Spectral Multigrid for Forced Internal Solitary Wave Computations

Keith Chor Hei Chan

BSc(Hons), Monash University, 2024

A thesis submitted for the degree of

Doctor of Philosophy

at Monash University in 2024

School of Mathematics
Monash University
Melbourne, Australia
2024

Copyright notice

© Keith Chor Hei Chan (2024)

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

Abstract

Newton-Krylov methods have been shown to be excellent solvers in solitary wave computations, outperforming existing ones such as the Petviashvili, imaginary time evolution or squared-operator iteration among others. Since multigrid is generally comparable to Krylov methods in efficiency, we aim to explore the viability of spectral multigrid for solitary wave computations. Variants of the nonlinear Schrödinger and the Kadomtsev–Petviashvili equation are solved. We find that spectral multigrid either performs similarly or outperforms Krylov methods, demonstrating their usefulness.

We then proceed to test spectral multigrid on the steady forced Korteweg De-Vries equation, which has application to stratified and free surface flows. Algebraically and exponentially decaying solitary waves are investigated for a single bump topographic forcing. Numerical results suggest there may exist an infinite set of exponentially decaying solitary waves corresponding to a nonlinear eigenvalue problem. Numerical continuation further shows the connection between the two types of decaying solutions. We then consider table-top solutions, which are the symmetric variant of asymmetric hydraulic fall solutions, to the steady forced Korteweg De-Vries equation with a double bump topographic forcing. The hydraulic fall parameter space is numerically mapped and extended in the negative topography region, showing hydraulic fall solutions periodically evolve into exponentially decaying solitary waves.

Finally, we investigate the Dubriel-Jacotin–Long equation with constant background current under the Boussinesq approximation in the presence of a bottom topography using spectral multigrid. We focus on single and double bump topographic forcing. The weakly nonlinear theory to the Dubriel-Jacotin–Long equation gives rise to the steady forced Korteweg De-Vries equation. A novel iterative numerical conformal mapping is introduced to handle the non-rectangular geometry due to the bottom topography. The weakly nonlinear theory and Dubriel-Jacotin–Long solutions are then compared for three stratifications up to the point of overturning.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Keith Chor Hei Chan
11 November 2024

To my parents.

Acknowledgements

My PhD has been a long and arduous journey and would not have been possible without the generous support from some key people. First and foremost I owe particular thanks my supervisor Simon Clarke for his support and guidance throughout the years. Without his expert insight and allowing my frequent knocks on his office door for questions, the completion of this PhD would not have been possible.

To my family, especially my parents for providing support in other aspects of my life, for allowing me to focus on my research during these years.

Finally to my fellow colleagues whom I shared an office with across the years. Special mentions to Hridya and Jiahao. Our badminton sessions, appetising lunches, bubble tea, and first year fundamental theorems have been an absolute pleasure.

Contents

Chapter 1

Introduction

1.1 Internal solitary waves and the Dubriel-Jacotin–Long equation

Internal waves are commonly found in the ocean [4] and in the atmosphere [23], and are driven by buoyancy forces due to fluid stratification, that is, the density changes within the fluid. This can be due to factors such as temperature and salinity in the ocean or compressibility effects in the case of the atmosphere. In the latter case, internal waves influence its thermal structure and wind speeds, whereas in the former, they travel within the interior of a fluid as oppose to observable surface waves and are particularly important as they affect mixing through the transport of energy.

Of particular interest here are internal solitary waves (ISWs) which are translating waves of permanent forms in stratified fluid. They have an important role in marine ecosystems, for example in conveying nutrients from deep water to shallower layers [98]. Furthermore, the understanding of them is necessary in the future design of deep-water production facilities [85]. Examples of ISWs in the ocean include mode 2 solitary waves in the Australia North West shelf [93], very large and steep internal waves of elevation near the Massachusetts coast [100] and large-amplitude ISWs observed in the northern South China Sea [77] among others. Figure 1.1 shows an ISW measured by ship borne sensors at the north of the Strait of Messina.

Stratified flow can be described by the incompressible stratified Euler equations. For steady solitary wave solutions, the Dubriel-Jacotin–Long (DJL) equation is equivalent to the full set of stratified Euler equations. Let $\eta(x^*, z^*)$ be the isopycnal displacement, that is the distance the density contour is displaced from the far upstream or downstream value, U the background current, H^* be the height of the fluid and N^2 the squared buoyancy

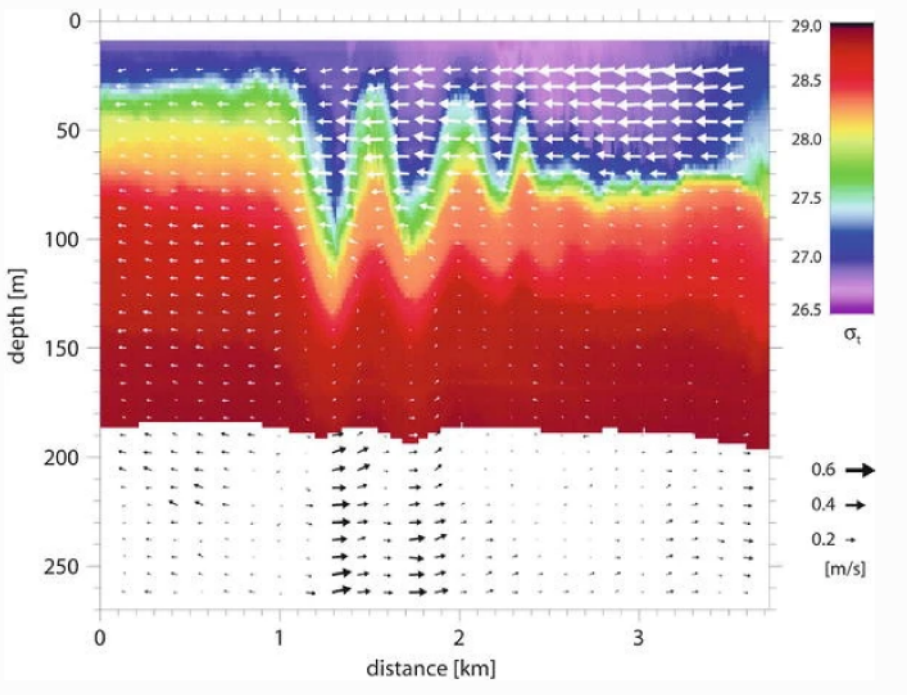

Figure 1.1: Figure taken from Alpers (2014) [3]. Density distribution of the water column and distribution of the velocity north of the Strait of Messina measured by ship borne sensors during the passage of a highly nonlinear internal wave packet on October 25, 1995.

frequency. The DJL equation then is

$$\nabla^2\eta + \frac{N^2(z^* - \eta)}{U^2}\eta = 0. \tag{1.1}$$

The isopycnal displacement tends to zero far upstream, or as $x^* \to -\infty$, and downstream, or as $x^* \to \infty$. Together with the rigid lid approximation gives the DJL boundary conditions

$$\eta(x^*, H^*) = 0, \tag{1.2a}$$
$$\eta(x^*, 0) = 0, \tag{1.2b}$$
$$\eta(\pm\infty, z^*) = 0. \tag{1.2c}$$

It was first derived by Dubriel-Jacotin [32] and popularised by Long [78] who considered this equation with constant buoyancy frequency. The DJL equation has been successful in modeling solitary waves observed in nature, for example those in the northwest South China Sea by Lien et al. [77] were modeled by He et al. [53]. Further numerical simulations [114] have been used to reproduce the trapped cores phenomena, which is an area of recirculating fluid, observed in nature and in experimentation [79].

The presence of a bottom topography gives rise to the forced DJL equation and can act as a forcing mechanism for internal waves in a stratified fluid. Let $d(x^*)$ be the bottom

topography, the boundary condition (1.2b) is then replaced by

$$\eta(x^*, d(x^*)) = d(x^*). \tag{1.3}$$

Many bottom topography types have been considered such as hills and corrugated bed [21, 114]. This is generally handled by a mapping [104] or embedding method [75, 103]. The former requires finding a map from the rectangular domain with bottom topography to a rectangular one before solving, whereas the latter iterates the solution to match the known bottom boundary. The DJL equation with constant background current have been solved by various methods, most notably the iterative algorithm of Turkington et al. [112]. It was later extended to non-constant background current by Dunphy et al. [33]. Another popular method is to introduce a time-like variable that extends the DJL equation to a parabolic form [103, 105]. The steady solution is then obtained by integrating forwards in time until a steady state is reached. For single bump topography, Soontiens et al. [103] and Stastna [105] investigated supercritical trapped waves over positive and negative bump topography. Dunphy et al. [33] also investigated a single bump topography while comparing the DJL solutions to weakly nonlinear (WNL) theory. Soontiens et al. [103] investigated double bump topography in the case of subcritical flows with the Euler equations to obtain asymmetric flows across the topographic maximum.

An assumption made in the derivation of the DJL equation based on conjugate flow theory [105] is that the streamlines connect to $\pm\infty$ in the horizontal or x direction [106, 105]. If any streamlines in the DJL solution are vertical, that is perpendicular to the x direction, there must exist a region of recirculating fluid also known as trapped cores, where they do not connect to the boundaries. We denote this behaviour as *overturning*. Therefore, a solution exhibiting it is invalid within the recirculating region. In this thesis, we shall ignore them.

Perturbation methods can be used on the stratified Euler equations to obtain the WNL equations, originally developed by Benney [7]. This leads to the Korteweg-de Vries (KdV) equation and its higher order extensions. Comparisons of the fully nonlinear model to the WNL theory were investigated by Lamb et al. [73, 74], showing agreement for small waves but breaking down for large ones. Further examples of comparisons of DJL to WNL have been performed by Stastna and Lamb [106], and Stastna [105] among others. In the case of a bottom topography, the WNL theory requires the solution to the forced KdV (fKdV) equation for the amplitude of the resonant internal wave mode,

$$A_t - \Delta A + 6AA_x + A_{xxx} = -\gamma f_x, \tag{1.4}$$

where x and t are the spatial and time variables, respectively, Δ is the detuning factor related to the Froude number via $\Delta \propto F - 1$, f is the external topographic forcing and γ

is the forcing coefficient.

1.2 The forced Korteweg-de Vries equation

In addition to application in internal wave dynamics, the fKdV equation also has applications to other branches of fluid dynamics such as Rossby [95] and free-surface waves generated by topographic disturbances in a two-dimensional channel [29, 50]. When near resonance, the generated surface waves can damage the integrity of water banks and hydraulic infrastructures. Rossby waves, also known as planetary waves, occur naturally in rotating fluids and are particularly important in the Earth's oceans and atmosphere, which have a major influence on weather.

The flow is characterised by the dimensionless flow rate given by the Froude number F that is, the ratio of the flow velocity and the linear long wave speed. We are particularly interested in steady fKdV solutions, which are utilised in the WNL theory for stratified flow over topography. The Froude number classifies the solution to the steady fKdV equation according to 11 basic steady flow types [8]. When $F < 1$ the flow is classified as subcritical, and supercritical for $F > 1$.

We focus on steady hydraulic fall solutions that involve a transition between subcritical and supercritical flow. This has been observed for various types of topographical disturbances including semi-circular (both numerically and experimentally [41]), bell-shaped [30], triangular [29], and delta function forcing [102]. We are especially interested in the parametric relationship between the two parameters Δ and bump topography height γ, for the existence of hydraulic fall solutions.

Asymptotic methods can be used to determine the hydraulic fall parameter space in various limits. Let positive and negative γ represent a bump and hole topography, respectively. Grimshaw and Smyth [50] used asymptotic analysis to show the parameter space in the small forcing limit $\gamma \to \pm 0$. In the large positive forcing limit, hydraulic approximation, that is where dispersive effects are negligible, can be used to give the relation $\Delta \propto \sqrt{\gamma}$ [34]. However, no known theoretical results exist in the large negative forcing limit.

Ee and Clarke [34] numerically determined the parametric relationship for hydraulic fall solutions for a sech^2 forcing. Their results were consistent with the theoretical asymptotic analysis. A minimisation algorithm based on the Hamiltonian of the fKdV equation was used. An identical topography added far downstream allows asymmetric hydraulic fall solutions to be analysed in a symmetric manner. They are known as tabletop solutions and are symmetric over the domain but are not over each bump. The flow transitions

from a subcritical to supercritical flow over the first bump, then transitions back to subcritical over the second bump. Despite one half being potentially unstable, it appears to be the symmetric variant of asymmetric hydraulic fall solutions, which are then recovered by only considering one half of the domain. Ee and Clarke observed that hydraulic fall solutions periodically transform into solitary waves in the negative γ region, where two such solutions were embedded in the parametric relationship. As the topographic forcing is exponentially decaying, these solitary waves therefore have exponential far field decay. Tabletop solutions have been similarly observed by Chardard et al. [22] with a sech^2 forcing and Lee and Whang [76] with a $\cos^4$ one. The latter was mapped by Cullen [27]. A minimisation of the wave drag [113] was instead used to filter for hydraulic fall solutions, with artificial boundary conditions for the truncated numerical domain. However, both Ee and Clarke, and Cullen experienced numerical difficulties in mapping the extended negative γ region.

Another type of forced solitary wave was observed by Keeler et al. [67], known as critical flow. These exist on a continuum in γ. Asymptotic analysis show that these solutions have algebraic far field decay. Furthermore numerical results show multiple types of algebraically decaying solitary solutions not predicted by their asymptotic analysis.

1.3 Solitary wave computations

There are several methods available for the numerical computation of solitary waves more general than the ones discussed previously for the DJL equation. The simplest are Newton's and shooting method [11], the former involves linearising the problem and inverting the resulting Jacobian matrix by either LU or QR factorization as part of Newton iterations [110]. Solving the matrix equation however can become difficult especially in higher dimensions where the resulting matrix is no longer tridiagonal. The shooting method instead transforms the boundary value problem to an initial value one. The unknown initial condition, which matches the correct boundary values of the original problem is then solved for. This is very efficient and accurate but unfortunately only applies to one-dimensional problems. Next, we discuss other more sophisticated methods available in literature.

The Petviashvili method was first introduced in 1976 [88] to solve constant-coefficient equations with power-law nonlinearity before extensions to more general wave equations were given by Pelinovsky et al. [87]. A modified version was used to calculate two-pulse solutions to the fifth-order KdV equation by Chugunova [24]. It is based on a fixed-point iteration with a stabilising factor. Although this method is fast and easy to implement, it only converges to the ground state, that is the lowest energy state, of nonlinear wave equations and diverges for non-ground or excited ones [81].

Imaginary time evolution methods (ITEM) turn solitary wave equations into a fictitious diffusion equation while normalizing the solution after each time step. A soliton solution would then be obtained if the evolution reaches a steady state. ITEM methods are quite slow but the recently developed Accelerated Imaginary Time Evolution methods (AITEM) [44] improved convergence speeds dramatically by introducing an acceleration operator making it competitive with Petviashvili methods [118]. However, similar to the Petviashvili method, AITEM can only converge to ground states.

To calculate excited states, squared-operator iteration methods (SOM) were developed by Yang et al. [117]. This class of method considers the time evolution of Hermitian and preconditioning operators, which allows convergence to any solitary wave both ground and excited states. An acceleration operator can be included to improve convergence speeds. Yang [117] introduced the modified squared-operator method (MSOM) which employs an eigenmode-elimination technique to allow for faster convergence giving it better performance over other squared-operator.

Newton-Krylov methods, in particular Newton-conjugate-gradient (Newton-CG) and Newton-biconjugate-gradient methods (Newton-BiCG), were introduced by Yang [115] for solitary wave computations. They use Fourier spectral methods and incorporates a Newton outer iteration in combination with a Krylov inner iteration. Yang tested this on a variety of physical models including the 2D nonlinear Schrödinger (NLS) equations (with and without periodic potentials), fifth-order Korteweg-de Vries (KdV) equation and the fifth-order Kadomtsev–Petviashvili (KP) equation with comparisons to leading existing methods; Petviashvili, AITEM and MSOM. The results show Newton-CG is much more efficient, often by orders of magnitude, over existing methods, while being more robust and converging for both ground and excited states. Comparisons of Newton-CG and nonlinear-conjugate-gradient methods also show that the former is less sensitive to initial conditions.

1.4 Spectral Multigrid

Motivated by Yang's results, a goal of this thesis it to explore the viability of spectral multigrid (SMG) for solitary wave computations. Since Krylov methods are known to be comparable to multigrid in efficiency when used with a FD or FE discretisation, we anticipate similar efficiency with spectral discretisation.

SMG was introduced by Zang et al. [121] for Fourier and Chebyshev collocation methods and combines spectral discretisation with multigrid concepts. SMG takes advantage of the ability of spectral methods to achieve accurate results with fewer grid points than

typical FD approximation due to their superior convergence. Multigrid ideas are then used to accelerate iterative schemes.

Originally popularized by Brandt [12] using FD discretisation, multigrid has become one of the fastest solvers for elliptic problems and can be easily extended to nonlinear ones through the full approximation scheme (FAS). Using local mode analysis, Brandt showed simple iteration schemes such as Gauss–Seidel and Jacobi are efficient at removing high frequency errors while perform poorly suppressing low frequency ones. Since low frequencies are interpreted as high ones on coarser grids, simple iterative schemes can efficiently remove all errors by iterating on multiple grid levels. Furthermore, iterations on coarse grids are significantly cheaper than on fine grids.

Combining spectral collocation methods with multigrid gives the basic idea of SMG. However, there are several challenges that need to be addressed, namely, the efficient relaxation schemes for dense spectral matrices and efficient spectrally accurate grid transfer operators. Zang [121, 122] addressed the relaxation scheme by using Richardson class of iterations, which allow the use of Fast Fourier Transforms (FFT). Spectrally accurate grid transfer operators used trigonometric interpolation, which also can utilise FFTs. SMG was then implemented successfully by Zang on simple model problems. Improvements to Fourier SMG was made by Brandt et al. [13] and comparisons of it to other solvers were made by Canuto et al. [19], demonstrating its usefulness. Further development by Heinrichs, include SMG for combined FD and Fourier problems [55], algebraic spectral multigrid methods (ASMG) [56] and domain decomposition problems [60].

SMG has since been applied in many fluid problems, initially for inviscid, compressible flows [65], then applications to transonic potential flow [107], followed by Navier–Stokes equations [62, 120]. Other applications of SMG to fluid dynamics problems include the solution of homogeneous turbulence problems [39], the Stokes problem in streamfunction formulation [58, 61], for barotropic nondivergent flow over irregular coastal topography [15], and variants of the Navier–Stokes equations [123, 70].

1.5 Thesis outline

This thesis has two aims. First is to explore the viability of SMG for solitary wave computations, and second, to investigate solitary wave solutions to the single and double bump forced DJL and fKdV equations, respectively. We begin by comparing SMG to Krylov methods for the problems investigated by Yang [115]. We then look to apply SMG to the one-dimensional fKdV equation with single and double bump topography using Fourier collocation methods, before proceeding with the two-dimensional DJL equation using Fourier-Chebyshev collocation methods.

The outline for this thesis is as follows. We begin with a review of spectral methods in Chapter 2, specifically Fourier and Chebyshev collocation methods. Efficient techniques for spectral transform, differentiation and integration are then explored. This allows us to discretise linear differential equations into spectral linear systems. Next we review solvers for spectral systems, where Ad Hoc exact methods are limited for Laplace and Helmholtz equations while iterative methods are used in all other situations. Since the matrices resulting from spectral systems are dense, we are restricted to Richardson and Krylov class of iterative methods. Preconditioners are also explored to accelerate the iterative methods. This ties into SMG, and the components of SMG that differs from FD multigrid: namely, the discretisation, relaxation scheme and grid transfer operators. Here Fourier and Chebyshev collocation is discussed for the discretisation, Richardson and Krylov methods for relaxation, and Fourier and Chebyshev interpolation for the grid transfer operator. Newton's method is discussed for nonlinear problems and used within predictor-corrector numerical continuation methods, namely, natural parameter and pseudo arc-length continuation. Finally, a novel numerical iterative conformal mapping method is proposed, specifically for the mapping of domain with a non-flat bottom, to a rectangular one. Fast Ad Hoc exact solvers are utilised within the numerical conformal mapping thus resulting in a total operation count of $O(N \log N)$. Results for a mapping of single and double bump topography to a rectangular domain are shown and discussed.

In Chapter 3, we compare SMG to Krylov methods for solitary wave computations. We use the same methodology, which utilise Fourier-Fourier collocation methods, and solve the same solitary wave problems investigated by Yang [115]. These include variants of the 2D NLS and the KP equation. Performance comparisons between SMG and Krylov methods are then documented.

Chapter 4 explores solitary wave and hydraulic fall solutions to the fKdV equation and serves as a one-dimensional test case for SMG. We focus initially on algebraic and exponentially decaying solitary wave solutions for a one bump topographic forcing. We then extend this to two bumps to study tabletop solutions, which are the symmetric version of asymmetric hydraulic fall solutions. Numerical continuation is used to extend the parameter space obtained by Ee and Clarke [34] for hydraulic fall solutions. Lastly, we connect algebraic decaying critical flow solutions obtained by Keeler et al. [67] to exponential decaying solitary wave ones by numerical continuation.

We extend our work to two-dimensions in Chapter 5, where we find solutions to the DJL equation with constant background current under the Boussinesq approximation. Perturbation methods, which utilise the fKdV solutions obtained in Chapter 4, are used

to generate the initial guess for Newton's method. One bump fKdV solitary wave and fKdV tabletop solutions are used in the initial guess for the one and two bump forced DJL equation respectively. We begin by investigating DJL solutions with no topographic forcing for a chosen set of stratifications. We then look for forced DJL solutions with a one and two bump topographic forcing. The numerical conformal mapping method in Chapter 2 is used to map the non-flat bottom domain to a rectangular one before spectral methods are used. Lastly we compare the WNL and DJL solutions to analyse the accuracy of WNL theory.

Chapter 2

Numerical Methods

Numerical methods are commonly used for problems with no known analytic solution. This is particularly useful for nonlinear differential equations. The first step often requires a suitable discretisation.

Numerical partial differential solvers come in two broad categories: local and global. The former calculates derivatives at a given point from its local neighbouring points. Examples of this include finite difference (FD) and finite element (FE) methods. The latter calculates derivatives at a given point from every point in the entire domain. Examples of this are spectral methods.

For sufficiently smooth problem and appropriate choice of basis functions, spectral methods exhibit spectral converge, that is, for a grid discretisation with N points, the error decrease as $O(c^N)$ where $0 < c < 1$. This is superior to local methods, which give algebraic convergence; error decreases as $O(N^{-\alpha})$ for some constant α. Therefore, to achieve the same desired accuracy, fewer grid points are required.

There are three types of spectral methods: Galerkin, Tau and collocation. Galerkin and Tau use the weak while collocation use the strong form of the differential equation [19]. Both are equivalent at the continuous level but are not after discretisation. We focus on collocation or pseudospectral methods throughout the rest of this thesis.

2.1 Spectral Collocation Methods

Spectral methods make use of global basis functions. The expansion of a given function u can be expressed as an infinite summation of orthogonal basis functions ϕ

$$u = \sum_{n=0}^{\infty} \hat{u}_n \phi_n(x). \tag{2.1}$$

Here $\hat{u}_n$ are the amplitude of each basis function ϕ_n, which are also known as the coefficients in spectral space of each eigenmode n. The basis functions are chosen so that

- $|\hat{u}_n|$ decay exponentially as $n \to \infty$, resulting in spectral convergence.
- It is simple to calculate the derivative in spectral space, that is for a given function u and corresponding $\hat{u}_n$, the coefficients for $\hat{u}'_n$ for $\frac{du}{dx}$ are found efficiently from $\hat{u}_n$.
- The spectral transform from u to $\hat{u}_n$ and vice versa, that is the conversion between real and spectral space, is computationally efficient.

Care must be taken when choosing appropriate basis functions to achieve spectral accuracy. Failing to do so will result in algebraic convergence, defeating the purpose for using spectral methods. The choice of appropriate basis functions depends on the computational domain and the boundary conditions of the problem.

Two commonly used spectral methods are Fourier and Chebyshev; the former use complex exponentials, while the latter use Chebyshev polynomials as the basis functions. When applied in a appropriate setting, spectral convergence is obtained. For Fourier, this occurs when the problem is periodic whereas in Chebyshev, a finite domain with fixed boundary conditions is required.

Both Fourier and Chebyshev spectral methods benefit from the use of the Fast Fourier transform (FFT) [26, 43]. The former utilise FFT while the latter utilise Fast Cosine transform (FCT) to perform fast differentiation and spectral transform. For N nodes, this reduces the operation count for matrix vector multiplication and conversion between real and spectral space, from $O(N^2)$ to $O(N \log_2 N)$.

Collocation methods require that the approximate solution satisfy the differential equation at specific *collocation points*. While the three types of spectral methods: Galerkin, Tau and collocation work equally well for linear problems, the latter is easier to apply for nonlinear ones.

Throughout the rest of the thesis we shall use the notation $\tilde{u}$ and $\hat{u}$ to denote the Fourier transform and Chebyshev transform of u respectively.

2.1.1 Fourier Collocation points

For periodic problems, the choice of Fourier basis functions is obvious. Spectral accuracy is achieved if the function is C^∞ and periodic. Boundary conditions are automatically satisfied by the choice of basis functions and do not need to be explicitly implemented.

We consider a spectral approximation using a truncated Fourier series. Let $\mathbf{x} = (x_0, x_1 \dots x_{N-1})$ be the discretisation in the domain $[0, 2\pi)$. The Fourier collocation points are N evenly spaced points given by

$$x_j = \frac{2\pi j}{N}, \quad j = 0, 1, \dots, N-1. \tag{2.2}$$

Let $\mathbf{u} = (u_0, u_1, \dots, u_{N-1})$ be the spectral approximation of u at the collocation points, where $u_j \simeq u(x_j)$ and let $\tilde{\mathbf{u}} = (\tilde{u}_{-N/2}, \tilde{u}_{-N/2+1}, \dots, \tilde{u}_{N/2-1})$ be the discrete Fourier Transform (DFT) of $\mathbf{u}$. Now let p be the wave numbers associated with the Fourier modes. The DFT gives the discrete Fourier coefficients

$$\tilde{u}_j = \frac{1}{N} \sum_{j=0}^{N-1} u_j e^{-ipx_j}, \quad p = -N/2, -N/2+1, \dots, N/2-1. \tag{2.3}$$

The inverse transform is then given by

$$u_j = \sum_{p=-N/2}^{N/2-1} \tilde{u}_p e^{ipx_j}, \quad j = 0, 1 \dots, N-1. \tag{2.4}$$

The DFT can also be written as a matrix operator given by

$$(C)_{pj} = \frac{1}{N} e^{-ipx_j}. \tag{2.5}$$

Similarly, the operator matrix for the inverse DFT is given by

$$(C^{-1})_{jp} = e^{ipx_j}. \tag{2.6}$$

Explicit matrix representations of the DFT and inverse DFT pair, while useful in theory, are rarely used in practice. This is because the operation count for multiplying a full $N \times N$ size matrix with a vector of length N is $O(N^2)$. Instead, the FFT algorithm is used in practice to perform the multiplication, reducing the operation count to $O(N \log_2 N)$.

2.1.2 Fourier Differentiation

In Fourier space the derivative of $\tilde{u}_j$ is easy to compute. Let Du denote the approximate spectral derivative of u then the derivative is simply given by

$$(Du)_j = \frac{1}{N} \sum_{p=-N/2+1}^{N/2-1} ip\tilde{u}_j e^{ipx_j}. \tag{2.7}$$

Note the mode $j = -N/2$ is removed in the summation. Given u is a real function, the term $\tilde{u}_{-N/2}$ contains a purely imaginary component and thus is set to 0. This is referred to as the filtered method [121].

The derivative can also be expressed in closed form using a Fourier differentiation matrix. This is a Toeplitz matrix with elements [19]:

$$(D)_{jk} = \begin{cases} 0 & \text{if} \quad j = k, \\ \frac{1}{2}(-1)^{j+k} \cot\left(\frac{(j-k)\pi}{N}\right) & \text{if} \quad j \neq k. \end{cases} \tag{2.8}$$

Similarly, second and higher order derivatives can be written in closed form [19].

Similar to the DFT matrix, differentiation matrices are rarely used in practice due to the inefficient matrix-vector multiplication required. Instead, transform methods utilizing FFTs are used in practice to perform fast differentiation.

Consider a simple one-dimensional model problem on $[0, 2\pi]$ with periodic boundary conditions given by

$$\frac{d^2u}{dx^2} = f. \tag{2.9}$$

The pseudospectral approximation with N nodes may be represented by the matrix system

$$A\mathbf{u} = \mathbf{f}, \tag{2.10}$$

where $\mathbf{u}$ and $\mathbf{f}$ are the vectors of unknowns and the right-hand side function f, respectively, at the collocation points,

$$\mathbf{u} = (u_0, u_1, \ldots, u_{N-1}),$$
$$\mathbf{f} = (f_0, f_1, \ldots, f_{N-1}),$$

and A is the second-order Fourier spectral derivative matrix. Let D be the diagonal matrix representing the nth-order differentiation in Fourier space given by

$$D^n_{pq} = (ip)^n \delta_{pq}. \tag{2.11}$$

The matrix A can then be expressed as

$$A = C^{-1} D^2 C. \tag{2.12}$$

While A appears as a matrix, no explicit matrix multiplication is performed in practice. Furthermore, memory requirements are lessened since $A\mathbf{u}$ is calculated through the use of transform operators without the need to explicitly store the matrix elements of A. In practice

$$A\mathbf{u} = \mathcal{F}^{-1}\{(ik)^2 \mathcal{F}\{\mathbf{u}\}\} \tag{2.13}$$

is used, where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the DFT and inverse DFT respectively, performed using FFTs.

2.1.3 Fourier Integration

Numerical integration can be performed using Gaussian quadrature methods given by

$$\int_a^b f(x)dx \approx \sum_{i=1}^{N} w_i f(x_i), \tag{2.14}$$

for appropriate weights w_i evaluated at nodes x_i.

For periodic and smooth functions, the quadrature points are the Fourier collocation nodes and the weights are simply the distance between subsequent Fourier nodes. This gives the well known trapezoidal rule

$$\int_a^b f(x)dx \approx \Delta x \sum_{i=1}^{N} f(x_i). \tag{2.15}$$

The trapezoidal rule has errors $O(1/N^2)$ for non-periodic functions, where N is the number of grid points. However, for a periodic smooth integrand $f(x)$, and the integration interval having the same length as its spatial period P, the trapezoidal rule becomes spectrally accurate [11].

Similar spectral accuracy can be obtained for a function $f(x)$ that is not periodic but decays exponentially quickly as $|x| \to \infty$ in an infinite domain. This is done by applying the trapezoidal rule on a large finite truncated domain $[-P, P]$ [11]

$$\int_{-\infty}^{\infty} f(x)dx \approx \int_{-P}^{P} f(x)dx \approx \Delta x \sum_{i=1}^{N} f(x_i). \tag{2.16}$$

2.1.4 Fourier pseudospectral method and dealiasing

Nonlinear terms in wave equations are often quadratic or cubic in nature. Consequently the spectral transform of a multiplication requires the evaluation of a convolution.

Here we consider the case of a quadratic type nonlinearity and let $w = uv$. To calculate the Fourier transform $\tilde{w}$ directly from $\tilde{u}$ and $\tilde{v}$ would require a convolution given by $\tilde{w} = \tilde{u} * \tilde{v}$. Direct computation of this takes $O(N^2)$ operations. To avoid this, an alternative is to evaluate this multiplication is by using a pseudospectral method

$$\tilde{\mathbf{w}} = \mathcal{F}\{\mathcal{F}^{-1}\{\tilde{\mathbf{u}}\}\mathcal{F}^{-1}\{\tilde{\mathbf{v}}\}\}. \tag{2.17}$$

Multiplications are easy to evaluate point-wise in a pseudospectral method, but more difficult with Galerkin or Tau due to requiring series multiplications. However, care must be taken due to aliasing effects, as first discovered by Orszag and Patterson [86].

We consider the aliasing effects of a multiplication of two functions. Let $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{s} = \mathbf{uv}$ be functions expanded in Fourier modes

$$\mathbf{u} = \sum_{p=-N/2}^{N/2-1} \tilde{u}_p e^{ipx_j}, \tag{2.18}$$

$$\mathbf{v} = \sum_{p=-N/2}^{N/2-1} \tilde{v}_p e^{ipx_j}, \tag{2.19}$$

$$\mathbf{s} = \sum_{p=-N/2}^{N/2-1} \tilde{s}_p e^{ipx_j}, \tag{2.20}$$

Now let $\tilde{\mathbf{S}} = (\tilde{S}_0, \tilde{S}_1, \ldots, \tilde{S}_{N-1})$ denote the discrete Fourier coefficients obtained using a pseudospectral method, and $\tilde{\mathbf{s}} = (\tilde{s}_0, \tilde{s}_1, \ldots, \tilde{s}_{N-1})$ to denote the true discrete Fourier coefficients of $\mathbf{s}$. We have

$$\begin{aligned} \tilde{S}_k &= \sum_{m+n=k} \tilde{u}_m \tilde{v}_n + \sum_{m+n=k\pm N} \tilde{u}_m \tilde{v}_n \\ &= \tilde{s}_k + \sum_{m+n=k\pm N} \tilde{u}_m \tilde{v}_n. \end{aligned} \tag{2.21}$$

On a finite grid of N points, Fourier components can only resolve wave numbers up to N. Higher modes $p > N/2$ are aliased into the low modes $p - N$. The resulting additional term is the aliasing error given by the second term on the right-hand side of equation (2.21).

Phillips [90] first showed simulations that exhibited nonlinear computational instability due to Fourier aliasing. This was further experienced by Bowman [9], where viscosity damping was required to combat the aliasing instability.

There are many methods for the removal of the aliasing error term, also known as dealiasing. Examples include zero padding [84], phase-shift [86] and implicit [10]. A comparison of these in large-eddy simulation of neutrally stratified atmospheric flows is shown by Margairaz et al. [80]. We only consider the simple zero padding known as the 3/2 rule [84] for quadratic type nonlinear terms. This involves padding the initial N grid points to $3N/2$ points before multiplication and removing the extra padded points. This however requires $\sim 50\%$ more operations, since the FFT computation now requires $O(3N/2 \log_2(3N/2))$ operations.

2.1.5 Chebyshev Collocation points

In the case of fixed boundary conditions, the basis function of choice is the Chebyshev polynomials of the first kind. These are the eigenfunctions of the singular Sturm–Liouville differential equation

$$\frac{d}{dx}\left(\sqrt{1-x^2}\frac{dT_n}{dx}(x)\right)+\frac{n^2}{\sqrt{1-x^2}}T_n(x)=0,\quad n=0,1,2\ldots, \tag{2.22}$$

also known as the Chebyshev differential equation. The eigenfunction solutions given by the Chebyshev polynomials $T_n(x)$ have been well studied [18]. An alternative expression for $T_n(x)$ after normalising so that $T_n(1)=1$, is given by

$$T_n(x)=\cos(n\cos^{-1}x),\quad x\in[-1,1]. \tag{2.23}$$

The trigonometric relation $\cos(n+1)\theta+\cos(n-1)\theta=2\cos\theta\cos n\theta$ gives the recursion relation

$$T_{n+1}(x)=2xT_n(x)-T_{n-1}(x),\quad n\geq 1, \tag{2.24}$$

with $T_0(x)=1$ and $T_1(x)=x$. Other useful properties of the Chebyshev polynomials which will be utilised for boundary conditions are

$$T_n(\pm1)=(\pm1)^n, \tag{2.25}$$

$$T_n'(\pm1)=(\pm1)^{n+1}n^2. \tag{2.26}$$

The orthogonality condition for Sturm–Liouville eigenfunctions relative to a weighted inner product gives the expression

$$\int_{-1}^{1}\frac{T_n(x)T_m(x)}{\sqrt{1-x^2}}dx=0,\quad \text{if}\quad n\neq m \tag{2.27}$$

and

$$\int_{-1}^{1}\frac{T_n^2(x)}{\sqrt{1-x^2}}dx=c_n\frac{\pi}{2}, \tag{2.28}$$

where

$$c_n=\begin{cases}2, & n=0,\\ 1, & n\geq 1.\end{cases} \tag{2.29}$$

The Chebyshev expansion of a function u is given by

$$u(x)=\sum_{n=0}^{\infty}\hat{u}_nT_n(x), \tag{2.30}$$

where $\hat{u}$ are the Chebyshev coefficients, or the Chebyshev transform of u.

Let $\mathbf{u} = (u_0, u_1, \ldots, u_{N-1})$ be the Chebyshev spectral approximation of u at the Chebyshev collocation points, where $u_j \simeq u(x_j)$. For the purposes of numerical schemes, we consider the summation truncated to N modes, evaluated at the collocation points x_n so that

$$\mathbf{u} = \sum_{n=0}^{N} \hat{u}_n T_n(x). \tag{2.31}$$

There are three forms of Chebyshev nodes or quadrature points with each having slight differences with regards to differentiation and integration. These are the *Chebyshev Gauss*, *Chebyshev Gauss–Radau* and the *Chebyshev Gauss–Lobatto* nodes.

For our work, we focus on the Chebyshev Gauss–Lobatto nodes as the collocation points, with their quadrature points and weights defined by

$$x_j = \cos\frac{\pi j}{N}, \quad w_j = \begin{cases} \dfrac{\pi}{2N}, & j = 0, N, \\ \dfrac{\pi}{N}, & j = 1, \ldots, N-1 \end{cases} \tag{2.32}$$

Boundary conditions (2.25) and (2.26) then give the properties

$$\mathbf{u}_0 = \sum_{n=0}^{N} \hat{u}_n, \qquad \mathbf{u}_N = \sum_{n=0}^{N} (-1)^n \hat{u}_n \tag{2.33}$$

and

$$\mathbf{u}'_0 = \sum_{n=0}^{N} n^2 \hat{u}_n, \qquad \mathbf{u}'_N = \sum_{n=0}^{N} (-1)^n n^2 \hat{u}_n, \tag{2.34}$$

where $\mathbf{u}'_0$ is the spectral approximation of $\frac{du}{dx}$ at x_0 and $\mathbf{u}'_N$ is the spectral approximation of $\frac{du}{dx}$ at x_N.

Let $\hat{\mathbf{u}} = (\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_N)$ be the discrete Chebyshev transform of $\mathbf{u}$ and let k be the Chebyshev wave numbers. The discrete Chebyshev transform based on the Chebyshev Gauss–Lobatto nodes is given by

$$\hat{u}_k = \frac{2}{N\bar{c}_k} \sum_{j=0}^{N} \frac{1}{\bar{c}_j} u_j \cos\frac{\pi j k}{N}, \quad k = 0, 1, \ldots, N, \tag{2.35}$$

and the inverse transform by

$$u_j = \sum_{k=0}^{N} \hat{u}_k \cos\frac{\pi j k}{N}, \quad j = 0, 1, \ldots, N, \tag{2.36}$$

where

$$\bar{c}_j = \begin{cases} 2, & j = 0, N, \\ 1, & j = 1, \ldots, N-1. \end{cases} \tag{2.37}$$

Similar to Fourier transforms, the Chebyshev transform can be written as a matrix operator given by

$$(C)_{kj} = \frac{2}{N\bar{c}_j\bar{c}_k}\cos\frac{\pi jk}{N} \tag{2.38}$$

Similarly, the operator matrix for the inverse Chebyshev transform matrix is given by

$$(C^{-1})_{jk} = \cos\frac{\pi jk}{N}. \tag{2.39}$$

Note that the Chebyshev transforms are simply a weighted cosine transform, and thus Fast Cosine Transforms (FCT) can be used. An algorithm for performing a Chebyshev transform using FFT for a given vector $\mathbf{v} = (v_0, \ldots, v_N)$ at the Chebyshev nodes is given in Algorithm 1.

Algorithm 1 Algorithm for the Chebyshev transform using FFT

1: Set $\mathbf{V}$ to be an even periodic extension of $\mathbf{v}$, obtained by concatenating $\mathbf{v}$ with $\overline{\mathbf{v}}$, where $\overline{\mathbf{v}} = (v_{N-1}, \ldots, v_1)$. That is let $\mathbf{V} = [\mathbf{v}|\overline{\mathbf{v}}]$. Note that $\mathbf{V}$ has length $2N-2$.
2: Set $\mathbf{w} = \tilde{\mathbf{V}}/(N-1)$ where $\tilde{\mathbf{V}}$ is the FFT of $\mathbf{V}$.
3: Halve the first and last elements of $\mathbf{w}$.
4: The Chebyshev transform of $\mathbf{v}$ is then given by the vector $\hat{\mathbf{v}} = (w_1/2, w_2, \ldots, w_{N-1}, w_N/2)$.

Similarly, inverse Chebyshev transform on a vector $\hat{\mathbf{v}}$ can be performed using Algorithm 2.

Algorithm 2 Algorithm for the inverse Chebyshev transform using FFT

1: Set $\tilde{\mathbf{w}} = (N-1)\,(2\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_{N-1}, 2\hat{v}_N)$.
2: Set $\tilde{\mathbf{W}}$ to be an even periodic extension of $\tilde{\mathbf{w}}$ by concatenating $\tilde{\mathbf{w}}$ with $\overline{\tilde{\mathbf{w}}}$ so that $\tilde{\mathbf{W}} = [\tilde{\mathbf{w}}|\overline{\tilde{\mathbf{w}}}]$.
3: Set $\mathbf{V} = \mathcal{F}^{-1}\{\tilde{\mathbf{W}}\}$ where $\mathbf{V}$ is the inverse FFT of $\mathbf{W}$.
4: The inverse Chebyshev transform given by the first $N+1$ elements of $\mathbf{V}$, that is $\mathbf{v} = (V_0, \ldots, V_N)$.

Boyd [11] showed a comparison of CPU time between matrix multiplication and FFT for the Chebyshev transform. For small values of N ($N = 2^5 \sim 2^9$) matrix multiplication can outperform FFT. However for $N > 2^9$ the FFT becomes the more efficient method. Hence the matrix representation of Chebyshev transforms is rarely used in practice.

2.1.6 Chebyshev Differentiation

Similarly to Fourier differentiation matrices, Chebyshev ones are also available in closed form. The first derivative matrix was first published by Gottlieb, Hussaini and Orszag

[47] and is given by

$$(D)_{jk} = \begin{cases} \dfrac{c_j}{c_k}\dfrac{(-1)^{j+k}}{x_j - x_k} & \text{if} \quad j \neq k, \\ \dfrac{x_k}{2(1-x_k^2)} & \text{if} \quad 1 \leq j = k \leq N-1, \\ \dfrac{2N^2+1}{6} & \text{if} \quad j = k = 0, \\ -\dfrac{2N^2+1}{6} & \text{if} \quad j = k = N. \end{cases} \tag{2.40}$$

The closed form for second derivative matrix were found by Ehrenstein and Peyret [35] and Peyret [89]. Other forms of improved Chebyshev differentiation matrices which minimize round off error is described by Canuto et al. [19], Baltensperger and Berrut [5] and Schneider and Werner [99].

Alternatively, transform methods for Chebyshev differentiation utilise (2.24). Differentiating (2.24) gives a recurrence relation for the first derivative Chebyshev polynomials

$$2T_k(x) = \frac{1}{k+1}T'_{k+1}(x) - \frac{1}{k-1}T'_{k-1}(x), \quad k \geq 1. \tag{2.41}$$

Similarly differentiating (2.31) gives

$$u'(x) = \sum_{k=0}^{\infty} \hat{u}_k T'_k(x) = \sum_{k=0}^{\infty} \hat{u}_k^{(1)} T_k(x). \tag{2.42}$$

where $\hat{u}_k^{(1)}$ denotes the Chebyshev coefficients of the derivative of u. The expression (2.41) gives the resulting recurrence relation

$$2k\hat{u}_k = c_{k-1}\hat{u}_{k-1}^{(1)} - \hat{u}_{k+1}^{(1)}, \quad k \geq 1. \tag{2.43}$$

More generally, for higher derivatives of order q we have

$$2k\hat{u}_k^{(q-1)} = c_{k-1}\hat{u}_{k-1}^{(q)} - \hat{u}_{k+1}^{(q)}, \quad k \geq 1, \tag{2.44}$$

which can be used iteratively to find the higher order derivative Chebyshev coefficients.

Since the discrete Chebyshev expansion assumes $\hat{u}_k^{(q)} = 0$ for $k \geq N$, equation (2.44) provides an efficient way to calculate the nonzero coefficients in decreasing order.

For small values of N, differentiation through differentiation matrices can outperform transform methods. This is because the heaviest computation in the latter is the FFT or FCT itself, for which the speed comparison between matrix multiplication and FFT has

been discussed in the previous section.

2.1.7 Chebyshev Integration

Spectrally accurate integration at Chebyshev nodes can be performed using Gaussian quadrature and is given by the equation

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\,dx \approx \sum_{n=0}^{N} w_n f(x_n). \tag{2.45}$$

The quadrature weights and nodes for the Chebyshev Gauss–Lobatto points are given by Fornberg [42]

$$x_n = \cos\frac{\pi n}{N}, \quad \omega_n = \begin{cases} \dfrac{\pi}{2N} & \text{for } n = 0, N \\ \dfrac{\pi}{N} & \text{for } n = 1, \ldots, N-1. \end{cases} \tag{2.46}$$

This is exact for polynomials up to degree $2N-1$ [46].

