Journal of Computational Physics 228 (2009) 5470–5489

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A numerical algorithm for viscous incompressible interfacial flows

Jin Wang [a,*], Greg Baker [b]

[a] Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529, USA
[b] Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA

ARTICLE INFO

Article history:
Received 2 January 2009
Received in revised form 2 April 2009
Accepted 21 April 2009
Available online 3 May 2009

Keywords:
Viscous flows
Interfacial motion
Stokes waves

ABSTRACT

We present a new algorithm to numerically simulate two-dimensional viscous incompressible flows with moving interfaces. The motion is updated in time by using the backward difference formula through an iterative procedure. At each iteration, the pseudo-spectral technique is applied in the horizontal direction. The resulting semi-discretized equations constitute a boundary value problem in the vertical coordinate which is solved by decoupling growing and decaying solutions. Numerical tests justify that this method achieves fully second-order accuracy in both the temporal variable and vertical coordinate. As an application of this algorithm, we study the motion of Stokes waves in the presence of viscosity. Our numerical results are consistent with the recently published asymptotic solution for Stokes waves in slightly viscous fluids.

1. Introduction

Viscous incompressible flows with interfaces occur in a wide variety of physical phenomena and a large number of technological processes. Consequently, numerical simulations to these problems have been making significant strides in many disciplines in science and engineering such as fluid dynamics, geophysics, oceanography, material science and mechanical engineering [8,25]. Mathematically, the motion is governed by the incompressible Navier–Stokes equations together with interfacial conditions [6]. There are two major difficulties associated with the numerical study of such problems. First, the incompressibility condition has to be satisfied (somehow in an implicit manner) at all times [15]. Second, 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 is crucial to have an accurate representation of it.

Several numerical methods have been developed for tracking or capturing the interfacial motion. Among the most popular ones are the volume-of-fluid (VOF), level set and boundary integral approaches. The VOF methods have been in use for several decades. Early work includes the SLIC algorithm of Noh and Woodward [27] and the SOLA-VOF algorithm of Hirt and Nicols [17]. Since then, significant progress has been made on VOF methods and a review of recent work can be found in [32]. In the VOF formulation, a volume fraction function C is defined and it satisfies an advection equation. At each time, the values of C 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 in general have difficulties in resolving the small structure of the interface, and are not good at capturing the fine-scale boundary layers [35]. The level set approach was first proposed by Osher and Sethian [29] and has since been widely applied to many interfacial/free-surface problems [35]. In these methods, a level set function ϕ is introduced such that its initial value gives the shortest distance between each point and the initial interface. The function ϕ then evolves in response to the propagation of the inter-

* Corresponding author.
E-mail addresses: j3wang@odu.edu (J. Wang), baker@math.ohio-state.edu (G. Baker).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.04.025

face and, at anytime, the zero level set $\phi = 0$ gives exactly the location of the interface. These methods, like the VOF, 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, without further modifications, have inherent numerical dissipation which will smooth the interface and lead to non-physical loss of mass. The boundary integral methods were developed for computing inviscid potential flows, and notable work in this category was made by Longuet-Higgins and Cokelet [26], Vinje and Brevig [39], Baker et al. [5], etc. In the boundary integral formulation, Laplace's equation is solved by using Green's functions, leading to Fredholm integral equations of the second kind. The dynamic and kinematic surface conditions are integrated to update the interface at each time. A distinct advantage of these methods is that the space dimension of the problem is reduced by one. Hence they offer an efficient way for the computation of inviscid and irrotational flows. Unfortunately, these methods are not applicable to general viscous motion. In addition, there are some other well known methods such as marker-and-cell [16,43], front tracking [14], phase field [37] and immersed interface methods [19,22,24], which have also achieved much success in multi-phase flow simulations. All these methods have their own strength and weakness, and a perfect approach does not yet exist.

The main purpose of the current work is to perform an accurate study on interfacial motion between two *slightly* viscous fluids, such as air and water. Despite the ubiquitous presence and importance of air–water systems, understanding of the viscous effects on water waves remains limited at present, due to the nonlinear phenomenon implicit in both air flow and water wave evolution [4]. The challenge in numerical study of such viscous effects is the demand for an algorithm which captures the many spatial scales on the water surface and which does not introduce numerical smoothing that can mask the true effects of viscosity. Those methods summarized above do not appear optimal to meet this demand.

