A NUMERICAL APPROACH FOR THE INTERFACIAL MOTION BETWEEN TWO IMMISCIBLE INCOMPRESSIBLE FLUIDS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the Graduate

School of the Ohio State University

By

Jin Wang, M.S.

* * * * *

The Ohio State University
2004

Dissertation Committee:

Prof. Greg Baker, Advisor

Prof. Saleh Tanveer

Prof. Ed Overman

Approved by

Advisor
Department Of Mathematics

ABSTRACT

Incompressible flows with interfaces occur in a wide variety of physical phenomena as well as technological processes. Mathematically, the motion is governed by the incompressible Navier-Stokes equations together with interfacial conditions.

In this thesis, we present a numerical approach to simulate the two-dimensional viscous, incompressible flows with interfaces. First we introduce some new coordinates so that the interface is mapped into a coordinate line which enables us to work on a rectangular domain instead of a deformed geometry. Then an iterative approach combined with an implicit time marching method is applied to update the motion in time. At each iterate, the Fourier transform and the pseudo-spectral technique are applied in the horizontal direction, X, under the assumption that the solutions are periodic in X. Then we write the semi-discretized equations as a 1st-order ODE system with respect to the vertical coordinate, Z, and an efficient ODE solver is developed to construct the solutions.

As an application of our numerical approach, we study the problem of steady progressive interfacial waves (Stokes waves). In contrast to all the previous work which was concerned with inviscid fluids, we study Stokes waves in the presence of viscosity. Our numerical results show that the effect of viscosity is somehow equivalent to the decay of the expansion parameter in the series expansion of the inviscid Stokes

waves. Our work suggests a new expansion form for Stokes waves in viscous fluids. In addition, we perform a similar study for the viscous effects on standing waves.

Finally, some analysis is applied for the linearized motion. In particular, the asymptotic solution to the linearized interfacial flow is derived.

Dedicated to my parents and wife

ACKNOWLEDGMENTS

I wish to express my heartiest thanks to my adviser, Greg Baker, for his guidance, encouragement and support throughout my graduate study at Ohio State. I have greatly enjoyed numerous discussions with him and benefited tremendously from his advice. I sincerely appreciate his patience in carefully reading this thesis and his many valuable suggestions for improvement.

I would like to thank Saleh Tanveer, Ed Overman and Bjorn Sandstede for their inspiring classes. Special thanks go to Saleh Tanveer and Ed Overman for serving on my Dissertation Committee.

I would like to thank the computer staff in the mathematics department, especially Dave Alden, for the assistance in computing. I would also like to thank the Ohio Supercomputer Center for the high performance computing resources.

Finally, I wish to express my deep gratitude to my wife for all her love and support.

VITA

1998	B.S. in Mathematics, University of Science and Technology of China.
1999	B.S. in Economics, University of Science and Technology of China.
2000	M.S. in Mathematics, University of Science and Technology of China.
2000 – present	Graduate Research and Teaching Associate, The Ohio State University.

PUBLICATIONS

1. J. Wang and R. X. Liu, A new approach to design high-order schemes, *J. Comput. Appl. Math.*, vol. 134, pp. 59-67, 2001.

2. J. Wang and R. X. Liu, Some generalizations of classical MPDE approach, *J. Univ. Sci. Tech. China*, vol. 31, pp. 143-150, 2001.

3. J. Wang and R. X. Liu, The remainder-effect analysis of upwind leapfrog schemes, *Math. Appl.*, vol. 13, pp. 84-90, 2000.

FIELDS OF STUDY

Major field: Mathematics

Specialization: Numerical Analysis and Scientific Computing

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview of computational fluid dynamics

It's well known the field of fluid mechanics generates many problems in the form of partial differential equations (PDE). These equations describe the motion and the energy of fluid flows and we have to solve such equations, together with some boundary conditions and/or initial conditions, to understand the physics involved. Basically there are three approaches to proceed: theoretical, experimental and computational.

The theoretical approach uses mathematical theory to seek the solution of a problem. For simple cases, an exact solution in closed form can be achieved which gives clean and general information about the physics involved in the problem. However, this can only be applied to a limited number of physical problems, usually in linear cases and with simple geometry. For the majority of problems in fluid mechanics, which usually possess nonlinearity and involve complex physics and geometry, we have to give up finding the exact solutions. A different theoretical approach, known as asymptotic, applies to problems involving one or more small parameters. Asymptotic methods construct a series expansion, called asymptotic expansion, in terms of the small parameter(s) to approximate the solution. This approximation

becomes increasingly accurate when letting the small parameter(s) tend to zero. The asymptotic technique can be applied to many nonlinear problems and the two most common asymptotic methods are the method of matched asymptotic expansions and the method of multiple scales [44][67].

The experimental approach uses apparatus and measuring devices in a lab to model and simulate and record results. If carefully handled, experiments can give very realistic results to many physical problems. Hence the experimental approach offers a direct way to demonstrate the real physics, which is a big advantage. However, it's not always possible to model a physical problem in a lab. For example, some important phenomena involved in liquid-gas interfaces often happen on such scales of space and time that experimental visualization is very difficult of even imposible. Another disadvantage of the experimental approach is that the costs for equipment are usually high.

The computational approach develops numerical methods and uses digital computers as tools to find approximate solutions. This approach can treat both linear and nonlinear problems, can handle both simple and complex geometry and is capable of attacking large-scale problems. Hence the computational approach can provide information not available by the other two approaches and can offer a powerful way to improve our understanding of complicated physics. In recent years this approach has been increasingly important in the study of fluid mechanics, as well as many other fields of science and engineering. More and more good numerical methods are being designed which enable us to attack more and more challenging problems. The continous improvement of computational power keeps extending the range of affordable

problems in numerical simulations. On the other hand, the computational approach also has limitations. Numerical methods cannot find exact solutions and they are always associated with numerical errors which can be disasterous in some situations. Meanwhile, a common difficulty to many numerial methods lies in the treatment of boundary conditions. (We will illustrate this issue in next section.)

It's important to note that these three approaches are closely related, rather than separated, from each other. The theoretical approach provides mathematical background to the other two approaches and often provides good insight, if not the solutions, to the problems. The experimental and the computational approaches can, in many cases, check and justify the results with each other. The results from these two approaches can, in turn, motivate the development of rigorous mathematical theory. In this thesis we are mainly concerned with the computational approach, though some discussion will be given to the asymptotic solutions.

The history of computational fluid dynamics (CFD) starts back in the early stage of the development of digital computers. A milestone in CFD was due to the famous paper by Courant, Friedrichs and Lewy in 1928 [14]. In this paper, existence and uniqueness questions were addressed for the numerical solutions of PDEs. In particular, a stability requirement, now commonly refered to as the CFL condition, was proposed for numerical solutions of hyperbolic PDEs. Later in the 1940's, von Neumann developed his method for evaluating the stability of numerical methods in time evolution problems. Since then this method has been the most widely applied technique for determining the numerical stability [47][54]. In the 1950's, much progress was made for numerically solving elliptic and parabolic equations and for

calculating shock waves. Typical work includes the successive overrelaxation (SOR) scheme by Frankel [23] for Laplace's equation, the conservative scheme by Peter Lax [39] for shock capturing, the alternating direction implicit (ADI) method by Douglas and Rachford [17], the particle-in-cell (PIC) method by Evans and Harlow [20], etc. In recent decades many new techniques, such as finite volume methods, boundary-integral equation methods and spectral methods, have been substantially extended. Meanwhile, many new ideas, such as the multigrid [43], the total variation diminishing (TVD) [31] and the essentially non-oscillating method (ENO) [32], have been introduced and widely applied. In recent years, both the number of researchers and the progress made in the area of CFD have been expanding rapidly with significant impact in not only the field of fluid mechanics, but also many other disciplines including material science, biological science, chemical engineering, industrial engineering, etc.

1.2 Numerical simulation to incompressible Navier-Stokes equations

Normally most fluids, including air and water, can be treated as incompressible fluids and their motions are described by the incompressible Navier-Stokes equations [8]. That's why the incompressible Navier-Stokes equations have a fundamental importance in fluid mechanics. Consequently, the numerical simulations to such equations have been a very active area in CFD.

We consider the two-dimensional unsteady incompressible Navier-Stokes equations

for a flow with constant density ρ and viscosity μ and without external forces. Let's denote the velocity vector by $V = (u, w)$, the pressure by p. The continuity of the momemtum and the mass give

$$\rho V_t + \rho V \cdot \bigtriangledown V = -\bigtriangledown p + \mu \bigtriangledown^2 V , \tag{1.1}$$

$$\bigtriangledown \cdot V = 0 , \tag{1.2}$$

where t is the temporal coordinate and $\bigtriangledown = (\frac{\partial}{\partial x}, \frac{\partial}{\partial z})$ the spacial gradient. Equation (1.2) is commonly called the incompressibility condition and it states that the velocity field is divergence free.

One major difficulty associated with the numerical study for equations (1.1) and (1.2) is that the incompressibility condition has to be satisfied at all times, but there is no time-derivative in equation (1.2). Thus, as the flow field is updated in time through (1.1), the incompressibility condition must somehow be satisfied implicitly through the computation.

There have been many numerical methods developed to solve equations (1.1) and (1.2). In what follows we briefly review some representative methods.

1. Pressure equation approach.

By taking the divergence of the momentum equation (1.1) and using the incompressibility condition, we obtain a Poisson equation for the pressure

$$\bigtriangledown^2 p = -\rho \bigtriangledown \cdot (V \cdot \bigtriangledown V) . \tag{1.3}$$

Numerically, at each time step, we solve the pressure equation (1.3) first, then do the time evolution to the momentum equation to find the velocity. The essential

idea of this method is that we use the pressure equation to replace the incompressibility condition and, consequently, the calculation of the velocity and the pressure is separated. On the other hand, this replacement can give rise to big numerical errors, leading to the violation of the incompressibility condition. Consequently, high resolution is required to ensure the incompressibility condition. Efforts have been made to overcome this problem by modifying the pressure equation. This includes the famous marker-and-cell (MAC) method [30][71]. Another issue with this method is that special treatment is needed for the pressure boundary conditions since there is no physical boundary condition for the pressure. An useful way to do this is to project the momemtum equation in the normal direction of the boundary [28][29].

2. Projection methods.

They belong to the category of fractional step methods. At each time step we first calculate an intermediate velocity V^*, which does not necessarily satisfy the incompressibility condition, then we project V^* into a divergence-free field to obtain the correct velocity.

There are many versions of the projection method (*e.g.* [9][37][68]). One of them is due to Van Kan [68]:

$$\frac{V^* - V^n}{\Delta t} + (V \cdot \bigtriangledown V)^{n+\frac{1}{2}} = \frac{1}{2} \bigtriangledown^2 (V^* + V^n) - \bigtriangledown p^n \,, \tag{1.4}$$

$$\frac{V^{n+1} - V^*}{\Delta t} = -\frac{1}{2} \bigtriangledown (p^{n+1} - p^n) \,, \tag{1.5}$$

$$\bigtriangledown \cdot V^{n+1} = 0 \,, \tag{1.6}$$

where $(V \cdot \bigtriangledown V)^{n+\frac{1}{2}}$ represents some average for the nonlinear term. Note that (1.5) and (1.6) yield a Poisson equation for the pressure and ensure the incompressibility

condition. It turns out that many projection methods only achieve 1st-order accuracy for the pressure [11][18], partly due to the difficulty in obtaining the pressure boundary conditions.

3. Vorticity-stream function formulation

Define the vorticity Ω by

$$\Omega = \frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}\,. \tag{1.7}$$

Also define the stream function Φ by

$$\frac{\partial \Phi}{\partial z} = u\,, \qquad \frac{\partial \Phi}{\partial x} = -w\,. \tag{1.8}$$

By taking the curl of the momentum equation (1.1) we can eliminate the pressure and obtain

$$\frac{\partial \Omega}{\partial t} + u\frac{\partial \Omega}{\partial x} + w\frac{\partial \Omega}{\partial z} = \frac{\mu}{\rho}\,(\frac{\partial^2 \Omega}{\partial x^2} + \frac{\partial^2 \Omega}{\partial z^2})\,. \tag{1.9}$$

Meanwhile, (1.7) and (1.8) give

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial z^2} = -\Omega\,. \tag{1.10}$$

As a result, we are able to transfer the mixed elliptic-parabolic Navier-Stokes equations into one parabolic equation (1.9) and one elliptic equation (1.10), which can then be solved separately. This is a great simplification for the numerical implementation. Unfortunately, for three-dimensional problems, this method no longer possessess such an easy simplification (actually gets much complicated); hence it is not favored for 3-D problems [3].

4. Artificial compressibility methods.

In order to overcome the difficulty with the incompressibility constraint, Chorin [13] introduced an artificial time derivative for the pressure and replaced the incompressibility condition by

$$\beta \frac{\partial p}{\partial t} + u_x + w_z = 0\,, \tag{1.11}$$

where β is a factor related to Δt. In this way one can update the velocity and the pressure simultaneously. The original method of Chorin was only valid for steady-state problems (when $\frac{\partial p}{\partial t} \to 0$). Similar methods for time-dependent problems were developed by Peyret and Taylor [52], Rogers *et al.* [55]. The disadvantage of these methods is that the equations may become highly stiff and require implicit treatment which requires a large amount of memory usage.

1.3 Computation of incompressible interfacial flows

We have already seen that it's not easy to compute the incompressible Navier-Stokes equations. Now, to make things more challenging, we want to simulate the flows of two immiscible and incompressible fluids with a sharp interface. Such interfacial flows occur in a wide variety of physical phenomena such as bubbles, droplets, cavities, ice melting in water, wind-water wave interaction, as well as a large number of technological processes such as jets, casting, mold filling, thin films, just to name a few. Consequently, numerical simulations to these problems have been making great strides in many disciplines in science and engineering, for example, geophysics, oceanography, material science, chemical engineering, and so on.

In addition to the difficulties in the simulation of the incompressible Navier-Stokes

equations, we now have a new challenge in that the domain of interest contains an unknown interface which evolves in time and which must be determined as part of the solution. The interface plays a major role in defining the system and it's important to have an accurate representation of the interface.

A couple of methods have been developed for tracking or capturing the interface. Some popular methods are summarized below.

1. Volume-of-fluid (VOF) methods.

VOF methods have been in use for several decades. Early work includes the SLIC algorithm of Noh and Woodward [45] and the SOLA-VOF algorithm of Hirt and Nicols [35]. Since then, significant progress has been made on VOF methods and a review of recent work can be found in [58].

In the VOF method, a volume fraction function is defined by

$$C(x,z) = \begin{cases} 1, & \text{if } (x,z) \text{ is in upper fluid}, \\ 0, & \text{if } (x,z) \text{ is in lower fluid}. \end{cases} \tag{1.12}$$

A point (x,z) lies in the interface if and only if $0 < C(x,z) < 1$. The function $C(x,z)$ satisfies the advection equation

$$\frac{\partial C}{\partial t} + V \cdot \bigtriangledown C = 0\,, \tag{1.13}$$

where the velocity V is obtained from the Navier-Stokes equations. At each time the values of $C(x,z)$ are used to reconstruct an approximation to the interface and this approximate interface is then used to update the volume fractions at the next time. VOF methods provide a simple way to handle the topological changes of the interface and are relatively easy to extend from two-dimensional to three-dimensional

domains. However, these methods cannot follow the small structure of the interface and are not good at capturing the fine-scale boundary layers near the interface.

2. Level set methods.

The level set approach was first proposed by Osher and Sethian [49] and has since been widely applied to many interfacial/free-surface problems including bubbles and drops, Rayleigh-Taylor instability, flow by mean curvature, etc. In these methods, a level set function $\phi(x, z, t)$ is introduced with the initial value

$$\phi(x, z, t = 0) = \pm d \tag{1.14}$$

where d is the shortest distance from the point (x, z) to the initial interface and where the sign of ϕ indicates whether (x, z) is in the upper or the lower fluid. The level set function ϕ evolves in response to the propagation of the interface and the evolution is given by

$$\frac{\partial \phi}{\partial t} + V \cdot \bigtriangledown \phi = 0 \tag{1.15}$$

which is similar to (1.13). At anytime, the zero level set, $\phi = 0$, gives exactly the location of the interface. These methods, like the VOF methods, do not require special procedures to treat topological changes of the interface and are relatively simple to generalize to three-dimensional problems. The disadvantages, however, are that level set methods have inherent numerical dissipation which will smooth the interface and lead to nonphysical loss of mass.

3. Boundary-integral equation (BIE) methods.

These methods were developed for potential flows and notable work in this category was made by Longuet-Higgins and Cokelet [42], Vinje and Brevig [69], Baker *et*

al [5], etc. In these methods, Laplace's equation is solved by using Green's functions, leading to Fredholm integral equations of the second kind. The dynamic and kinamatic interfacial/free-surface conditions are integrated to update the interface/free surface at each time. A distinct advantage of the BIE methods is that the space dimension of the problem is reduced by one. Hence the BIE methods offer an efficient way for the computation of inviscid and irrotational flows. Unfortunately, these methods are not applicable to general viscous flows.

In addition, there are some other methods such as marker-and-cell method [30][71], front-tracking method [26], which have also achieved much success in the interface simulations. All these methods have their strength and weakness and a perfect approach does not yet exist.

Probably the most important two-fluid system is the one with air and water with particular importance in geophysical flows. An example is the generation of sea surface waves which can affect both the local and the global climate, and affect all commercial activities related to the oceans. Despite its ubiquitous presence and importance, understanding of the physics involved remains limited due to the nonlinear phenomenon implicit in both air flow and water wave evolution [4].

We hope to perform a careful numerical study of the interface evolution with as much accuracy as possible for the interface profile and the boundary layers, under the assumption that the interface remain single-valued and no dramatic change occurs for its topology. The methods summarized above do not appear optimal for an accurate

study of the detailed interactions of the two fluids at the interface. Hence it's worthwhile to explore new approaches to improve our understanding of the fundamental physics involved.

A new approach has been developed in our research to simulate the two-dimensional viscous incompressible flows with interfaces. The basic idea of the approach is as follows. New coordinates, referred to as logical coordinates, are introduced so that the interface is mapped into a coordinate line which enables us to work on a rectangular domain instead of the deformed geometry. An iterative approach combined with the Crank-Nicolson scheme or the backward difference formula is applied for the evolution of the interface to ensure time-stepping stability. To perform the space discretization in the horizontal direction, X, the Fourier transform and the pseudo-spectral technique are applied under the assumption that the solutions are periodic in X. Then we write the semi-discretized equations as a 1st-order ODE system with respect to the vertical coordinate, Z, and an efficient ODE solver is developed to construct the solutions. The incompressibility condition is treated as one equation in the ODE system so that it's automatically satisfied at each time step. The methods achieve uniform order of accuracy for the velocities, the pressure and the interface profile: 2nd order for both t and Z, and spectral accuracy for X.

1.4 Interfacial waves

By applying our numerical methods we are able to study interfacial waves moving between two different fluids. In this thesis we consider two kinds of such waves:

Stokes waves and standing waves. In future, our plan is to numerically simulate the generation of water waves (like sea surface waves) by wind forcing and study their subsequent interaction. This will be an interesting topic in our future research.

The problem of steady progressive free-surface or interfacial waves (Stokes waves) is one of the oldest in the field of mathematical fluid mechanics and a large body of work has been done on this subject. Stokes [63] was the first to systematically study the properties of surface water waves by using the technique of series expansion called the Stokes' expansion. He computed the solution to fifth-order for the deep-water case and to 3rd-order for finite depth. Thereafter, much effort has been devoted to Stokes' theory by various investigators. Levi Civita [41] proved the convergence of Stokes' expansion for sufficiently small waves. De [16] published a fifth-order solution to general depth. Schwartz [59] were able to calculate the expansion to extremely high orders by using the digital computer to perform the coefficient arithmetic. By noting that "waves which occur in nature are never, in fact, free surface waves", Tsuji and Nagata [65] and Holyer [36] applied the Stokes' expansion to the interfacial waves moving between two fluids of different densities. We note that all the works mentioned here were concerned with inviscid fluids.

We let x, z be the horizontal and the vertical coordinates, respectively, and t the temporal coordinate. In Stokes' expansion, the profile of a surface/interfacial wave in a frame moving with the phase speed can be written in a non-dimensional form

$$z = \sum_{k=1}^{\infty} A_k(A) \cos kx \ , \tag{1.16}$$

where the coefficients A_k $(k = 1, 2, 3, \cdots)$ only depend on a free parameter A. One of

the main objectives in the work mentioned above is to calculate these coefficients to high-order so that they can give good approximations to the highest stable wave. This is not the goal of the present work. Instead, our interest is to study the viscous effects on Stokes waves. While there are no truly free surface waves in nature, there are also no truly inviscid fluids in nature either. We ask: What happens to a Stokes wave in the presence of viscosity? It turns out the coefficients A_k $(k = 1, 2, 3, \cdots)$ appear to decay with time in a 'nice' pattern, i.e., the effect of the viscosities is somehow equivalent to the decay of the expansion parameter A in the series expansion of the inviscid Stokes waves. Our work also suggests a new expansion form for Stokes waves in viscous fluids.

Another topic in fluid mechanics, which is as attractive as Stokes waves, is the motion of standing waves. A standing wave is stationary in the horizontal direction and makes periodic oscillations between crest and trough in the vertical direction. It can be expanded in a similar way as (1.16)

$$z = \sum_{k=1}^{\infty} A_k(A, t) \cos kx \ , \tag{1.17}$$

where the coefficients A_k $(k = 1, 2, 3, \cdots)$ depend on both the time and the parameter A. Various investigators have studied such waves [51][56][60] in the case of inviscid fluids. It's of our interest to ask the same question as for the Stokes waves: What happens to a standing wave in the presence of viscosity? Based on the results for Stokes waves, we speculate that a similar decay pattern also holds for standing waves. Hoewever, our numerical tests show that there is a disagreement with this pattern,

starting from the 4th mode A_4. The reason for this disagreement is not clearly understood yet and that remains an open question in our research.

1.5 Summary of the thesis

This thesis is mainly concerned with the numerical methods developed for computing the two-dimensional interfacial flows between two immiscible and incompressible viscous fluids. The thesis is organized as follows.

In Chapter 2, we review some basic techniques in numerical methods including finite difference methods, discrete Fourier transform, ODE solvers and iterative methods for linear systems. The ideas closely related to our numerical methods applied to interfacial flows are emphasized.

In Chapter 3, we describe in detail our numerical methods for computing incompressible flows with interfaces. Major discussion is devoted for the first approach which is constructed through the extraction of linearized terms. In addition, we also discuss a different approach which employs the generalized minimum residual (GMRES) algorithm.

In Chapter 4, we provide the numerical results from our methods. Two examples serve for the numerical verification of the accuracy of our methods. Then the Stokes waves in the presence of viscosity are investigated and a new expansion form is suggested. Similar study is also performed for standing waves. Finally the parallelization of the numerical methods is briefly discussed.

In Chapter 5, we apply some analysis to the linearized motion. This includes

the asymptotic solution of the linear problem and the justification of the order of accuracy of our numerical methods applied to the linear problem.

CHAPTER 2

BACKGROUND

2.1 Finite difference methods

Among various numerical methods, finite difference methods have the longest history and probably the widest applications. The basic idea of finite difference methods is that we use differences, constructed on appropriate grid points, to approximate the derivatives of functions. For example,

$$\left.\frac{\mathrm{d}u}{\mathrm{d}t}\right|_{t=n\Delta t} \approx \frac{u^{n+1}-u^n}{\Delta t} \approx \frac{u^{n+1}-u^{n-1}}{2\Delta t}\,, \qquad \left.\frac{\mathrm{d}^2u}{\mathrm{d}t^2}\right|_{t=n\Delta t} \approx \frac{u^{n+1}-2u^n+u^{n-1}}{(\Delta t)^2}\,, \qquad \text{etc.}$$

where u^n denotes the value of u at $t = n\Delta t$. In our notation, t usually refers to the temporal coordinate and x, z the spacial coordinates. By using finite differences, we can transfer a differential equation defined in a continuous space into one or more algebraic equations defined in a discrete space. Then we solve these algebraic equations to obtain the numerical solutions to that differential equation.

For a given differential equation there can be many ways to discretize the equation by using finite difference methods. Some of these methods give better approximation to the solution than other ones. That means, different methods have different

accuracy. Consider a general mth-order ODE with respect to t,

$$L\big(t, u, \frac{\mathrm{d}u}{\mathrm{d}t}, \cdots, \frac{\mathrm{d}^m u}{\mathrm{d}t^m}\big) = 0\,. \tag{2.1}$$

Let

$$L_{\Delta t}\big(t, u, \frac{\mathrm{d}u}{\mathrm{d}t}, \cdots, \frac{\mathrm{d}^m u}{\mathrm{d}t^m}\big)$$

be a finite difference approximation with step size Δt to $L\big(t, u, \frac{\mathrm{d}u}{\mathrm{d}t}, \cdots, \frac{\mathrm{d}^m u}{\mathrm{d}t^m}\big)$. By performing the Taylor's series expansion for $L_{\Delta t}$ at some moment, say $t = n\Delta t$, we obtain, generally,

$$L_{\Delta t} = L + T\,, \qquad \text{with} \quad T = C_p \Delta t^p + C_{p+1}\Delta t^{p+1} + \cdots\,, \tag{2.2}$$

where $C_p\,, C_{p+1}\,, \cdots$ are constants and $C_p \neq 0$. We call T in (2.2) the *(global) truncation error* or *discretization error* of the method and p the *order of accuracy* for the method.

There are many other concepts related to finite difference methods such as stability, consistency, convergence, dispersion, etc. We refer to the book of Richtmyer and Morton [54] and that of Anderson, Tannehill and Pletcher [3] for a comprehensive study of these issues.

In what follows, we consider a typical 1st-order ODE

$$\frac{\mathrm{d}u}{\mathrm{d}t} + g(u,t) = f(u,t)\ , \tag{2.3}$$

where g usually stands for linear terms and f the nonlinear terms. We illustrate some commonly used second-order finite difference methods here. These ideas are closely

related to our numerical methods applied to the interfacial flow.

1) second-order BDF method:

$$\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta t} + g^{n+1} = f^{n+1} \ . \tag{2.4}$$

2) BDF with a 2nd-order extrapolation for f:

$$\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta t} + g^{n+1} = 2f^n - f^{n-1} \ . \tag{2.5}$$

3) General BDF/extrapolation formula:

$$\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta t} + \beta g^{n+1} + (1-\beta)(2g^n - g^{n-1}) = \alpha f^{n+1} + (1-\alpha)(2f^n - f^{n-1}), \tag{2.6}$$

where α, β are two real parameters and we usually require $0 \le \alpha \le 1$, $0 \le \beta \le 1$. In particular, when $\alpha = \beta = 1$, (2.6) is reduced to (2.4) and when $\alpha = 0$, $\beta = 1$, (2.6) is reduced to (2.5).

4) Crank-Nicolson method:

$$\frac{u^{n+1} - u^n}{\Delta t} + \frac{1}{2}(g^{n+1} + g^n) = \frac{1}{2}(f^{n+1} + f^n) \ . \tag{2.7}$$

5) Crank-Nicolson with Adams-Bashforth:

$$\frac{u^{n+1} - u^n}{\Delta t} + \frac{1}{2}(g^{n+1} + g^n) = \frac{3}{2}f^n - \frac{1}{2}f^{n-1} \ . \tag{2.8}$$

6) Leapfrog method:

$$\frac{u^{n+1} - u^{n-1}}{2\Delta t} + g^n = f^n \ . \tag{2.9}$$

7) Leapfrog with Crank-Nicolson:

$$\frac{u^{n+1}-u^{n-1}}{2\Delta t}+g^n=\frac{1}{2}(f^{n+1}+f^{n-1})\ . \tag{2.10}$$

8) General Leapfrog/Crank-Nicolson formula:

$$\frac{u^{n+1}-u^{n-1}}{2\Delta t}+\beta g^n+(1-\beta)\frac{1}{2}(g^{n+1}+g^{n-1})=\alpha f^n+(1-\alpha)\frac{1}{2}(f^{n+1}+f^{n-1})\,, \tag{2.11}$$

where $0\le\alpha\le 1$, $0\le\beta\le 1$. In particular, when $\alpha=\beta=1$, we recover (2.9); when $\alpha=0,\ \beta=1$, we recover (2.10).

In all the above methods, we approximate the governing equation (2.3) at $t=(n+1)\Delta t$. That's to say, $n+1$ refers to the current time level and $n, n-1$, etc., refer to previous time levels. Hence, variables with the superscript $n+1$ are unknowns and must be computed, while variables with superscripts n or $n-1$ are already known.

We notice that in the methods (2.5), (2.8) and (2.9) f^{n+1} doesn't appear. Instead, some linear combinations of f^n and f^{n-1} are used to discretize f. We call them *explicit* treatments of f. In the other methods, f^{n+1} appears and we call them *implicit* treatments of f. Similarly we can discuss the explicit/implicit treatments of g. If a method treats both f and g explicitly, we say the method is explicit. Otherwise we say the method is implicit. The advantage of explicit methods is that, they usually generate simpler algebraic equations in the discrete space and numerical solutions can be obtained with less effort. The disadvantage is that many explicit methods are numerically unstable unless the time step is prohibitively small. On the

other hand, implicit methods generally have good stability properties and that's the main reason why they are prefered in many numerical applications. However, implicit methods usually result in more complicated algebraic equations and more than often some iterative approaches have to be applied to obtain the solutions.

2.2 Discrete Fourier transform

A real function f which is periodic with period X can be represented by the Fourier series

$$f(x) = a_0 + \sum_{k=1}^{\infty} \left(a_k \cos \frac{2\pi}{X} kx + b_k \sin \frac{2\pi}{X} kx \right) , \tag{2.12}$$

where

$$\begin{aligned} a_0 &= \frac{1}{X} \int_0^X f(x)\, \mathrm{d}x \ , \\ a_k &= \frac{2}{X} \int_0^X f(x) \cos kx \, \mathrm{d}x \, , \qquad k = 1, 2, \cdots \ , \\ b_k &= \frac{2}{X} \int_0^X f(x) \sin kx \, \mathrm{d}x \, , \qquad k = 1, 2, \cdots \ . \end{aligned} \tag{2.13}$$

The complex version of a Fourier series takes the form

$$f(x) = \sum_{k=-\infty}^{\infty} c_k \exp \left(\frac{2\pi}{X} ikx \right) \ , \tag{2.14}$$

where

$$c_k = \frac{1}{X} \int_0^X f(x) \exp \left(- \frac{2\pi}{X} ikx \right) \mathrm{d}x \, , \qquad k = 0, \pm 1, \pm 2, \cdots \ . \tag{2.15}$$

When f is real, $c_{-k} = c_k^*$ ($*$ refers to the complex conjugate) and the real Fourier coefficients and the complex Fourier coefficients are related by

$$\begin{aligned} c_0 &= a_0\,, \\ c_k &= \frac{1}{2}(a_k - ib_k)\,, \qquad k = 1, 2, \cdots, \\ c_{-k} &= \frac{1}{2}(a_k + ib_k)\,, \qquad k = 1, 2, \cdots. \end{aligned} \tag{2.16}$$

In practical applications, we can only compute a finite number of Fourier coefficients and we are more interested in the discretized form of the Fourier transform.

We pick two positive integers N, M with $N = 2M + 1$ and partition the domain $[0,\, X]$ into N equally spaced intervals by $x_j = j\,X/N\,,\ \ j = 0, 1, 2, \cdots, N-1\,.$ Then we write

$$f(x_j) = \sum_{k=-M}^{M} \hat{c}_k \exp\big(\frac{2\pi}{X} ikx_j\big)\,, \qquad j = 0, 1, \cdots, N-1\,. \tag{2.17}$$

It's more convenient to write (2.17) as

$$f_j = \sum_{k=-M}^{M} \hat{c}_k \exp\big(\frac{2\pi}{N} ikj\big)\,, \qquad j = 0, 1, \cdots, N-1\,. \tag{2.18}$$

If we treat $\{\hat{c}_k\}$ as unknowns in (2.18), we can solve the linear system to obtain

$$\hat{c}_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j \exp\big(\frac{-2\pi}{N} ikj\big)\,, \qquad k = -M, -M+1, \cdots, M\,. \tag{2.19}$$

Equations (2.19) and (2.18) define the discrete Fourier transform and the inverse discrete Fourier transform, respectively. Note that the discrete Fourier coefficient $\hat{c}_k$ defined in (2.19) is generally different from the continuous Fourier coefficient c_k defined in (2.15). However, $\hat{c}_k \to c_k$ as $N \to \infty$. In a computer, (2.19) or (2.18)

can be computed rapidly by the well-established algorithm called the Fast Fourier Transform (FFT). The FFT is most efficient when N is an even number. Hence, we usually drop the coefficient $\hat{c}_M$ when performing the transform. That means, we set $N = 2M$ and replace M by $M - 1$ in (2.19) and (2.18).

The discrete Fourier transform is widely applied in numerical computations when dealing with functions which possess periodicity. Let's illustrate the basic idea by considering the Burgers' equation

$$f_t + f f_x = \nu f_{xx}\,, \qquad x \in [0, 2\pi]\,, \quad t \geq 0\,, \tag{2.20}$$

where ν is a constant parameter. We assume periodic boundary conditions. Before performing the discrete Fourier transform, we need to calculate the discrete Fourier series expansions for f_x and f_{xx}. We notice that (2.17) corresponds to the expansion in continuous space

$$f(x) = \sum_{k=-M}^{M} \hat{c}_k \exp\left(ikx\right), \qquad x \in [0, 2\pi]\,. \tag{2.21}$$

By taking the derivative with respect to x on both sides, we obtain

$$f_x(x) = \sum_{k=-M}^{M} ik\hat{c}_k \exp\left(ikx\right), \qquad x \in [0, 2\pi]\,. \tag{2.22}$$

In discrete form,

$$f_x(x_j) = \sum_{k=-M}^{M} ik\hat{c}_k \exp\left(ikx_j\right), \qquad j = 0, 1, \cdots, N-1\,. \tag{2.23}$$

In a similar way, we obtain

$$f_{xx}(x_j) = \sum_{k=-M}^{M} (-k^2)\hat{c}_k \exp\left(ikx_j\right), \qquad j = 0, 1, \cdots, N-1\,. \tag{2.24}$$

Now, we formally perform the discrete Fourier transform by substituting (2.23), (2.24) and (2.17) (Note that $X = 2\pi$) into (2.20),

$$\begin{aligned} & \frac{\partial}{\partial t}\Big(\sum_{k=-M}^{M} \hat{c}_k \exp(ikx_j)\Big) + \Big(\sum_{k=-M}^{M} \hat{c}_k \exp(ikx_j)\Big)\Big(\sum_{k=-M}^{M} ik\hat{c}_k \exp(ikx_j)\Big) \\ = \;& \nu \sum_{k=-M}^{M} (-k^2)\hat{c}_k \exp(ikx_j)\,, \qquad j = 0, 1, \cdots, N-1\,. \end{aligned} \tag{2.25}$$

From (2.25) we obtain

$$\begin{aligned} \sum_{k=-M}^{M} \frac{\mathrm{d}\hat{c}_k}{\mathrm{d}t} \exp(ikx_j) \;+\; & \sum_{k=-M}^{M} \hat{d}_k \exp(ikx_j) = \sum_{k=-M}^{M} (-\nu k^2)\hat{c}_k \exp(ikx_j)\,, \\ j \;=\; & 0, 1, \cdots, N-1\,, \end{aligned} \tag{2.26}$$

where

$$\hat{d}_k = \sum_{m=-M}^{M} \hat{c}_{k-m}(ik\hat{c}_m)\,, \qquad k = -M, -M+1, \cdots, M\,, \tag{2.27}$$

and where we have used the property that

$$\exp\big(ikx_j\big) = \exp\big(i(k+nN)x_j\big)\,, \qquad \hat{c}_k = \hat{c}_{k+nN}\,, \qquad \text{for } n = 0, \pm 1, \pm 2, \cdots\,. \tag{2.28}$$

Equation (2.26) implies that

$$\frac{\mathrm{d}\hat{c}_k}{\mathrm{d}t} + \hat{d}_k = (-\nu k^2)\hat{c}_k\,, \qquad k = -M, -M+1, \cdots, M\,, \tag{2.29}$$

where $\hat{d}_k$ are defined in (2.27). Now the original PDE (2.20) is transformed into a set of ODEs (2.29) for $\{\hat{c}_k\}$ with respect to t. What's left is to update $\{\hat{c}_k\}$ in time, and then use the inverse discrete Fourier transform (2.18) to recover $\{f_j\}$. If we treat the nonlinear term $\hat{d}_k$ explicitly, then (2.29) can be solved separately for each k. On the other hand, if an implicit form is used for $\hat{d}_k$, then all $\hat{c}_k$s are coupled with each other in (2.29) and usually an iterative method is employed to construct the solutions.

We point out that on deriving equation (2.29) we directly implemented the discrete Fourier transform on the nonlinear term ff_x, which resulted in the formula (2.27) called convolution. It takes $O(N^2)$ operations to calculate all the $\hat{d}_k$s. This is not desirable when N is large. A better way is to use the pseudo-spectral technique [53]. Generally, to perform the discrete Fourier transform of a product of two functions, say u and v, we perform 3 steps:

1. Use the inverse discrete Fourier transform to recover $\{u_j\}$ and $\{v_j\}$, the discrete values of the two functions;

2. Form the products $u_j\, v_j$, $\quad j = 0, 1, \cdots, N-1$;

3. Use the discrete Fourier transform to obtain the Fourier coefficients associated with the products obtained in Step 2.

The above procedure requires $O(N \log_2 N)$ operations, which is much cheaper than the convolution. In practical applications of the pseudo-spectral approach, one may need to de-alias. A commonly used de-aliasing approach is the "$\frac{3}{2}N$ rule" [48].

As a summary, to apply the discrete Fourier transform (often with other numerical methods) to solve a differential equation, the general procedure is: First use the discrete Fourier transform to derive equations for the discrete Fourier coefficients $\{\hat{c}_k\}$. The pseudo-spectral technique is often used to treat the nonlinear terms. Now the problem is transformed into the discrete Fourier space. Then use some other numerical methods, depending on the nature of the equations for $\{\hat{c}_k\}$, to construct the solutions in the discrete Fourier space. Finally, use the inverse discrete Fourier transform to recover the solutions of the original differential equation. If the solutions are sufficiently smooth, say, infinitely differentiable, then the discrete

Fourier transform can achieve spectral accuracy, i.e., the errors decay exponentially with the number of points, N. This is the greatest advantage of the discrete Fourier transform.