Alternatively, integration at Chebyshev Gauss–Lobatto points can be performed using Fejér's rule [40] or Clenshaw–Curtis quadrature [25]. This has the advantage of omitting the weight function $\frac{1}{\sqrt{1-x^2}}$ in the integral and also allows the use of FFTs. However this is only exact for polynomials up to degree N.

Consider the integral at the Chebyshev Gauss–Lobatto points with the change of variables $x = \cos\theta$

$$I = \int_{-1}^{1} f(x)\,dx = \int_{0}^{\pi} f(\cos\theta)\sin\theta\,d\theta, \tag{2.47}$$

such that θ are N evenly spaced Fourier collocation points in $[0, \pi]$

$$\theta_j = \frac{\pi j}{N}, \quad j = 0, \ldots, N-1. \tag{2.48}$$

Here $f(\cos\theta)$ can be expanded as a discrete cosine series

$$f(\cos\theta) \approx \sum_{n=0}^{N} a_n \cos n\theta, \tag{2.49}$$

where a_n are the discrete cosine (or Chebyshev) coefficients. Evaluating the integral (2.47) with the substitution (2.49) gives

$$\sum_{n=0}^{N} a_n \int_{0}^{\pi} \cos n\theta \sin\theta\,d\theta \approx \sum_{k=0}^{N} \frac{2a_{2k}}{1-(2k)^2}. \tag{2.50}$$

Alternatively, equation (2.50) can be derived by expanding $f(x)$ in a Chebyshev series

$f(x) = \sum_{n=0}^{N} a_n T_n(x)$ and using the fact that

$$\int_{-1}^{1} T_n(x)\,dx = \begin{cases} \dfrac{2}{1-n^2} & \text{if } n \text{ is even,} \\ 0 & \text{if } n \text{ is odd.} \end{cases} \tag{2.51}$$

It follows that the quadrature weights

$$w_n = \begin{cases} \dfrac{2}{1-n^2} & \text{if } n \text{ is even,} \\ 0 & \text{if } n \text{ is odd,} \end{cases} \tag{2.52}$$

can be used to evaluate the integral

$$I = \int_{-1}^{1} f(x)\,dx = \sum_{i=0}^{N} w_n a_n, \tag{2.53}$$

where the Chebyshev coefficients a_n can be efficiently calculated using FFTs in $O(N \log N)$ operations, an improvement over Gaussian quadrature. As noted earlier the Clenshaw–Curtis method is exact for polynomials up to degree N.

It may appear that Gaussian quadrature is twice as accurate since Gaussian quadrature is exact for polynomials up to degree $2N - 1$ whereas Clenshaw–Curtis method is exact for polynomials up to degree N. However numerous examples have shown that Clenshaw–Curtis performs comparably with Gaussian quadrature [82, 37, 72]. Trefethen [109] compares Gaussian quadrature, albeit at Legendre nodes, with Clenshaw–Curtis and showed the comparable accuracy for a variety of integrands.

2.2 Exact Spectral Solvers

Discretisation of differential equations using local methods generally results in sparse matrices. Many efficient sparse matrix solvers are available. Classical references include the books by Greenbaum [49] and Saad [96]. Spectral methods, however, generally result in dense matrices. We explore several methods to tackle spectral solutions.

Spectral exact solvers fall into two main categories: direct solution of the dense spectral discretisation system and Ad Hoc methods. For direct methods, the spectral matrix is full in 1D, and block-wise full in higher dimensions, thus inversion is costly. For a problem in d dimensions, the cost of Gaussian elimination is $O(N^{3d})$ operations and $O(N^{2d})$ storage. LU decomposition and QR methods can reduce this to $O(N^{2d})$ operations while also requiring $O(N^{2d})$ memory to store every element of the spectral differentiation matrix. This is only acceptable for small N.

In Ad Hoc methods, exact solutions to the discretised matrix can be obtained via spectral transforms. Fourier and Chebyshev methods that can exploit the FFT can yield solutions in $O(N^d \log_2 N)$ operations and $O(N^d)$ storage since the full spectral matrix does not need to be stored. However, Ad Hoc methods are restricted to classes of constant coefficient Poisson or Helmholtz type problems.

2.2.1 Fourier Ad Hoc method

Fourier Ad Hoc methods are very simple to apply using FFTs. Consider a simple one-dimensional Helmholtz problem on $[0, 2\pi]$ with periodic boundary conditions,

$$\frac{d^2u}{dx^2} + \lambda u = f, \tag{2.54}$$

where λ is a constant, and $f = f(x)$.

Let C be the Fourier transform matrix, C^{-1} be the inverse Fourier transform matrix, and D be the differentiation matrix in Fourier space: a diagonal matrix with elements ik. This gives the solution $\mathbf{u} = (C^{-1}DC + \lambda I)^{-1}\mathbf{f}$.

In practice no matrix multiplication is performed or stored, instead they can be computed using FFTs. This is done by discretising (2.54) with Fourier collocation method with N points and taking the Fourier transform on both sides. Rearranging for $\tilde{u}_p$ yields

$$\tilde{u}_p = \frac{\tilde{f}_p}{-p^2 + \lambda}, \quad p = -N/2, -N/2 + 1, \ldots, N/2 - 1. \tag{2.55}$$

The solution $\mathbf{u}$ can then be recovered by taking the inverse DFT. The number of operations is $O(N \log N)$ to solve the equation. However if λ is not a constant but an arbitrary function of x, Fourier Ad Hoc method do not work since a convolution is required.

2.2.2 Chebyshev Tau Ad Hoc method

Similarly fast Chebyshev solvers are available for a class of Helmholtz type equations [19]. Consider the one-dimensional Helmholtz equation on $[-1, 1]$ with homogeneous Dirichlet conditions

$$\begin{aligned} \frac{d^2u}{dx^2} + \lambda u &= f \\ u(-1) = u(1) &= 0, \end{aligned} \tag{2.56}$$

where λ is a constant and $f = f(x)$.

The Chebyshev Tau approximation is then

$$\begin{aligned} \hat{u}_k^{(2)} + \lambda \hat{u}_k &= \hat{f}_k, \quad k = 0, 1, \ldots, N-2 \\ \sum_{k=0}^{N} \hat{u}_k = 0, &\quad \sum_{k=0}^{N} (-1)^k \hat{u}_k = 0. \end{aligned} \tag{2.57}$$

An efficient solution method can be obtained via using the recursion relation (2.44) for $q = 2$

$$2k\hat{u}_k^{(1)} = c_{k-1}\hat{u}_{k-1}^{(2)} - \hat{u}_{k+1}^{(1)}, \quad k \geq 1. \tag{2.58}$$

Substituting this into (2.57) gives after simplification

$$\begin{aligned} -\lambda \left(\frac{c_{k-2}}{4k(k-1)} \right) u_{k-2} + \left(1 + \frac{\lambda \beta_k}{2(k^2-1)} \right) u_k - \lambda \frac{\beta_{k+2}}{4k(k+1)} u_{k+2} \\ = -\frac{c_{k-2}}{4k(k-1)} f_{k-2} + \frac{\beta_k}{2(k^2-1)} f_k - \frac{\beta_{k+2}}{4k(k+1)} f_{k+2}, \quad k = 2, \ldots, N \end{aligned} \tag{2.59}$$

where

$$\beta_k = \begin{cases} 1, & 0 \leq k \leq N-2, \\ 0, & k > N-2. \end{cases} \tag{2.60}$$

We see that (2.59) are uncoupled for odd and even coefficients except the boundary conditions, which can be rewritten as

$$\sum_{\substack{k=0 \\ k \text{ even}}}^{N} \hat{u}_k = 0, \tag{2.61}$$

$$\sum_{\substack{k=1 \\ k \text{ odd}}}^{N} \hat{u}_k = 0. \tag{2.62}$$

This results in two linear quasi-tridiagonal matrices, one for the odd and one for the even coefficients. The former is given by (2.59) except with odd k values with the boundary condition (2.62). Similarly the latter is given by (2.59) except with even k values with the boundary condition (2.61). A small modification to the Thomas algorithm can yield improved efficiency, resulting in the cost of solving both systems being $16N$ operations [19].

2.2.3 Chebyshev-Fourier Ad Hoc method

Combining the ideas from the previous two sections on Fourier and Chebyshev Ad Hoc solvers, we obtain an efficient solver for the two-dimensional Fourier Chebyshev Helmholtz

problem

$$au_{xx} + bu_{yy} + \lambda u = f, \tag{2.63}$$

where a, b and λ are constants and $f = f(x, y)$.

Let x be the Fourier discretisation and y be the Chebyshev discretisation direction respectively. Taking the DFT of the spectral approximation of (2.63) gives

$$(\tilde{u}_{yy})_p + \frac{(\lambda - ap^2)}{b}\tilde{u}_p = \frac{\tilde{f}_p}{b}. \tag{2.64}$$

This is equivalent to the Helmholtz equation, which can be solved using the Chebyshev Tau technique in the previous section at each y value. In the case $\lambda = 0$, the zero-mean solution is taken by assigning $\tilde{u}_0 = 0$.

Since the $\tilde{\mathbf{u}}$ has imaginary components in general, care must be taken to not only take the real parts during any Chebyshev transform as one would if the solution is strictly real.

2.3 Iterative methods for Spectral Systems

For problems where exact Ad Hoc methods are not available, we turn to iterative methods. In general we are seeking iterative solvers for the dense linear system

$$A\mathbf{u} = \mathbf{f} \tag{2.65}$$

where $A \in \mathbb{R}^{N\times N}$, $\mathbf{u} \in \mathbb{R}^N$ and $\mathbf{f} \in \mathbb{R}^N$ resulting from the spectral discretisation of a general linear differential equation. We note that the first and last rows of A and f are reserved for boundary conditions if applicable. Since the spectral differentiation matrices are dense, A is generally also dense, thus classes of iterative solvers available are restricted. Iterative techniques such as Gauss–Seidel (GS) and Successive Over-Relaxation (SOR) cannot be used efficiently, rather Jacobi / Richardson iterations are preferred. This is because GS and SOR use a mix of both old and new values calculated at the same iteration and cannot utilise fast transform techniques. Richardson iterations on the other hand update all unknown values simultaneously which allows the use of FFT in the algorithm.

2.3.1 Richardson

One of the simplest iterative schemes is the Richardson iteration [94]. This takes the form

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \omega\mathbf{r}^{(k)}, \tag{2.66}$$

where $\mathbf{v}$ is the best estimate of the true solution $\mathbf{u}$, ω is the relaxation parameter, $\mathbf{r}$ is the residual given by

$$\mathbf{r}^{(k)} = \mathbf{f} - A\mathbf{v}^{(k)}, \tag{2.67}$$

and k is the iteration number.

The error in the new iteration can be related to the error in the previous iteration through

$$(\mathbf{v}^{(k+1)} - \mathbf{u}) = (I - \omega A)(\mathbf{v}^{(k)} - \mathbf{u}). \tag{2.68}$$

This implies that the iterative scheme converges if the spectral radius ρ of $(I - \omega A)$ is less than one. This is equivalent to

$$|I - \omega\lambda| < 1, \tag{2.69}$$

for all the eigenvalues λ of A. The convergence criteria is then $0 < \omega < \omega_{max}$ where

$$\omega_{max} = \frac{2}{\lambda_{max}}. \tag{2.70}$$

The simplest Richardson scheme is stationary Richardson (SR), where ω is fixed for all relaxation sweeps using the optimal value of ω which minimises ρ, given by

$$\omega = \frac{2}{\lambda_{max} + \lambda_{min}}, \tag{2.71}$$

where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of the matrix A. This choice of ω gives the spectral radius to be

$$\rho = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}} \simeq 1 - 2\frac{\lambda_{min}}{\lambda_{max}}, \quad \text{for} \quad \lambda_{min} \ll .\lambda_{max} \tag{2.72}$$

An improvement to the simple SR iteration is to allow the parameter ω to change between relaxations. One such relaxation is nonstationary Richardson (NSR) with the optimum parameters derived by Young [119] by means of first-order Chebyshev polynomials. A relaxation cycle, with the optimum parameters, cycles through a fixed number of κ parameters with

$$\omega_j = \frac{2}{(\lambda_{max} - \lambda_{min}) \cos\frac{(2j-1)\pi}{2\kappa} + \lambda_{max} + \lambda_{min}}, \quad j = 1, \ldots, \kappa. \tag{2.73}$$

For the case $\kappa = 1$ this simply reduces to the SR scheme. Zang et al. [122] suggested the use of $\kappa = 3$, as any higher value does not justify the cost.

The spectral radius of NSR is given by Canuto et al [19]

$$\rho = \frac{1}{\left[T_k\left(\frac{\mathcal{K}+1}{\mathcal{K}-1}\right)\right]^{1/\kappa}}, \tag{2.74}$$

where T_k are the Chebyshev polynomials and $\mathcal{K}$ is the condition number given by

$$\mathcal{K} = \frac{\lambda_{max}}{\lambda_{min}}. \tag{2.75}$$

However, both SR and NSR require the knowledge of the maximum and minimum eigenvalues, which are not generally known for an arbitrary linear differential equation. Estimates are available for mth-order differential equations [11] which give

$$\lambda_{max} \sim O(N^m)\lambda_{min}. \tag{2.76}$$

For a second-order differential equation, this implies that $O(N^2)$ iterations are required for convergence if the iterative scheme is used alone.

An alternative parameter free iterative scheme introduced by Zang et al. [121] is the Minimum Residual Richardson (MRR), which does not perform poorly with complex eigenvalues, unlike SR and NSR. The choice of relaxation parameter aims to minimize the residual with

$$\omega = \frac{(\mathbf{r}^{(k)} \cdot A\mathbf{r}^{(k)})}{(A\mathbf{r}^{(k)} \cdot A\mathbf{r}^{(k)})}, \tag{2.77}$$

where $(\cdot)$ denotes the dot product. Note that the only convergence criterion for Richardson type iterations for the chosen optimal ω, is that the eigenvalues lie strictly in the $Re(z) > 0$ of the complex plane [11, 36]. The algorithm for MRR for a chosen tolerance tol, given an initial guess $\mathbf{v}^{(0)}$ is given in Algorithm 3.

Algorithm 3 Algorithm for the Minimum Residual Richardson method

1: Set $\mathbf{v}^{(0)}$, $\mathbf{r}^{(0)} = \mathbf{f} - A\mathbf{v}^{(0)}$.
2: **for** $k = 1, \ldots, n$ **do**
3: **if** $|\mathbf{r}^{(k-1)}| < tol$ **then**
4: **return** $\mathbf{v}^{(k-1)}$
5: **end if**
6: $\omega = \frac{(\mathbf{r}^{(k-1)} \cdot A\mathbf{r}^{(k-1)})}{(A\mathbf{r}^{(k-1)} \cdot A\mathbf{r}^{(k-1)})}$
7: $\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} + \omega\mathbf{r}^{(k-1)}$
8: $\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - \omega\mathbf{r}^{(k-1)}$
9: **end for**
10: **return** $\mathbf{v}^{(n)}$

The only change in Algorithm 3 for SR and NSR is the calculation of ω in step 6. Canuto and Quarteroni [20] showed MRR is superior to SR and NSR methods when ap-

plied to their model one and two-dimensional Chebyshev spectral discretised differential equation.

To estimate the number of iterations required for convergence, we define the rate of convergence $\mathcal{R}$ to be

$$\mathcal{R} = -\log \rho \tag{2.78}$$

and the reciprocal of $\mathcal{R}$ to be $\mathcal{J}$, which measures the number of iterations required to reduce the error by a factor of e. For the Richardson method, the number of iterations scale as [19]

$$\mathcal{J} \simeq \frac{1}{2}\mathcal{K}. \tag{2.79}$$

There are three main methods to improve the convergence rate:

(1) Consider Krylov subspace methods;

(2) Utilise a preconditioner;

(3) Accelerate using multigrid.

In the first option, Krylov subspace methods scales more favourably with condition number than the Richardson class of methods, resulting in faster convergence. Alternatively, a preconditioner can be used to reduce the condition number of the operator matrix. Lastly, multigrid concepts can be used to enhance convergence. Note that any combination of the three methods can also be used in tandem. We will delve into each of the three methods more extensively in the upcoming sections.

2.3.2 Krylov subspace methods

An improvement to the class of Richardson iterations is Krylov subspace methods. These are used in algorithms to find solutions to high-dimensional linear algebra problems. Excellent resources on Krylov methods are detailed by Saad [96].

One of the more popular Krylov subspace methods is the conjugate-gradient (CG) method. It was originally developed by Hestene and Stiefel [64] to solve symmetric and positive definite sparse $N \times N$ linear systems exactly in N iterations. Subsequently, this was adopted as an iterative method. CG seeks to find the minimum of the function

$$\phi(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T A\mathbf{u} - \mathbf{u}^T\mathbf{b}, \tag{2.80}$$

which is equivalent to solving (2.65).

While Fourier spectral matrices are not sparse, the Helmholtz operator $-\frac{d^2}{dx^2} + \lambda$ when used with the Fourier differentiation matrix is symmetric, positive definite and diagonally

dominant. CG has been shown to be very effective in solving such Fourier spectral systems [115].

The conjugate gradient method for solving (2.65) is given in Algorithm 4.

Algorithm 4 Algorithm for the conjugate-gradient method

Set $\mathbf{v}^{(0)}$, $\mathbf{r}^{(0)} = \mathbf{f} - A\mathbf{v}^{(0)}$, $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$.
for $k = 1, \ldots, n$ **do**
  if $|\mathbf{r}^{(k-1)}| < tol$ **then**
    return $\mathbf{v}^{(k-1)}$
  end if
  $\alpha_{k-1} = \frac{(\mathbf{r}^{(k-1)} \cdot \mathbf{r}^{(k-1)})}{(\mathbf{p}^{(k-1)} \cdot A\mathbf{p}^{(k-1)})}$
  $\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} + \alpha_{k-1}\mathbf{p}^{(k-1)}$
  $\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - \alpha_{k-1}A\mathbf{p}^{(k-1)}$
  $\beta_{k-1} = \frac{(\mathbf{r}^{(k)} \cdot \mathbf{r}^{(k)})}{(\mathbf{r}^{(k-1)} \cdot \mathbf{r}^{(k-1)})}$
  $\mathbf{p}^{(k)} = \mathbf{r}^{(k)} + \beta_{k-1}\mathbf{p}^{(k-1)}$
end for
return $\mathbf{v}^{(n)}$

Saad [96] gives the convergence properties of the CG method to be

$$||\mathbf{u} - \mathbf{v}^{(k)}||_2 \leq 2\left[\frac{\sqrt{\mathcal{K}} - 1}{\sqrt{\mathcal{K}} + 1}\right]^k ||\mathbf{u} - \mathbf{v}^{(0)}||_2, \tag{2.81}$$

where $\mathbf{u}$ is the exact solution and $\mathbf{v}^{(k)}$ is the kth iteration estimate. This implies

$$\mathcal{J} \simeq \frac{1}{2}\sqrt{\mathcal{K}}, \tag{2.82}$$

which has a faster convergence than the Richardson class of methods.

In the case of non symmetric matrices such as Chebyshev spectral matrices, examples of Krylov methods of choice are the biconjugate-gradient stabilized method (BiCGSTAB) and generalised minimum residual method (GMRES). A comparison of preconditioned Krylov subspace methods was investigated by Ghai et al. [45], with GMRES being one of the most effective methods. Here we will focus on GMRES.

GMRES was developed by Saad and Schultz [97] and is the generalisation of the minimum residual method for non-symmetric matrices. It seeks to find the solution $\mathbf{v}_n$ that minimises the residual norm $\mathbf{r}_n = ||b - A\mathbf{v}_n||_2$. The algorithm for GMRES is given in Algorithm 5.

As shown by Saad [96], the convergence rate of GMRES is more complicated than

Algorithm 5 Algorithm for the GMRES method

Set $\mathbf{v}^{(0)}$, $\mathbf{r}^{(0)} = \mathbf{f} - A\mathbf{v}^{(0)}$, $\beta = ||\mathbf{r}^{(0)}||$, $\mathbf{v}^{(1)} = \mathbf{r}^{(0)}/\beta$.
for $j = 1, \ldots, n$ **do**
    $\mathbf{w}^{(j)} = A\mathbf{v}^{(j)}$
    for $i = 1, \ldots, j$ **do**
        $g_{ij} = (\mathbf{v}^{(i)})^T \mathbf{w}^{(j)}$
        $\mathbf{w}^{(j)} = \mathbf{w}^{(j)} - g_{ij}\mathbf{v}_i$
    end for
    $g_{j+1,j} = ||\mathbf{w}^{(j)}||$
    if $g_{j+1,j} = 0$ set $n = j$ **then**
        Goto (12)
    end if
    $\mathbf{v}^{(j+1)} = \mathbf{w}^{(j)}/g_{j+1,j}$
end for
$W_n = [\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}]$, $\hat{G}_n = \{g_{ij}\}$, $1 \leq j \leq n$, $\leq i \leq j+1$
Compute $\mathbf{q}^{(\mathbf{n})}$, the minimiser of $||\beta\mathbf{e}_1 - \hat{G}_n\mathbf{q}||$
Set $\mathbf{v}^{(n)} = \mathbf{v}^0 + W_n\mathbf{q}^{(n)}$

CG. In practice, GMRES performs well and scales similarly to CG.

Iterative methods such as the Richardson class and Krylov subspace are rarely used alone in practice but are generally used with a preconditioner. This has the benefit of reducing the condition number of the system, hence improving the convergence rate.

2.4 Preconditioning

In the previous section, we showed the convergence rate of iterative methods depends on the condition number of the matrix A. If it is high, it is often beneficial to instead seek the solution to the (left) preconditioned system

$$H^{-1}A\mathbf{x} = H^{-1}\mathbf{f}, \tag{2.83}$$

for some chosen preconditioning matrix H. We will focus on the case of left preconditioning in the rest of this thesis. For example, the preconditioned Richardson iteration has the form

$$H(\mathbf{v}^{(k+1)} - \mathbf{v}^{(k)}) = \omega\mathbf{r}^{(k)}. \tag{2.84}$$

Consider two extreme choices for the preconditioning matrix; $H = A$ and $H = I$. The former case results in $H^{-1}A$ reducing to the identity matrix I with a condition number of 1. Iterative methods would converge very rapidly in this case, however computing H^{-1} is identical in difficulty to solving the original system. In the latter extreme, $H = I$ results in the original system with no improvements. Thus the optimum choice lies somewhere between these two extremes. Specifically we require that the preconditioning matrix have the following properties

(1) H must be easily invertible;

(2) The condition number of $H^{-1}A$ is smaller than that of A; that is $H \approx A$ in some sense.

An obvious choice for the preconditioning matrix for the spectral discretisation matrix would be the FD approximation to the differential operator [83]. For second-order one-dimensional problems, the chosen preconditioning results in a tridiagonal system which can be solved using Thomas' algorithm in $O(N)$ operations. Higher dimension FD discretisation results in a sparse block tridiagonal matrix, which can be calculated from one-dimensional FD discretisation matrices through the use of the Kronecker tensor product.

Another benefit of preconditioning is the removal of negative eigenvalues, for a suitable choice of preconditioning matrix [11]. Negative eigenvalues are often seen in the Helmholtz and advection-diffusion problems. This poses a problem for many iterative schemes such as Richardson and conjugate-gradient type methods since their convergence criteria require that all eigenvalues are strictly positive.

Here we summarise the algorithms in the previous section while incorporating a preconditioner. The algorithm for the preconditioned minimum residual method (PMRR) with preconditioner H is given in Algorithm 6.

Algorithm 6 Algorithm for Preconditioned Minimum Residual Richardson method

Set $\mathbf{v}^{(0)}$, $\mathbf{r}^{(0)} = \mathbf{f} - A\mathbf{v}^{(0)}$, $\mathbf{z}^{(0)} = H^{-1}\mathbf{r}^{(0)}$.
for $k = 1, \ldots, n$ **do**
    if $|\mathbf{r}^{(k-1)}| < tol$ **then**
        return $\mathbf{v}^{(k-1)}$
    end if
    $\omega = \frac{(\mathbf{r}^{(k-1)} \cdot A\mathbf{z}^{(k-1)})}{(A\mathbf{z}^{(k-1)} \cdot A\mathbf{z}^{(k-1)})}$
    $\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} + \omega \mathbf{z}^{(k-1)}$
    $\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - \omega A\mathbf{z}^{(k-1)}$
    $\mathbf{z}^{(k)} = H^{-1}\mathbf{r}^{(k)}$
end for
return $\mathbf{v}^{(n)}$

Note that H^{-1} is never explicitly calculated, instead a sparse linear solver is used to solve the system $H\mathbf{z} = \mathbf{r}$.

Similarly, the algorithm for the preconditioned conjugate-gradient method (PCG) for a preconditioner H, is given in Algorithm 7.

Finally, the algorithm for the preconditioned GMRES with preconditioner H is given in Algorithm 8.

Algorithm 7 Algorithm for the Preconditioned conjugate-gradient method

Set $\mathbf{v}^{(0)}$, $\mathbf{r}^{(0)} = \mathbf{f} - A\mathbf{v}^{(0)}$, $\mathbf{z}^{(0)} = H^{-1}\mathbf{r}^{(0)}$, $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$.
for $k = 1, \ldots, n$ **do**
    if $|\mathbf{r}^{(k-1)}| < tol$ **then**
        return $\mathbf{v}^{(k-1)}$
    end if
    $\alpha_{k-1} = \frac{(\mathbf{r}^{(k-1)} \cdot \mathbf{r}^{(k-1)})}{(\mathbf{p}^{(k-1)} \cdot A\mathbf{p}^{(k-1)})}$
    $\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} + \alpha_{k-1}\mathbf{p}^{(k-1)}$
    $\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - \alpha_{k-1}A\mathbf{p}^{(k-1)}$
    $\beta_{k-1} = \frac{(\mathbf{r}^{(k)} \cdot \mathbf{r}^{(k)})}{(\mathbf{r}^{(k-1)} \cdot \mathbf{r}^{(k-1)})}$
    $\mathbf{p}^{(k)} = \mathbf{r}^{(k)} + \beta_{k-1}\mathbf{p}^{(k-1)}$
end for
return $\mathbf{v}^{(n)}$

Algorithm 8 Algorithm for the Preconditioned GMRES method

Set $\mathbf{v}^{(0)}$, $H\mathbf{r}^{(0)} = \mathbf{f} - A\mathbf{v}^{(0)}$, $\beta = ||\mathbf{r}^{(0)}||$, $\mathbf{v}^{(1)} = \mathbf{r}^{(0)}/\beta$.
for $j = 1, \ldots, n$ **do**
    Compute $H\mathbf{w}^{(j)} = A\mathbf{v}^{(j)}$
    for $i = 1, \ldots, j$ **do**
        $g_{ij} = (\mathbf{v}^{(i)})^T\mathbf{w}^{(j)}$
        $\mathbf{w}^{(j)} = \mathbf{w}^{(j)} - g_{ij}\mathbf{v}_i$
    end for
    $g_{j+1,j} = ||\mathbf{w}^{(j)}||$
    if $g_{j+1,j} = 0$ set $n = j$ **then**
        Goto (12)
    end if
    $\mathbf{v}^{(j+1)} = \mathbf{w}^{(j)}/g_{j+1,j}$
end for
$W_n = [\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}]$, $\hat{G}_n = \{g_{ij}\}$, $1 \leq j \leq n$, $\leq i \leq j+1$
Compute $\mathbf{q}^{(\mathbf{n})}$, the minimiser of $||\beta\mathbf{e}_1 - \hat{G}_n\mathbf{q}||$
Set $\mathbf{v}^{(n)} = \mathbf{v}^0 + W_n\mathbf{q}^{(n)}$

Next we discuss the FD preconditioners for both Fourier and Chebyshev collocation methods.

2.4.1 Fourier Collocation

Canuto et al. [19] explored the eigenvalues of some preconditioned Fourier spectral operators, namely, second and fourth-order FD as well as the strong and weak form of the FE preconditioning for a range of spectral operators. Here we will only focus on the second-order FD preconditioning.

As the Fourier collocation nodes are evenly spaced, we introduce the standard second-order FD approximation for uniform mesh. It is derived using a Taylor series and trun-

cating higher order terms. The second-order accurate centered FD approximations for the first and second derivatives are given by

$$\left(\frac{du}{dx}\right)_j \approx \frac{u_{j+1} - u_{j-1}}{2\Delta x} + O(\Delta x^2) \tag{2.85}$$

$$\left(\frac{d^2u}{dx^2}\right)_j \approx \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} + O(\Delta x^2) \tag{2.86}$$

for $j = 0, 1, \ldots, N-1$, where $u_j \simeq u(x_j)$, $x_j = j\Delta x$ and Δx is the uniform step size. Higher order derivatives and accuracy can be similarly obtained through Taylor series. The FD derivative approximations can then be used to generate the FD matrix approximation to the differential operator of interest.

Boyd [11] showed that for sufficiently large N, the WKB approximation [6] shows the eigenmodes of large eigenvalues λ must locally resemble a constant coefficient equation. This suggests that the eigenvalues are dominated by the second derivative term, unless the coefficients of such is $O(1/N)$ in comparison than the first derivative terms or $O(1/N^2)$ in comparison to the undifferentiated term. Hence we shall consider the differential operators investigated by Canuto et al. [19] that most resembles the Jacobian calculation within the Newton iteration for the fKdV and DJL equation under this approximation.

The second-order model problem with differential operator $\mathcal{L} = -d^2/dx^2$ and periodic boundary condition on $[0, 2\pi)$ is first considered. The periodic boundary condition results in the second-order FD matrix $H^{(fd2)}$ being cyclic tridiagonal. The eigenvalues of $H^{(fd2)}$ are given by

$$\lambda_p = \frac{4\sin^2\left(\dfrac{p\Delta x}{2}\right)}{(\Delta x)^2}, \tag{2.87}$$

where $p = -N/2 + 1, \ldots, N/2 - 1$ are the Fourier wave numbers. The preconditioned matrix $(H^{(fd2)})^{-1}A$ eigenvalues are then given by

$$\Lambda_p = \frac{(p\Delta x/2)^2}{\sin^2(p\Delta x/2)}. \tag{2.88}$$

From this we can conclude the eigenvalues of the preconditioned matrix $(H^{(fd2)})^{-1}A$ for the operator $\mathcal{L} = -d^2/dx^2$ with periodic boundary conditions are bounded between $[1, \pi^2/4]$. Note that the condition number of the preconditioned matrix is independent of N.

The second consideration of interest is the Helmholtz problem with the operator $\mathcal{L} =$

$-d^2/dx^2 - \lambda$ where $\lambda > 0$. Second-order FD preconditioning gives the results

$$\lambda_p = 4\frac{\sin^2\left(\frac{p\Delta x}{2}\right)}{\Delta x^2} - \lambda \tag{2.89}$$

$$\Lambda_p = (p^2 - \lambda)/\left(\frac{p^2\sin^2\left(\frac{p\Delta x}{2}\right)}{\left(\frac{p\Delta x}{2}\right)^2} - \lambda\right). \tag{2.90}$$

Second-order FD preconditioning likely leads to a mix of both positive and negative eigenvalues according to Canuto et al. [19], where a range of p could lead to $\Lambda_p < 0$.

2.4.2 Chebyshev Collocation

While preconditioners are optional in Fourier methods, they are a necessity for Chebyshev methods. This is because the condition number of Chebyshev differentiation matrix scales as $O(N^4)$ [19].

Canuto et al. [19] also explored the preconditioned eigenvalues for the operator $\mathcal{L} = -d^2/dx^2$ on $[-1, 1]$ with boundary conditions $u(-1) = u(1) = 0$. Second-order non-uniform FD and finite-element preconditioning in strong form were considered. Similar to the Fourier collocation preconditioning, we focus only on the second-order non-uniform FD preconditioning.

Since Chebyshev nodes are non-uniform, the FD approximation for non-uniform grids must be used. The second-order FD approximations for first and second derivatives respectively is given by

$$\left(\frac{du}{dx}\right)_j \approx \left(\frac{1}{\Delta x_j + \Delta x_{j+1}}\right)\left(\frac{\Delta x_j}{\Delta x_{j+1}}(u_{j+1} - u_j) + \frac{\Delta x_{j+1}}{\Delta x_j}(u_j - u_{j-1})\right), \tag{2.91}$$

$$\left(\frac{d^2u}{dx^2}\right)_j \approx \left(\frac{u_{j+1} - u_j}{\Delta x_{j+1}} - \frac{u_j - u_{j-1}}{\Delta x_j}\right)\left(\frac{2}{\Delta x_{j+1} + \Delta x_j}\right). \tag{2.92}$$

where $\Delta x_j = x_{j+1} - x_j$. These are similarly derived from the Taylor series on non-uniform grids. For Chebyshev collocation nodes we also have that $u_0 = 0$ and $u_N = 0$ with $x_j = \cos \pi j/N$.

The analytic eigenvalues of the preconditioned matrix $H^{-1}A$ are given by Haldenwang

et al. [52]

$$\Lambda_k = \frac{k(k-1)\sin^2\dfrac{\pi}{2N}\cos\dfrac{\pi}{2N}}{\sin\dfrac{(k-1)\pi}{2N}\sin\dfrac{\pi}{2N}}, \qquad k = 2, 3, \ldots, N, \tag{2.93}$$

where k are the Chebyshev wave numbers. From this we can see that

$$\Lambda_{min} = 1, \qquad \Lambda_{max} = N(N-1)\sin^2\frac{\pi}{2N} \tag{2.94}$$

where the upper bound is $\lim_{N\to\infty} \Lambda_{max} = \pi^2/4$, which is the same upper bound for the second-order FD preconditioned Fourier operator.

While the second-order model and Helmholtz problem are not strictly the problems we are working with, the eigenvalues of the preconditioned Jacobian during each Newton iteration for our problems of interest, the fKdV and DJL equations, can be approximated by the second-order model and Helmholtz problem. This is because the eigenvalues are dominated by the second-order derivative term [11].

2.5 Spectral Multigrid

Spectral multigrid (SMG) was initially proposed by Zang et al. [121, 122] and combines elements of spectral methods with multigrid concepts. We shall refer to the iterative scheme used within a multigrid setting as relaxations. While iterative schemes aim to solve the system, relaxations aim to remove the upper half frequency spectrum of the errors. In this section we shall denote the upper half frequencies as high frequencies. We will assume the reader is familiar with basic multigrid concepts and expand the ideas to SMG. Classical references on multigrid include the books by Briggs et al. [14], Hackbusch [51] and Trottenberg et al. [111].

There are several key components in a multigrid scheme, namely the type of discretisation, relaxation, and grid transfer operators. SMG also has the same key components, but differ in the methods used and available for each component. Details of each will be discussed in detail in the upcoming sections. Other components of multigrid that are the same as SMG include the coarse grid solver, iteration types such as the V-cycle, F-cycle, W-cycle and Full Multigrid (FMG), and the correction scheme for linear problems or Full Approximation Scheme (FAS) for nonlinear multigrid. The choice for our SMG application are the correction scheme, the V-cycle and FMG with preconditioned Krylov methods as the coarse grid solver.

2.5.1 Discretisation

Traditional multigrid typically uses FD or FE discretisation, which results in a sparse linear system. Instead, SMG uses spectral discretisation, resulting in a diagonally dominant, but full matrix. We focus on the commonly used spectral methods, namely Fourier and Chebyshev collocation methods as discussed in prior sections, due to the availability of fast transforms. These fast transforms will be utilised in many components of SMG.

Here we discuss the discretisation between nodes in a fine grid x_f with N_f grid points and nodes in a coarse grid x_c with N_c grid points. For Fourier collocation nodes we have the fine and coarse nodes given by

$$(x_f)_j = \frac{2\pi j}{N_f}, \quad j = 0, 1, \ldots, N_f - 1 \tag{2.95}$$

$$(x_c)_j = \frac{2\pi j}{N_c}, \quad j = 0, 1, \ldots, N_c - 1. \tag{2.96}$$

We also choose the ratio of the fine and coarse grid points to be $N_f = 2N_c$. Similarly, the Chebyshev Gauss–Lobatto collocation nodes are given by

$$(x_f)_j = \cos\frac{\pi j}{N_f}, \quad j = 0, 1, \ldots, N_f \tag{2.97}$$

$$(x_c)_j = \cos\frac{\pi j}{N_c}, \quad j = 0, 1, \ldots, N_c. \tag{2.98}$$

Due to the extra boundary point in Chebyshev nodes, we have that $N_f = 2N_c - 1$. Thus is it advantageous to choose the number of nodes to be $N = 2^m$ or $N = 2^m + 1$ for m grids used in SMG for Fourier and Chebyshev methods respectively. With this choice of ratio between fine and coarse grid, we only need to focus on relaxations that damp high frequency errors.

2.5.2 Relaxations

The choices of relaxation is restricted due to the dense properties of the spectral matrix. One commonly used in FD multigrid is the GS method, which utilises a mix of new and old values during each iterative step. GS is very efficient when applied to a sparse FD discretisation matrix, requiring $O(N)$ operations per iteration, and is very effective at damping of high frequency modes, making it an ideal choice of relaxation in FD multigrid. However, since spectral discretisation matrices are full, GS becomes extremely inefficient, requiring $O(N^2)$ operations for each iteration. Furthermore, since it is not possible to vectorise the process of new and old values during each iterative step, fast transform methods cannot be used.

Instead, we focus on Richardson iterations as relaxations in SMG, with small adjustments to the relaxation parameter ω. For use in a multigrid relaxation, it is only important that the high frequencies are being damped.

Zang [121] suggested replacing λ_{min} with λ_{mid} in both SR (2.71) and NSR (2.73) to obtain the optimal relaxation parameter for use in SMG. The optimal parameter in SR as a multigrid relaxation is then given by

$$\omega_{MG} = \frac{2}{\lambda_{max} + \lambda_{mid}}. \tag{2.99}$$

The resulting multigrid smoothing factor is then defined to be

$$\mu_{MG} = \frac{\lambda_{max} - \lambda_{mid}}{\lambda_{max} + \lambda_{mid}}, \tag{2.100}$$

which measures the damping rate of high frequency modes. This can be rewritten as

$$\mu_{MG} = \frac{\mathcal{K} - 1}{\mathcal{K} + 1}, \tag{2.101}$$

where the multigrid condition number is given by

$$\mathcal{K}_{MG} = \lambda_{max}/\lambda_{mid}. \tag{2.102}$$

When SR is applied to the second-order model Fourier problem (2.9), a smoothing factor of $\mu_{MG} = 0.6$ is achieved [121]. However, GS as a relaxation in FD multigrid tends to achieve $\mu_{MG} \simeq 0.1$. We explore alternative improvements to the SR relaxation.

In the residual smoothing method (RSM) introduced by Brandt et al. [13], the updating of $\mathbf{v}$ uses the residual of the surrounding neighbouring points. The weighted residual relaxation scheme in two dimensions is given by

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} - \omega \begin{bmatrix} \gamma & \beta & \gamma \\ \beta & \alpha & \beta \\ \gamma & \beta & \gamma \end{bmatrix} \mathbf{r}^{(k)}, \tag{2.103}$$

where α, β and γ are weights for the surrounding points. This is interpreted as

$$\begin{aligned} v_{ij}^{(k+1)} = v_{ij}^{(k)} - \omega[\alpha r_{ij}^{(k)} + \beta(r_{i-1,j}^{(k)} + r_{i+1,j}^{(k)} + r_{i,j-1}^{(k)} + r_{i,j+1}^{(k)}) \\ +\gamma(r_{i-1,j-1}^{(k)} + r_{i+1,j+1}^{(k)} + r_{i-1,j+1}^{(k)} + r_{i+1,j-1}^{(k)})], \end{aligned} \tag{2.104}$$

where i and j are the index for the collocation points (x_i, y_j). The analogous one-

dimensional version is given by

$$v_i^{(k+1)} = v_i^{(k)} - \omega[\beta r_{i-1}^{(k)} + \alpha r_i^{(k)} + \beta r_{i+1}^{(k)}]. \tag{2.105}$$

Extensions to high dimensions are straightforward. An example of a three-dimensional weighted residual is described by Canuto et al. [19].

The optimal weights for the surrounding points can be determined using a smoothing analysis through local mode analysis [12]. For only a single parameter α and $\beta = \gamma = 0$, this reduces to the Richardson scheme. When more parameters are involved, the optimal choices must be solved numerically. The optimal parameters for RSM applied to one, two-dimensional and three-dimensional self-adjoint Fourier problem were computed by Canuto et al. [19]. This allows for a smoothing rate of approximately $\mu_{MG} \simeq 0.1$ [19, 38] compared to $\mu_{MG} \simeq 0.6$ for NSR and $\mu_{MG} \simeq 0.7$ for SR.

Anisotropy

The two-dimensional problem

$$\mathcal{L} = a\frac{\partial^2}{\partial x^2} + b\frac{\partial^2}{\partial y^2}, \tag{2.106}$$

is defined as isotropic for constants a and b, if

$$\frac{a}{h_x^2} = \frac{b}{h_y^2} \tag{2.107}$$

where h_x and h_y are the step sizes in the respective directions.

RSM is shown to perform exceptionally when applied to self-adjoint, and isotropic problems. However, it performs poorly when applied to the Helmholtz problem or anisotropic problems [38]. This agrees with our experience when using RSM for the calculations of the Jacobian of the fKdV equation.

Anisotropy is a common problem when considering discretisation in higher dimensions. It occurs when the length scales of different dimensions vary considerably. This is especially critical in two-dimensional Fourier-Chebyshev discretisation.

In the case of anisotropic periodic problems in two-dimensions, Brandt [13] suggests the use of line relaxations. This is essentially a FD preconditioner applied to the spectral system, but instead of solving the FD matrix exactly, line relaxation is used on the preconditioner.

Line relaxation is a commonly used smoother in anisotropic FD multigrid schemes

[14]. It consists of first relaxing along lines of constant x then analogous sweeps along lines of constant y. This can be improved with high parallelism by relaxing all the odd lines first, then all the even lines. This scheme is called the alternating direction zebra (ADZ). However, line relaxation cannot be implemented efficiently on dense spectral matrices. Fortunately, it performs well as a preconditioner for Richardson relaxations. The relaxation parameter ω can be found through smoothing analysis shown as by Brandt [13].

Heinrich [54] verified the usefulness of ADZ as a preconditioner for SMG and extended it to non-periodic problems. Comparisons of smoothing rates of ADZ and variants of incomplete LU-decomposition (ILU) preconditioners in combination with SR, NSR and MRR were made for a range of two-dimensional Chebyshev-Chebyshev and Fourier-Chebyshev problems. Results show MRR with ADZ preconditioning being superior to other combinations. Furthermore, Heinrich [57] extended the idea to three-dimensional plane relaxations. Alternating plane relaxation (APR) was proposed where one step of APR consist of one step of (x, y), one step of (x, z), and one step of (y, z) plane relaxations, where each APR is preformed by 2D multigrid methods using ADZ.

A comparison of relaxation methods for SMG is shown by Canuto et al. [19]. RSM is found to be superior only for isotropic self-adjoint problems. For all other problems such as the Helmholtz or anisotropic problems, MRR with a FD preconditioner is most robust and will be our chosen method.

2.5.3 Grid transfer Operators

In general, the transfer operators used in multigrid must have comparable accuracy to the discretisation scheme. In FD multigrid with polynomial order accuracy, interpolation with the same order is commonly used. Therefore in SMG we utilise trigonometric and Chebyshev interpolation for Fourier and Chebyshev SMG respectively.

Given our choice of fine and coarse grid points given in (2.95) and (2.96) for Fourier and (2.97) and (2.98) for Chebyshev discretisation, it would appear restriction through injection is an obvious choice since the coarse grid points coincide with the fine grid points. However, Brandt [13] compared injection with Fourier interpolation and showed the latter gives slightly better results. Hence we shall take Fourier and Chebyshev interpolation as our restriction of choice.

For the Fourier restriction operator, interpolation can be efficiently performed using the FFT. Let $\mathbf{v}_f$ be a vector on the fine grid at the Fourier collocation nodes with N_f points. Now let the restriction of $\mathbf{v}_f$ be denoted as $\mathbf{v}_c$ with N_c points, that is $\mathbf{v}_c = R[\mathbf{v}_f]$. The algorithm for the Fourier restriction operator R is then given in Algorithm 9.

Algorithm 9 Algorithm for the Fourier restriction operator

1: Let $\tilde{\mathbf{v}}_f = \mathcal{F}\{\mathbf{v}_f\}$.
2: Define $\tilde{\mathbf{v}}_c = ((\tilde{v}_f)_1, (\tilde{v}_f)_2, \dots, (\tilde{v}_f)_{N_c})$.
3: Let $\mathbf{v}_c = \mathcal{F}^{-1}\{\tilde{\mathbf{v}}_c\}$.

Note the algorithm simply takes the Fourier transform, removes the upper half frequencies before taking the inverse.

Similarly, the prolongation operator involves taking the Fourier transform, padding the upper half frequencies with zeros then taking the inverse. The algorithm for the Fourier prolongation operator P such that $\mathbf{v}_f = P[\mathbf{v}_c]$ is given in Algorithm 10.