In this paper, we present a numerical algorithm with high accuracy to simulate the two-dimensional viscous incompressible flows with moving interfaces. Since our focus is the study of viscous effects, we may reasonably assume that the interface remains single-valued and no dramatic change occurs for its topology. We then introduce new coordinates, referred to as logical coordinates, to map the interface into a coordinate line which enables us to work on a rectangular domain instead of the deformed geometry. The backward difference formula (BDF) [1] combined with an iterative procedure are 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 pseudo-spectral technique [28,30] are applied under the assumption that the solutions are periodic in X. Then we write the semi-discretized equations as a first-order ODE system with respect to the vertical coordinate, Z. Together with interfacial conditions and far-field boundary conditions, they form a boundary value problem in Z which is solved by decoupling growing and decaying solutions. The incompressibility condition is treated as one equation in the ODE system, so that it is automatically satisfied at each time step. The method achieves uniform order of accuracy for the velocities, the pressure and the interface profile: second-order for the temporal variable and vertical coordinate, and spectral accuracy for the horizontal coordinate. In addition, this algorithm can be easily applied in parallel computation, making large-scale simulations possible.

As an application of this numerical approach, we study the viscous effects on a special type of traveling waves, the Stokes waves [36], in the present paper. Stokes waves are nonlinear, periodic and steady progressive free-surface or interfacial motions. The study of Stokes waves in inviscid fluids is one of the oldest in the field of mathematical fluid mechanics and a large body of work has been devoted to it. Stokes [36] was the first to systematically study the properties of surface water waves by using the technique of series expansion named the Stokes' expansion. Levi Civita [23] proved the convergence of Stokes' expansion for sufficiently small waves. De [11] published a fifth-order solution to general water depth. Schwartz [33,34] calculated the expansion to extremely high orders by using the digital computer to perform the coefficient arithmetic. Tsuji and Nagata [38] and Holyer [18] extended the Stokes' expansion to the interfacial waves moving between two fluids of different densities. All the aforementioned work was concerned with inviscid fluids. Recently, Wang [40] proposed an expansion form, based on an asymptotic analysis, for Stoke waves in the presence of small viscosity. The asymptotic form shows that the effects of viscosity can be approximated by the decay of the expansion parameter in the inviscid Stokes' expansion. This result is numerically verified in the present work. By computing the motion of Stokes waves sufficiently in time, we are able to conduct a careful numerical analysis of the viscous effects on wave motion. In particular, the decay rates for principal wave modes and the relationship between different modes in the course of viscous damping are thoroughly discussed. To our best knowledge, such direct numerical simulation result on viscous Stokes waves is the first of its kind.

The remaining part of this paper is organized as follows. In Section 2, we describe in detail the numerical method for computing two-dimensional viscous incompressible flows with interfaces. In Section 3, we provide the computational results to demonstrate the algorithm. Two test examples serve for the numerical verification of accuracy, followed by a brief discussion of the parallel implementation of the numerical algorithm. Then the simulation results for viscous Stokes waves are presented and thoroughly discussed. Finally, conclusions are made in Section 4.

2. Numerical algorithm

2.1. Basic formulation

Let us denote the spatial 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 motion in each of the two fluids is described by the incompressible Navier–Stokes equations

$$\rho u_t + \rho u u_x + \rho w u_z = -P_x + \mu(u_{xx} + u_{zz}), \tag{1}$$
$$\rho w_t + \rho u w_x + \rho w w_z = -P_z + \mu(w_{xx} + w_{zz}), \tag{2}$$
$$u_x + w_z = 0, \tag{3}$$

where $P = p + \rho g z$ is referred to as the hydrodynamic pressure. Eqs. (1)–(3) hold in both the upper and lower fluids, and their solutions are connected through the interfacial conditions, to be provided in Eqs. (6)–(8). In addition, we will assume that solutions are periodic in the horizontal direction, and exponentially decay away from the interface.

Let us represent the interface in the form

$$(x, z) = (x, h(x, t)). \tag{4}$$

The profile of h is determined by the kinematic condition

$$h_t + u^{(I)} h_x = w^{(I)}, \tag{5}$$

where $u^{(I)}$, $w^{(I)}$ are the interfacial velocity components. Due to the continuity of velocity at the interface, we have

$$u^{(1)} = u^{(2)} = u^{(I)}, \quad w^{(1)} = w^{(2)} = w^{(I)}, \tag{6}$$

where the superscripts (1) and (2) distinguish the upper and lower domains. Moreover, the balance of stresses provides two more interfacial conditions [6] and, in two-dimensional case, they yield