2.3 Boundary value problem (BVP) solvers

In this section we are concerned with ODEs and boundary values. A simple example is given by

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y) \tag{2.30}$$

with $y(a)$ and $y(b)$ specified. There have been a number of numerical methods developed to solve such equations. Here we just name a few most common ODE solvers.

1) Euler method:

$$y_{j+1} = y_j + \Delta x \, f(x_j \, , \, y_j) \tag{2.31}$$

which is first-order accurate.

2) Trapezoidal rule:

$$y_{j+1} = y_j + \frac{\Delta x}{2} \, (\, f(x_{j+1} \, , \, y_{j+1}) + f(x_j \, , \, y_j) \,) \tag{2.32}$$

which is second-order accurate.

3) Adams-Bashforth method:

$$y_{j+1} = y_j + \frac{\Delta x}{2} \, (\, 3f(x_j \, , \, y_j) - f(x_{j-1} \, , \, y_{j-1}) \,) \tag{2.33}$$

which is second-order accurate.

4) Simpson's rule:

$$y_{j+1} = y_{j-1} + \frac{\Delta x}{3} \, (\, f(x_{j+1} \, , \, y_{j+1}) + 4f(x_j \, , \, y_j) + f(x_{j-1} \, , \, y_{j-1}) \,) \tag{2.34}$$

which is fourth-order accurate.

We note that among all the methods with second-order accuracy the trapezoidal rule is the unique one that employes points at only two levels, $j+1$ and j. This feature gives great simplicity for implementing the method since we don't need a start-up procedure. Also the trapezoidal rule possesses good stability properties. Therefore, in what follows we discuss this method in detail.

We focus on a special case of (2.30)

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \lambda y + r\,, \qquad a \le x \le b\,, \tag{2.35}$$

where $\lambda \neq 0$ is a real constant and r is a function of x. Equation (2.35) has the importance that many problems in numerical computations can be reduced to differential equations in this form or in its vector counterparts, which we will address soon. Let's apply the trapezoidal rule to (2.35),

$$\frac{y_{j+1} - y_j}{\Delta x} = \frac{\lambda}{2}(y_{j+1} + y_j) + \frac{1}{2}(r_{j+1} + r_j)\,. \tag{2.36}$$

That is,

$$(1 - \frac{\Delta x}{2}\lambda)\, y_{j+1} - (1 + \frac{\Delta x}{2}\lambda)\, y_j = \frac{\Delta x}{2}(r_j + r_{j+1})\,. \tag{2.37}$$

Suppose we partition the domain $[a,\, b]$ into J intervals so that $j = 0,\, 1,\, \cdots,\, J$ with $J\Delta x = b - a$. We have to discuss (2.37) in two different cases.

If $\lambda < 0$, we use

$$y_{j+1} = \frac{1 + \frac{\Delta x}{2}\lambda}{1 - \frac{\Delta x}{2}\lambda}\, y_j + \frac{\frac{\Delta x}{2}}{1 - \frac{\Delta x}{2}\lambda}\,(r_j + r_{j+1})\,. \tag{2.38}$$

We calculate y_{j+1} from $j = 0$ to $j = 1,\, 2, \cdots$, until $j = J - 1$. That means, we do

the calculation with the index j increasing. In this case y_0 has to be given at the left boundary condition. Since $\left|\dfrac{1+\frac{\Delta x}{2}\lambda}{1-\frac{\Delta x}{2}\lambda}\right| < 1$, (2.38) is numerically stable.

On the other hand, if $\lambda > 0$, we use

$$y_j = \frac{1-\frac{\Delta x}{2}\lambda}{1+\frac{\Delta x}{2}\lambda}\, y_{j+1} - \frac{\frac{\Delta x}{2}}{1+\frac{\Delta x}{2}\lambda}\,(r_j + r_{j+1})\ . \tag{2.39}$$

We calculate y_j from $j = J-1$ to $j = J-2,\ J-3, \cdots$, until $j = 0$. That means, we do the calculation with the index j decreasing. In this case y_J has to be given at the right boundary condition. Since $\left|\dfrac{1-\frac{\Delta x}{2}\lambda}{1+\frac{\Delta x}{2}\lambda}\right| < 1$, (2.39) is numerically stable.

Now let's consider a matrix equation

$$\frac{\mathrm{d}}{\mathrm{d}x}\,Y = AY + R\,, \qquad a \le x \le b\,, \tag{2.40}$$

where the unknown $Y = (y_1,\, y_2,\, \cdots,\, y_n)^T$, where $A = (a_{ij})$ is an $n \times n$ constant matrix and where $R = (r_1,\, r_2,\, \cdots,\, r_n)^T$ with each r_i a function of x. Equation (2.40) can be regarded as the extension of (2.35) in vector form and we want to apply the trapezoidal rule to solve (2.40) as well. There are two ways to proceed. First let's consider using the trapezoidal rule directly,

$$(I - \frac{\Delta x}{2}A)\,Y_{j+1} - (I + \frac{\Delta x}{2}A)\,Y_j = \frac{\Delta x}{2}(R_j + R_{j+1})\ . \tag{2.41}$$

Suppose the boundary conditions for (2.40) are given at both the end points, $x = a$ and $x = b$,

$$C\,Y(a) + D\,Y(b) = \widetilde{R_2}\ , \tag{2.42}$$

where $C,\ D$ are constant matrices and $\widetilde{R_2}$ is a vector of length n.

The discretized version of (2.42) is

$$C\,Y_0 + D\,Y_J = \widetilde{R_2}\ . \tag{2.43}$$

Combine (2.41) (for $j = 0,\, 1,\, \cdots,\, J-1$) and (2.43) to obtain a partitioned system

$$\begin{bmatrix} D_1 & D_2 & & & & & \\ & D_1 & D_2 & & & & \\ & & \ddots & \ddots & & & \\ & & & \ddots & \ddots & & \\ & & & & D_1 & D_2 & \\ D & & & & & C & \end{bmatrix} \begin{bmatrix} Y_J \\ Y_{J-1} \\ \vdots \\ \vdots \\ Y_1 \\ Y_0 \end{bmatrix} = \begin{bmatrix} \frac{\Delta x}{2}(R_J + R_{J-1}) \\ \frac{\Delta x}{2}(R_{J-1} + R_{J-2}) \\ \vdots \\ \vdots \\ \frac{\Delta x}{2}(R_1 + R_0) \\ \widetilde{R_2} \end{bmatrix}, \qquad (2.44)$$

where

$$D_1 = I - \frac{\Delta x}{2} A\,, \qquad D_2 = -(I + \frac{\Delta x}{2} A)\,. \qquad (2.45)$$

To find a fast way to solve (2.44), we write it in a compact form

$$\begin{bmatrix} B & S_1 \\ S_2 & C \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \widetilde{R_1} \\ \widetilde{R_2} \end{bmatrix}, \qquad (2.46)$$

where

$$\begin{aligned} V_1 &= [\, Y_J\,,\, Y_{J-1}\,,\, \cdots,\, Y_1\,]^T\,, \quad V_2 = Y_0\,, \\ \widetilde{R_1} &= [\, \frac{\Delta x}{2}(R_J + R_{J-1})\,,\, \frac{\Delta x}{2}(R_{J-1} + R_{J-2})\,,\, \cdots,\, \frac{\Delta x}{2}(R_1 + R_0)\,]^T\,, \end{aligned} \qquad (2.47)$$

and $B\,,\ S_1\,,\ S_2$ are the corresponding matrix blocks. Note that B is in block bi-diagonal form and a linear system with the matrix B can be solved very rapidly by using the standard block bi-diagonal solver (see, *e.g.* [27]).

The solution to (2.46) is given by

$$\begin{aligned} V_2 &= (\, C - S_2\, B^{-1}\, S_1\,)^{-1}\, (\, \widetilde{R_2} - S_2\, B^{-1}\, \widetilde{R_1}\,)\,, \\ V_1 &= B^{-1}\, (\, \widetilde{R_1} - S_1\, V_2\,)\,. \end{aligned} \qquad (2.48)$$

The procedure to obtain (2.48) is as follows.

(1) Solve matrix equations $BX = S_1\,,\ BX = \widetilde{R_1}$ to obtain $B^{-1}S_1\,,\ B^{-1}\widetilde{R_1}$, respectively.

(2) Do the matrix multiplications $S_2\,B^{-1}\,S_1\,,\ S_2\,B^{-1}\,\widetilde{R_1}$.

(3) Solve $(\,C - S_2\,B^{-1}\,S_1\,)\,V_2 = \widetilde{R_2} - S_2\,B^{-1}\,\widetilde{R_1}$ to obtain V_2.

(4) Do the multiplication $S_1\,V_2$.

(5) Solve $BV_1 = \widetilde{R_1} - S_1\,V_2$ to obtain V_1.

The problem in this procedure is that there is no guarantee that the block bidiagonal solver associated with the matrix B is numerically stable. If it is unstable, then the method will fail. An alternative approach applies when the coefficient matrix A in (2.40) is diagonalizable, i.e., A has n linearly independent eigenvectors.

Let $\lambda_1\,,\ \lambda_2\,,\cdots,\lambda_n$ be the n eigenvalues of A and $\beta_1\,,\ \beta_2\,,\cdots,\beta_n$ the corresponding eigenvectors. Let $Q = [\beta_1\,,\ \beta_2\,,\cdots,\beta_n]$. Then

$$Q^{-1}AQ = \widetilde{A} = \mathrm{diag}(\lambda_1\,,\ \lambda_2\,,\cdots,\lambda_n)\,. \tag{2.49}$$

We apply the transformation

$$Y = Q\widetilde{Y}\ , \tag{2.50}$$

where $\widetilde{Y} = [\,\widetilde{y_1}\,,\ \widetilde{y_2}\,,\cdots,\widetilde{y_n}\,]$. Then the original system (2.40) becomes

$$\frac{\mathrm{d}}{\mathrm{dx}}\widetilde{Y} = \widetilde{A}\widetilde{Y} + \widetilde{R}\ , \tag{2.51}$$

where $\widetilde{R} = [\,\widetilde{r_1}\,,\ \widetilde{r_2}\,,\cdots,\widetilde{r_n}\,] = Q^{-1}R$. The system (2.51) is reduced to n scalar ODEs,

$$\frac{\mathrm{d}\widetilde{\mathrm{y}}_\mathrm{j}}{\mathrm{dx}} = \lambda_j\,\widetilde{y_j} + \widetilde{r_j}\,, \qquad j = 1,\ 2,\cdots,n\,. \tag{2.52}$$

For $\lambda_j < 0$, we use a formula similar to (2.38) to solve (2.52). For $\lambda_j > 0$, a formula similar to (2.39) will apply. In this way, we ensure numerical stability. Once $\widetilde{Y}$ is constructed, we can recover the solution for the original system by (2.50).

2.4 Iterative methods for linear systems

2.4.1 General idea

There are two major classes of methods for solving linear systems. The well-known Gaussian elimination, LU factorization, LDL^T factorization, etc., belong to the class of direct methods. These methods are suitable for systems with small matrices. When the coefficient matrices are large and sparse, the direct methods become impractical unless the matrices have special structures. In contrast to the direct methods are the iterative methods which generate a sequence of approximate solutions such that they converge to the exact solution. In this section, we give some general idea of the iterative methods, with emphasis on the GMRES method.

Suppose we want to solve a linear system

$$Ax = b\,, \tag{2.53}$$

where $A = (a_{ij})$ is a non-singular $n \times n$ matrix and where b is a vector of length n.

If we split A

$$A = A_1 + (A - A_1)\,, \tag{2.54}$$

then (2.53) yields

$$A_1\, x = (A_1 - A)x + b\,, \tag{2.55}$$

or

$$x = A_1^{-1}(A_1 - A)x + A_1^{-1}b\,. \tag{2.56}$$

If we denote $A_1^{-1}(A_1 - A)$ by Q and $A_1^{-1}b$ by f, then (2.56) can be written as

$$x = Qx + f\,. \tag{2.57}$$

Given a vector $x^{(0)}$, called the initial guess, we can construct a sequence of approximate solutions $\{x^{(m)}\}$ by

$$x^{(m+1)} = Qx^{(m)} + f\,, \qquad m = 0, 1, 2, \cdots\,. \tag{2.58}$$

Assume the iterates $\{x^{(m)}\}$ converge, $\lim_{m\to\infty} x^{(m)} = a$, then (2.58) implies

$$a = Qa + f\,, \tag{2.59}$$

which yields

$$Aa = b\,. \tag{2.60}$$

Hence a, the limit of $\{x^{(m)}\}$, is the exact solution to (2.53).

The next question is: When does the sequence $\{x^{(m)}\}$ converge? We have the following result [22]:

Let $\rho(Q)$ be the spectral radius (i.e., the absolute value of the biggest eigenvalue) of the matrix Q, then the iterates defined in (2.58) converge for any initial guess $x^{(0)}$ if and only if $\rho(Q) < 1$.

Proof Let x be the exact solution of (2.53), then x satisfies (2.57). Let $e^{(m)} = x - x^{(m)}$, then (2.57) and (2.58) yield

$$e^{(m)} = Qe^{(m-1)} = Q^2e^{(m-2)} = \cdots = Q^me^{(0)}$$

Since $e^{(0)} = x - x^{(0)}$ is arbitrary, $\lim_{m\to\infty} e^{(m)} = 0$ if and only if $\lim_{m\to\infty} Q^m = 0$, which is equivalent to the condition $\rho(Q) < 1$.

We note that, for any matrix norm $\|\cdot\|$, $\|Q\| \geq \rho(Q)$. Hence, if $\|Q\| < 1$, then the iterates in (2.58) converge to the exact solution of (2.53). This is very useful since practically it's often easier to calculate some norm of a matrix than to find its spectral radius.

A common way to split the matrix A is to let

$$A = D - L - U\,, \tag{2.61}$$

where

$$D = \mathrm{diag}(a_{11}\,,\, a_{22}\,, \cdots\,,\, a_{nn}) \tag{2.62}$$

and where

$$L = -\begin{pmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \cdots & \cdots & \cdots & & \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \qquad U = -\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ & 0 & a_{23} & \cdots & a_{2n} \\ & & \cdots & \cdots & \cdots \\ & & & 0 & a_{n-1,n} \\ & & & & 0 \end{pmatrix}. \tag{2.63}$$

Some of the classical iterative methods are listed below.

(1) The Jacobi Iteration:

$$x_i^{(m+1)} = \frac{-1}{a_{ii}}\left(\sum_{j=1}^{i-1} a_{ij}\, x_j^{(m)} + \sum_{j=i+1}^{n} a_{ij}\, x_j^{(m)} - b_i \right), \qquad i = 1,\, 2, \cdots,\, n\,. \tag{2.64}$$

We can write (2.64) in matrix form,

$$x^{(m+1)} = D^{-1}(L+U)x^{(m)} + D^{-1}b\,. \tag{2.65}$$

If the matrix A is diagonally dominant, i.e., $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ for $i = 1, 2, \cdots, n$, then the Jacobi Iteration (2.65) converges.

(2) The Gauss-Siedel Iteration:

$$x_i^{(m+1)} = \frac{-1}{a_{ii}} \left(\sum_{j=1}^{i-1} a_{ij}\, x_j^{(m+1)} + \sum_{j=i+1}^{n} a_{ij}\, x_j^{(m)} - b_i \right), \qquad i = 1, 2, \cdots, n. \tag{2.66}$$

Or in matrix form,

$$x^{(m+1)} = D^{-1}(Lx^{(m+1)} + Ux^{(m)}) + D^{-1}b. \tag{2.67}$$

That is,

$$x^{(m+1)} = (D - L)^{-1}Ux^{m} + (D - L)^{-1}b. \tag{2.68}$$

If the matrix A is diagonally dominant, or symmetric and positive definite, then the Gauss-Siedel Iteration (2.68) converges.

(3) The Successive Over-Relaxation (SOR) Iteration:

$$x_i^{(m+1)} = (1 - \omega)x_i^{(m)} - \frac{\omega}{a_{ii}} \Big(\sum_{j=1}^{i-1} a_{ij}\, x_j^{(m+1)} + \sum_{j=i+1}^{n} a_{ij}\, x_j^{(m)} - b_i \Big), \quad i = 1, 2, \cdots, n, \tag{2.69}$$

where ω is a real number called the relaxation parameter. Equation (2.69) can be expressed in matrix form as

$$x^{(m+1)} = (1 - \omega)x^{(m)} + \omega D^{-1}\big(Lx^{(m+1)} + Ux^{(m)} \big) + \omega D^{-1}b, \tag{2.70}$$

which implies

$$x^{(m+1)} = (D - \omega L)^{-1}\big[(1 - \omega)D + \omega U \big]x^{(m)}. + \omega(D - \omega L)^{-1}b \tag{2.71}$$

The convergence of SOR requires

$$0 < \omega < 2 \, . \tag{2.72}$$

In particular, when $\omega = 1$, SOR reduces to the Gauss-Siedel Iteration.

Another popular iterative method is the Conjugate Gradient method, which works for symmetric positive definite matrices. For details, see, for example, [22][27].

2.4.2 Preconditioning

The technique of preconditioning is frequently used in iterative methods to improve the efficiency and to accelerate the convergence, especially when the coefficient matrix A is ill-conditioned. Let's illustrate the basic idea of preconditioning as follows. For simplicity of notation, from now on we refer to the m-th iterates by the subscript m instead of the superscript. Suppose M is a matrix such that $M^{-1}A \approx I$ in the sense that

$$(I - M^{-1}A)^m \to 0 \qquad \text{when} \quad m \to \infty \, . \tag{2.73}$$

The matrix M is also subject to the constraint that linear systems with M are easy to solve. Such a matrix M is called a preconditioner to the oringinal system. Then we can construct a simple iteration by

$$x_{m+1} = x_m + M^{-1}(b - Ax_m) \, . \tag{2.74}$$

Let x_0 be an initial guess. We have the following procedure to compute the solution to $Ax = b$:

$$\begin{aligned} r_0 &= b - Ax_0\,, \\ \text{solve } My_0 &= r_0 \text{ for } y_0\,, \\ \text{while } (r_m &\neq 0) \\ m &= m+1\,, \\ x_m &= x_{m-1} + y_{m-1}\,, \\ r_m &= b - Ax_m\,, \\ \text{solve } My_m &= r_m \text{ for } y_m\,, \\ \text{end}\,. & \end{aligned} \tag{2.75}$$

We can also generate the initial guess x_0 by using the preconditioner and solve

$$Mx_0 + c = b\,, \tag{2.76}$$

where the vector $c \approx 0$ and is intended to make x_0 a better approximation to x. We may put $c = 0$ for simplicity.

If we define the error of the procedure (2.75) to be $e_m = A^{-1}b - x_m$, then we have

$$\begin{aligned}
e_0 &= A^{-1}b - x_0 \ , \\
e_1 &= A^{-1}b - x_1 = A^{-1}b - (x_0 + y_0) \\
&= A^{-1}b - [x_0 + M^{-1}(b - Ax_0)] \\
&= A^{-1}b - x_0 - M^{-1}A(A^{-1}b - x_0) \\
&= e_0 - M^{-1}Ae_0 \\
&= (I - M^{-1}A)e_0 \ , \\
e_2 &= A^{-1}b - x_2 \\
&= A^{-1}b - [x_1 + M^{-1}(b - Ax_1)] \\
&= e_1 - M^{-1}Ae_1 \\
&= (I - M^{-1}A)e_1 \\
&= (I - M^{-1}A)^2 e_0 \ .
\end{aligned}$$

By induction, we easily obtain

$$e_m = (I - M^{-1}A)^m e_0 \ . \tag{2.77}$$

The convergence of the iteration is established by noting (2.73).

2.4.3 The GMRES method

The Generalized Minimum RESidual (GMRES) method was proposed by Saad and Schultz in 1986 [57] in order to solve large and non-symmetric linear systems. In this section we give a brief discription of the fundamentals of GMRES. The presentation

in what follows is essentially from the book of Golub and Van Loan [27] in Ch.5, Ch.9 and Ch.10 and from the report by Fraysse *et al.* [24].

The Arnoldi process

From linear algebra we know that if an $n \times n$ matrix A is symmetric, then we can construct an orthogonal matrix Q such that $Q^T AQ$ is tridiagonal. When A is not symmetric, the orthogonal tridiagonalization $Q^T AQ$ generally does not exist. An alternative way to proceed is via the Arnoldi process, which genearates an orthogonal matrix Q such that $Q^T AQ = H$ is in the upper Hessenberg form. A matrix $H = (h_{ij})$ is upper Hessenberg if $h_{ij} = 0\,,\ i > j + 1\,.$

Let $Q = [q_1,\ q_2, \cdots,\ q_n]\,.$ By comparing columns in $AQ = QH$, we obtain

$$Aq_m = \sum_{i=1}^{m+1} h_{im}\, q_i\,, \quad 1 \le m \le n-1\ . \tag{2.78}$$

If we separate the last term in the summation, we obtain

$$h_{m+1,\,m}\, q_{m+1} = Aq_m - \sum_{i=1}^{m} h_{im}\, q_i \stackrel{\Delta}{=} r_m\ , \tag{2.79}$$

where $h_{im} = q_i^T Aq_m$ for $i = 1, \cdots, m$. It follows that if $r_m \neq 0$, then q_{m+1} is specified by

$$q_{m+1} = r_m \,/\, h_{m+1,\,m}\ , \tag{2.80}$$

where $h_{m+1,\,m} = \|r_m\|_2$ since the 2-norm of each column vector of an orthogonal

matrix is equal to 1. These equations define the Arnoldi process which is described as follows:

$$\begin{aligned}
r_0 &= q_1\,, \\
h_{10} &= 1\,, \\
m &= 0\,, \\
\text{while } (h_{m+1,\,m} &\neq 0) \\
q_{m+1} &= r_m \,/\, h_{m+1,\,m}\,, \\
m &= m+1\,, \\
r_m &= A q_m\,, \\
\text{for } i &= 1 : m \\
h_{im} &= q_i^T r_m\,, \\
r_m &= r_m - h_{im}\, q_i\,, \\
&\text{end}\,, \\
h_{m+1,\,m} &= \|r_m\|_2\,, \\
&\text{end}\,.
\end{aligned} \tag{2.81}$$

We assume that q_1 is a given unit 2-norm starting vector. The q_m are called the Arnoldi vectors and they define an orthonormal basis for the Krylov subspace

$$\mathcal{K}(A,\, q_1,\, m) = \text{span}\{q_1,\, \cdots,\, q_m\} = \text{span}\{q_1,\, Aq_1,\, \cdots,\, A^{m-1}q_1\}\,. \tag{2.82}$$

The situation after m steps is summarized by the Arnoldi factorization (note that

q_{m+1} in (2.79) is not calculated until the $(m+1)$-th step)

$$AQ_m = Q_m H_m + r_m e_m^T \ , \tag{2.83}$$

where $Q_m = [q_1, \cdots, q_m]$, $e_m = I_m(:, m)$ is the m-th canonical vector and

$$H_m = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1m} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2m} \\ 0 & h_{32} & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & h_{m,\,m-1} & h_{mm} \end{bmatrix} .$$

If $r_m = 0$, then the columns of Q_m define an invariant subspace, which means the Krylov subspace $\mathcal{K}(A, q_1, m)$ is invariant under A.

Givens rotations and least squares

An m-by-n matrix A can be factorized by

$$A = QR \ , \tag{2.84}$$

where $Q \in \mathbf{R}^{m \times m}$ is orthogonal and $R \in \mathbf{R}^{m \times n}$ is upper triangular. This is called the QR factorization.

The method of Givens rotations is a common way to compute the QR factorization of a matrix. Givens rotations are matrices of the form

$$G = G(i, m, \theta) = \begin{bmatrix} I & & & & \\ & c & & s & \\ & & I & & \\ & -s & & c & \\ & & & & I \end{bmatrix} , \tag{2.85}$$

where $G_{ii} = c = \cos\theta$, $G_{im} = s = \sin\theta$ for some θ and the Is are identity matrices of the appropriate dimensions. Such matrices are clearly orthogonal. Premultiplication by $G(i, m, \theta)^T$ amounts to a counterclockwise rotation of θ radians in the $(i,\ m)$ coordinate plane. If $x \in \mathbf{R}^n$ and $y = G(i, m, \theta)^T x$, then the components of y read

$$y_j = \begin{cases} cx_i - sx_m\,, & j = i\,, \\ sx_i + cx_m\,, & j = m\,, \\ x_j\,, & j \neq i,\, m\,. \end{cases}$$

If we set

$$c = \frac{x_i}{\sqrt{x_i^2 + x_m^2}} , \qquad s = \frac{-x_m}{\sqrt{x_i^2 + x_m^2}} , \tag{2.86}$$

then $y_m = 0$. Thus, it is a simple matter to zero a specific entry in a vector by using the Givens rotation.

Now, we apply the Givens rotations to an m-by-n matrix A . We can first zero all the lower diagonal entries in the first column, A_{i1} , in the order of $i = m,\ m-1,\ \cdots,\ 2$. Then we zero all the lower diagonal entries in the second column, A_{i2} , in the order of $i = m,\ m-1,\ \cdots,\ 3$. If we continue this process, column by column, we will finally

obtain an upper triangular matrix R. If G_j denotes the j-th Givens rotation in the reduction, then we have

$$Q^T A = R \ , \tag{2.87}$$

where $Q = G_1 G_2 \cdots , G_J$ and J is the total number of rotations. Equation (2.87) gives the QR factorization of the matrix A.

The QR factorization is a powerful method to find the least squares solution of overdetermined systems of equations, i.e., the minimization of $\|Ax - b\|_2$ where $A \in \mathbf{R}^{m \times n}$ with $m \geq n$ and $b \in \mathbf{R}^m$. Here we assume that rank(A)=n. Suppose that an orthogonal matrix $Q \in \mathbf{R}^{m \times m}$ has been computed such that

$$Q^T A = R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \ , \quad \text{where} \ \ R_1 \in \mathbf{R}^{n \times n} \ , \tag{2.88}$$

is upper triangular. Let

$$Q^T b = \begin{bmatrix} c \\ d \end{bmatrix} \ , \quad \text{where} \ \ c \in \mathbf{R}^n \ \ \text{and} \ \ d \in \mathbf{R}^{m-n} \ . \tag{2.89}$$

Since the 2-norm is preserved under an orthogonal transformation, we have

$$\|Ax - b\|_2^2 = \|Q^T Ax - Q^T b\|_2^2 = \|R_1 x - c\|_2^2 + \|d\|_2^2 \tag{2.90}$$

for any $x \in \mathbf{R}^n$. Clearly, if rank(A)=rank(R_1)= n, then the least squares solution x is determined by the upper triangular system

$$R_1 x = c \, . \tag{2.91}$$

Hence we conclude that the full rank least squares problem can be readily solved once we have computed the QR factorization of A.

The GMRES algorithm

Let A be a non-singular $n \times n$ real matrix, and b be a vector of length n. We want to solve the linear system

$$Ax = b \ . \tag{2.92}$$

Let x_0 be an initial guess for this linear system and $r_0 = b - Ax_0$ be its corresponding residual.

The GMRES algorithm builds an approximation to the solution of (2.92) in the form

$$x_m = x_0 + Q_m y \ , \tag{2.93}$$

where Q_m is an orthonormal basis for the Krylov subspace

$$\mathcal{K}(A,\, r_0,\, m) = \text{span}\{r_0,\, Ar_0,\, \cdots,\, A^{m-1} r_0\} \ , \tag{2.94}$$

and where the vector y is determined so that the 2-norm of the residual $r_m = b - Ax_m$ is minimal over $\mathcal{K}(A,\, r_0,\, m)$.

The basis Q_m for the Krylov subspace $\mathcal{K}(A,\, r_0,\, m)$ is constructed via the Arnoldi process discussed before. After m steps of the Arnoldi iteration, we get the relationship (2.83), which can be rewritten as

$$AQ_m = Q_{m+1} \tilde{H}_m \ , \tag{2.95}$$

where

$$Q_{m+1} = [Q_m,\, q_{m+1}], \qquad \tilde{H}_m = \begin{bmatrix} & H_m & \\ 0 \cdots 0 & & h_{m+1,\,m} \end{bmatrix} \in \mathbf{R}^{(m+1)\times m} \ .$$

If $q_1 = r_0/\beta_0$ where $\beta_0 = \|r_0\|_2$, then it follows that

$$\begin{aligned} r_m &= b - Ax_m = b - A(x_0 + Q_m y) \\ &= r_0 - AQ_m y = r_0 - Q_{m+1}\tilde{H}_m y \\ &= \beta_0 q_1 - Q_{m+1}\tilde{H}_m y \\ &= Q_{m+1}(\beta_0\, e_1 - \tilde{H}_m y)\ . \end{aligned} \tag{2.96}$$

Since Q_{m+1} is an orthonormal matrix, the residual norm $\|r_m\|_2 = \|\beta_0\, e_1 - \tilde{H}_m y\|_2$ is minimized when y solves the linear least-squares problem

$$\min_{y \in \mathbf{R}^m} \|\beta_0\, e_1 - \tilde{H}_m y\|_2\ . \tag{2.97}$$

Equation (2.97) can be efficiently solved by using the Givens rotations discussed in the previous subsection. We denote the solution of (2.97) by y_m. Then $x_m = x_0 + Q_m y_m$ gives an approximate solution of (2.92) for which the residual is minimized over

$\mathcal{K}(A,\, r_0,\, m)$. Now we are ready to write out the basic GMRES algorithm as follows:

$$
\begin{aligned}
r_0 &= b - Ax_0\,, \\
h_{10} &= \|r_0\|_2\,, \\
m &= 0\,, \\
\text{while } (h_{m+1,\,m} &> 0) \\
q_{m+1} &= r_m \,/\, h_{m+1,\,m}\,, \\
m &= m + 1\,, \\
r_m &= Aq_m\,, \\
\text{for } i &= 1 : m \\
h_{im} &= q_i^T\, r_m\,, \\
r_m &= r_m - h_{im}\, q_i\,, \\
&\text{end}\,, \\
h_{m+1,\,m} &= \|r_m\|_2\,, \\
x_m &= x_0 + Q_m y_m \ \text{ where } \|h_{10}\, e_1 - \tilde{H}_m y_m\|_2 = \min\,, \\
&\text{end}\,, \\
x = x_m\,. &
\end{aligned}
\tag{2.98}
$$

In practice, to make GMRES effective, preconditioning is almost always required. The preconditioning process in GMRES is similar to that discussed before except the iteration form is different from that in (2.74). We have found in our numerical experiments that preconditioning is a key to improve the efficiency of GMRES. More

than often a good preconditioner can use just 1/20 or less iterations to achieve the same accuracy, compared to the case without a preconditioner.

We refer to the above preconditioning process as left preconditioning. Sometimes people may also consider right preconditioning together with left preconditioning. That means, we construct two matrices $M_1,\ M_2$ such that $M_1^{-1}AM_2^{-1}$ is close to the identy matrix I and that linear systems with M_1 are easy to solve. Then we solve the linear system

$$M_1^{-1}AM_2^{-1}y = M_1^{-1}b \tag{2.99}$$

with $x = M_2^{-1}y$. However, in all our numerical tests we have found this is not necessary since a left preconditioner seems good enough for the GMRES. Hence we won't discuss right preconditioning in any further detail.

Another issue related to the GMRES algorithm (2.74) is that the storage of the orthogonal basis Q_m might be demanding, especially when the convergence is slow. The restarted GMRES method is developed to cope with this memory drawback. Given a fixed number j, the restarted GMRES method computes a sequence of approximate solutions x_m until the convergence is achieved or $m = j$. If $m = j$ and the solution is not found yet, then a new starting vector is chosen on which GMRES is applied again. Often GMRES is restarted from the last computed approximation, i.e., $x_0 = x_j$. The process is repeated until the convergence is achieved.

Finally, though the above discussion of GMRES is concerned with real linear system, the ideas can be naturally extended to complex case. Only a few minor changes are needed for this extension: the matrix Q_m^T has to be replaced by Q_m^*

which is the conjugate transposition of Q and the least-square problem (2.97) has to be solved in the complex space $\mathcal{C}^m$.

A GMRES package

There are several software packages that implement GMRES. We will consider the one written by Fraysse *et al.* [24] for simplicity and portability. This package contains GMRES routines for both real and complex, single and double precision calculations. One important feature of this package is that the GMRES solvers are implemented by the reverse communication mechanism for the matrix-vector multiplication, the dot product and the preconditioning computations. That means, the package leaves these jobs to the user. They need to be supplied in a series of subroutines. This ensures the portability and the flexibility of the package since only the user knows what matrix is being treated, how the preconditioner should be chosen and how the data should be organized.

Another advantage of this data structure is that the GMRES routines don't care about the explicit structure of the matrix. Actually the matrix is not stored at all in the routines. We will exploit this nature of the software package in the design of our numerical method.

CHAPTER 3

NUMERICAL METHODS

3.1 Basic formulation

We now go to the main point of this thesis – to numerically study the two-dimensional interfacial flows between two immiscible and incompressible viscous fluids. Let's denote the spacial coordinates by (x, z), the temporal coordinate by t, the velocity components by (u, w), the pressure by p, the density by ρ, the dynamic viscosity by μ and the gravitational acceleration by g. The equations of motion, in each of the two fluids, are given by the Navier-Stokes equations

$$\rho u_t + \rho u u_x + \rho w u_z = -P_x + \mu(u_{xx} + u_{zz}), \tag{3.1}$$

$$\rho w_t + \rho u w_x + \rho w w_z = -P_z + \mu(w_{xx} + w_{zz}), \tag{3.2}$$

where P is the hydrodynamic pressure which includes the gravity term, $P = p + \rho g z$. The incompressibility condition is

$$u_x + w_z = 0. \tag{3.3}$$

The equations (3.1)-(3.3) hold in both the upper fluid and the lower fluid. Their solutions are connected through the interfacial conditions. Let's represent the interface

in the form

$$(x, z) = (x, h(x,t)) . \tag{3.4}$$

h is determined by the kinematic condition

$$h_t + u^{(I)} h_x = w^{(I)} , \tag{3.5}$$

where $u^{(I)}, w^{(I)}$ are the velocity components at the interface.

The continuity of velocity at the interface gives

$$u^{(1)} = u^{(2)} = u^{(I)} , \quad w^{(1)} = w^{(2)} = w^{(I)} , \tag{3.6}$$

where the superscripts $(1), (2)$ distinguish the upper and the lower domains. Moreover, we have two stress conditions, or dynamical interfacial conditions,

$$\begin{aligned} & (h_x^2 - 1) \, [\, \mu^{(1)} (u_z^{(1)} + w_x^{(1)}) - \mu^{(2)} (u_z^{(2)} + w_x^{(2)}) \,] \\ + \; & 2 h_x \, [\, \mu^{(1)} (u_x^{(1)} - w_z^{(1)}) - \mu^{(2)} (u_x^{(2)} - w_z^{(2)}) \,] = 0 , \end{aligned} \tag{3.7}$$

$$\begin{aligned} (P^{(1)} - P^{(2)}) \; - \; & g h (\rho^{(1)} - \rho^{(2)}) + h_x \, [\, \mu^{(1)} (u_z^{(1)} + w_x^{(1)}) - \mu^{(2)} (u_z^{(2)} + w_x^{(2)}) \,] \\ - \; & 2 \, [\, \mu^{(1)} w_z^{(1)} - \mu^{(2)} w_z^{(2)} \,] - 2 T \kappa \, = \, 0 , \end{aligned} \tag{3.8}$$

where T is the surface tension of water and where κ is the mean curvature of the interface,

$$2\kappa = \frac{h_{xx}}{(1 + h_x^2)^{3/2}} \, . \tag{3.9}$$

We are seeking those solutions that are 2π-periodic in x . Hence we don't need additional boundary conditions in the horizontal direction. However we do need the boundary conditions at the two ends in the vertical direction to complete the system and we are only interested in those solutions which are exponentially decaying with respect to $|z|$.

3.2 The mapped equations

The evolving interface $h(x,t)$ between the two fluids makes the design of numerical methods difficult. To overcome this difficulty we map the deformed geometry into a rectangular shape in new, logical coordinates at the cost of changing the details of the governing equations and the interfacial conditions. Our numerical methods are then constructed on these mapped equations.

Let's introduce the new coordinates, (X, Z, τ), through the mapping

$$x = X\,, \tag{3.10}$$

$$z = F(X, Z, \tau)\,, \tag{3.11}$$

$$t = \tau\,, \tag{3.12}$$

where

$$F(X,Z,\tau) \triangleq \begin{cases} Z + h(X,\tau)\,\exp(-\alpha Z), & Z \geq 0\,, \\ Z + h(X,\tau)\,\exp(\alpha Z), & Z \leq 0\,, \end{cases} \tag{3.13}$$

where $\alpha \geq 0$ is a constant. Clearly, when $Z = 0$,

$$z = h(x,t) \tag{3.14}$$

marks the location of the interface. When far from the interface, Z is relaxing exponentially to z if $\alpha \neq 0$.