Algorithm 10 Algorithm for the Fourier prolongation operator

1: Let $\tilde{\mathbf{v}}_c = \mathcal{F}\{\mathbf{v}_c\}$.
2: Define $\tilde{\mathbf{v}}_f = ((\tilde{v}_c)_1, (\tilde{v}_c)_2, \dots, (\tilde{v}_c)_{N_c}, 0, \dots, 0)$ such that the length of $\mathbf{v}_f$ is N_f.
3: Let $\mathbf{v}_f = \mathcal{F}^{-1}\{\tilde{\mathbf{v}}_f\}$.

Algorithms 9 and 10 allows Fourier restriction and prolongation to be computed using two FFTs respectively. Explicit matrix representations of the operators is given by Zang et al. [122]

$$R_{jl} = \frac{1}{N_f} \sum_{p=-(N_f/4)+1}^{(N_f/4)-1} e^{2\pi i p(2j-l)/N_f}, \tag{2.108}$$

$$P_{jl} = \frac{2}{N_c} \sum_{p=-(N_c/4)+1}^{(N_c/4)-1} e^{2\pi i p(j-2l)/N_c}. \tag{2.109}$$

Similarly, Chebyshev restriction and prolongation operators in closed form is given by Zang et al. [122]

$$R_{jl} = \frac{2}{N_f \bar{c}_l} \sum_{p=0}^{N_f/2} \bar{c}_p^{-1} \cos\left(\frac{2\pi jp}{N_f}\right) \cos\left(\frac{\pi lp}{N_f}\right), \tag{2.110}$$

$$P_{jl} = \frac{4}{N_c \bar{c}_l} \sum_{p=0}^{N_c/2} \bar{c}_p^{-1} \cos\left(\frac{2\pi lp}{N_c}\right) \cos\left(\frac{\pi jl}{N_c}\right), \tag{2.111}$$

where c is defined in (2.29).

Zang noted that it is computationally beneficial to force R to be adjoint to P for residual transfers, as is widely used in FD multigrid. This gives rise to the new restriction operator

$$R_{jl} = \frac{2}{N_f \bar{c}_j} \sum_{p=0}^{N_f/2} \bar{c}_p^{-1} \cos\left(\frac{2\pi jp}{N_f}\right) \cos\left(\frac{\pi lp}{N_f}\right). \tag{2.112}$$

Note that any other type of grid transfer, such as for the coefficients of the differential equation, will use (2.110) instead.

The algorithm for Chebyshev restriction and prolongation operators is very similar to the Fourier case, both requiring a spectral transform along with removing or padding upper half frequencies. The main difference is managing of boundary conditions, and the adjustment in (2.112) to ensure the restriction and prolongation operators are adjoint. Let $\mathcal{C}$ and $\mathcal{C}^{-1}$ denote the Chebyshev transform and its inverse respectively, and $\mathbf{v}_f$ be a vector on the fine grid at the Chebyshev collocation nodes with N_f points. The Chebyshev restriction algorithm is then given in Algorithm 11.

Algorithm 11 Algorithm for the Chebyshev restriction operator

1: Let $\hat{\mathbf{v}}_f = \mathcal{C}\{\mathbf{v}_f\}$.
2: Apply adjoint operator adjustment $(\hat{v}_f)_{N_c} \leftarrow (\hat{v}_f)_{N_c}/2$.
3: Define $\hat{\mathbf{v}}_c = ((\hat{v}_f)_1, (\hat{v}_f)_2, \ldots, (\hat{v}_f)_{N_c})$.
4: Let $\mathbf{v}_c = \mathcal{C}^{-1}\{\hat{\mathbf{v}}_c\}$.
5: Apply boundary conditions $(v_c)_1 = (v_f)_1$ and $(v_c)_{N_c} = (v_f)_{N_f}$.

Note that step 2 in Algorithm 11 is omitted for grid transfers other than for the residual. Similarly, the Chebyshev prolongation operator algorithm is given in Algorithm 12.

Algorithm 12 Algorithm for the Chebyshev prolongation operator

1: Let $\hat{\mathbf{v}}_c = \mathcal{C}\{\mathbf{v}_c\}$.
2: Define $\hat{\mathbf{v}}_f = ((\hat{v}_c)_1, (\hat{v}_c)_2, \ldots, (\hat{v}_c)_{N_c}, 0, \ldots, 0)$ such that the length of $\mathbf{v}_f$ is N_f.
3: Let $\mathbf{v}_f = \mathcal{C}^{-1}\{\tilde{\mathbf{v}}_f\}$.
4: Apply boundary conditions $(v_f)_1 = (v_c)_1$ and $(v_f)_{N_f} = (v_c)_{N_c}$.

We shall denote the method in Algorithm 11 and 12 as *Dirichlet restriction* and *Dirichlet prolongation*, respectively, which can be performed efficiently using 2 FFTs since they both utilise the Chebyshev transform. Note that the boundary conditions in Algorithm 11 and 12 are of Dirichlet type. For Neumann or mixed boundary conditions, (2.34) is used.

For Dirichlet boundary conditions, the residual is generally not smooth near the boundaries since the ends are set to zero. This results in the Chebyshev transform in step 1 of the Algorithms 11 and 12, when applied to the residual, to exhibit Gibbs phenomenon near the Dirichlet boundary, introducing a source of high frequency modes. This can be mitigated by first calculating the residual values at the Dirichlet boundary as if a homogeneous Neumann boundary is present, before applying the Chebyshev restriction or prolongation algorithm.

Consider a vector $\mathbf{v}$ of length N at the Chebyshev collocation nodes $x_0, \ldots, x_N$, to be restricted or prolongated, where $v_0 = v_N = 0$. We begin by only considering the internal nodes $v_1, \ldots, v_{N-1}$ and look to solve for the new boundaries $\overline{v}_0$ and $\overline{v}_N$ assuming a homogeneous Neumann condition. To do this, we use the first and last rows of Chebyshev differentiation matrices [108] to obtain the equations

$$\begin{aligned} \mathbf{v}'(x_0) = 0 &\simeq \frac{2(N-1)^2+1}{6}\overline{v}_0 + \sum_{j=1}^{N-1} 2\frac{(-1)^j}{1-x_j}v_j + \frac{1}{2}(-1)^{N-1}\overline{v}_N, \\ \mathbf{v}'(x_N) = 0 &\simeq -\frac{1}{2}(-1)^{N-1}\overline{v}_0 + \sum_{j=1}^{N-1} -2\frac{(-1)^{N-1+j}}{1+x_j}v_j - \frac{2(N-1)^2+1}{6}\overline{v}_N. \end{aligned} \tag{2.113}$$

The resulting linear system can then be solved for $\overline{v}_0$ and $\overline{v}_N$. The new vector $\overline{\mathbf{v}} = (\overline{v}_0, v_1, \ldots, v_{N-1}, \overline{v}_N)$ is then used in the Chebyshev restriction or prolongation algorithm. We shall call this method the *Neumann restriction* and *Neumann prolongation*.

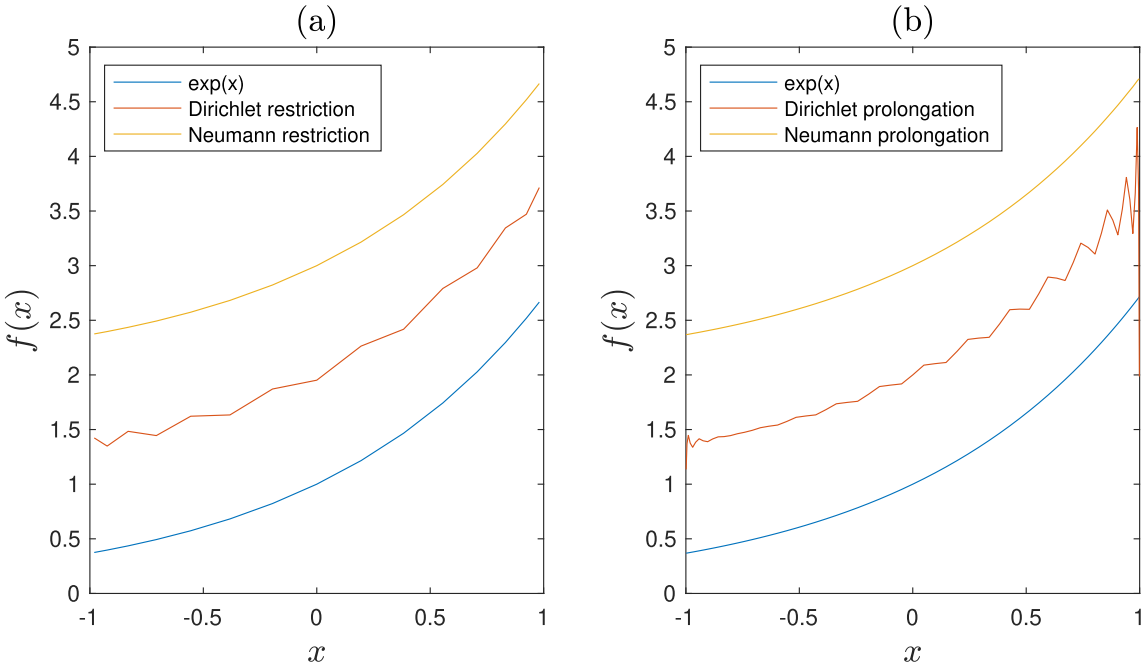

Figure 2.1: Comparison of Dirichlet and Neumann restriction (a), and Dirichlet and Neumann prolongation (b) to the original function $f(x) = e^x$, respectively. Only internal points are plotted and a vertical translation is used to separate plots for viewing.

Figure 2.1 shows the Gibbs phenomenon appearing due to the boundary discontinuity when using Dirichlet prolongation and restriction. However, Figure 2.2 shows Gibbs phenomenon only appearing for Dirichlet prolongation but not restriction. This suggests that the appearance of Gibbs phenomenon depends on the function being restricted or prolongated. We also see that the Neumann restriction and prolongation method removes Gibbs phenomenon. When applied to SMG, Neumann restriction and Neumann prolongation

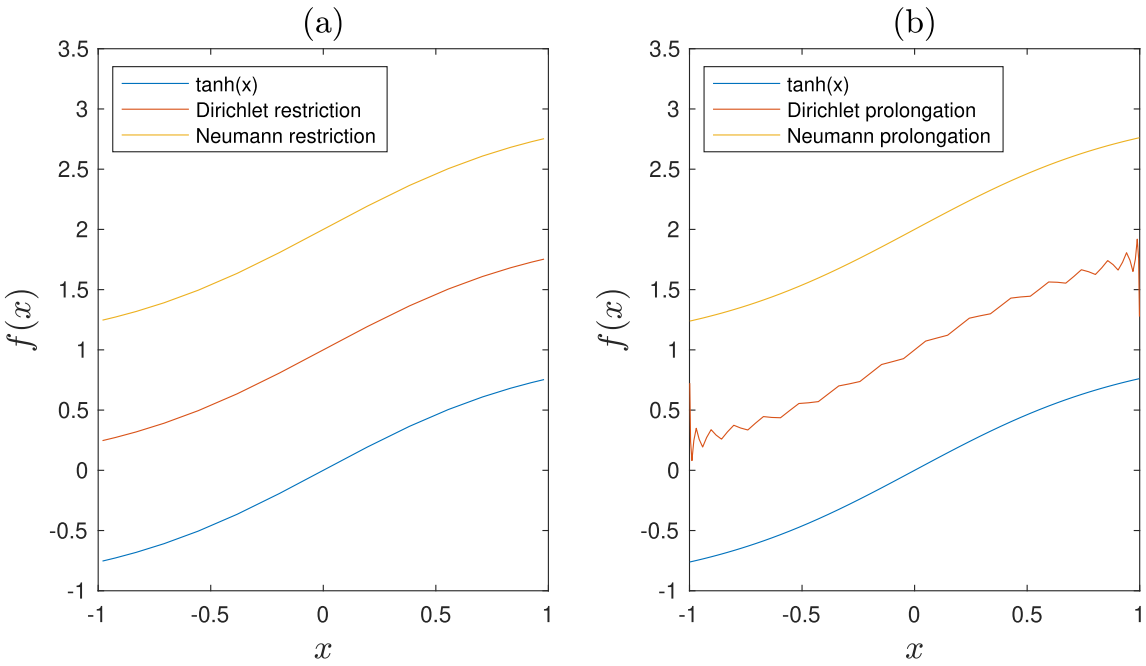

Figure 2.2: Comparison of Dirichlet and Neumann restriction (a), and Dirichlet and Neumann prolongation (b) to the original function $f(x) = \tanh x$, respectively. Only internal points are plotted and a vertical translation is used to separate plots for viewing.

result in a marginal improvement in SMG convergence.

2.6 Nonlinear solvers

The solvers mentioned in the prior sections are limited to linear problems. For nonlinear ones, Newton's method is most commonly employed. It iteratively solves for the correction to an initial guess of the solution until the required tolerance for the nonlinear residual is met. At each iteration, the correction requires the solution to a linear problem. In essence, Newton's method turns a solving a nonlinear problem into iteratively solving linear ones.

Consider a nonlinear function $f : \mathbb{R} \to \mathbb{R}$, for which we are interested in finding the roots of

$$f(x) = 0. \tag{2.114}$$

Assume we have a good initial guess to the solution of (2.114) at $x \approx x^{(0)}$. Using a Taylor series expansion we obtain

$$f(x) = f(x^{(0)}) + f'(x^{(0)})[x - x^{(0)}] + O([x - x^{(0)}]^2). \tag{2.115}$$

Now ignoring quadratic and higher terms, and rearranging gives the Newton–Raphson

method

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \qquad k = 0, 1, 2, \ldots \tag{2.116}$$

where k is the iteration number. In the case where the derivative $f'(x)$ is not readily known, an approximate gradient can be used in its place. This gives the secant method

$$x^{(k+1)} = x^{(k)} - f(x^{(k)}) \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})}. \tag{2.117}$$

Now generalising to the higher dimensional problem, and working with a differential operator $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \tag{2.118}$$

we obtain the expression for Newton's iteration

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - J(\mathbf{x}^{(k)})^{-1} \mathbf{f}(\mathbf{x}^{(k)}), \tag{2.119}$$

where J is the Jacobian matrix.

There are several practical difficulties when applying Newton's method. Firstly, obtaining a good initial guess, and secondly, the calculation of $J^{-1}(\mathbf{x}_n)$ at each iteration. We typically tackle the first problem using perturbation methods, where the leading or first-order perturbation solution is used as the initial guess. The second problem is minimised by applying Newton's method directly to the differential equation, also known as the Newton–Kantorovich method. In practice (2.119) is rarely used since the pseudospectral Jacobian matrix is a full matrix and would require $O(N^3)$ operations to determine its inverse.

To use the Newton–Kantorovich method, we require the notion of Fréchet derivatives [11], which is the generalisation of Taylor series to general nonlinear functions. The Fréchet differential of the operator $\mathcal{N}$ in the direction of Δ is defined by

$$\mathcal{N}_u \Delta = \lim_{\epsilon \to 0} \frac{\mathcal{N}(u + \epsilon\Delta(x)) - \mathcal{N}(u)}{\epsilon} = \left. \frac{\partial \mathcal{N}(u + \epsilon\Delta)}{\partial \epsilon} \right|_{\epsilon=0}. \tag{2.120}$$

This gives the linear term in the generalised Taylor expansion for any operator $\mathcal{N}(u)$

$$\mathcal{N}(u + \Delta) = \mathcal{N}(u) + \mathcal{N}_u(u)\Delta + O(\Delta)^2. \tag{2.121}$$

As an example, consider the second-order differential equation

$$u_{xx} = F(u(x)) \tag{2.122}$$

Assume we have a good initial estimate of the solution $u^{(k)}$ to the true solution $u(x)$. Now

expanding (2.122) in a Taylor series around $u^{(k)}$ gives

$$u_{xx} = F\left(u^{(k)}(x)\right) + F_u\left(u^{(k)}(x)\right)\left[u(x) - u^{(k)}(x)\right] + O\left(\left[u(x) - u^{(k)}(x)\right]^2\right) \tag{2.123}$$

where F_u is the derivative of the function $F(x, u)$ with respects to u. Ignoring higher order terms and rearranging (2.123) gives the iteration

$$u_{xx}^{(k+1)} - F_u\left(u^{(k)}(x)\right) u^{(k+1)} = F\left(u^{(k)}(x)\right) - F_u\left(u^{(k)}(x)\right) u^{(k)} \tag{2.124}$$

The iteration is equivalent to solving for the correction term Δ in

$$u^{(k+1)}(x) = u^{(k)}(x) + \Delta(x) \tag{2.125}$$

where $\Delta(x)$ is the solution to the linear differential equation

$$\Delta_{xx} - F_u\left(u^{(k)}(x)\right)\Delta = F\left(u^{(k)}(x)\right) - u_{xx}^{(k)}. \tag{2.126}$$

Notice the second term on the left hand side is the definition of the Fréchet differential, which can be calculated using equation (2.120).

2.7 Numerical Continuation

Numerical continuation techniques are useful to determine an approximate family of solutions to a system of nonlinear equations as a function of some parameter value. Numerical continuation applies to the equation

$$\mathbf{F}(\mathbf{u}, \lambda) = \mathbf{0} \tag{2.127}$$

for an arbitrary nonlinear function $\mathbf{F} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ and a parameter value $\lambda \in \mathbb{R}$. We seek to find the family of solutions to this equation as λ varies. In general $\mathbf{F}$ can be a system of nonlinear equations and λ can be a vector of parameters. Here we will only discuss the simple case where $\mathbf{F}$ is a single nonlinear equation and λ is a scalar. Note the vector case can be easily generalised.

One class of numerical continuation technique uses a predictor-corrector method [71]. They involve a number of important aspects to be carefully addressed:

(1) ensuring the efficiency of the predictor step,

(2) ensuring the efficiency of the corrector step,

(3) ensuring the efficiency of the step size adaptation,

(4) handling of turning points, bifurcations etc.

The predictor method of choice used is the tangent approximation. This involves linear extrapolation to the predicted solution. Higher order alternatives are also available, but are less common in practice. For our corrector approach, we employ Newton's method as the chosen nonlinear solver.

Step size control is critical in predictor-corrector methods. If it is too small, the method's efficiency would suffer along flat branches whereas if step size is too large Newton's method may not converge. There are several approaches to choosing step size, which can be based on asymptotic estimates, convergence of the corrector or on empirical performances [2, 101]. Here, we consider the basic step size control based empirically on the convergence of the Newton method.

In our work we only focus on the handling of folds or vertical turning points where the gradient $\frac{d\mathbf{u}}{d\lambda}$ approaches infinity. Various numerical continuation techniques exist, which parameterises the continuation curve in different ways. For problems without folds, natural parameter continuation is used. Pseudo arc-length continuation is used to handle cases where folds are present.

2.7.1 Natural parameter continuation

One of the simplest numerical continuation methods is the natural parameter continuation [48], which takes λ as the parameter. The method we use includes the tangent predictor and Newton corrector.

Suppose we have a solution $\mathbf{u}_0 = \mathbf{u}(\lambda_0)$ to equation (2.127), and some initial gradient vector $\dot{\mathbf{u}}_0 = \frac{d\mathbf{u}_0}{d\lambda}$. At continuation step ν, we look for the solution $\mathbf{u}_{\nu+1}$ at $\lambda_{\nu+1} = \lambda_\nu + \Delta\lambda_\nu$ for a chosen $\Delta\lambda_\nu$ by solving the nonlinear equation

$$\mathbf{F}(\mathbf{u}_{\nu+1}, \lambda_{\nu+1}) = \mathbf{0} \tag{2.128}$$

using a suitable nonlinear solver. We use a tangent predictor at step ν

$$\mathbf{u}_{\nu+1}^{(1)} = \mathbf{u}_\nu + \dot{\mathbf{u}}_\nu \Delta\lambda_\nu, \tag{2.129}$$

in combination with a Newton's method corrector. The resulting Newton corrector is then given by the iteration

$$\begin{aligned} \Delta\mathbf{F}_\mathbf{u}(\mathbf{u}_{\nu+1}^{(k)}, \lambda_{\nu+1})\mathbf{u}_{\nu+1}^{(k)} &= -\mathbf{F}(\mathbf{u}_{\nu+1}^{(k)}, \lambda_{\nu+1}) \\ \mathbf{u}_{\nu+1}^{(k+1)} &= \mathbf{u}_{\nu+1}^{(k)} + \Delta\mathbf{u}_{\nu+1}^{(k)}, \qquad k = 0, 1, 2, \ldots, \end{aligned} \tag{2.130}$$

where k is the iteration number and $\mathbf{F}_\mathbf{u}$ is the Jacobian.

Once Newton's method corrector converges to a required tolerance, a new direction vector $\dot{\mathbf{u}}$ is required for the next predictor step. We derive the expression for $\dot{\mathbf{u}}$ by first differentiating (2.127) with respect to the parameter λ which gives

$$\mathbf{F}_{\mathbf{u}}\frac{d\mathbf{u}}{d\lambda} + \mathbf{F}_{\lambda} = \mathbf{0}. \tag{2.131}$$

Evaluating this at $(\mathbf{u}_{\nu+1}, \lambda_{\nu+1})$ and rearranging for $\dot{\mathbf{u}}_{\nu+1}$ gives the expression for the new tangent predictor

$$\dot{\mathbf{u}}_{\nu+1} = -\mathbf{F}_{\mathbf{u}}^{-1}(\mathbf{u}_{\nu+1}, \lambda_{\nu+1})\mathbf{F}_{\lambda}(\mathbf{u}_{\nu+1}, \lambda_{\nu+1}). \tag{2.132}$$

It is advantageous that the term $\mathbf{F}_{\mathbf{u}}$ does not need to be computed since it is known from the Newton's correction calculation (2.130) and $\mathbf{F}_{\lambda}$ is generally easy to calculate. In the case where $\mathbf{F}_{\lambda}$ is not easily calculated, an alternative simple backward Euler can be used to calculate the tangent approximation for the next step in the continuation

$$\dot{\mathbf{u}}_{\nu+1} \approx \frac{\mathbf{u}_{\nu+1} - \mathbf{u}_{\nu}}{\Delta\lambda_{\nu}}. \tag{2.133}$$

Natural parameter continuation is simple. However, it is not possible to use past a fold, where $|\frac{d\mathbf{u}}{d\lambda}| \to \infty$. To combat this, often the parametrisation of the arc-length also known as pseudo arc-length continuation is used.

2.7.2 Pseudo arc-length continuation

The method of pseudo arc-length continuation was first developed by Keller [68]. It uses the ideal parametrisation of a curve, namely the arc-length as the parameter. Its main benefit is that it can be used past folds where natural parameter continuation fails.

Here the parametrisation is set to be the normalised arc-length given by

$$ds^2 = ||\dot{\mathbf{u}}||^2 + \dot{\lambda}^2 = 1. \tag{2.134}$$

The same Euler predictor and Newton correction are used. Given an initial solution $(\mathbf{u}_0, \lambda_0)$ and derivative $\dot{\mathbf{u}}_0$, the tangent predictor at step ν is given by

$$\begin{aligned} \mathbf{u}_{\nu+1} &= \mathbf{u}_{\nu} + \dot{\mathbf{u}}_{\nu}\Delta s_{\nu}, \\ \lambda_{\nu+1} &= \lambda_{\nu} + \dot{\lambda}_{\nu}\Delta s_{\nu}, \end{aligned} \tag{2.135}$$

where Δs_{ν} is a chosen arc-length step size at step ν in the continuation.

This results in solving the nonlinear system at step ν

$$\begin{aligned} \mathbf{F}(\mathbf{u}_{\nu+1}, \lambda_{\nu+1}) &= 0, \\ (\mathbf{u}_{\nu+1} - \mathbf{u}_\nu) \cdot \dot{\mathbf{u}}_\nu + (\lambda_{\nu+1} - \lambda_\nu)\dot{\lambda}_\nu - \Delta s_\nu &= 0 \end{aligned} \tag{2.136}$$

using a suitable nonlinear solver. Note that the second equation of (2.136) comes from the approximation of (2.134).

Newton's method can be used to solve (2.136) resulting in the iterative matrix system

$$\begin{pmatrix} \mathbf{F}_\mathbf{u}(\mathbf{u}_{\nu+1}^{(k)}, \lambda_{\nu+1}^{(k)}) & \mathbf{F}_\lambda(\mathbf{u}_{\nu+1}^{(k)}, \lambda_{\nu+1}^{(k)}) \\ \dot{\mathbf{u}}_\nu & \dot{\lambda}_\nu \end{pmatrix} \begin{pmatrix} \Delta\mathbf{u}_{\nu+1}^{(k)} \\ \Delta\lambda_{\nu+1}^{(k)} \end{pmatrix} = - \begin{pmatrix} \mathbf{F}(\mathbf{u}_{\nu+1}^{(k)}, \lambda_{\nu+1}^{(k)}) \\ (\mathbf{u}_{\nu+1}^{(k)} - \mathbf{u}_\nu) \cdot \dot{\mathbf{u}}_\nu + (\lambda_{\nu+1}^{(k)} - \lambda_\nu)\dot{\lambda}_\nu - \Delta s_\nu \end{pmatrix} \tag{2.137}$$

with the next Newton iteration values for $\mathbf{u}$ and λ found by

$$\left.\begin{aligned} \mathbf{u}_{\nu+1}^{(k+1)} &= \mathbf{u}_{\nu+1}^{(k)} + \Delta\mathbf{u}_{\nu+1}^{(k)}, \\ \lambda_{\nu+1}^{(k+1)} &= \lambda_{\nu+1}^{(k)} + \Delta\lambda_{\nu+1}^{(k)}. \end{aligned}\right\} \qquad \mu = 0, 1, 2, \ldots \tag{2.138}$$

This matrix equation (2.137) can be split using the block matrix inversion formula, resulting in a two step method [31, 69]. At Newton iteration μ, the first step is solving the linear system

$$\begin{aligned} \mathbf{F}_\mathbf{u}(\mathbf{u}_{\nu+1}, \lambda_{\nu+1})\mathbf{z}_1 &= -\mathbf{F}(\mathbf{u}_{\nu+1}, \lambda_{\nu+1}), \\ \mathbf{F}_\mathbf{u}(\mathbf{u}_{\nu+1}, \lambda_{\nu+1})\mathbf{z}_2 &= \mathbf{F}_\lambda(\mathbf{u}_{\nu+1}, \lambda_{\nu+1}) \end{aligned} \tag{2.139}$$

for $\mathbf{z}_1$ and $\mathbf{z}_2$. The second step, we solve for $\Delta\lambda$ and $\Delta\mathbf{u}$ via the equations

$$\begin{aligned} \Delta\lambda_{\nu+1} &= \frac{-\mathbf{F}(\mathbf{u}_{\nu+1}, \lambda_{\nu+1}) - \dot{\mathbf{u}}_\nu \cdot \mathbf{z}_1}{\dot{\lambda}_\nu - \dot{\mathbf{u}}_\nu \cdot \mathbf{z}_2}, \\ \Delta\mathbf{u}_{\nu+1} &= \mathbf{z}_1 - \Delta\lambda_{\nu+1}\mathbf{z}_2. \end{aligned} \tag{2.140}$$

Equation (2.138) can then be used accordingly to update the next Newton iteration.

Once Newton's method converges to a required tolerance, the new tangent vector is found by solving the matrix system

$$\begin{pmatrix} \mathbf{F}_\mathbf{u} & \mathbf{F}_\lambda \\ \dot{\mathbf{u}}_k & \dot{\lambda}_\nu \end{pmatrix} \begin{pmatrix} \dot{\mathbf{u}}_{\nu+1} \\ \dot{\lambda}_{\nu+1} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}. \tag{2.141}$$

The first equation in the matrix system comes from taking the derivative of (2.127) with respect to the arc-length s, whereas the second is the approximation of (2.134).

Using the block matrix inversion formula, this simplifies to a two step method. First we solve for $\mathbf{z}_1$ and $\mathbf{z}_2$ in the equation

$$\mathbf{F}_\mathbf{u}\mathbf{z}_1 = \mathbf{0}, \tag{2.142}$$

$$\mathbf{F}_\mathbf{u}\mathbf{z}_2 = \mathbf{F}_\lambda. \tag{2.143}$$

The solution to (2.142) is simply $\mathbf{z}_1 = \mathbf{0}$, assuming that a bifurcation is not present, so that the Jacobian is invertible [101]. Also, the solution to (2.143) is known from the Newton correction (2.139). The second step then simplifies to the equations

$$\begin{aligned} \dot{\lambda}_{\nu+1} &= \frac{1}{\lambda_\nu - \mathbf{u}_\nu \cdot \mathbf{z}_2}, \\ \dot{\mathbf{u}}_{\nu+1} &= -\dot{\lambda}_{\nu+1}\mathbf{z}_2. \end{aligned} \tag{2.144}$$

A generalisation proposed by Keller [68] is to use

$$\psi(\mathbf{u}_{\nu+1} - \mathbf{u}_\nu) \cdot \dot{\mathbf{u}}_\nu + (1-\psi)(\lambda_{\nu+1} - \lambda_\nu)\dot{\lambda}_\nu - \Delta s_\nu = 0 \tag{2.145}$$

instead of the second equation in (2.136), for a chosen tuning factor of $0 < \psi < 1$. Adjusting this allows for more or less emphasis on either parameters $\mathbf{u}$ or λ.

Pseudo arc-length continuation has the advantage of continuation past folds. However, this requires the calculation of $\mathbf{F}_\lambda$. In the cases where this becomes difficult, for example, when the parameter is part of the boundary condition, natural parameter continuation is used instead.

2.7.3 Step size control

It is desirable to incorporate a step size control based on how close the predicted solution is to the true solution. One way to estimate this is based on the number of Newton iterations required to converge in the previous continuation step ν [101].

Assume an optimal number of Newton iterations N_{opt}. We choose this to be roughly $N_{opt} \simeq 6$, following the approach taken by Seydel [101]. Let N_ν denote the number of Newton iterations needed for the previous step. If $N_\nu > N_{opt}$, we take a smaller step size in the next correction. Similarly, we take a larger step size if $N_\nu < N_{opt}$. The goal is to adjust the step size so that during each continuation step roughly N_{opt} corrections are used. To adjust the next step size, we introduce the parameter

$$\xi = \frac{N_{opt}}{N_\nu}. \tag{2.146}$$

We can then multiply ξ by the previous step size Δs after each Newton iteration to update

it. To avoid step size changing too rapidly, we define ξ as

$$\xi = \begin{cases} 0.5 & \text{if} \quad \xi < 0.5, \\ \xi & \text{if} \quad 0.5 \leq \xi \leq 2, \\ 2 & \text{if} \quad 2 < \xi. \end{cases} \tag{2.147}$$

Another important consideration during continuation is to avoid reversing directions. To ensure this, it is important that the following condition is satisfied after every step

$$\dot{\mathbf{u}}_\nu \dot{\mathbf{u}}_{\nu+1} + \dot{\lambda}_\nu \dot{\lambda}_{\nu+1} > 0. \tag{2.148}$$

If the condition above does not hold, we reverse the signs of the current tangent to ensure the orientation of the path is preserved.

2.8 Conformal mapping

The forced DJL equation is defined on a non-rectangular domain due to the presence of a bottom topography. In order to use spectral methods, we require a mapping from the physical domain to a rectangular domain at the spectral collocation points. The illustration of the mapping is shown in Figure 2.3. Spectral methods can then used to solve the forced DJL equation in the rectangular domain using Fourier-Chebyshev methods, before being mapped back to the physical domain that we refer to as the forced DJL domain.

There are many mapping functions possible. One such method applied to the forced DJL domain is boundary mapping [104]. However, this method has the downside of requiring the solution to an elliptic problem containing mixed derivatives for the mapping function. Therefore, fast exact solvers cannot be used. Soontiens et al. [104] resorted to GMRES with a variety of preconditioners to solve the elliptic problem. This was later shown to be inefficient and impractical for larger resolutions [103]. We propose a conformal mapping algorithm instead.

2.8.1 Conformal mapping algorithm

Consider a general conformal map $x(u, v)$ and $z(u, v)$. In the (u, v) domain there are several properties that must be satisfied. These are Cauchy's Integral Theorem

$$\int_{\partial w} (x + iz)\, dw = 0, \tag{2.149}$$

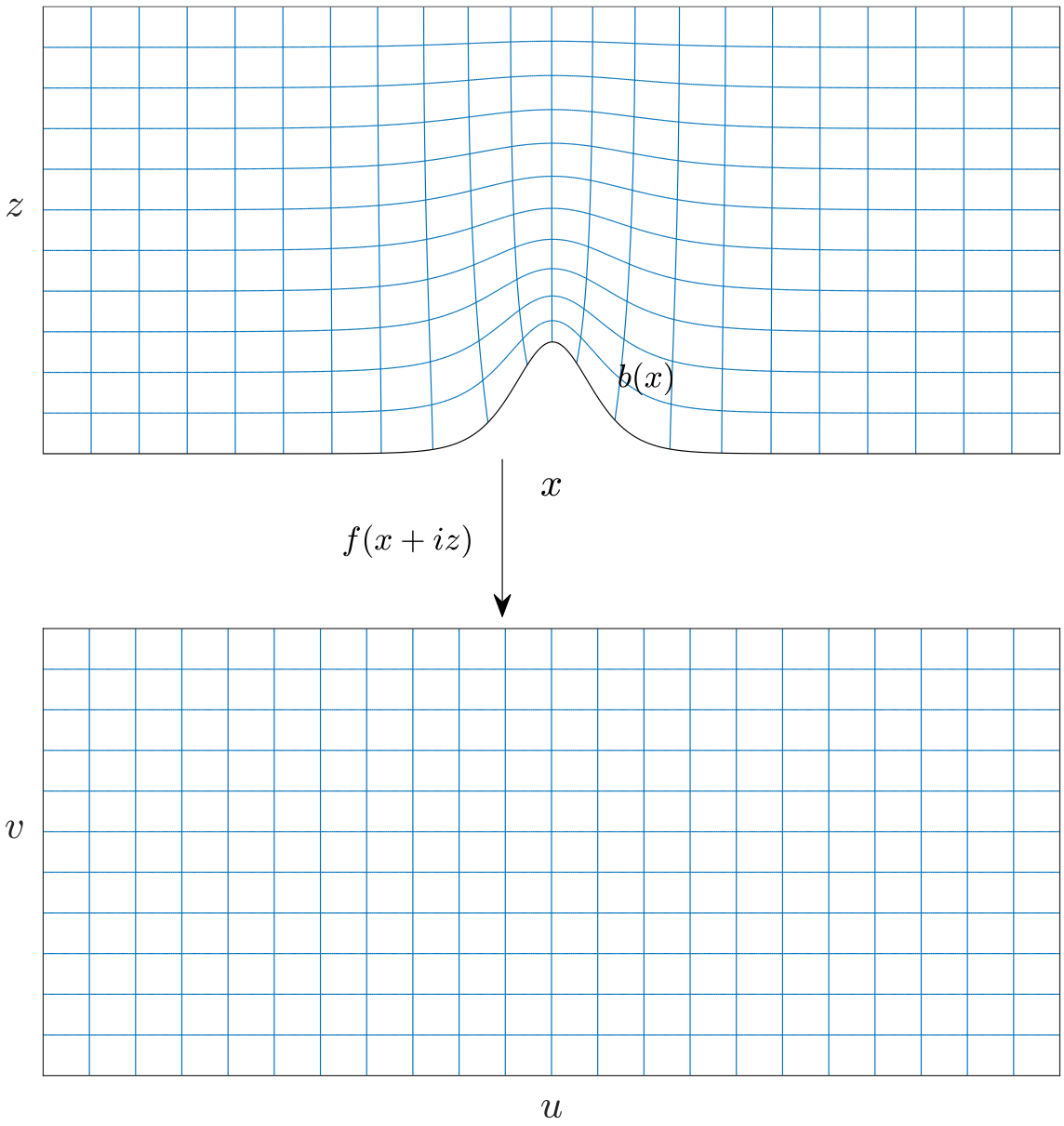

Figure 2.3: A conformal mapping of the (x, z) forced DJL domain with a $b(x)$ bottom topography to the (u, v) rectangular domain at the spectral collocation points. The conformal mapping function f is defined to be $u + iv = f(x + iz)$.

where $w = u + iv$; the Cauchy–Riemann equations

$$\begin{aligned} \frac{\partial x}{\partial u} &= \frac{\partial z}{\partial v}, \\ \frac{\partial x}{\partial v} &= -\frac{\partial z}{\partial u}, \end{aligned} \tag{2.150}$$

and the requirement that x and z are harmonic:

$$\begin{aligned} \nabla^2 x &= 0 \\ \nabla^2 z &= 0 \end{aligned} \tag{2.151}$$

with appropriate boundary conditions. We shall use these properties to propose a conformal mapping algorithm that converts the forced DJL domain to a rectangular domain.

In the forced DJL domain (x, z), we have a near rectangular domain. Let $b(x)$ be the bottom topography shape with $|b(x)| \leq 1$ and $\alpha < 1$. The domain is then given by $x \in [-L, L)$ and $z \in [\alpha b(x), 1]$, with periodic and homogeneous boundary conditions in x and z, respectively. In the rectangular domain (u, v), we have $u \in [-L, L)$ and $v \in [0, H_w]$ with similar periodic and homogeneous boundary conditions in u and v, respectively. We look to solve for the mapping functions $x(u, v)$ and $z(u, v)$, where $x(u, v)$ has periodic boundary conditions, and $z(u, v)$ satisfy

$$\begin{aligned} z(u, 0) &= \alpha b(x(u, 0)), \\ z(u, 1) &= 1. \end{aligned} \tag{2.152}$$

We propose an iterative numerical conformal mapping algorithm by iterating the bottom boundary. For the initial guess, let $x^{(0)}(u, 0) = u$. Then

(1) Solve for H_w using Cauchy's Integral Theorem (2.149).

Since the (u, v) domain is a rectangle, the integral simplifies to

$$\int_{\partial w} x + iz \, dw = \int_{-L}^{L} z(u, 0) \, du + \int_{0}^{H_w} x(L, v) \, dv + \int_{L}^{-L} z(u, H_w) \, du + \int_{H_w}^{0} x(-L, v) \, dv = 0. \tag{2.153}$$

using $x(L, v) = x(-L, v)$ and solving gives

$$H_w = 1 - \frac{1}{2L} \int_{-L}^{L} z(u, 0) \, du = 1 - \frac{1}{2L} \int_{-L}^{L} \alpha b(x^{(k)}(u, 0)) \, du, \tag{2.154}$$

where the integration is performed using Fourier integration methods.

(2) Solve Laplace's equation $\nabla^2 z = 0$ in the (u, v) domain with boundary conditions (2.152).

Laplace's equation can be solved efficiently using Chebyshev-Fourier Ad Hoc methods.

(3) Integrate Cauchy–Riemann equations (2.150) along the boundary $v = 0$ to obtain $x^{(k+1)}(u, 0)$, the updated values on the boundary.

Here let $x^{(k+1)}(u, 0) = u + e^{(k+1)}$ for some unknown error $e^{(k+1)}$. Integrating the Cauchy–Riemann equation gives

$$e^{(k+1)} = \int \frac{\partial z}{\partial v}(u, 0) - 1 \, du. \tag{2.155}$$

Note that $\frac{\partial z}{\partial v}$ is found using Chebyshev integration and the solution with zero mean is taken when performing Fourier integration.

(4) Repeat steps 1–3 with updated boundary $x^{k+1}(u, 0)$ until required tolerance is met.

We use $||x^{k+1}(u, 0) - x^k(u, 0)||_2 < tol$ as the convergence criteria.

2.8.2 Numerical conformal mapping results

We begin by testing the performance of our iterative numerical conformal mapping algorithm. Consider a forced DJL domain (x, z) with $x \in [-\pi, \pi)$ and $z \in [\alpha b(x), 1]$, where $b(x)$ is a topography of one or two bumps of the form

$$b(x) = \operatorname{sech}^2(\pi x), \tag{2.156}$$

and

$$b(x) = \operatorname{sech}^2\left(\frac{3}{2}\pi\left(x + \frac{\pi}{3}\right)\right) + \operatorname{sech}^2\left(\frac{3}{2}\pi\left(x - \frac{\pi}{3}\right)\right) \tag{2.157}$$

respectively. We map the (x, z) onto a rectangular (u, v) domain with $u \in [-\pi, \pi)$ and $v \in [0, H_w]$. A grid resolution of $2^N \times 2^{(N-1)} + 1$ is used in all our experiments with the tolerance set to 10^{-10}. The numerical conformal maps of a one and two bump positive topography is shown in Figures 2.4 and 2.5, respectively. Figure 2.6 and 2.7 show the numerical conformal maps of a negative one and two bump topography, respectively.

At convergence, the Cauchy–Riemann equations are checked to confirm the mapping is conformal. Let the Cauchy–Riemann 2-norms be denoted as $(CR)_1 = \sqrt{\iint_D |\frac{\partial x}{\partial u} - \frac{\partial z}{\partial v}|^2 \, dA}$ and $(CR)_2 = \sqrt{\iint_D |\frac{\partial x}{\partial v} + \frac{\partial z}{\partial u}|^2 \, dA}$, where D is the (u, v) domain.

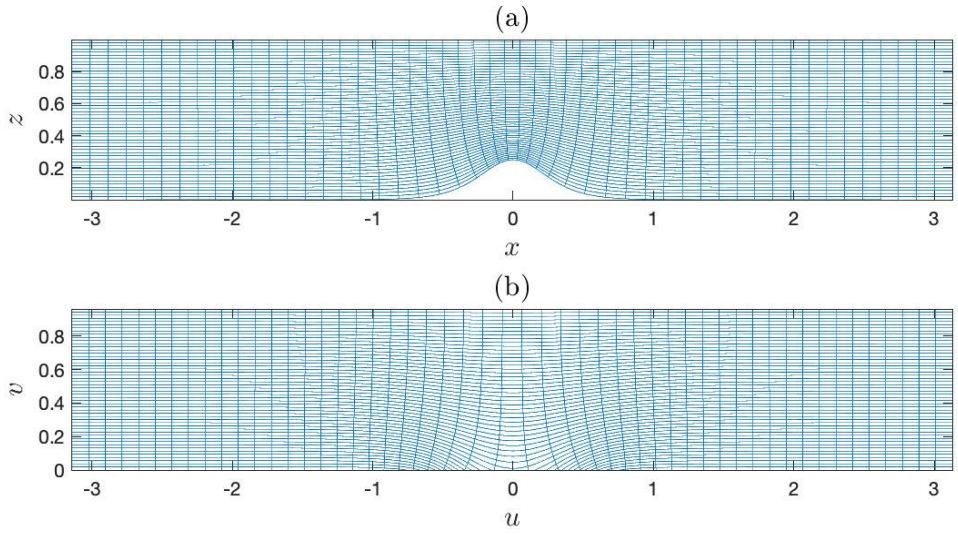

Figure 2.4: Conformal map of a rectangular domain with a one bump bottom topography $x \in [-\pi, \pi)$, $z \in [\alpha b(x), 1]$ with $\alpha = 0.25$ and $N = 9$ to rectangular domain $u \in [-\pi, \pi)$, $v \in [0, H_w]$ where $H_w \approx 0.9628$. Plot of lines of constant (u, v) in the (x, z) domain (a) and lines of constant (x, z) in the (u, v) domain (b).

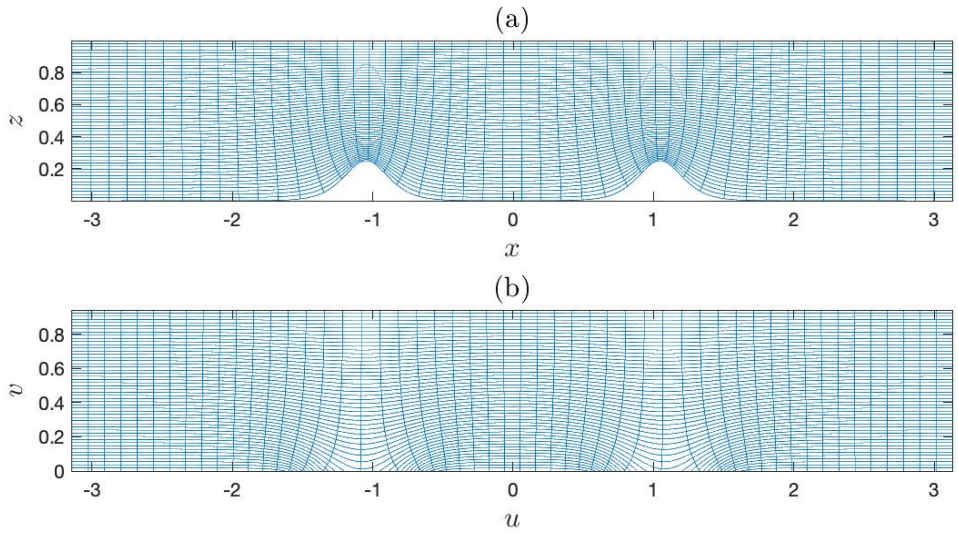

Figure 2.5: Conformal map of a rectangular domain with a two bump bottom topography $x \in [-\pi, \pi)$, $z \in [\alpha b(x), 1]$ with $\alpha = 0.25$ and $N = 9$ to rectangular domain $u \in [-\pi, \pi)$, $v \in [0, H_w]$ where $H_w \approx 0.9433$. Plot of lines of constant (u, v) in the (x, z) domain (a) and lines of constant (x, z) in the (u, v) domain (b).