$$(h_x^2 - 1)\left[\mu^{(1)}\left(u_z^{(1)} + w_x^{(1)}\right) - \mu^{(2)}\left(u_z^{(2)} + w_x^{(2)}\right)\right] + 2h_x\left[\mu^{(1)}\left(u_x^{(1)} - w_z^{(1)}\right) - \mu^{(2)}\left(u_x^{(2)} - w_z^{(2)}\right)\right] = 0, \tag{7}$$
$$(P^{(1)} - P^{(2)}) - gh(\rho^{(1)} - \rho^{(2)}) + h_x\left[\mu^{(1)}\left(u_z^{(1)} + w_x^{(1)}\right) - \mu^{(2)}\left(u_z^{(2)} + w_x^{(2)}\right)\right] - 2\left[\mu^{(1)} w_z^{(1)} - \mu^{(2)} w_z^{(2)}\right] - \gamma\kappa = 0, \tag{8}$$

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

$$\kappa = \frac{h_{xx}}{(1 + h_x^2)^{3/2}}. \tag{9}$$

2.2. Mapped equations

The evolving interface $h(x, t)$ between the two fluids makes the design of accurate 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 us introduce the new coordinates, (X, Z, τ), through the mapping [41]

$$x = X, \tag{10}$$
$$z = F(X, Z, \tau), \tag{11}$$
$$t = \tau, \tag{12}$$

where

$$F(X, Z, \tau) \triangleq \begin{cases} Z + h(X, \tau)\exp(-\alpha Z), & Z \geqslant 0, \\ Z + h(X, \tau)\exp(\alpha Z), & Z \leqslant 0, \end{cases} \tag{13}$$

where $\alpha > 0$ is a constant which can be used to adjust the grid spacing near the interface. Clearly, the coordinate line $Z = 0$ corresponds to the location of the interface $z = h(x, t)$. When far from the interface, Z is relaxing exponentially to the physical coordinate z so that the far-field boundary conditions can be easily handled.

If we define

$$G_0 = \frac{F_\tau}{F_Z}, \quad G_1 = \frac{F_X}{F_Z}, \quad G_3 = \frac{1}{F_Z}, \tag{14}$$

then the transformed derivatives can be calculated by

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - G_0 \frac{\partial}{\partial Z}, \tag{15}$$
$$\frac{\partial}{\partial x} = \frac{\partial}{\partial X} - G_1 \frac{\partial}{\partial Z}, \tag{16}$$
$$\frac{\partial}{\partial z} = G_3 \frac{\partial}{\partial Z}, \tag{17}$$
$$\frac{\partial^2}{\partial x^2} = \frac{\partial^2}{\partial X^2} + (G_1)^2 \frac{\partial^2}{\partial Z^2} - 2G_1 \frac{\partial^2}{\partial X \partial Z} + [G_1 (G_1)_Z - (G_1)_X] \frac{\partial}{\partial Z}, \tag{18}$$
$$\frac{\partial^2}{\partial z^2} = (G_3)^2 \frac{\partial^2}{\partial Z^2} + G_3 (G_3)_Z \frac{\partial}{\partial Z}. \tag{19}$$

Let us further define

$$g_2 = (G_1)^2 + (G_3)^2, \quad g_3 = -2G_1, \quad g_4 = G_1\frac{\partial G_1}{\partial Z} + G_3\frac{\partial G_3}{\partial Z} - \frac{\partial G_1}{\partial X}. \tag{20}$$

Then we can write the Laplacian in new variables as

$$\mathcal{L} \triangleq \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} = \frac{\partial^2}{\partial X^2} + g_2\frac{\partial^2}{\partial Z^2} + g_3\frac{\partial^2}{\partial X \partial Z} + g_4\frac{\partial}{\partial Z}. \tag{21}$$

We note that the coefficients G_i $(i = 0, 1, 3)$, g_i $(i = 2, 3, 4)$ are different in the upper and lower domains.

Now we substitute the transformation rules (15)–(21) for the derivatives directly into the basic equations (1)–(8) to obtain

$$u_\tau - G_0 u_Z + u(u_X - G_1 u_Z) + wG_3 u_Z = -\frac{1}{\rho}P_X + \frac{1}{\rho}G_1 P_Z + \nu\mathcal{L}\{u\}, \tag{22}$$

$$w_\tau - G_0 w_Z + u(w_X - G_1 w_Z) + wG_3 w_Z = -\frac{1}{\rho}G_3 P_Z + \nu\mathcal{L}\{w\}, \tag{23}$$

$$u_X - G_1 u_Z + G_3 w_Z = 0, \tag{24}$$

where $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity. The kinematic condition becomes

$$h_\tau + u^{(l)} h_X = w^{(l)}. \tag{25}$$

Though there is no change to the velocity interfacial conditions (6), the two stress conditions are now:

$$\begin{aligned}&\mu^{(1)}\left(G_3^{(1)}u_Z^{(1)} + w_X^{(1)}\right) - \mu^{(2)}\left(G_3^{(2)}u_Z^{(2)} + w_X^{(2)}\right) + \left(\frac{4h_X}{h_X^2 - 1} + \frac{G_1^{(1)}}{G_3^{(1)}}\right)\mu^{(1)}\left(u_X^{(1)} - G_1^{(1)}u_Z^{(1)}\right)\\ &\quad - \left(\frac{4h_X}{h_X^2 - 1} + \frac{G_1^{(2)}}{G_3^{(2)}}\right)\mu^{(2)}\left(u_X^{(2)} - G_1^{(2)}u_Z^{(2)}\right) = 0,\end{aligned} \tag{26}$$

$$(P^{(1)} - P^{(2)}) + \left(2 - \frac{4h_X^2}{h_X^2 - 1}\right)\left[\mu^{(1)}\left(u_X^{(1)} - G_1^{(1)}u_Z^{(1)}\right) - \mu^{(2)}\left(u_X^{(2)} - G_1^{(2)}u_Z^{(2)}\right)\right] = gh(\rho^{(1)} - \rho^{(2)}) + \gamma\kappa. \tag{27}$$

We note that in order to obtain (26) and (27), we have eliminated $w_z^{(1)}$ and $w_z^{(2)}$ in Eqs. (7) and (8) by using the incompressibility condition (3). It will be clear soon that such eliminations are necessary to facilitate our numerical formulation.

Before we describe the numerical method in detail, it is best to first summarize the overall strategy. The equations are written in the form of linear terms and nonlinear terms separately. The second-order backward difference formula (BDF) [1] is applied to update the motion in time. The method is fully implicit and so require the solution of a nonlinear system of equations for the unknowns at the new time level. The linear terms thus provide a simple iterative procedure. At each iterate, the Fourier transform and the pseudo-spectral technique [28,30] are applied in the horizontal direction X, which possesses periodicity, to achieve spectral accuracy in X. Then a linear system of first-order ODEs with respect to the vertical coordinate, Z, is solved.

As a start, we describe the separation of the equations into linear and nonlinear parts. Let us 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 as well as the mapping-associated terms will be collected to the right-hand sides. We have, first of all,

$$u_Z - q = 0. \tag{28}$$

In what follows we replace u_Z by q. From the momentum equation (22) we obtain

$$u_\tau + \frac{1}{\rho}P_X - \nu(u_{XX} + q_Z) = R_u \triangleq G_0 q + \frac{1}{\rho}G_1 P_Z - [u(u_X - G_1 q) + wG_3 q] + \nu[(g_2 - 1)q_Z + g_3 q_X + g_4 q]. \tag{29}$$

Next, the incompressibility condition (24) gives

$$u_X + w_Z = R_c \triangleq G_1 q + (1 - G_3)w_Z. \tag{30}$$

Finally, from the momentum equation (23) we have

$$w_\tau + \frac{1}{\rho}P_Z - \nu(w_{XX} + w_{ZZ}) = G_0 w_Z + \frac{1}{\rho}(1 - G_3)P_Z - [u(w_X - G_1 w_Z) + wG_3 w_Z] + \nu[g_2 w_{ZZ} + g_3 w_{XZ} + g_4 w_Z - w_{ZZ}]. \tag{31}$$

We note that in the linear case, the incompressibility condition implies

$$w_{ZZ} + q_X = 0. \tag{32}$$

Adding $\nu(w_{ZZ} + q_X)$ to both sides of (31), we obtain

$$w_\tau + \frac{1}{\rho}P_Z - \nu(w_{XX} - q_X) = R_w \triangleq G_0 w_Z + \frac{1}{\rho}(1 - G_3)P_Z - [u(w_X - G_1 w_Z) + wG_3 w_Z] + \nu[g_2 w_{ZZ} + g_3 w_{XZ} + g_4 w_Z + q_X]. \tag{33}$$