If we define

$$G_0 = \frac{F_\tau}{F_Z}\,, \qquad G_1 = \frac{F_X}{F_Z}\,, \qquad G_3 = \frac{1}{F_Z}\,, \tag{3.15}$$

then

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - G_0 \frac{\partial}{\partial Z} , \tag{3.16}$$

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial X} - G_1 \frac{\partial}{\partial Z} , \tag{3.17}$$

$$\frac{\partial}{\partial z} = G_3 \frac{\partial}{\partial Z} , \tag{3.18}$$

$$\frac{\partial^2}{\partial x^2} = \frac{\partial^2}{\partial X^2} + (G_1)^2 \frac{\partial^2}{\partial Z^2} - 2G_1 \frac{\partial^2}{\partial X \partial Z} + (G_1(G_1)_Z - (G_1)_X) \frac{\partial}{\partial Z} , \tag{3.19}$$

$$\frac{\partial^2}{\partial z^2} = (G_3)^2 \frac{\partial^2}{\partial Z^2} + G_3 (G_3)_Z \frac{\partial}{\partial Z} . \tag{3.20}$$

Let's further define

$$g_2 = (G_1)^2 + (G_3)^2 , \quad g_3 = -2G_1 , \quad g_4 = G_1 \frac{\partial G_1}{\partial Z} + G_3 \frac{\partial G_3}{\partial Z} - \frac{\partial G_1}{\partial X} . \tag{3.21}$$

Then we can write the Laplacian in new variables as

$$\begin{aligned} \mathcal{L} &\triangleq \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} \\ &= \frac{\partial^2}{\partial X^2} + g_2 \frac{\partial^2}{\partial Z^2} + g_3 \frac{\partial^2}{\partial X \partial Z} + g_4 \frac{\partial}{\partial Z} . \end{aligned} \tag{3.22}$$

We remark that in the upper domain ($Z \geq 0$) and the lower domain ($Z \leq 0$), each of G_i ($i = 0, 1, 3$), g_i ($i = 2, 3, 4$) generally has different expressions, according to (3.13). In particular, if we set $\alpha = 0$ in (3.13), we get, in both domains,

$$G_0 = h_\tau , \quad G_1 = h_X , \quad G_3 = 1 , \tag{3.23}$$

$$g_2 = 1 + (h_X)^2 , \quad g_3 = -2h_X , \quad g_4 = -h_{XX} . \tag{3.24}$$

Now we substitute the transformation rules (3.16)-(3.22) for the derivatives directly into the basic equations (3.1)-(3.8). The equations of motion become:

$$u_\tau - G_0 u_Z + u(u_X - G_1 u_Z) + wG_3 u_Z = -\frac{1}{\rho}P_X + \frac{1}{\rho}G_1 P_Z + \nu\mathcal{L}\{u\}\,, \quad (3.25)$$

$$w_\tau - G_0 w_Z + u(w_X - G_1 w_Z) + wG_3 w_Z = -\frac{1}{\rho}G_3 P_Z + \nu\mathcal{L}\{w\}\,, \quad (3.26)$$

where $\nu = \frac{\mu}{\rho}$ is called the kinematic viscosity, and the incompressibility condition is

$$u_X - G_1 u_Z + G_3 w_Z = 0\,. \quad (3.27)$$

At the interface $Z = 0$, we have continuity of the velocity

$$u^{(1)} = u^{(2)} = u^{(I)}\,, \quad w^{(1)} = w^{(2)} = w^{(I)}\,, \quad (3.28)$$

and the kinematic condition

$$h_\tau + u^{(I)} h_X = w^{(I)}\,. \quad (3.29)$$

The stress conditions (or dynamic interfacial condtions) are now:

$$\begin{aligned}\mu^{(1)}(G_3^{(1)}u_Z^{(1)} + w_X^{(1)}) &- \mu^{(2)}(G_3^{(2)}u_Z^{(2)} + w_X^{(2)}) + \big(\frac{4h_X}{h_X^2 - 1} + \frac{G_1^{(1)}}{G_3^{(1)}}\big)\mu^{(1)}(u_X^{(1)} - G_1^{(1)}u_Z^{(1)}) \\ &- \big(\frac{4h_X}{h_X^2 - 1} + \frac{G_1^{(2)}}{G_3^{(2)}}\big)\mu^{(2)}(u_X^{(2)} - G_1^{(2)}u_Z^{(2)}) = 0\,, \quad (3.30)\end{aligned}$$

$$\begin{aligned}(P^{(1)} - P^{(2)}) &+ \big(2 - \frac{4h_X^2}{h_X^2 - 1}\big)\big[\mu^{(1)}(u_X^{(1)} - G_1^{(1)}u_Z^{(1)}) - \mu^{(2)}(u_X^{(2)} - G_1^{(2)}u_Z^{(2)})\big] \\ &= gh(\rho^{(1)} - \rho^{(2)}) + 2T\kappa\,. \quad (3.31)\end{aligned}$$

We note that in order to obtain (3.30) and (3.31), we have eliminated $w_z^{(1)}$ and $w_z^{(2)}$ in equations (3.7) and (3.8) by using the incompressiblity condition (3.3). It will be

clear soon that such eliminations are necessary to be consistent with our numerical method for the Navier-Stokes equations.

Before we start a detailed description of the numerical methods, it is best to describe first the overall strategy. The equations are written in the form of linear terms and nonlinear terms separately, and they have the appearance of (2.3). Then the methods described there may be used to advance the solution in time. In particular, implicit methods will be used for reasons of numerical stability and iterative method will be used to construct the solution. Two specific choices will be made, the Crank-Nicolson method (2.7) and the second-order BDF method (2.4). These methods are fully implicit and so require the solution of a nonlinear system of equations for the unknowns at the new time level. The linear terms provide a simple iterative procedure. For example, the iteration

$$\frac{u^{(n,m)} - u^n}{\Delta t} + \frac{1}{2}(g^{(n,m)} + g^n) = \frac{1}{2}(f^{(n,m-1)} + f^n) \tag{3.32}$$

may be applied to the Crank-Nicolson method (2.7) and when the iterations have reached a satisfactory level of convergence, $u^{n+1} = u^{(n,m)}$. The advantage of such an iteration is that only a linear system must be solved, but the disadvantage may be the need to reduce Δt to ensure convergence below values that are sufficient for numerical stability. As to the spacial discretization, the Fourier transform is performed along the X-direction which possesses periodicity to achieve spectral accuracy in X. For each iteration at each time step, a linear system of 1st-order ODEs with respect to Z is solved.

As a start, we describe the separation of the equations into linear and nonlinear

parts. Let's introduce a new variable $q = u_Z$. Then we extract the linear parts of these equations and put them to the left-hand sides. All the nonlinear terms and the mapping-associated terms will be collected to the right-hand sides.

We have, first of all,

$$u_Z - q = 0 \ . \tag{3.33}$$

In all the following equations, we replace u_Z by q. Secondly, from the momemtum equation (3.25) we have

$$\begin{aligned} u_\tau + \frac{1}{\rho} P_X - \nu(u_{XX} + q_Z) = R_u \triangleq G_0 q + \frac{1}{\rho} G_1 P_Z \\ -\left[u(u_X - \ G_1 q) + w G_3 q\right] + \nu\left[(g_2 - 1) q_Z + g_3 q_X + g_4 q\right] \ . \end{aligned} \tag{3.34}$$

Thirdly, the incompressibility condition (3.27) gives

$$u_X + w_Z = R_c \triangleq G_1 q + (1 - G_3) w_Z \ . \tag{3.35}$$

Finally, from the momentum equation (3.26) we have

$$\begin{aligned} w_\tau + \frac{1}{\rho} P_Z - \nu(w_{XX} + w_{ZZ}) \ &= \ G_0 w_Z + \frac{1}{\rho}(1 - G_3) P_Z \\ -[u(w_X - G_1 w_Z) + w G_3 w_Z] \ &+ \ \nu\left[g_2 w_{ZZ} + g_3 w_{XZ} + g_4 w_Z - w_{ZZ}\right] . \end{aligned} \tag{3.36}$$

We don't want the w_{ZZ} term on the left-hand side since it's a 2nd-order derivative with respect to Z. We note that, in the linear case the incompressibility condition reads

$$u_X + w_Z = 0 \ , \tag{3.37}$$

which implies

$$w_{ZZ} + q_X = 0 \ . \tag{3.38}$$

Hence, we hope to replace w_{ZZ} by $-q_X$ on the linear part, i.e., the left-hand side, of (3.36). This is done by adding $\nu(w_{ZZ}+q_X)$ to both sides of (3.36) and maintaining the equality,

$$w_\tau + \frac{1}{\rho}P_Z - \nu(w_{XX}-q_X) = R_w \triangleq G_0 w_Z + \frac{1}{\rho}(1-G_3)P_Z$$
$$-\big[u(w_X - G_1 w_Z) + wG_3 w_Z\big] + \nu\big[g_2 w_{ZZ} + g_3 w_{XZ} + g_4 w_Z + q_X\big]\,. \tag{3.39}$$

We will use (3.39) instead of (3.36) as one of the governing equations.

Similarly, the two stress conditions (3.30) and (3.31) may be expressed as

$$\mu^{(1)}(q^{(1)}+w_X^{(1)}) - \mu^{(2)}(q^{(2)}+w_X^{(2)}) = S_1 \triangleq \mu^{(1)}(1-G_3^{(1)})q^{(1)} - \mu^{(2)}(1-G_3^{(2)})q^{(2)}$$
$$-\Big(\frac{4h_X}{h_X^2-1}+\frac{G_1^{(1)}}{G_3^{(1)}}\Big)\mu^{(1)}(u_X^{(1)}-G_1^{(1)}q^{(1)})$$
$$+\Big(\frac{4h_X}{h_X^2-1}+\frac{G_1^{(2)}}{G_3^{(2)}}\Big)\mu^{(2)}(u_X^{(2)}-G_1^{(2)}q^{(2)})\,, \tag{3.40}$$
$$P^{(1)} - P^{(2)} + 2(\mu^{(1)}u_X^{(1)} - \mu^{(2)}u_X^{(2)}) = S_2 \triangleq gh(\rho^{(1)}-\rho^{(2)}) + 2T\kappa$$
$$+2(\mu^{(1)}G_1^{(1)}q^{(1)} - \mu^{(2)}G_1^{(2)}q^{(2)})$$
$$+\frac{4h_X^2}{h_X^2-1}\big[\mu^{(1)}(u_X^{(1)}-G_1^{(1)}q^{(1)}) - \mu^{(2)}(u_X^{(2)}-G_1^{(2)}q^{(2)})\big]\,, \tag{3.41}$$

while the other two interfacial conditions (3.28) and the kinematic condition (3.29) remain unchanged.

3.3 Time marching

Suppose we know the numerical solution at the time step n, $\{h^n, u^n, q^n, w^n, P^n\}$, and we want to advance the solution to the next time step $n+1$, $\{h^{n+1}, u^{n+1}, q^{n+1}, w^{n+1}, P^{n+1}\}$.

Since the equations are nonlinear, an iterative method is used to construct the solution. The iteration is based on the linear part of the system as described in (3.32) for the Crank-Nicolson method.

3.3.1 The Crank-Nicolson method

The Crank-Nicolson method (2.7) applied to (3.33), (3.34), (3.35) and (3.39) yields

$$\begin{aligned} u_Z^{n+1} - q^{n+1} &= 0 \,, && (3.42)\\ \frac{u^{n+1}-u^n}{\Delta\tau} + \frac{1}{2\rho}(P_X^{n+1}+P_X^n) &- \frac{\nu}{2}(u_{XX}^{n+1}+q_Z^{n+1}+u_{XX}^n+q_Z^n) \\ &= \frac{1}{2}R_u^{n+1}+\frac{1}{2}R_u^n \,, && (3.43)\\ w_Z^{n+1}+u_X^{n+1} &= R_c^{n+1} \,, && (3.44)\\ \frac{w^{n+1}-w^n}{\Delta\tau} + \frac{1}{2\rho}(P_Z^{n+1}+P_Z^n) &- \frac{\nu}{2}(w_{XX}^{n+1}-q_X^{n+1}+w_{XX}^n-q_X^n) \\ &= \frac{1}{2}R_w^{n+1}+\frac{1}{2}R_w^n \,, && (3.45) \end{aligned}$$

where R_u , R_c , R_w contain all the nonlinear terms. In addition, the evolution of the interface is approximated by

$$\frac{h^{n+1}-h^n}{\Delta\tau} = \frac{1}{2}(w^{(I)}-u^{(I)}h_X)^{n+1} + \frac{1}{2}(w^{(I)}-u^{(I)}h_X)^n \,, \qquad (3.46)$$

while the interfacial conditions are approximated by

$$\begin{aligned} (\, u^{(1)} - u^{(2)} \,)^{n+1} &= 0 \,, && (3.47)\\ (\, \mu^{(1)}(q^{(1)}+w_X^{(1)}) - \mu^{(2)}(q^{(2)}+w_X^{(2)}) \,)^{n+1} &= S_1^{n+1} \,, && (3.48)\\ (\, w^{(1)} - w^{(2)} \,)^{n+1} &= 0 \,, && (3.49)\\ (\, P^{(1)} - P^{(2)} + 2(\mu^{(1)}u_X^{(1)} - \mu^{(2)}u_X^{(2)}) \,)^{n+1} &= S_2^{n+1} \,, && (3.50) \end{aligned}$$

where S_1, S_2 have all the nonlinear terms.

Unfortunately, the equations for the updated solution are nonlinear and challenging to solve in general. A simple iterative method, as described in (3.32), is obtained by evaluating all nonlinear terms with a previous guess. Specifically, let $\{h^{(n,m-1)}, u^{(n,m-1)}, q^{(n,m-1)}, w^{(n,m-1)}, P^{(n,m-1)}\}$ be the solution at the $(m-1)$ th iteration $(m \geq 1)$, where

$$\{h^{(n,0)}, u^{(n,0)}, q^{(n,0)}, w^{(n,0)}, P^{(n,0)}\} \stackrel{\Delta}{=} \{h^n, u^n, q^n, w^n, P^n\} .$$

The next iterate is obtained as follows.

First we update the interface h by

$$\frac{h^{(n,m)} - h^n}{\Delta\tau} = \frac{1}{2}(w^{(I)} - u^{(I)}h_X)^{(n,m-1)} + \frac{1}{2}(w^{(I)} - u^{(I)}h_X)^n . \tag{3.51}$$

Once $h^{(n,m)}$ is known, the mappings (3.10)-(3.12) are evaluated and the mapping-associated coefficients $G_i^{(n,m)}$ $(i = 0, 1, 3)$, $g_i^{(n,m)}$ $(i = 2, 3, 4)$ are readily calculated. Then we compute $\{u^{(n,m)}, q^{(n,m)}, w^{(n,m)}, P^{(n,m)}\}$ by

$$\begin{aligned}
u_Z^{(n,m)} - q^{(n,m)} &= 0 , &(3.52)\\
\frac{u^{(n,m)} - u^n}{\Delta\tau} + \frac{1}{2\rho}(P_X^{(n,m)} + P_X^n) &- \frac{\nu}{2}(u_{XX}^{(n,m)} + q_Z^{(n,m)} + u_{XX}^n + q_Z^n) \\
&= \frac{1}{2}R_u^{(n,m-1)} + \frac{1}{2}R_u^n , &(3.53)\\
w_Z^{(n,m)} + u_X^{(n,m)} &= R_c^{(n,m-1)} , &(3.54)\\
\frac{w^{(n,m)} - w^n}{\Delta\tau} + \frac{1}{2\rho}(P_Z^{(n,m)} + P_Z^n) &- \frac{\nu}{2}(w_{XX}^{(n,m)} - q_X^{(n,m)} + w_{XX}^n - q_X^n) \\
&= \frac{1}{2}R_w^{(n,m-1)} + \frac{1}{2}R_w^n , &(3.55)
\end{aligned}$$

with the corresponding interfacial conditions

$$(\, u^{(1)} - u^{(2)} \,)^{(n,m)} = 0 \,, \tag{3.56}$$

$$(\, \mu^{(1)}(q^{(1)} + w_X^{(1)}) - \mu^{(2)}(q^{(2)} + w_X^{(2)}) \,)^{(n,m)} = S_1^{(n,m-1)} \,, \tag{3.57}$$

$$(\, w^{(1)} - w^{(2)} \,)^{(n,m)} = 0 \,, \tag{3.58}$$

$$(\, P^{(1)} - P^{(2)} + 2(\mu^{(1)} u_X^{(1)} - \mu^{(2)} u_X^{(2)}) \,)^{(n,m)} = S_2^{(n,m-1)} \,. \tag{3.59}$$

We set the stopping criterion of the iterations to be

$$\begin{aligned} &\frac{\|h^{(n,m)} - h^{(n,m-1)}\|_2}{\|h^{(n,m-1)}\|_2} + \frac{\|u^{(n,m)} - u^{(n,m-1)}\|_2}{\|u^{(n,m-1)}\|_2} + \frac{\|q^{(n,m)} - q^{(n,m-1)}\|_2}{\|q^{(n,m-1)}\|_2} \\ &+ \frac{\|w^{(n,m)} - w^{(n,m-1)}\|_2}{\|w^{(n,m-1)}\|_2} + \frac{\|P^{(n,m)} - P^{(n,m-1)}\|_2}{\|P^{(n,m-1)}\|_2} < E \,, \end{aligned} \tag{3.60}$$

where E is some tolerance and where the L_2-norm $\|\cdot\|_2$ is taken at all the grid points. Once (3.60) is satisfied, we set

$$\{h^{n+1}, u^{n+1}, q^{n+1}, w^{n+1}, P^{n+1}\} = \{h^{(n,m)}, u^{(n,m)}, q^{(n,m)}, w^{(n,m)}, P^{(n,m)}\}$$

and the advancement of solution to the time step $n+1$ is complete. Clearly, when convergence is achieved, we have effectively applied the Crank-Nicolson scheme to (3.51)-(3.55) and treated all the interfacial condtions in a fully implicit manner.

3.3.2 The backward differentiation formula (BDF)

An alternative way to do time marching is based on the second-order BDF method (2.4). The application of the BDF method to (3.33), (3.34), (3.35) and (3.39) yields

$$u_Z^{n+1} - q^{n+1} = 0 \,, \tag{3.61}$$

$$\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta\tau} + \frac{1}{\rho}P_X^{n+1} - \nu(u_{XX}^{n+1} + q_Z^{n+1}) = R_u^{n+1} \,, \tag{3.62}$$

$$w_Z^{n+1} + u_X^{n+1} = R_c^{n+1} \,, \tag{3.63}$$

$$\frac{3w^{n+1} - 4w^n + w^{n-1}}{2\Delta\tau} + \frac{1}{\rho}P_Z^{n+1} - \nu(w_{XX}^{n+1} - q_X^{n+1}) = R_w^{n+1} \,. \tag{3.64}$$

The evolution of the interface is approximated by

$$\frac{3h^{n+1} - 4h^n + h^{n-1}}{2\Delta\tau} = (w^{(I)} - u^{(I)}h_X)^{n+1} \,, \tag{3.65}$$

and the interfacial conditions are approximated by (3.47)-(3.50). An iterative process, similar to that applied to the Crank-Nicolson method, is constructed by using a previous guess to evaluate the nonlinear terms.

Using the same notation as before, we obtain the m-th ($m \geq 1$) iterate for h from

$$\frac{3h^{(n,m)} - 4h^n + h^{n-1}}{2\Delta\tau} = (w^{(I)} - u^{(I)}h_X)^{(n,m-1)} \,. \tag{3.66}$$

Knowing $h^{(n,m)}$ we can evaluate $G_i^{(n,m)}$ $(i = 0, 1, 3)$, $g_i^{(n,m)}$ $(i = 2, 3, 4)$. Then we compute $\{u^{(n,m)}, q^{(n,m)}, w^{(n,m)}, P^{(n,m)}\}$ by

$$u_Z^{(n,m)} - q^{(n,m)} = 0 \,, \tag{3.67}$$

$$\frac{3u^{(n,m)} - 4u^n + u^{n-1}}{2\Delta\tau} + \frac{1}{\rho}P_X^{(n,m)} - \nu(u_{XX}^{(n,m)} + q_Z^{(n,m)}) = R_u^{(n,m-1)} \,, \tag{3.68}$$

$$w_Z^{(n,m)} + u_X^{(n,m)} = R_c^{(n,m-1)} \,, \tag{3.69}$$

$$\frac{3w^{(n,m)} - 4w^n + w^{n-1}}{2\Delta\tau} + \frac{1}{\rho}P_Z^{(n,m)} - \nu(w_{XX}^{(n,m)} - q_X^{(n,m)}) = R_w^{(n,m-1)} \,, \tag{3.70}$$

with the corresponding interfacial condtions given by (3.56)-(3.59). The stopping criterion is the same as (3.60). Once convergence is achieved, we have effectively applied the BDF scheme to (3.66)-(3.70) and treated all the interfacial condtions in a fully implicit manner.

Next, we turn to the spacial discretization which allows the construction of the m-th iterate.

3.4 The Fourier transform

We choose the spacial domain to be a rectangle

$$\{\,(X,Z)\,|\,0 \le X \le 2\pi\,, -H \le Z \le H\,\} \tag{3.71}$$

where H, the vertical computational length, is a prescribed constant. We often pick H to be big enough so that the far-field boundary conditions can be set to 0, accommodating the exponential decay of solutions. Then we make uniform grids with $2K$ points in the X-direction and $2J+1$ points in the Z-direction. Consequently, $\Delta X = \dfrac{\pi}{K}$ and $\Delta Z = \dfrac{H}{J}$.

Since we have assumed all the solutions are periodic in the X-direction, we can take advantage of the discrete Fourier transform to achieve spectral accuracy for X.

In the previous section, we described two ways to march forward in time. For the Crank-Nicolson method, we apply the discrete Fourier transform at each iteration to (3.51)-(3.55) as well as the corresponding interfacial conditions. For the left-hand sides of these equations, we simply replace $\dfrac{\partial}{\partial x}$ by ik, $\dfrac{\partial^2}{\partial x^2}$ by $-k^2$ and

the physical variables $\{h, u, q, w, P\}$ by their k-th Fourier coefficients, where $k = -K, -K+1, \cdots, 0, 1, \cdots, K-1$. For the right-hand sides containning all the non-linear terms and the mapping-associated terms, we carry out the well-known pseudo-spectral approach which consists of 3 steps:

1. use the inverse discrete Fourier transform to recover the physical variables;

2. evaluate the expressions in physical space;

3. use the discrete Fourier transform to obtain the Fourier coefficients of the expressions.

From now on the subscript k will denote the k-th Fourier coefficient of the physical variables. After applying the Fourier transform, the iterative formula (3.52)-(3.55) can be written as as a 4 by 4 linear system of 1st-order ODEs with respect to Z

$$\frac{d}{dZ} Y_k \;=\; B_k(\Delta\tau)\, Y_k + R_k \;, \tag{3.72}$$

wehre

$$\begin{aligned} Y_k \;&\triangleq\; \left(u_k^{(n,m)},\; q_k^{(n,m)},\; w_k^{(n,m)},\; P_k^{(n,m)}\right)^T \;, \\ B_k(\Delta\tau) \;&\triangleq\; \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{\nu\Delta\tau}(2+\nu k^2\Delta\tau) & 0 & 0 & \frac{1}{\rho\nu} ik \\ -ik & 0 & 0 & 0 \\ 0 & -\rho\nu ik & -\frac{\rho}{\Delta\tau}(2+\nu k^2\Delta\tau) & 0 \end{bmatrix} \;, \end{aligned} \tag{3.73}$$

and where the vector R_k contains all the explicit terms, i.e., the terms associated

with the n th time level and the $(m-1)$ th iteration, and is given by the k-th Fourier coefficient of the vector

$$\begin{pmatrix} 0 \\ \frac{-2}{\nu}\left[\frac{1}{2}(R_u^{(n,m-1)}+R_u^n)+\frac{u^n}{\Delta\tau}-\frac{1}{2\rho}P_X^n+\frac{\nu}{2}(u_{XX}^n+q_Z^n)\right] \\ R_c^{n,m-1} \\ 2\rho\left[\frac{1}{2}(R_w^{(n,m-1)}+R_w^n)+\frac{w^n}{\Delta\tau}-\frac{1}{2\rho}P_Z^n+\frac{\nu}{2}(w_{XX}^n-q_X^n)\right] \end{pmatrix} . \tag{3.74}$$

Similarly, for the BDF method (3.67)-(3.70) we obtain

$$\frac{d}{dZ}Y_k \;=\; B_k(\frac{4}{3}\Delta\tau)\,Y_k+\hat{R}_k\;, \tag{3.75}$$

where R_k is the k-th Fourier coefficient of the following vector which contains all the explicit terms,

$$\begin{pmatrix} 0 \\ \frac{-1}{\nu}\left[R_u^{(n,m-1)}+\frac{2u^n}{\Delta\tau}-\frac{u^{n-1}}{2\Delta\tau}\right] \\ R_c^{(n,m-1)} \\ \rho\left[R_w^{(n,m-1)}+\frac{2w^n}{\Delta\tau}-\frac{w^{n-1}}{2\Delta\tau}\right] \end{pmatrix} . \tag{3.76}$$

The interfacial condtions for both the systems (3.72) and (3.75) are given by

$$T_k^{(1)}Y_k^{(1)}-T_k^{(2)}Y_k^{(2)}=r_k\;, \tag{3.77}$$

where

$$T_k \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mu & ik\mu & 0 \\ 0 & 0 & 1 & 0 \\ 2ik\mu & 0 & 0 & 1 \end{bmatrix}, \qquad r_k \triangleq \begin{pmatrix} 0 \\ (S_1^{(n,m-1)})_k \\ 0 \\ (S_2^{(n,m-1)})_k \end{pmatrix} . \tag{3.78}$$

We must now construct numerical solutions to either of the ODE systems (3.72), (3.75) subject to the interfacial conditions (3.77) and the far-field conditions (decaying solutions). The two systems (3.72) and (3.75) have the same structures except that the right-hand-side vectors, R_k and $\hat{R}_k$, are different. Hence an ODE solver working for one system can be very easily modified to work for the other one. So, in the following discussion we will only use the system (3.72) to illustrate the numerical procedure.

3.5 The boundary value problem (BVP)

We notice that the 4 eigenvalues of the matrix $B_k(\Delta\tau)$ in (3.73) are given by

$$\lambda_1 = k \ , \quad \lambda_2 = -k \ , \quad \lambda_3 = \psi(k) \ , \quad \lambda_4 = -\psi(k) \ , \tag{3.79}$$

where

$$\psi(k) \triangleq \sqrt{k^2 + \frac{2}{\nu\Delta\tau}} \ . \tag{3.80}$$

The shooting method is the simplest to apply but there is a difficulty in the choice of shooting parameters for decaying solutions. Further, the rapid exponential growth of the solutions typically causes blow up prior to the numerical solution reaching the interface.

The remedy is to diagonalize the system (3.72) prior to applying the shooting

method (see Section 2.3). The eigenvectors associated with the four eigenvalues in (3.79) are

$$e_1 = \begin{pmatrix} k \\ k^2 \\ -ik \\ \frac{2\rho i}{\Delta\tau} \end{pmatrix}, \; e_2 = \begin{pmatrix} 1 \\ -k \\ i \\ \frac{2\rho i}{k\Delta\tau} \end{pmatrix}, \; e_3 = \begin{pmatrix} \psi(k) \\ \psi^2(k) \\ -ik \\ 0 \end{pmatrix}, \; e_4 = \begin{pmatrix} 1 \\ -\psi(k) \\ \frac{ik}{\psi(k)} \\ 0 \end{pmatrix} \tag{3.81}$$

for $k \neq 0$, and

$$e_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 0 \\ -\frac{\Delta\tau}{2\rho} \\ 0 \end{pmatrix}, \quad e_3 = \begin{pmatrix} \psi(0) \\ \psi^2(0) \\ 0 \\ 0 \end{pmatrix}, \quad e_4 = \begin{pmatrix} 1 \\ -\psi(0) \\ 0 \\ 0 \end{pmatrix} \tag{3.82}$$

for $k = 0$. Define the matrix

$$Q_k = (e_1, e_2, e_3, e_4) \tag{3.83}$$

and introduce the transformation

$$Y_k = Q_k \widetilde{Y}_k \; . \tag{3.84}$$

Then the system (3.72) becomes

$$\frac{d}{dZ}\widetilde{Y}_k = \widetilde{B}_k \widetilde{Y}_k + \widetilde{R}_k \; , \tag{3.85}$$

where $\widetilde{R}_k \triangleq Q_k^{-1} R_k$ and

$$\widetilde{B}_k \triangleq Q_k^{-1} B_k Q_k = \begin{cases} \begin{pmatrix} k & & & \\ & -k & & \\ & & \psi(k) & \\ & & & -\psi(k) \end{pmatrix}, & k \neq 0 , \\ \begin{pmatrix} 0 & 1 & & \\ & 0 & & \\ & & \psi(0) & \\ & & & -\psi(0) \end{pmatrix}, & k = 0 . \end{cases} \tag{3.86}$$

At the interface, the jump condition (3.77) becomes

$$T_k^{(1)} Q_k^{(1)} \widetilde{Y}_k^{(1)} - T_k^{(2)} Q_k^{(2)} \widetilde{Y}_k^{(2)} = r_k , \tag{3.87}$$

where T_k is defined in (3.78) and Q_k in (3.83).

Now the system (3.85) is reduced to four scalar equations in the form of

$$\frac{d}{dZ} \widetilde{y} = \lambda \widetilde{y} + \widetilde{r} \tag{3.88}$$

and we apply the trapezoidal rule, which is second-order accurate in ΔZ, to each of them. Following the discussion after (2.35), we use

$$\widetilde{y}_{j+1} = \frac{1}{(1 - \frac{\Delta Z}{2}\lambda)} \left[(1 + \frac{\Delta Z}{2}\lambda) \widetilde{y}_j + \frac{\Delta Z}{2} (\widetilde{r}_j + \widetilde{r}_{j+1}) \right] , \quad j = -J, -J+1, \cdots, -1 , \tag{3.89}$$

when $\lambda < 0$. That means we start from the bottom, where $\widetilde{y}_{-J}$ is known, and apply (3.89) with increasing j until we reach the interface $j = 0$.

If $\lambda > 0$, we use

$$\widetilde{y}_j = \frac{1}{(1+\frac{\Delta Z}{2}\lambda)}\left[\,(1-\frac{\Delta Z}{2}\lambda)\widetilde{y}_{j+1} - \frac{\Delta Z}{2}(\widetilde{r}_j + \widetilde{r}_{j+1})\,\right]\,,\ j = J-1, J-2\cdots, 0\,. \quad (3.90)$$

That means we start from the top, where $\widetilde{y}_J$ is known, and apply (3.90) with decreasing j until we reach the interface $j = 0$.

We apply this one-way shooting method in three steps. First, we shoot from the top and bottom using only those ODEs that have positive and negative eigenvalues, respectively. Consequently, some of the unknowns are now determined, in particular, at the interface. Second, we use the known quantities to write the interfacial conditions as a linear system of algebraic equations for the remaining unknowns. Once this algebraci system has been solved, we use the known values at the interface to complete the third step. We shoot from the interface upwards and downwards using those ODEs that have negative and positive eigenvalues, respectively.

The details of this procedure for the case $k > 0$ are as follows. From (3.86) we see the two positive eigenvalues correspond to the 1st and the 3rd components of $\widetilde{Y}_k$, denoted by $\widetilde{y}_1$ and $\widetilde{y}_3$, respectively. They are determined by the recursion (3.90). The two negative eigenvalues correspond to the 2nd and the 4th components of $\widetilde{Y}_k$, denoted by $\widetilde{y}_2$ and $\widetilde{y}_4$, respectively. They are determined by the recursion (3.89). At the interface $j = 0$, (3.87) gives

$$T_k^{(1)}\,Q_k^{(1)}\begin{pmatrix}\widetilde{y}_1^{(1)}\\ \widetilde{y}_2^{(1)}\\ \widetilde{y}_3^{(1)}\\ \widetilde{y}_4^{(1)}\end{pmatrix}_{j=0} - T_k^{(2)}\,Q_k^{(2)}\begin{pmatrix}\widetilde{y}_1^{(2)}\\ \widetilde{y}_2^{(2)}\\ \widetilde{y}_3^{(2)}\\ \widetilde{y}_4^{(2)}\end{pmatrix}_{j=0} = r_k\,, \quad (3.91)$$

where $\{\widetilde{y}_1^{(1)}, \widetilde{y}_3^{(1)}, \widetilde{y}_2^{(2)}, \widetilde{y}_4^{(2)}\}_{j=0}$ are already known. Hence (3.91) forms a 4 by 4 linear system for the unknowns $\{\widetilde{y}_2^{(1)}, \widetilde{y}_4^{(1)}, \widetilde{y}_1^{(2)}, \widetilde{y}_3^{(2)}\}_{j=0}$. Once they are solved, we can continue the computation for $\widetilde{y}_1, \widetilde{y}_3$ by following (3.90) with the indices shifted to: $j = -1, -2, \cdots, -J$, and for $\widetilde{y}_2, \widetilde{y}_4$ by following (3.89) with the indices shifted to: $j = 0, 1, \cdots, J-1$.

The situation is a little different in the case $k = 0$ since we have two zero eigenvalues, corresponding to $\widetilde{y}_1$ and $\widetilde{y}_2$. Moreover, the calculation of $\widetilde{y}_1$ requires $\widetilde{y}_2$ (see (3.86)). Therefore, when applying the shooting method, we always shoot from the same place and go through the same direction (either upwards or downwards) for the two ODEs that have zero eigenvalues. There is no change for the other parts in this method.

Finally, we remind the reader that the discussion in this subsection is made for the matrix $B_k(\Delta\tau)$. When our strategy is applied to to the system (3.75), where the matrix is $B_k(\frac{4}{3}\Delta\tau)$, we have to replace $\Delta\tau$ by $\frac{4}{3}\Delta\tau$.

3.6 A different approach

The method described above is the one we apply to most of our numerical simulations. Here, for comparison, we discuss a different method which employs the GMRES algorithm. This method is also constructed on the mapped equations and has some similar parts as the previous method. The important feature of this new method is that it doesn't require the extraction of linear terms. Instead, we directly perform the temporal and the spacial discretizations to the full mapped equations, which results

in a huge linear system with variable coefficients. Then the GMRES algorithm is applied to find the solutions.

The method is based on applying different time marching algorithms to different terms in the equations. As a simple illustration, consider

$$\frac{\partial u}{\partial t} + g(u) = f(u) \,, \tag{3.92}$$

where $g(u)$ contains the diffusion terms and $f(u)$ represents the advection terms. The application of the method (2.8) to (3.92) yields

$$\frac{u^{n+1} - u^n}{\Delta t} + \frac{1}{2}\big(g(u^{n+1}) + g(u^n)\big) = \frac{3}{2} f(u^n) - \frac{1}{2} f(u^{n-1}) \,, \tag{3.93}$$

where the Crank-Nicolson scheme is applied to the linear diffusion terms in $g(u)$ and the Adams-Bashforth scheme is applied to the nonlinear advection terms in $f(u)$. A linear system must be solved to find u^{n+1} and this is where GMRES proves useful.

This method is also second-order accurate and with good stability properties. Our numerical tests show that this method actually has better stability properties than the previous method, i.e., permitting bigger values of Δz and Δt. The other reason to introduce this method is that it can provide a consistency check on all the numerical results. We found very good agreement for the results from these different methods and we are therefore confident about the validity of our results. The disadvantage of this new method is that much more computing effort is required since, as we will see soon, we are working with a linear system with a huge coefficient matrix.

3.6.1 The numerical method

For convenience, we list again the governing equations with the unknowns $\{u, q, w, P\}$:

$$u_Z = q \,, \tag{3.94}$$

$$u_\tau - G_0 q + u(u_X - G_1 q) + w G_3 q = -\frac{1}{\rho} P_X + \frac{1}{\rho} G_1 P_Z + \nu \mathcal{L}\{u\} \,, \tag{3.95}$$

$$u_X - G_1 q + G_3 w_Z = 0 \,, \tag{3.96}$$

$$w_\tau - G_0 w_Z + u(w_X - G_1 w_Z) + w G_3 w_Z = -\frac{1}{\rho} G_3 P_Z + \nu \mathcal{L}\{w\} \,, \tag{3.97}$$

where $\mathcal{L}$ is the mapped Laplacian and

$$\mathcal{L}\{u\} = u_{XX} + g_2\, q_Z + g_3\, q_X + g_4\, q \,. \tag{3.98}$$

We note that, by differentiating (3.96) with respect to X and Z, respectively, we obtain

$$w_{XZ} = -\frac{1}{G_3} [\, u_{XX} - (G_1)_X\, q - G_1\, q_X + (G_3)_X\, w_Z \,] \,, \tag{3.99}$$

$$w_{ZZ} = -\frac{1}{G_3} [\, q_X - (G_1)_Z\, q - G_1\, q_Z + (G_3)_Z\, w_Z \,] \,. \tag{3.100}$$

Hence we have

$$\begin{aligned} \mathcal{L}\{w\} &= w_{XX} + g_2\, w_{ZZ} + g_3\, w_{XZ} + g_4\, w_Z \\ &= w_{XX} - \frac{1}{G_3} [\, g_2\, (q_X - (G_1)_Z\, q) + g_3\, (u_{XX} - (G_1)_X\, q - G_1\, q_X) \,] \\ &\quad - [\, \frac{1}{G_3} (\, g_2 (G_3)_Z + g_3 (G_3)_X \,) - g_4 \,] w_Z + \frac{G_1}{G_3} g_2\, q_Z \,. \end{aligned} \tag{3.101}$$

All the derivatives with respect to Z are first-order in the governing system (3.94)-(3.97).

The numerical approach is as follows: The Crank-Nicolson method is applied to the diffusion terms and the Adams-Bashforth method to the advection terms to advance the solution in time. The discrete Fourier transform is applied along the horizontal direction, X, which is assumed to possess periodicity. Then we obtain a first-order ODE system with respect to Z, the vertical coordinate. The trapezoidal rule is applied to this ODE system which results in a linear system, where the coefficient matrix is huge and has variable entries. We will use the GMRES method to solve the linear system at each time step.

