu, C.-W. (1989). Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework. *Mathematics of Computation*, *52*(186), 411–435.

Cockburn, B., & Shu, C.-W. (1991a). *The p (1-) rkdg method for two-dimensional euler equations of gas dynamics* (Tech. Rep.). DTIC Document.

Cockburn, B., & Shu, C.-W. (1991b). The runge-kutta local projection p1-discontinuous galerkin finite element method for scalar conservation laws. *RAIRO Modél. Math. Anal. Numér*, *25*(3), 337–361.

Cockburn, B., & Shu, C.-W. (1998a). The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, *35*(6), 2440–2463.

Cockburn, B., & Shu, C.-W. (1998b). The runge–kutta discontinuous galerkin method for conservation laws v: multidimensional systems. *Journal of Computational Physics*, *141*(2), 199–224.

Dawson, C., Kubatko, E. J., Westerink, J. J., Trahan, C., Mirabito, C., Michoski, C., & Panda, N. (2011). Discontinuous galerkin methods for modeling hurricane storm surge. *Advances in Water Resources*, *34*(9), 1165–1176.

Dean, R. G., & Dalrymple, R. A. (1991). *Water wave mechanics for engineers and scientists.* World Scientific.

Dingemans, M. (1994). Comparison of computations with boussinesq-like models and laboratory measurements. *Report H-1684.12*, *32*.

Engsig-Karup, A. P. (2006). *Unstructured nodal dg-fem solution of high order boussinesq type equations* (Unpublished doctoral dissertation).

Engsig-Karup, A. P., Hesthaven, J. S., Bingham, H. B., & Madsen, P. A. (2006). Nodal dg-fem solution of high-order boussinesq-type equations. *Journal of engineering mathematics*, *56*(3), 351–370.

Eskilsson, C., & Sherwin, S. J. (2006). Spectral hp discontinuous galerkin methods for modelling 2d boussinesq equations. *Journal of Computational Physics*, *212*(2), 566–589.

Gobbi, M. F., & Kirby, J. T. (1999). Wave evolution over submerged sills: tests of a high-order boussinesq model. *Coastal Engineering*, *37*(1), 57–96.

Green, A., & Naghdi, P. (1976). A derivation of equations for wave propagation in water of variable depth. *Journal of Fluid Mechanics*, *78*(02), 237–246.

Hesthaven, J. S., & Warburton, T. (2007). *Nodal discontinuous galerkin methods: algorithms, analysis, and applications* (Vol. 54). Springer.

Johnson, C., & Pitkäranta, J. (1986). An analysis of the discontinuous galerkin method for a scalar hyperbolic equation. *Mathematics of computation*, *46*(173), 1–26.

Kennedy, A. B., Chen, Q., Kirby, J. T., & Dalrymple, R. A. (2000). Boussinesq modeling of wave transformation, breaking, and runup. i: 1d. *Journal of waterway, port, coastal, and ocean engineering*, *126*(1), 39–47.

Kennedy, A. B., Kirby, J. T., & Gobbi, M. F. (2002). Simplified higher-order boussinesq equations: I. linear simplifications. *Coastal engineering*, *44*(3), 205–229.

Kirby, R. M., & Karniadakis, G. E. (2003). De-aliasing on non-uniform grids: algorithms and applications. *Journal of Computational Physics*, *191*(1), 249–264.

Lannes, D., & Bonneton, P. (2009). Derivation of asymptotic two-dimensional time-dependent equations for surface water wave propagation. *Physics of fluids*, *21*, 016601.

Lesaint, P., & Raviart, P.-A. (1974). *On a finite element method for solving the neutron transport equation.* Univ. Paris VI, Labo. Analyse Numérique.

Lin, P., Chang, K.-A., & Liu, P. L.-F. (1999). Runup and rundown of solitary waves on sloping beaches. *Journal of waterway, port, coastal, and ocean engineering*, *125*(5), 247–255.

Madsen, P. A., & Sørensen, O. R. (1992). A new form of the boussinesq equations with improved linear dispersion characteristics. part 2. a slowly-varying bathymetry. *Coastal Engineering*, *18*(3), 183–204.

Mase, H., & Kirby, J. T. (1992). Hybrid frequency-domain kdv equation for random wave transformation. *Coastal Engineering Proceedings*, *1*(23).

Maxworthy, T. (1976). Experiments on collisions between solitary waves. *Journal of Fluid Mechanics*, *76*(01), 177–186.

Nwogu, O. (1993). Alternative form of boussinesq equations for nearshore wave propagation. *Journal of waterway, port, coastal, and ocean engineering*, *119*(6), 618–638.

Peregrine, D. (1967). Long waves on a beach. *J. Fluid Mech*, *27*(4), 815–827.

Power, H., & Chwang, A. T. (1984). On reflection of a planar solitary wave at a vertical wall. *Wave Motion*, *6*(2), 183–195.

Rayleigh, L. (1876). On waves. *Phil. Mag*, *1*(5), 257–279.

Reed, W. H., & Hill, T. (1973). Triangularmesh methodsfor the neutrontransportequation. *Los Alamos Report LA-UR-73-479*.

Serre, F. (1953). Contribution à l'étude des écoulements permanents et variables dans les canaux. *La Houille Blanche*(6), 830–872.

Shields, J. J., & Webster, W. C. (1988). On direct methods in water-wave theory. *Journal of Fluid Mechanics*, *197*, 171–199.

Su, C., & Mirie, R. M. (1980). On head-on collisions between two solitary waves. *Journal of Fluid Mechanics*, *98*(03), 509–525.

Ting, F. C., & Kirby, J. T. (1994). Observation of undertow and turbulence in a laboratory surf zone. *Coastal Engineering*, *24*(1), 51–80.

Ting, F. C., & Kirby, J. T. (1995). Dynamics of surf-zone turbulence in a strong plunging breaker. *Coastal Engineering*, *24*(3), 177–204.

Van Leer, B. (1974). Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *Journal of computational physics*, *14*(4), 361–370.

Webster, W., & Kim, D. (1991). The dispersion of large-amplitude gravity waves in deep water.

Yan, J., & Shu, C.-W. (2002). A local discontinuous galerkin method for kdv type equations. *SIAM Journal on Numerical Analysis*, *40*(2), 769–791.

Zhang, Y., Kennedy, A. B., Panda, N., Dawson, C., & Westerink, J. J. (2013).

Boussinesq–green–naghdi rotational water wave theory. *Coastal Engineering*, *73*, 13–27.

Zhang, Y., Kennedy, A. B., Panda, N., Dawson, C., & Westerink, J. J. (2014). Generating–absorbing sponge layers for phase-resolving wave models. *Coastal Engineering*, *84*, 1–9.