We will use (33) to replace (31) as one of the governing equations. Similarly, the two stress conditions (26) and (27) may be expressed as

$$\mu^{(1)}\left(q^{(1)} + w_X^{(1)}\right) - \mu^{(2)}\left(q^{(2)} + w_X^{(2)}\right) = S_1 \triangleq \mu^{(1)}\left(1 - G_3^{(1)}\right)q^{(1)} - \mu^{(2)}\left(1 - G_3^{(2)}\right)q^{(2)} - \left(\frac{4h_X}{h_X^2 - 1} + \frac{G_1^{(1)}}{G_3^{(1)}}\right)\mu^{(1)}\left(u_X^{(1)} - G_1^{(1)}q^{(1)}\right) + \left(\frac{4h_X}{h_X^2 - 1} + \frac{G_1^{(2)}}{G_3^{(2)}}\right)\mu^{(2)}\left(u_X^{(2)} - G_1^{(2)}q^{(2)}\right), \tag{34}$$

$$P^{(1)} - P^{(2)} + 2\left(\mu^{(1)}u_X^{(1)} - \mu^{(2)}u_X^{(2)}\right) = S_2 \triangleq gh(\rho^{(1)} - \rho^{(2)}) + \gamma\kappa + 2\left(\mu^{(1)}G_1^{(1)}q^{(1)} - \mu^{(2)}G_1^{(2)}q^{(2)}\right) + \frac{4h_X^2}{h_X^2 - 1}\left[\mu^{(1)}\left(u_X^{(1)} - G_1^{(1)}q^{(1)}\right) - \mu^{(2)}\left(u_X^{(2)} - G_1^{(2)}q^{(2)}\right)\right], \tag{35}$$

while the other two interfacial conditions (6) and the kinematic condition (25) remain unchanged.

2.3. Time marching

Due to the strong nonlinearity of the problem and the incompressibility constraint on the velocity field, an explicit time-marching method, such as the Adams-Bashforth [31], would easily cause numerical instability or produce non-physical solution [15]. Hence, implicit methods and fractional step techniques are commonly used in the numerical simulation of the incompressible Navier–Stokes equations. The projection approach [7,9,12], for example, is one of the most popular fractional step methods and has been widely applied to compute Navier–Stokes flows. In the projection formulation, an intermediate velocity is first calculated which does not necessarily satisfy the incompressibility condition; it is then projected into a divergence-free field to obtain the correct velocity. However, when applying the projection method to our two-phase flow problem, we encountered difficulty in handling the physical interfacial conditions. We have, therefore, adopted a fully implicit method, the backward difference formula (BDF), accomplished in an iterative manner. The details are described below.

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 $n+1$, $\{h^{n+1}, u^{n+1}, q^{n+1}, w^{n+1}, P^{n+1}\}$.

The BDF method applied to Eqs. (28)–(30) and (33) yields

$$u_Z^{n+1} - q^{n+1} = 0, \tag{36}$$

$$\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta\tau} + \frac{1}{\rho}P_X^{n+1} - \nu(u_{XX}^{n+1} + q_Z^{n+1}) = R_u^{n+1}, \tag{37}$$

$$w_Z^{n+1} + u_X^{n+1} = R_c^{n+1}, \tag{38}$$

$$\frac{3w^{n+1} - 4w^n + w^{n-1}}{2\Delta\tau} + \frac{1}{\rho}P_Z^{n+1} - \nu(w_{XX}^{n+1} - q_X^{n+1}) = R_w^{n+1}, \tag{39}$$

where R_u, R_c and R_w are defined in Eqs. (29), (30) and (33), respectively, and contain all the nonlinear terms. In addition, the evolution of the interface is approximated by

$$\frac{3h^{n+1} - 4h^n + h^{n-1}}{2\Delta\tau} = (w^{(I)} - u^{(I)}h_X)^{n+1}, \tag{40}$$

while the interfacial conditions are approximated by

$$(u^{(1)} - u^{(2)})^{n+1} = 0, \tag{41}$$

$$\left(\mu^{(1)}\left(q^{(1)} + w_X^{(1)}\right) - \mu^{(2)}\left(q^{(2)} + w_X^{(2)}\right)\right)^{n+1} = S_1^{n+1}, \tag{42}$$

$$(w^{(1)} - w^{(2)})^{n+1} = 0, \tag{43}$$

$$\left(P^{(1)} - P^{(2)} + 2\left(\mu^{(1)}u_X^{(1)} - \mu^{(2)}u_X^{(2)}\right)\right)^{n+1} = S_2^{n+1}, \tag{44}$$

where S_1, S_2 are defined in Eqs. (34) and (35).

Unfortunately, the equations for the updated solution are nonlinear and challenging to solve in general. A simple iterative method 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, where

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

The next iterate is obtained as follows.