Let's consider the details of the approximation in time. By using the Crank-Nicolson scheme for the linear terms and the Adam-Bashforth scheme for the nonlinear terms, we obtain

$$\begin{aligned}
\frac{u^{n+1}-u^n}{\Delta\tau} &- \frac{1}{2}[\,(G_0 q)^{n+1} + (G_0 q)^n\,] \\
&+ \frac{3}{2}[\,u(u_X - G_1 q) + wG_3 q\,]^n - \frac{1}{2}[\,u(u_X - G_1 q) + wG_3 q\,]^{n-1} \\
&= \frac{1}{2}\,[\,-\frac{1}{\rho}P_X + \frac{1}{\rho}G_1 P_Z + \nu\mathcal{L}\{u\}\,]^{n+1} + \\
&\quad \frac{1}{2}\,[\,-\frac{1}{\rho}P_X + \frac{1}{\rho}G_1 P_Z + \nu\mathcal{L}\{u\}\,]^n\,, \qquad (3.102) \\
\frac{w^{n+1}-w^n}{\Delta\tau} &- \frac{1}{2}[\,(G_0 w_Z)^{n+1} + (G_0 w_Z)^n\,] \\
&+ \frac{3}{2}[\,u(w_X - G_1 w_Z) + wG_3 w_Z\,]^n - \frac{1}{2}[\,u(w_X - G_1 w_Z) + wG_3 w_Z\,]^{n-1} \\
&= \frac{1}{2}\,[\,-\frac{1}{\rho}G_3 P_Z + \nu\mathcal{L}\{w\}\,]^{n+1} + \frac{1}{2}\,[\,-\frac{1}{\rho}G_3 P_Z + \nu\mathcal{L}\{w\}\,]^n\,. \qquad (3.103)
\end{aligned}$$

By substituting (3.98) and (3.101) into (3.102) and (3.103) and rearranging terms,

we obtain

$$
\begin{aligned}
u_Z^{n+1} &= q^{n+1} \,, && (3.104)\\
\frac{1}{2\rho} G_1^{n+1} P_Z^{n+1} &+ \frac{\nu}{2} g_2^{n+1} q_Z^{n+1} = U^{n+1} + E \,, && (3.105)\\
G_3^{n+1} w_Z^{n+1} &= -u_X^{n+1} + G_1^{n+1}\, q^{n+1} \,, && (3.106)\\
-\frac{1}{2\rho} G_3^{n+1} P_Z^{n+1} &+ [\, \frac{1}{2} G_0 + \frac{\nu}{2} g_4 - \frac{\nu}{2} \frac{1}{G_3} (\, g_2 (G_3)_Z + g_3 (G_3)_X \,) \,]^{n+1}\, w_Z^{n+1} \\
&+ \frac{\nu}{2} (\frac{G_1}{G_3} g_2)^{n+1}\, q_Z^{n+1} = V^{n+1} + F \,, && (3.107)
\end{aligned}
$$

where

$$
\begin{aligned}
U^{n+1} &= \frac{u^{n+1}}{\Delta\tau} - \frac{1}{2} (G_0\, q)^{n+1} - \frac{1}{2} [\, -\frac{1}{\rho} P_X + \nu\, (u_{XX} + g_3\, q_X + g_4\, q) \,]^{n+1} \,, && (3.108)\\
V^{n+1} &= \frac{w^{n+1}}{\Delta\tau} - \frac{\nu}{2} \{\, w_{XX} - \frac{1}{G_3} [\, g_2\, (q_X - (G_1)_Z\, q) \\
&\quad + g_3\, (u_{XX} - (G_1)_X\, q - G_1\, q_X) \,] \,\}^{n+1} \,, && (3.109)
\end{aligned}
$$

and where E, F denote the explicit terms

$$
\begin{aligned}
E &= -\frac{u^n}{\Delta\tau} - \frac{1}{2} (G_0\, q)^n \\
&\quad + \frac{3}{2} [\, u(u_X - G_1 q) + w G_3 q \,]^n - \frac{1}{2} [\, u(u_X - G_1 q) + w G_3 q \,]^{n-1} \\
&\quad - \frac{1}{2} [\, -\frac{1}{\rho} P_X + \frac{1}{\rho} G_1 P_Z + \nu\, (u_{XX} + g_2\, q_Z + g_3\, q_X + g_4\, q) \,]^n \,, && (3.110)\\
F &= -\frac{w^n}{\Delta\tau} - \frac{1}{2} (G_0\, w_Z)^n \\
&\quad + \frac{3}{2} [\, u(w_X - G_1 w_Z) + w G_3 w_Z \,]^n - \frac{1}{2} [\, u(w_X - G_1 w_Z) + w G_3 w_Z \,]^{n-1} \\
&\quad - \frac{1}{2} \{\, -\frac{1}{\rho} G_3 P_Z + \nu\, [\, w_{XX} - \frac{1}{G_3} [\, g_2\, (q_X - (G_1)_Z\, q) + g_3\, (u_{XX} - (G_1)_X\, q - G_1\, q_X) \,] \\
&\quad - [\, \frac{1}{G_3} (\, g_2 (G_3)_Z + g_3 (G_3)_X \,) - g_4 \,] w_Z + \frac{G_1}{G_3} g_2\, q_Z \,] \,\}^n \,. && (3.111)
\end{aligned}
$$

Now (3.104)-(3.107) form a first-order linear system of ODEs with respect to Z. We apply the trapezoid rule to perform the integration of this system and obtain

$$\frac{u_{j+1}^{n+1} - u_j^{n+1}}{\Delta Z} - \frac{1}{2}(q_{j+1}^{n+1} + q_j^{n+1}) = 0 , \tag{3.112}$$

$$\begin{aligned} \frac{1}{2\rho}(G_1^{n+1})_{j+\frac{1}{2}} \frac{P_{j+1}^{n+1} - P_j^{n+1}}{\Delta Z} &+ \frac{\nu}{2}(g_2^{n+1})_{j+\frac{1}{2}} \frac{q_{j+1}^{n+1} - q_j^{n+1}}{\Delta Z} \\ -\frac{(U^{n+1})_{j+1} + (U^{n+1})_j}{2} &= \frac{E_{j+1} + E_j}{2} , \end{aligned} \tag{3.113}$$

$$\begin{aligned} (G_3^{n+1})_{j+\frac{1}{2}} \frac{w_{j+1}^{n+1} - w_j^{n+1}}{\Delta Z} &+ \frac{1}{2}[\,(u_X^{n+1})_{j+1} + (u_X^{n+1})_j\,] \\ -\frac{1}{2}\Big[\,(G_1^{n+1}q^{n+1})_{j+1} &+ (G_1^{n+1}q^{n+1})_j\,\Big] = 0 , \end{aligned} \tag{3.114}$$

$$\begin{aligned} -\frac{1}{2\rho}(G_3^{n+1})_{j+\frac{1}{2}} \frac{P_{j+1}^{n+1} - P_j^{n+1}}{\Delta Z} &+ \frac{\nu}{2}(\frac{G_1}{G_3}g_2)_{j+\frac{1}{2}}^{n+1} \frac{q_{j+1}^{n+1} - q_j^{n+1}}{\Delta Z} \\ +\Big[\,\frac{1}{2}G_0 + \frac{\nu}{2}g_4 - \frac{\nu}{2}\frac{1}{G_3}\,\big(\,g_2(G_3)_Z &+ g_3(G_3)_X\,\big)\,\Big]_{j+\frac{1}{2}}^{n+1} \frac{w_{j+1}^{n+1} - w_j^{n+1}}{\Delta Z} \\ -\frac{(V^{n+1})_{j+1} + (V^{n+1})_j}{2} &= \frac{F_{j+1} + F_j}{2} . \end{aligned} \tag{3.115}$$

We note that all the terms in the left-hand sides of the above equations are evaluated at $t = (n+1)\Delta t$ with superscipt $n+1$ and all the terms in the right-hand sides are evaluated at previous times with superscripts n or $n-1$. We have yet to apply an approximation in the X direction, where we plan to take advantage of the Fourier transform to achieve spectral accuracy. Since the coefficients in the above equations, such as G_0, G_1, G_3, g_2, g_3, g_4, all depend on X, application of the Fourier transform to any product with these coefficients will require the convolution rule and make the results far too complicated. Fortunately, the GMRES method doesn't require the explicit structure of the iteration matrix. This enables us to

apply the pseudo-spectral approach to equations (3.112)-(3.115) to obtain a linear system for each Fourier mode k,

$$A_k\, Y_k \;=\; S_k\;, \tag{3.116}$$

where the unknown, Y_k, is composed of the k-th Fourier coefficients of u, q, w and P at all the vertical positions, $j = -M, -M+1, \cdots, -1, 0, 1, \cdots, M$ and where the right-hand side vector S_k comes from the k-th Fourier coefficients of the right-hand sides in equations (3.112)-(3.115).

3.6.2 The GMRES iterations

We use GMRES to solve equation (3.116). The main steps in performing a GMRES iteration are as follows:

(1) Calculate the right-hand side vector S_k, which we have discussed in detail.

(2) Make an initial guess for the solutions.

(3) Apply a good preconditioner.

(4) Apply the pseudo-spectral approach to perform the matrix-vector multiplication.

It's relatively easy to perform the matrix-vector multiplication, say $A_k\,\widetilde{Y}$. This is achieved by substituting the vector $\widetilde{Y}$ into the left-hand sides of equations (3.112)-(3.115) and applying the pseudo-spectral approach. As a result, we obtain the right-hand sides corresponding to that vector $\widetilde{Y}$ and they give exactly $A_k\,\widetilde{Y}$. Note that the interfacial conditions (3.118)-(3.121) are also included in the calculation with the pseudo-spectral approach.

Finally, we need boundary conditions to close the system (3.112)-(3.115). We need four conditions at the two ends and four conditions at the interface. We use

$$u = 0\,, \quad w = 0 \tag{3.117}$$

on both the top and the bottom. At the interface, we have the continuity of the velocities and two stress conditions. They read:

$$u^{(1)} - u^{(2)} = 0\,, \tag{3.118}$$

$$w^{(1)} - w^{(2)} = 0\,, \tag{3.119}$$

$$\begin{aligned} \mu^{(1)}(G_3^{(1)}q^{(1)} + w_X^{(1)}) &- \mu^{(2)}(G_3^{(2)}q^{(2)} + w_X^{(2)}) + (\frac{4h_X}{h_X^2 - 1} + \frac{G_1^{(1)}}{G_3^{(1)}})\mu^{(1)}(u_X^{(1)} - G_1^{(1)}q^{(1)}) \\ &- (\frac{4h_X}{h_X^2 - 1} + \frac{G_1^{(2)}}{G_3^{(2)}})\mu^{(2)}(u_X^{(2)} - G_1^{(2)}q^{(2)}) = 0\,, \end{aligned} \tag{3.120}$$

$$\begin{aligned} (P^{(1)} - P^{(2)}) &+ (2 - \frac{4h_X^2}{h_X^2 - 1})[\mu^{(1)}(u_X^{(1)} - G_1^{(1)}q^{(1)}) - \mu^{(2)}(u_X^{(2)} - G_1^{(2)}q^{(2)})] \\ &= gh(\rho^{(1)} - \rho^{(2)}) + 2T\kappa\,. \end{aligned} \tag{3.121}$$

These conditions are evaluated at $t = t^{n+1}$ and $Z = 0$. That means they are treated implicitly.

Now we write out the details for making the initial guess. Let's go back to the system (3.104)-(3.107). We linearize all the terms evaluated at $t = (n+1)\Delta t$ (i.e., terms with superscripts $n+1$) so that the coefficients are approximated by

$$G_0 = 0, \quad G_1 = 0, \quad G_3 = 1, \quad g_2 = 1, \quad g_3 = 0, \quad g_4 = 0\,. \tag{3.122}$$

We are led to the approximate system

$$
\begin{aligned}
u_Z^{n+1} &= q^{n+1} \,, && (3.123)\\
\frac{\nu}{2} q_Z^{n+1} &= \frac{u^{n+1}}{\Delta\tau} + \frac{1}{2\rho} P_X^{n+1} - \frac{\nu}{2} u_{XX}^{n+1} + E \,, && (3.124)\\
w_Z^{n+1} &= -u_X^{n+1} \,, && (3.125)\\
-\frac{1}{2\rho} P_Z^{n+1} &= \frac{w^{n+1}}{\Delta\tau} + \frac{\nu}{2} q_X^{n+1} - \frac{\nu}{2} w_{XX}^{n+1} + F \,. && (3.126)
\end{aligned}
$$

Then we perform the Fourier transform in X on the above equations (3.123)-(3.126), which yields the following ODE system for each Fourier mode k,

$$
\frac{d}{dZ} Y_k \;=\; B_k(\Delta\tau)\, Y_k + R_k \,, \qquad (3.127)
$$

where

$$
B_k(\Delta\tau) \triangleq \begin{bmatrix}
0 & 1 & 0 & 0 \\
\frac{1}{\nu\Delta\tau}(2+\nu k^2 \Delta\tau) & 0 & 0 & \frac{1}{\rho\nu} ik \\
-ik & 0 & 0 & 0 \\
0 & -\rho\nu ik & -\frac{\rho}{\Delta\tau}(2+\nu k^2\Delta\tau) & 0
\end{bmatrix} , \qquad (3.128)
$$

and

$$
Y_k \triangleq \begin{pmatrix} u_k^{n+1} \\ q_k^{n+1} \\ w_k^{n+1} \\ P_k^{n+1} \end{pmatrix} , \qquad R_k \triangleq \begin{pmatrix} 0 \\ \frac{2}{\nu} E_k \\ 0 \\ -2\rho F_k \end{pmatrix} . \qquad (3.129)
$$

As before, the subscript k refers to the k-th Fourier coefficient of the corresponding physical variable.

We also linearize the interfacial conditions (3.118)-(3.121) to obtain

$$u^{(1)} - u^{(2)} = 0 \,, \tag{3.130}$$

$$\mu^{(1)}(q^{(1)} + w_X^{(1)}) - \mu^{(2)}(q^{(2)} + w_X^{(2)}) = 0 \,, \tag{3.131}$$

$$w^{(1)} - w^{(2)} = 0 \,, \tag{3.132}$$

$$(P^{(1)} - P^{(2)}) + 2(\mu^{(1)} u_X^{(1)} - \mu^{(2)} u_X^{(2)}) = gh(\rho^{(1)} - \rho^{(2)}) + T h_{XX} \,. \tag{3.133}$$

After the Fourier transform is applied, the above equations can be written as

$$T_k^{(1)} Y_k^{(1)} - T_k^{(2)} Y_k^{(2)} = r_k \,, \tag{3.134}$$

where

$$T_k \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mu & ik\mu & 0 \\ 0 & 0 & 1 & 0 \\ 2ik\mu & 0 & 0 & 1 \end{bmatrix} , \qquad r_k \triangleq \begin{pmatrix} 0 \\ 0 \\ 0 \\ \left(g(\rho^{(1)} - \rho^{(2)}) - k^2 T\right) h_k \end{pmatrix} . \tag{3.135}$$

Note that the ODE system (3.127) with the interfacial conditions (3.134) are in the same form as (3.72) and (3.77) in Section 3.4, except that the right-hand side vectors R_k and r_k are different. Hence we can use the same technique as before to compute the solutions. Specifically, we first diagonalize the system (3.127), then apply the trapezoidal rule to the transformed diagonal system and shoot in the appropriate directions as described in Section 3.5. Once the solutions for the diagonal system are obtained, we transform them back to obtain the solutions for the original system (3.127). These solutions Y_k, obtained for all the vertical points, will serve as the initial guess for GMRES.

Fortunately, the preconditioner works in the same way as making the initial guess, except that the right-hand side vectors R_k in (3.127) and r_k in (3.134) are different. They are the residuals in the iteration and are provided by GMRES. We simply substitute them into (3.127) and (3.134) and apply the procedure described above to construct the solutions.

In our numerical simulation of the interfacial flows, we found the GMRES typically requires 6-10 iterations at each time step to achieve convergence. Overall, this method requires about three times as much CPU time as the previous method does.

3.7 Conversion to dimensionless units

To reduce the number of computations, it's convenient to use dimensionless variables. Let β be the wave number and introduce the dimensionless variables

$$\begin{aligned} \bar{x} = \beta x\,, \qquad \bar{z} = \beta z\,, \qquad \bar{t} = \sqrt{g\beta}\, t\,, \\ \bar{u} = \sqrt{\frac{\beta}{g}}\, u\,, \qquad \bar{w} = \sqrt{\frac{\beta}{g}}\, w\,, \qquad \bar{P} = \frac{\beta}{\rho g} P\,, \quad \bar{h} = \beta h\,. \end{aligned} \tag{3.136}$$

Let's further define the dimensionless parameters

$$\bar{\rho} = \frac{\rho^{(1)}}{\rho^{(2)}}\,, \quad \bar{\mu} = \frac{\mu^{(1)}}{\mu^{(2)}}\,, \quad \bar{T} = \frac{\beta^2}{\rho^{(2)} g} T\,, \quad \bar{\kappa} = \frac{1}{k}\kappa\,, \quad R_e = \frac{\sqrt{g}}{\sqrt{\beta^3}\,\nu}\,. \tag{3.137}$$

Then we can write all the equations in dimensionless form,

$$\bar{u}_{\bar{t}} + \bar{u}\bar{u}_{\bar{x}} + \bar{w}\bar{u}_{\bar{z}} = -\bar{P}_{\bar{x}} + \frac{1}{R_e}(\bar{u}_{\bar{x}\bar{x}} + \bar{u}_{\bar{z}\bar{z}})\,, \tag{3.138}$$

$$\bar{w}_{\bar{t}} + \bar{u}\bar{w}_{\bar{x}} + \bar{w}\bar{w}_{\bar{z}} = -\bar{P}_{\bar{z}} + \frac{1}{R_e}(\bar{w}_{\bar{x}\bar{x}} + \bar{w}_{\bar{z}\bar{z}})\,, \tag{3.139}$$

$$\bar{u}_{\bar{x}} + \bar{w}_{\bar{z}} = 0\,. \tag{3.140}$$

The kinematic condition (3.5) and the continuity of velocity at interface (3.6) keep the same form,

$$\bar{h}_{\bar{t}} + \bar{u}^{(I)}\bar{h}_{\bar{x}} = \bar{w}^{(I)} \,, \tag{3.141}$$

$$\bar{u}^{(1)} = \bar{u}^{(2)} = \bar{u}^{(I)} \,, \quad \bar{w}^{(1)} = \bar{w}^{(2)} = \bar{w}^{(I)} \,. \tag{3.142}$$

The dynamical interfacial conditions (3.7) and (3.8) are changed into

$$\begin{aligned} & (\bar{h}_{\bar{x}}^2 - 1)\,[\,\bar{\mu}(\bar{u}_{\bar{z}}^{(1)} + \bar{w}_{\bar{x}}^{(1)}) - (\bar{u}_{\bar{z}}^{(2)} + \bar{w}_{\bar{x}}^{(2)})\,] \\ + \;& 2\bar{h}_{\bar{x}}\,[\,\bar{\mu}(\bar{u}_{\bar{x}}^{(1)} - \bar{w}_{\bar{z}}^{(1)}) - (\bar{u}_{\bar{x}}^{(2)} - \bar{w}_{\bar{z}}^{(2)})\,] \;=\; 0 \,, \end{aligned} \tag{3.143}$$

$$\begin{aligned} (\bar{\rho}\bar{P}^{(1)} - \bar{P}^{(2)}) \;-\; & (\bar{\rho} - 1)\bar{h} + \frac{1}{R_e^{(2)}}\,\bar{h}_{\bar{x}}\,[\,\bar{\mu}(\bar{u}_{\bar{z}}^{(1)} + \bar{w}_{\bar{x}}^{(1)}) - (\bar{u}_{\bar{z}}^{(2)} + \bar{w}_{\bar{x}}^{(2)})\,] \\ - \;& \frac{2}{R_e^{(2)}}\,[\,\bar{\mu}\bar{w}_{\bar{z}}^{(1)} - \bar{w}_{\bar{z}}^{(2)}\,] \;-\; 2\bar{T}\bar{\kappa} \;=\; 0 \,. \end{aligned} \tag{3.144}$$

Our numerical methods can be applied directly to equations (3.138)-(3.144). One simple way to convert the dimentional numerical methods into the dimensionless form is to set

$$\rho^{(1)} = \rho^{(1)} = 1 \,, \quad \nu^{(1)} = \frac{1}{R_e^{(1)}} \,, \quad \nu^{(2)} = \frac{1}{R_e^{(2)}} \,, \quad \mu^{(1)} = \bar{\mu} \,, \quad \mu^{(2)} = 1 \,.$$

What's left is just a few modifications of the coefficients in the 2nd dynamical interfacial condition, which is easy to accomplish.

CHAPTER 4

NUMERICAL RESULTS

4.1 Numerical verification of accuracy

While there is little doubt about the spectral accuracy in the X-direction where the Fourier transform is applied, the order of accuracy for the time marching and the discretization in the Z-direction is to be justified by numerical experiments. Two examples serve for that purpose. These tests are performed on a 2.4GHz Xeon dual-processor workstation.

In the first example, we consider the incompressible Navier-Stokes equations (3.1)-(3.3) with the exact solutions

$$\begin{aligned} u &= t\sin(2x)e^{-2z}\,, \\ w &= t\cos(2x)e^{-2z}\,, \\ P &= \frac{\rho}{2}(\cos(2x)e^{-2z} - t^2e^{-4z})\,, \end{aligned} \tag{4.1}$$

which are exponentially decaying in the vertical direction and periodic in the horizontal direction. The spatial domain is defined as

$$\{\,(x,z)\,|\,0 \le x \le 2\pi\,,\; h(x,t) \le z \le 1\,\}\,, \tag{4.2}$$

where h, the bottom, is artificially set as

$$h(x,t) = 0.1\sin(x-t)\ . \tag{4.3}$$

Only one fluid is concerned and there is actually no interface. Nevertheless, when applying our methods the wavy boundary $h(x,t)$ serves as the "interface", on which the mappings (3.10)-(3.12) are readily formed (for $Z \geq 0$ only) and all the parts in our approach are readily applied. That means the mapped equations and the formulation as a BVP in Z will be thoroughly tested. Obviously, the numerical treatment of the interfacial conditions will not be tested. The initial and the boundary values for u, w, P are taken from the exact solution (4.1).

We perform the computation for $\rho = 1$, $\mu = 0.313$ (corresponding to $R_e \doteq 100$) and run the codes until $\tau = 0.4$. We use 32 points in the X-direction so that the errors assoiciated with ΔX is much much smaller than those associated with Δt and ΔZ. Let N be the number of time steps and J the number of points in the Z-direction. We keep doubling N and J to check the error pattern. The results are presented in Table 4.1, where $E(u,N,J)$ denotes the L_2-norm of the errors for u with the resolution of N time steps and J points in the Z-direction, and where $R(u,N,J)$ denotes the quantity

$$\sqrt{\frac{E(u,N/2,J/2)}{E(u,N,J)}}$$

Similar notations hold for $E(w,N,J)$, $R(w,N,J)$ and $E(P,N,J)$, $R(P,N,J)$. The results clearly indicate the 2nd-order convergence in both $\Delta\tau$ and ΔZ.

In the second example, we consider the two-fluid case with an interface. Due to the presence of the nonlinear interfacial conditions (3.6)-(3.8), an analytical form of

N	J	$E(u,N,J)$ ($R(u,N,J)$)	$E(w,N,J)$ ($R(w,N,J)$)	$E(P,N,J)$ ($R(P,N,J)$)
40	40	1.824×10^{-5} ($--$)	6.069×10^{-6} ($--$)	4.469×10^{-5} ($--$)
80	80	4.680×10^{-6} (1.97)	$1.512E \times 10^{-6}$ (2.00)	1.207×10^{-5} (1.92)
160	160	1.194×10^{-6} (1.98)	3.833×10^{-7} (1.99)	3.018×10^{-6} (2.00)

Table 4.1: Results for the first test case

solutions is not available in this case. However, we know there are exact solutions for the linearized problem. If we use that linearized solutions as the initial conditions and set the amplitude of the interface h to be small enough, then the influence of the nonlinear terms in both the governing equations and the interfacial conditions becomes unimportant since they are in the order of $O(h^2)$ and we expect the solutions of our nonlinear problem will be very close to that of the linearized problem. Hence we will use the linearized solutions as the reference solutions to test the accuracy of our numerical methods. At the same time, the numerical treatment of the interfacial conditions will be tested at least at the linear level.

Solutions for the linear motion of interfacial flows are available in [12] and most

recently in [7]. They take the form of

$$\begin{pmatrix} u_k \\ w_k \\ P_k \\ h_k \end{pmatrix} = \begin{pmatrix} u_k^0 \\ w_k^0 \\ P_k^0 \\ h_k^0 \end{pmatrix} e^{\sigma(k)\,t} \ , \tag{4.4}$$

where the subscript k specifies the k-th Fourier coefficient and the superscript 0 indicates the initial state. The value of $\sigma(k)$ is found through the dispersion relation

$$\begin{aligned} & [\rho^{(1)}\sqrt{\nu^{(1)}}(\Omega^{(1)} + \sqrt{\nu^{(1)}}k) + \rho^{(2)}\sqrt{\nu^{(2)}}(\Omega^{(2)} + \sqrt{\nu^{(2)}}k)] \\ \times \quad & [(\rho^{(2)} - \rho^{(1)})gk + Tk^3 + (\rho^{(2)} + \rho^{(1)})\sigma^2(k)] \\ + \quad & 4(\rho^{(1)}\sqrt{\nu^{(1)}}\Omega^{(1)} + \rho^{(2)}\nu^{(2)}k)(\rho^{(2)}\sqrt{\nu^{(2)}}\Omega^{(2)} + \rho^{(1)}\nu^{(1)}k)\sigma(k)k = 0 \ , \end{aligned} \tag{4.5}$$

where $\Omega^{(1)} = \sqrt{\sigma(k) + \nu^{(1)}k^2}$, $\Omega^{(2)} = \sqrt{\sigma(k) + \nu^{(2)}k^2}$. Newton's method is applied to numericaly find the roots of the nonlinear equation (4.5). Once $\sigma(k)$ is determined, the initial values $\{u_k^0\,,\, w_k^0\,,\, P_k^0\,,\, h_k^0\}$ are determined as follows.

In the upper domain:

$$\begin{aligned} w_k^0 &= A\exp(-|k|z) + B\exp(-\frac{\Omega^{(1)}}{\sqrt{\nu^{(1)}}}\,z) \ , \\ u_k^0 &= -\frac{i|k|}{k}A\exp(-|k|z) - \frac{i}{k}\frac{\Omega^{(1)}}{\sqrt{\nu^{(1)}}}B\exp(-\frac{\Omega^{(1)}}{\sqrt{\nu^{(1)}}}\,z) \ , \\ P_k^0 &= \frac{\rho^{(1)}\sigma(k)}{|k|}A\exp(-|k|z) \ . \end{aligned} \tag{4.6}$$

In the lower domain:

$$\begin{aligned} w_k^0 &= C\exp(|k|z) + D\exp(\frac{\Omega^{(2)}}{\sqrt{\nu^{(2)}}}\,z)\,, \\ u_k^0 &= \frac{i|k|}{k}C\exp(|k|z) + \frac{i}{k}\frac{\Omega^{(2)}}{\sqrt{\nu^{(2)}}}D\exp(\frac{\Omega^{(2)}}{\sqrt{\nu^{(2)}}}\,z)\,, \\ P_k^0 &= -\frac{\rho^{(2)}\sigma(k)}{|k|}C\exp(|k|z)\,. \end{aligned} \tag{4.7}$$

For the interface:

$$h_k^0 = \frac{a}{2}\,, \tag{4.8}$$

where a is a small real constant that specifies the amplitude and where A, B, C, D are constants determined by a and $\sigma(k)$,

$$\begin{aligned} A &= \frac{aS\,\sigma(k)}{2(S-R)}\,, \qquad\qquad B = \frac{aR\,\sigma(k)}{2(R-S)}\,, \\ C &= -\frac{\sqrt{\nu^{(2)}}k + \Omega^{(2)}}{\sqrt{\nu^{(2)}}k - \Omega^{(2)}}A - \frac{\sqrt{\nu^{(2)}}\Omega^{(1)} + \sqrt{\nu^{(1)}}\Omega^{(2)}}{\sqrt{\nu^{(2)}}k - \Omega^{(2)}}\frac{B}{\sqrt{\nu^{(1)}}}\,, \\ D &= A + B - C\,, \end{aligned}$$

where

$$\begin{aligned} R &= 2\rho^{(1)}\nu^{(1)}k^2 + 2\rho^{(2)}\sqrt{\nu^{(2)}}\,k\Omega^{(2)}\,, \\ S &= \rho^{(1)}\big(\sigma(k) + 2\nu^{(1)}k^2\big) + \rho^{(2)}\Omega^{(1)}\sqrt{\frac{\nu^{(2)}}{\nu^{(1)}}}\big(\sqrt{\nu^{(2)}}k + \Omega^{(2)}\big) - \rho^{(2)}\sqrt{\nu^{(2)}}k\big(\sqrt{\nu^{(2)}}k - \Omega^{(2)}\big)\,. \end{aligned}$$

In our test we pick $k = 1$, $a = 0.01$ and consider the air-water case with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$ (corresponding to $R_e^{(1)} \doteq 208.70$, $R_e^{(2)} \doteq 2845.9$). The domain of computation is chosen to be a rectangle as is defined in (3.71) with $H = 1$. The initial conditions and the boundary values at the

N	J	$E(u,N,J)$ $(\ R(u,N,J)\ )$	$E(w,N,J)$ $(\ R(w,N,J)\ )$	$E(P,N,J)$ $(\ R(P,N,J)\ )$	$E(h,N,J)$ $(\ R(h,N,J)\ )$
40	40	3.717×10^{-2} $(--)$	2.220×10^{-2} $(--)$	7.017×10^{-1} $(--)$	1.343×10^{-3} $(--)$
80	80	1.155×10^{-2} $(\ 1.79\ )$	6.486×10^{-3} $(\ 1.85\ )$	2.134×10^{-1} $(\ 1.81\ )$	4.023×10^{-4} $(\ 1.83\ )$
160	160	3.171×10^{-3} $(\ 1.90\ )$	1.736×10^{-3} $(\ 1.93\ )$	5.795×10^{-2} $(\ 1.92\ )$	1.086×10^{-4} $(\ 1.92\ )$
320	320	8.093×10^{-4} $(\ 1.98\ )$	4.406×10^{-4} $(\ 1.99\ )$	1.475×10^{-2} $(\ 1.98\ )$	2.761×10^{-5} $(\ 1.98\ )$

Table 4.2: Results for the second test case

two ends ($Z = \pm 1$) are taken from the linear solutions. We advance the solution until $\tau = 0.4$, using fixed 32 points in the X-direction, $2J + 1$ points in the Z-direction and N time steps. The results are shown in Table 4.2, where the quantities E and R are defined as before. We observe bigger numerical errors than those in the first test case. The reasons are: (1) We don't have the exact solution in this case and we are using an approximate solution instead as the reference solution; (2) The viscosities are much smaller (or in other words, the Reynolds numbers are much bigger), so that higher resolution is required to achieve good accuracy. Nevertheless, the results in the table clearly indicate that second-order convergence is approached as the resolution is refined.

Further evidence of the accuracy and reliability of the numerical code is provided by the remarkable pattern in the simulation of viscous effects on the motion of Stokes waves.

4.2 Numerical simulation of viscous Stokes waves

A large body of research has been conducted on steady progressive waves (Stokes waves) [59][60][63][65]. Certainly, such waves, no matter on a free surface or at an interface, can only exist for inviscid fluids and so we call them inviscid Stokes waves. We ask: What happens if we start with an inviscid Stokes wave and then turn on the viscosity? It can be expected that the wave will decay due to viscous effects. A more delicate question is: In what pattern does the viscosity damp the wave? The results reported in this section will try to answer that question. For convenience, we

will call such waves viscous Stokes waves. In what follows we neglect the effects of the surface tension.

We consider a two-fluid system in a frame moving with the phase speed c and use the expansion formula from the paper of Tsuji and Nagata [65] to obtain initial conditions. The wave profile h can be expanded in a dimensionless form by a Fourier cosine series

$$h = \sum_{k=1}^{\infty} A_k(A) \cos kx \ . \tag{4.9}$$

The first five Fourier coefficients are:

$$\begin{aligned}
A_1 &= A \ , \\
A_2 &= \tfrac{1}{2} \tfrac{\rho^{(2)}-\rho^{(1)}}{\rho^{(2)}+\rho^{(1)}} \left(1 + \tfrac{17(\rho^{(2)})^2-38\rho^{(2)}\rho^{(1)}+17(\rho^{(1)})^2}{12(\rho^{(2)}+\rho^{(1)})^2} A^2 \right) A^2 \ , \\
A_3 &= \tfrac{3(\rho^{(2)})^2-10\rho^{(2)}\rho^{(1)}+3(\rho^{(1)})^2}{8(\rho^{(2)}+\rho^{(1)})^2} A^3 + \\
&\quad \tfrac{459(\rho^{(2)})^4-2468(\rho^{(2)})^3\rho^{(1)}+4130(\rho^{(2)})^2(\rho^{(1)})^2-2468\rho^{(2)}(\rho^{(1)})^3+459(\rho^{(1)})^4}{384(\rho^{(2)}+\rho^{(1)})^4} A^5 \ , \\
A_4 &= \tfrac{(\rho^{(2)}-\rho^{(1)})((\rho^{(2)})^2-6\rho^{(2)}\rho^{(1)}+(\rho^{(1)})^2)}{3(\rho^{(2)}+\rho^{(1)})^3} A^4 \ , \\
A_5 &= \tfrac{125(\rho^{(2)})^4-1516(\rho^{(2)})^3\rho^{(1)}+3118(\rho^{(2)})^2(\rho^{(1)})^2-1516\rho^{(2)}(\rho^{(1)})^3+125(\rho^{(1)})^4}{384(\rho^{(2)}+\rho^{(1)})^4} A^5 \ ,
\end{aligned} \tag{4.10}$$

and the phase speed c is given by

$$c^2 = \tfrac{\rho^{(2)}-\rho^{(1)}}{\rho^{(2)}+\rho^{(1)}} \left(1 + \tfrac{(\rho^{(2)})^2+(\rho^{(1)})^2}{(\rho^{(2)}+\rho^{(1)})^2} A^2 + \tfrac{(\rho^{(2)}-\rho^{(1)})^2(5(\rho^{(2)})^2-14\rho^{(2)}\rho^{(1)}+5(\rho^{(1)})^2)}{4(\rho^{(2)}+\rho^{(1)})^4} A^4 \right) . \tag{4.11}$$

Tsuji and Nagata were able to give the explicit expressions for the Fourier coefficients in the series expansions of the stream functions for both the fluids up to the fifth order. Consequently the velocities and the pressures can be easily calculated from the stream functions. These solutions are used as the initial values in our codes. Then we will turn on the viscosities for both the fluids and start the computation.

In our numerical simulations we set the reference frame to be moving at the inviscid phase speed c so that the wave is nearly stationary in the horizontal direction except for a very small phase shift due to viscous contribution. What we are most interested in is the wave motion in the vertical direction, i.e., the change of the amplitude, due to viscous effects. We consider three choices for the amplitude parameter A by using

(1) a small value $A = 0.01$;

(2) a moderate value $A = 0.1$;

(3) a relatively big value $A = 0.2$.

We also consider two choices for the viscosities:

(1) The typical air-water case with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$, or in nondimensional units, $\bar{\rho} = 0.0012$, $\bar{\mu} = 13.636$, $R_e^{(1)} \doteq 208.70$, $R_e^{(2)} \doteq 2845.9$.

(2) An artificial case where the densities are the same with, but the viscosities are 10 times bigger than, the air-water case. Specifically, $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$, or in nondimensional units, $\bar{\rho} = 0.0012$, $\bar{\mu} = 13.636$, $R_e^{(1)} \doteq 20.870$, $R_e^{(2)} \doteq 284.59$.

The spacial domain of computation is the same as defined in (3.71) with H to be big enough so that it reasonably represents two layers of infinite thickness, which is the case considered in the paper of Tsuji and Nagata. We perform the computation on a 2.4GHz Xeon dual-processor workstation from $\tau = 0$ until $\tau = 20T$, where T is one wave period. Figures 4.1 and 4.2 show the wave profiles at $\tau = 0$ and $\tau = 20T$ for $A = 0.01$, 0.1 and the two choices of the viscosites, respectively. One can clearly

see the decay of the wave amplitude. In the case of bigger viscosities (Figure 4.2) the wave decays faster than in the air-water case (Figure 4.1). Moreover with the same viscosities the decay rate is approximately the same for the two choices of A. Clearly each Fourier mode of the wave, A_k, is dependent on the temporal variable τ and we use $A_k(\tau)$ to indicate such a dependence. To do a quantitative study, we define the decay rate, $\widehat{\sigma}(k)$, for each mode A_k through the relation

$$A_k(\tau) = A_k(0)\, e^{\,\widehat{\sigma}(k)\,\tau}\ . \tag{4.12}$$

The value of $\widehat{\sigma}(k)$ is numerically calculated by

$$\widehat{\sigma}(k) = \frac{\mathrm{Ln}(A_k(20T)) - \mathrm{Ln}(A_k(0))}{20T}\ , \tag{4.13}$$

where Ln is the natural logarithm function. We calculate the decay rate for each of the first five Fourier modes and compare that with the value in the purely linear case, i.e., the real part of $\sigma(k)$ as determined by equation (4.5). The comparison is made in dimensionless units and the results are shown in Tables 4.3 and 4.4, respectively.

We see that the decay rate for the mode A_1 is close to that in the linear case, the smaller the A, the smaller the difference. Even when $A = 0.2$, the decay rate is only about 10% different from the linear prediction. But the most notable feature of the pattern in decay rates is that the decay rate $\widehat{\sigma}(k)$ for the kth mode is approximately $\widehat{\sigma}(k) = k\,\widehat{\sigma}(1)$, at least for $k = 1, 2, \cdots, 5$. These values are distinct from the linear predictions and suggest that nonlinear interactions remain important during the viscous damping of the wave. There is a single disagreement with this pattern in Table 4.3, the decay rate of the fourth mode when $A = 0.2$. There are several

Mode	Linear case	$A = 0.01$	$A = 0.1$	$A = 0.2$
A_1	-8.03×10^{-4}	-8.05×10^{-4}	-8.15×10^{-4}	-9.33×10^{-4}
A_2	-3.00×10^{-3}	-1.55×10^{-3}	-1.66×10^{-3}	-1.82×10^{-3}
A_3	-6.53×10^{-3}	-2.27×10^{-3}	-2.51×10^{-3}	-2.80×10^{-3}
A_4	-1.14×10^{-2}	-2.99×10^{-3}	-2.99×10^{-3}	-2.83×10^{-3}
A_5	-1.75×10^{-2}	-3.71×10^{-3}	-3.70×10^{-3}	-4.48×10^{-3}

Table 4.3: Decay rates in the air-water case

possibilities for this discrepancy, but we will delay discussion of it until after we view the results from a different perspective.

Tables 4.3 and 4.4 give the average decay rate over a time interval. We now want to study the decay pattern for each mode in detail. From the expansion formula (4.10) we know the analytic relationship between these modes. The complex version of (4.9) predicts that all the complex Fourier coefficients are purely real. However, the numerical calculations for the viscous Stokes waves generate complex Fourier coefficients with both real and imaginary parts. Thus the magnitude $|A_k|$ and the phase ϕ_k will be studied. For the inviscid case, the results in (4.10) suggest one way to view the family of Stokes waves is to consider the curves $A_k(A_1)$, or equivalently, $|A_k|(|A_1|)$. Then the effects of viscosity can be studied by viewing the deviation of the numerical results from these curves.

We draw the curves by using (4.10) for the modes $|A_2|$ versus $|A_1|$, $|A_3|$ versus $|A_1|$, $|A_4|$ versus $|A_1|$, $|A_5|$ versus $|A_1|$, etc., and refer to these curves as inviscid

Mode	Linear case	$A = 0.01$	$A = 0.1$	$A = 0.2$
A_1	-7.04×10^{-3}	-7.05×10^{-3}	-7.09×10^{-3}	-7.15×10^{-3}
A_2	-2.68×10^{-2}	-1.40×10^{-2}	-1.43×10^{-2}	-1.45×10^{-2}
A_3	-5.82×10^{-2}	-2.10×10^{-2}	-2.16×10^{-2}	-2.20×10^{-2}
A_4	-1.00×10^{-1}	-2.83×10^{-2}	-2.86×10^{-2}	-2.81×10^{-2}
A_5	-1.52×10^{-1}	-3.57×10^{-2}	-3.59×10^{-2}	-3.52×10^{-2}

Table 4.4: Decay rates in the case with 10 times bigger viscosities

solutions. On the other hand, we have the numerical solutions which give the time evolution for the amplitude of each mode. We can plot these amplitudes in the same way as $|A_2|$ versus $|A_1|$, $|A_3|$ versus $|A_1|$, $|A_4|$ versus $|A_1|$, $|A_5|$ versus $|A_1|$, etc. In Figures 4.3 – 4.8 we compare the numerical solutions for the three choices of the amplitude parameter A and the two choices of the viscosities to the analytic inviscid solutions. The numerical solutions are plotted from $\tau = 0$ and for every period, T, until $\tau = 20T$. Figures 4.3, 4.5, 4.7 give the results in the air-water case for $A = 0.01,\ 0.1,\ 0.2$, respectively. Figures 4.4, 4.6, 4.8 give the results in the case with 10 times bigger viscosities for $A = 0.01,\ 0.1,\ 0.2$, respectively. These results, together with results for the decay rates, suggest a very clear interpretation: viscous effects simply reduce the magnitude of the Stokes wave while allowing it to remain a member of the family. Without viscosity, A is fixed. With viscosity it is reduced while maintaining the ratio of the amplitudes.