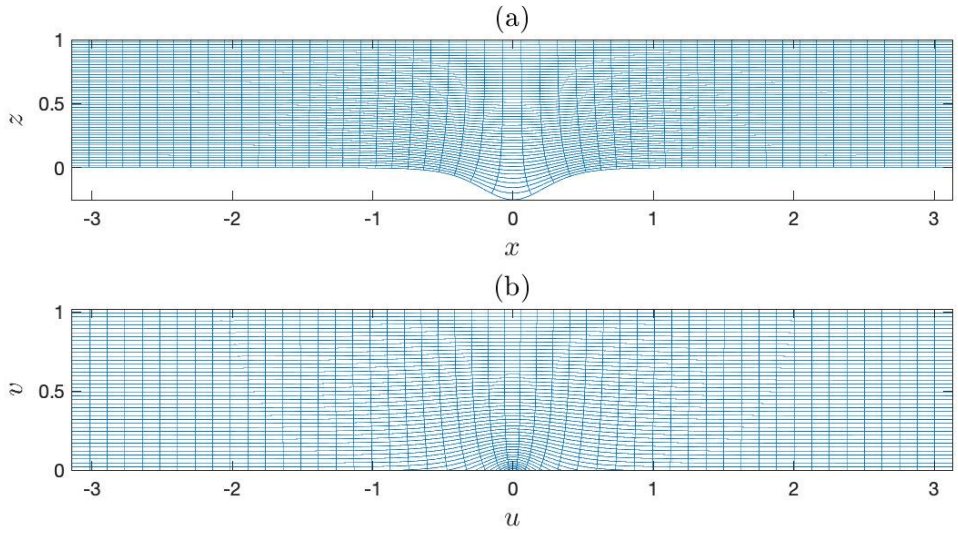

Figure 2.6: Conformal map of a rectangular domain with a one bump bottom topography $x \in [-\pi, \pi)$, $z \in [\alpha b(x), 1]$ with $\alpha = -0.25$ and $N = 9$ to rectangular domain $u \in [-\pi, \pi)$, $v \in [0, H_w]$ where $H_w \approx 1.018$. Plot of lines of constant (u, v) in the (x, z) domain (a) and lines of constant (x, z) in the (u, v) domain (b).

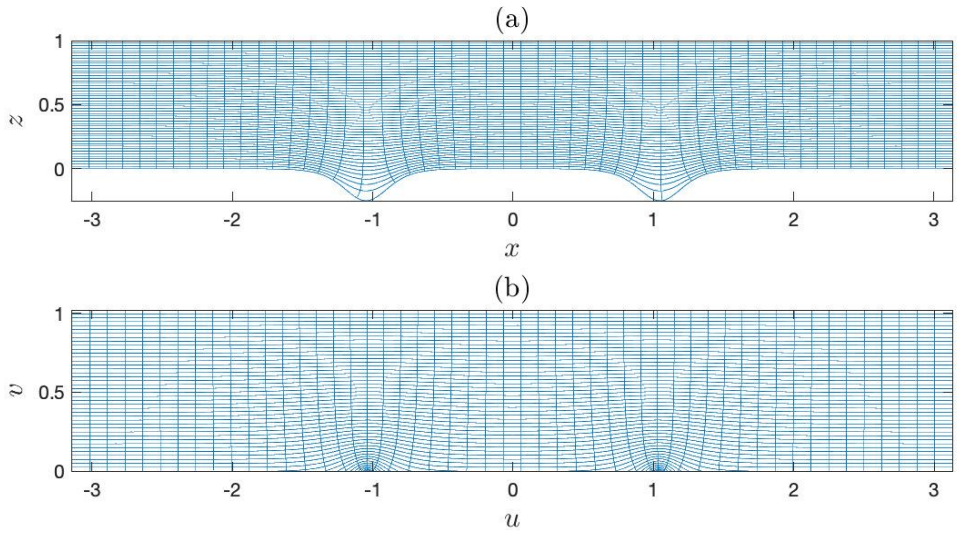

Figure 2.7: Conformal map of a rectangular domain with a two bump bottom topography $x \in [-\pi, \pi)$, $z \in [\alpha b(x), 1]$ with $\alpha = -0.25$ and $N = 9$ to rectangular domain $u \in [-\pi, \pi)$, $v \in [0, H_w]$ where $H_w \approx 1.021$. Plot of lines of constant (u, v) in the (x, z) domain (a) and lines of constant (x, z) in the (u, v) domain (b).

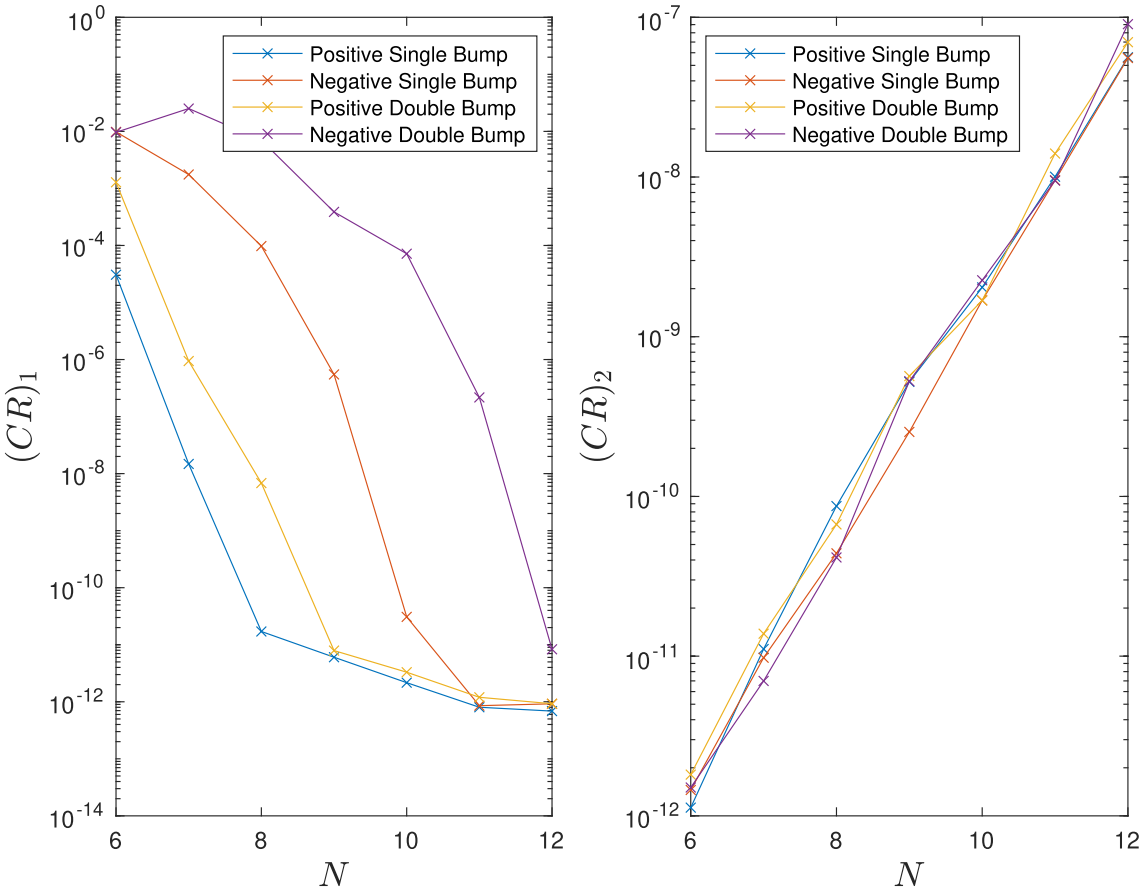

Figure 2.8: Plot of resolution vs Cauchy–Riemann 2-norms for single and double bump topography. Positive topography uses $\alpha = 0.25$ while negative topography uses $\alpha = -0.25$.

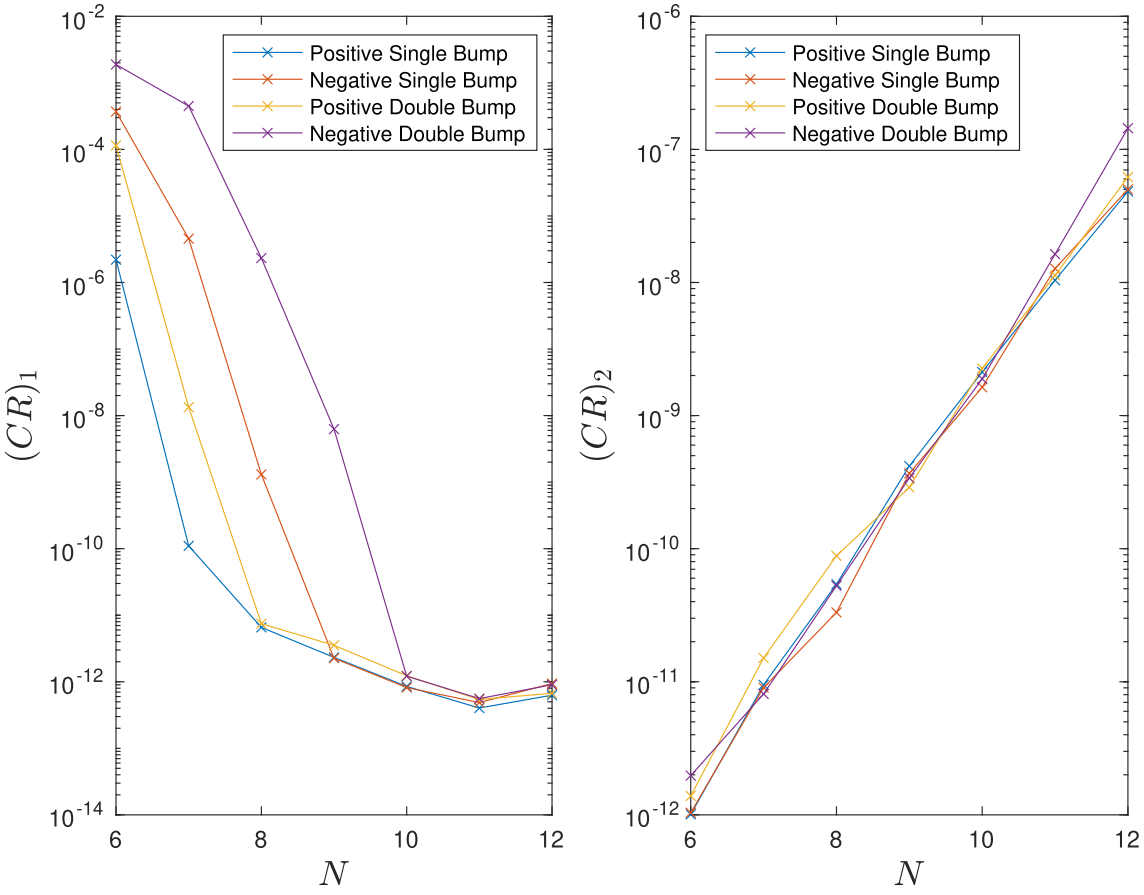

Figure 2.9: Plot of resolution vs Cauchy–Riemann 2-norms for single and double bump topography. Positive topography uses $\alpha = 0.10$ while negative topography uses $\alpha = -0.10$.

In Figure 2.8 and 2.9 we see $(CR)_2$ increases exponentially with resolution but is largely independent of the height and type of bottom topography. This suggest round-off errors may affect the calculation. However, while $(CR)_1$ decreases as resolution increases for all topography types, $(CR)_1$ is significantly larger for negative topography, larger α and double bump over single bump topography.

Next we investigate the performance of our iterative numerical conformal mapping algorithm to changes in resolution and topography height.

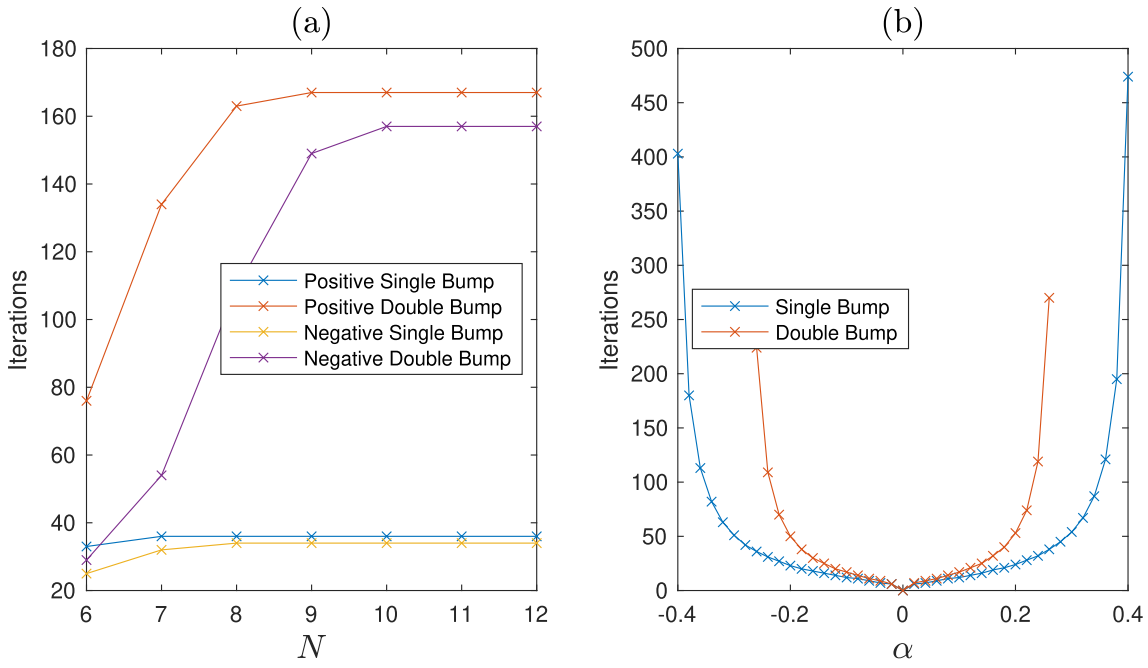

Figure 2.10: Plot of resolutions at a fixed positive topography height $\alpha = 0.25$ and fixed negative topography height $\alpha = -0.25$ vs number of iterations to reach tolerance for single and double bump topography (a). Plot of topography height α with fixed resolution $N = 9$ vs number of iterations to reach tolerance for single and double bump topography (b).

In Figure 2.10 we see the number of iterations is independent of the resolution past $N = 7$ and $N = 9$ for positive single and double bump topography, respectively, and $N = 8$ and $N = 10$ for the negative single and double cases. This suggests $O(N \log N)$ operations are required to generate a conformal map using our algorithm. Increasing $|\alpha|$ on the other hand degrades performance. At $\alpha = 0.42$ and $\alpha = 0.28$ the algorithm fails to convergence for positive single and double bump topography. In the negative cases, the algorithm fails at $\alpha = -0.42$ for single and $\alpha = -0.28$ for double bump.

2.9 Summary

In this chapter we introduced the numerical methods required to solve the fKdV and DJL problems in the upcoming chapters. We started with pseudospectral methods as the discretisation, specifically Fourier collocation methods for periodic boundary condition problems and Chebyshev collocation methods for fixed boundary conditions. For Chebyshev discretisation, the focus was on the Chebyshev Gauss–Lobatto nodes. For N nodes, we defined a method efficient if it requires $O(N \log N)$ operations to perform.

Next, efficient spectral differentiation methods, which do not use matrix multiplication but instead utilise spectral transforms, were explored. Following this, we summarised efficient spectrally accurate integration methods. For Fourier methods, this was simply the trapezoidal rule, while Clenshaw–Curtis is used for Chebyshev integration.

We then utilised pseudospectral and differentiation methods to discretise linear differential equations into a matrix one. However, unlike FD or FE discretised matrices, spectral ones are dense. Efficient exact solvers for spectral systems were introduced, but are only implemented for Laplace, Poisson and Helmholtz differential equations. For all other differential equations, iterative techniques are used. Due to spectral matrices being dense, efficient iterative methods only consists of the class of Richardson iterations and Krylov methods. However, iterative methods used alone are generally slow, which can be accelerated with the used of a preconditioner. Here we only focused on second-order FD methods as preconditioners.

Another method of accelerating iterative methods discussed is multigrid, where we introduced SMG. Richardson iterations are used as SMG relaxations on each grid, while Krylov methods are used as the coarse grid solver. While FD and FE multigrid use polynomial interpolation, SMG utilises efficient trigonometric interpolation for the grid transfer operators. An improved Chebyshev interpolation for Dirichlet boundary conditions is introduced, which removes Gibbs phenomenon and marginally improves SMG performance.

Nonlinear problems are tackled using Newton's method, where SMG is used to invert the Jacobian linear system at every Newton step. It is further used in numerical continuation techniques, which are useful to determine the family of solutions to a system of nonlinear differential equations. We considered specifically the class of predictor-corrector, specifically natural parameter continuation when the Jacobian is not readily known, and pseudo arc-length continuation for problems containing folds.

Finally, we developed an iterative numerical conformal mapping algorithm to map the forced DJL domain to a rectangular one. Fast spectral solvers were used in the algo-

rithm which is significantly faster than the boundary mapping method used by Soontiens et al. [104]. The algorithm is also robust, able to handle both positive and negative bottom topographies. We further showed the number of iterations to convergence is almost independent of the resolution. Thus, our algorithm has the same operation count of $O(N \log N)$ as the spectral solver.

However, it deteriorated as the topography height increased, and eventually failed to converge. Cauchy–Riemann 2-norms also suggests higher resolutions are required for double bump topography and increasing topography height. Hence, we shall ignore negative double bump topography for the DJL equation in the upcoming chapters. The major limitation of our algorithm is that it can only be applied to the forced DJL domain mapped to a rectangular domain.

Chapter 3

Comparison of SMG to CG for solitary wave computations

In this chapter, we compare the performance of SMG to Krylov methods for solitary wave problems. Yang [115, 116] demonstrated CG and biconjugate-gradient (BiCG) significantly outperformed existing methods for solitary wave computations when using a spectral discretisation. Since multigrid and Krylov methods often have comparable convergence properties when used with FD and FE discretisation, we anticipate similar results for spectral when comparing SMG and Krylov methods. We begin by outlining the methodology used by Yang, then compare the performance of Newton-SMG to Newton-Krylov for the same selection of solitary wave problems investigated by Yang.

3.1 Methodology

Yang's Newton-CG method has three main components: the Fourier pseudospectral discretisation, Newton's iterations, and a suitable preconditioner. Yang showed an alternative simpler practical usage of the Newton–Kantorovich method. Consider a general nonlinear differential equation

$$\mathcal{L}_{NL}\mathbf{u} = \mathbf{0}. \tag{3.1}$$

Suppose an initial approximate solution close to the exact solution at iteration step k is known and given by $\mathbf{u}^{(k)} \simeq \mathbf{u}$. Thus we have

$$\mathbf{u} = \mathbf{u}^{(k)} + \mathbf{e}^{(k)}, \tag{3.2}$$

where $\mathbf{e}^{(k)}$, with $|\mathbf{e}^{(k)}| \ll 1$, is the error at the kth step. Substituting (3.2) into (3.1) and expanding about $\mathbf{u}^{(k)}$ gives

$$\mathcal{L}_{NL}\mathbf{u}^{(k)} + \mathcal{L}_{L}\mathbf{e}^{(k)} = O(|\mathbf{e}^{(k)}|^2), \tag{3.3}$$

where $\mathcal{L}_{NL}$ is the original nonlinear operator in (3.1) and $\mathcal{L}_L$ is the linearised operator or Jacobian of $\mathcal{L}_{NL}$. Note that $\mathcal{L}_L$ depends on $\mathbf{u}^{(k)}$. Ignoring terms of order $O(|\mathbf{e}^{(k)}|^2)$, we obtain

$$\mathcal{L}_L \mathbf{e}^{(k)} = -\mathbf{r}^{(k)}, \tag{3.4}$$

where $\mathbf{r}^{(k)} = \mathcal{L}_{NL}\mathbf{u}^{(k)}$ is the nonlinear residual at step k. Equation (3.4) is simply a linear inhomogeneous differential equation for $\mathbf{e}^{(k)}$. This gives the iterative algorithm to update the approximate solution at every step

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{e}^{(k)}. \tag{3.5}$$

As an example, consider the steady fKdV equation

$$-\Delta A + 3A^2 + A_{xx} + \gamma f = 0 \tag{3.6}$$

where Δ and γ are known constants, and $f = f(x)$. Now let $A = v^{(k)} + e^{(k)}$, we obtain

$$-\Delta v^{(k)} + 3(v^{(k)})^2 + v^{(k)}_{xx} - \Delta e^{(k)} + 6v^{(k)}e^{(k)} + 3(e^{(k)})^2 + e^{(k)}_{xx} + \gamma f = 0. \tag{3.7}$$

Ignoring higher order terms in $e^{(k)}$ and rearranging gives the linear inhomogeneous differential equation for the correction

$$\Delta e^{(k)} + 6v^{(k)}e^{(k)} + e^{(k)}_{xx} = -(\Delta v^{(k)} + 3(v^{(k)})^2 + v^{(k)}_{xx} + \gamma f). \tag{3.8}$$

Turning to the preconditioner, Yang introduced a spectral preconditioner as an alternative to FD preconditioning. The Fourier spectral preconditioner, which can be implemented efficiently using FFT, is of the form

$$H = c - \mathcal{L}_{LC}[v], \tag{3.9}$$

where c is a given constant, and $\mathcal{L}_{LC}$ is the Fourier spectral operator consisting strictly of the linear component of the given differential equation. For the problems investigated by Yang, the exact solver Fourier Ad Hoc methods outlined in Chapter 2 can then be used to invert the preconditioner efficiently.

3.2 Results

Yang [115] showed Newton-CG is superior to nonlinear CG and other solitary wave computation methods for symmetric problems. Therefore, we only compare the performance of Newton-SMG to Newton-CG for a selection of problems that were investigated by Yang. The difference between the two is the linear solver used to invert Jacobian linear system (3.4), where Newton-SMG uses SMG and Newton-CG uses CG as the linear solver, re-

spectively. We investigate the performance of the linear solvers for symmetric problems. For non-symmetric ones, BiCGSTAB is used in place of CG, and is preferred over the very similar BiCG, used by Yang, for stability purposes.

Identical Newton's iterations, preconditioner, selection of problems and their computational domain to Yang [115, 116] will be used for both methods for an unambiguous comparison. The solitary wave problems chosen, with details on the physical model is given by Yang [115, 116]. We omit these details in this chapter and only focus on numerical performance comparisons.

We choose FMG for the SMG scheme with one iteration of PMRR after each restriction and prolongation. For each problem, CG or BiCGSTAB is used as the coarse grid solver in the Newton-SMG method for symmetric and asymmetric problems, respectively. The tolerance for the linear and nonlinear residual 2-norm is chosen to be 10^{-10}. One FMG cycle is used within each Newton iteration. This is sufficient to reduce the residual 2-norm to the required tolerance. For a fair comparison to CG and BiCGSTAB, the linear residual to the Jacobian after one FMG cycle is recorded and used as the tolerance for CG and BiCGSTAB within each Newton iteration.

3.2.1 Ground state of the 2D nonlinear Schrödinger equation

The first problem investigated is the computation of the ground state of the 2D nonlinear Schrödinger (NLS) equation given by

$$iU_t + U_{xx} + U_{yy} + |U|^2 U = 0. \tag{3.10}$$

Ground states come in the form $U(x, y, t) = u(x, y)e^{i\mu t}$, where $u(x, y)$ is a positive function satisfying

$$u_{xx} + u_{yy} + u^3 = \mu u. \tag{3.11}$$

The linearised operator or Jacobian of (3.11) is given by

$$\mathcal{L}_L = \partial_{xx} + \partial_{yy} + 3u^2 - \mu, \tag{3.12}$$

and the preconditioner used is

$$H = c - \partial_{xx} - \partial_{yy}. \tag{3.13}$$

Following Yang's approach, the parameters chosen are $\mu = 1$, $c = 1$ and the computational domain is taken to be a square with $x \in [-15, 15]$ and $y \in [-15, 15]$. The initial guess for Newton's method is

$$u_0(x, y) = 2.2e^{-x^2-y^2}. \tag{3.14}$$

Results shown in Figure 3.1 for the computation of the ground state of the 2D NLS

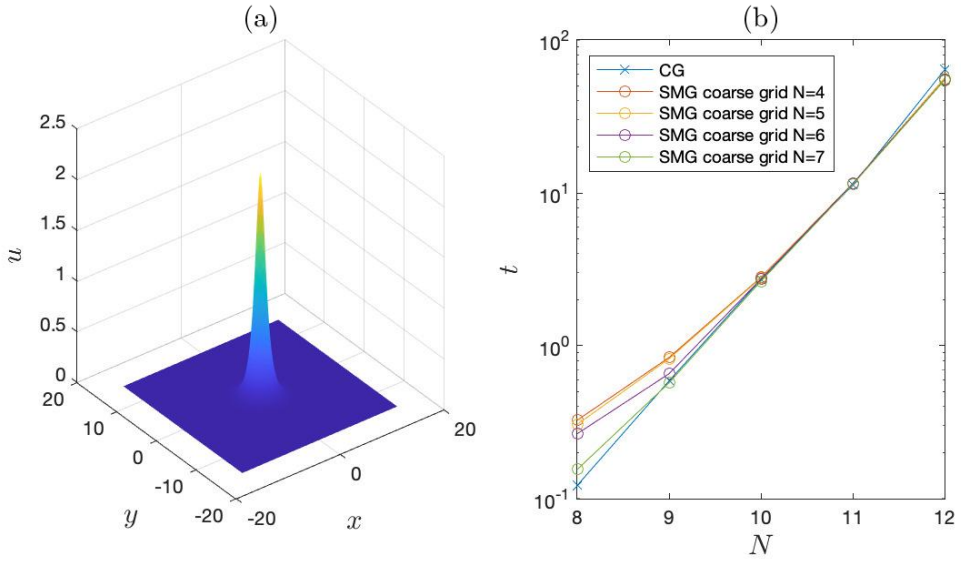

Figure 3.1: Solution to the ground state of the 2D NLS equation (a) and corresponding comparison between CG and SMG time taken for different grid resolutions and different coarse grids (b). The resolution is taken to be $2^N \times 2^N$, with 2^N points in the x and y direction respectively. We note a grid resolution of lower than $N = 4$ results in Newton-CG failing to converge.

show Newton-CG and Newton-SMG have similar efficiency and scaling behaviour as larger resolution are used and are largely independent on the coarse grid chosen. Newton-SMG at $N = 8$ appears to be an outlier to the scaling behaviour, where Newton-CG slightly outperforms Newton-SMG. This is likely due to the use of only two grids in SMG. A higher resolution coarse grid appears to be preferable in the outlier region, but is indistinguishable at higher resolutions. The performance of Newton-SMG edges out Newton-CG only at the finest resolutions but since SMG is more complicated to implement than CG, it does not seem worthwhile for this problem.

3.2.2 Semi-infinite-gap solitons in the 2D NLS equation

The second problem chosen is the computation of semi-infinite-gap solitons in the 2D NLS equation with periodic potentials given by the equation

$$u_{xx} + u_{yy} - V_0(\sin^2 x + \sin^2 y)u + \sigma|u|^2 u = -\mu u, \tag{3.15}$$

where V_0 is the strength of the periodic potential, μ is the propagation constant and $\sigma = \pm 1$, corresponds to self-focusing or self-defocusing nonlinearity.

The Jacobian for (3.15) is given by

$$\mathcal{L}_L = \partial_{xx} + \partial_{yy} - V_0(\sin^2 x + \sin^2 y) + 3\sigma u^2. \tag{3.16}$$

We consider two solutions to (3.15). In the first solution, the parameters $\sigma = 1$, $\mu = 4.11$, $V_0 = 6$ are used. The same preconditioner is used as given in (3.13) but with $c = 3$ for this problem. The computational domain is taken to be a square with $x \in [-5\pi, 5\pi]$ and $y \in [-5\pi, 5\pi]$ and the initial guess for Newton's method is chosen to be

$$u_0(x, y) = 1.15\,\mathrm{sech}(2\sqrt{x^2 + y^2}). \tag{3.17}$$

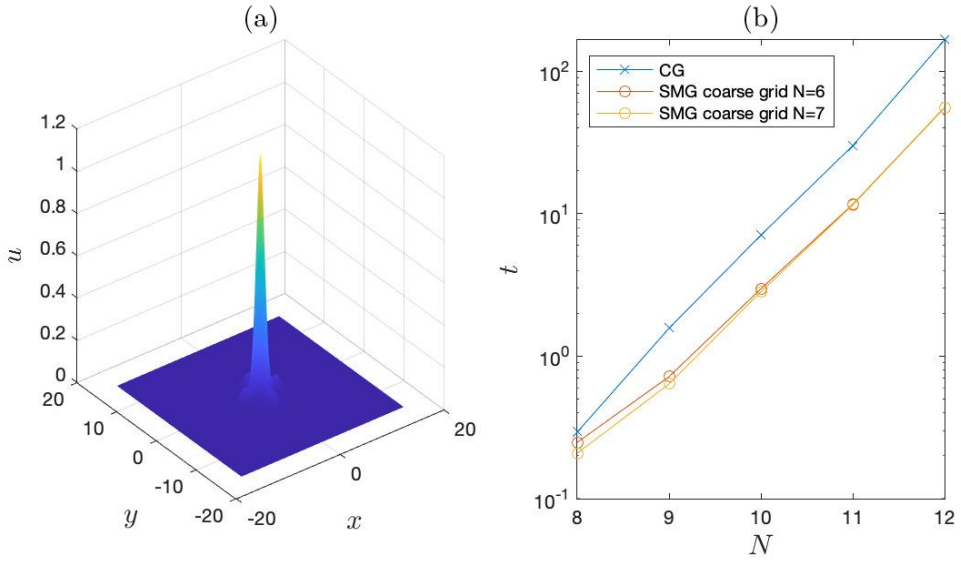

Figure 3.2: Solution to the semi-infinite-gap solitons in the 2D NLS equation with parameters $\sigma = 1$, $\mu = 4.11$, $V_0 = 6$ (a) and corresponding comparison between CG and SMG time taken for different grid resolutions and different coarse grids (b). The resolution is taken to be $2^N \times 2^N$, with 2^N points in the x and y direction respectively. We note a grid resolution of lower than $N = 6$ results in Newton-CG failing to converge.

Similarly to the previous problem, the computation of the semi-infinite-gap solitons in the 2D NLS show similar scaling behaviour after $N = 9$ for both Newton-CG and Newton-SMG as seen in Figure 3.2. A similar scaling outlier is seen at $N = 8$ for Newton-SMG likely due to the same reason as the previous problem. A larger coarse grid also appears to be slightly more preferable in Newton-SMG. However, for this problem, past $N = 9$, Newton-SMG clearly outperforms Newton-CG by approximately a factor of 3 in time taken.

The second solution uses the parameters $\sigma = 1$, $\mu = 3.71$, $V_0 = 6$ with the initial guess

$$u_0(x, y) = 0.49\,\mathrm{sech}(2\sqrt{x^2 + y^2}). \tag{3.18}$$

The computational domain is increased to $x \in [-10\pi, 10\pi]$ and $y \in [-10\pi, 10\pi]$, with the preconditioner chosen identical to the first problem.

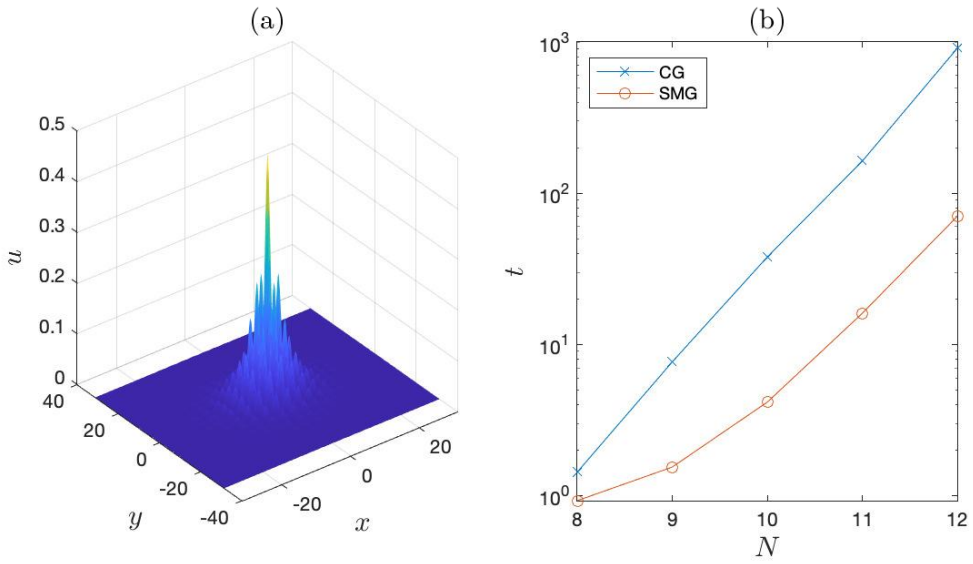

Figure 3.3: Solution to the semi-infinite-gap solitons in the 2D NLS equation with parameters $\sigma = 1$, $\mu = 3.71$, $V_0 = 6$ (a) and corresponding comparison between CG and SMG time taken for different grid resolutions (b). The resolution is taken to be $2^N \times 2^N$, with 2^N points in the x and y direction respectively. We note a grid resolution of lower than $N = 7$ results in Newton-CG failing to converge.

We observe a grid resolution of $N = 6$ or lower results in Newton-CG failing to converge, hence limits the coarse grid to $N = 7$ in Newton-SMG. In Figure 3.3 we see the scaling outlier for Newton-SMG extends to $N = 9$ before similar scaling is seen for both Newton-CG and Newton-SMG. Despite this, Newton-SMG clearly outperforms Newton-CG, with performance increases by approximately an order of magnitude past $N = 9$.

3.2.3 On-site positive lattice solitons in the 2D NLS equation

The third problem chosen is the on-site positive lattice solitons in the 2D NLS equation with a periodic potential and self-focusing nonlinearity. The equation is given by (3.15), except now with $\sigma = -1$ due to the self-focusing nonlinearity. The parameters $\mu = 5$ and $V_0 = 6$ are chosen for the first solution with same preconditioner given in (3.13) with $c = 3$. The computational domain is taken to be a square $x \in [-5\pi, 5\pi]$, $y \in [-5\pi, 5\pi]$ with initial condition

$$u_0(x, y) = 1.15\,\mathrm{sech}^2(\sqrt{x^2 + y^2}\cos x \cos y). \tag{3.19}$$

We see in Figure 3.4 a similar scaling outlier at $N = 8$ for Newton-SMG, but otherwise Newton-CG and Newton-SMG display the same scaling behaviour after $N = 9$. A larger coarse grid also appears preferable in the outlier zone. For this problem, Newton-SMG outperforms Newton-CG by approximately a factor of 3 past $N = 9$.

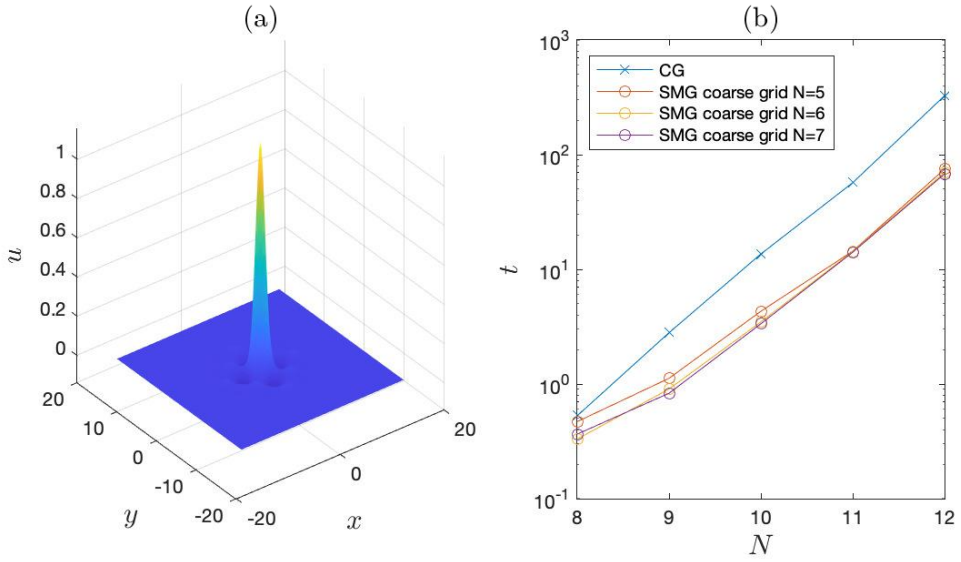

Figure 3.4: Solution to the on-site positive lattice solitons in the 2D NLS equation with parameters $\sigma = -1$, $\mu = 5$, $V_0 = 6$ (a) and corresponding comparison between CG and SMG time taken for different grid resolutions and different coarse grids (b). The resolution is taken to be $2^N \times 2^N$, with 2^N points in the x and y direction respectively. We note a grid resolution of lower than $N = 5$ results in Newton-CG failing to converge.

The second solution uses the parameters $\mu = 4.56$ with the domain increased to $x \in [-10\pi, 10\pi]$, $y \in [-10\pi, 10\pi]$. The preconditioner used is identical to the first solution with the initial guess

$$u_0(x, y) = 0.56\,\mathrm{sech}^2(\sqrt{x^2 + y^2}\cos x \cos y). \tag{3.20}$$

Similar to the second solution in Section 3.2.2, we find a grid resolution of $N = 6$ or lower results in Newton-CG failing to converge, hence limits the coarse grid to $N = 7$ in Newton-SMG. Figure 3.5 show Newton-SMG scaling outliers at $N = 8$ and $N = 9$. For $N > 9$, Newton-CG and Newton-SMG show similar scaling behaviour. For this problem, Newton-SMG outperforms Newton-CG by approximately an order of magnitude past the outlier zones.

3.2.4 Depression and elevation waves in the fifth-order Kadomtsev–Petviashvili equation equation

Next we consider a non-symmetric problem, namely, the depression and elevation waves in the fifth-order Kadomtsev–Petviashvili (KP) equation

$$(u_{xxxx} + 2u_{xx} + 3u^2 - \nu u)_{xx} + u_{yy} = 0, \tag{3.21}$$

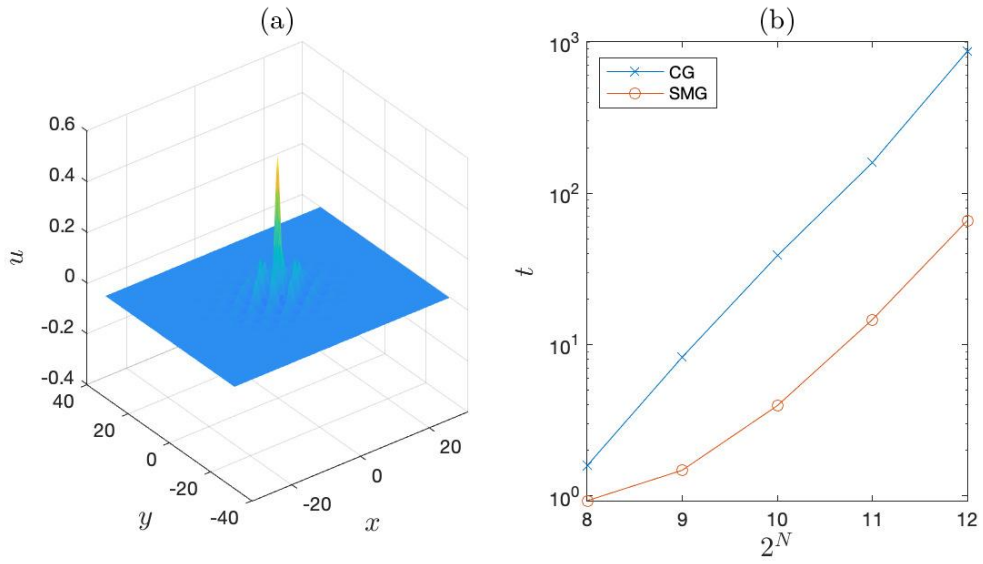

Figure 3.5: Solution to the on-site positive lattice solitons in the 2D NLS equation with parameters $\sigma = -1$, $\mu = 4.56$, $V_0 = 6$ (a) and corresponding comparison between CG and SMG time taken for different grid resolutions (b). The resolution is taken to be $2^N \times 2^N$, with 2^N points in the x and y direction respectively. We note a grid resolution of lower than $N = 7$ results in Newton-CG failing to converge.

with the chosen parameter $\nu = -1.2$ and preconditioner given by the operator

$$H = c - \partial_{xx}(\partial_{xxxx} + 2\partial_{xx} - \nu) - \partial_{yy} \tag{3.22}$$

with $c = 0.0001$. The computational domain is taken to be $x \in [-60\pi, 60\pi]$, $y \in [-30\pi, 30\pi]$. The anisotropy results in a non-symmetric Jacobian matrix where Newton-BiCGSTAB is used in place of Newton-CG.

The grid resolution is taken to be $2^N \times 2^{(N-3)}$ points in the x direction and y direction, respectively. The initial guess for the depression wave for Newton's method is given by

$$u_0(x, y) = -0.43\,\mathrm{sech}(0.3\sqrt{x^2 + y^2}\cos x), \tag{3.23}$$

while the initial guess for the elevation wave is $-u_0$.

Since equation (3.21) admits a continuous family of solutions, the integral condition

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(x, y)\,dx\,dy = 0 \tag{3.24}$$

is used to filter for the zero-mean solution, imposed after every Newton iteration. This is equivalent to the mass conservation law for the KP equation. This is performed by setting the first element of the 2D Fourier transform of $v^{(k)}$, the solution after each Newton

iteration, to zero, that is,

$$\tilde{v}^{(k)}(1,1) = 0. \tag{3.25}$$

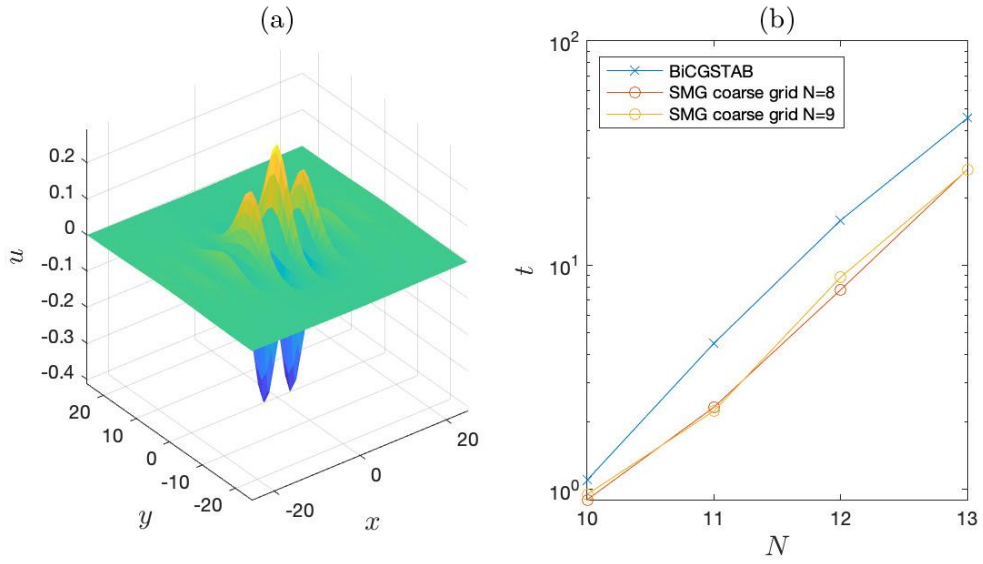

Figure 3.6: Solution to the depression waves in the fifth-order KP equation (a) and corresponding comparison between BiCGSTAB and SMG time taken for different grid resolutions and different coarse grids (b). The resolution is taken to be $2^N \times 2^{(N-3)}$, with 2^N points in the x direction and $2^{(N-3)}$ points in the y direction respectively. We note a grid resolution of lower than $N = 7$ results in Newton-BiCGSTAB failing to converge.

The residual tolerances are adjusted for the depression and elevation waves of the KP equation for different grid resolutions. This is because the stagnation of the residual indicates that the error has reached machine precision. This is likely due to the growing condition number of the discretised Jacobian. The tolerances are set just above where the residual stagnates as given in Table 3.1. Note that the largest resolution in either the x or y direction determines the residual tolerance. For example, if a grid has $2^{10} \times 2^{13}$ grid points, the tolerance will be set at 10^{-7}.

N	Residual 2-norm tolerance
10	10^{-10}
11	10^{-9}
12	10^{-8}
13	10^{-7}
14	10^{-5}

Table 3.1: Residual 2-norm tolerances set for the depression and elevation waves of the KP equation for different resolutions.

Results for the depression waves in the fifth-order KP equation shown in Figure 3.6 show similar scaling behaviour of Newton-BiCGSTAB and Newton-SMG past the scaling

outlier at $N = 10$. A higher resolution coarse grid appears slightly preferable at $N = 10$ and $N = 12$. For this problem, Newton-SMG outperforms Newton-BiCGSTAB by approximately a factor of 4 past the outlier at $N = 10$.