First we update the interface h by

$$\frac{3h^{(n,m)} - 4h^n + h^{n-1}}{2\Delta\tau} = \left(w^{(l)} - u^{(l)} h_X\right)^{(n,m-1)}. \tag{45}$$

Once $h^{(n,m)}$ is known, the mappings (10)–(12) are evaluated and the 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

$$u_Z^{(n,m)} - q^{(n,m)} = 0, \tag{46}$$

$$\frac{3u^{(n,m)} - 4u^n + u^{n-1}}{2\Delta\tau} + \frac{1}{\rho} P_X^{(n,m)} - \nu\left(u_{XX}^{(n,m)} + q_Z^{(n,m)}\right) = R_u^{(n,m-1)}, \tag{47}$$

$$w_Z^{(n,m)} + u_X^{(n,m)} = R_c^{(n,m-1)}, \tag{48}$$

$$\frac{3w^{(n,m)} - 4w^n + w^{n-1}}{2\Delta\tau} + \frac{1}{\rho} P_Z^{(n,m)} - \nu\left(w_{XX}^{(n,m)} - q_X^{(n,m)}\right) = R_w^{(n,m-1)}, \tag{49}$$

with the corresponding interfacial conditions

$$(u^{(1)} - u^{(2)})^{(n,m)} = 0, \tag{50}$$

$$\left(\mu^{(1)}\left(q^{(1)} + w_X^{(1)}\right) - \mu^{(2)}\left(q^{(2)} + w_X^{(2)}\right)\right)^{(n,m)} = S_1^{(n,m-1)}, \tag{51}$$

$$(w^{(1)} - w^{(2)})^{(n,m)} = 0, \tag{52}$$

$$\left(P^{(1)} - P^{(2)} + 2\left(\mu^{(1)} u_X^{(1)} - \mu^{(2)} u_X^{(2)}\right)\right)^{(n,m)} = S_2^{(n,m-1)}. \tag{53}$$

We set the stopping criterion of the iterations as

$$\frac{\|h^{(n,m)} - h^{(n,m-1)}\|_2}{\|h^{(n,m-1)}\|_2} + \frac{\|u^{(n,m)} - u^{(n,m-1)}\|_2}{\|u^{(n,m-1)}\|_2} + \frac{\|q^{(n,m)} - q^{(n,m-1)}\|_2}{\|q^{(n,m-1)}\|_2} + \frac{\|w^{(n,m)} - w^{(n,m-1)}\|_2}{\|w^{(n,m-1)}\|_2} + \frac{\|P^{(n,m)} - P^{(n,m-1)}\|_2}{\|P^{(n,m-1)}\|_2} < E, \tag{54}$$

where E is some tolerance and where the L_2-norm $\|\cdot\|_2$ is taken at all the grid points. Once (54) 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 BDF scheme to (45)–(49) and treated all the interfacial conditions in a fully implicit manner.

Next, we turn to the spatial discretization which allows the construction of the mth iterate.

2.4. The Fourier transform

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. Based on the method of marching forward in time, we apply the discrete Fourier transform at each iteration to (45)–(49) as well as the corresponding interfacial conditions. For the left-hand sides of these equations, we simply replace $\frac{\partial}{\partial x}$ by ik, $\frac{\partial^2}{\partial x^2}$ by $-k^2$ and the physical variables $\{h, u, q, w, P\}$ by their kth Fourier coefficients, where $k = -K, -K+1, \ldots, 0, 1, \ldots, K-1$. For the right-hand sides containing all the nonlinear and mapping-associated terms, we carry out the well-known pseudo-spectral approach [30] which consists of three 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 kth Fourier coefficient of the physical variables. After applying the Fourier transform, the iterative formula (46)–(49) can be written as a four-by-four linear system of first-order ODEs with respect to Z

$$\frac{d}{dZ} Y_k = B_k Y_k + R_k, \tag{55}$$

where

$$Y_k \triangleq \left(u_k^{(n,m)}, q_k^{(n,m)}, w_k^{(n,m)}, P_k^{(n,m)}\right)^T,$$

$$B_k \triangleq \begin{bmatrix} 0 & 1 & 0 & 0 \\ k^2 + \frac{3}{2\nu\Delta\tau} & 0 & 0 & \frac{1}{\rho\nu} ik \\ -ik & 0 & 0 & 0 \\ 0 & -\rho\nu ik & -\rho\left(\nu k^2 + \frac{3}{2\Delta\tau}\right) & 0 \end{bmatrix}, \tag{56}$$

and where R_k is the kth Fourier coefficient of the following vector which contains all the explicit terms,

$$\begin{pmatrix} 0 \\ \frac{-1}{\nu}\left[R_u^{(n,m-1)} + \frac{2u^n}{\Delta\tau} - \frac{u^{n-1}}{2\Delta\tau}\right] \\ R_c^{(n,m-1)} \\ \rho\left[R_w^{(n,m-1)} + \frac{2w^n}{\Delta\tau} - \frac{w^{n-1}}{2\Delta\tau}\right] \end{pmatrix}. \tag{57}$$