The evidence is strongest for $A = 0.01$ and $A = 0.1$. For $A = 0.2$, there is a

deviation in the pattern for A_4 and A_5 but it is confined to the first a few periods of the motion. The reason is that our initial conditions correspond to the inviscid Stokes wave where the tangential velocities are discontinous but the pressure is continuous across the interface. As soon as the computation is started in the presence of viscosity, boundary layers form to ensure the velocities become continuous and the stresses become important in the balance of pressure across the interface. When the wave amplitude is big, like $A = 0.2$, such an adjustment from the inviscid solution to the viscous solution can affect the fourth and fifth digits of the numerical results. Since this spontaneous adjustment is relatively small, it is observed in the fourth and fifth modes where amplitudes are of comparable size to the adjustments. The numerical results show that the deviation in the pattern of amplitudes quickly dies away and the Stokes wave is fully restored, albeit at a smaller amplitude. This is also indicated by the case with bigger viscosities (see Figure 4.8) where the modes decay much faster and the numerical solution and the inviscid solution show pretty good agreement when $\tau \geq 10T$.

One more evidence is provided in Figure 4.9, where we match the numerical solutions of viscous Stokes waves at $\tau = 20T$ by using some analytic solutions from inviscid Stokes waves. The air-water case is considered and two choices for the initial wave amplitude are made: $A = 0.01$ and 0.1. From the numerical solutions we are able to obtain the magnitude of the mode A_1 at $\tau = 20T$ in both cases, which are approximately 0.009038 and 0.09031, respectively. Then we set the amplitude parameter A to be these two numbers, respectively, and substitute A into the expansion (4.9) to obtain an inviscid solution. The numerical solutions and the inviscid

solutions are plotted for both cases in Figure 4.9 and we find excellent agreement between them.

Now that we have thoroughly studied the amplitude of each mode, we turn to investigating its phase. Since the reference frame is moving at the inviscid phase speed c in our numerical simulations, the wave motion is not purely stationary in the horizontal direction – there is a small shift of phase due to the contribution from viscosity. Accordingly, the complex Fourier coefficients of wave modes contain small imaginary parts. Let $A_{k,\,r}$ and $A_{k,\,i}$ denote the real and imaginary parts of A_k, respectively, and P_k the phase shift for mode A_k. Then P_k is determined by $\tan P_k = \dfrac{A_{k,\,i}}{A_{k,\,r}}$.

In Figures 4.10 and 4.11 we plot P_k versus the time t for the first four modes of the Stokes wave in the air-water case with $A = 0.01$. The phase shift for mode A_1 clearly indicates a straight line with respect to time and the slope of the line, $\dfrac{P_1}{t}$, gives the shift of phase speed $\Delta c \doteq 1.23 \times 10^{-4}$. On the other hand, the linear theory predicts the difference between the inviscid phase speed and viscous phase speed is approximately 1.26×10^{-4}, which is very close to the numerical result.

Unfortunately, the numerical solution of the phase shift for modes A_2, A_3 and A_4 shows significant oscillations. The reason for such oscillations is not clearly understood yet. Nevertheless, the trend of the numerical data can be revealed by using the standard linear least square approximations. That means, given a set of points $\{(t^n,\ P_k^n)\}_{n=1}^N$, we seek a linear function $P_k = a_k\, t + b_k$ such that the quantity

$$\sum_{n=1}^{N} \left(a_k\, t^n + b_k - P_k^n\right)^2$$

phase shift	a_k	b_k
P_1	1.23×10^{-4}	-1.01×10^{-5}
P_2	2.50×10^{-4}	-9.39×10^{-3}
P_3	3.30×10^{-4}	-1.67×10^{-2}
P_4	3.62×10^{-4}	-3.45×10^{-2}

Table 4.5: Linear least square approximations for the phase shift

is minimized. The two coefficients a_k and b_k are determined by

$$\begin{aligned} a_k &= \frac{N \sum_{n=1}^{N} t^n P_k^n - \left(\sum_{n=1}^{N} t^n\right)\left(\sum_{n=1}^{N} P_k^n\right)}{N \sum_{n=1}^{N} (t^n)^2 - \left(\sum_{n=1}^{N} t^n\right)^2}, \\ b_k &= \frac{\left(\sum_{n=1}^{N} (t^n)^2\right)\left(\sum_{n=1}^{N} P_k^n\right) - \left(\sum_{n=1}^{N} t^n\right)\left(\sum_{n=1}^{N} t^n P_k^n\right)}{N \sum_{n=1}^{N} (t^n)^2 - \left(\sum_{n=1}^{N} t^n\right)^2}, \end{aligned}$$

and the results for $1 \leq k \leq 4$ are presented in Table 4.5.

One notable feature of the linear least square approximations to the phase shift is that a_2 and a_3 are about two and three times of a_1, respectively. This indicates that the rate of phase shift for mode A_k is k times that for mode A_1. There is a disagreement with this pattern for mode A_4, which is apparently associated with the strong oscillations in the numerical calculation of the phase shift P_4. The case we are considering is for $A = 0.01$, which is close to linear case but the nonlinear interaction between different modes remains important. When the amplitude is big, the values of the parameter A and viscosities are both important in determining the phase speed and the shift of phase can be no longer approximated by a straight line. Nevertheless,

the pattern of the phase shift in Table 4.5 together with that of the decay rate in Tables 4.3 and 4.4 suggest that viscosity maintains the ratio of both the amplitude and the phase shift between each mode and allows a Stokes wave to remain a member of the family.

In conclusion, the Stokes wave appears to be a stably-attracting state. Mathematically, the above observations imply the following interpretation. In the presence of viscosity, if the reference frame is moving with the viscous phase speed $\hat{c}$ instead of the inviscid speed c, the expansion for a Stokes wave may take the form

$$h = \sum_{k=1}^{\infty} A_k \big(A\, f(A, \rho, \nu, t) \big) \cos kx \ . \tag{4.14}$$

That means the parameter A in the expansion (4.9) is now replaced by the product of A and a time-dependent function f. Clearly the function f has the following properties:

(1) $0 \le f \le 1$ and $f = 1$ when $t = 0$;

(2) f is decreasing with respect to t;

(3) The decay rate of f is determined by the densities and the viscosities of the fluids as well as the wave amplitude.

Now suppose the densities of the fluids are constant. Let's also take the ratio of the viscosities, $\dfrac{\nu^{(2)}}{\nu^{(1)}}$, to be fixed. We are particularly interested in the case with small viscosity $\nu^{(1)}$ and small amplitude A. One possible form for the function f is,

$$f = \exp\big(\widehat{\sigma}(1)\, t\big) \ , \tag{4.15}$$

where $\widehat{\sigma}(1)$ has the same meaning as that defined in (4.12), i.e., the nonlinear decay

rate of the first mode A_1. Consequently, from (4.10) we have, $\widehat{\sigma}(2) \doteq 2\widehat{\sigma}(1)$, $\widehat{\sigma}(3) \doteq 3\widehat{\sigma}(1), \cdots$, which agrees with our numerical results presented in Tables 4.3 and 4.4.

One possibility is that $\widehat{\sigma}(1)$ can be expanded in terms of the amplitude parameter A,

$$\widehat{\sigma}(1) = \sigma_0 + \sigma_1 A + \sigma_2 A^2 + \cdots , \tag{4.16}$$

where σ_0 is the linear decay rate, and then each σ_m $(m = 0, 1, \cdots)$ can be expanded in terms of the viscosity $\nu = \nu^{(1)}$,

$$\sigma_m = \sigma_{m,0} + \nu\sigma_{m,1} + \nu^2\sigma_{m,2} + \cdots . \tag{4.17}$$

Meanwhile, the viscous phase speed $\hat{c}$ may be expanded in a similar way as $\widehat{\sigma}(1)$ in terms of A and ν. This possibility must be checked by performing an asymptotic study of the viscous Stokes waves. In Chapter 5 we provide the details of the asymptotic expansions in the linear case. The asymptotic calculation in the nonlinear case will be a topic in our future research.

Finally we plot the vorticity contours in Figure 4.12 for $A = 0.1$ and with the two choices of viscosities. In both cases the vorticity in the upper fluid is dominant and the maximal value of the vorticity occurs near the interface. In the air-water case, there is only a very thin layer of vorticity in the lower fluid. In the other case, the vorticity is much weaker due to the bigger viscosities but extends further into the fluid away from the interface. To have a closer look at the vorticity distribution in the lower fluids, we zoom in the vorticity contours in the lower domains for both cases and present the enlarged pictures in Figure 4.13.

4.3 Numerical simulation of viscous standing waves

The motion of two-dimensional standing waves at a fluid interface is also an attractive topic in fluid mechanics and many studies have been performed in the case of inviscid fluids [51][56][61]. An inviscid standing wave does not propagate but makes periodic oscillations between crest and trough in the vertical direction. It can be expanded in a similar form as in (4.9) but with time-dependent coefficients,

$$h = \sum_{k=1}^{\infty} A_k(A,t)\cos kx \ . \tag{4.18}$$

Based on our results from the viscous Stokes waves, we would expect that a similar decay pattern of amplitude holds for standing waves in the presence of viscosity.

Here we use the fifth-order free-surface wave expansion formula from the paper of Penney and Price [51]. Initially the wave is at its peak,

$$\begin{aligned} A_1 &= A + \frac{1}{32}A^3 - \frac{47}{1344}A^5 \ , \\ A_2 &= \frac{1}{2}A^2 - \frac{79}{672}A^4 \ , \\ A_3 &= \frac{3}{8}A^3 - \frac{12563}{59136}A^5 \ , \\ A_4 &= \frac{1}{3}A^4 \ , \\ A_5 &= \frac{295}{768}A^5 \ . \end{aligned} \tag{4.19}$$

At this moment both fluids are at rest, i.e., the velocities u and w are zero everywhere.

The numerical simulation is performed in a similar way as that for Stokes waves. We use the above initial conditions and turn on the viscosities to start the computation. The numerical solution is recorded at every period T, when the wave attains its

peak, until $\tau = 20T$. The decay pattern for the modes $|A_2|$ versus $|A_1|$, $|A_3|$ versus $|A_1|$, $|A_4|$ versus $|A_1|$ and $|A_5|$ versus $|A_1|$ are plotted in Figures 4.14 and 4.15 for the air-water case and the case with 10 times bigger viscosities, respectively. The amplitude parameter $A = 0.1$ in both cases. We observe that, for the first 3 modes, i.e., for $|A_2|$ versus $|A_1|$ and $|A_3|$ versus $|A_1|$, the numerical solutions (the squares) closely follow the inviscid solutions (the curves). However, for the modes $|A_4|$ versus $|A_1|$ and $|A_5|$ versus $|A_1|$, there is a significant deviation from the inviscid curve (see Figure 4.14). The disagreement is small, about 10^{-5}. There is no improvement with higher numerical resolution. Similar results also hold for smaller or bigger values of A. The situation appears different from the behavior of Stokes waves with bigger amplitude $A = 0.2$. The reason for this disagreement is not clearly understood yet.

4.4 Parallelization

One of the advantages of the numerical method described in Chapter 3 is that it can be easily adapted to parallel computer architectures. The details are presented below.

Suppose the domain of computation is a rectangle as is in (3.71) and there are $2K$ points in the X-direction and $2J+1$ points in the Z-direction. Let M be the number of processors. We use row-wise striped partitioning when updating the solution in time. Each processor except one, to which we refer as processor 0, is assigned $2J/M$ rows. Processor 0, instead, contains one more row which marks the interface. Each

No. of processors n	1	2	4	8	16	32
CPU time $T(n)$ (in seconds)	1658	876	450	220	114	64
Speedup $S(n)$	1.00	1.89	3.68	7.54	14.54	25.91
Efficiency $E(n)$	1.00	0.95	0.92	0.94	0.91	0.81

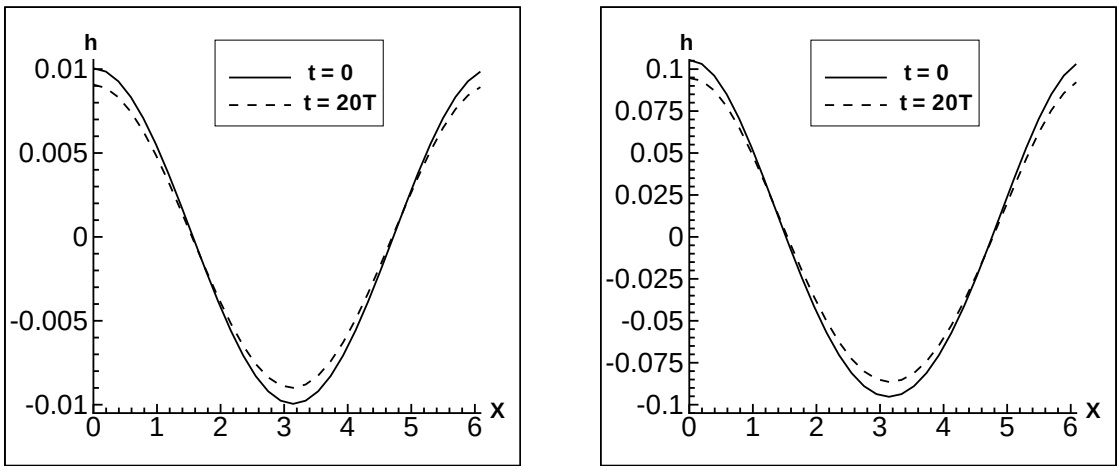
Table 4.6: Performance of parallelization

processor performs the Fourier transform along the X-direction, calculates the right-hand side vectors R_k (see equation (3.72)), and prepares the data for the BVP. Then we switch to column-wise striped partitioning, by way of an all-to-all communication, to solve the BVP (3.85). Each processor now handles $2K/M$ columns and works out the transformed solutions $\widetilde{Y}_k$. Finally, we go back to the row-wise striped partitioning by using the all-to-all communication again and recover the original solutions Y_k. That completes one iteration for marching in time. The biggest overhead in this parallel agorithm is the switches between row-wise and column-wise partitionings at each iteration.

A test of the performance of the parallelization by using MPI is made on an IA-64 Cluster with 900 MHz Itanium-2 processors, for a problem with moderate size: $K = 32$, $J = 1600$ and 400 time steps. The CPU time is compared for different number of processors and in each multi-processor case the CPU time is measured from the beginning of computation until the last processor finishes execution. We use n to denote the number of processors and $T(n)$ the CPU time measured with n

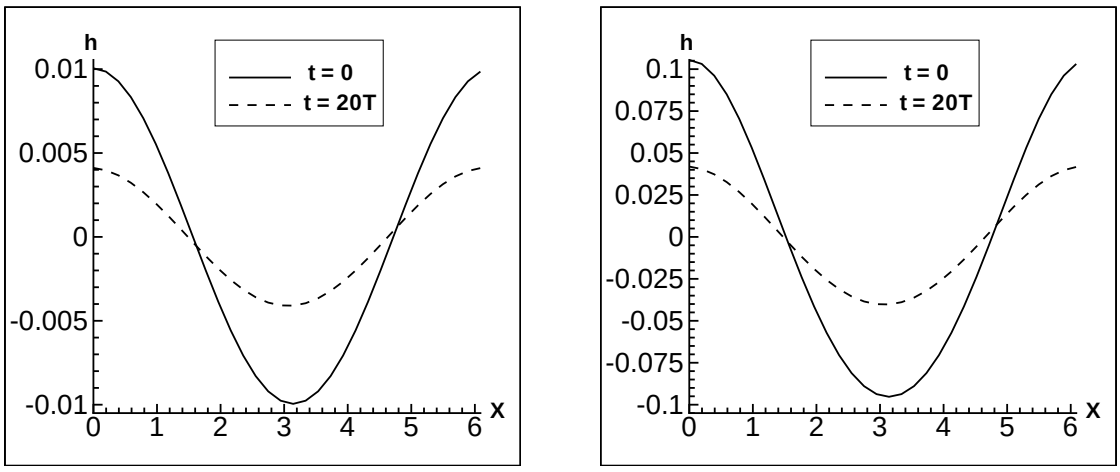
processors. Meanwhile we calculate the speedup $S(n) = T(1)/T(n)$ and the efficiency $E(n) = S(n)/n$. The results are shown in Table 4.6.

The drop in performance for 32 processors is associated with $K = 32$. Communication costs are beginning to be important. We expect performance will improve again when K is much larger.



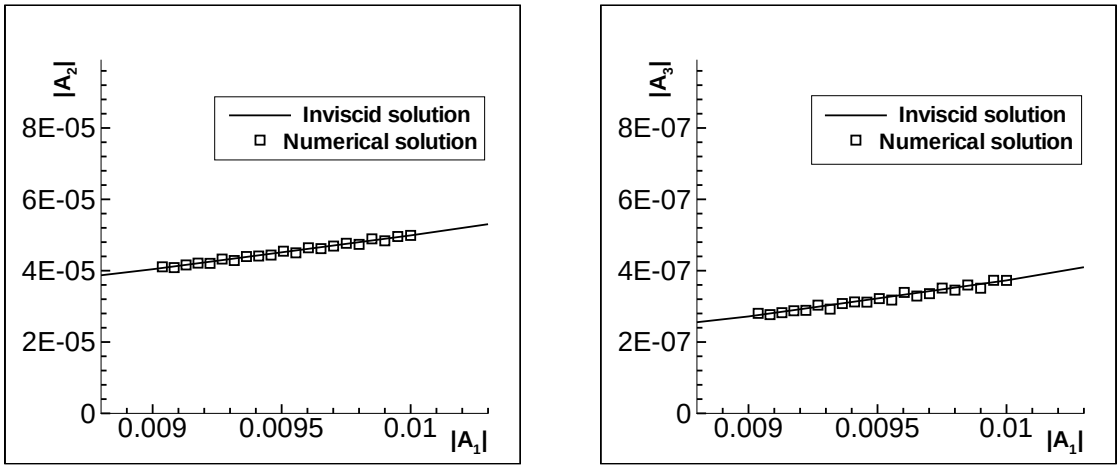
(a) (b)

Figure 4.1: The interface profiles from the numerical simulation of Stokes waves at $t = 0$ and $t = 20T$, where T is one wave period, with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8\times10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1\times10^{-2}$ and two choices for the amplitude parameter A: (a) $A = 0.01$; (b) $A = 0.1$.



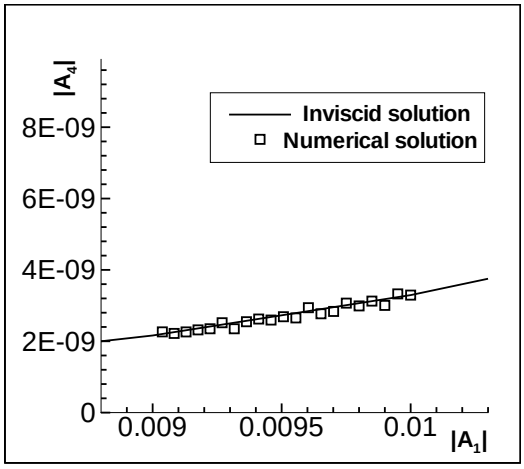
(a) (b)

Figure 4.2: The interface profiles from the numerical simulation of Stokes waves at $t = 0$ and $t = 20T$, where T is one wave period, with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1E \times 10^{-1}$ and two choices for the amplitude parameter A: (a) $A = 0.01$; (b) $A = 0.1$.

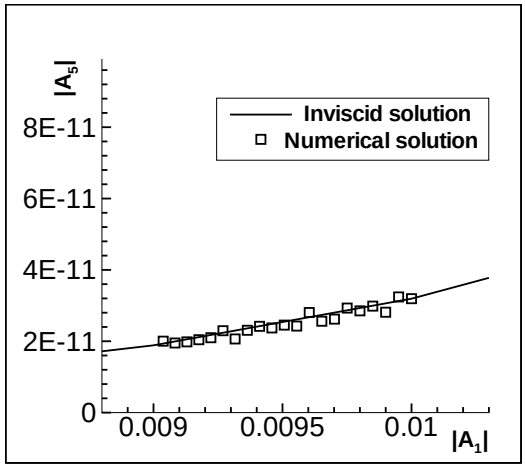


(a)

(b)



(c)



(d)

Figure 4.3: Comparison between the inviscid solution and the numerical solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$ and the amplitude parameter $A = 0.01$. The numerical solution is displayed from $\tau = 0$ and for every period, T, until $\tau = 20T$. (a) modes $|A_2|$ versus $|A_1|$; (b) modes $|A_3|$ versus $|A_1|$; (c) modes $|A_4|$ versus $|A_1|$; (d) modes $|A_5|$ versus $|A_1|$.

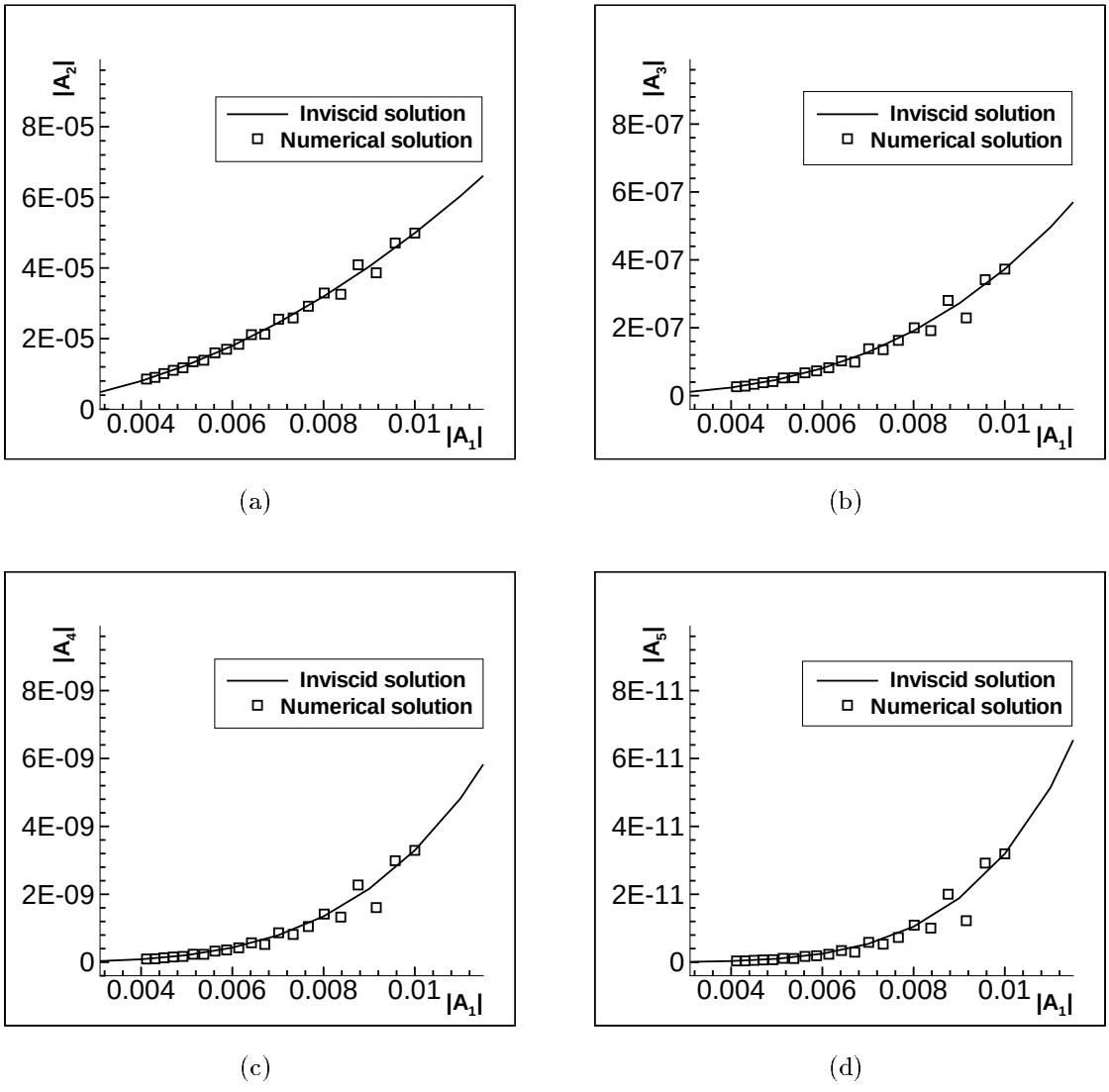


Figure 4.4: Comparison between the inviscid solution and the numerical solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$ and the amplitude parameter $A = 0.01$. The numerical solution is displayed from $\tau = 0$ and for every period, T, until $\tau = 20T$. (a) modes $|A_2|$ versus $|A_1|$; (b) modes $|A_3|$ versus $|A_1|$; (c) modes $|A_4|$ versus $|A_1|$; (d) modes $|A_5|$ versus $|A_1|$.

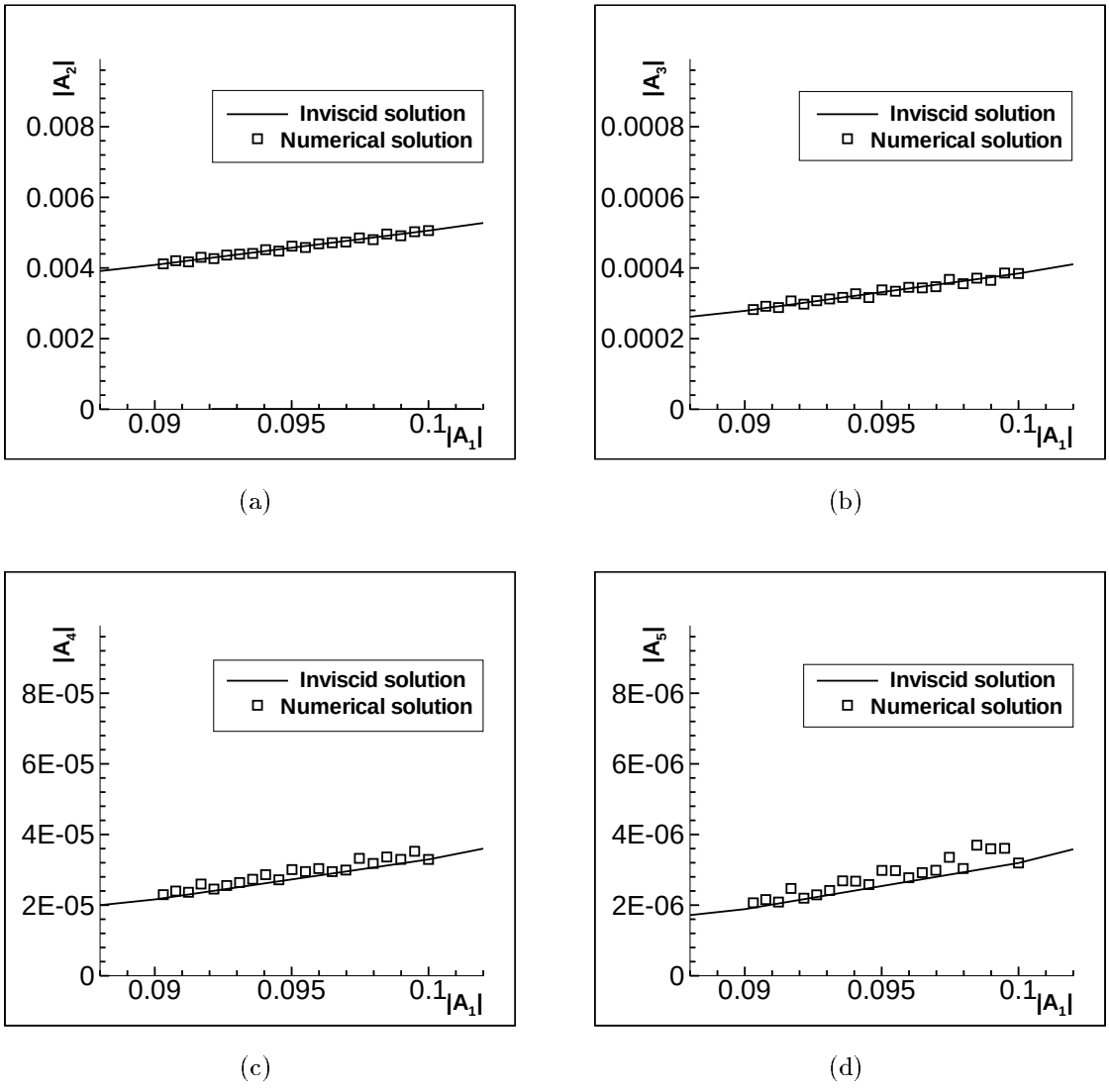


Figure 4.5: Comparison between the inviscid solution and the numerical solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$ and the amplitude parameter $A = 0.1$. The numerical solution is displayed from $\tau = 0$ and for every period, T, until $\tau = 20T$. (a) modes $|A_2|$ versus $|A_1|$; (b) modes $|A_3|$ versus $|A_1|$; (c) modes $|A_4|$ versus $|A_1|$; (d) modes $|A_5|$ versus $|A_1|$.

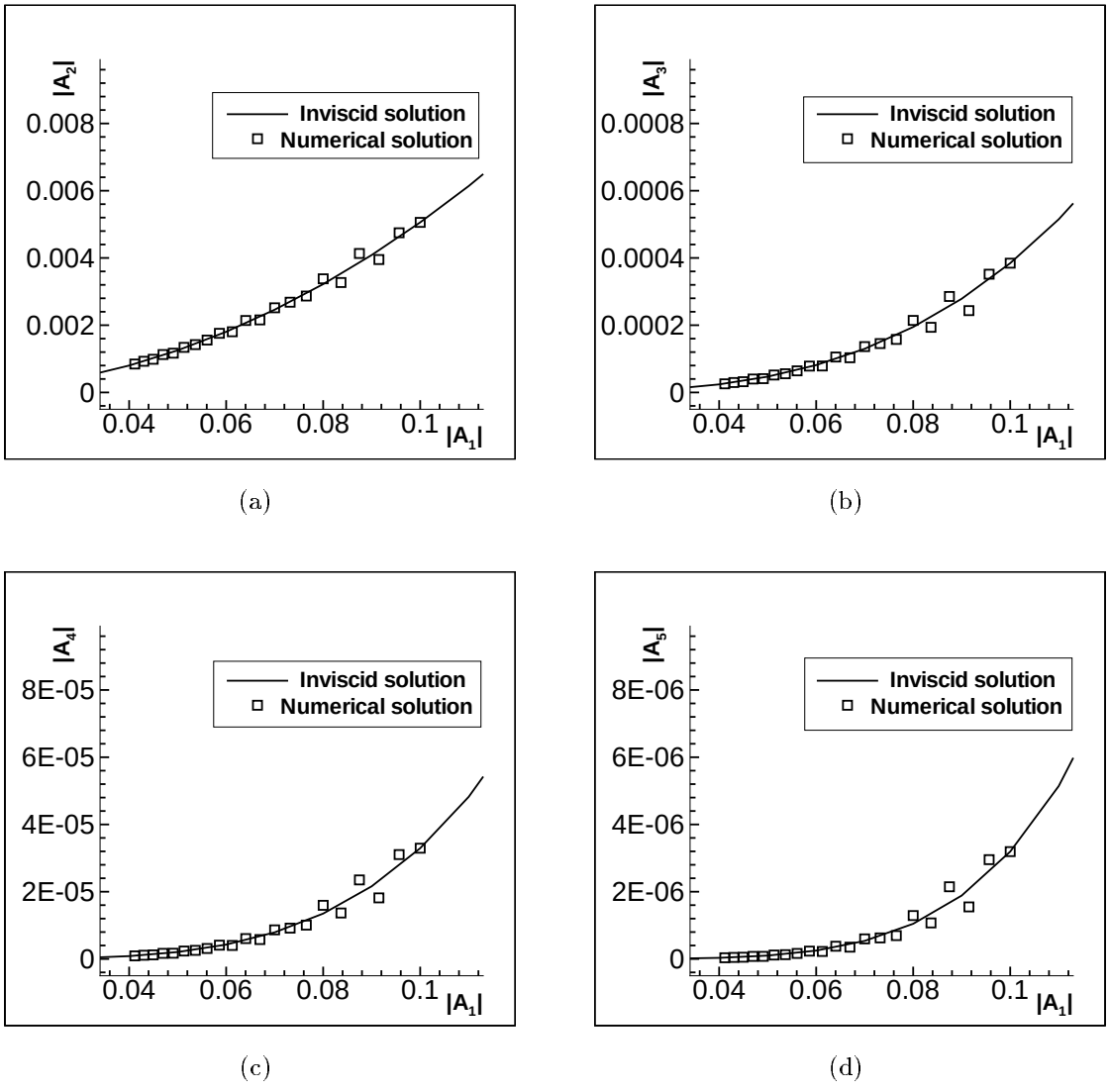


Figure 4.6: Comparison between the inviscid solution and the numerical solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$ and the amplitude parameter $A = 0.1$. The numerical solution is displayed from $\tau = 0$ and for every period, T, until $\tau = 20T$. (a) modes $|A_2|$ versus $|A_1|$; (b) modes $|A_3|$ versus $|A_1|$; (c) modes $|A_4|$ versus $|A_1|$; (d) modes $|A_5|$ versus $|A_1|$.

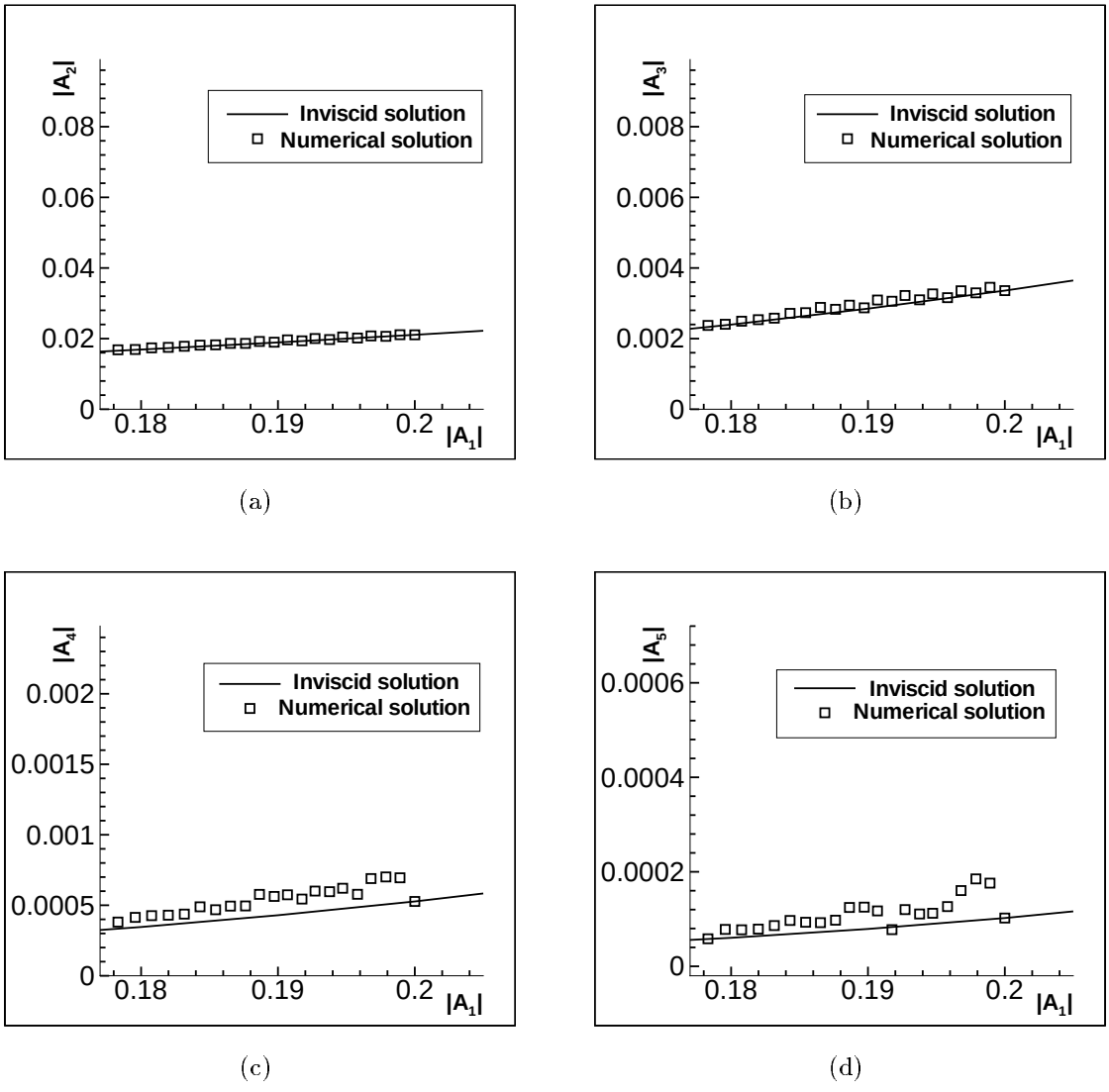


Figure 4.7: Comparison between the inviscid solution and the numerical solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$ and the amplitude parameter $A = 0.2$. The numerical solution is displayed from $\tau = 0$ and for every period, T, until $\tau = 20T$. (a) modes $|A_2|$ versus $|A_1|$; (b) modes $|A_3|$ versus $|A_1|$; (c) modes $|A_4|$ versus $|A_1|$; (d) modes $|A_5|$ versus $|A_1|$.