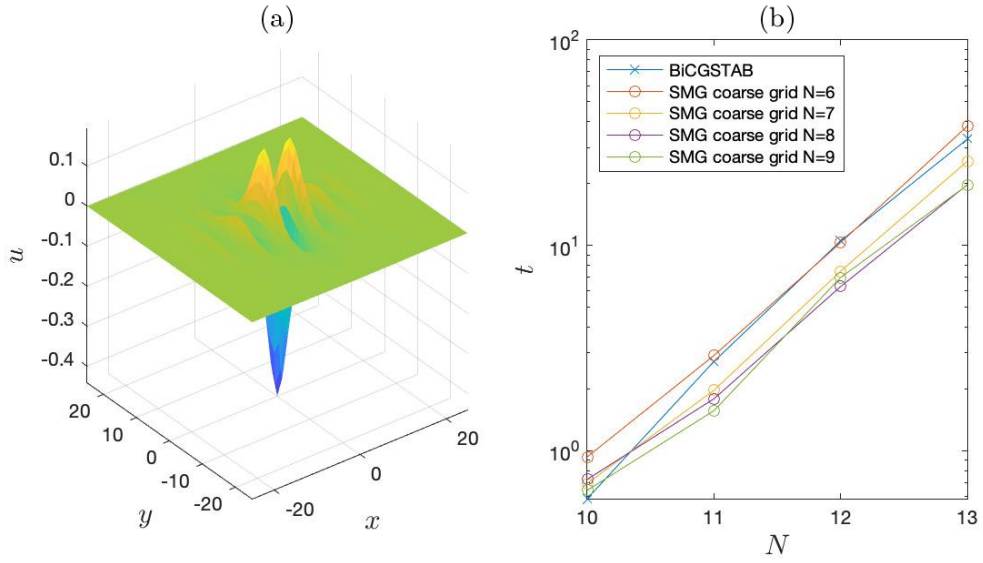

Figure 3.7: Solution to the elevation waves in the fifth-order KP equation (a) and corresponding comparison between BiCGSTAB and SMG time taken for different grid resolutions and different coarse grids (b). The resolution is taken to be $2^N \times 2^{(N-3)}$, with 2^N points in the x direction and $2^{(N-3)}$ points in the y direction respectively. We note a grid resolution of lower than $N = 6$ results in Newton-BiCGSTAB failing to converge.

Results for the elevation waves in the fifth-order KP equation shown in Figure 3.7 show Newton-BiCGSTAB outperforms Newton-SMG at $N = 10$. For $N > 10$, Newton-SMG outperforms Newton-BiCGSTAB by approximately a factor of 2. A larger resolution coarse grid appears to be preferable over a smaller coarse grid resolution for this problem.

We observe that the fifth-order KP equation and the grid resolution chosen exhibits anisotropy. Since PMRR degrades as a result, we anticipate improved performance by letting $h_x = h_y$. For the domain size given, the resolution $2^{(N-1)} \times 2^N$ is chosen.

In Figure 3.8 we observe Newton-SMG consistently outperforming Newton-BiCGSTAB at all fine grids and coarse grids chosen when $h_x = h_y$. However, the coarse grid performance is less consistent than with the anisotropy grid.

3.3 Summary

We conducted a variety of solitary wave computations and compared Newton-Krylov methods developed by Yang to Newton-SMG. Since the same hardware used throughout,

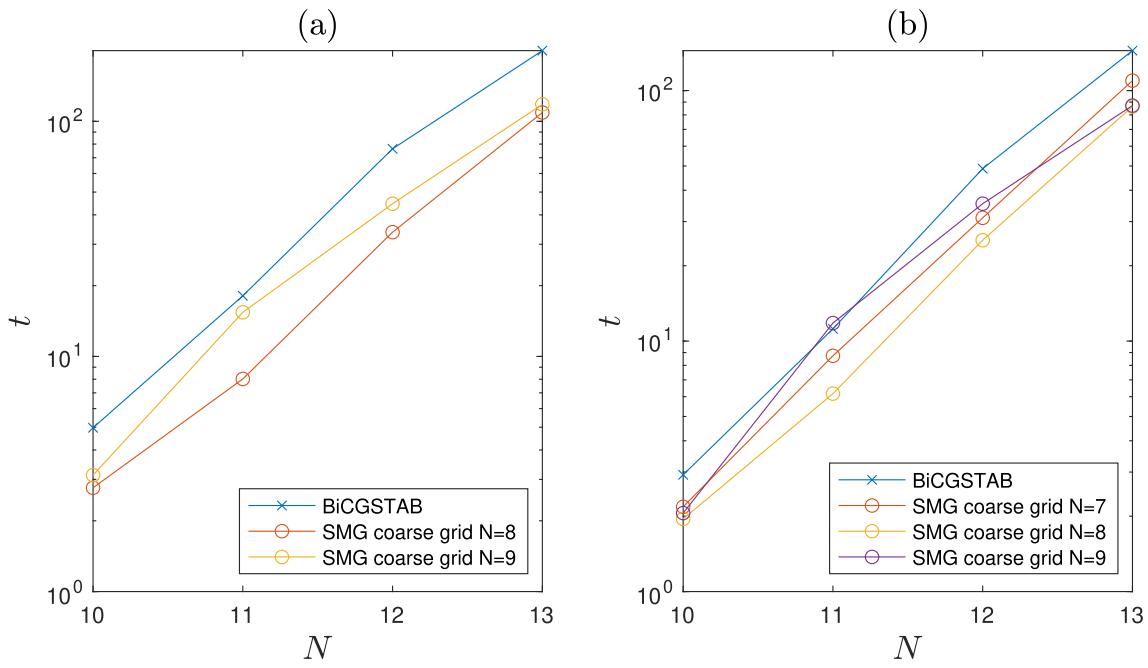

Figure 3.8: Comparison between BiCGSTAB and SMG time taken for different grid resolutions and different coarse grids for the solution of the fifth-order KP equation for the depression wave (a) and elevation wave (b). The resolution is taken to be $2^{(N-1)} \times 2^N$ points in the x direction and y direction respectively. We note a grid resolution of lower than $N = 8$ and $N = 7$ results in Newton-BiCGSTAB failing to converge for the depression and elevation waves respectively.

this gives a good indication on relative performance.

For the computation of the semi-infinite-gap solutions in the 2D NLS equation and the on-site positive lattice solitons in the 2D NLS equation, Newton-SMG outperforms Newton-CG by a factor of 3 to 10 in computational time for larger resolutions. However, for the computation of the ground state of the 2D NLS equation, they perform very similarly for all resolutions. Since SMG is more difficult to implement than CG, Newton-SMG does not seem worthwhile for this problem. Lastly, for the computation of depression and elevation waves in the fifth-order KP equation, Newton-SMG also outperforms Newton-BiCGSTAB by a factor of 2 to 4. This shows the robustness of SMG when used in an anisotropic problem.

There exist a minimum grid resolution, where Newton-Krylov fails to converge, ranging from $N = 4$ to $N = 7$ depending on the problem being solved. Initially aliasing effects were suspected. However, testing showed dealiasing did not improve the minimum grid resolution.

To achieve an unambiguous comparison, the linear residual 2-norm is recorded after each FMG cycle and used as the tolerance for Krylov methods. After one FMG cycle

it is generally in the order of 10^{-10}. However, the linear solver during each Newton iteration does not generally require solving to such accuracy. In Yang's paper [115] it is set to $10^{-2}r_{NL}$ where r_{NL} is the nonlinear residual at each Newton step. While Krylov methods require more iterations and thus have more gradual decreases in residual that can benefit from this lower threshold, SMG can not, since half a V-cycle or FMG cycle is not possible. Further testing with Yang's tolerances show small improvements to Newton-Krylov methods, though not changing the conclusions of the results shown here.

Chapter 4

The forced Korteweg-de Vries equation

The main goals of this chapter are to investigate critical flow and hydraulic fall solutions to the steady fKdV equation, that arises from a perturbation expansion of the stratified Euler equations [105]. In dimensionless form this is given by

$$-\Delta A + 3A^2 + A_{xx} = -\gamma f, \tag{4.1}$$

obtained by considering the steady state of (1.4) and integrating the resultant equation in x. In the application to free surface flow [50], A is the free surface displacement

The downstream boundary conditions, which defines the constant of integration to zero, are given by

$$A \to 0, \quad A_x \to 0, \quad f \to 0 \quad \text{as} \quad x \to -\infty. \tag{4.2}$$

We follow Ee and Clarke [34] and use the steady fKdV equation in the form (4.1), with $f = \operatorname{sech}^2 x$ as the chosen bump forcing. There are several reasons for these choices. Firstly, this allows comparison to the numerical results of Ee and Clarke [34]. Moreover, several analytic solitary wave solutions are available for the chosen equation form and bump forcing. Perturbation analysis can be performed around these, which provides a natural starting point in the numerical continuation process. Throughout this chapter, we focus on hydraulic fall and solitary wave solutions to equation (4.1).

4.1 Hamiltonian and solution types

Different steady solution types to the fKdV equation can be found using the Hamiltonian for (4.1) given by

$$\mathcal{H}(A, A_x) = \gamma f A - \frac{1}{2}\Delta A^2 + \frac{1}{2}A_x^2 + A^3. \tag{4.3}$$

An understanding of hydraulic fall solution dynamics can be found by observing the Hamiltonian in the far field, where topographic effects are negligible, that is $f = 0$. The phase portrait is then given by

$$A_x^2 = \Delta A^2 - 2A^3 + C. \tag{4.4}$$

The phase plane has two critical points. In the case $\Delta > 0$, they are a saddle point at $(A, A_x) = (0, 0)$ and a centre (minimum) at $(A, A_x) = (\Delta/3, 0)$. Hydraulic fall solutions traverse from one critical point to the other in the phase space. The $\Delta < 0$ case simply reverses the nature of the critical points, where $(A, A_x) = (0, 0)$ becomes the saddle point while $(A, A_x) = (\Delta/3, 0)$ becomes the centre. Converting from $\Delta > 0$ to the $\Delta < 0$ case can be made simply through the transformation $A \leftarrow A - \Delta/3$. Thus, without loss of generality we assume $\Delta > 0$. Similarly, without loss of generality, we assume the downstream limit as the saddle point since the flow is reversible. That is, we define the upstream and downstream directions as $x \to \pm\infty$.

In the special case of critical flow, when $\Delta = 0$, the critical points coincide and a solitary wave solution path traverses from the origin $(A, A_x) = (0, 0)$ back to the origin.

4.1.1 Solitary wave solutions

We begin by investigating solitary wave solutions to (4.1) that exist in the critical flow case with $\Delta = 0$, that satisfy

$$A_{xx} + 3A^2 = -\gamma \operatorname{sech}^2(x), \tag{4.5}$$

with boundary conditions $A \to 0$ as $x \to \pm\infty$. Solitary wave solutions for $\Delta \neq 0$ also exist as shown by Camassa and Wu [17].

Keeler et al. [67] (henceforth referenced as KBB) showed the condition $\gamma \int_{-\infty}^{\infty} f(x)dx \leq 0$ is necessary for the existence of a solitary wave solution. This can be obtained by integrating (4.1) and applying the far field boundary conditions (4.2). Therefore, solitary wave solutions only exist for negative values of γ for our chosen bump forcing.

Topographical effects become negligible as $x \to \pm\infty$, thus solitary wave solutions far upstream and downstream must satisfy the equation

$$3A^2 + A_{xx} = 0. \tag{4.6}$$

If we assume a solution of the form $A = cx^n$, substituting into and ignoring the trivial

solution $c = 0$, (4.6) yields $c = -2$ and $n = -2$ giving

$$A = -2/x^2. \tag{4.7}$$

This suggests there exists solitary waves that exhibits algebraic decay as $x \to \pm\infty$. We shall denote these as algebraic decaying solitary waves (ADSW). An example of ADSWs were found by KBB for the scaled equation in the form $u_{\xi\xi} + u^2 = \alpha e^{-\xi^2}$, i.e., with a Gaussian forcing. KBB used matched asymptotics in the limits $\alpha \ll 1$ and $\alpha \gg 1$, confirming this algebraic far field decay. This was further supported by numerical results confirming the predicted matched asymptotic behaviour.

Analytic exponential decaying solitary wave (EDSW) solutions are also known for equation (4.5). These can be found by assuming a solution in the form $A = \alpha\,\mathrm{sech}^2(x)$ for some unknown constant α [17]. Solving for the constant α gives $A = 2\,\mathrm{sech}^2(x)$ at $\gamma = -8$.

To investigate additional EDSWs following Cullen's method [27], a 4th-order shooting method was applied to (4.5) with initial conditions $(A, A_x) = (0, 0)$ as $x \to \infty$. We set $x = 100$ as the downstream limit and integrate to $x = 0$. The behaviour of the solution derivative at $x = 0$ is observed for a range of γ values as shown in Figure 4.1. Since solitary waves must satisfy the condition $A_x(0) = 0$ due to symmetry, we apply a secant method to solve for the γ values that satisfy this condition. The first and second solitary waves found match the EDSW analytic solution at $\gamma = -8$ and numerical ones at $\gamma \simeq -24.3$ which agrees with Ee and Clarke [34] within 1%. Several more solitary wave solutions were found at various γ values as shown in Figure 4.2. Hence, equation (4.5) corresponds to a nonlinear eigenvalue problem, with eigenvalues γ and EDSW eigenfunctions. The first 11 eigenvalues are shown in Table 4.1, with their location following a power-law relation numerically found to be $\gamma \sim -1.23 \times n^{3.4}$ for $n \gg 1$.

In the limit $\gamma \to -\infty$, we follow KBB's matched asymptotic approach to determine the solution behaviour in the vicinity of the topography $|x| \ll 1$. We begin by ignoring the dispersion term in (4.5) since $|A_{xx}| \ll A^2$ in the limit $|x| \ll 1$. The resulting quadratic equation can then be solved for A. Now taking the amplitude and noting that $\lim_{x\to 0} \mathrm{sech}^2(x) = 1$, we obtain the asymptotic behaviour

$$|A(0)| \sim \sqrt{\frac{-\gamma}{3}}. \tag{4.8}$$

Figure 4.2 shows agreement with the predicted large γ behaviour.

The numerical results suggest there may be an infinite number of eigenvalues and corresponding EDSW eigenfunctions to (4.5).

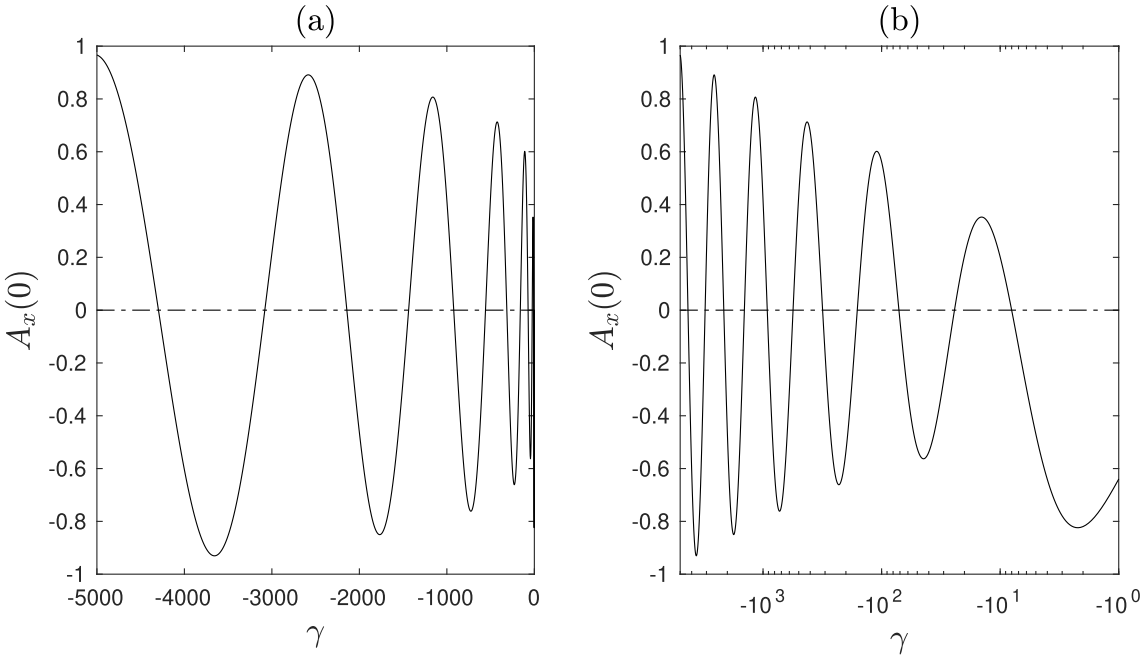

Figure 4.1: Plot of $A_x(0)$ vs γ for the solutions obtained from the shooting method with γ in linear (a) and log scale (b). Solitary wave solutions are located, where the curve intercepts the γ axis, or when $A_x(0) = 0$.

4.1.2 Hydraulic fall solutions

Next we investigate hydraulic fall solutions, to (4.1) in the general case $\Delta \neq 0$, which traverse between the two critical points in the phase plane. We begin by exploring analytic perturbation and asymptotic results.

Analytic results

Asymptotic analysis of the parameter space $\gamma(\Delta)$ for hydraulic fall solutions in the limit $\gamma \to 0$ was shown by Grimshaw and Smyth [50] to give the relationship

$$\left(\frac{\Delta}{3}\right)^3 = (K\gamma)^2 \tag{4.9}$$

with $K = \int_{-\infty}^{\infty} f \, dx$.

In the limit as $\gamma \to \infty$, the dispersion term A_{xx} in equation (4.1) becomes negligible. This results in a quadratic equation in A, where the parametric relationship for large positive γ matched at the topographical maximum [34] is given by

$$\Delta = \sqrt{12\gamma}. \tag{4.10}$$

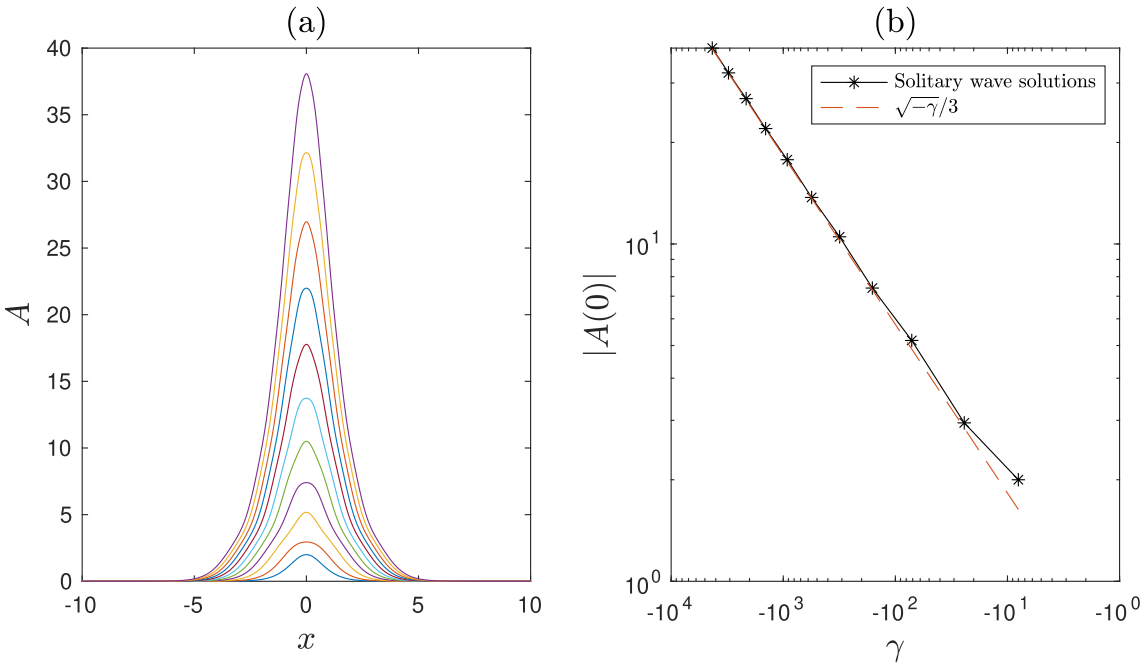

Figure 4.2: First 11 EDSW solutions to equation (4.5) obtained using a shooting method (a). EDSW amplitude vs γ for the solutions obtained and comparison to the large γ limit $|A| \sim \sqrt{\frac{-\gamma}{3}}$ (b).

n	γ
1	-8
2	-24.29
3	-71.42
4	-160.44
5	-314.51
6	-558.37
7	-921.42
8	-1437.12
9	-2140.18
10	-3080.82
11	-4296.34

Table 4.1: First 11 eigenvalues γ to 2 decimal places, corresponding to EDSW eigenfunctions to equation (4.5).

To our knowledge, there are no known asymptotic solutions for $\gamma \to -\infty$.

Perturbation methods can be used to find the parametric relationship $\Delta(\gamma)$ in the vicinity of the analytic solution at $\gamma = -8$. Let $\Delta = \epsilon$ for $0 < \epsilon \ll 1$, for which

$$-\epsilon A + 3A^2 + A_{xx} = -\gamma f. \tag{4.11}$$

We assume A and γ are regular perturbation series in powers of ϵ given by

$$\gamma = \gamma^{(0)} + \epsilon\gamma^{(1)} + \epsilon^2\gamma^{(2)} + ... = \sum_{n=0}^{\infty} \epsilon^n \gamma^{(n)}, \tag{4.12}$$

$$A = A^{(0)} + \epsilon A^{(1)} + \epsilon^2 A^{(2)} + ... = \sum_{n=0}^{\infty} \epsilon^n A^{(n)}, \tag{4.13}$$

and noting that $A^{(0)} = 2\,\mathrm{sech}^2(x)$ and $\gamma^{(0)} = -8$. Substituting (4.12) and (4.13) into (4.11) and equating $O(\epsilon^1)$ terms give the first-order problem

$$-A^{(0)} + 6A^{(0)}A^{(1)} + A^{(1)}_{xx} = -\gamma^{(1)} f. \tag{4.14}$$

Using the substitution $\theta = \tanh(x)$, the equation simplifies to the Legendre differential equation with a constant right hand side

$$\frac{d}{d\theta}\left[(1-\theta^2)\frac{dA^{(1)}}{d\theta}\right] + 12A^{(1)} = -\gamma^{(1)} + 2. \tag{4.15}$$

We use the hydraulic fall boundary from the phase plane

$$A \to 0 \quad \text{as} \quad x \to -\infty \quad \text{and} \quad A \to \Delta/3 = \epsilon/3 \quad \text{as} \quad x \to \infty, \tag{4.16}$$

to give the boundary conditions for the first-order problem to be

$$\begin{aligned} \lim_{\theta \to -1} A^{(1)} &= 0 \\ \lim_{\theta \to 1} A^{(1)} &= 1/3. \end{aligned} \tag{4.17}$$

The solution is given by $\gamma^{(1)} = 0$ and $A^{(1)} = \frac{1}{6} - \frac{1}{6}P_3(\theta)$, where P_3 is the Legendre polynomial of order three.

Now consider $O(\epsilon^2)$ terms. Similarly, using the substitution $\theta = \tanh(x)$, we obtain the equation

$$(1-\theta^2)\left[\frac{d}{d\theta}\left[(1-\theta^2)\frac{dA^{(2)}}{d\theta}\right] + 12A^{(2)}\right] = -\gamma^{(2)}(1-\theta^2) + A^{(1)} - 3(A^{(1)})^2. \tag{4.18}$$

Substituting in $A^{(1)}$ and simplifying gives

$$\frac{d}{d\theta}\left[(1-\theta^2)\frac{dA^{(2)}}{d\theta}+12A^{(2)}\right]=-\gamma^{(2)}+b_0+b_2P_2(\theta)+b_4P_4(\theta), \tag{4.19}$$

with $b_0=\frac{11}{72}$, $b_2=\frac{115}{504}$ and $b_4=\frac{5}{42}$. Subsequent boundary conditions for higher order perturbed problem are

$$\begin{aligned}\lim_{\theta\to-1}A^{(n)}&=0,\\ \lim_{\theta\to1}A^{(n)}&=0\end{aligned} \tag{4.20}$$

for $n\geq 2$. The solution for the second-order problem is then given by

$$A^{(2)}=-\frac{\gamma^{(2)}}{12}+\frac{b_0}{12}+\frac{b_2}{6}P_2(\theta)-\frac{b_4}{8}P_4(\theta), \tag{4.21}$$

$$\gamma^{(2)}=b_0+2b_2-\frac{3}{2}b_4=\frac{31}{72}. \tag{4.22}$$

Thus in the vicinity of $\gamma=-8$, the parametric curve for hydraulic solutions to $O(\epsilon^2)$ must lie on the curve

$$\gamma=-8+\frac{31}{72}\Delta^2. \tag{4.23}$$

Now turning to solitary wave solutions, we apply the boundary condition so that the upstream and downstream depths are zero for the first-order and higher order problems, that is

$$\begin{aligned}\lim_{\theta\to-1}A^{(n)}&=0,\\ \lim_{\theta\to1}A^{(n)}&=0\end{aligned} \tag{4.24}$$

for $n=0,1,2,\ldots$. This gives the solution $A^{(1)}=0$ and $\gamma^{(1)}=-2$. Repeatedly solving the higher order problems gives the solution $A^{(n)}=0$ and $\gamma^{(n)}=0$ for $n\geq 2$. This is consistent with the known result that solitary wave solutions in the form $A=2\,\mathrm{sech}^2(x)$ exist on the line $\Delta=-\frac{1}{2}(\gamma+8)$ [17].

The perturbation analysis agrees with the numerical results obtained by Ee and Clarke but is unable to obtain the negative branch emanating from $\gamma=-8$. Thus, we resort to numerical results for the extended negative region.

Numerical results

The steady fKdV equation (4.1) contains two free parameters and requires an additional condition for a unique solution. Ee and Clarke [34] and Cullen [27] both utilised a minimisation algorithm based on the Hamiltonian to locate hydraulic fall solutions. Cullen used a minimisation of the wave-resistance coefficient introduced by Wu [113]. By imposing

the upstream and downstream boundary conditions for hydraulic fall solutions: That is $\mathcal{H}(0,0) = 0$ and $\mathcal{H}(\Delta/3, 0) = \Delta^3/54$, the net change in Hamiltonian between $x = \pm\infty$ is calculated to be

$$\delta\mathcal{H} = \Delta^3/54. \tag{4.25}$$

Differentiating and then integrating the Hamiltonian (4.3) over the real line, the wave-resistance coefficient or wave-drag constraint is obtained, which must equal the net change in Hamiltonian given by

$$\int_{-\infty}^{\infty} \frac{d\mathcal{H}}{dx} dx = \int_{-\infty}^{\infty} \gamma f_x A dx = \frac{\Delta^3}{54}. \tag{4.26}$$

The parametric space $\Delta(\gamma)$ for hydraulic fall solutions was first numerically determined by Ee and Clarke [34]. A shooting method in combination with a minimisation algorithm was used to locate solutions, which converged to the downstream saddle point. This was built upon a consideration of the Hamiltonian. The conditions imposed ensure the downstream amplitude lies within the homoclinic orbit in the phase plane. The minimisation conditions used in their algorithm are

$$\begin{aligned} \mathcal{H}(\mathcal{H} - \mathcal{H}_s) &\leq 0, \\ -\frac{\Delta}{3} \leq A &\leq \frac{\Delta}{6}, \end{aligned} \tag{4.27}$$

where $\mathcal{H}_s = \mathcal{H}(\Delta/3, 0) = \Delta^3/54$. Ee and Clarke utilised tabletop solutions allowing the use of Fourier spectral methods. Their results show that the hydraulic fall solutions evolve into the analytic solitary wave solution at $\gamma = -8$. Furthermore, a second solitary wave solution was found numerically at $\gamma \simeq -24.55$. Between the zero solution at $\gamma = 0$ and the solitary wave at $\gamma = -8$, and the two subsequent solitary waves at $\gamma = -8$ and $\gamma \simeq -24.55$, the connecting parametric curve $\Delta(\gamma)$ followed an arc shape. The hydraulic fall and solitary wave solutions in the continuation exhibit exponential far field decay. The EDSWs discussed in Section 4.1.1 also coincide with the solitary waves at $\Delta = 0$ in the hydraulic fall solution. Both the small γ approximation (4.9) and large positive γ limit (4.10) are consistent with the numerical results of Ee and Clarke.

Cullen [27] that showed asymmetric solutions must correspond to a minimum of the wave-resistance coefficient (4.26), with equality for hydraulic fall solutions. A Chebyshev collocation method as detailed in Cullen and Clarke [28] was used for a truncated numerical domain with artificial boundary conditions simulating zero far field boundary conditions. This method potentially introduces nonlinear waves in the downstream limit. To combat this, a minimisation process was chosen over the integral condition (4.26). A genetic algorithm was used for the minimisation criteria and was able to replicate the parametric relationship obtained by Ee and Clarke for the region $-10 \leq \gamma \leq 5$, however their numerical algorithm failed to converge for $\gamma \leq -10$.

We aim to extend the hydraulic fall parametric relationship results by Ee and Clarke, and Cullen. Their results suggest that between each subsequent EDSW solutions in Figure 4.2 there may also exist some connecting arc-shaped parametric relationship $\gamma(\Delta)$. The numerically obtained EDSW solutions from the shooting method and their corresponding γ value locations can then be used as a natural starting point for the continuation process.

Both Ee and Clarke, and Cullen encountered numerical difficulties in mapping the extended negative region. We develop several improvements for numerical stability. Firstly, we use the wave-drag constraint (4.26) directly, without a minimisation procedure. Secondly, similar to the approach by Ee and Clarke, a second identical bump forcing is added downstream which converts the asymmetric problem into a symmetric one. Boundary conditions are automatically enforced using Fourier spectral methods.

4.2 Numerical Method

To obtain tabletop solutions to (4.1), we consider forcing consisting of two identical bumps separated by distance $2h$, given by $f(x) = \text{sech}^2(x-h) + \text{sech}^2(x+h)$. Since the wave-drag constraint applies to asymmetric hydraulic fall solutions, only half of the domain is considered. We look for the solution $A(\gamma, \Delta)$ and the hydraulic parametric relationship $\Delta(\gamma)$ to the nonlinear system

$$A_{xx} + 3A^2 - \Delta A = -\gamma f(x), \tag{4.28}$$

$$\int_0^\infty \gamma f_x A dx = \int_{-\infty}^0 \gamma f_x A dx = \frac{\Delta^3}{54} \tag{4.29}$$

on $x \in (-\infty, \infty)$ with the hydraulic fall boundary conditions (4.16). The problem is not strictly symmetric but can be treated symmetrically on a sufficiently large numerical domain. Newton's method is used to solve the system of nonlinear equations with the Jacobian of (4.28) given by

$$J(v) = v_{xx} + (\Delta + 6A)v. \tag{4.30}$$

Numerical continuation methods are required to generate the hydraulic fall parametric space $\gamma(\Delta)$. While Ee and Clarke, and Cullen used a simple natural parameter continuation, we employ a pseudo arc-length continuation.

The details of pseudo arc-length continuation is discussed in Chapter 2. Here we mention the nuances involved since the wave-drag constraint in (4.29) is an integral condition, and the parameter $\boldsymbol{\lambda}_\nu$ is a vector $[\gamma_\nu, \Delta_\nu]$ at each continuation step ν. Let $[F, G]$ be the approximation to (4.28) and (4.29), respectively, at the Fourier collocation points and ν

the index at each numerical continuation step. Then the resulting system is given by

$$F = (A_{xx})_\nu + 3A_\nu^2 - \Delta_\nu A_\nu + \gamma_\nu f(x), \tag{4.31}$$

$$G = \sum_{j=0}^{N/2} w_j \gamma_\nu f_x(x_j)(A_j)_\nu - \frac{\Delta_\nu^3}{54}, \tag{4.32}$$

where w_j are simply the trapezoidal rule weights, $\Delta_\nu = \Delta(\gamma_\nu)$ and $(A_j)_\nu = A(x_j, \gamma_\nu, \Delta(\gamma_\nu))$. We solve the system $[F, G] = 0$ using pseudo arc-length continuation with Euler predictor and Newton corrector. This requires the solution to three systems of linear equations at each step of the Newton corrector.

4.3 Numerical Results

4.3.1 Tabletop Solutions

The numerical results for tabletop solutions presented here utilise 2^{13} grid points, a domain of $L = 200$ and bump spacing of $h = 10$ for negative γ and $h = 14$ for positive γ. We find these choices of different h for the positive and negative region most numerically stable. It is important that h is large enough so that the bumps are adequately separated to minimise tail influence. However, a large h requires a larger numerical domain L for the periodic assumption to hold. This in turn determines the minimum grid points needed to resolve the wave. Provided these conditions are met, the results are largely independent of the parameters L and h. The coarsest grid in the multigrid algorithm is set at 2^5 grid points. One V-cycle consisting of 1 PMRR relaxation after each restriction and prolongation, with second-order FD preconditioning, is used within each Newton iteration, with a residual tolerance set to 10^{-10}. Examples of tabletop solutions obtained using SMG for positive and negative forcing are given in Figure 4.3. This approach improves upon the methods used by both Ee and Clarke, and Cullen, since it eliminates the need for a minimisation algorithm. However, care must be taken in the choice of numerical domain and the distance between bumps.

Numerical results, which are supported by the results of Lee and Whang [76], suggest that tabletop solutions only depend on the topography height and the Froude number and are independent of the distance between topographies h, provided h is large enough to avoid overlap and trailing tail influence. This is to be expected if tabletop solutions are symmetric variants of asymmetric hydraulic fall solutions.

4.3.2 Critical flow ADSW solutions

Next we investigate ADSW solutions to (4.5) in the critical flow case. In the numerical results by KBB, several extra types of critical flow solutions were found not predicted by

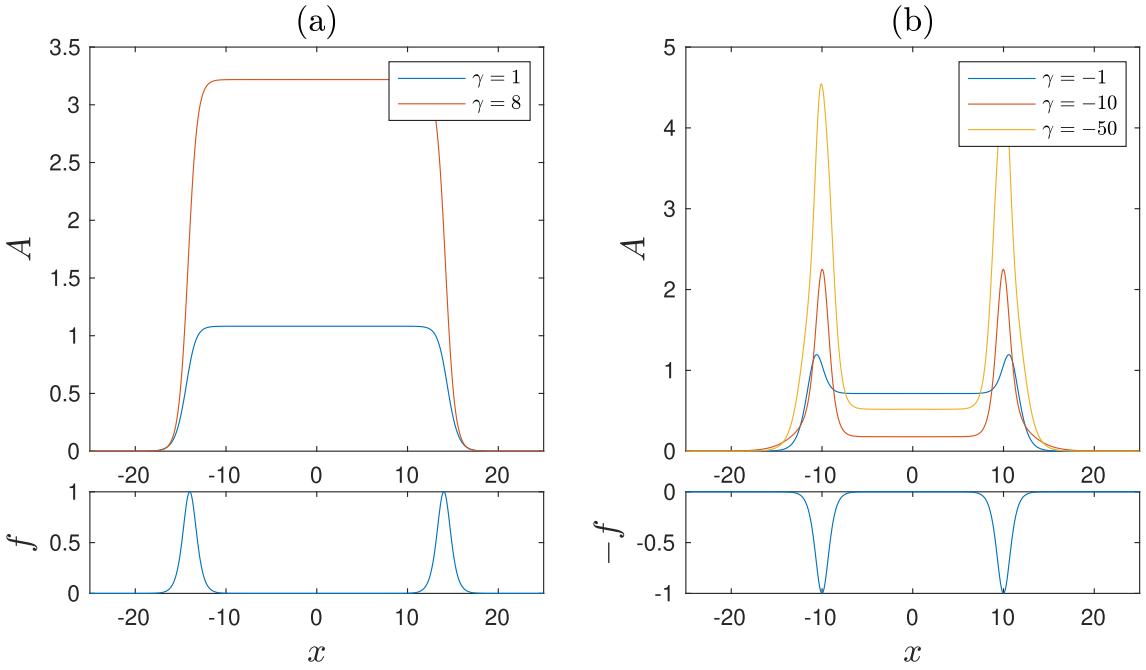

Figure 4.3: Tabletop solutions (upper) to equation (4.1) and scaled topography (lower) for $(\gamma, \Delta) \approx (1.00, 3.25)$ and $(\gamma, \Delta) \approx (8.00, 9.66)$ (a) and $(\gamma, \Delta) \approx (-1.00, 2.14)$, $(\gamma, \Delta) \approx (-9.99, 0.54)$ and $(\gamma, \Delta) \approx (-50.0, 1.56)$ (b).

their asymptotic analysis. We follow the labels used by KBB and note these as type I, II, IIa, IIb, IIc and IId [67, Figure 3]. In contrast to type I solutions, all type II solutions have a termination point in phase space.

Our numerical investigations suggest the numerical ADSWs obtained by KBB originate from EDSWs. The reasoning is two-fold. Firstly, the termination point for different types of ADSWs in KBB appear to coincide with the location of the EDSWs. Secondly, during the continuation process used to find the hydraulic parametric relationship, after reaching a solitary wave eigenvalue, the solution would occasionally jump to a branch of ADSW identical to that observed by KBB. Furthermore, the type of ADSW obtained at the jump depended on which eigenvalue was approached. For example, type I solutions obtained after the zero solution, type II after the first EDSW, type IIa after the second EDSW etc.

We test this hypothesis by using the numerically obtained EDSW from the shooting method as the starting point for the continuation process. The solution obtained from the shooting method is used in the pseudo arc-length continuation with parameter γ. Solution types I, II, IIa, IIb, IIc, IId from KBB are replicated in Figure 4.4 with each type originating from a different EDSW. A small turning point is observed briefly during the transition from EDSW to ADSW. Numerical results from our shooting method suggests there may be an infinite set of EDSW. If each EDSW also evolves into a separate type

of ADSW, this suggest there may be infinitely many types of ADSW existing on $\Delta = 0$, with a new solution type continuum appearing after each new EDSW.

Figure 4.5 shows the far field decay rate of ADSW type I solution approaching the predicted decay rate of equation (4.6). Similar behaviour is also observed for other ADSW solution types.

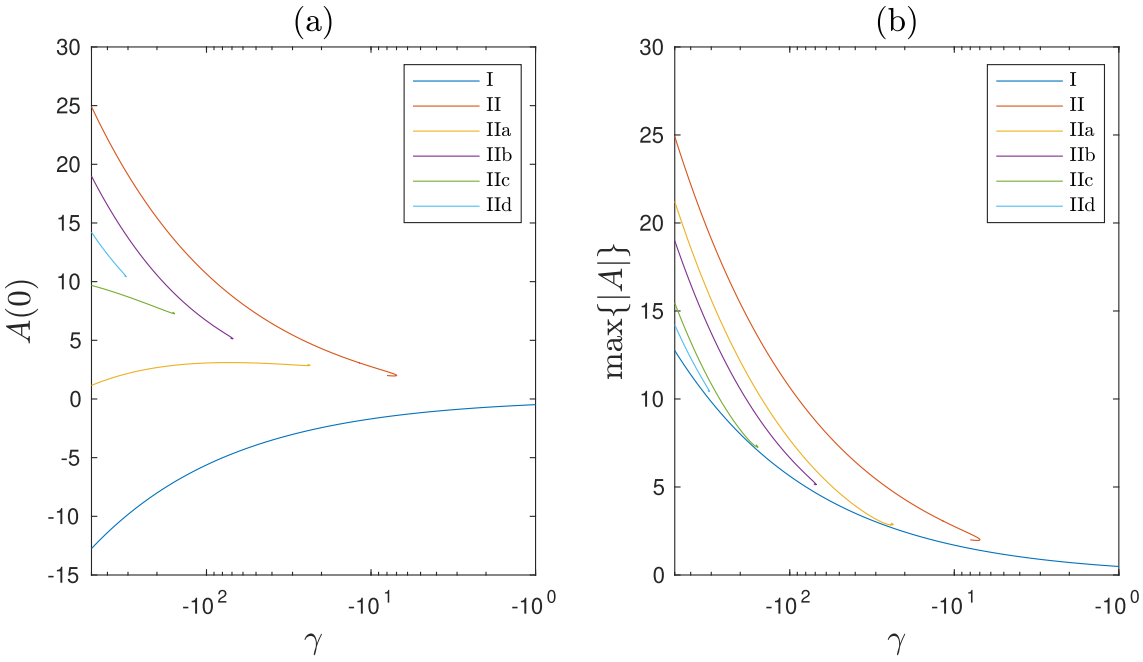

Figure 4.4: Solution space of ADSW solutions to equation (4.28) similar to that obtained by KBB. The turning points are enlarged for visibility, capturing the transition from EDSW to ADSW. Plot labels are consistent with KBB for comparison. Plot of γ vs A(0) (a) and vs maximum amplitude (b).

4.3.3 Hydraulic parametric results

Lastly, we aim to extend the parametric space obtained by Ee and Clarke. The parametric relationship $\gamma(\Delta)$ for hydraulic fall solutions, generated using pseudo arc-length, is consistent with that obtained by Ee and Clarke in the region $\gamma \approx -24.5$ to $\gamma = 10$. Our results also agree with the asymptotic analysis in the limit $\gamma \to 0$, large positive γ limit and the perturbation analysis around $\gamma = -8$. Furthermore, our numerical results are able to be significantly extended into the negative γ region, as shown in Figure 4.6. Figure 4.7 show the agreement of the numerical parametric space to the perturbation solution in the vicinity of the analytic solution.

Locations of each solitary wave solution coincide with the EDSW locations found using the shooting method. The EDSW solutions at $\Delta = 0$ cannot be obtained using pseudo

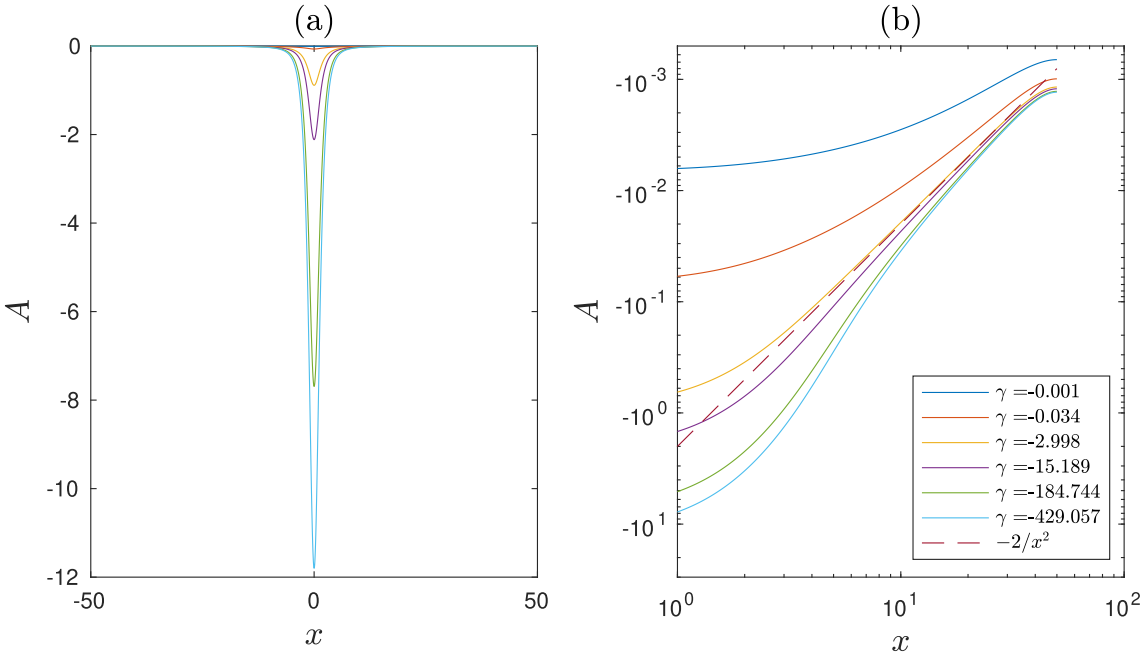

Figure 4.5: Evolution of Keeler type I solution to equation (4.28) (a) and corresponding decay rate in log-log scale obtained at various γ values (b). The behaviour approaches the predicted rate in equation (4.7). Results near $x = \pm 50$ are unreliable due to Fourier methods enforcing periodic boundary conditions.