The interfacial conditions for the system (55) are given by

$$J_k^{(1)} Y_k^{(1)} - J_k^{(2)} Y_k^{(2)} = r_k, \tag{58}$$

where

$$J_k \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mu & ik\mu & 0 \\ 0 & 0 & 1 & 0 \\ 2ik\mu & 0 & 0 & 1 \end{bmatrix}, \quad r_k \triangleq \begin{pmatrix} 0 \\ \left(S_1^{(n,m-1)}\right)_k \\ 0 \\ \left(S_2^{(n,m-1)}\right)_k \end{pmatrix}. \tag{59}$$

We must now construct numerical solutions to the ODE system (55) subject to the interfacial conditions (58) and the far-field conditions (decaying solutions).

2.5. The boundary value problem

We notice that the four eigenvalues of the matrix B_k in (56) are given by

$$\lambda_1 = k, \quad \lambda_2 = -k, \quad \lambda_3 = \psi(k), \quad \lambda_4 = -\psi(k), \tag{60}$$

where

$$\psi(k) \triangleq \sqrt{k^2 + \frac{3}{2\nu\Delta\tau}}. \tag{61}$$

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 (55) prior to applying the shooting method. The eigenvectors associated with the four eigenvalues in (60) are

$$e_1 = \begin{pmatrix} k \\ k^2 \\ -ik \\ \frac{3\rho i}{2\Delta\tau} \end{pmatrix}, \quad e_2 = \begin{pmatrix} 1 \\ -k \\ i \\ \frac{3\rho i}{2k\Delta\tau} \end{pmatrix}, \quad e_3 = \begin{pmatrix} \psi(k) \\ \psi^2(k) \\ -ik \\ 0 \end{pmatrix}, \quad e_4 = \begin{pmatrix} 1 \\ -\psi(k) \\ \frac{ik}{\psi(k)} \\ 0 \end{pmatrix} \tag{62}$$

for $k \neq 0$, and

$$e_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 0 \\ -\frac{\Delta\tau}{2\rho} \\ 0 \end{pmatrix}, \quad e_3 = \begin{pmatrix} \psi(0) \\ \psi^2(0) \\ 0 \\ 0 \end{pmatrix}, \quad e_4 = \begin{pmatrix} 1 \\ -\psi(0) \\ 0 \\ 0 \end{pmatrix} \tag{63}$$

for $k = 0$. Define the matrix

$$Q_k = (e_1, e_2, e_3, e_4) \tag{64}$$

and introduce the transformation

$$Y_k = Q_k \widetilde{Y}_k. \tag{65}$$

Then the system (55) becomes

$$\frac{d}{dZ}\widetilde{Y}_k = \widetilde{B}_k \widetilde{Y}_k + \widetilde{R}_k, \tag{66}$$

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

$$\widetilde{B}_k \triangleq Q_k^{-1} B_k Q_k = \begin{cases} \begin{pmatrix} k & & & \\ & -k & & \\ & & \psi(k) & \\ & & & -\psi(k) \end{pmatrix}, & k \neq 0, \\ \begin{pmatrix} 0 & 1 & & \\ & 0 & & \\ & & \psi(0) & \\ & & & -\psi(0) \end{pmatrix}, & k = 0. \end{cases} \tag{67}$$

At the interface, the jump condition (58) becomes

$$J_k^{(1)} Q_k^{(1)} \widetilde{Y}_k^{(1)} - J_k^{(2)} Q_k^{(2)} \widetilde{Y}_k^{(2)} = r_k, \tag{68}$$

where J_k is defined in (59) and Q_k in (64).