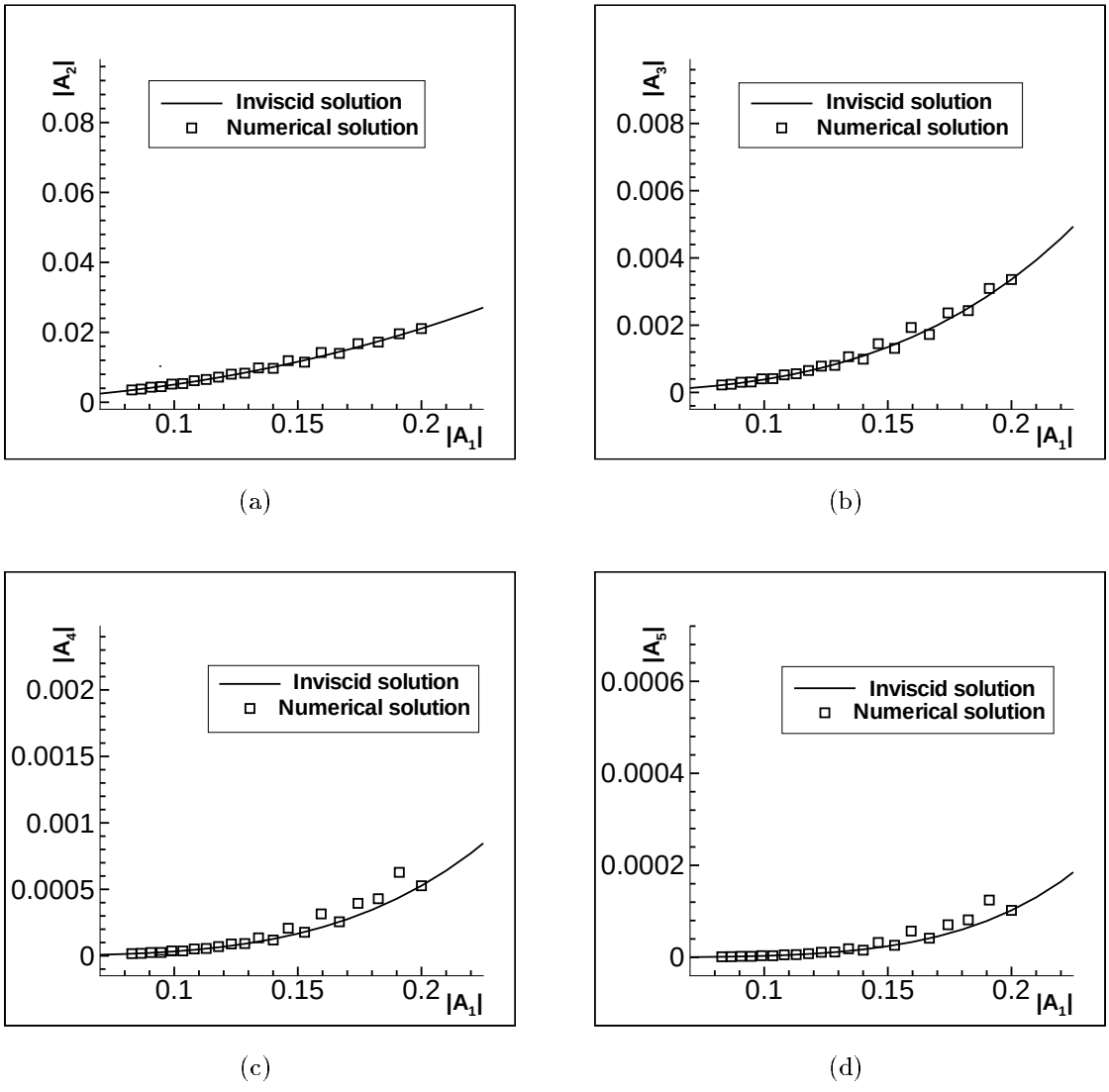
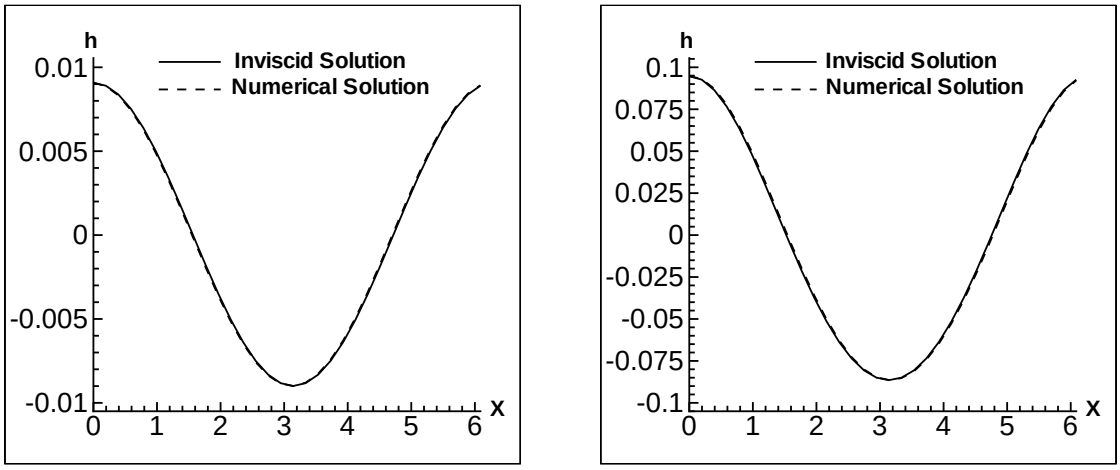
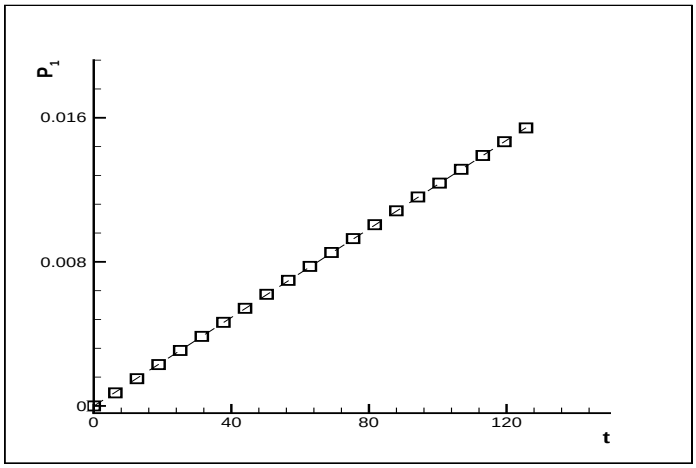


Figure 4.8: Comparison between the inviscid solution and the numerical solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$ and the amplitude parameter $A = 0.2$. The numerical solution is displayed from $\tau = 0$ and for every period, T, until $\tau = 20T$. (a) modes $|A_2|$ versus $|A_1|$; (b) modes $|A_3|$ versus $|A_1|$; (c) modes $|A_4|$ versus $|A_1|$; (d) modes $|A_5|$ versus $|A_1|$.

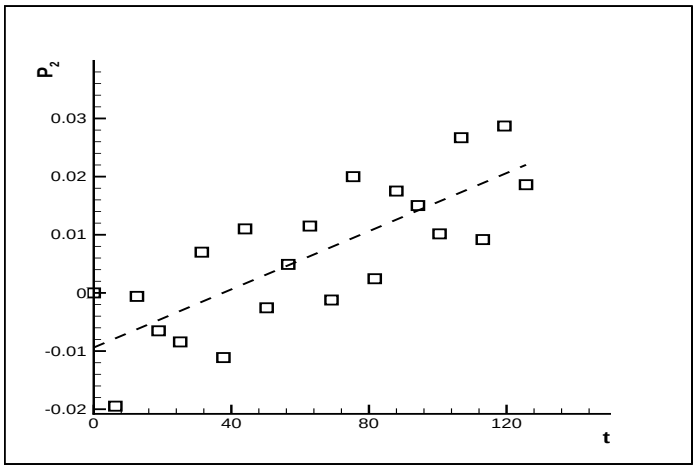


(a) (b)

Figure 4.9: Comparison between the inviscid solution and the numerical solution for the profiles of Stokes waves with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$. (a) The numerical solution starts from $A = 0.01$ and is plotted at $t = 20T$, while the inviscid solution is plotted with $A \doteq 0.009038$. (b) The numerical solution starts from $A = 0.1$ and is plotted at $t = 20T$, while the inviscid solution is plotted with $A \doteq 0.09031$.

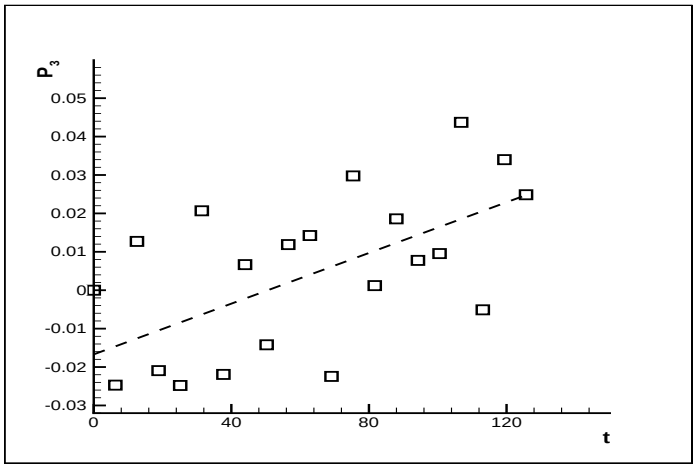


(a)

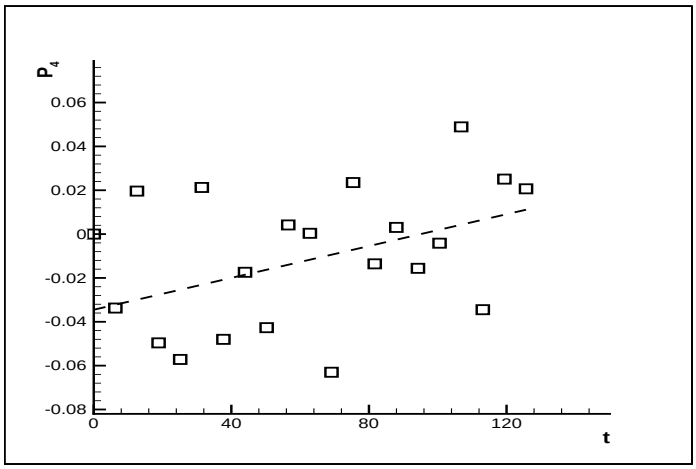


(b)

Figure 4.10: The phase shift in the numerical solution of the Stokes wave with $A = 0.01$ and $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$. (a) Phase shift of mode A_1 versus time; (b) Phase shift of mode A_2 versus time. □ numerical solution; -- linear least square approximation.

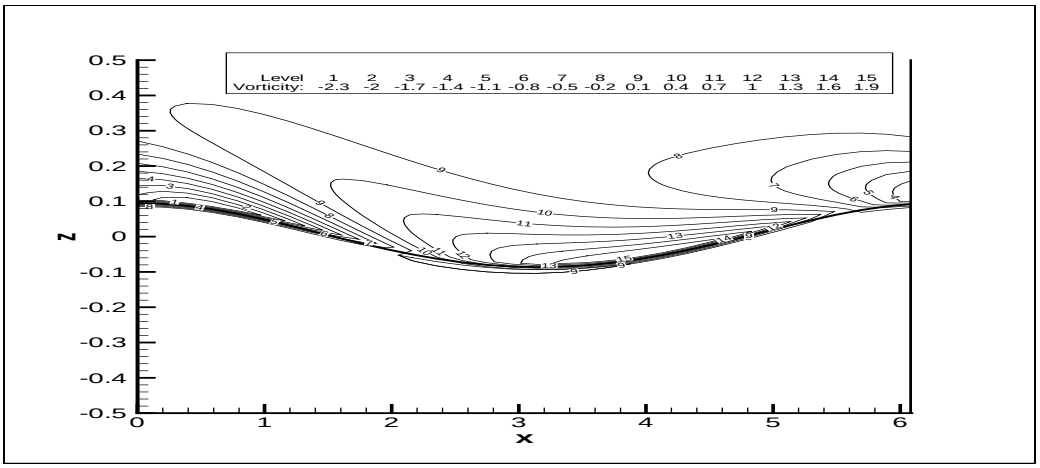


(a)

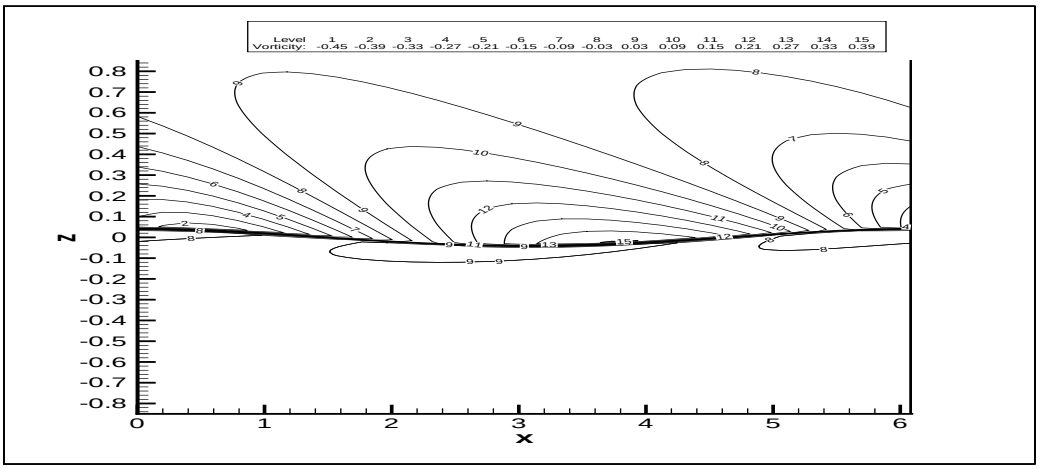


(b)

Figure 4.11: The phase shift in the numerical solution of the Stokes wave with $A = 0.01$ and $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$. (a) Phase shift of mode A_3 versus time; (b) Phase shift of mode A_4 versus time. □ numerical solution; -- linear least square approximation.

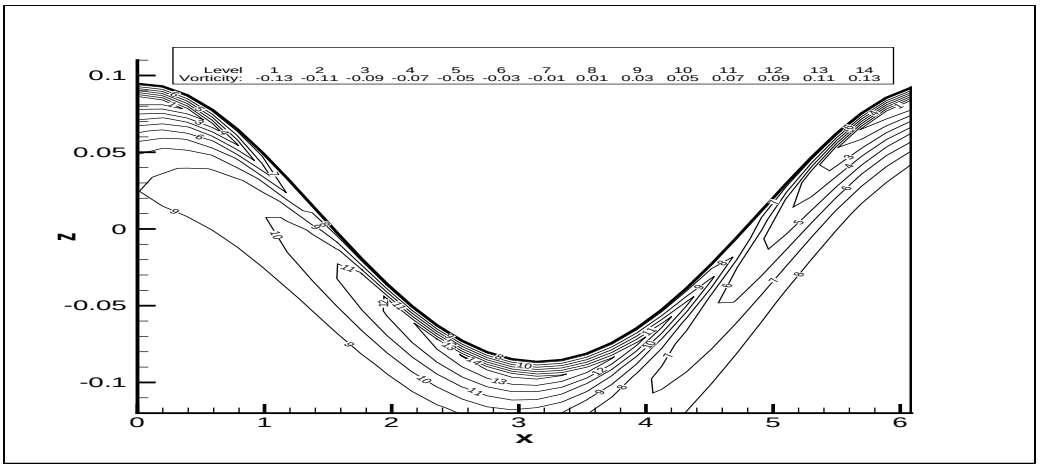


(a)

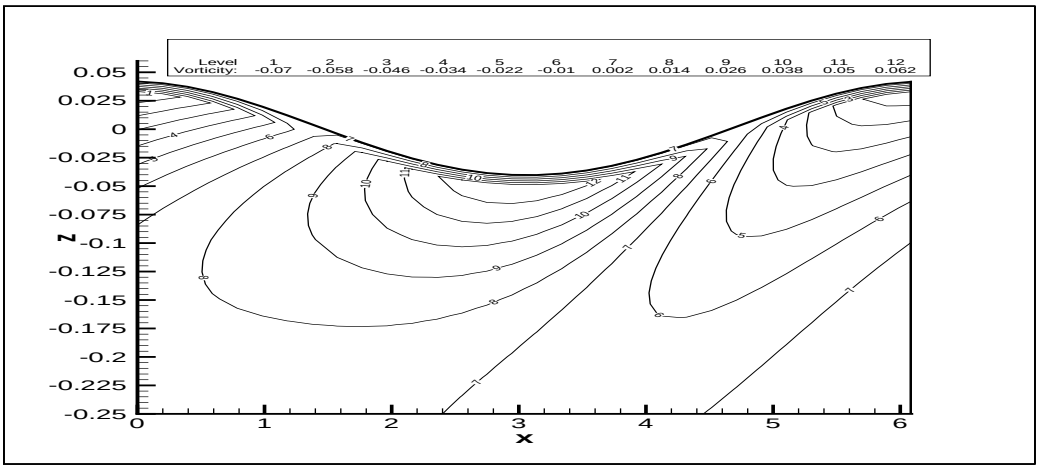


(b)

Figure 4.12: The vorticity contours when the amplitude parameter $A = 0.1$ for the two choices of the viscosities: (a) $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8\times10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$; (b) $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$.



(a)



(b)

Figure 4.13: The vorticity contours in the lower fluid when the amplitude parameter $A = 0.1$ for the two choices of the viscosities: (a) $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$; (b) $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$.

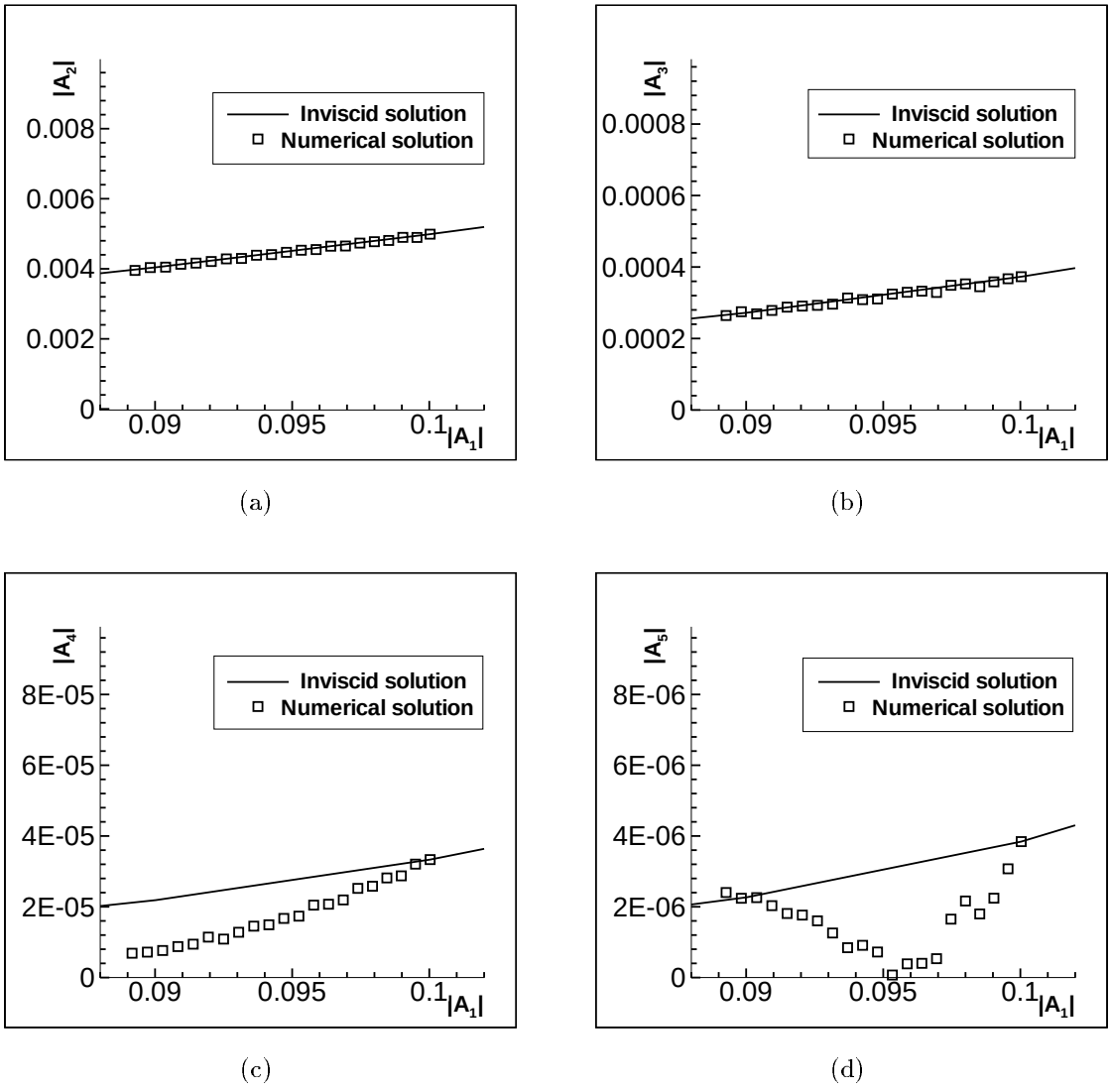


Figure 4.14: Comparison between the inviscid solution and the numerical solution of the standing wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$ and the amplitude parameter $A = 0.1$. The numerical solution is displayed from $\tau = 0$ and for every period, T, until $\tau = 20T$. (a) modes $|A_2|$ versus $|A_1|$; (b) modes $|A_3|$ versus $|A_1|$; (c) modes $|A_4|$ versus $|A_1|$; (d) modes $|A_5|$ versus $|A_1|$.

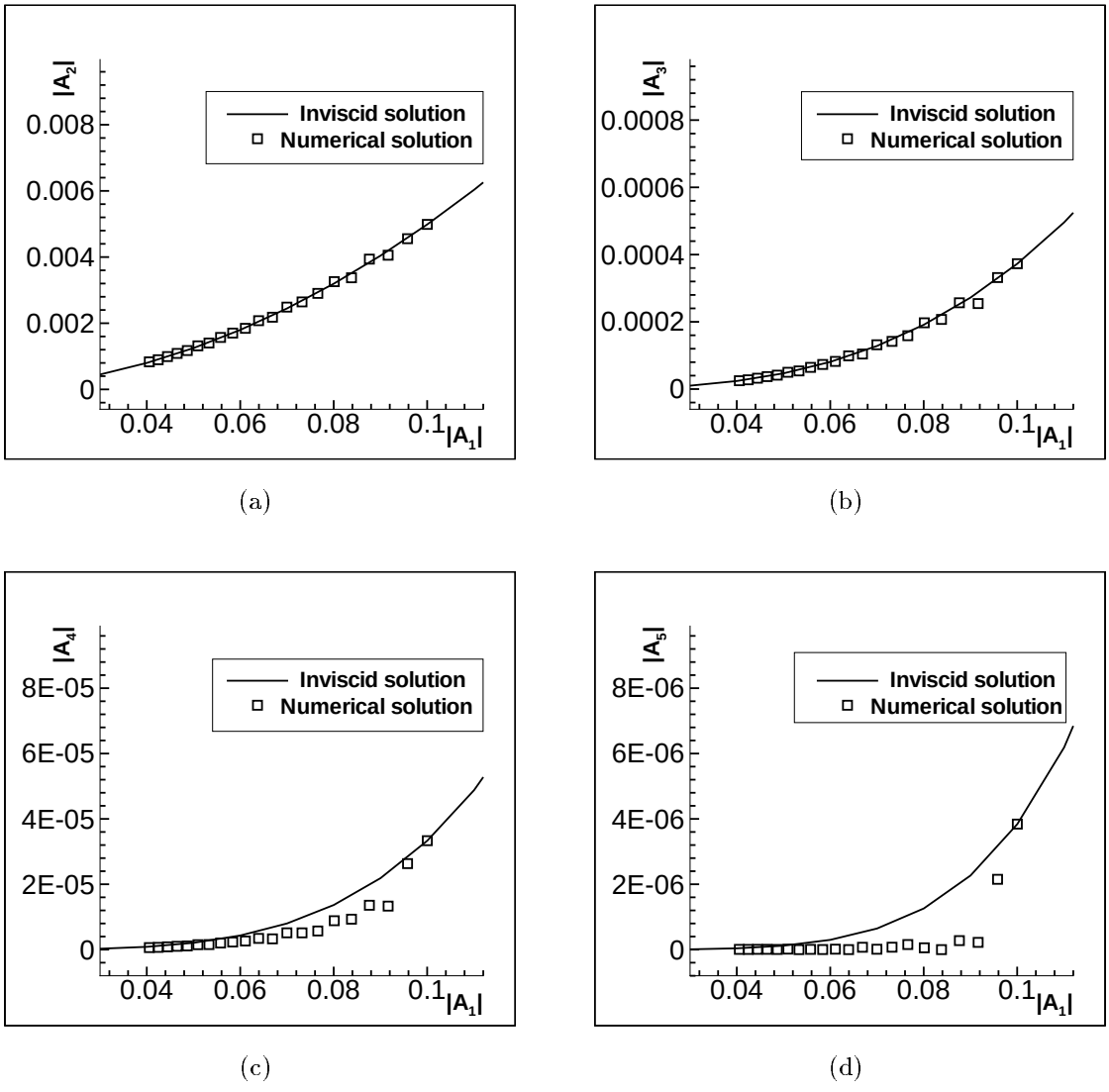


Figure 4.15: Comparison between the inviscid solution and the numerical solution of the standing wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$ and the amplitude parameter $A = 0.1$. The numerical solution is displayed from $\tau = 0$ and for every period, T, until $\tau = 20T$. (a) modes $|A_2|$ versus $|A_1|$; (b) modes $|A_3|$ versus $|A_1|$; (c) modes $|A_4|$ versus $|A_1|$; (d) modes $|A_5|$ versus $|A_1|$.

CHAPTER 5

LINEAR ANALYSIS

5.1 Asymptotic study for the linear problem

In this section we derive asymptotic expansions, in terms of small viscosities, for the solutions of the linear Navier-Stokes equations with linear interfacial conditions. Since we are able to work out the analytical solutions for this linear case, the asymptotic solutions provide a good way to understand the influence of viscosity on the motion of a linear (Stokes) wave.

The power of asymptotic methods is that they can be applied to not only linear problems but also many non-linear problems where exact solutions are impossible to obtain. Furthermore, asymptotic method and the numerical method can benefit each other by providing a good check on their accuracy.

5.1.1 Asymptotic expansions

We consider a linear version of the Navier-Stokes equations

$$u_t = -\frac{1}{\rho}P_x + \nu(u_{xx} + u_{zz}) \,, \tag{5.1}$$

$$w_t = -\frac{1}{\rho}P_z + \nu(w_{xx} + w_{zz}) \,, \tag{5.2}$$

$$iku + w_z = 0 \,, \tag{5.3}$$

with the interfacial conditions

$$u^{(1)} = u^{(2)} \,, \tag{5.4}$$

$$h_t = w^{(1)} = w^{(2)} \,, \tag{5.5}$$

$$\rho^{(1)}\nu^{(1)}\big(u_z^{(1)} + w_x^{(1)}\big) = \rho^{(2)}\nu^{(2)}\big(u_z^{(2)} + w_x^{(2)}\big) \,, \tag{5.6}$$

$$(\rho^{(2)} - \rho^{(1)})gh + P^{(1)} - P^{(2)} - 2\big(\rho^{(1)}\nu^{(1)}w_z^{(1)} - \rho^{(2)}\nu^{(2)}w_z^{(2)}\big) = Th_{xx} \,. \tag{5.7}$$

Here and in what follows, we use the superscripts (1) and (2) to distinguish the quantities in the upper and the lower fluids but we shall use them only when it is important to distinguish which fluid is being considered.

We assume solutions in the form

$$\begin{pmatrix} u \\ w \\ P \\ h \end{pmatrix} = e^{ikx}\, e^{\sigma t} \begin{pmatrix} \mathcal{U} \\ \mathcal{W} \\ \mathcal{P} \\ a \end{pmatrix} \tag{5.8}$$

where $k > 0$ and where a is a fixed number which measures the initial amplitude of the interface. Our goal is then to derive asymptotic expansions for $\mathcal{U}$, $\mathcal{W}$, $\mathcal{P}$ and σ in terms of small viscosity ν.

By substituting (5.8) into the equations (5.1)-(5.7), we obtain

$$\begin{aligned} \sigma\mathcal{U} &= -\frac{ik}{\rho}\mathcal{P} + \nu(-k^2\mathcal{U} + \mathcal{U}_{zz}) \,, && (5.9)\\ \sigma\mathcal{W} &= -\frac{1}{\rho}\mathcal{P}_z + \nu(-k^2\mathcal{W} + \mathcal{W}_{zz}) \,, && (5.10)\\ & ik\mathcal{U} + \mathcal{W}_z = 0 \,, && (5.11) \end{aligned}$$

and

$$\begin{aligned} \mathcal{U}^{(1)} &= \mathcal{U}^{(2)} \,, && (5.12)\\ a\,\sigma = \mathcal{W}^{(1)} &= \mathcal{W}^{(2)} \,, && (5.13)\\ \rho^{(1)}\nu^{(1)}\big(\mathcal{U}_z^{(1)} + ik\mathcal{W}^{(1)}\big) &= \rho^{(2)}\nu^{(2)}\big(\mathcal{U}_z^{(2)} + ik\mathcal{W}^{(2)}\big) \,, && (5.14)\\ (\rho^{(2)} - \rho^{(1)})ga + \mathcal{P}^{(1)} - \mathcal{P}^{(2)} &- 2\big(\rho^{(1)}\nu^{(1)}\mathcal{W}_z^{(1)} - \rho^{(2)}\nu^{(2)}\mathcal{W}_z^{(2)}\big) \\ &= -k^2 T a \,. && (5.15) \end{aligned}$$

We introduce the dimensionless parameters

$$r = \frac{\rho^{(1)}}{\rho^{(2)}} \,, \qquad R = \sqrt{\frac{\nu^{(2)}}{\nu^{(1)}}} \,, \tag{5.16}$$

and keep r and R fixed. We will use the method of multiple scales [34][44] to derive the asymptotic expansions. In considering the boundary layers near the interface which have thickness proportional to $\sqrt{\nu}$, we introduce scaled vertical coordinates η_0 , η_1 as

$$\eta_0 = \frac{z}{\sqrt{\nu}} \,, \qquad \eta_1 = \sqrt{\nu}\,\eta_0 = z \,. \tag{5.17}$$

Consequently,

$$\begin{aligned} \frac{\partial}{\partial z} &= \frac{1}{\sqrt{\nu}}\frac{\partial}{\partial \eta_0} + \frac{\partial}{\partial \eta_1} \,, \\ \frac{\partial^2}{\partial z^2} &= \frac{1}{\nu}\frac{\partial^2}{\partial \eta_0^2} + \frac{2}{\sqrt{\nu}}\frac{\partial^2}{\partial \eta_0\,\partial \eta_1} + \frac{\partial^2}{\partial \eta_1^2} \,. && (5.18) \end{aligned}$$

Then we assume the following expansions:

$$\begin{aligned}
\mathcal{U} &= u_0(\eta_0\,,\eta_1) + \sqrt{\nu}\,u_1(\eta_0\,,\eta_1) + \nu\,u_2(\eta_0\,,\eta_1) + \cdots\,, \\
\mathcal{W} &= w_0(\eta_0\,,\eta_1) + \sqrt{\nu}\,w_1(\eta_0\,,\eta_1) + \nu\,w_2(\eta_0\,,\eta_1) + \cdots\,, \\
\mathcal{P} &= P_0(\eta_0\,,\eta_1) + \sqrt{\nu}\,P_1(\eta_0\,,\eta_1) + \nu\,P_2(\eta_0\,,\eta_1) + \cdots\,, \\
\sigma &= \sigma_0 + \sqrt{\nu}\,\sigma_1 + \nu\,\sigma_2 + \cdots\,. \qquad (5.19)
\end{aligned}$$

Note that σ is the same in the upper and the lower fluids but with different expansions. They are related by

$$\sigma_0^{(1)} = \sigma_0^{(2)}\,; \qquad \sigma_m^{(1)} = R^m\,\sigma_m^{(2)}\,, \quad m = 1, 2, \cdots\,. \qquad (5.20)$$

By substituting (5.19) into (5.9), we obtain

$$\begin{aligned}
&(\sigma_0 + \sqrt{\nu}\,\sigma_1 + \nu\,\sigma_2 + \cdots)\,u_0 + \sqrt{\nu}\,(\sigma_0 + \sqrt{\nu}\,\sigma_1 + \nu\,\sigma_2 + \cdots)\,u_1 + \\
&\nu\,(\sigma_0 + \sqrt{\nu}\,\sigma_1 + \nu\,\sigma_2 + \cdots)\,u_2 + \cdots = -\frac{ik}{\rho}\,P_0 - \sqrt{\nu}\,\frac{ik}{\rho}\,P_1 - \nu\,\frac{ik}{\rho}\,P_2 + \cdots \\
&+\nu\,\Big(-k^2u_0 - \sqrt{\nu}\,k^2u_1 - \nu\,k^2u_2 - \cdots + \frac{1}{\nu}\,\frac{\partial^2 u_0}{\partial\eta_0^2} + \frac{2}{\sqrt{\nu}}\,\frac{\partial^2 u_0}{\partial\eta_0\,\partial\eta_1} + \frac{\partial^2 u_0}{\partial\eta_1^2} + \\
&\frac{1}{\sqrt{\nu}}\,\frac{\partial^2 u_1}{\partial\eta_0^2} + 2\frac{\partial^2 u_1}{\partial\eta_0\,\partial\eta_1} + \sqrt{\nu}\,\frac{\partial^2 u_1}{\partial\eta_1^2} + \frac{\partial^2 u_2}{\partial\eta_0^2} + 2\sqrt{\nu}\,\frac{\partial^2 u_2}{\partial\eta_0\,\partial\eta_1} + \nu\,\frac{\partial^2 u_2}{\partial\eta_1^2} + \cdots\Big)\,. \qquad (5.21)
\end{aligned}$$

Comparison of the coefficients of ν^n yields,

$$\text{order } \nu^0\ : \qquad \sigma_0\,u_0 = -\frac{ik}{\rho}\,P_0 + \frac{\partial^2 u_0}{\partial\eta_0^2}\,, \qquad (5.22)$$

$$\text{order } \nu^{\frac{1}{2}}\ : \qquad \sigma_1\,u_0 + \sigma_0\,u_1 = -\frac{ik}{\rho}\,P_1 + 2\frac{\partial^2 u_0}{\partial\eta_0\,\partial\eta_1} + \frac{\partial^2 u_1}{\partial\eta_0^2}\,, \qquad (5.23)$$

$$\begin{aligned}
\text{order } \nu^1\ : \qquad &\sigma_2\,u_0 + \sigma_1\,u_1 + \sigma_0\,u_2 = -\frac{ik}{\rho}\,P_2 - k^2u_0 \\
&+\frac{\partial^2 u_0}{\partial\eta_1^2} + 2\frac{\partial^2 u_1}{\partial\eta_0\,\partial\eta_1} + \frac{\partial^2 u_2}{\partial\eta_0^2}\,. \qquad (5.24)
\end{aligned}$$

If we substitute (5.19) into (5.10) and equate the coefficients of like powers of ν^n, we obtain,

$$\text{order } \nu^{-\frac{1}{2}} : \quad \frac{\partial P_0}{\partial \eta_0} = 0 \,, \tag{5.25}$$

$$\text{order } \nu^{0} : \quad \sigma_0 \, w_0 = -\frac{1}{\rho} \frac{\partial P_0}{\partial \eta_1} - \frac{1}{\rho} \frac{\partial P_1}{\partial \eta_0} + \frac{\partial^2 w_0}{\partial \eta_0^2} \,, \tag{5.26}$$

$$\text{order } \nu^{\frac{1}{2}} : \quad \sigma_1 \, w_0 + \sigma_0 \, w_1 = -\frac{1}{\rho} \frac{\partial P_1}{\partial \eta_1} - \frac{1}{\rho} \frac{\partial P_2}{\partial \eta_0} + 2 \frac{\partial^2 w_0}{\partial \eta_0 \, \partial \eta_1} + \frac{\partial^2 w_1}{\partial \eta_0^2} \,. \tag{5.27}$$

Similarly, the substitution of (5.19) into (5.11) yields,

$$\text{order } \nu^{-\frac{1}{2}} : \quad \frac{\partial w_0}{\partial \eta_0} = 0 \,, \tag{5.28}$$

$$\text{order } \nu^{0} : \quad iku_0 + \frac{\partial w_0}{\partial \eta_1} + \frac{\partial w_1}{\partial \eta_0} = 0 \,, \tag{5.29}$$

$$\text{order } \nu^{\frac{1}{2}} : \quad iku_1 + \frac{\partial w_1}{\partial \eta_1} + \frac{\partial w_2}{\partial \eta_0} = 0 \,. \tag{5.30}$$

We also expand the interfacial conditions (5.12)-(5.15). Let $\nu = \nu^{(1)}$. Using (5.16), the substitution of (5.19) and (5.20) into (5.12) yields,

$$\text{order } \nu^{0} : \quad u_0^{(1)} = u_0^{(2)} \,, \tag{5.31}$$

$$\text{order } \nu^{\frac{1}{2}} : \quad u_1^{(1)} = R \, u_1^{(2)} \,. \tag{5.32}$$

Substitution into (5.13) yields,

$$\text{order } \nu^{0} : \quad a \, \sigma_0^{(1)} = w_0^{(1)} = w_0^{(2)} = a \, \sigma_0^{(2)} \,, \tag{5.33}$$

$$\text{order } \nu^{\frac{1}{2}} : \quad a \, \sigma_1^{(1)} = w_1^{(1)} = R \, w_1^{(2)} = aR \, \sigma_1^{(2)} \,. \tag{5.34}$$

Substitution into (5.14) yields,

$$\text{order } \nu^{-\frac{1}{2}} : \quad r \, \frac{\partial u_0^{(1)}}{\partial \eta_0} = R \, \frac{\partial u_0^{(2)}}{\partial \eta_0} \,, \tag{5.35}$$

$$\text{order } \nu^{0} : \quad r \left(\frac{\partial u_0}{\partial \eta_1} + \frac{\partial u_1}{\partial \eta_0} + ikw_0 \right)^{(1)} = R^2 \left(\frac{\partial u_0}{\partial \eta_1} + \frac{\partial u_1}{\partial \eta_0} + ikw_0 \right)^{(2)} \,. \tag{5.36}$$

Finally, substitution into (5.15) yields,

$$\text{order } \nu^0 \ : \qquad (\rho^{(2)} - \rho^{(1)})ga + P_0^{(1)} - P_0^{(2)} = -k^2 T a \ , \tag{5.37}$$

$$\text{order } \nu^{\frac{1}{2}} \ : \qquad P_1^{(1)} - R\, P_1^{(2)} - 2\Big(\rho^{(1)}\, \frac{\partial w_0^{(1)}}{\partial \eta_0} - R\rho^{(2)}\, \frac{\partial w_0^{(2)}}{\partial \eta_0}\Big) = 0 \ . \tag{5.38}$$

5.1.2 Lowest-order solutions

We start the calculation by seeking solutions at the lowest order, i.e., the solutions to u_0, w_0, P_0 and σ_0. Application of secularity conditions in the higher order equations will then be used to determine the additional dependency of the solutions on the scaled variables η_0, η_1. The interfacial conditions at the lowest orders are applied to determine the coefficients in the solutions. Details are as follows.