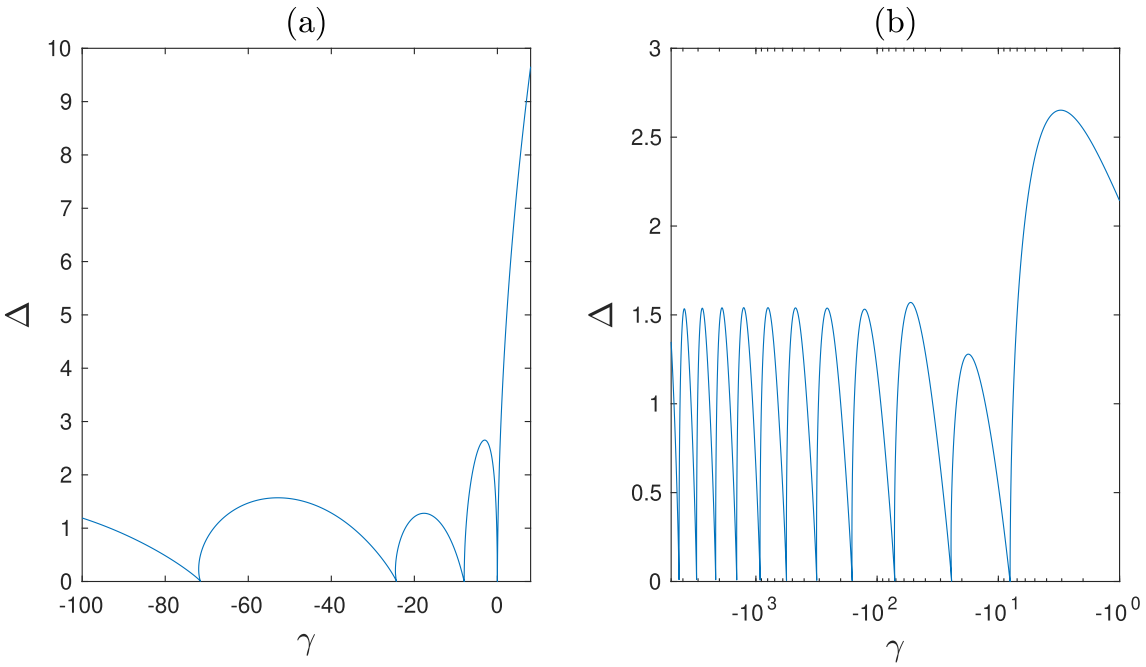

Figure 4.6: Parametric space for hydraulic fall solutions to equation (4.1) with γ axis linear (a) and log scaling (b). A small vertical turning point is present before every root (except at $\gamma = 0, -8$)

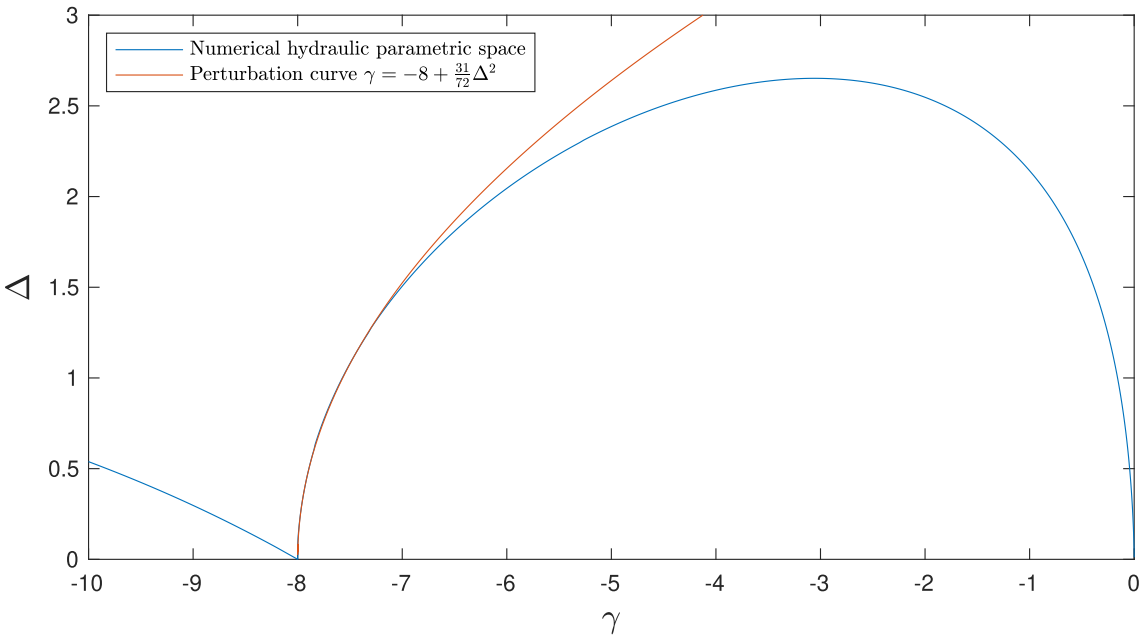

Figure 4.7: Plot of the numerical hydraulic parametric space $\gamma(\Delta)$ and the perturbation curve given by (4.23), near $\gamma = -8$

arc-length continuation, although solutions for very small Δ are available. This is likely due to the existence of a continuum of solutions on the line $\Delta = 0$ shown by KBB resulting in Newton's method being unable to converge.

An arch type behaviour is observed in the parametric space $\gamma(\Delta)$ between subsequent EDSW locations with a similar shape and height. There also appears to be a vertical turning point just before each negative eigenvalue (except at $\gamma = -8$), which had not been observed by Ee and Clarke. This is due to their minimisation algorithm being unable to handle vertical turning points. Pseudo arc-length has no such issues with traversing turning points. This is also the likely explanation for the small discrepancy at the second solitary wave location between our results and Ee and Clarke.

4.4 Summary

In this chapter, we investigated solitary wave and hydraulic fall solutions to the steady fKdV equation. We commenced with solitary waves and determined the existence of both ADSW and EDSW solutions. A shooting method was used to locate several solitary waves. Numerical evidence suggests that there may exist an infinite set of EDSW solutions to the steady fKdV equation along $\Delta = 0$ as $\gamma \to -\infty$, corresponding to a nonlinear eigenvalue problem with eigenvalues γ and EDSW eigenfunctions.

Next we investigated hydraulic fall solutions and reviewed analytic results in literature for the small and large positive γ limit. Perturbation analysis was then used to generate

the parametric relationship for hydraulic fall solutions in the vicinity of the only known analytic solution.

We then turned to numerical methods. Numerical continuation is used to show the connection between EDSW and ADSW solutions, extending the results obtained by KBB. The connection between EDSW and ADSW solutions further suggests the existence of infinitely many types of ADSW continuum solutions.

Lastly, an improved numerical method has been developed to generate the extended hydraulic fall parameter space for solutions to the steady fKdV equation. This is accomplished using tabletop solutions obtained using twin topographies and a wave-drag constraint condition. Pseudo arc-length continuation is used with a SMG solver for each Newton iteration. Our method appears to be more numerically stable than the approach used by Ee and Clarke [34], and Cullen [27]. We showed that hydraulic fall solutions periodically evolve into EDSW solutions, with the hydraulic fall parametric relationship $\gamma(\Delta)$ follows a similar arch-type behaviour connecting subsequent EDSW.

We note that CG is significantly faster than SMG during each Newton iteration. This is to be expected since the problem is one-dimensional, and multigrid is rarely used in a one-dimensional setting. In the next chapter we investigate SMG when applied to the two-dimensional Dubreil-Jacotin–Long equation.

Chapter 5

The Dubriel-Jacotin–Long equation

5.1 Introduction

In this chapter we seek solutions using SMG to the DJL equation with a bottom topography under the Boussinesq approximation and constant background current, repeated here for the reader's convenience

$$\nabla^2\eta + \frac{N^2(z^* - \eta)}{U^2}\eta = 0, \tag{5.1}$$

with boundary conditions

$$\eta(x^*, H^*) = 0, \tag{5.2a}$$
$$\eta(x^*, d(x^*)) = d(x^*), \tag{5.2b}$$
$$\eta(\pm\infty, z^*) = 0. \tag{5.2c}$$

The derivation of the DJL equation with a bottom topography from the stratified Euler equations is outlined by Stastna [105] and Soontiens et al. [104]. Figure 5.1 illustrates the definition of η.

The squared buoyancy frequency is related to the density distribution ρ, the gravitational constant g and the background density ρ_0. Under the Boussinesq approximation, N^2 is given by the expression

$$N^2(z^*) = -\frac{g}{\rho_0}\frac{d\rho}{dz^*}(z^*). \tag{5.3}$$

Here we will assume density changes are small relative to the background density, and define the density function to be

$$\rho(z^*) = \rho_0(1 - \beta\overline{\rho}(z^*)), \tag{5.4}$$

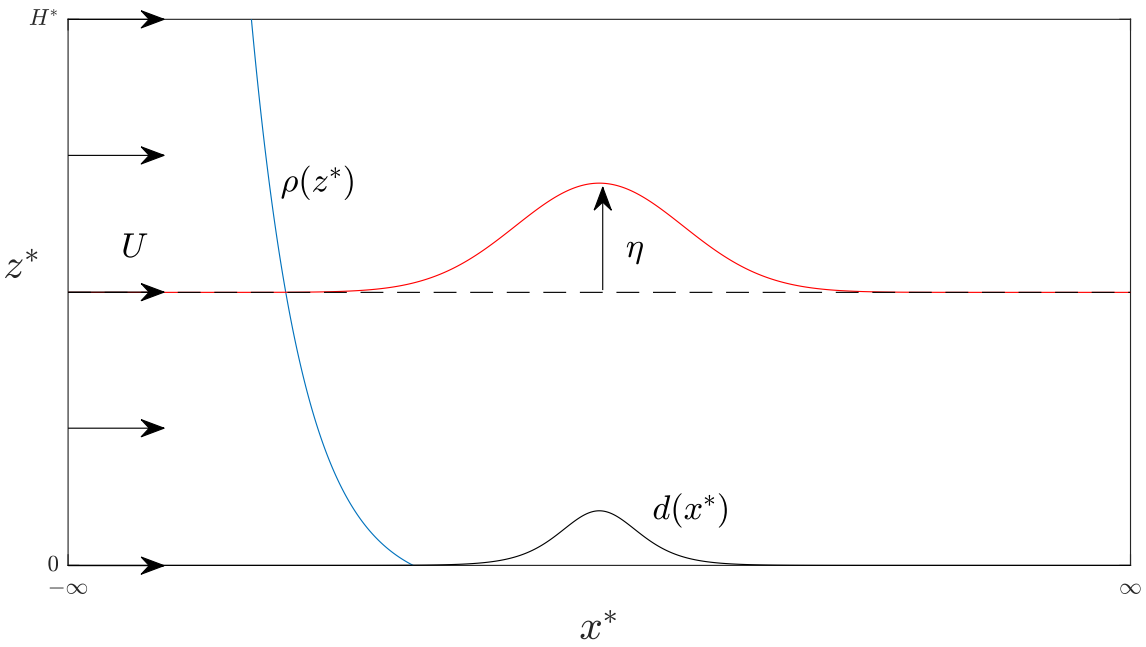

Figure 5.1: A single density contour $\rho(z^* - \eta)$ (red) drawn for a given density profile $\rho(z^*)$ (blue) over some topography $d(x^*)$ (black) with a constant background current U. The isopycnal displacement $\eta(x^*, z^*)$ is defined to be the distance the density contour is displaced from the far upstream or downstream value.

for a constant $0 < \beta \ll 1$ and a chosen function $\overline{\rho}(z^*)$. Note that $\rho'(z^*) < 0$ is required for stability, therefore this restricts $\overline{\rho}'(z^*) > 0$. This is simply because low density fluid must be above high density fluid for stability.

Next we non-dimensionalise the DJL equation by scaling all lengths by the height H^*. Let

$$\eta = H^*\hat{\zeta}, \quad z = H^*\hat{z}, \quad x^* = H^*\hat{x}, \quad d(x^*) = H^*b(\hat{x}). \tag{5.5}$$

Substituting this into equation (5.1) gives

$$\hat{\nabla}^2\hat{\zeta} + g\beta H^*\frac{\overline{\rho}'(H^*(\hat{z} - \hat{\zeta}))}{U^2}\hat{\zeta} = 0. \tag{5.6}$$

Define the reduced gravity

$$g' = \beta g, \tag{5.7}$$

the dimensionless velocity scale

$$\hat{u} = \frac{U}{\sqrt{g'H^*}} \tag{5.8}$$

and let

$$M^2(\hat{z}) = \overline{\rho}'(H^*\hat{z}) \tag{5.9}$$

to obtain the dimensionless DJL equation

$$\hat{\nabla}^2\hat{\zeta} + \frac{M^2(\hat{z}-\hat{\zeta})}{\hat{u}^2}\hat{\zeta} = 0 \tag{5.10}$$

with boundary conditions

$$\hat{\zeta} = 0 \quad \text{at} \quad \hat{z} = 1, \tag{5.11a}$$
$$\hat{\zeta} = \hat{b}(\hat{x}) \quad \text{at} \quad \hat{z} = \hat{b}(\hat{x}), \tag{5.11b}$$
$$\hat{\zeta} = 0 \quad \text{as} \quad \hat{x} \to \pm\infty. \tag{5.11c}$$

Next we look for perturbation solutions to the dimensionless DJL equation.

5.2 Perturbation series

Similar to the perturbation series used on the stratified Euler equations by Benney [7], in this section we consider small amplitude solutions to the system (5.10)–(5.11). Let ϵ be the isopycnal scale, μ^{-1} the topographic length scale, α the topographic height scale, and $\epsilon\Delta$ the perturbation to the long wave speed c. This scaling gives

$$\hat{\zeta} = \epsilon\zeta, \quad \hat{x} = \mu^{-1}x, \quad \hat{b} = \alpha b, \quad \hat{z} = z, \quad \hat{u} = c + \epsilon\Delta. \tag{5.12}$$

Substituting (5.12) into (5.10) gives

$$\zeta_{zz} + \mu^2\zeta_{xx} + \frac{M^2(z-\epsilon\zeta)}{(c+\epsilon\Delta)^2}\zeta = 0 \tag{5.13}$$

with boundary conditions

$$\zeta = 0 \quad \text{at} \quad z = 1, \tag{5.14a}$$
$$\zeta = \frac{\alpha}{\epsilon}b(x) \quad \text{at} \quad z = \alpha b(x), \tag{5.14b}$$
$$\zeta = 0 \quad \text{as} \quad x \to \pm\infty. \tag{5.14c}$$

We are interested in solutions where the dispersion term $\mu^2\zeta_{xx}$ and the boundary condition in (5.14b) are in balance to $O(\epsilon)$. For the term on the RHS of the boundary conditions to be $O(\epsilon)$ (note that b is $O(1)$ after scaling by α), we require that $\frac{\alpha}{\epsilon} = O(\epsilon)$ or $\alpha = O(\epsilon^2)$. Similarly for the term $\mu^2\zeta_{xx}$ to be $O(\epsilon)$, we require that $\mu^2 = O(\epsilon)$.

Next we substitute the binomial approximation

$$\frac{1}{(c+\epsilon\Delta)^2} \approx \frac{1}{c^2}\left(1 - \frac{2\epsilon\Delta}{c} + \dots\right), \tag{5.15}$$

and the Taylor series expansion

$$M^2(z-\epsilon\zeta) \approx M^2(z) - \epsilon\zeta(M^2(z))_z + \dots, \tag{5.16}$$

into (5.13) to obtain

$$\zeta_{zz} + \epsilon\zeta_{xx} + \frac{1}{c^2}\left(1 - \frac{2\epsilon\Delta}{c} + \dots\right)\left(M^2(z) - \epsilon\zeta(M^2(z))_z + \dots\right)\zeta = 0. \tag{5.17}$$

Finally, we expand ζ in a regular perturbation series

$$\zeta = \zeta^{(0)} + \epsilon\zeta^{(1)} + \dots, \tag{5.18}$$

and substitute into (5.17), while equating terms of similar order. Note that the boundary conditions can also be expanded in a Taylor series. The boundary condition

$$\zeta = \epsilon b \quad \text{on} \quad z = \epsilon^2 b \tag{5.19}$$

expanded around $z = 0$ gives

$$\zeta + \epsilon^2 b\zeta_z + \dots = \epsilon b \quad \text{on} \quad z = 0. \tag{5.20}$$

Substituting (5.18) in (5.20) gives the $O(1)$ boundary condition

$$\zeta^{(0)} = 0 \quad \text{on} \quad z = 0, \tag{5.21}$$

the $O(\epsilon)$ and $O(\epsilon^2)$ boundary condition

$$\zeta^{(1)} = b \quad \text{on} \quad z = 0 \tag{5.22}$$

and

$$\zeta^{(2)} + b\zeta_z^{(0)} = 0 \quad \text{on} \quad z = 0, \tag{5.23}$$

respectively.

Higher order boundary conditions can be similarly determined by equating higher order coefficients in (5.20).

5.2.1 Leading order

Equating terms of $O(1)$ in (5.17), we obtain the leading order equation

$$\zeta_{zz}^{(0)} + \frac{M^2(z)}{c^2}\zeta^{(0)} = 0, \tag{5.24}$$

with the leading order boundary conditions

$$\zeta^{(0)} = 0 \quad \text{on} \quad z = 1, \tag{5.25a}$$
$$\zeta^{(0)} = 0 \quad \text{on} \quad z = 0, \tag{5.25b}$$
$$\zeta^{(0)} = 0 \quad \text{as} \quad x \to \pm\infty. \tag{5.25c}$$

Since only z derivatives are present in the leading order equation, we use separation of variables. Let $\zeta^{(0)} = A(x)\phi(z)$. This results in the eigenvalue problem for ϕ also known as the *Taylor–Goldstein* equation

$$\phi_{zz} + \frac{M^2(z)}{c^2}\phi = 0, \tag{5.26}$$

with boundary conditions

$$\phi = 0 \quad \text{on} \quad z = 1, \tag{5.27a}$$
$$\phi = 0 \quad \text{on} \quad z = 0. \tag{5.27b}$$

This is a one-dimensional linear eigenvalue problem, where the $n = 1, \ldots, \infty$ eigenvalues and eigenfunctions, $1/c_n^2$ and $\phi^{(n)}$ respectively, can be found using a suitable eigenvalue solver. We shall refer to them as simply $1/c^2$ and ϕ here until we make use of eigenfunction expansion in the next section.

In order to solve for $A(x)$, we equate $O(\epsilon)$ terms in the perturbation series (5.17) to obtain

$$\zeta_{zz}^{(1)} + \frac{M^2(z)}{c^2}\zeta^{(0)} = -\zeta_{xx}^{(0)} + \frac{(M^2(z))_z}{c^2}(\zeta^{(0)})^2 + \frac{2\Delta}{c^3}M^2(z)\zeta^{(0)}, \tag{5.28}$$

with boundary conditions

$$\zeta^{(1)} = 0 \quad \text{on} \quad z = 1, \tag{5.29a}$$
$$\zeta^{(1)} = b(x) \quad \text{on} \quad z = 0, \tag{5.29b}$$
$$\zeta^{(1)} = 0 \quad \text{as} \quad x \to \pm\infty. \tag{5.29c}$$

We multiply (5.28) by ϕ and integrate over z to obtain the *compatibility equation* for $A(x)$

$$-A_{xx}\int_0^1 \phi^2\,dz - \frac{3}{2}A^2\int_0^1 \phi_z^3\,dz + \frac{2\Delta}{c}A\int_0^1 \phi_x^2\,dz = b(x)\phi_z(0). \tag{5.30}$$

Next we divide (5.30) through by $-\frac{2}{c}\int_0^1 \phi_x^2\,dz$ and use the substitution

$$r = \frac{3c\int_0^1 \phi_z^3\,dz}{2\int_0^1 \phi_z^2\,dz}, \quad s = \frac{c\int_0^1 \phi^2\,dz}{2\int_0^1 \phi_z^2\,dz}, \quad \gamma = \frac{c\int_0^1 \phi_z(0)\,dz}{2\int_0^1 \phi_z^2\,dz}, \tag{5.31}$$

to obtain the familiar steady fKdV equation

$$-\Delta A + \frac{r}{2}A^2 + sA_{xx} = -\gamma b(x), \tag{5.32}$$

with boundary conditions

$$A = 0 \quad \text{as} \quad x \to \pm\infty. \tag{5.33}$$

To transform the fKdV equation into the form (4.1) given in Chapter 4, we use the substitution $\chi = \sigma x$ and $A(x) = \frac{6s\mu^2}{r}B(\chi)$. The fKdV equation then becomes

$$-\Delta^* B + 3B^2 + B_{\chi\chi} = -\gamma^* b(\chi), \tag{5.34}$$

with boundary conditions

$$B = 0 \quad \text{as} \quad \chi \to \pm\infty, \tag{5.35}$$

where

$$\Delta^* = \frac{\Delta}{s\sigma^2} \tag{5.36}$$

and

$$\gamma^* = \frac{\gamma r}{6s^2\sigma^4}. \tag{5.37}$$

Using the fKdV equation in this form allows us to use the solutions generated in Chapter 4 to generate perturbation solutions to the DJL equation.

The eigenvalue solution $\phi(z)$ and the fKdV solution $A(x)$ then combine to give the leading order perturbation $\zeta^{(0)}$ to the DJL equation

$$\zeta^{(0)} = A(x)\phi(z) = \frac{6s\mu^2}{r}B(\chi)\phi(z). \tag{5.38}$$

Note that the leading order solution does not contain boundary conditions due to the topography. It does however, have topography contributions from the fKdV solution.

5.2.2 Order ϵ

Next we equate $O(\epsilon)$ terms in the DJL perturbation series (5.17). Since the $\zeta^{(1)}$ boundary conditions are inhomogeneous, we let $\zeta^{(1)} = \Phi + B(x)$ where $B(x) = b(x)(1-z)$ giving the equation

$$\Phi_{zz} + \frac{M^2(z)}{c^2}\Phi = -\frac{M^2(z)}{c^2}b(x)(1-z) - \zeta^{(0)}_{xx} + \frac{(M^2(z))_z}{c^2}(\zeta^{(0)})^2 + \frac{2\Delta}{c^3}M^2(z)\zeta^{(0)}, \tag{5.39}$$

with homogeneous and asymptotic boundary conditions

$$\Phi = 0 \quad \text{on} \quad z = 1, \tag{5.40a}$$

$$\Phi = 0 \quad \text{on} \quad z = 0, \tag{5.40b}$$

$$\Phi = 0 \quad \text{as} \quad x \to \pm\infty. \tag{5.40c}$$

Equation (5.39) is then solved using eigenfunction expansions. Let m be the mode of interest, and express Φ as the eigenfunction expansion in $\phi^{(n)}$ given by $\Phi = \sum_{n=0}^{\infty} a_n \phi^{(n)}$. Now let $\zeta^{(0)} = A\phi^{(m)}$, $c = c_m$ and $\phi = \phi^{(m)}$. Next we multiply (5.39) by $\phi^{(m)}$ and integrate over z, while noting the orthogonality condition from Sturm–Liouville theory similarly used by Benney [7]

$$\int_0^1 M^2(z)\phi^{(n)}\phi^{(m)}dz = 0 \quad \text{if} \quad n \neq m. \tag{5.41}$$

Equation (5.39) then simplifies to

$$\frac{c_n^2 - c_m^2}{c_m^2} a_n \int_0^1 (\phi_z^{(n)})^2 dz = -b(x)\frac{c_n^2}{c_m^2}\phi_z^{(n)}(0) - A_{xx}\int_0^1 \phi^{(n)}\phi^{(m)}dz + A^2\left(\frac{c_m^2}{2c_n^2} - 2\right)\int_0^1 (\phi_z^{(m)})^2\phi_z^{(n)}dz. \tag{5.42}$$

If $n = m$ we recover the compatibility equation (5.30) as expected. For $n \neq m$, the solution to the expansion coefficients a_n are

$$a_n = -b(x)\frac{c_n^2}{c_n^2 - c_m^2}\frac{\phi_z(0)}{\int_0^1(\phi_z^{(n)})^2dz} - A_{xx}\frac{\int_0^1 \phi_m\phi_n dz}{\int_0^1(\phi_z^{(n)})^2dz}\frac{c_m^2}{c_n^2 - c_m^2} + A^2\left(\frac{c_m^2}{2c_n^2} - 2\right)\frac{\int_0^1(\phi_z^{(n)})^2\phi_z^{(n)}dz}{\int_0^1(\phi_z^{(n)})^2dz}\frac{c_m^2}{c_n^2 - c_m^2}. \tag{5.43}$$

The $O(\epsilon)$ perturbation $\zeta^{(1)}$ is then simply

$$\zeta^{(1)} = b(x)(1 - z) + \sum_{\substack{n=0 \\ n\neq m}}^{\infty} a_n\phi^{(n)}. \tag{5.44}$$

In summary, the DJL perturbation series to $O(\epsilon)$ is

$$\begin{aligned}\zeta &\approx \zeta^{(0)} + \epsilon\zeta^{(1)} \\ &= A(x)\phi^{(m)} + \epsilon\left(b(x)(1 - z) + \sum_{\substack{n=0 \\ n\neq m}}^{\infty} a_n\phi^{(n)}\right)\end{aligned} \tag{5.45}$$

Note that the $\phi^{(m)}$ mode is $O(1)$ while $\phi^{(n)}$ for $n \neq m$ modes are of $O(\epsilon)$.

Subsequent higher order approximations can similarly be derived by equating higher order terms in the perturbation expansion (5.17) with the appropriate boundary conditions found using Taylor series. Taking to $O(\epsilon^2)$ and higher, the compatibility equation then becomes variants of the extended fKdV equation.

5.3 DJL Numerical Method

In the perturbation expansion, it is beneficial to work with equation (5.32) in the dimensionless DJL coordinates. The transformation back to these are given by

$$\hat{A} = \epsilon A, \quad \hat{\Delta} = \epsilon \Delta, \quad \hat{b} = \alpha b, \quad \hat{x} = \mu^{-1} x. \tag{5.46}$$

Substituting this in equation (5.32) and let $\hat{r} = r$, $\hat{s} = \frac{s\epsilon}{\mu^2}$ and $\hat{\gamma} = \frac{\gamma\epsilon^2}{\alpha}$, we obtain

$$-\hat{\Delta}\hat{A} + \frac{\hat{r}}{2}\hat{A}^2 + \hat{s}\hat{A}_{\hat{x}\hat{x}} = -\hat{\gamma} b. \tag{5.47}$$

Since $\alpha = O(\epsilon^2)$, and $\mu^2 = O(\epsilon)$ from the assumption made in the perturbation series, we approximate $\epsilon \simeq \mu^2$ and $\epsilon^2 \simeq \alpha$, or equivalently

$$\mu \simeq \alpha^4. \tag{5.48}$$

Equation (5.47) is then transformed to the familiar form

$$-\hat{\Delta}^* \hat{B} + 3\hat{B}^2 + \hat{B}_{\hat{\chi}\hat{\chi}} = -\hat{\gamma}^* \hat{b}(\hat{\chi}), \tag{5.49}$$

via the substitution

$$\hat{A} = \frac{6\hat{s}\mu^2}{\hat{r}} \hat{B}, \tag{5.50}$$

where $\hat{\chi} = \mu\hat{x}$ and

$$\hat{\Delta}^* = \frac{\hat{\Delta}}{\hat{s}\mu^2}, \tag{5.51}$$

$$\hat{\gamma}^* = \frac{\hat{\gamma}\hat{r}\alpha}{6\hat{s}^2\mu^4}. \tag{5.52}$$

Working in the dimensionless DJL coordinates has the advantage of removing the dependence of extra scaling parameters ϵ and σ that appeared in the scaled DJL coordinates. In the dimensionless DJL coordinates, the long wave speed perturbation is given by

$$u = c + \hat{\Delta}, \tag{5.53}$$

and the perturbation solution is given by

$$\hat{\zeta} \approx \hat{\zeta}^{(0)} + \hat{\zeta}^{(1)} + \dots . \tag{5.54}$$

The accuracy of the perturbation solution depends on two conditions. First is that Δ is required to be a small parameter, and second is that the approximation (5.48) must hold. Subsequent sections only deal with dimensionless DJL coordinates, and the hat notation is dropped. From here we will denote the perturbation solution as the weakly nonlinear solution (WNL).

To construct the WNL, the Chebyshev collocation method is used in the z domain and MATLAB's inbuilt `eig` function is used to numerically solve the linear eigenvalue equation (5.26) for ϕ. The fKdV equation is solved using the same methodology as Chapter 4 using Fourier collocation methods in x with grid points matching that of the DJL discretisation.

The eigenvalue equation (5.26) for ϕ only depends on the squared buoyancy frequency M^2. This remains true for DJL solutions with or without topographic forcing. We consider three different stratifications given in Table 5.1, which are similar to those used by Brown et al. [16]. Each stratification is also normalised by the constant a to ensure $\int_0^1 M^2(\eta)\, d\eta = 1$.

Stratification	$M^2(\eta)$	a
Linear	$a\eta$	2
Exponential	ae^{η}	$\frac{1}{e-1}$
Sech^2	$a\,\mathrm{sech}^2(\eta - 1)$	$\frac{1}{\tanh(1)}$

Table 5.1: Table of stratifications and normalising constant used in the numerical experiments.

The eigenvalues are found numerically in a finite computational domain with N_z points, thus we obtain N_z eigenvalue-eigenvector pairs. We take $N_z/2$ terms to the Nyquist frequency in the summation (5.45) to obtain

$$\begin{aligned} \zeta &\approx \zeta^{(0)} + \epsilon\zeta^{(1)} \\ &= A(x)\phi^{(m)} + \epsilon \left(b(x)(1-z) + \sum_{\substack{n=0 \\ n\neq m}}^{N_z/2} a_n \phi^{(n)} \right). \end{aligned} \tag{5.55}$$

This is repeated for a higher and lower resolution to ensure spurious eigenvalues are filtered.

In order to use the WNL as an initial guess in Newton's method, two-dimensional interpolation is required. This is because it is defined on a rectangular domain while the forced DJL one has a bottom topography. Let the WNL domain be denoted as (x', z') where $x' \in [-L, L)$, $z' \in [0, 1]$, the forced DJL domain be denoted as (x, z) where $x \in [-L, L)$, $z \in [\alpha b(x), 1]$, and the conformal mapping domain discussed in Chapter 2 be denoted as (u, v) where $u \in [-L, L)$ and $z \in [0, H_w]$. Note that if no topography is present, (x', z'), (x, z) and (u, v) are equivalent. In the case of a bottom topography, the WNL domain points on (x', z'), which are the spectral collocation nodes, must be interpolated to the points in $(x(u, v), z(u, v))$, before the WNL can be used as the initial guess in Newton's method. Interpolation is performed numerically using two-dimensional splines. The boundary condition $\zeta = \alpha b(x)$ on $z = \alpha b(x)$ is then applied after interpolation. This causes discontinuities, which is remedied by introducing a boundary layer between $z = \alpha b(x)$ and $z = \alpha b(x) + 0.05$ where spline interpolation is used to smooth the solution. While this removes discontinuities, it does not improve the performance of Newton's method.

In the case of a positive forcing, the initial guess on (x', z') consists of some unused and redundant nodes, namely the nodes located between $z = \alpha b(x)$ and $z = 0$. This does not pose a problem since the nodes can easily be discarded during interpolation.

However, in the case of negative forcing, there exists missing initial data, where the hole is located, specifically in the region between $z = \alpha b(x)$ and $z = 0$. Interpolation from the boundary is used to remedy this. The missing nodes in the WNL can be found by interpolating the boundary condition $\zeta(x, \alpha b(x)) = \alpha b(x)$ with the internal points on $z \in [0, 1]$. However, spline interpolation cannot be used in this case since the derivatives at $\zeta(x, \alpha b(x))$ are unknown. Modified Akima (makima) interpolation is used instead. The interpolation from the boundary performs well for small negative α with small number missing data, but becomes problematic for larger negative values of α since the interpolation from the boundary is akin to extrapolation.

In the case of no topographic forcing, we solve the dimensionless DJL equation (5.10) with boundary conditions (5.11). The $x \in [-L/2, L/2)$ domain is discretised using Fourier collocation methods, with the asymptotic boundary conditions in (5.11c) approximated by periodic boundary conditions for a sufficiently large chosen L. The $z \in [0, 1]$ domain is discretised using Chebyshev collocation methods.

In the case of topographic forcing, the DJL equation is solved in the (u, v) domain with Fourier collocation methods in $u \in [-L/2, L/2)$ and Chebyshev collocation methods in $v \in [0, H_w]$. Note H_w is determined through the conformal mapping algorithm. The

DJL equation in the (u, v) coordinates is given by

$$\frac{1}{J(u,v)}\nabla^2\zeta + \frac{M^2(z(u,v)-\zeta)}{u^2}\zeta = 0, \tag{5.56}$$

where $J(u, v)$ is the Jacobian, which can be simplified using the Cauchy–Riemann conditions to give

$$J(u,v) = \left(\frac{\partial z}{\partial v}\right)^2 + \left(\frac{\partial z}{\partial u}\right)^2. \tag{5.57}$$

The boundary conditions are

$$\zeta = 0 \quad \text{at} \quad v = H_w, \tag{5.58a}$$

$$\zeta = \alpha b(x(u,v)) \quad \text{at} \quad v = 0, \tag{5.58b}$$

with periodic boundary conditions in x. The numerical conformal mapping is then used to map the rectangular (u, v) grid to the physical (x, z) grid.

For each stratification in Table 5.1, a small Δ is initially chosen to obtain the $O(1)$ and $O(\epsilon)$ WNL. Newton's method is used on the DJL equation (5.10) with no topography or (5.56) with topography where the initial guess $\overline{\zeta}$ is taken to be the $O(\epsilon)$ WNL

$$\overline{\zeta} \approx \zeta^{(0)} + \zeta^{(1)}, \tag{5.59}$$

interpolated from the (x', z') grid to the (x, z) grid. This is performed using MATLAB's inbuilt `interp2` function. The tolerance in Newton's method is set to 10^{-7}. For all DJL solutions, one FMG cycle consisting of one PMRR relaxation on the up and down cycle, with second-order FD preconditioning, is used within each Newton iteration. GMRES is used as coarse grid solver at the grid with $2^3 \times 2^2$ grid points. This is chosen over BiCG and BiCGSTAB in Chapter 3 due to its increased robustness.

Our results only consider valid DJL solutions before overturning. To determine overturning, we check for vertical isopycnals with the following condition

$$\frac{\partial}{\partial z}(z - \zeta) < 0, \tag{5.60}$$

which is equivalent to

$$\frac{\partial \zeta}{\partial z} > 1, \tag{5.61}$$

or

$$\frac{\partial(x,\zeta)}{\partial(x,z)} > 1, \tag{5.62}$$

in the (u, v) coordinates using the chain rule, where $\partial(x, \zeta) = \frac{\partial x}{\partial u}\frac{\partial \zeta}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial \zeta}{\partial u}$. Similar overturning predictions can be obtained for the WNL by replacing ζ with the perturbation

solution $\zeta^{(0)}$ or $\zeta^{(0)}+\zeta^{(1)}$, and z with z' in the overturning condition (5.61). Thus we can compare the overturning predictions from the WNL to the overturning location in the DJL. Stastna and Lamb [106] showed the maximum wave amplitude occurred at one of three possibilities: The onset of wave breaking or overturning, conjugate flow amplitude or the failure associated with shear instability. In our numerical experiments, conjugate flow and shear instability did not occur, hence the overturning point also corresponds to the maximum wave amplitude.

Once a valid DJL solution is obtained, numerical continuation is used with the parameter Δ until overturning is reached. The same tolerance of 10^{-7} is set for the Newton correction. The WNL continuation, is obtained from a pseudo arc-length continuation in the fKdV equation with parameter Δ^*. The conversion from Δ^* to Δ can be easily determined through (5.51).

The DJL and WNL continuation is then compared with a plot of momentum versus Δ, where the momentum P for a DJL solution is defined by the integral over the forced DJL domain D. This integral is more easily computed in the rectangular domain D',

$$P = \iint_D \zeta^2(x,z)\,dx\,dz = \iint_{D'} |J(u,v)|\,\zeta^2(u,v)\,du\,dv. \tag{5.63}$$

There are two main methods to compute the momentum for the WNL. The first involves taking the integral over the WNL domain D''

$$P = \iint_{D''} (\zeta^{(0)}(x',z'))^2\,dx'\,dz'. \tag{5.64}$$

This has the advantage of being simple to compute since D'' is a simple rectangular domain. However, since the domain D'' is not physical, the integral includes (or omits in the case of negative topography) non-physical points where the topography lies. We shall call this source of error *Boundary Error*.

The second method is to compute the integral in the forced DJL domain using (5.63) but with ζ replaced with $\zeta^{(0)}$ or $\zeta^{(1)}$ for $O(1)$ and $O(\epsilon)$ WNL, respectively. This, however, requires interpolation of the (x',z') points to the $(x(u,v),z(u,v))$ points before integrating in the D' domain. This method introduces *Interpolation Error* in the computation. Furthermore, the derivative calculation for overturning in (5.61) becomes problematic for negative bump topography due to errors in the interpolated points.

The two methods produce very close results in the positive bump topography case, with the momentum difference in the order of 10^{-7}. However, in the negative bump topography case, it is advantageous to compute the integral using the first method due to

the large source of interpolation error. In the rest of the chapter, we choose to compute the WNL momentum using the second method for positive, and the first method for negative bump topography.

5.4 DJL Numerical Results

5.4.1 No topography

We begin by investigating solutions to the DJL equation with no topographic forcing, corresponding to $b(x) = 0$. With no topography, the topographic length scale μ is arbitrary and conformal mapping is not required. For simplicity we set $\mu = 1$.

The compatibility equation simplifies from the fKdV equation to the KdV equation. The exact solitary wave solution to the KdV equation is well known and can be found by assuming the solitary wave solution in the form $B = B_0 \operatorname{sech}^2 \theta x$. Substituting into (5.49) and solving for the unknowns B_0 and θ gives

$$B = \frac{\Delta^*}{2} \operatorname{sech}^2 \left(\frac{\sqrt{\Delta^*}}{2} x \right). \tag{5.65}$$

Upon using (5.50) and (5.51), the solution for A reduces to

$$A = \frac{3\Delta}{r} \operatorname{sech}^2 \left(\frac{1}{2} \sqrt{\frac{\Delta}{s}} x \right). \tag{5.66}$$

The $O(1)$ WNL is then given by

$$\zeta^{(0)} = A(x)\phi(z) = \frac{3\Delta}{r} \operatorname{sech}^2 \left(\sqrt{\frac{\Delta}{s}} \frac{x}{2} \right) \phi. \tag{5.67}$$

The $O(\epsilon)$ WNL is found using (5.43) and (5.44) with $b(x) = 0$. Note that with no topography, the it only depend on the parameter Δ.

Numerical continuation of the KdV solution is not required for the WNL continuation since a closed form solution is known. A grid resolution of $2^6 \times (2^6 + 1)$ is used with the domain size L in $x \in [-L/2, L/2)$ chosen so that the KdV solution has a relative tail tolerance of 10^{-8}, that is $B(\pm L/2)/B(0) = 10^{-8}$. This gives

$$L = \frac{4}{\sqrt{\Delta^*}} \operatorname{sech}^{-1}(\sqrt{10^{-8}}). \tag{5.68}$$

The choice of domain then remains fixed during continuation.

Isopycnals, that is the contours of $z - \zeta$, for the sech^2 stratification for mode 1 and 2 is shown in Figure 5.2. Since these are the isopycnals just before the overturning point, some isopycnals are near vertical.

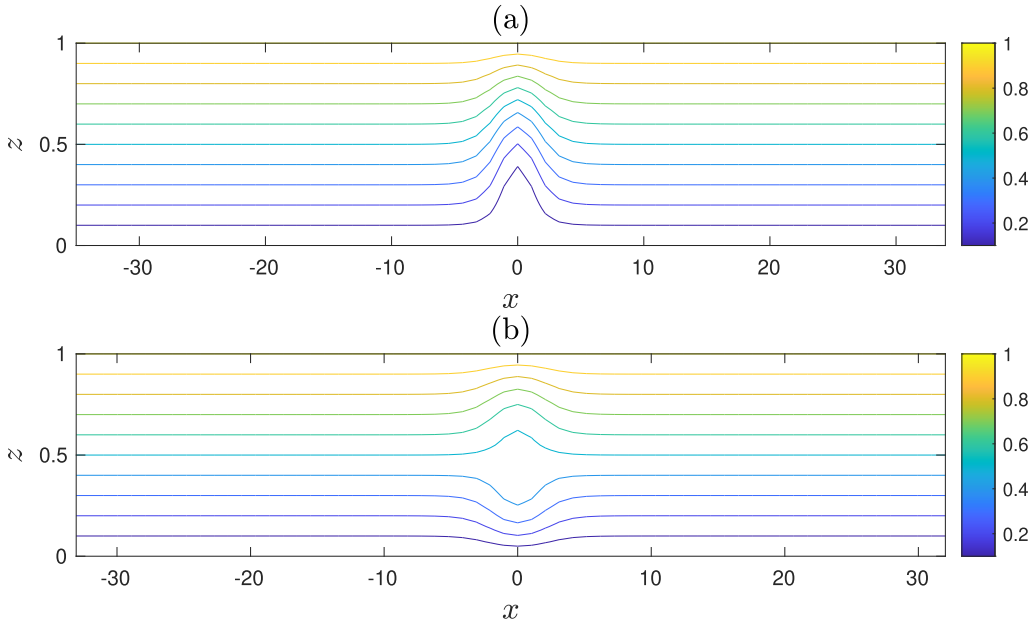

Figure 5.2: Isopycnal just before overturning for sech^2 stratification in Table 5.1 for mode 1 (a) and mode 2 (b).

In the continuation plots in Figure 5.3 and 5.4, we see for small Δ/c, WNL provides a good approximation to the true DJL solutions for all stratifications. However, as Δ/c increases, a larger divergence is seen between the two. This is to be expected since the former uses a perturbation expansion in Δ. Interestingly, the difference between $O(1)$ and $O(\epsilon)$ WNL continuation is minimal except the overturning location, where the latter predicts it at a larger Δ compared to the former.

Figure 5.4 shows mode 2 solitary waves have significantly faster divergence between the WNL and DJL solutions than mode 1. Note that Newton's method failed to converge for the mode 2 exponential stratification. The difficulty in mode 2 convergence, here and in later sections with topography forcing, of Newton's method is likely explained by the asymptotics beyond all order [1] phenomena, resulting in mode 2 waves develop oscillatory tails consisting of lower mode short waves, hence the inaccuracy of WNL theory.

5.4.2 Single bump topography

Next we investigate the effect of a single bump topography $b(\chi) = \text{sech}^2(\chi)$. In this case we have five parameters and three equations. The five parameters are γ^* and Δ^* the location in parameter space of the single bump fKdV solution, μ the topographic length

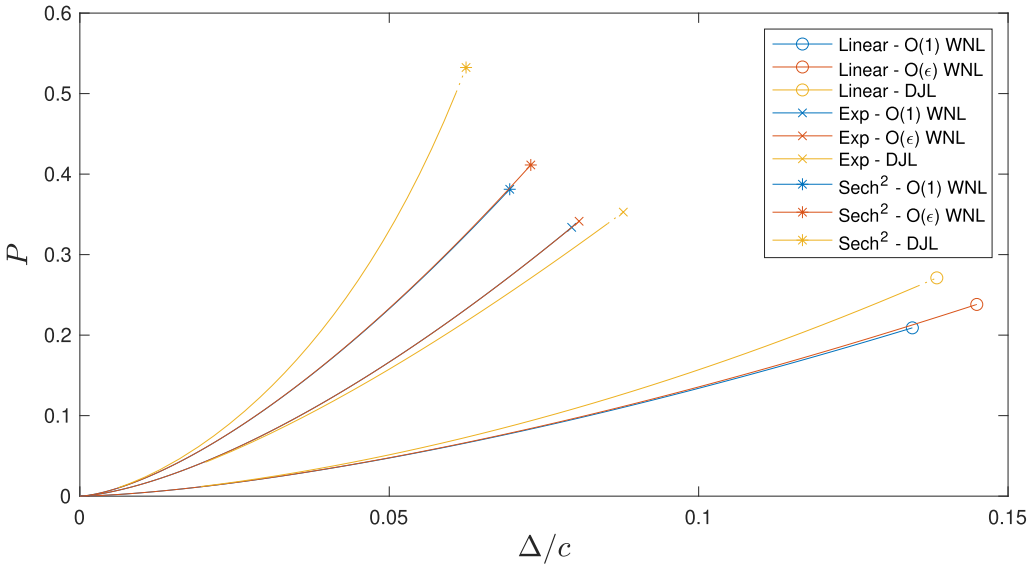

Figure 5.3: Plot of momentum P vs normalised wave speed Δ/c for mode 1 solitary waves for $O(1)$ and $O(\epsilon)$ WNL and DJL solutions with linear, exponential and sech^2 stratification in Table 5.1. Solid lines show the continuation curves. Plot markers $\circ, \times, *$ show the first overturning solution obtained in the continuation. Dashed lines, barely visible between the solid lines and the marker, indicate the gap between the last valid DJL solution and the first overturning solution obtained during continuation.