Equation (5.25) implies that P_0 is independent of η_0, i.e.,

$$P_0 = P_0(\eta_1) \ . \tag{5.39}$$

Equation (5.28) implies that

$$w_0 = w_0(\eta_1) \ . \tag{5.40}$$

From (5.22) we obtain

$$\big(\frac{\partial^2}{\partial \eta_0^2} - \sigma_0\big)\, u_0 = \frac{ik}{\rho}\, P_0(\eta_1) \ . \tag{5.41}$$

Since we want exponentially decaying solutions, (5.41) yields

$$u_0 = \begin{cases} B_1(\eta_1)\, e^{-\sqrt{\sigma_0}\,\eta_0} \ - \ \dfrac{ik}{\rho\,\sigma_0}\, P_0(\eta_1) \quad \text{for} \quad z \geq 0 \ , \\ B_2(\eta_1)\, e^{\sqrt{\sigma_0}\,\eta_0} \ - \ \dfrac{ik}{\rho\,\sigma_0}\, P_0(\eta_1) \quad \text{for} \quad z \leq 0 \ , \end{cases} \tag{5.42}$$

where $B_1(\eta_1)$ and $B_2(\eta_1)$ are to be determined. The Substitution of (5.42) and (5.40) into (5.29) yields, for $z \geq 0$,

$$ikB_1(\eta_1)\, e^{-\sqrt{\sigma_0}\,\eta_0} + \frac{k^2}{\rho\,\sigma_0}\, P_0(\eta_1) + \frac{\mathrm{d}w_0(\eta_1)}{\mathrm{d}\eta_1} + \frac{\partial w_1}{\partial \eta_0} = 0 \ . \tag{5.43}$$

Elimination of secular terms yields

$$\frac{k^2}{\rho\,\sigma_0}\, P_0(\eta_1) + \frac{\mathrm{d}w_0(\eta_1)}{\mathrm{d}\eta_1} = 0 \ . \tag{5.44}$$

Note that (5.44) also holds for $z \leq 0$. Meanwhile, substitution of (5.39) and (5.40) into (5.26) yields

$$\sigma_0\, w_0(\eta_1) = -\frac{1}{\rho}\, \frac{\mathrm{d}P_0(\eta_1)}{\mathrm{d}\eta_1} - \frac{1}{\rho}\, \frac{\partial P_1}{\partial \eta_0} \ . \tag{5.45}$$

To remove the secularity we require

$$\sigma_0\, w_0(\eta_1) = -\frac{1}{\rho}\, \frac{\mathrm{d}P_0(\eta_1)}{\mathrm{d}\eta_1} \ . \tag{5.46}$$

By substituting (5.46) into (5.44), we obtain

$$\Big(\frac{\mathrm{d}^2}{\mathrm{d}\eta_1^2} - k^2\Big)\, w_0 = 0 \ , \tag{5.47}$$

which implies

$$w_0 = \begin{cases} A_1\, e^{-k\eta_1} & \text{for} \quad z \geq 0 \ , \\ A_2\, e^{k\eta_1} & \text{for} \quad z \leq 0 \ , \end{cases} \tag{5.48}$$

where A_1 , A_2 are constants. Equations (5.44) and (5.48) show that

$$P_0 = -\frac{\rho\,\sigma_0}{k^2}\, \frac{\mathrm{d}w_0(\eta_1)}{\mathrm{d}\eta_1} = \begin{cases} \dfrac{\rho^{(1)}\,\sigma_0}{k}\, A_1\, e^{-k\eta_1} & \text{for} \quad z \geq 0 \ , \\ - \dfrac{\rho^{(2)}\,\sigma_0}{k}\, A_2\, e^{k\eta_1} & \text{for} \quad z \leq 0 \ . \end{cases} \tag{5.49}$$

Equations (5.42) and (5.49) show that

$$u_0 = \begin{cases} B_1(\eta_1)\, e^{-\sqrt{\sigma_0}\,\eta_0} \;-\; iA_1\, e^{-k\eta_1} & \text{for} \quad z \geq 0\ , \\ B_2(\eta_1)\, e^{\sqrt{\sigma_0}\,\eta_0} \;+\; iA_2\, e^{k\eta_1} & \text{for} \quad z \leq 0\ . \end{cases} \tag{5.50}$$

Now we determine the forms of $B_1(\eta_1)$ and $B_2(\eta_1)$. First notice that (5.45) and (5.46) indicate that

$$P_1 = P_1(\eta_1)\ . \tag{5.51}$$

By substituting (5.50) into (5.23) we obtain, for $z \geq 0$,

$$\big(\frac{\partial^2}{\partial\eta_0^2} - \sigma_0\big)\, u_1 = \frac{ik}{\rho}\, P_1(\eta_1) + \big(\sigma_1\, B_1(\eta_1) + 2\sqrt{\sigma_0}\,\frac{\mathrm{d}B_1(\eta_1)}{\mathrm{d}\eta_1}\big)\, e^{-\sqrt{\sigma_0}\,\eta_0} - iA_1\,\sigma_1\, e^{-k\eta_1}\ . \tag{5.52}$$

The secularity condition requires

$$\sigma_1\, B_1(\eta_1) + 2\sqrt{\sigma_0}\,\frac{\mathrm{d}B_1(\eta_1)}{\mathrm{d}\eta_1} = 0\ , \tag{5.53}$$

which implies

$$B_1(\eta_1) = b_1\, e^{-\frac{\sigma_1}{2\sqrt{\sigma_0}}\,\eta_1}\ , \tag{5.54}$$

where b_1 is a constant. Similarly, for $z \leq 0$, we get

$$\sigma_1\, B_2(\eta_1) + 2\sqrt{\sigma_0}\,\frac{\mathrm{d}B_2(\eta_1)}{\mathrm{d}\eta_1} = 0\ , \tag{5.55}$$

which implies

$$B_2(\eta_1) = b_2\, e^{\frac{\sigma_1}{2\sqrt{\sigma_0}}\,\eta_1}\ , \tag{5.56}$$

where b_2 is a constant.

Now the interfacial conditions at the lowest order, (5.31), (5.33), (5.35) and (5.37), will determine the unknowns A_1, A_2, b_1, b_2 and σ_0, while σ_1 will be determined by the next-order solutions.

Equation (5.33) implies that

$$a\sigma_0 = A_1 = A_2 \ , \tag{5.57}$$

where $\sigma_0 = \sigma_0^{(1)} = \sigma_0^{(2)}$. By substituting (5.49) and (5.57) into (5.37), we obtain

$$\sigma_0^2 = -\Big(\frac{\rho^{(2)} - \rho^{(1)}}{\rho^{(2)} + \rho^{(1)}}\, gk + \frac{k^3\, T}{\rho^{(2)} + \rho^{(1)}}\Big) \triangleq -F^2 \ . \tag{5.58}$$

By substituting (5.54), (5.56) and (5.57) into (5.50) and applying the condition (5.31), we have

$$b_1 - ia\,\sigma_0 = b_2 + ia\,\sigma_0 \ . \tag{5.59}$$

Meanwhile, (5.35) implies that

$$r\,(-\sqrt{\sigma_0})\, b_1 = R\,(\sqrt{\sigma_0})\, b_2 \ . \tag{5.60}$$

From (5.59) and (5.60), we find

$$b_1 = \frac{2iaR\sigma_0}{R+r} \ , \qquad b_2 = -\frac{2iar\sigma_0}{R+r} \ . \tag{5.61}$$

Then, the solutions at the lowest order are

$$u_0 \;=\; \begin{cases} \dfrac{2iaR\sigma_0}{R+r}\, e^{-\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1}\, e^{-\sqrt{\sigma_0}\,\eta_0} \;-\; ia\sigma_0\, e^{-k\eta_1} & \text{for} \quad z \geq 0 \ , \\ -\dfrac{2iar\sigma_0}{R+r}\, e^{\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1}\, e^{\sqrt{\sigma_0}\,\eta_0} \;+\; ia\sigma_0\, e^{k\eta_1} & \text{for} \quad z \leq 0 \ , \end{cases} \tag{5.62}$$

$$w_0 \;=\; \begin{cases} a\sigma_0\, e^{-k\eta_1} & \text{for} \quad z \geq 0 \ , \\ a\sigma_0\, e^{k\eta_1} & \text{for} \quad z \leq 0 \ , \end{cases} \tag{5.63}$$

$$P_0 \;=\; \begin{cases} \dfrac{\rho^{(1)}\, a\sigma_0^2}{k}\, e^{-k\eta_1} & \text{for} \quad z \geq 0 \ , \\ -\dfrac{\rho^{(2)}\, a\sigma_0^2}{k}\, e^{k\eta_1} & \text{for} \quad z \leq 0 \ , \end{cases} \tag{5.64}$$

where σ_0 is given by (5.58) and σ_1 will be determined by solutions at the next order.

5.1.3 First-order solutions

Here the first-order solutions refer to u_1, w_1, P_1 and σ_1. The idea is essentially the same as that in calculating the lowest-order solutions. The governing equations at the present order and the secularity conditions from equations at the next order determine the forms of the solutions. Then the interfacial conditions at the present order determine the coefficients in the solutions. Here are the details.

For $z \geq 0$, (5.43), (5.44) and (5.54) imply that

$$ikb_1\, e^{-\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1}\, e^{-\sqrt{\sigma_0}\,\eta_0} + \frac{\partial w_1}{\partial \eta_0} = 0\,. \tag{5.65}$$

Hence

$$w_1 = \frac{ikb_1}{\sqrt{\sigma_0}}\, e^{-\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1}\, e^{-\sqrt{\sigma_0}\,\eta_0} + F_1(\eta_1)\,. \tag{5.66}$$

Similarly, for $z \leq 0$ we obtain

$$w_1 = -\frac{ikb_2}{\sqrt{\sigma_0}}\, e^{\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1}\, e^{\sqrt{\sigma_0}\,\eta_0} + F_2(\eta_1)\,. \tag{5.67}$$

Here b_1 and b_2 are given in (5.61) while $F_1(\eta_1)$ and $F_2(\eta_1)$ are to be determined.

Equations (5.52), (5.53) and (5.57) imply that, for $z \geq 0$,

$$\Big(\frac{\partial^2}{\partial \eta_0^2} - \sigma_0\Big)\, u_1 = \frac{ik}{\rho}\, P_1(\eta_1) - ia\sigma_0\, \sigma_1\, e^{-k\eta_1}\,. \tag{5.68}$$

Similarly for $z \leq 0$, we obtain

$$\Big(\frac{\partial^2}{\partial \eta_0^2} - \sigma_0\Big)\, u_1 = \frac{ik}{\rho}\, P_1(\eta_1) + ia\sigma_0\, \sigma_1\, e^{k\eta_1}\,. \tag{5.69}$$

Hence

$$u_1 = \begin{cases} D_1(\eta_1)\, e^{-\sqrt{\sigma_0}\,\eta_0} \;+\; \dfrac{ia\sigma_0\, \sigma_1\, e^{-k\eta_1} - \frac{ik}{\rho}\, P_1(\eta_1)}{\sigma_0} & \text{for} \quad z \geq 0\,, \\ D_2(\eta_1)\, e^{\sqrt{\sigma_0}\,\eta_0} \;+\; \dfrac{-ia\sigma_0\, \sigma_1\, e^{k\eta_1} - \frac{ik}{\rho}\, P_1(\eta_1)}{\sigma_0} & \text{for} \quad z \leq 0\,. \end{cases} \tag{5.70}$$

Substitute (5.70) and (5.66) into (5.30) to obtain, for $z \geq 0$,

$$ikD_1(\eta_1)\, e^{-\sqrt{\sigma_0}\,\eta_0} + \tfrac{ikb_1}{\sqrt{\sigma_0}}\big(-\tfrac{\sigma_1}{2\sqrt{\sigma_0}}\big)\, e^{-\frac{\sigma_1}{2\sqrt{\sigma_0}}\,\eta_1}\, e^{-\sqrt{\sigma_0}\,\eta_0} \quad + \\ \big[-ak\sigma_1\, e^{-k\eta_1} + \tfrac{k^2}{\rho\,\sigma_0}\, P_1(\eta_1) + \frac{\mathrm{d}F_1(\eta_1)}{\mathrm{d}\eta_1}\big] = -\frac{\partial w_2}{\partial \eta_0} \quad . \tag{5.71}$$

To remove the secularity, we must have

$$-ak\sigma_1\, e^{-k\eta_1} + \frac{k^2}{\rho\,\sigma_0}\, P_1(\eta_1) + \frac{\mathrm{d}F_1(\eta_1)}{\mathrm{d}\eta_1} = 0\ . \tag{5.72}$$

Meanwhile, substitution of (5.51), (5.63) and (5.66) into (5.27) yields, for $z \geq 0$,

$$a\sigma_0\,\sigma_1\, e^{-k\eta_1} + \sigma_0\, F_1(\eta_1) = -\frac{1}{\rho}\,\frac{\mathrm{d}P_1(\eta_1)}{\mathrm{d}\eta_1} - \frac{1}{\rho}\,\frac{\partial P_2}{\partial \eta_0}\ . \tag{5.73}$$

The secularity condition requires

$$a\sigma_0\,\sigma_1\, e^{-k\eta_1} + \sigma_0\, F_1(\eta_1) = -\frac{1}{\rho}\,\frac{\mathrm{d}P_1(\eta_1)}{\mathrm{d}\eta_1}\ . \tag{5.74}$$

Now by substituting (5.72) into (5.74), we obtain

$$\big(\frac{\mathrm{d}^2}{\mathrm{d}\eta_1^2} - k^2\big)\, F_1 = 0\ , \tag{5.75}$$

which implies

$$F_1(\eta_1) = f_1\, e^{-k\eta_1}\ , \tag{5.76}$$

where f_1 is a constant. Similarly for $z \leq 0$, we obtain

$$\big(\frac{\mathrm{d}^2}{\mathrm{d}\eta_1^2} - k^2\big)\, F_2 = 0\ , \tag{5.77}$$

which implies

$$F_2(\eta_1) = f_2\, e^{k\eta_1}\ , \tag{5.78}$$

where f_2 is a constant. Hence,

$$w_1 = \begin{cases} \dfrac{ikb_1}{\sqrt{\sigma_0}}\, e^{-\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1}\, e^{-\sqrt{\sigma_0}\,\eta_0} + f_1\, e^{-k\eta_1} & \text{for} \quad z \geq 0\ , \\ - \quad \dfrac{ikb_2}{\sqrt{\sigma_0}}\, e^{\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1}\, e^{\sqrt{\sigma_0}\,\eta_0} + f_2\, e^{k\eta_1} & \text{for} \quad z \leq 0\ . \end{cases} \tag{5.79}$$

Consequently, we obtain

$$P_1 = \begin{cases} \dfrac{\rho^{(1)}\,\sigma_0}{k}\left(a\sigma_1^{(1)} + f_1\right) e^{-k\eta_1} & \text{for} \quad z \geq 0\ , \\ - \quad \dfrac{\rho^{(2)}\,\sigma_0}{k}\left(a\sigma_1^{(2)} + f_2\right) e^{k\eta_1} & \text{for} \quad z \leq 0\ . \end{cases} \tag{5.80}$$

Substitute (5.80) into (5.70) to obtain

$$u_1 = \begin{cases} D_1(\eta_1)\, e^{-\sqrt{\sigma_0}\,\eta_0} \quad - \quad if_1\, e^{-k\eta_1} & \text{for} \quad z \geq 0\ , \\ D_2(\eta_1)\, e^{\sqrt{\sigma_0}\,\eta_0} \quad + \quad if_2\, e^{k\eta_1} & \text{for} \quad z \leq 0\ . \end{cases} \tag{5.81}$$

Now we determine the forms of $D_1(\eta_1)$ and $D_2(\eta_1)$. First notice that (5.73) and (5.74) indicate that

$$P_2 = P_2(\eta_1)\ . \tag{5.82}$$

Substitute (5.62), (5.81) and (5.82) into (5.24) to obtain, for $z \geq 0$,

$$\begin{aligned} \Big[\big(\sigma_2 + k^2 - \frac{\sigma_1^2}{4\sigma_0}\big)\, b_1\, e^{-\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1} \quad + \quad \sigma_1\, D_1(\eta_1) + 2\sqrt{\sigma_0}\,\frac{\mathrm{d}D_1(\eta_1)}{\mathrm{d}\eta_1}\Big]\, e^{-\sqrt{\sigma_0}\,\eta_0} \\ -i\big(a\sigma_0\,\sigma_2 + f_1\,\sigma_1\big)\, e^{-k\eta_1} \quad + \quad \frac{ik}{\rho}\, P_2(\eta_1) = \big(\frac{\partial^2}{\partial\eta_0^2} - \sigma_0\big)\, u_2\ . \end{aligned} \tag{5.83}$$

Elimination of secularity in (5.83) requires that

$$\big(\sigma_2 + k^2 - \frac{\sigma_1^2}{4\sigma_0}\big)\, b_1\, e^{-\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1} + \sigma_1\, D_1(\eta_1) + 2\sqrt{\sigma_0}\,\frac{\mathrm{d}D_1(\eta_1)}{\mathrm{d}\eta_1} = 0\ . \tag{5.84}$$

From (5.84) we can determine the solution for D_1,

$$D_1(\eta_1) = \Big[d_1 - \frac{b_1}{2\sqrt{\sigma_0}}\big(\sigma_2 + k^2 - \frac{\sigma_1^2}{4\sigma_0}\big)\,\eta_1\Big]\, e^{-\frac{\sigma_1}{2\sqrt{\sigma_0}}\eta_1}\ , \tag{5.85}$$

where b_1 is given in (5.61) and d_1 is a constant to be determined. Corresponding to (5.83), for $z \leq 0$, we obtain

$$\begin{aligned} \Big[\big(\sigma_2 + k^2 - \frac{\sigma_1^2}{4\sigma_0}\big)\, b_2\, e^{\frac{\sigma_1}{2\sqrt{\sigma_0}}\,\eta_1} \quad &+ \quad \sigma_1\, D_2(\eta_1) - 2\sqrt{\sigma_0}\,\frac{\mathrm{d}D_2(\eta_1)}{\mathrm{d}\eta_1}\Big]\, e^{\sqrt{\sigma_0}\,\eta_0} \\ -i\big(a\sigma_0\,\sigma_2 - f_2\,\sigma_1\big)\, e^{k\eta_1} \quad &+ \quad \frac{ik}{\rho}\, P_2(\eta_1) = \big(\frac{\partial^2}{\partial\eta_0^2} - \sigma_0\big)\, u_2\,. \end{aligned} \tag{5.86}$$

Elimination of secularity of in (5.86) requires that

$$\big(\sigma_2 + k^2 - \frac{\sigma_1^2}{4\sigma_0}\big)\, b_2\, e^{\frac{\sigma_1}{2\sqrt{\sigma_0}}\,\eta_1} + \sigma_1\, D_2(\eta_1) - 2\sqrt{\sigma_0}\,\frac{\mathrm{d}D_2(\eta_1)}{\mathrm{d}\eta_1} = 0\,. \tag{5.87}$$

From (5.87) we can determine the solution for D_2,

$$D_2(\eta_1) = \Big[d_2 + \frac{b_1}{2\sqrt{\sigma_0}}\,\big(\sigma_2 + k^2 - \frac{\sigma_1^2}{4\sigma_0}\big)\,\eta_1\Big]\, e^{\frac{\sigma_1}{2\sqrt{\sigma_0}}\,\eta_1}\,, \tag{5.88}$$

where b_2 is given in (5.61) and d_2 is a constant to be determined. We note that, in both (5.85) and (5.88), η_1 occurs in the square brackets in the form

$$\pm\frac{b_1}{2\sqrt{\sigma_0}}\,\big(\sigma_2 + k^2 - \frac{\sigma_1^2}{4\sigma_0}\big)\,\eta_1$$

and its origin is from the expansion of

$$\exp\,\Big[\pm\,\frac{b_1}{2\sqrt{\sigma_0}}\,\big(\sigma_2 + k^2 - \frac{\sigma_1^2}{4\sigma_0}\big)\,\sqrt{\nu}\,\eta_1\Big]\,. \tag{5.89}$$

This exponential form can be recovered by introducing another scaled coordinate $\eta_2 = \sqrt{\nu}\,\eta_1 = \sqrt{\nu}\,z$. By following the same procedure as for the determination of η_1 in (5.85) and (5.88), the elimination of secularity determines the dependency on η_2.

At this stage, the solutions at the current order are expressed in (5.79), (5.80) and (5.81), with $D_1(\eta_1)$ and $D_2(\eta_1)$ given by (5.85) and (5.88), respectively. What remains

is to use the interfacial conditions (5.32), (5.34), (5.36) and (5.38) to determine the unknown coefficients f_1, f_2, d_1, d_2 and σ_1, while σ_2 has to be determined by solutions at the next order. First, by substituting (5.79) into (5.34), we obtain

$$a\sigma_1^{(1)} = \frac{ikb_1}{\sqrt{\sigma_0}} + f_1 = -\frac{ikb_2R}{\sqrt{\sigma_0}} + Rf_2 = aR\,\sigma_1^{(2)} \,. \tag{5.90}$$

Substitution of (5.61) yields

$$f_1 = a\sigma_1^{(1)} + \frac{2akR\sqrt{\sigma_0}}{R+r} \,, \qquad f_2 = a\sigma_1^{(2)} + \frac{2akr\sqrt{\sigma_0}}{R+r} \,. \tag{5.91}$$

Combine (5.80) and (5.91) to obtain

$$P_1 = \begin{cases} \dfrac{\rho^{(1)}\,\sigma_0}{k}\left(2a\sigma_1^{(1)} + \dfrac{2akR\sqrt{\sigma_0}}{R+r}\right) e^{-k\eta_1} & \text{for} \quad z \geq 0 \,, \\ -\dfrac{\rho^{(2)}\,\sigma_0}{k}\left(2a\sigma_1^{(2)} + \dfrac{2akr\sqrt{\sigma_0}}{R+r}\right) e^{k\eta_1} & \text{for} \quad z \leq 0 \,. \end{cases} \tag{5.92}$$

By substituting (5.92) into (5.38) and recalling (5.40), we obtain

$$\sigma_1^{(1)} = R\,\sigma_1^{(2)} = -\frac{2kRr\,\sqrt{\sigma_0}}{(R+r)(1+r)} \,. \tag{5.93}$$

Finally we calculate d_1 and d_2. By substituting (5.81), (5.85) and (5.88) into (5.32) we obtain

$$d_1 - if_1 = Rd_2 + iRf_2 \,, \tag{5.94}$$

or

$$d_1 = Rd_2 + 2ia\sigma_1^{(1)} + \frac{2iakR\sqrt{\sigma_0}}{R+r}\,(1+r) \,. \tag{5.95}$$

Meanwhile, (5.36) yields

$$r\left(2iak\sigma_0 - \sqrt{\sigma_0}\,d_1\right) = R^2\left(2iak\sigma_0 + \sqrt{\sigma_0}\,d_2\right) \,. \tag{5.96}$$

Combine (5.95) and (5.96) to obtain

$$\begin{aligned} d_1 &= \frac{1}{R+r}\left\{2iaR\sigma_1^{(1)} + 2iak\sqrt{\sigma_0}\left[R(1+r)+r-R^2-\frac{Rr(1+r)}{R+r}\right]\right\}, \\ d_2 &= \frac{1}{R(R+r)}\left\{-2iar\sigma_1^{(1)} + 2iak\sqrt{\sigma_0}\left[r-R^2-\frac{Rr(1+r)}{R+r}\right]\right\}. \end{aligned} \tag{5.97}$$

Except for σ_2, the calculations of the first-order solutions are complete. This solution procedure can be carried to even higher orders, though the calculations become more and more complicated. Fortunately, the solutions constructed prove adequate to test the numerical results.

5.2 Accuracy of the numerical methods

5.2.1 Truncation errors for a simple model

Let's first consider a simple model problem which captures the essential parts of our numerical method. In this case, we are able to calculate the truncation errors in closed form. The model is a one-dimensional linear diffusion equation

$$u_t = \nu u_{zz} \ . \tag{5.98}$$

Here we are not giving any initial conditions nor boundary conditions since they are not needed in calculating the numerical truncation errors.

After applying the Crank-Nicolson approximation to the time derivative in (5.98), we have

$$u^{n+1} - u^n = \frac{\nu\Delta t}{2}(u_{zz}^{n+1} + u_{zz}^n) \ . \tag{5.99}$$

Then we introduce a new variable $q = u_z$ and write (5.99) as a linear system of equations,

$$\frac{d}{dz}Y = AY + R \ , \tag{5.100}$$

where

$$Y \triangleq \begin{pmatrix} u^{n+1} \\ q^{n+1} \end{pmatrix} \ , \qquad A \triangleq \begin{pmatrix} 0 & 1 \\ \frac{2}{\nu\Delta t} & 0 \end{pmatrix} \ , \qquad R \triangleq \begin{pmatrix} 0 \\ r \end{pmatrix} \ , \tag{5.101}$$

and

$$r = -(\frac{2}{\nu\Delta t}u^n + u^n_{zz}) \ . \tag{5.102}$$

If we directly apply the trapezoidal rule to (5.100), we have

$$(I - \frac{\Delta z}{2}A)\, Y_{j+1} - (I + \frac{\Delta z}{2}A)\, Y_j = \frac{\Delta z}{2}(R_j + R_{j+1}) \ , \tag{5.103}$$

where I is the identity matrix. Consequently, we obtain

$$Y_{j+1} = (I - \frac{\Delta z}{2}A)^{-1}\left[(I + \frac{\Delta z}{2}A)\, Y_j + \frac{\Delta z}{2}(R_j + R_{j+1})\right] \ , \tag{5.104}$$

which can be then implemented numerically. All that is needed to complete this step is an approximation for R_j and R_{j+1}. Since R involves r, we need an approximation for u^n_{zz}. Two methods are tried: $(u_{zz})^n_j = \dfrac{(u_z)^n_{j+1} - (u_z)^n_{j-1}}{2\Delta z}$ or $\dfrac{(u_{zz})^n_j + (u_{zz})^n_{j+1}}{2} = \dfrac{(u_z)^n_{j+1} - (u_z)^n_j}{\Delta z}$. Numerically, we find no significant difference.

However, as we mentioned in Chapter 3, direct application of the trapezoidal rule is numerically unstable. The remedy is to diagonalize the system (5.100). It's clear to see the two eigenvalues of the matrix A in (5.100) are $\lambda_1 = \lambda \triangleq \sqrt{2/(\nu\Delta t)}$, $\lambda_2 = -\lambda \triangleq -\sqrt{2/(\nu\Delta t)}$ and they have the corresponding eigenvectors $e_1 = \begin{pmatrix} 1 \\ \lambda \end{pmatrix}$, $e_2 =$

$\begin{pmatrix} -1 \\ \lambda \end{pmatrix}$. Let the transformatin matrix $P \triangleq (e_1, e_2)$ and transform $Y = P\widetilde{Y}$. That is,

$$\begin{pmatrix} u^{n+1} \\ q^{n+1} \end{pmatrix} = P \begin{pmatrix} \widetilde{u}^{n+1} \\ \widetilde{q}^{n+1} \end{pmatrix} = \begin{pmatrix} \widetilde{u}^{n+1} - \widetilde{q}^{n+1} \\ \lambda(\widetilde{u}^{n+1} + \widetilde{q}^{n+1}) \end{pmatrix} , \tag{5.105}$$

where $\widetilde{u}, \widetilde{q}$ are the transformed variables and they can be expressed by the inverse of (5.105),

$$\begin{pmatrix} \widetilde{u}^{n+1} \\ \widetilde{q}^{n+1} \end{pmatrix} = P^{-1} \begin{pmatrix} u^{n+1} \\ q^{n+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(u^{n+1} + q^{n+1}/\lambda) \\ -\frac{1}{2}(u^{n+1} - q^{n+1}/\lambda) \end{pmatrix} . \tag{5.106}$$

Now the system (5.100) becomes

$$\frac{d}{dz}\widetilde{u}^{n+1} = \lambda_1 \widetilde{u}^{n+1} + \widetilde{R}_1 , \tag{5.107}$$

$$\frac{d}{dz}\widetilde{q}^{n+1} = \lambda_2 \widetilde{q}^{n+1} + \widetilde{R}_2 , \tag{5.108}$$

or, in block form

$$\frac{d}{dz}\widetilde{Y} = \widetilde{A}\widetilde{Y} + \widetilde{R} , \tag{5.109}$$

where $\widetilde{A} \triangleq \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} = \begin{pmatrix} \lambda & \\ & -\lambda \end{pmatrix}$, $\widetilde{R} \triangleq \begin{pmatrix} \widetilde{R}_1 \\ \widetilde{R}_2 \end{pmatrix} = P^{-1}R = \begin{pmatrix} r/(2\lambda) \\ r/(2\lambda) \end{pmatrix}$.

Now apply the trapezoidal rule to the two scalar ODEs (5.107), (5.108), respectively,

$$(1 - \frac{\Delta z}{2}\lambda_1)\widetilde{u}_{j+1}^{n+1} - (1 + \frac{\Delta z}{2}\lambda_1)\widetilde{u}_j^{n+1} = \frac{\Delta z}{2}\big((\widetilde{R}_1)_{j+1} + (\widetilde{R}_1)_j \big) , \tag{5.110}$$

$$(1 - \frac{\Delta z}{2}\lambda_2)\widetilde{q}_{j+1}^{n+1} - (1 + \frac{\Delta z}{2}\lambda_2)\widetilde{q}_j^{n+1} = \frac{\Delta z}{2}\big((\widetilde{R}_2)_{j+1} + (\widetilde{R}_2)_j \big) . \tag{5.111}$$

Since $\lambda_1 > 0$, recursion (5.110) is applied backwards, i.e., $\widetilde{u}_j^{n+1}$ is calculated knowing $\widetilde{u}_{j+1}^{n+1}$. In this way, a decreasing sequence is obtained which is numerically stable. Alternatively, (5.111) is applied in the forward direction.

On the other hand, we are concerned with the numerical accuracy here. It turns out the current method has the same truncation errors as (5.103)(5.104). To see this, write (5.110), (5.111) in block form

$$(I - \frac{\Delta z}{2}\widetilde{A})\widetilde{Y}_{j+1} - (I + \frac{\Delta z}{2}\widetilde{A})\widetilde{Y}_j = \frac{\Delta z}{2}(\widetilde{R}_{j+1} + \widetilde{R}_j) \,. \tag{5.112}$$

Multiply with the matrix P on both sides to obtain

$$P(I - \frac{\Delta z}{2}\widetilde{A})P^{-1}P\widetilde{Y}_{j+1} - P(I + \frac{\Delta z}{2}\widetilde{A})P^{-1}P\widetilde{Y}_j = \frac{\Delta z}{2}P(\widetilde{R}_{j+1} + \widetilde{R}_j) \,. \tag{5.113}$$

Equation (5.113) is identical to (5.103) since $P\widetilde{A}P^{-1} = A\,, \quad P\widetilde{Y} = Y\,, \quad P\widetilde{R} = R\,.$ Therefore, we may go back to (5.104) and start calculating the truncation errors from there.

By substituting (5.101) into (5.104), we obtain

$$u_{j+1}^{n+1} = \frac{1}{1-\beta}\big[(1+\beta)u_j^{n+1} + \Delta z q_j^{n+1} + \frac{\Delta z^2}{4}(r_j + r_{j+1})\big] \,, \tag{5.114}$$

$$q_{j+1}^{n+1} = \frac{1}{1-\beta}\big[(1+\beta)q_j^{n+1} + \frac{2\Delta z}{\nu\Delta t}u_j^{n+1} + \frac{\Delta z}{2}(r_j + r_{j+1})\big] \,, \tag{5.115}$$

where $\beta = \frac{\Delta z^2}{2\nu\Delta t}$. Equation (5.115) still holds when we replace j by $j-1$,

$$q_j^{n+1} = \frac{1}{1-\beta}\big[(1+\beta)q_{j-1}^{n+1} + \frac{2\Delta z}{\nu\Delta t}u_{j-1}^{n+1} + \frac{\Delta z}{2}(r_{j-1} + r_j)\big] \,. \tag{5.116}$$

From (5.114) we find

$$q_j^{n+1} = \frac{1}{\Delta z}\big[(1-\beta)u_{j+1}^{n+1} - (1+\beta)u_j^{n+1} - \frac{\Delta z^2}{4}(r_j + r_{j+1})\big] \,. \tag{5.117}$$

Replace j by $j-1$ in (5.117),

$$q_{j-1}^{n+1} = \frac{1}{\Delta z}\Big[(1-\beta)u_j^{n+1} - (1+\beta)u_{j-1}^{n+1} - \frac{\Delta z^2}{4}(r_{j-1}+r_j)\Big] \ . \tag{5.118}$$

Substituting (5.117) and (5.118) into (5.116) and collecting terms, we obtain

$$(1-\beta)u_{j+1}^{n+1} - 2(1+\beta)u_j^{n+1} + (1-\beta)u_{j-1}^{n+1} = \frac{\Delta z^2}{4}(r_{j+1}+2r_j+r_{j-1}) \ . \tag{5.119}$$

By substituting (5.102), the right-hand side of (5.119) reads

$$\begin{aligned}&\frac{\Delta z^2}{4}(r_{j+1}+2r_j+r_{j-1}) = \\ &-\frac{\Delta z^2}{4}\Big[\frac{2}{\nu\Delta t}(u_{j+1}^n+2u_j^n+u_{j-1}^n) + \big((u_{zz})_{j+1}^n + 2(u_{zz})_j^n + (u_{zz})_{j-1}^n\big)\Big] \ .\end{aligned} \tag{5.120}$$

We have two methods for calculating the $(u_{zz})_j^n$ contributions to r_j – see the discussion following (5.104). Both methods result in

$$(u_{zz})_{j+1}^n + 2(u_{zz})_j^n + (u_{zz})_{j-1}^n = 4(u_{zz})_j^n + C\Delta z^2(u_{zzzz})_j^n \ , \tag{5.121}$$

where the constant C depends on the choice of the numerical approximation. Now we can write (5.119) as

$$\begin{aligned}&(1-\beta)u_{j+1}^{n+1} - 2(1+\beta)u_j^{n+1} + (1-\beta)u_{j-1}^{n+1} = \\ &-\beta(u_{j+1}^n+2u_j^n+u_{j-1}^n) - \Delta z^2(u_{zz})_j^n - C\frac{\Delta z^4}{4}(u_{zzzz})_j^n \ .\end{aligned} \tag{5.122}$$

Finally we carry out standard Taylor's series expansion for (5.122) about (t_n, z_j) to determine the truncation errors. After some algebra, we obtain

$$u_t - \nu u_{zz} = -\frac{C+2}{8}\nu\Delta z^2 u_{zzzz} + \frac{1}{12}\nu^3\Delta t^2 u_{zzzzzz} + o(\Delta t^2, \Delta z^2) \ , \tag{5.123}$$

which shows that the truncation error for the method is $O(\Delta t^2, \Delta z^2)$.

5.2.2 Order of accuracy for the linear problem

We now go to the main point of our error analysis: study the accuracy of our numerical methods applied to the two-fluid system in the linear case. Now we have a 4×4 system and it's difficult to apply a similar procedure as that for the previous model problem to compute the truncation errrors. On the other hand, since we are most interested in the motion of the interfacial flows, we have to consider the interfacial conditions in our analysis and it's not good enough to just calculate the truncation errors. Consequently, our goal here is to theoretically justify the second-order accuracy for our numerical methods applied to the linear problem, without worrying too much about the constant coefficients in the error expressions. Both the governing equations and the boundary conditions will be considered in our study.

Let's first investigate the temporal discretizations by using the normal mode analysis [18][64]. The governing equations, under the Fourier transform in x , are

$$u_t = -\frac{ik}{\rho}P + \nu(u_{zz} - k^2 u) \,, \tag{5.124}$$

$$w_t = -\frac{1}{\rho}P_z + \nu(w_{zz} - k^2 w) \,, \tag{5.125}$$

$$iku + w_z = 0 \,, \tag{5.126}$$

where $k > 0$. By taking the Laplace transform in t to the above equations we obtain

$$\big[\frac{d^2}{dz^2} - (\frac{\sigma}{\nu} + k^2)\big]\,\hat{u} = \frac{ik}{\mu}\hat{P} \,, \tag{5.127}$$

$$\big[\frac{d^2}{dz^2} - (\frac{\sigma}{\nu} + k^2)\big]\,\hat{w} = \frac{1}{\mu}\hat{P}_z \,, \tag{5.128}$$

$$(\frac{d^2}{dz^2} - k^2)\,\hat{P} = 0 \,, \tag{5.129}$$

which is a linear system of ODEs. Together with the boundary conditions we can calculate the analytical solution, denoted by $Y(\sigma)$.

Now we apply the Crank-Nicolson method in t to approximate (5.124)-(5.126),

$$\frac{u^{n+1}-u^n}{\Delta t} = -\frac{ik}{2\rho}(P^{n+1}+P^n)+\frac{\nu}{2}\left[u_{zz}^{n+1}+u_{zz}^n-k^2(u^{n+1}+u^n)\right], \quad (5.130)$$

$$\frac{w^{n+1}-w^n}{\Delta t} = -\frac{1}{2\rho}(P_z^{n+1}+P_z^n)+\frac{\nu}{2}\left[w_{zz}^{n+1}+w_{zz}^n-k^2(w^{n+1}+w^n)\right], \quad (5.131)$$

$$iku^{n+1}+w_z^{n+1}=0\,. \quad (5.132)$$

By taking the Laplace transform in t to (5.130)-(5.132), we obtain

$$\left[\frac{d^2}{dz^2}-(\frac{\bar\sigma}{\nu}+k^2)\right]\hat u=\frac{ik}{\mu}\hat P\,, \quad (5.133)$$

$$\left[\frac{d^2}{dz^2}-(\frac{\bar\sigma}{\nu}+k^2)\right]\hat w=\frac{1}{\mu}\hat P_z\,, \quad (5.134)$$

$$(\frac{d^2}{dz^2}-k^2)\,\hat P=0\,, \quad (5.135)$$

where

$$\bar\sigma=\frac{2}{\Delta t}\frac{e^{\sigma\Delta t}-1}{e^{\sigma\Delta t}+1}=\sigma+O(\sigma^3\Delta t^2)\,. \quad (5.136)$$

Similarly, if we apply the second-order BDF method to approximate (5.124)-(5.126), we obtain

$$\frac{3u^{n+1}-4u^n+u^{n-1}}{2\Delta t} = -\frac{ik}{\rho}P^{n+1}+\nu(u_{zz}^{n+1}-k^2u^{n+1})\,, \quad (5.137)$$

$$\frac{3w^{n+1}-4w^n+w^{n-1}}{2\Delta t} = -\frac{1}{\rho}P_z^{n+1}+\nu(w_{zz}^{n+1}-k^2w^{n+1})\,, \quad (5.138)$$

$$iku^{n+1}+w_z^{n+1}=0\,, \quad (5.139)$$

which recovers (5.133)-(5.135) but with a different $\bar\sigma$,

$$\bar\sigma=\frac{3-4e^{-\sigma\Delta t}+e^{-2\sigma\Delta t}}{2\Delta t}=\sigma+O(\sigma^3\Delta t^2)\,. \quad (5.140)$$

It's clear to see the similarity between the two systems (5.133)-(5.135) and (5.127)-(5.129). Meanwhile, we notice that there is no time derivative in the boundary conditions. In our numerical methods, we use the implicit treatment for all the boundary conditions so that the discretized equations take essentially the same form as the analytic ones. Therefore, we can readily see the solution to (5.133)-(5.135) will be $Y(\bar{\sigma})$. We have

$$Y(\bar{\sigma}) = Y(\sigma + C\Delta t^2) = Y(\sigma) + O(\Delta t^2) \ , \tag{5.141}$$

which shows that our numerical methods are second-order in time.

However, our numerical methods also include the discretization in the vertical spacial direction, z, and the numerical errors associated with Δt and Δz are coupled with each other. So we turn to the numerical errors associated with the discretization in z, i.e., the ODE solver in our methods, and determine the overall accuracy for both Δt and Δz.

Let's first consider the analytic equations (5.127)-(5.129). Following the ideas in our numerical method, we introduce a new variable

$$\hat{q} = \hat{u}_z \ , \tag{5.142}$$

and transfer (5.127)-(5.129) into a first-order linear ODE system with respect to z,

$$\frac{\mathrm{d}}{\mathrm{d}z} Y = B(\sigma)\, Y \ , \tag{5.143}$$

where

$$Y \triangleq \begin{pmatrix} \hat{u} \\ \hat{q} \\ \hat{w} \\ \hat{P} \end{pmatrix}, \quad B(\sigma) \triangleq \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{\nu}(\sigma + \nu k^2) & 0 & 0 & \frac{1}{\rho\nu} ik \\ -ik & 0 & 0 & 0 \\ 0 & -\rho\nu ik & -\rho(\sigma + \nu k^2) & 0 \end{bmatrix} . \quad (5.144)$$

Accordingly, the interfacial conditions become

$$S^{(1)} Y^{(1)}(0) - S^{(2)} Y^{(2)}(0) = r \ , \quad (5.145)$$

where the matrix S and the vector r are given by

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mu & ik\mu & 0 \\ 0 & 0 & 1 & 0 \\ 2ik\mu & 0 & 0 & 1 \end{bmatrix}, \qquad r = \begin{pmatrix} 0 \\ 0 \\ 0 \\ (\rho^{(1)} - \rho^{(2)}) g\hat{h} - k^2 T\hat{h} \end{pmatrix} . \quad (5.146)$$

The matrix $B(\sigma)$ has four distinct eigenvalues

$$\lambda_1(\sigma) = k \ , \quad \lambda_2(\sigma) = -k \ , \quad \lambda_3(\sigma) = \sqrt{k^2 + \frac{\sigma}{\nu}} \ , \quad \lambda_4(\sigma) = -\sqrt{k^2 + \frac{\sigma}{\nu}} \ . \quad (5.147)$$

The eigenvectors associated with these eigenvalues are

$$e_1 = \begin{pmatrix} -ki \\ -k^2 i \\ -k \\ \rho\sigma \end{pmatrix}, \ e_2 = \begin{pmatrix} -ki \\ k^2 i \\ k \\ \rho\sigma \end{pmatrix}, \ e_3 = \begin{pmatrix} -\sqrt{k^2 + \frac{\sigma}{\nu}} \\ -(k^2 + \frac{\sigma}{\nu}) \\ ik \\ 0 \end{pmatrix}, \ e_4 = \begin{pmatrix} \sqrt{k^2 + \frac{\sigma}{\nu}} \\ -(k^2 + \frac{\sigma}{\nu}) \\ ik \\ 0 \end{pmatrix} . \quad (5.148)$$

We can then use these eigenvectors to diagonalize the matrix $B(\sigma)$. Define a matrix

$$Q(\sigma) = (e_1, e_2, e_3, e_4) \,, \tag{5.149}$$

and perform the transformation

$$Y = Q(\sigma)\, \widetilde{Y} \,. \tag{5.150}$$

Then the system (5.143) becomes

$$\frac{\mathrm{d}}{\mathrm{d}z}\widetilde{Y} = \widetilde{B}(\sigma)\, \widetilde{Y} \,, \tag{5.151}$$

where

$$\widetilde{Y} = \begin{pmatrix} \widetilde{y}_1 \\ \widetilde{y}_2 \\ \widetilde{y}_3 \\ \widetilde{y}_4 \end{pmatrix} \,, \quad \widetilde{B}(\sigma) = Q^{-1}(\sigma)\, B(\sigma)\, Q(\sigma) = \begin{pmatrix} \lambda_1(\sigma) & & & \\ & \lambda_2(\sigma) & & \\ & & \lambda_3(\sigma) & \\ & & & \lambda_4(\sigma) \end{pmatrix} . \tag{5.152}$$

Correspondingly, the interfacial conditions (5.145) become

$$S^{(1)}\, Q^{(1)}(\sigma)\, \widetilde{Y}^{(1)}(0) - S^{(2)}\, Q^{(2)}(\sigma)\, \widetilde{Y}^{(2)}(0) = r \,. \tag{5.153}$$

Now the system (5.151) may be separated into four scalar equations

$$\frac{\mathrm{d}}{\mathrm{d}z}\widetilde{y}_m = \lambda_m\, \widetilde{y}_m \,, \qquad m = 1, 2, 3, 4 \,, \tag{5.154}$$

whose general solutions are

$$\widetilde{y}_m(z) = c_m\, e^{\lambda_m z} \,, \qquad m = 1, 2, 3, 4 \,, \tag{5.155}$$

where the coefficients c_m will be determined by the boundary conditions. We consider the domain $-H \le z \le H$ where H is a fixed number. We pick a positive integer J such that $H = J\Delta z$ and denote

$$z_j = j\,\Delta z\,, \qquad \text{where} \quad j = \begin{cases} J, J-1, \cdots, 0\,, & \text{in upper fluid}\,, \\ 0, -1, \cdots, -J\,, & \text{in lower fluid}\,. \end{cases} \tag{5.156}$$

When $j = 0$, i.e., at the interface, we will use the superscripts (1) and (2) to distinguish the points in the upper and the lower fluids.

Since we are considering $k > 0$, we have $\lambda_1\,,\ \lambda_3 > 0$ and $\lambda_2\,,\ \lambda_4 < 0$. For the two positive eigenvalues $\lambda_1\,,\ \lambda_3$, our numerical integration starts from the top $z = z_J$. Correspondingly, we assume the boundary conditions for the analytic solutions are specified on the top and, consequently, the analytic solutions in the upper fluid are given by

$$\widetilde{y}_m(z) = \widetilde{y}_m(z_J)\, e^{\lambda_m\,(z - z_J)}\,, \qquad z \ge 0\,, \qquad m = 1, 3\,. \tag{5.157}$$

When evaluated at the grid points $z_j = j\Delta z$, (5.157) yields

$$\widetilde{y}_m(z_j) = \widetilde{y}_m(z_J)\, e^{\lambda_m\,(j-J)\,\Delta z}\,, \qquad j = J, J-1, \cdots, 0\,, \qquad m = 1, 3\,. \tag{5.158}$$

On the other hand, for the two negative eigenvalues $\lambda_2\,,\ \lambda_4$, our numerical integration starts from the bottom $z = z_{-J}$. Correspondingly, we assume the boundary conditions for the analytic solutions are specified on the bottom and obtain the analytic solutions in the lower fluid by

$$\widetilde{y}_m(z) = \widetilde{y}_m(z_{-J})\, e^{\lambda_m\,(z - z_{-J})}\,, \qquad z \le 0\,, \qquad m = 2, 4\,. \tag{5.159}$$

At the grid points $z_j = j\Delta z$,

$$\widetilde{y}_m(z_j) = \widetilde{y}_m(z_{-J})\, e^{\lambda_m\,(j+J)\,\Delta z}\,, \qquad j = -J, -J+1, \cdots, 0\,, \qquad m = 2, 4\,. \tag{5.160}$$

At the interface $j = 0$, we obtain $\widetilde{y}_1^{(1)}(0)$ and $\widetilde{y}_3^{(1)}(0)$ by following (5.158), $\widetilde{y}_2^{(2)}(0)$ and $\widetilde{y}_4^{(2)}(0)$ by following (5.160). To proceed, we need to determine $\widetilde{y}_1^{(2)}(0)$, $\widetilde{y}_3^{(2)}(0)$ and $\widetilde{y}_2^{(1)}(0)$, $\widetilde{y}_4^{(1)}(0)$. This is achieved by applying the interfacial condition (5.153). Let's denote the column vectors of the matrix $S^{(1)}\,Q^{(1)}(\sigma)$ by $g_m^{(1)},\ m = 1,2,3,4$, and those of the matrix $S^{(2)}\,Q^{(2)}(\sigma)$ by $g_m^{(2)},\ m = 1,2,3,4$. Then (5.153) becomes

$$\Big(g_1^{(1)},\, g_2^{(1)},\, g_3^{(1)},\, g_4^{(1)}\Big) \begin{pmatrix} \widetilde{y}_1^{(1)}(0) \\ \widetilde{y}_2^{(1)}(0) \\ \widetilde{y}_3^{(1)}(0) \\ \widetilde{y}_4^{(1)}(0) \end{pmatrix} - \Big(g_1^{(2)},\, g_2^{(2)},\, g_3^{(2)},\, g_4^{(2)}\Big) \begin{pmatrix} \widetilde{y}_1^{(2)}(0) \\ \widetilde{y}_2^{(2)}(0) \\ \widetilde{y}_3^{(2)}(0) \\ \widetilde{y}_4^{(2)}(0) \end{pmatrix} = r\ . \quad (5.161)$$

Collecting the unknowns to the left-hand side and the knowns to the right-hand side, we obtain

$$\Big(-g_1^{(2)},\, g_2^{(1)},\, -g_3^{(2)},\, g_4^{(1)}\Big) \begin{pmatrix} \widetilde{y}_1^{(2)}(0) \\ \widetilde{y}_2^{(1)}(0) \\ \widetilde{y}_3^{(2)}(0) \\ \widetilde{y}_4^{(1)}(0) \end{pmatrix} = \Big(-g_1^{(1)},\, g_2^{(2)},\, -g_3^{(1)},\, g_4^{(2)}\Big) \begin{pmatrix} \widetilde{y}_1^{(1)}(0) \\ \widetilde{y}_2^{(2)}(0) \\ \widetilde{y}_3^{(1)}(0) \\ \widetilde{y}_4^{(2)}(0) \end{pmatrix} + r\,. \quad (5.162)$$

Or, written in a compact form,

$$M_1(\sigma)\,U = M_2(\sigma)\,V + r\ , \quad (5.163)$$

where U denotes the unknowns $\big(\widetilde{y}_1^{(2)}(0),\, \widetilde{y}_2^{(1)}(0),\, \widetilde{y}_3^{(2)}(0),\, \widetilde{y}_4^{(1)}(0)\big)^T$ and the solution of (5.163) is given by

$$U = M_1^{-1}(\sigma)\,M_2(\sigma)\,V + M_1^{-1}(\sigma)\,r\ . \quad (5.164)$$

Once U is obtained, we can continue the calculation into the other half domain. For λ_1 and λ_3,

$$\widetilde{y}_m(z_j) = \widetilde{y}_m^{(2)}(0)\, e^{\lambda_m\, j\Delta z}\,, \qquad j = 0, -1, \cdots, -J\,, \qquad m = 1, 3\,. \tag{5.165}$$

For λ_2 and λ_4,

$$\widetilde{y}_m(z_j) = \widetilde{y}_m^{(1)}(0)\, e^{\lambda_m\, j\Delta z}\,, \qquad j = 0, 1, \cdots, J\,, \qquad m = 2, 4\,. \tag{5.166}$$

In summary, the solutions of the system (5.151) are given by

$$\widetilde{Y}(z_j) = \begin{pmatrix} \widetilde{y}_1(z_J)\, e^{\lambda_1\,(j-J)\Delta z} \\ \widetilde{y}_2^{(1)}(0)\, e^{\lambda_2\, j\Delta z} \\ \widetilde{y}_3(z_J)\, e^{\lambda_3\,(j-J)\Delta z} \\ \widetilde{y}_4^{(1)}(0)\, e^{\lambda_4\, j\Delta z} \end{pmatrix} \quad \text{in fluid 1}\,, \qquad \begin{pmatrix} \widetilde{y}_1^{(2)}(0)\, e^{\lambda_1\, j\Delta z} \\ \widetilde{y}_2(z_{-J})\, e^{\lambda_2\,(j+J)\Delta z} \\ \widetilde{y}_3^{(2)}(0)\, e^{\lambda_3\, j\Delta z} \\ \widetilde{y}_4(z_{-J})\, e^{\lambda_4\,(j+J)\Delta z} \end{pmatrix} \quad \text{in fluid 2}\,, \tag{5.167}$$

where $\lambda_m = \lambda_m(\sigma)\,,\ m = 1, 2, 3, 4$. The solutions to the original system (5.143) are recovered by using (5.150).

Now we consider the numerical solutions and compare with the above analytical solutions. The procedure is similar to the above. However σ has to be replaced by $\bar{\sigma}$. Corresponding to (5.143) we have

$$\frac{\mathrm{d}}{\mathrm{d}z}\, Y = B(\bar{\sigma})\, Y\,. \tag{5.168}$$

By using the transformation

$$Y = Q(\bar{\sigma})\, \widetilde{Y}\,, \tag{5.169}$$

where the matrix Q is defined in (5.148) and (5.149), the system (5.168) is transformed into

$$\frac{\mathrm{d}}{\mathrm{d}z}\widetilde{Y} = \widetilde{B}(\bar{\sigma})\, \widetilde{Y}\,, \tag{5.170}$$

where $\widetilde{B}(\bar{\sigma}) = \text{diag}\Big(\lambda_1(\bar{\sigma}),\, \lambda_2(\bar{\sigma}),\, \lambda_3(\bar{\sigma}),\, \lambda_4(\bar{\sigma})\Big)$. We represent the numerical solution of (5.170) at the grid points $z_j = j\Delta z$ as

$$\widetilde{Y}_j = \big(\widetilde{y}_{1,\,j}\,,\, \widetilde{y}_{2,\,j}\,,\, \widetilde{y}_{3,\,j}\,,\, \widetilde{y}_{4,\,j}\big)\,, \qquad \text{where} \quad j = \begin{cases} J, J-1, \cdots, 0\,, & \text{in upper fluid}\,, \\ 0, -1, \cdots, -J\,, & \text{in lower fluid}\,. \end{cases} \tag{5.171}$$

Since we are seeking solutions that are exponentially decaying away from the interface, we can always pick the height of the domain, H, to be big enough so that the numerical errors in the far fields are as small as possible. Hence, for simplicity of discussion, we assume the exact solutions are given on the top $z = z_J$ and the bottom $z = z_{-J}$. That means, $\widetilde{y}_{m,\,J} = \widetilde{y}_m(z_J)$ for $m = 1, 3$ and $\widetilde{y}_{m,\,-J} = \widetilde{y}_m(z_{-J})$ for $m = 2, 4$.

Then, for each of the 4 scalar equations in (5.154), we apply the trapezoidal rule to obtain

$$\big(1 - \frac{\Delta z}{2}\lambda_m\big)\,\widetilde{y}_{m,\,j+1} - \big(1 + \frac{\Delta z}{2}\lambda_m\big)\,\widetilde{y}_{m,\,j} = 0\,. \tag{5.172}$$

For λ_1 and λ_3, where we start the calcualtions from the top, (5.172) yields

$$\widetilde{y}_{m,\,j} = \widetilde{y}_{m,\,J}\Big(\frac{1 - \frac{\Delta z}{2}\lambda_m}{1 + \frac{\Delta z}{2}\lambda_m}\Big)^{J-j} = \widetilde{y}_m(z_J)\Big(\frac{1 - \frac{\Delta z}{2}\lambda_m}{1 + \frac{\Delta z}{2}\lambda_m}\Big)^{J-j}\,, \tag{5.173}$$

where $j = J, J-1, \cdots, 0\,;\ m = 1, 3$. For λ_2 and λ_4, where we start the calculations from the bottom, (5.172) yields

$$\widetilde{y}_{m,\,j} = \widetilde{y}_{m,\,-J}\Big(\frac{1 + \frac{\Delta z}{2}\lambda_m}{1 - \frac{\Delta z}{2}\lambda_m}\Big)^{J+j} = \widetilde{y}_m(z_{-J})\Big(\frac{1 + \frac{\Delta z}{2}\lambda_m}{1 - \frac{\Delta z}{2}\lambda_m}\Big)^{J+j}\,, \tag{5.174}$$

where $j = -J, -J+1, \cdots, 0\,;\ m = 2, 4$.

At the interface, we need to determine the four unknowns $\big(\,\widetilde{y}^{(2)}_{1,0},\,\widetilde{y}^{(1)}_{2,0},\,\widetilde{y}^{(2)}_{3,0},\,\widetilde{y}^{(1)}_{4,0}\,\big)^T \triangleq \bar{U}$. This is achieved in a similar way to (5.163),

$$M_1(\bar{\sigma})\,\bar{U} = M_2(\bar{\sigma})\,\bar{V} + r\ , \tag{5.175}$$

where

$$\bar{V} = \big(\,\widetilde{y}^{(1)}_{1,0},\,\widetilde{y}^{(2)}_{2,0},\,\widetilde{y}^{(1)}_{3,0},\,\widetilde{y}^{(2)}_{4,0}\,\big)^T \tag{5.176}$$

is known by following (5.173) and (5.174). The solution of (5.175) is given by

$$\bar{U} = M_1^{-1}(\bar{\sigma})\,M_2(\bar{\sigma})\,\bar{V} + M_1^{-1}(\bar{\sigma})\,r\ . \tag{5.177}$$

Then we can proceed to calculate the solutions in the other half domain.

$$\widetilde{y}_{m,\,j} = \widetilde{y}^{(2)}_{m,\,0}\Big(\frac{1-\frac{\Delta z}{2}\lambda_m}{1+\frac{\Delta z}{2}\lambda_m}\Big)^{-j}, \quad j = 0,-1,\cdots,-J\,,\ \ m = 1,3, \tag{5.178}$$

and

$$\widetilde{y}_{m,\,j} = \widetilde{y}^{(1)}_{m,\,0}\Big(\frac{1+\frac{\Delta z}{2}\lambda_m}{1-\frac{\Delta z}{2}\lambda_m}\Big)^{j}, \quad j = 0,1,\cdots,J\,,\ \ m = 2,4. \tag{5.179}$$

Therefore, the solutions for the system (5.170) are given by

$$\widetilde{Y}_j = \begin{pmatrix} \widetilde{y}_1(z_J)\Big(\frac{1-\frac{\Delta z}{2}\lambda_1}{1+\frac{\Delta z}{2}\lambda_1}\Big)^{J-j} \\ \widetilde{y}^{(1)}_{2,0}\Big(\frac{1+\frac{\Delta z}{2}\lambda_2}{1-\frac{\Delta z}{2}\lambda_2}\Big)^{j} \\ \widetilde{y}_3(z_J)\Big(\frac{1-\frac{\Delta z}{2}\lambda_3}{1+\frac{\Delta z}{2}\lambda_3}\Big)^{J-j} \\ \widetilde{y}^{(1)}_{4,0}\Big(\frac{1+\frac{\Delta z}{2}\lambda_4}{1-\frac{\Delta z}{2}\lambda_4}\Big)^{j} \end{pmatrix} \quad \text{in fluid 1}\,, \qquad \begin{pmatrix} \widetilde{y}^{(2)}_{1,0}\Big(\frac{1-\frac{\Delta z}{2}\lambda_1}{1+\frac{\Delta z}{2}\lambda_1}\Big)^{-j} \\ \widetilde{y}_2(z_{-J})\Big(\frac{1+\frac{\Delta z}{2}\lambda_2}{1-\frac{\Delta z}{2}\lambda_2}\Big)^{J+j} \\ \widetilde{y}^{(2)}_{3,0}\Big(\frac{1-\frac{\Delta z}{2}\lambda_3}{1+\frac{\Delta z}{2}\lambda_3}\Big)^{-j} \\ \widetilde{y}_4(z_{-J})\Big(\frac{1+\frac{\Delta z}{2}\lambda_4}{1-\frac{\Delta z}{2}\lambda_4}\Big)^{J+j} \end{pmatrix} \quad \text{in fluid 2}\,, \tag{5.180}$$

where $\lambda_m = \lambda_m(\bar{\sigma})\,,\ m = 1,2,3,4$. The solutions for the original system (5.168) are recovered by applying (5.169).

The numerical errors can then be analyzed by comparing the numerical solutions (5.180) with the analytical solutions (5.167). We only need to do that for the solutions in the upper fluid since the discussion is essentially the same for that in the lower fluid.

From (5.136) or (5.140) we know $\bar{\sigma} = \sigma + C\Delta t^2$. Here and in what follows we use the common notation C to denote any constant scalar, vector or matrix that is of order $O(1)$. There is no need to distinguish these constants in our analysis here. Then we have, from (5.147)

$$\lambda_m(\bar{\sigma}) = \lambda_m(\sigma) + C\Delta t^2\,, \qquad m = 1,2,3,4\,. \tag{5.181}$$

As a result, for $m = 1,3$,

$$\begin{aligned}
\frac{1-\frac{\Delta z}{2}\lambda_m(\bar{\sigma})}{1+\frac{\Delta z}{2}\lambda_m(\bar{\sigma})} &= \Big(1-\frac{\Delta z}{2}\lambda_m(\bar{\sigma})\Big)\Big(1-\frac{\Delta z}{2}\lambda_m(\bar{\sigma})+\frac{\Delta z^2}{4}\lambda_m^2(\bar{\sigma})-\frac{\Delta z^3}{8}\lambda_m^3(\bar{\sigma})+\cdots\Big) \\
&= 1-\lambda_m(\bar{\sigma})\Delta z+\frac{\Delta z^2}{2}\lambda_m^2(\bar{\sigma})+C\Delta z^3 \\
&= 1-\lambda_m(\sigma)\Delta z+\frac{\Delta z^2}{2}\lambda_m^2(\sigma)+C\Delta z^3+C\Delta z\Delta t^2 \\
&= e^{-\lambda_m(\sigma)\Delta z}+C\Delta z^3+C\Delta z\Delta t^2\,.
\end{aligned} \tag{5.182}$$

Hence,

$$\Big(\frac{1-\frac{\Delta z}{2}\lambda_m(\bar{\sigma})}{1+\frac{\Delta z}{2}\lambda_m(\bar{\sigma})}\Big)^{J-j} = e^{\lambda_m(\sigma)(j-J)\Delta z}+C\Delta z^2+C\Delta t^2\,. \tag{5.183}$$

By comparing (5.180) with (5.167) for the solutions in fluid 1 and using (5.183), we can readily see that, for $m = 1,3$,

$$\widetilde{y}_{m,j} = \widetilde{y}_m(z_j)+C\Delta z^2+C\Delta t^2\,. \tag{5.184}$$

Similarly, for $m = 2, 4$, we can obtain

$$\Big(\frac{1 + \frac{\Delta z}{2}\lambda_m(\bar{\sigma})}{1 - \frac{\Delta z}{2}\lambda_m(\bar{\sigma})}\Big)^j = e^{\lambda_m(\sigma) j \Delta z} + C\Delta z^2 + C\Delta t^2 \,. \tag{5.185}$$

To prove that (5.184) also holds for $m = 2, 4$, it remains to show

$$\widetilde{y}^{(1)}_{m,0} = \widetilde{y}^{(1)}_m(0) + C\Delta z^2 + C\Delta t^2 \,, \qquad m = 2, 4 \,. \tag{5.186}$$

It suffices to show

$$\bar{U} = U + C\Delta z^2 + C\Delta t^2 \,, \tag{5.187}$$

where $\bar{U}$ is given by (5.177) and U by (5.164). By performing similar calculations as above we find $\bar{V} = V + C\Delta z^2 + C\Delta t^2$. Meanwhile by checking the entries of the matrices M_1 and M_2, defined in (5.163), we obtain $M_1(\bar{\sigma}) = M_1(\sigma) + C\Delta t^2$, $M_2(\bar{\sigma}) = M_2(\sigma) + C\Delta t^2$. Hence (5.177) indicates

$$\begin{aligned}
\bar{U} &= \big(M_1(\sigma) + C\Delta t^2\big)^{-1}\big(M_2(\sigma) + C\Delta t^2\big)\big(V + C\Delta z^2 + C\Delta t^2\big) + \big(M_1(\sigma) + C\Delta t^2\big)^{-1} r \\
&= \big(I + M_1^{-1}(\sigma) C\Delta t^2\big)^{-1} M_1^{-1}(\sigma)\big(M_2(\sigma) + C\Delta t^2\big)\big(V + C\Delta z^2 + C\Delta t^2\big) + \\
&\quad \big(I + M_1^{-1}(\sigma) C\Delta t^2\big)^{-1} M_1^{-1}(\sigma)\, r \\
&= \big(I - M_1^{-1}(\sigma) C\Delta t^2 + C\Delta t^4\big) M_1^{-1}(\sigma)\big(M_2(\sigma) + C\Delta t^2\big)\big(V + C\Delta z^2 + C\Delta t^2\big) + \\
&\quad \big(I - M_1^{-1}(\sigma) C\Delta t^2 + C\Delta t^4\big) M_1^{-1}(\sigma)\, r \\
&= M_1^{-1}(\sigma) M_2(\sigma) V + M_1^{-1}(\sigma)\, r + C\Delta z^2 + C\Delta t^2 \,,
\end{aligned} \tag{5.188}$$

where I is the identity matrix and where we have assumed Δt is small enough so that the spectral radius of the matrix $M_1^{-1}(\sigma) C\Delta t^2$ is smaller than 1. Substitution of (5.164) into (5.188) yields (5.187).

Therefore, we obtain

$$\widetilde{Y}_j = \widetilde{Y}(z_j) + C\Delta z^2 + C\Delta t^2 \; . \tag{5.189}$$

Finally, applying (5.150) and (5.169) to recover the original variables and noting that $Q(\bar{\sigma}) = Q(\sigma) + C\Delta t^2$, we have

$$\begin{aligned} Y_j &= \big(Q(\sigma) + C\Delta t^2\big)\big(\widetilde{Y}(z_j) + C\Delta z^2 + C\Delta t^2\big) \\ &= Y(z_j) + C\Delta z^2 + C\Delta t^2 \; , \end{aligned} \tag{5.190}$$

which completes our proof that the numerical solutions are 2nd-order accurate in both Δt and Δz .

BIBLIOGRAPHY

[1] Acheson, D. J., *Elementary Fluid Dynamics*, Oxford University Press, 1990.

[2] Al-Zanaidi, M. A. and Hui, W. H., Turbulent air flow over water waves - a numerical study, *J. Fluid Mech.*, vol. 148, pp. 225-246, 1984.

[3] Anderson, D. A., Tannehill, J. C. and Pletcher, R. H., *Computational fluid mechanics and heat transfer*, Hemisphere Publishing Corporation, 1984.

[4] Baker, G. R., Berger, K. M. and Johnson, J. T., Numerical studies of the nonlinear interaction between turbulent air flow and sea surface waves, with application to ocean surface wave turbulence, 2001 ITR/AP NSF Grant.

[5] Baker, G. R., Meiron, D. I. and Orszag, S. A., Generalized vortex methods for free surface flow problems, *J. Fluid Mech.*, vol. 123, pp. 477-501, 1982.

[6] Baker, G. R. and Overman, E. A., *The Art of Scientific Computing*, Draft VI, 2000.

[7] Baker, G. R., Wang, J., Johnson, J. T. and Hayslip, A. R., The linear stability at the interface between two immiscible incompressible fluids, in preparation.

[8] Batchelor, G. K., *An introduction to fluid dynamics*, Cambridge University Press, 1967.

[9] Bell, J. B., Colella, P. and Glaz, H. M., A second order projection method for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, vol. 85, pp. 257-283, 1989.

[10] Billingham, J. and King, A. C., *Wave Motion*, Cambridge University Press, 2000.

[11] Brown, D. L., Cortez, R. and Minion, M. L., Accurate projection methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, vol. 168, pp. 464-499, 2001.

[12] Chandrasekhar, S., *Hydrodynamic and Hydromagnetic Stability*, Oxford, Clarendon Press, 1961.

[13] Chorin, A. J., A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.*, vol. 2, pp. 12-26, 1967.

[14] Courant, R., Friedrichs, K. O. and Lewy, H., Uber die partiellen differenzengleichungen der Mathematischen Physik, *Mathematishe Annalen*, vol. 100, pp. 32-74. 1928.

[15] Craik, A. D. D., *Wave Interactions and Fluid Flows*, Cambridge University Press, 1985.

[16] De, S. C., Contribution to the theory of Stokes waves, *Proc. Cambridge Phil. Soc.*, vol. 51, pp. 713-736, 1955.

[17] Douglas, J. and Rachford, H. H., On the numerical solution of heat conduction problems in two or three space variables, *Trans. Amer. Math. Soc.*, vol. 82, pp. 421-439, 1956.

[18] E, Weinan and Liu, J.-G., Projection method I: convergence and numerical boundary layers, *SIAM J. Numer. Anal.*, vol. 32, no. 4, pp. 1017-1057, 1995.

[19] E, Weinan and Liu, J.-G., Projection method II: Godunov-Ryabenki analysis, *SIAM J. Numer. Anal.*, vol. 33, no. 4, pp. 1597-1621, 1996.

[20] Evans, M. E. and Harlow, F. H., The particle-in-cell method for hydrodynamic calculations, Los Alamos Scientific Laboratory Report LA-2139, Los Alamos, New Mexico, 1957.

[21] Fenton, J. D., A fifth-order Stokes theory for steady waves, *J. Waterw. Port Coastal Ocean Eng.*, vol. 111, no. 2, pp. 216-234, 1985.

[22] Ferziger, J. H. and Peric, M., *Computational Methods for Fluid Dynamics*, Springer, 2002.

[23] Frankel, S. P., Convergence rates of iterative treatments of partial differential equations, *Mathematical Tables and Other Aids to Computation*, vol. 4, pp. 65-75, 1950.

[24] Fraysse, V., Giraud, L. and Gratton, S., A set of GMRES routines for real and complex arithmetics, CERFACS Technical Report TR/PA/97/49, France, 1997.

[25] Gent, P. R. and Taylor, P. A., A numerical model of the air flow above water waves, *J. Fluid Mech.*, vol. 77, pp. 105-128, 1976.

[26] Glimm, J., McBryan, O., Menikoff, R. and Sharp, D., Front tracking applied to Rayleigh-Taylor instability, *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 230-251, 1986.

[27] Golub, G. H. and Van Loan, C. F., *Matrix Computations*, The Johns Hopkins University Press, 1996.

[28] Gresho, P. M., Incompressible fluid dynamics: Some fundamental formulation issues, *Annu. Rev. Fluid Mech.*, vol. 23, pp. 413-453, 1991.

[29] Gresho, P. M. and Sani, R. L., On pressure boundary conditions for the incompressible Navier-Stokes equations, *Int. J. Numer. Methods Fluids*, vol. 7, pp. 1111-1145, 1987.

[30] Harlow, F. H. and Welch, J. E., Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids*, vol. 8, pp. 2182-2189, 1965.

[31] Harten, A., High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.*, vol. 49, pp. 357-393, 1983.

[32] Harten, A., Engquist B., Osher S. and Chakravarthy S., Uniformly high order accurate essentially non-oscillatory schemes, III, *J. Comput. Phys.*, vol. 71, pp. 231-303, 1987.

[33] Lomax, H., Pulliam, T. H. and Zingg, D. W., *Fundamentals of Computational Fluid Dynamics*, Springer, 2001.

[34] Hinch, E. J., *Perturbation methods*, Cambridge University Press, 1991.

[35] Hirt, C. W. and Nichols, B. D., Volume of fluid (VOF) method for dynamics of free boundaries, *J. Comput. Phys.*, vol. 39, pp. 201-225, 1981.

[36] Holyer, J. Y., Large amplitude progressive interfacial waves, *J. Fluid Mech.*, vol. 93, pp. 433-448, 1979.

[37] Kim, J. and Moin, P., Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comput. Phys.*, vol. 59, pp. 308-323, 1985.

[38] Kumar, V., Grama, A., Gupta, A. and Karypis, G., *Introduction to Parallel Computing: Design and Analysis of Algorithms*, The Benjamin/Cummings Publishing Company, 1994.

[39] Lax, P. D., Weak solutions of nonlinear hyperbolic equations and their numerical computation, *Comm. Pure Appl. Math.*, vol. 7, pp. 159-193, 1954.

[40] Lax, P. D. and Wendroff B., Systems of conservation laws, *Comm. Pure Appl. Math.*, vol. 13, pp. 217-237, 1960.

[41] Levi Civita, M. T., Determination rigoureuse des ondes permanentes d'ampleur finie, *Math Ann.*, vol. 93, pp. 264-314, 1925.

[42] Longuet-Higgins, M. S. and Cokelet, E. D., The deformation of steep surface waves on water, I: A numerical method of computation, *Proc. R. Soc. Lodon A*, vol. 95, pp. 1-26, 1976.

[43] McCormick, S. F., *Multigrid Methods (Frontiers in Applied Mathematics 3)*, SIAM, Philadelphia, 1987.

[44] Nayfeh, A. H., *Perturbation methods*, John Wiley & Sons, 1973.

[45] Noh, W. F. and Woodward, P. R., SLIC (simple line interface calculation), in *Proc. 5th Int. Conf. Fluid Dyn.*, vol. 59, *Lect. Notes Phys.*, pp. 330-340, Berlin: Springer-Verlag, 1976.

[46] Ockendon, H. and Ockendon, J. R., *Viscous Flow*, Cambridge University Press, 1995.

[47] O'Brien, G. G., Hyman, M. A. and Kaplan, S., A study of the numerical solution of partial differential equations, *J. Math. Phys.*, vol. 29, pp. 223-251, 1950.

[48] Orszag, S. A., Numerical simulation of incompressible flows within simple boundaries I: Galerkin (spectral) representations, *Stud. Appl. Math.*, vol. 50, pp. 293-327, 1971.

[49] Osher, S. and Sethian, J. A., Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton- Jacobi formulations, *J. Comput. Phys.*, vol. 79, pp. 12-49, 1988.

[50] Overman, E. A., *Matlab Overview*, Draft, 2000.

[51] Penney, W. G. and Price, A. T., Some gravity wave problems in the motion of perfect liquids, part II: Finite periodic stationary gravity waves in a perfect liquid, *Phil. Trans. R. Soc. Lond. A*, vol. 244, pp. 251-284, 1952.

[52] Peyret, R. and Taylor, T. D., *Computational Methods for Fluid Flow*, Springer, Berlin, 1983.

[53] Peyret, R., *Spectral Methods for Incompressible flow*, Springer, 2002.

[54] Richtmyer, R. D. and Morton, K. W., *Difference method for initial value problems*, John Wiley & Sons, 1967.

[55] Rogers, S. E., Kwak, D. and Kiris, C., AIAA Paper 89-0463, 1989 (unpublished).

[56] Rottman, J. W., Steep standing waves at a fluid interface, *J. Fluid Mech.*, vol. 124, pp. 283-306, 1982.

[57] Saad, Y. and Schultz, M., GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 856-869, 1986.

[58] Scardovelli, R. and Zaleski, S., Direct numerical simulation of free surface and interfacial flow, *Annu. Rev. Fluid Mech.*, vol. 31, pp. 567-603, 1999.

[59] Schwartz, L. W., Computer extension and analytic continuation of Stokes' expansion for gravity waves, *J. Fluid Mech.*, vol. 62, pp. 553-578, 1974.

[60] Schwartz, L. W. and Fenton, J. D., Strongly Nonlinear Waves, *Ann. Rev. Fluid Mech.*, vol. 14, pp. 39-60, 1982.

[61] Schwartz, L. W. and Whitney, A. K., A semi-analytic solution for nonlinear standing waves in deep water, *J. Fluid Mech.*, vol. 107, pp. 147-171, 1981.

[62] Sethian, J. A., *Level set methods and fast marching methods*, Cambridge University Press, 2000.

[63] Stokes, G. G., On the theory of oscillatory waves, *Trans. Cambridge Philos. Soc.*, vol. 8, pp. 441-455, 1847.

[64] Strikwerda, J. C. and Lee, Y. S., The accuracy of the fractional step method, *SIAM J. Numer. Anal.*, vol. 37, pp. 37-47, 1999.

[65] Tsuji, Y. and Nagata, Y., Stokes' expansion of internal deep water waves to the fifth order, *J. Ocean. Soc. Japan*, vol. 29, pp. 61-69, 1973.

[66] Sullivan, P. P., McWilliams, J. C. and Moeng, C. H., Simulation of turbulent flow over idealized water waves, *J. Fluid Mech.*, vol. 404, pp. 47-85, 2000.

[67] Van Dyke, M., *Perturbation methods in fluid mechanics*, Academic Press, 1964.

[68] Van Kan, J., A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 870-891, 1986.

[69] Vinje, T. and Brevig, P., Numerical simulation of breaking waves, *Adv. Water Resources*, vol. 4, pp. 77-82, 1981.

[70] Wang, J. and Baker, G., A Numerical Approach for Computing Two-Dimensional Viscous Incompressible Flows with Interfaces, in preparation.

[71] Welch, J. E., Harlow, F. H., Shannon, J. P. and Daly, B. J., The MAC method, Los Alamos Scientific Laboratory Report LA-3425, Los Alamos, New Mexico, 1966.

[72] Whitham, G. B., *Linear and Nonlinear Waves*, John Wiley & Sons, 1974.