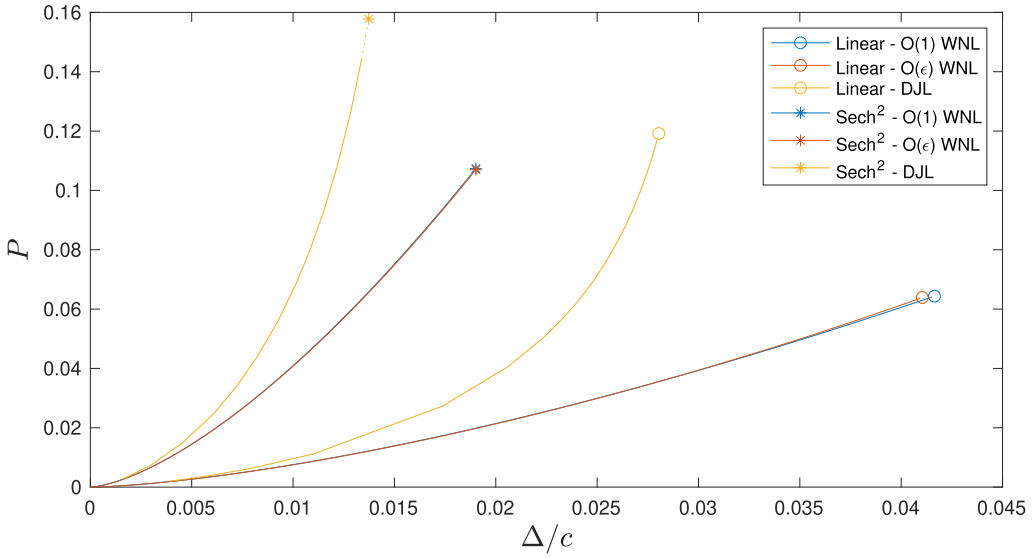

Figure 5.4: Plot of momentum P vs normalised wave speed Δ/c for mode 2 solitary waves for $O(1)$ and $O(\epsilon)$ WNL and DJL solutions with linear and sech^2 stratification in Table 5.1. Solid lines show the continuation curves. Plot markers $\circ, *$ show the first overturning solution obtained in the continuation. Dashed lines, barely visible between the solid lines and the marker, indicate the gap between the last valid DJL solution and the first overturning solution obtained during continuation.

scale, Δ the perturbation to the long wave speed, and α the topography height. The three equations are for Δ^* and γ^* given in (5.51) and (5.52), respectively, and

$$\Delta^* = g(\gamma^*), \tag{5.69}$$

where g is the relationship between Δ^* and γ^* for the existence of a fKdV solitary wave solution. Examples of the relationship in (5.69), also known as bifurcation diagrams, for single bump fKdV solutions can be found in the paper by Lee and Whang [76].

There are two free parameters available. It is advantageous to one of them as Δ^* which then determines γ^*, so that fKdV particular solution types can be targeted. Since the continuation uses Δ, the second parameter is chosen to be the topographic length scale μ, which then determines the topography height α through (5.52). This allows the two parameters μ and α related to the topography to be fixed during continuation, so that only one conformal mapping is required for each chosen μ.

The fKdV domain size is set to $L = 40$, which corresponds to $\chi \in [-20, 20)$. It is important to ensure that the domain size is large enough to approximate the asymptotic boundary conditions with periodicity. This is true for both the fKdV and DJL equations. While larger domain size does not affect the solution obtained, it requires higher resolution. Therefore, it is advantageous to pick a domain just large enough so that the tail ends of the solitary wave solution reach a certain acceptable tolerance. The fKdV domain size chosen results in a tolerance of approximately 10^{-7} for the tail end of the fKdV solution.

We begin by generating the parameter space (5.69) for a single bump fKdV solitary wave using (5.49). Here we fix a range of Δ^* values and perform a continuation in γ^*.

The results in Figure 5.5 are very similar to those obtained by Cullen [27], who used a $\cos^4$ shape topography while varying the topography length scale. We choose the continuation at $\Delta^* = 1.5$. Along this curve there are three types of solutions: a negative bump, a positive bump, and a transition between one to two positive bump solitary waves as shown in Figure 5.6. Picking $\gamma^* = -0.5$ and $\gamma^* = 0.25$ enables us to capture the three solution types, as well as positive and negative bump topography. Thus, we have a total of three solutions for $\gamma^* = -0.5$ and two for $\gamma^* = 0.25$.

Each of the five solutions are separately chosen as the initial WNL guess used in Newton's method. For each case we choose $\mu = 0.7, 0.8$ and 0.9. Smaller values of μ correspond to a large topography length, and thus require higher resolution. Furthermore, small μ results in small α, which then closely resemble the no-topography case. Values of μ cannot exceed 1 since that would violate our assumptions on the perturbation expansion

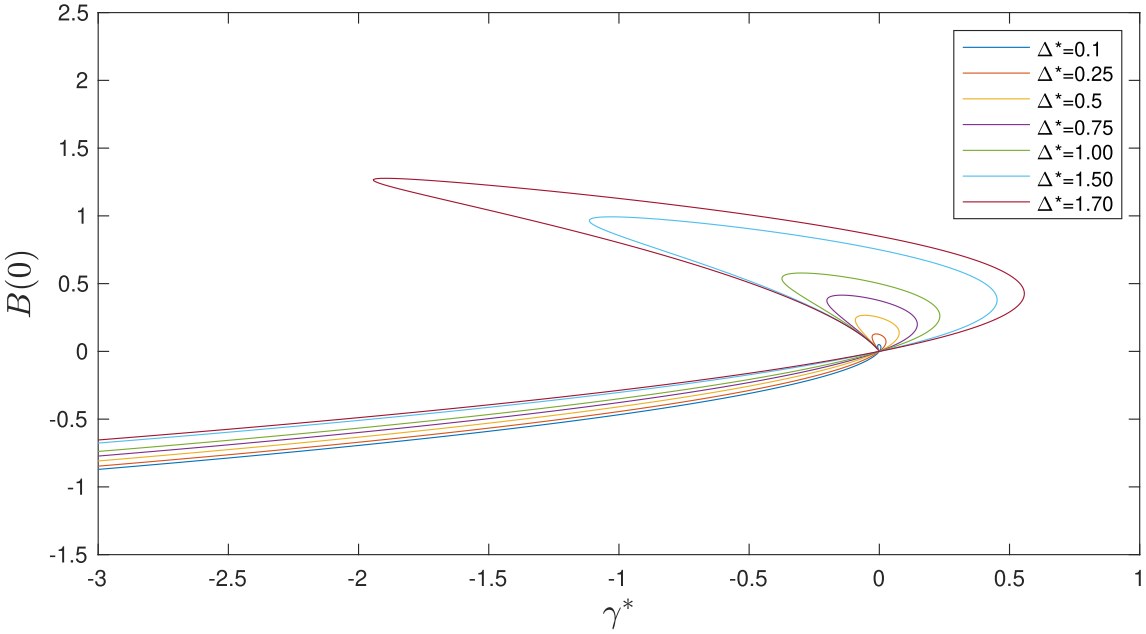

Figure 5.5: Single bump fKdV continuation in γ^* for a range of Δ^* values.

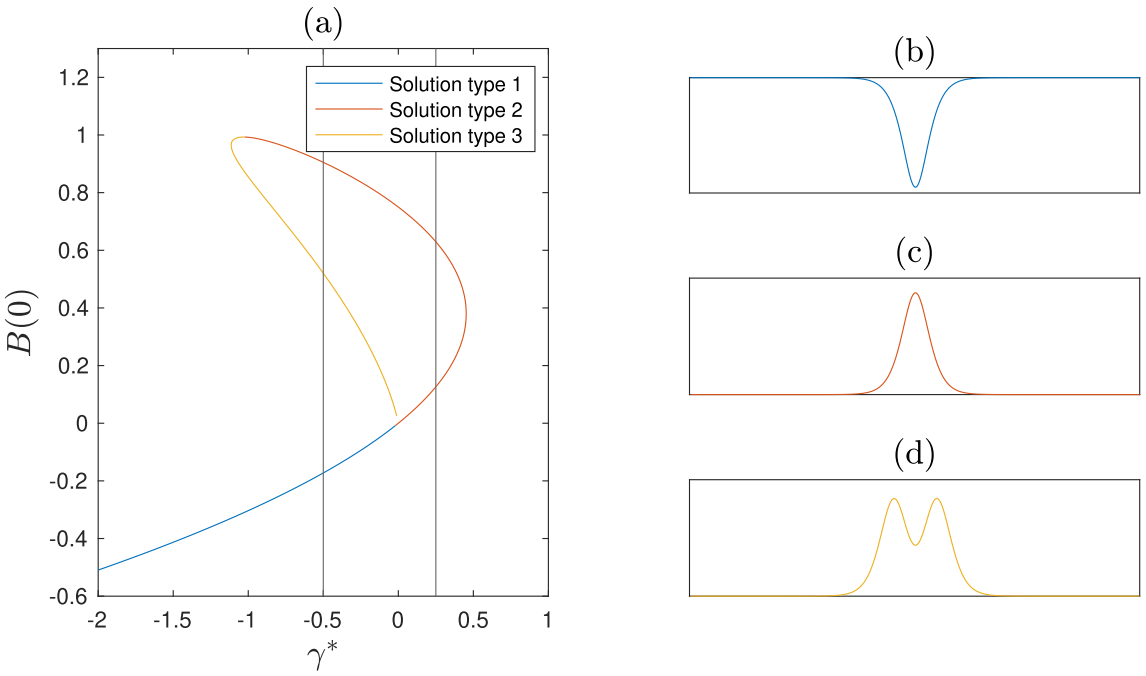

Figure 5.6: Single bump fKdV continuation in γ^* at $\Delta^* = 1.5$ with vertical lines at the chosen $\gamma^* = -0.5$ and $\gamma^* = 0.25$ (a). Plot of solitary wave solution types 1 (b), 2 (c) and 3 (d).

(5.48).

Other parameters α and Δ are calculated using (5.51) and (5.52) rearranged for the chosen variable as

$$\Delta = s\mu^2\Delta^*, \tag{5.70a}$$

$$\alpha = \frac{6\Delta^2\gamma^*}{\gamma r(\Delta^*)^2}. \tag{5.70b}$$

It is important to check that Δ remains small for the perturbation solution to be valid.

Once a DJL solution is obtained from Newton's method, similar to the no-topography case, we perform numerical continuation on the parameter Δ until overturning is reached. However, since no closed-form solution exists for the single bump fKdV equation, numerical continuation in Δ^* for a fixed γ^* is used to obtain the WNL continuation. The fKdV continuation in Δ^* can be compared to the DJL one in Δ through equation (5.51) and (5.38) for the $O(1)$ or (5.45) for the $O(\epsilon)$ WNL. Given the existence of saddle node bifurcations, for both the fKdV and DJL, pseudo arc-length is used over natural parameter continuation. A grid resolution of $2^9\times(2^8+1)$ is used for all single bump DJL solutions.

For our choice of parameters Δ^*, γ^* and μ, the ratio μ^4/α remains constant throughout continuation. The α values for region 1 and 3 are identical, and are shown in Table 5.2, while the α values for region 2 are shown in Table 5.3.

$M^2(\eta)$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$
2η	4.57×10^{-3}	7.79×10^{-3}	1.25×10^{-2}
$\frac{1}{e-1}e^{\eta}$	7.64×10^{-3}	1.30×10^{-2}	2.09×10^{-2}
$\frac{1}{\tanh(1)}\operatorname{sech}^2(\eta-1)$	8.99×10^{-3}	1.53×10^{-2}	2.46×10^{-2}

Table 5.2: Table of α values to two decimal places for $\gamma^* = -0.5$ at regions 1 and 3 for choices of different stratifications and μ values.

$M^2(\eta)$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$
2η	-2.28×10^{-3}	-3.90×10^{-3}	-6.24×10^{-3}
$\frac{1}{e-1}e^{\eta}$	-3.82×10^{-3}	-6.51×10^{-3}	-1.04×10^{-2}
$\frac{1}{\tanh(1)}\operatorname{sech}^2(\eta-1)$	-4.50×10^{-3}	-7.67×10^{-3}	-1.23×10^{-2}

Table 5.3: Table of α values to two decimal places for $\gamma^* = 0.25$ at region 2 for choices of different stratifications and μ values.

Despite there being five different solutions from the chosen Δ^* and γ^* values, we find

that each solution type in the fKdV parameter space belongs to the same WNL and DJL continuation curve. Therefore, while there are three WNL and DJL solutions on the branch $\gamma^* = -0.5$, there are only two continuation curves. Similarly, the two WNL and DJL solutions on the branch $\gamma^* = 0.25$ correspond to the same continuation curve. We shall denote these as the region 1, region 2 and region 3 DJL continuation, respectively.

Only mode 1 DJL solutions are considered in the single bump topography case since Newton's method failed to converge for mode 2.

In region 1, Figure 5.7 shows the $O(1)$ WNL is a better approximation to the DJL solution as $\Delta/c \to 0$ whereas the $O(\epsilon)$ solution is more accurate as $\Delta/c \to \infty$. This is likely due to the better boundary condition approximation used in the $O(\epsilon)$ WNL.

In Figure 5.8 or 5.9, in log scale, we see that with linear stratification, the WNL is an excellent approximation to the DJL solution for small Δ/c, with deviations increasing with Δ/c for all μ values in region 2. Additionally, the $O(\epsilon)$ predicts the DJL overturning location more accurately than the $O(1)$ WNL. However, for the more nonlinear exponential and sech^2 stratification, they diverge significantly, and more so as μ increases.

In region 3, Figure 5.10 shows that WNL approximates the DJL solution well only for $\mu = 0.7$, and when Δ/c is small. As μ and/or Δ/c increase, they diverge further. In this region, WNL are generally poor approximations to the DJL counterparts, likely explaining why the more nonlinear sech^2 compared to exponential stratification failed to converge for this region.

The isopycnals for the exp stratification at $\mu = 0.9$ at various locations in region 1,2 and 3 are shown in Figure 5.11.

5.4.3 Two bump topography

In the case of two bump topography we use the tabletop solutions obtained from the fKdV hydraulic fall parameter space given in Figure 4.6. The bottom topography is of the form

$$b(\chi) = \text{sech}^2(\chi - h) + \text{sech}^2(\chi + h). \tag{5.71}$$

The parameters chosen in the fKdV tabletop solutions are $L = 50$, so that $\chi \in [-25, 25)$, and $h = 12$.

Similar to the single bump solutions, there are five parameters γ^*, Δ^*, μ, Δ, α and three equations (5.69), (5.51) and (5.52). We also consider Δ^* and μ as the freely chosen

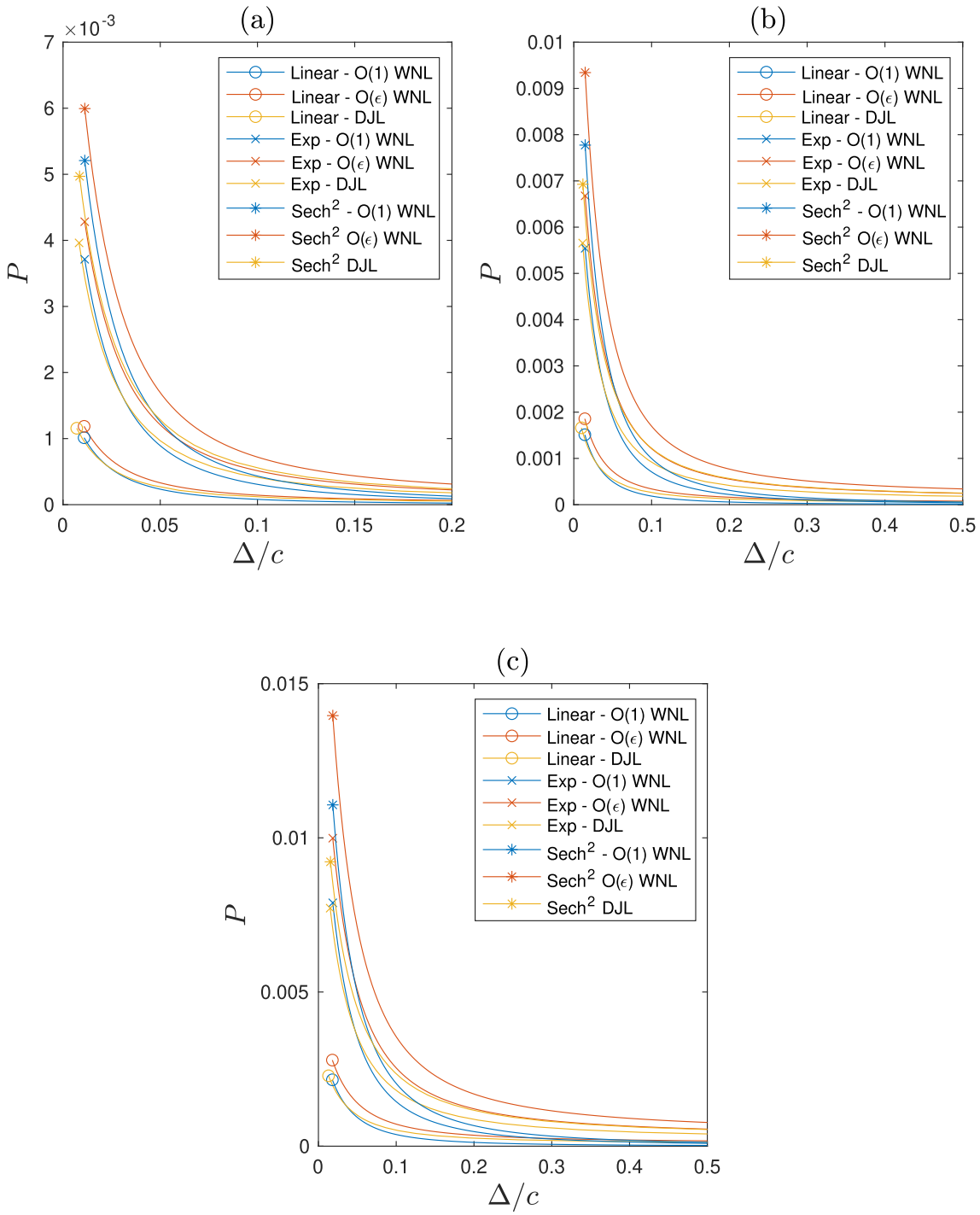

Figure 5.7: Region 1 DJL and WNL continuation for linear, exponential and sech^2 stratifications for $\mu = 0.7$ (a) $\mu = 0.8$ (b) and $\mu = 0.9$ (c). The symbols $\circ \times *$ show the location where continuation halts due to minimum step size of 10^{-6} reached.

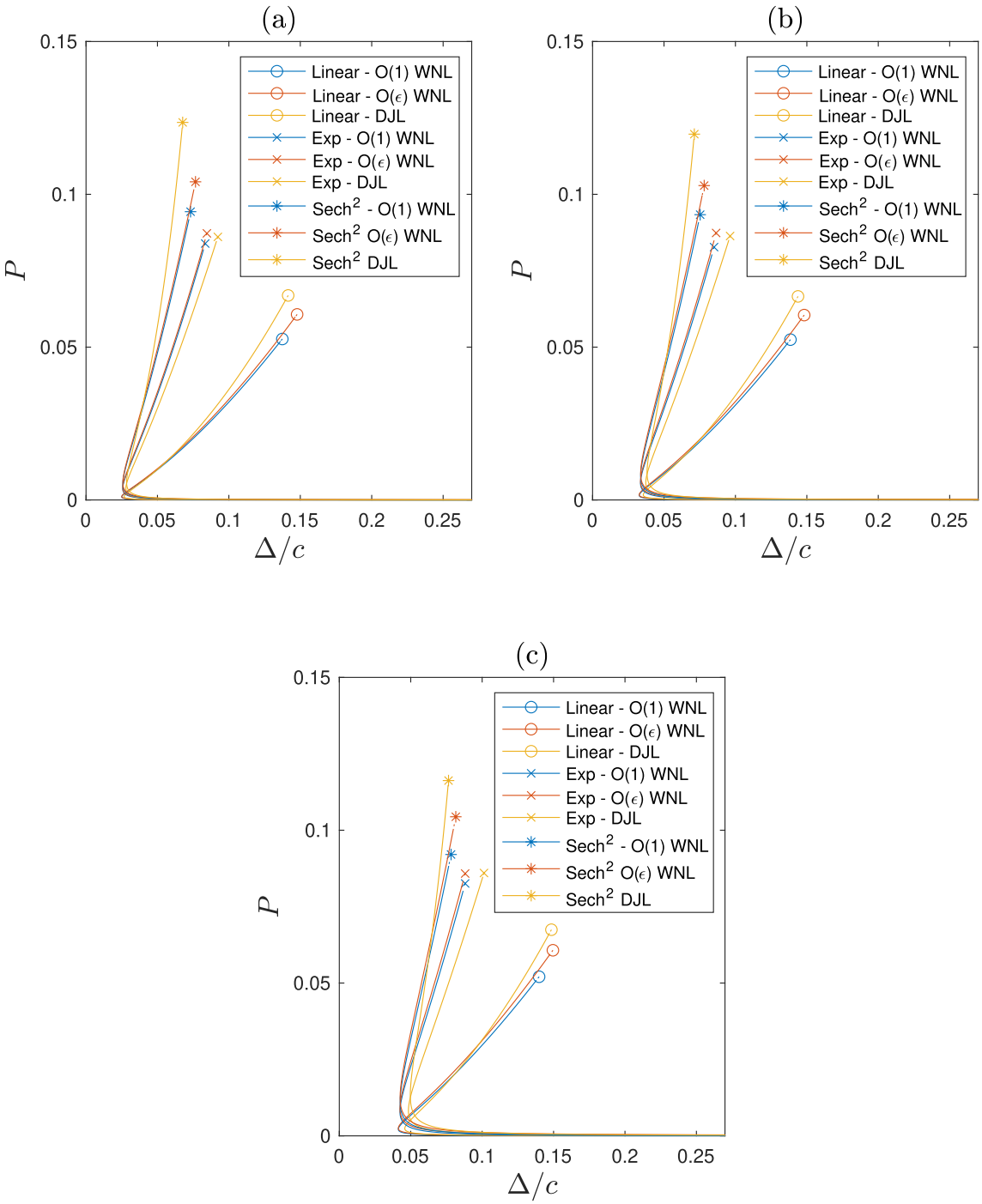

Figure 5.8: Region 2 DJL and WNL continuation for linear, exponential and sech^2 stratifications for $\mu = 0.7$ (a) $\mu = 0.8$ (b) and $\mu = 0.9$ (c). Overturning locations are indicated by the symbols $\circ \times *$. Dashed lines indicate the gap between the last valid DJL solution and the first overturning solution obtained during continuation.

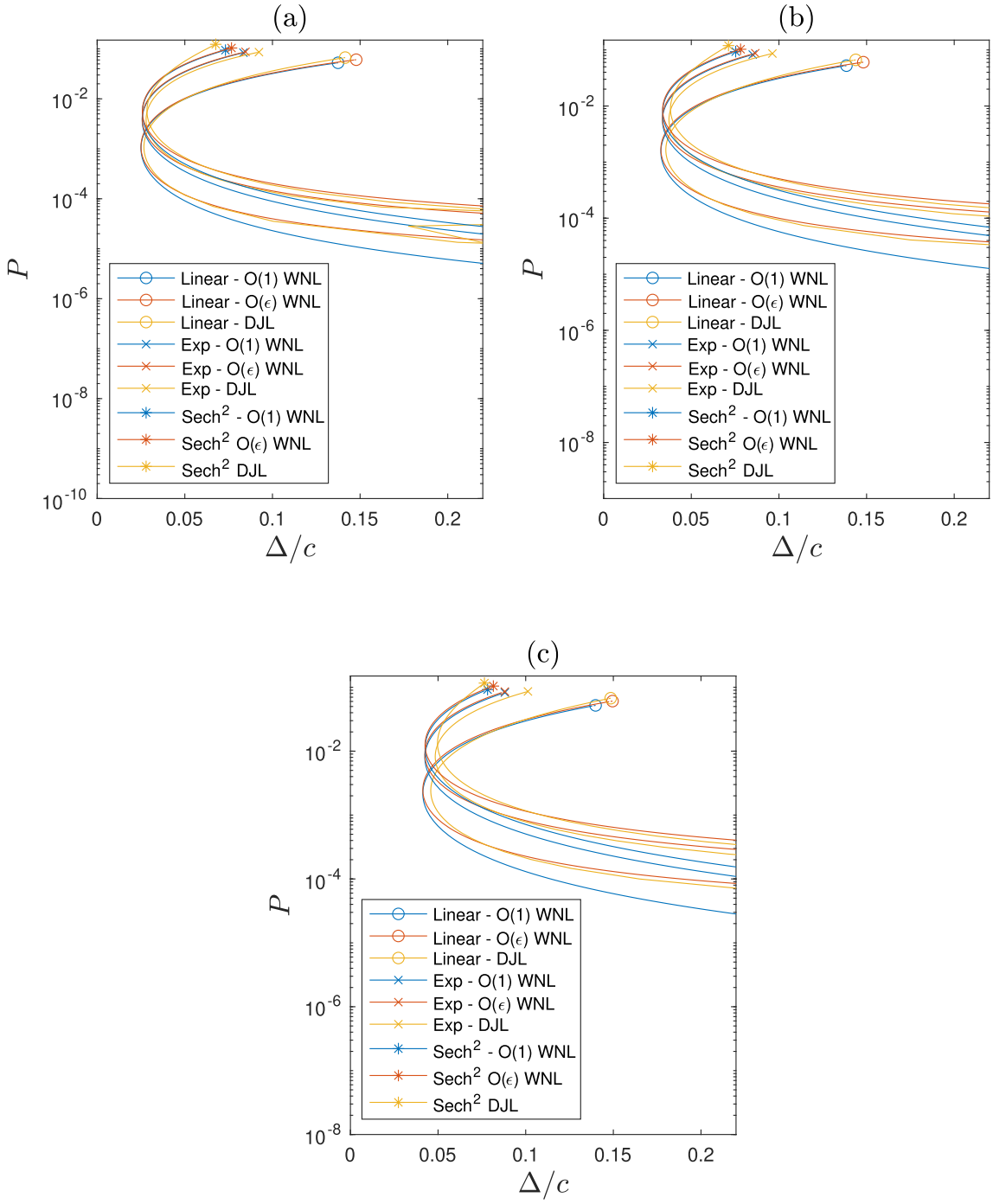

Figure 5.9: Region 2 DJL and WNL continuation for linear, exponential and sech^2 stratifications for $\mu = 0.7$ (a) $\mu = 0.8$ (b) and $\mu = 0.9$ (c). Same as Figure 5.8 but in log scale to better capture the turning point location. Overturning locations are indicated by the symbols $\circ \times *$. Dashed lines indicate the gap between the last valid DJL solution and the first overturning solution obtained during continuation.

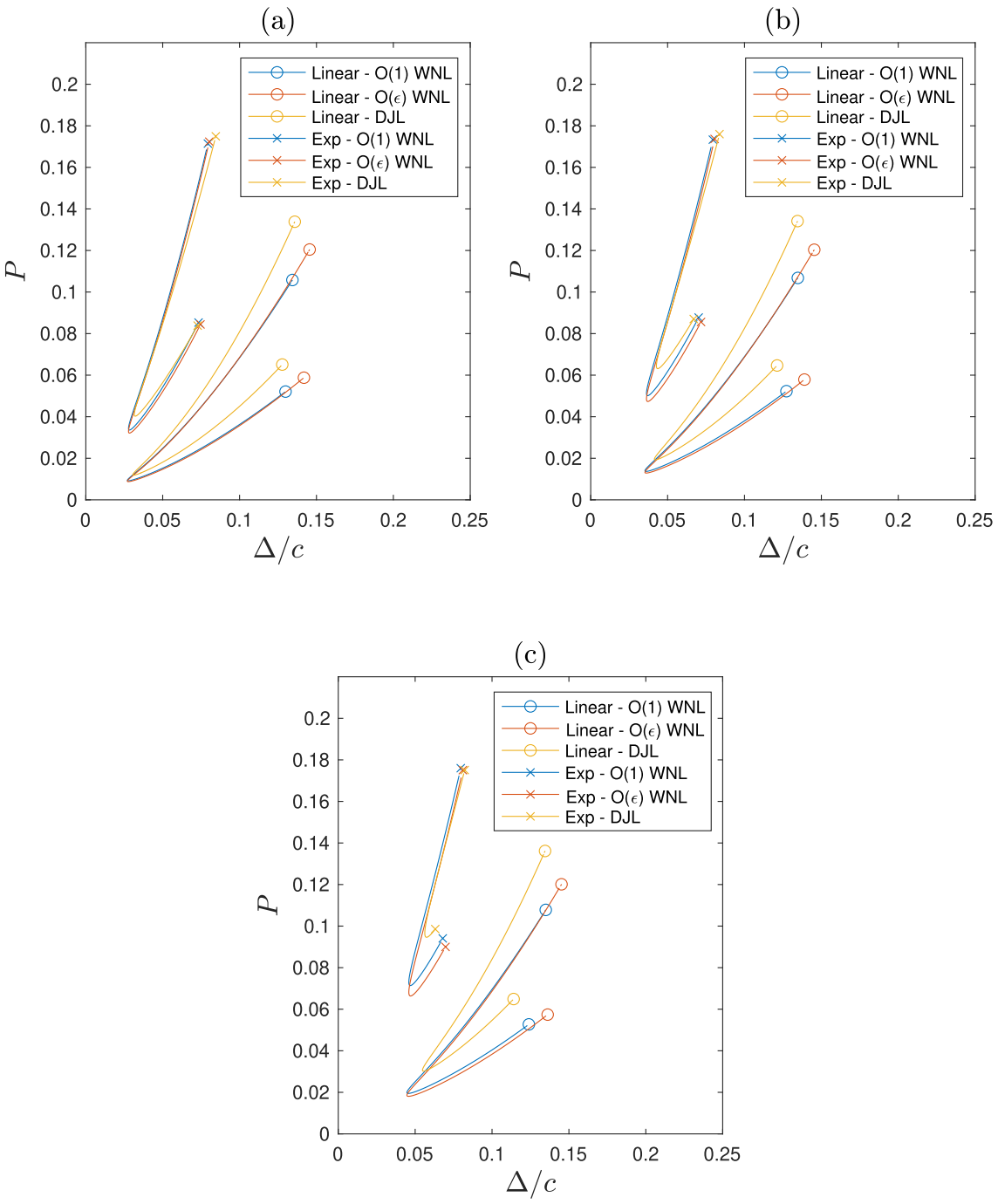

Figure 5.10: Region 3 DJL and WNL continuation for linear and exponential stratifications for $\mu = 0.7$ (a) $\mu = 0.8$ (b) and $\mu = 0.9$ (c). Overturning locations are indicated by the symbols $\circ \times *$. Dashed lines indicate the gap between the last valid DJL solution and the first overturning solution obtained during continuation.

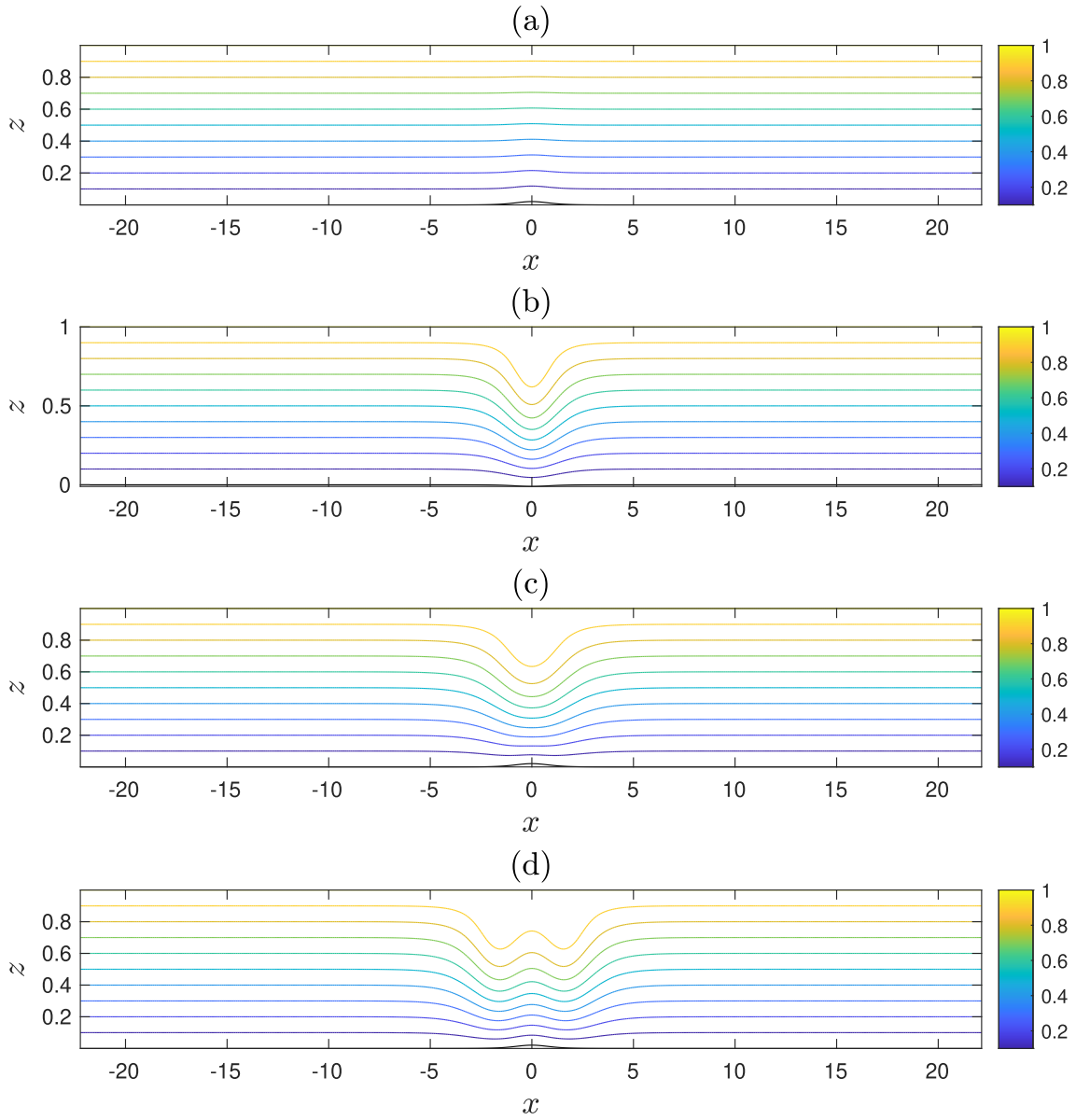

Figure 5.11: Isopycnals for exp stratification with $\mu = 0.9$ at region 1 at the location given by $*$ in Figure 5.7 (a), region 2 at the overturning location in Figure 5.8 (b), and region 3 at the overturning locations in Figure 5.10 (c) and (d). Bottom topography is shown in black.

parameters. Conformal mapping is performed with topography given by (5.71). We focus only on positive two bump topography because the Cauchy–Riemann plots in Figure 2.9 and 2.8 suggests large negative two bump topographies have unreliable conformal maps. Similarly, we consider only the positive branch of the hydraulic fall parameter space in Figure 4.6 when used in (5.69). The stratification $M^2(\eta)$ flipped along the mid axis $\eta = 0.5$ shown in Table 5.4 is used instead to ensure $\alpha > 0$.

Flipped Stratification	$M^2(\eta)$	a
Linear	$a(1-\eta)$	2
Exponential	$ae^{1-\eta}$	$\frac{1}{e-1}$
Sech^2	$a\,\text{sech}^2(\eta)$	$\frac{1}{\tanh(1)}$

Table 5.4: Table of flipped stratifications to Table 5.1 and normalising constant used for two bump topography experiments.

Unlike the single bump case, the hydraulic fall parameter space in Figure 4.6 for the positive branch is a one-to-one curve. Therefore, the chosen parameter Δ^* automatically determines γ^*. Since Δ is required to be small in the perturbation, this translates to a small Δ^* through equation (5.51). A small positive $\Delta^* \ll 1$ is picked initially for the WNL, which is then used as the initial guess in Newton's method. The corresponding γ^*, obtained from Figure 4.6 for the chosen Δ^*, gives the initial starting topography height α from (5.52).

DJL tabletop method

We are interested in finding specific DJL solutions that have a similar flat top to fKdV tabletop ones. This will be referred to as DJL tabletop solutions. To filter fKdV tabletop solutions, we utilised the wave drag constraint equation (4.26). However, no such condition exist for the DJL equation. Therefore, to filter DJL tabletop solutions we resort to an iterative algorithm.

Consider a DJL solution at a fixed topography height α. If $\Delta \approx \Delta_c$, where Δ_c is the critical value for DJL tabletop solutions, the DJL solution is similar to a tabletop one, except it exhibits small amplitude waves in the x direction where the flat top would be as shown in Figure 5.12.

In Figure 5.12, we observe that the wave crests and troughs of the small amplitude waves are at fixed locations in x despite changes in Δ. This suggests that the wavelength is constant for a given topography height α, while the amplitude of the waves changes with Δ. At $\Delta = \Delta_c$ the amplitude of the waves is zero.

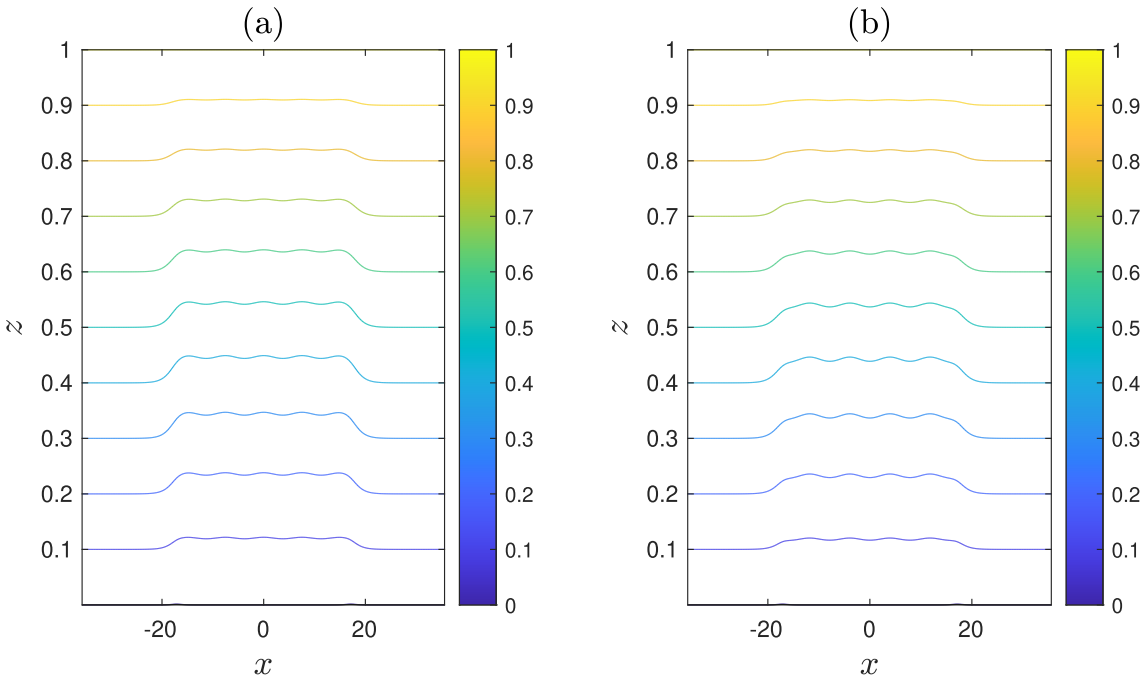

Figure 5.12: Isopycnals for linear stratification at $\gamma^* = 0.25$, $\mu = 0.7$ and $\alpha = 8.49 \times 10^{-4}$ where $\Delta = 1.0012\Delta_c$ (a) and $\Delta = 0.9988\Delta_c$ (b). Bottom topography is shown in black but not visible due to the small topography height.

Observing this behaviour, we propose an iterative algorithm to find DJL tabletop solutions ζ_c at (Δ_c, α). We solve for this and Δ_c at a fixed topography height α, by utilising a secant method to find the zeros of the amplitude of the waves. The secant method requires two DJL solutions $\zeta^{(1)}$ at $(\Delta^{(1)}, \alpha)$ and $\zeta^{(2)}$ at $(\Delta^{(2)}, \alpha)$ near ζ_c. In this section, $\zeta^{(1)}$ and $\zeta^{(2)}$ represent the iteration number used in the secant method and not the WNL.

Since the wavelength is fixed, we pick a point in $\zeta^{(1)}$, where a wave crest is located (x_c, z_c), and another point where a wave trough is (x_t, z_t) and calculate the difference $y^{(1)}$

$$y^{(1)} = \zeta^{(1)}(x_c, z_c) - \zeta^{(1)}(x_t, z_t) \quad \text{at} \quad (\Delta^{(1)}, \alpha). \tag{5.72}$$

At the same crest and trough locations we calculate the difference for $y^{(2)}$

$$y^{(2)} = \zeta^{(2)}(x_c, z_c) - \zeta^{(2)}(x_t, z_t) \quad \text{at} \quad (\Delta^{(2)}, \alpha). \tag{5.73}$$

The secant method is then used to find zero on the difference y, which corresponds to zero amplitude waves. Given two DJL solutions near ζ_c, namely, $\zeta^{(1)}$ at $(\Delta^{(1)}, \alpha)$ and $\zeta^{(2)}$ at $(\Delta^{(2)}, \alpha)$, the secant method to calculate a DJL tabletop solution is given in Algorithm 13.

The tolerance in step 4 of the Algorithm 13 is chosen to be 10^{-5}. We find that in step 7, using the initial guess $\zeta^{(k+1)}$ in Newton's method performs better than the interpolated

Algorithm 13 Algorithm for the secant method to calculate a DJL tabletop solution

1: Set $\zeta^{(1)}$ at $(\Delta^{(1)}, \alpha)$ and $\zeta^{(2)}$ at $(\Delta^{(2)}, \alpha)$
2: Set $k = 1$
3: Find location of the wave crest (x_c, z_c) and wave trough (x_t, z_t) in $\zeta^{(1)}$ and $\zeta^{(2)}$
4: **while** $|y^{(k+1)} - y^{(k)}| < tol$ **do**
5: Calculate $y^{(k+1)}$ and $y^{(k)}$ using equations (5.72) and (5.73)
6: Use secant method to calculate $\Delta^{(k+2)} = \Delta^{(k+1)} - y^{(k+1)} \frac{\Delta^{(k+1)} - \Delta^{(k)}}{y^{(k+1)} - y^{(k)}}$
7: Use Newton's method on the DJL equation (5.56) with $\zeta^{(k+1)}$ as the initial guess at $u = c + \Delta^{(k+2)}$ to calculate $\zeta^{(k+2)}$
8: $k \leftarrow k + 1$
9: **end while**

solution

$$\zeta^{(k+2)} = \zeta^{(k+1)} - y^{(k+1)} \frac{\zeta^{(k+1)} - \zeta^{(k)}}{y^{(k+1)} - y^{(k)}}. \tag{5.74}$$

This is most likely because (5.74) is not generally a solution to the DJL equation.

Algorithm 13 is used in 14 to find a DJL tabletop solution from the WNL approximation. In step 1, Δ^* and γ^* is picked from the positive branch of Figure 4.6. It is beneficial that both parameters are small and near the origin. Step 2 uses (5.45), with the known fKdV tabletop solution obtained in Chapter 4 at the chosen Δ^* and γ^*. This results in $\overline{\zeta}$ being a DJL tabletop solution. However, it does not remain so after Newton's method in step 3.

Algorithm 14 Algorithm for the computation of a DJL tabletop solution from the WNL approximation

1: Set Δ^* and γ^*
2: Calculate $\Delta^{(1)}$ and α using (5.51) and (5.52)
3: Use the $O(\epsilon)$ WNL to generate the initial guess $\overline{\zeta}$ at $(\Delta^{(1)}, \alpha)$
4: Use Newton's method with $\overline{\zeta}$ as the initial guess to obtain a solution to the DJL equation $\zeta^{(1)}$ at $(\Delta^{(1)}, \alpha)$.
5: Choose $\Delta^{(2)} = \Delta^{(1)} + \epsilon$, where $\epsilon \ll 1$.
6: Use Newton's method, with $\zeta^{(1)}$ as the initial guess, to obtain another DJL solution $\zeta^{(2)}$ at $(\Delta^{(2)}, \alpha)$.
7: Use the Algorithm 13 with $\zeta^{(1)}$ at $(\Delta^{(1)}, \alpha)$ and $\zeta^{(2)}$ at $(\Delta^{(2)}, \alpha)$ to obtain ζ_c at (Δ_c, α)

DJL tabletop continuation

Once a DJL tabletop solution is found for a given topography height α, we seek the continuation of it with parameter α. This is performed using natural parameter continuation. The reason is twofold: First is that the term $\frac{\partial \zeta}{\partial \alpha}$ required in the pseudo arc-length continuation is difficult to compute. Second is that it is likely the DJL tabletop continuation closely resembles the hydraulic fall parameter space for the fKdV equation in Figure 4.6

being a one-to-one curve with no folds, thus, natural parameter continuation is sufficient. Note that a new conformal mapping is required at every continuation step due to the change in topography height.

At each step in the natural parameter continuation, two DJL solutions at different values of Δ are required for the secant method. At continuation step ν, let $\zeta_\nu^{(1)}(\Delta_\nu^{(1)}, \alpha_\nu)$ and $\zeta_\nu^{(2)}(\Delta_\nu^{(2)}, \alpha_\nu)$ denote the two to be used in the secant method, and $\zeta_\nu(\Delta_\nu, \alpha_\nu)$ to be the DJL tabletop solution obtained from the secant method. We begin by initialising the natural parameter continuation with the first DJL tabletop solution $\zeta_0(\Delta_0, \alpha_0)$. In the first continuation step we take $\zeta_1^{(1)}$ to be the Newton correction to the constant or zeroth-order approximation in Δ for the predictor step. Let $\mathcal{N}\{\zeta, (\Delta_\nu, \alpha_\nu)\}$ denote the Newton correction of the predictor ζ at (Δ_ν, α_ν). The zeroth-order approximation at step ν is given by

$$\begin{aligned} \zeta_\nu^{(1)}(\Delta_\nu^{(1)}, \alpha_\nu) &= \mathcal{N}\{\zeta_{\nu-1}(\Delta_{\nu-1}, \alpha_{\nu-1}), (\Delta_\nu^{(1)}, \alpha_\nu)\}, \\ \Delta_\nu^{(1)} &= \Delta_{\nu-1}. \end{aligned} \tag{5.75}$$

Since in the first step there is no tangent approximation, we look for the second DJL solution at a small perturbation in $\Delta^{(1)}$, using the first as the initial guess. We shall call this the *zeroth-order perturbation approximation*. At step ν this is given by

$$\begin{aligned} \zeta_\nu^{(2)}(\Delta_\nu^{(2)}, \alpha_\nu) &= \mathcal{N}\{\zeta_\nu^{(1)}(\Delta_\nu^{(1)}, \alpha_\nu), (\Delta_\nu^{(2)}, \alpha_\nu)\}, \\ \Delta_\nu^{(2)} &= \Delta_\nu^{(1)} + \epsilon, \end{aligned} \tag{5.76}$$

where $0 < \epsilon \ll 1$.

In the second continuation step, we take the zeroth and first-order approximations in ζ and Δ as the two DJL solutions used in the secant method. The tangent or first-order approximation at step ν is given by

$$\begin{aligned} \zeta_\nu^{(2)}(\Delta_\nu^{(2)}, \alpha_\nu) &= \mathcal{N}\left\{ \frac{\zeta_{\nu-1}(\Delta_{\nu-1}, \alpha_{\nu-1}) - \zeta_{\nu-2}(\Delta_{\nu-2}, \alpha_{\nu-2})}{\delta\alpha_{\nu-1}} \delta\alpha_\nu, (\Delta_\nu^{(1)}, \alpha_\nu) \right\}, \\ \Delta_\nu^{(2)} &= \frac{\Delta_{\nu-1} - \Delta_{\nu-2}}{\delta\alpha_{\nu-1}} \delta\alpha_\nu, \end{aligned} \tag{5.77}$$

where $\delta\alpha_\nu = \alpha_\nu - \alpha_{\nu-1}$. Note that the first-order approximation requires known DJL tabletop solutions at two previous steps $\zeta_{\nu-1}(\Delta_{\nu-1}, \alpha_{\nu-1})$ and $\zeta_{\nu-2}(\Delta_{\nu-2}, \alpha_{\nu-2})$, hence can only be used for steps $\nu \geq 1$.

For continuation steps $\nu \geq 2$, we take the first and second-order approximation as the two DJL solutions in the secant method. The quadratic or second-order approximation

at step ν is given by

$$\zeta_\nu^{(2)}(\Delta_\nu^{(2)}, \alpha_\nu) = \mathcal{N}\Bigg\{ \frac{\zeta_{\nu-3}(\delta\alpha_{\nu-2})^2 - \zeta_{\nu-2}(\delta\alpha_{\nu-2} + \delta\alpha_{\nu-1})^2 + \zeta_{\nu-1}\delta\alpha_{\nu-1}(2\delta\alpha_{\nu-2} + \delta\alpha_{\nu-1})}{\delta\alpha_{\nu-2}\delta\alpha_{\nu-1}(\delta\alpha_{\nu-2} + \delta\alpha_{\nu-1})}\delta\alpha_\nu,$$
$$(\Delta_\nu^{(1)}, \alpha_\nu)\Bigg\},$$
$$\Delta_\nu^{(2)} = \frac{\Delta_{\nu-3}(\delta\alpha_{\nu-2})^2 - \Delta_{\nu-2}(\delta\alpha_{\nu-2} + \delta\alpha_{\nu-1})^2 + \Delta_{\nu-1}\delta\alpha_{\nu-1}(2\delta\alpha_{\nu-2} + \delta\alpha_{\nu-1})}{\delta\alpha_{\nu-2}\delta\alpha_{\nu-1}(\delta\alpha_{\nu-2} + \delta\alpha_{\nu-1})}\delta\alpha_\nu. \tag{5.78}$$

Note that the second-order approximation requires known DJL tabletop solutions at three previous steps $\zeta_{\nu-1}(\Delta_{\nu-1}, \alpha_{\nu-1})$, $\zeta_{\nu-2}(\Delta_{\nu-2}, \alpha_{\nu-2})$ and $\zeta_{\nu-3}(\Delta_{\nu-3}, \alpha_{\nu-3})$, hence can only be used for steps $\nu \geq 2$.

In the case, where at a continuation step ν, the first and second-order or the zeroth and first-order approximation are too close, that is

$$|\Delta_\nu^{(2)} - \Delta_\nu^{(1)}| < tol, \tag{5.79}$$

we revert one of the solution to the zeroth-order perturbation approximation (5.76). The tolerance in (5.79) is set to 10^{-5}. Similar to the single bump topography case, we focus only on mode 1 DJL solutions since Newton's method failed to converge for mode 2 ones.

Plots of γ^* vs Δ^* for the DJL tabletop parameter space with stratifications given in Table 5.4 and $\mu = 0.7, 0.8$ and 0.9 are shown in Figure 5.13. The sech^2 stratification failed to converge in the initial Newton iteration and is not shown. The comparison to the fKdV hydraulic fall parametric space is shown, with the conversion of the parameters Δ to Δ^* and α to γ^* given by (5.51) and (5.52) respectively. The isopycnals for the exponential stratification is shown in Figure 5.14, showing a tabletop like shape.

We see that the fKdV hydraulic fall parameter space is a good approximation to DJL tabletop one only for $\Delta^* \ll 1$ and $\gamma^* \ll 1$, or, equivalently, $\Delta \ll 1$ and $\alpha \ll 1$. They then diverge as Δ and γ increases, with increasing divergence for higher nonlinear stratifications and for larger μ. This is consistent with the perturbation expansion assumptions.

Overturning predictions for the WNL are not shown since the difference between the DJL tabletop and fKdV hydraulic fall parametric space is significant at the DJL tabletop overturning locations. For linear and exponential stratification these also occur at decreasing values of γ^* as the stratification becomes strongly nonlinear, and as μ increases.

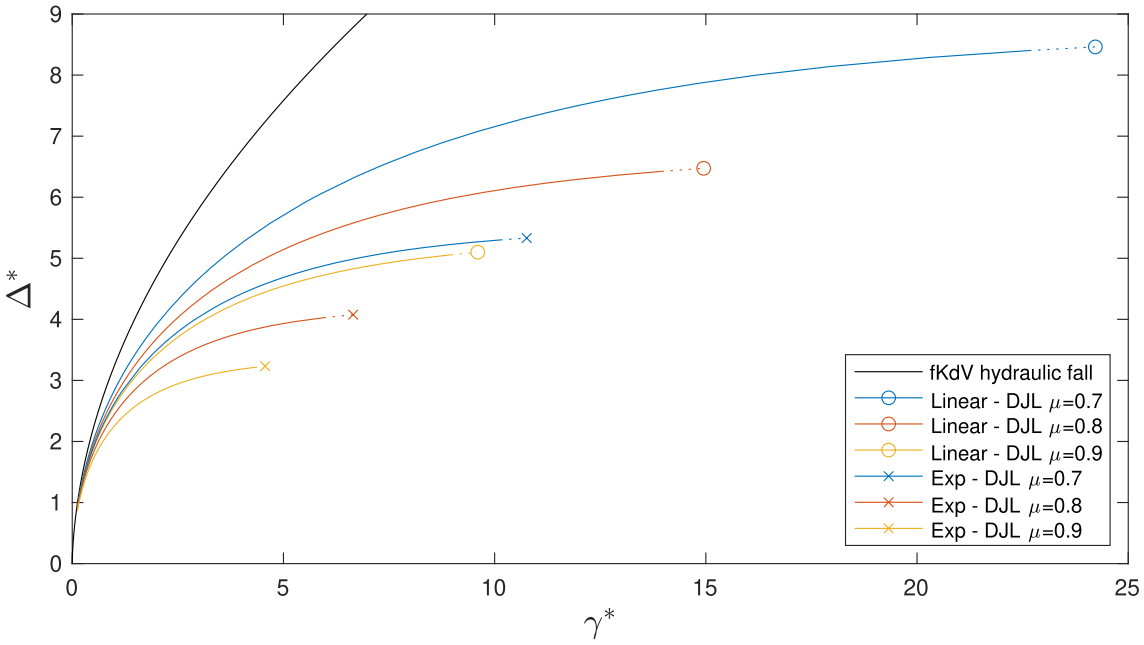

Figure 5.13: Plot of γ^* vs Δ^* for the fKdV hydraulic and DJL tabletop parameter space for linear and exponential stratifications for various μ values. Sech^2 stratification is not shown since it failed to converge in the initial Newton step. Overturning locations for linear and exponential stratifications are indicated by the symbols $\circ$ and $\times$ respectively. Dashed lines indicate the gap between the last valid DJL solution and the first overturning solution obtained during continuation.

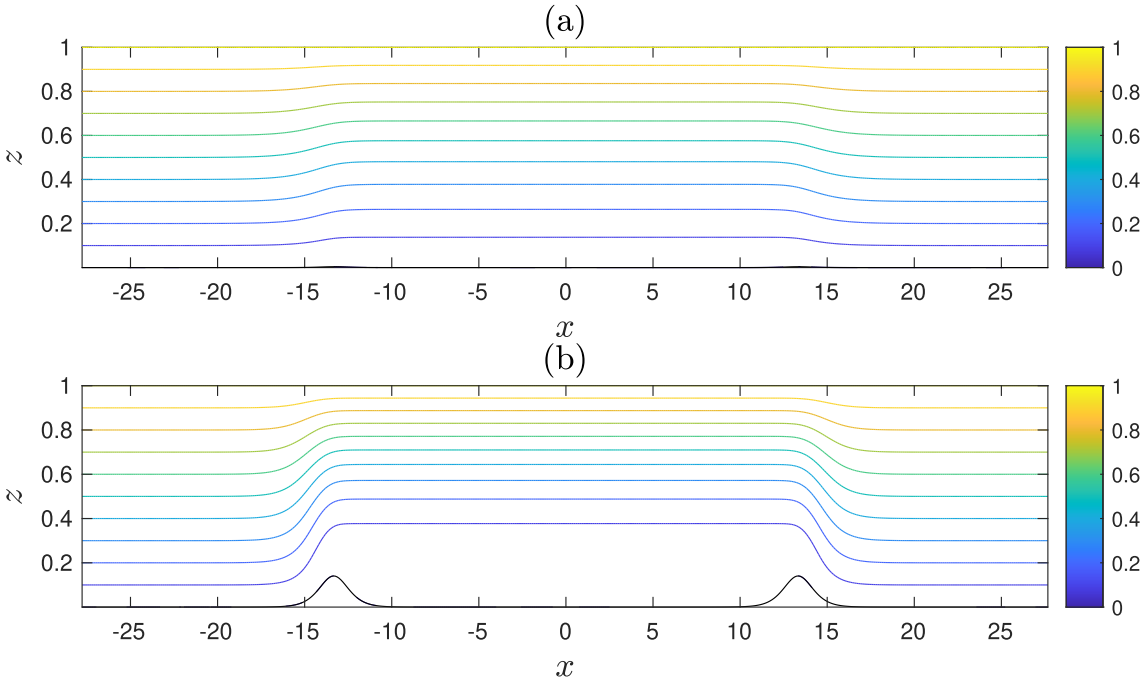

Figure 5.14: Isopycnals for exponential stratification at $\mu = 0.9$ and $\gamma^* = 0.1$ (a) and at overturning location $\times$ from Figure 5.13 (b). Bottom topography is shown in black.

5.5 Summary

In this chapter we numerically solved the DJL equation under the Boussinesq approximation and with no background current. Linear, exponential and sech^2 stratifications were considered. Perturbation methods were used to generate the initial guess for Newton's method. We were interested in comparing WNL and DJL solutions with no-topography, single bump and double bump bottom topography.

In the no-topography case, our numerical method is robust and able to obtain both mode 1 and mode 2 DJL solutions with the exception of mode 2 exponential stratification. Numerical continuation is used with parameter Δ to find the solution family for both WNL and DJL. The $O(1)$ and $O(\epsilon)$ WNL are almost identical, except the predicted overturning location, where $O(\epsilon)$ predict it at a larger Δ. We find for all topographies, they is a good approximation to the DJL solution for small Δ/c values, as expected from perturbation theory. However, it becomes poorer as Δ/c increases.

Next we solved the DJL equation with a single bump topography forcing. The numerical conformal mapping algorithm was used to map the forced DJL domain to a rectangular one before spectral methods were used. The WNL requires the fKdV continuation with a single bump bottom topography, which were obtained using the same numerical methods as in Chapter 4. We focused on three fKdV solutions types in regions 1,2 and 3. Numerical interpolation of the grid points in the WNL domain to the forced DJL one is also required so that the perturbation solution can be used as the initial guess in Newton's method. The WNL and DJL continuations in Δ in these regions were then compared. In region 1, we found the $O(1)$ WNL better approximates the DJL solutions for $\Delta/c \ll 1$, whereas the $O(\epsilon)$ counterpart for $\Delta/c \to \infty$ for all stratifications and μ values. This is likely due to the improved boundary condition approximation of the $O(\epsilon)$ perturbation. In region 2, it is a good approximation to the DJL solution but then begins to deviate as overturning is approached for all stratifications and μ values. However, the it performs poorly in region 3, with large deviations from the DJL solutions for all stratifications, and worse as μ increases. Our numerical method is less robust in the single bump topography case, being unable to generate mode 2 solutions for any stratification likely due to the asymptotics beyond all order phenomena.

Finally, we investigated double bump topography DJL tabletop solutions with a similar flat top to fKdV ones, which were obtained in Chapter 4 to generate the WNL. A secant algorithm is used to filter DJL tabletop solutions, which is also used in the natural parameter continuation to find its parameter space. Results show the fKdV hydraulic fall parameter space is a good approximation to the DJL tabletop counterpart only for $\Delta^* \ll 1$ and $\gamma^* \ll 1$, as the two curves diverge rapidly as γ^* and μ increases and for

stronger nonlinear stratifications.

Chapter 6

Conclusion

We solved the DJL equation with constant background current under the Boussinesq approximation with a single and double bump bottom topography. We further demonstrated the viability of SMG for solitary wave computations including the fKdV and DJL equation. The project was motivated by the success of Yang's Newton-Krylov methods for solving various solitary wave equations [115].

We began by introducing the numerical methods used for the fKdV and DJL equations. Fourier and Chebyshev pseudospectral methods were explored with focus on efficiency, that is, ones that have $O(N \log N)$ operations for spectral transformation, differentiation and integration. Efficient Ad Hoc spectral solvers were discussed next, which were limited to Laplace, Poisson and Helmholtz type problems. Efficient spectral iterative techniques were then explored for the more general case, which include Richardson and Krylov methods with the option of their acceleration with second-order FD preconditioners.

Another approach to accelerating iterative methods is SMG, which is one of the main focus in this thesis. SMG differs from traditional multigrid with its spectral discretisation, restricted relaxation choices, and grid transfer operators. It is crucial that every component of SMG is efficient and can utilise the FFT. Therefore, Richardson class of methods is used for relaxation and trigonometric interpolation for grid transfer operations. Within SMG, we developed an improved Dirichlet restriction and prolongation operator for the Chebyshev pseudospectral method. This involved initially only considering the internal points and solving for the boundary ones assuming a homogeneous Neumann condition is present, before grid operators are applied. This removes the presence of Gibbs phenomenon and thus improves SMG convergence.

Newton's method is implemented for nonlinear problems, where within each iteration, SMG is used as the linear Jacobian solver. Predictor-corrector natural parameter and pseudo arc-length continuation methods are then used to solve a system of parameterised nonlinear equations.

Finally, we developed an iterative numerical conformal mapping algorithm to map the forced DJL domain to a rectangular one, so that spectral methods can be used. Our algorithm more efficient than the boundary mapping method by Soontiens et al. [104], since exact Ad Hoc spectral solvers are significantly faster than preconditioned GMRES.

SMG was then tested for performance comparisons between Newton-SMG and Newton-Krylov methods for a variety of solitary wave computations. Yang [115] showed Newton-CG outperformed all existing solitary wave computation methods including nonlinear CG. Our results showed Newton-SMG either performs similarly or outperforms Newton-Krylov methods. This demonstrates the usefulness of SMG, especially for higher resolution grids.

In Chapter 4 we investigated solitary and hydraulic fall solutions to the steady fKdV equation under a single and double bump forcing. This also served as a one-dimensional test case for SMG. A shooting method was utilised to generate several EDSWs, which were consistent with the results by Cullen [27]. We then showed using numerical continuation that each type of EDSW evolved into a different type of ADSW solution obtained by Keeler et al. [67]. Numerical evidence suggests that there may exist an infinite set of EDSW solutions, further implying the existence of infinite types of ADSW solutions. Asymmetric hydraulic fall parameter space were then investigated through tabletop solutions and the wave-drag constraint equation. The methodology chosen is more numerically stable than the approach used by Ee and Clarke [34] and Cullen [27], and was able to generate the extended negative γ region of the hydraulic fall parameter space. We showed the hydraulic fall solutions periodically evolved into the EDSW solutions obtained with single bump topography. Krylov methods were more efficient than SMG when applied to the steady fKdV equation. This is to be expected since SMG is not competitive for problems in one dimension as shown by Canuto et al. [19], but is typically used for problems in higher dimensions.

Finally, in Chapter 5 we solved the DJL equation under the Boussinesq approximation and with no background current. This served as a two-dimensional test case for SMG with Fourier-Chebyshev discretisation. We began with the perturbation series, which formed the WNL and the initial guess for Newton's method. Once a DJL solution was obtained, numerical continuation was used with parameter Δ until overturning was reached.

The DJL equation with no topography was first considered, where the WNL utilised the KdV solution. For the no topography case, several mode 2 DJL solutions were obtained. We found that the WNL is a good approximation to the DJL only for $\Delta/c \ll 1$. Single bump topography was then considered, where the WNL utilised the single bump fKdV solution. The iterative numerical conformal mapping algorithm introduced in Chap-

ter 2 was used to map the forced DJL domain to a rectangular one so that spectral methods could be used. We found that the WNL is a good approximation for $\Delta/c \ll 1$ only in region 1 and 2, but is poor in region 3. Finally, two bump topography was investigated, where the WNL utilised fKdV tabletop solutions obtained in Chapter 4. A secant method was developed to filter for DJL tabletop solutions. Its parameter space was then compared to the hydraulic fall counterpart, where the latter was only a good approximation to the former for $0 < \gamma^* \ll 1$, before diverging rapidly.

There are several areas of improvement in SMG, which have not been explored in this thesis. Our results utilised the MRR with direction inversion of the second-order FD preconditioner using sparse linear solvers. In our experience, this proves to be the most robust for both Jacobian calculations in the fKdV and the DJL equations. However, in two-dimensions, the computation is expensive. To combat this, many alternative relaxations to the FD preconditioner have been proposed in literature. Line relaxation was first utilised in Chebyshev SMG by Kang et al. [66] and extended to Fourier SMG by Brandt et al. [13]. Further investigation into this for SMG was performed by Heinrichs [54] and found improved convergence as suggested by smoothing analysis. Phillips [91] investigated incomplete LU decompositions (ILU) of the five-point second-order and nine-point fourth-order FD matrix. Comparisons were made on a self-adjoint elliptic equation between line relaxation, approximate factorisation (AF) and ILU for both Fourier and Chebyshev SMG, with all relaxation methods showing effective smoothing.

Further improvements can also be made with the choice of preconditioner. Zang [122] initially considered variants of ILU decomposition for Chebyshev SMG that are easy to invert. Phillips et al. [92] further investigated alternative efficient preconditioners for the Chebyshev SMG, which involve various variants of ILU decomposition of the FD and approximate spectral matrix. Canuto et al. [19] then compared inexact factorisation of low-order preconditioners, which show ILU to be more efficient than FD as preconditioners.

Another extension to SMG would be SMG preconditioned Krylov subspace methods. While they are weak relaxations alone, multigrid preconditioned Krylov methods are quite commonly used throughout literature with great success. Heinrich [59] compared different types of multigrid strategies and showed that FE multigrid preconditioned Richardson relaxation being the most efficient strategy. It is likely a strategy that combines elements of Krylov methods and SMG could be more efficient than Krylov or SMG methods alone.

An unexplored area in this thesis is SMG utilising nonlinear relaxations and the full approximation scheme (FAS). Nonlinear relaxations have seen very little research for use in SMG. This is an alternative to using Newton's method for solving nonlinear problems.

One popular option used in FD multigrid is the nonlinear Jacobi. Instead of an outer Newton iteration, it incorporates an inner one within each Jacobi iteration. However, initial testing with FAS SMG on simple model problems proved unsuccessful and was not pursued further. Other examples of nonlinear relaxation used in SMG is by Street et al. [107], where AF was successfully applied to solve transonic potential flow. Yang [115] showed Newton-CG outperformed nonlinear CG method. It would be interesting to compare the SMG counterpart, that is, nonlinear SMG with FAS to Newton-SMG.

In our iterative numerical conformal mapping algorithm, we observed performance degradation for double bump and negative topographies, and for increasing topography height. It is suspected that the spectral discretisation may be a factor. This is because the Cauchy–Riemann 2-norms are large near the boundaries, sometimes exhibiting Gibbs phenomenon. It is unclear if alternative spectral, FD or FE discretisation, would show the same behaviour. Furthermore, performance comparisons have not been made with the improved embedded boundary method of Laprise and Peltier [75]. This involves iterating the solution to match the known bottom topography and boundary condition. Soontiens et al. [103] showed a remarkable increase in computational efficiency of the embedding method compared to the boundary mapping counterpart. A comparison of our iterative numerical conformal mapping algorithm with the embedded boundary method would be of considerable interest.

For the fKdV equation, while analytic asymptotics exist for large positive topography height and local compact topography limits, there is little known analytic or asymptotic results in the negative γ^* region for the hydraulic fall parameter space. Similarly, little is known about the set of EDSW solutions, which appear to be given by a nonlinear eigenvalue problem.

In Chapter 5, our work considered stratifications in the form of linear, exponential and sech^2. One area of interest may be to consider the transition between these. For linear and exponential, this can be captured by introducing a parameter γ such that $M^2(\eta) = e^{\gamma\eta}$. For $\gamma \ll 1$, this is equivalent to linear stratification after normalising. Between exponential and sech^2, a stratification of the form $M^2(\eta) = \gamma e^{\eta} + (1-\gamma)\operatorname{sech}^2\eta$ can be used while adjusting the parameter $\gamma \in [0, 1]$. Another popular stratification used in literature and not considered in the thesis due to the failures of Newton's method, is the thermocline stratification given by $M^2(\eta) = \operatorname{sech}^2\left(\frac{\eta-\eta_0}{d}\right)$ for constants η_0 and d [106].

The continuation of our solutions ceased at overturning. An extension of our work past this would require resolving the solution within the recirculating region, where the DJL equation is not valid. This has been demonstrated by Helfrich and White [63] with laboratory experiments confirming the analytic results [79].

In the single bump topography case, we only considered a continuation with parameter Δ. This resulted in a very small value of topography height α on the order of 10^{-2} to 10^{-3} relative to the domain size. In this case, conformal mapping is hardly required since the DJL and the WNL domain are very similar, i.e. $(x, z) \simeq (x', z')$. Additionally, the ratio (5.48) is of the order of $10^1 - 10^2$ and fixed during continuation since Δ and α remain unchanged. An extension of our work would be to consider continuation in both wave speed Δ and topography height α for both the fKdV and the DJL equation for comparison. It is anticipated the WNL would be more accurate as the ratio μ/α^4 approaches 1.

To remove large boundary errors when negative topographies are used, an alternative proposed method is to extend the z domain to $z \in [\alpha, 1]$, for $\alpha < 0$, which would affect the calculation of the eigenfunction ϕ. The benefit is that it avoids the large interpolation from the boundary errors akin to extrapolation and can handle large negative forcings. This, however, has higher complexity due to the boundary conditions on ϕ and hence also for ζ at α.

For double bump topography, we only considered the positive branch of the hydraulic fall parameter space shown in Figure 4.6. Therefore, the flipped stratification shown in Table 5.4 is used to ensure α remains positive. This is because of the conformal mapping results in Chapter 2, showing degraded performance for negative and double bump topographies. If a reliable conformal map can be obtained, and extrapolation minimised, an extension would be to consider the original stratifications in Table 5.1 in the positive branch of the fKdV continuation. A further extension would be to consider the negative branch of the hydraulic fall parameter space. It is uncertain if our methodology is capable or needs improvement to generate results in this region.

Our Newton-SMG method for the DJL equation is likely slower and less robust than the DJLES iterative method by Dunphy et al. [33]. This is primarily due to the FD preconditioner used with SMG. A similar spectral preconditioner by Yang [115] was attempted but failed to converge. If an efficient spectral preconditioner, which allows Ad Hoc methods to invert, similarly to Yang's preconditioner [115] can be developed for the DJL equation, Newton-SMG may be competitive to DJLES in efficiency. Furthermore, SMG may also be incorporated to speed up the Poisson solver in the DJLES algorithm. We also anticipate improved robustness of Newton-SMG if an alternative spectral discretisation with fixed boundaries at infinity is used over Fourier discretisation. This is because Newton's method frequently converged to a periodic, but not asymptotically decaying solution. Since the domain is infinite, alternative choices of spectral discretisation include rational Chebyshev, sinc and Hermite functions [11].

Finally, the natural extension of our works would be the investigation of stability and time evolution for our steady fKdV and DJL solutions.

Bibliography

[1] Akylas, T. and Grimshaw, R. (1992). Solitary internal waves with oscillatory tails. *Journal of Fluid Mechanics*, 242:279–298.

[2] Allgower, E. L. and Georg, K. (2003). *Introduction to Numerical Continuation Methods*. SIAM.

[3] Alpers, W. (2014). *Ocean Internal Waves*, pages 433–437. Springer New York, New York, NY.

[4] Apel, J. R. (2002). Oceanic Internal Waves and Solitons. *An Atlas of Oceanic Internal Solitary Waves*, 1:1–40.

[5] Baltensperger, R. and Berrut, J. P. (1999). The errors in calculating the pseudospectral differentiation matrices for Chebyshev-Gauss-Lobatto points. *Computers & Mathematics with Applications*, 37(1):41–48.

[6] Bender, C. M. and Orszag, S. A. (1999). *Advanced Mathematical Methods for Scientists and Engineers*. Springer.

[7] Benney, D. J. (1966). Long non-linear waves in fluid flows. *Journal of Mathematics and Physics*, 45(1-4):52–63.

[8] Binder, B. J. (2019). Steady two-dimensional free-surface flow past disturbances in an open channel: solutions of the Korteweg–de Vries equation and analysis of the weakly nonlinear phase space. *Fluids*, 4(1):24.

[9] Bowman, J. C. (2014). How Important is Dealiasing for Turbulence Simulations? *online slides: http://helper. ipam. ucla. edu/publications/mtws1/mtws1*, 12187.

[10] Bowman, J. C. and Roberts, M. (2011). Efficient dealiased convolutions without padding. *SIAM Journal on Scientific Computing*, 33(1):386–406.

[11] Boyd, J. P. (2001). *Chebyshev and Fourier Spectral Methods*. Courier Corporation.

[12] Brandt, A. (1977). Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390.

[13] Brandt, A., Fulton, S. R., and Taylor, G. D. (1985). Improved spectral multigrid methods for periodic elliptic problems. *Journal of Computational Physics*, 58(1):96–112.

[14] Briggs, W. L., Henson, V. E., and McCormick, S. F. (2000). *A Multigrid Tutorial*. SIAM.

[15] Broutman, D. and Grimshaw, R. (1990). Spectral multigrid and collocation methods for barotropic nondivergent flow over irregular coastal topography. *Geophysical & Astrophysical Fluid Dynamics*, 52(1-3):1–23.

[16] Brown, D. J. and Christie, D. R. (1998). Fully nonlinear solitary waves in continuously stratified incompressible Boussinesq fluids. *Physics of Fluids*, 10(10):2569–2586.

[17] Camassa, R. and Wu, T. Y. T. (1991). Stability of forced steady solitary waves. *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, 337(1648):429–466.

[18] Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. (1988). *Spectral Method in Fluid Dynamics*. Springer-Verlag, New York.

[19] Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. (2006). *Spectral Methods*. Springer.

[20] Canuto, C. and Quarteroni, A. (1985). Preconditioned minimal residual methods for Chebyshev spectral calculations. *Journal of Computational Physics*, 60(2):315–337.

[21] Carr, M., Stastna, M., and Davies, P. A. (2010). Internal solitary wave-induced flow over a corrugated bed. *Ocean Dynamics*, 60:1007–1025.

[22] Chardard, F., Dias, F., Nguyen, H. Y., and Vanden-Broeck, J. M. (2011). Stability of some stationary solutions to the forced KdV equation with one or two bumps. *Journal of Engineering Mathematics*, 70(1-3):175–189.

[23] Christie, D. R. and White, R. (1992). The morning glory of the Gulf of Carpentaria. *Australian Meteorological Magazine*, 41:21–60.

[24] Chugunova, M. and Pelinovsky, D. (2006). Two-pulse solutions in the fifth-order KdV equation: rigorous theory and numerical approximations. *arXiv preprint nlin/0605043*.

[25] Clenshaw, C. W. and Curtis, A. R. (1960). A method for numerical integration on an automatic computer. *Numerische Mathematik*, 2:197–205.

[26] Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

[27] Cullen, A. C. (2018). *A novel numerical solver for nonlinear boundary value problems, with applications to the forced Gardner equation.* PhD thesis, Monash University.

[28] Cullen, A. C. and Clarke, S. R. (2019). A fast, spectrally accurate homotopy based numerical method for solving nonlinear differential equations. *Journal of Computational Physics*, 385:106–118.

[29] Dias, F. and Vanden-Broeck, J. M. (1989). Open channel flows with submerged obstructions. *Journal of Fluid Mechanics*, 206:155–170.

[30] Dias, F. and Vanden-Broeck, J. M. (2002). Generalised critical free-surface flows. *Journal of Engineering Mathematics*, 42(3-4):291–301.

[31] Dijkstra, H. A. (2005). *Nonlinear Physical Oceanography: A Dynamical Systems Approach to the Large Scale Ocean Circulation and El Nino*, volume 532. Springer.

[32] Dubreil-Jacotin, M. L. (1932). Sur les ondes de type permanent dans les liquides hétérogenes. *Atti Accad. Naz. Lincei Rend*, 6:814–819.

[33] Dunphy, M., Subich, C., and Stastna, M. (2011). Spectral methods for internal waves: indistinguishable density profiles and double-humped solitary waves. *Nonlinear Processes in Geophysics*, 18(3):351–358.

[34] Ee, B. K. and Clarke, S. R. (2007). Weakly dispersive hydraulic flows in a contraction: Parametric solutions and linear stability. *Physics of Fluids*, 19(5):056601.

[35] Ehrenstein, U. and Peyret, R. (1989). A Chebyshev collocation method for the Navier–Stokes equations with application to double-diffusive convection. *International Journal for Numerical Methods in Fluids*, 9(4):427–452.

[36] Eisenstat, S. C., Elman, H. C., and Schultz, M. H. (1983). Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis*, 20(2):345–357.

[37] Engels, H. (1980). Numerical quadrature and cubature. *Academic Press*.

[38] Erlebacher, G., Zang, T. A., and Hussaini, M. Y. (1987). Spectral multigrid methods for the numerical simulation of turbulence. *Multigrid Methods, Marcel Dekker, New York*, pages 177–194.

[39] Erlebacher, G., Zang, T. A., and Hussaini, M. Y. (1988). Spectral multigrid methods for the solution of homogeneous turbulence problems. *Multigrid Methods: Theory, Applications, and Supercomputing, SF McCormick, ed*, 110:177–194.

[40] Fejér, L. (1933). Mechanische quadraturen mit positiven cotesschen zahlen. *Mathematische Zeitschrift*, 37(1):287–309.

[41] Forbes, L. K. (1988). Critical free-surface flow over a semi-circular obstruction. *Journal of Engineering Mathematics*, 22(1):3–13.

[42] Fornberg, B. (1998). *A Practical Guide to Pseudospectral Methods*. Cambridge University Press.

[43] Frigo, M. and Johnson, S. G. (2005). The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231.

[44] García-Ripoll, J. J. and Pérez-García, V. M. (2001). Optimizing Schrödinger functionals using Sobolev gradients: Applications to quantum mechanics and nonlinear optics. *SIAM Journal on Scientific Computing*, 23(4):1316–1334.

[45] Ghai, A., Lu, C., and Jiao, X. (2019). A comparison of preconditioned Krylov subspace methods for large-scale nonsymmetric linear systems. *Numerical Linear Algebra with Applications*, 26(1):e2215.

[46] Golub, G. H. and Welsch, J. H. (1969). Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230.

[47] Gottlieb, D. A., Hussaini, M. Y., and Orszag, S. A. (1984). Theory and applications of spectral methods.

[48] Govaerts, W. J. F. (2000). *Numerical Methods for Bifurcations of Dynamical Equilibria*. SIAM.

[49] Greenbaum, A. (1997). *Iterative Methods for Solving Linear Systems*. SIAM.

[50] Grimshaw, R. H. J. and Smyth, N. (1986). Resonant flow of a stratified fluid over topography. *Journal of Fluid Mechanics*, 169:429–464.

[51] Hackbusch, W. (2013). *Multi-grid Methods and Applications*, volume 4. Springer Science & Business Media.

[52] Haldenwang, P., Labrosse, G., Abboudi, S., and Deville, M. (1984). Chebyshev 3-D spectral and 2-D pseudospectral solvers for the Helmholtz equation. *Journal of Computational Physics*, 55(1):115–128.

[53] He, Y., Lamb, K. G., and Lien, R.-C. (2019). Internal solitary waves with subsurface cores. *Journal of Fluid Mechanics*, 873:1–17.

[54] Heinrichs, W. (1988a). Line relaxation for spectral multigrid methods. *Journal of Computational Physics*, 77(1):166–182.

[55] Heinrichs, W. (1988b). Multigrid methods for combined finite difference and Fourier problems. *Journal of Computational Physics*, 78(2):424–436.

[56] Heinrichs, W. (1990). Algebraic spectral multigrid methods. *Computer Methods in Applied Mechanics and Engineering*, 80(1-3):281–286.

[57] Heinrichs, W. (1991). A 3D spectral multigrid method. *Applied Mathematics and Computation*, 41(2):117–128.

[58] Heinrichs, W. (1992). A spectral multigrid method for the Stokes problem in stream-function formulation. *Journal of Computational Physics*, 102(2):310–318.

[59] Heinrichs, W. (1993a). Finite element preconditioning for spectral multigrid methods. *Applied Mathematics and Computation*, 59(1):19–40.

[60] Heinrichs, W. (1993b). Spectral multigrid methods for domain decomposition problems using patching techniques. *Applied Mathematics and Computation*, 59(2-3):165–176.

[61] Heinrichs, W. (1993c). Spectral multigrid methods for the reformulated Stokes equations. *Journal of Computational Physics*, 107(2):213–224.

[62] Heinrichs, W. (1993d). Spectral multigrid techniques for the Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 106(3):297–314.

[63] Helfrich, K. R. and White, B. L. (2010). A model for large-amplitude internal solitary waves with trapped cores. *Nonlinear Processes in Geophysics*, 17(4):303–318.

[64] Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436.

[65] Hussaini, M. Y., Salas, M. D., and Zang, T. A. (1983). Spectral methods for inviscid, compressible flows.

[66] Kang, Y. and Fulton, S. R. (1993). Relaxation schemes for Chebyshev spectral multigrid methods. In *NASA. Langley Research Center, The Sixth Copper Mountain Conference on Multigrid Methods, Part 1.*

[67] Keeler, J. S., Binder, B. J., and Blyth, M. G. (2017). On the critical free-surface flow over localised topography. *Journal of Fluid Mechanics*, 832:73–96.

[68] Keller, H. B. (1977). Numerical solution of bifurcation and nonlinear eigenvalue problems. *Application of Bifurcation Theory*, pages 359–384.

[69] Keller, H. B. (1987). Lectures on numerical methods in bifurcation problems. *Applied Mathematics*, 217:50.

[70] Kim, T. Y., Dolbow, J. E., and Fried, E. (2011). The Navier–Stokes-$\alpha\beta$ equations as a platform for a spectral multigrid method to solve the Navier–Stokes equations. *Computers & Fluids*, 44(1):102–110.

[71] Kuznetsov, Y. A., Kuznetsov, I. A., and Kuznetsov, Y. (1998). *Elements of Applied Bifurcation Theory*, volume 112. Springer.

[72] Kythe, P. K. and Schäferkotter, M. R. (2004). *Handbook of Computational Methods for Integration*. CRC Press.

[73] Lamb, K. G., Duda, T., and Farmer, D. (1999). Theoretical descriptions of shallow-water solitary internal waves: Comparisons with fully-nonlinear waves. In *The 1998 WHOI/IOS/ONR Internal Solitary Wave Workshop: Contributed Papers*, volume 209. July.

[74] Lamb, K. G. and Yan, L. (1996). The evolution of internal wave undular bores: comparisons of a fully nonlinear numerical model with weakly nonlinear theory. *Journal of Physical Oceanography*, 26(12):2712–2734.

[75] Laprise, R. and Peltier, W. R. (1989). On the structural characteristics of steady finite-amplitude mountain waves over bell-shaped topography. *Journal of the Atmospheric Sciences*, 46(4):586–595.

[76] Lee, S. and Whang, S. (2015). Trapped supercritical waves for the forced KdV equation with two bumps. *Applied Mathematical Modelling*, 39(9):2649–2660.

[77] Lien, R.-C., Henyey, F., Ma, B., and Yang, Y. J. (2014). Large-amplitude internal solitary waves observed in the northern South China Sea: properties and energetics. *Journal of Physical Oceanography*, 44(4):1095–1115.

[78] Long, R. R. (1953). Some aspects of the flow of stratified fluids: A theoretical investigation. *Tellus*, 5(1):42–58.

[79] Luzzatto-Fegiz, P. and Helfrich, K. R. (2014). Laboratory experiments and simulations for solitary internal waves with trapped cores. *Journal of Fluid Mechanics*, 757:354–380.

[80] Margairaz, F., Giometto, M. G., Parlange, M. B., and Calaf, M. (2018). Comparison of dealiasing schemes in large-eddy simulation of neutrally stratified atmospheric flows. *Geoscientific Model Development*, 11(10):4069–4084.

[81] Musslimani, Z. H. and Yang, J. (2004). Self-trapping of light in a two-dimensional photonic lattice. *JOSA B*, 21(5):973–981.

[82] O'hara, H. and Smith, F. J. (1968). Error estimation in the Clenshaw–Curtis quadrature formula. *The Computer Journal*, 11(2):213–219.

[83] Orszag, S. (1980). Spectral methods for problems in complex geometries. *Journal of Computational Physics*, 37(1):70–92.

[84] Orszag, S. A. (1971). On the elimination of aliasing in finite-difference schemes by filtering high-wavenumber components. *Journal of Atmospheric Sciences*, 28(6):1074–1074.

[85] Osborne, A. and Burch, T. (1980). Internal solitons in the Andaman sea. *Science*, 208(4443):451–460.

[86] Patterson, G. S. and Orszag, S. A. (1971). Spectral calculations of isotropic turbulence: Efficient removal of aliasing interactions. *The Physics of Fluids*, 14(11):2538–2541.

[87] Pelinovsky, D. E. and Stepanyants, Y. A. (2004). Convergence of Petviashvili's iteration method for numerical approximation of stationary solutions of nonlinear wave equations. *SIAM Journal on Numerical Analysis*, 42(3):1110–1127.

[88] Petviashvili, V. I. (1976). Equation of an extraordinary soliton. *Fizika Plazmy*, 2:469–472.

[89] Peyret, R. (1986). Introduction to spectral methods with application to fluid mechanics. *In Von Karman Inst. for Fluid Dynamics Computational Fluid Dynamics*, 1.

[90] Phillips, N. A. (1959). An example of non-linear computational instability. *The Atmosphere and the Sea in Motion*, 501:504.

[91] Phillips, T. N. (1987). Relaxation schemes for spectral multigrid methods. *Journal of Computational and Applied Mathematics*, 18(2):149–162.

[92] Phillips, T. N., Zang, T. A., and Hussaini, M. Y. (1986). Preconditioners for the spectral multigrid method. *IMA Journal of Numerical Analysis*, 6(3):273–292.

[93] Rayson, M. D., Jones, N. L., and Ivey, G. N. (2019). Observations of large-amplitude mode-2 nonlinear internal waves on the Australian North West shelf. *Journal of Physical Oceanography*, 49(1):309–328.

[94] Richardson, L. F. (1911). IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210(459-470):307–357.

[95] Rossby, C.-G. (1939). Relation between variations in the intensity of the zonal circulation of the atmosphere and the displacements of the semi-permanent centers of action. *Journal of Marine Research*, 2:38–55.

[96] Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. SIAM.

[97] Saad, Y. and Schultz, M. H. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869.

[98] Sandstrom, H. and Elliott, J. (1984). Internal tide and solitons on the Scotian Shelf: A nutrient pump at work. *Journal of Geophysical Research: Oceans*, 89(C4):6415–6426.

[99] Schneider, C. and Werner, W. (1986). Some new aspects of rational interpolation. *Mathematics of Computation*, 47(175):285–299.

[100] Scotti, A. and Pineda, J. (2004). Observation of very large and steep internal waves of elevation near the Massachusetts coast. *Geophysical Research Letters*, 31(22).

[101] Seydel, R. (2009). *Practical Bifurcation and Stability Analysis*, volume 5. Springer Science & Business Media.

[102] Shen, S. S. P. (1991). Locally forced critical surface waves in channels of arbitrary cross section. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 42(1):122–138.

[103] Soontiens, N., Stastna, M., and Waite, M. L. (2013). Trapped internal waves over topography: Non-boussinesq effects, symmetry breaking and downstream recovery jumps. *Physics of Fluids*, 25(6).

[104] Soontiens, N., Subich, C., and Stastna, M. (2010). Numerical simulation of supercritical trapped internal waves over topography. *Physics of Fluids*, 22(11).

[105] Stastna, M. (2022). *Internal Waves in the Ocean: Theory and Practice*, volume 9. Springer Nature.

[106] Stastna, M. and Lamb, K. G. (2002). Large fully nonlinear internal solitary waves: The effect of background current. *Physics of Fluids*, 14(9):2987–2999.

[107] Streett, C. L., Zang, T. A., and Hussaini, M. Y. (1985). Spectral multigrid methods with applications to transonic potential flow. *Journal of Computational Physics*, 57(1):43–76.

[108] Trefethen, L. N. (2000). *Spectral Methods in MATLAB*. SIAM.

[109] Trefethen, L. N. (2008). Is Gauss quadrature better than Clenshaw–Curtis? *SIAM Review*, 50(1):67–87.

[110] Trefethen, L. N. and Bau, D. (2022). *Numerical Linear Algebra*. SIAM.

[111] Trottenberg, U., Oosterlee, C. W., and Schuller, A. (2000). *Multigrid*. Elsevier.

[112] Turkington, B., Eydeland, A., and Wang, S. (1991). A computational method for solitary internal waves in a continuously stratified fluid. *Studies in Applied Mathematics*, 85(2):93–127.

[113] Wu, T. Y.-T. (1987). Generation of upstream advancing solitons by moving disturbances. *Journal of Fluid Mechanics*, 184:75–99.

[114] Xu, C., Subich, C., and Stastna, M. (2016). Numerical simulations of shoaling internal solitary waves of elevation. *Physics of Fluids*, 28(7).

[115] Yang, J. (2009). Newton-conjugate-gradient methods for solitary wave computations. *Journal of Computational Physics*, 228(18):7007–7024.

[116] Yang, J. (2010). *Nonlinear Waves in Integrable and Nonintegrable Systems*. SIAM.

[117] Yang, J. and Lakoba, T. I. (2007). Universally-convergent squared-operator iteration methods for solitary waves in general nonlinear wave equations. *Studies in Applied Mathematics*, 118(2):153–197.

[118] Yang, J. and Lakoba, T. I. (2008). Accelerated imaginary-time evolution methods for the computation of solitary waves. *Studies in Applied Mathematics*, 120(3):265–292.

[119] Young, D. (1954). Iterative methods for solving partial difference equations of elliptic type. *Transactions of the American Mathematical Society*, 76(1):92–111.

[120] Zang, T. A. and Hussaini, M. Y. (1986). On spectral multigrid methods for the time-dependent Navier–Stokes equations. *Applied Mathematics and Computation*, 19(1-4):359–372.

[121] Zang, T. A., Wong, Y. S., and Hussaini, M. Y. (1982). Spectral multigrid methods for elliptic equations. *Journal of Computational Physics*, 48(3):485–501.

[122] Zang, T. A., Wong, Y. S., and Hussaini, M. Y. (1984). Spectral multigrid methods for elliptic equations II. *Journal of Computational Physics*, 54(3):489–507.

[123] Zhang, W., Zhang, C. H., and Xi, G. (2010). An explicit Chebyshev pseudospectral multigrid method for incompressible Navier–Stokes equations. *Computers & Fluids*, 39(1):178–188.