Now the system (66) is reduced to four scalar equations in the form of

$$\frac{d}{dZ}\tilde{y} = \lambda \tilde{y} + \tilde{r} \tag{69}$$

and we apply the trapezoidal rule, which is second-order accurate in ΔZ, to each of them to obtain

$$\left(1 - \frac{\Delta Z}{2}\lambda\right)\tilde{y}_{j+1} - \left(1 + \frac{\Delta Z}{2}\lambda\right)\tilde{y}_j = \frac{\Delta Z}{2}(\tilde{r}_j + \tilde{r}_{j+1}). \tag{70}$$

To ensure numerical stability, we have to consider (70) in two different cases. If $\lambda < 0$, we use

$$\tilde{y}_{j+1} = \frac{1}{(1 - \frac{\Delta Z}{2}\lambda)}\left[\left(1 + \frac{\Delta Z}{2}\lambda\right)\tilde{y}_j + \frac{\Delta Z}{2}(\tilde{r}_j + \tilde{r}_{j+1})\right], \quad j = -J, -J+1, \ldots, -1. \tag{71}$$

That means we start from the bottom, where $\tilde{y}_{-J}$ is known, and apply (71) with increasing j until we reach the interface $j = 0$. If $\lambda > 0$, we use

$$\tilde{y}_j = \frac{1}{(1 + \frac{\Delta Z}{2}\lambda)}\left[\left(1 - \frac{\Delta Z}{2}\lambda\right)\tilde{y}_{j+1} - \frac{\Delta Z}{2}(\tilde{r}_j + \tilde{r}_{j+1})\right], \quad j = J-1, J-2, \ldots, 0. \tag{72}$$

That means we start from the top, where $\tilde{y}_J$ is known, and apply (72) 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 algebraic 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 (67) we see the two positive eigenvalues correspond to the first and the third components of $\widetilde{Y}_k$, denoted by $\tilde{y}_1$ and $\tilde{y}_3$, respectively. They are determined by the recursion (72). The two negative eigenvalues correspond to the second and the fourth components of $\widetilde{Y}_k$, denoted by $\tilde{y}_2$ and $\tilde{y}_4$, respectively. They are determined by the recursion (71). At the interface $j = 0$, (68) gives

$$J_k^{(1)} Q_k^{(1)} \begin{pmatrix} \tilde{y}_1^{(1)} \\ \tilde{y}_2^{(1)} \\ \tilde{y}_3^{(1)} \\ \tilde{y}_4^{(1)} \end{pmatrix}_{j=0} - J_k^{(2)} Q_k^{(2)} \begin{pmatrix} \tilde{y}_1^{(2)} \\ \tilde{y}_2^{(2)} \\ \tilde{y}_3^{(2)} \\ \tilde{y}_4^{(2)} \end{pmatrix}_{j=0} = r_k, \tag{73}$$

where $\left\{\tilde{y}_1^{(1)}, \tilde{y}_3^{(1)}, \tilde{y}_2^{(2)}, \tilde{y}_4^{(2)}\right\}_{j=0}$ are already known. Hence, (73) forms a four-by-four linear system for the unknowns $\left\{\tilde{y}_2^{(1)}, \tilde{y}_4^{(1)}, \tilde{y}_1^{(2)}, \tilde{y}_3^{(2)}\right\}_{j=0}$. Once they are solved, we can continue the computation for $\tilde{y}_1$, $\tilde{y}_3$ by following (72) with the indices shifted to: $j = -1, -2, \ldots, -J$, and for $\tilde{y}_2$, $\tilde{y}_4$ by following (71) with the indices shifted to: $j = 0, 1, \ldots, J-1$.

The situation is a little different in the case $k = 0$ since we have two zero eigenvalues, corresponding to $\tilde{y}_1$ and $\tilde{y}_2$. Moreover, the calculation of $\tilde{y}_1$ requires $\tilde{y}_2$ (see Eq. (67)). 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.

3. Numerical results

In this section, we present computational results by applying the numerical method described before. We will first use two examples to verify the order of accuracy. The tests will be conducted on the order of the time marching and

the discretization in the Z-direction, as there is little doubt about the spectral accuracy in the X-direction where the Fourier transform is applied. Next, we will briefly discuss the parallel implementation of the algorithm, and present some test results for the performance. Then, we will focus our attention on the numerical study of viscous effects on Stokes waves. Simulation results for Stokes waves with various amplitudes and viscosities will be presented and carefully discussed.

3.1. One-phase flow test

As the first example, we test the numerical accuracy by computing exact solution to the Navier–Stokes equations in the one-fluid case. An exact solution to the incompressible Navier–Stokes equations (1)–(3) which is periodic in the horizontal direction and exponentially decaying in the vertical direction, is given by

$$\begin{aligned} u &= t\sin(2x)e^{-2z}, \\ w &= t\cos(2x)e^{-2z}, \\ P &= \frac{\rho}{2}(\cos(2x)e^{-2z} - t^2e^{-4z}), \end{aligned} \tag{74}$$

We define the spatial domain as

$$\{(x,z)|0 \leqslant x \leqslant 2\pi,\ h(x,t) \leqslant z \leqslant 1\}, \tag{75}$$

where the wavy bottom, h, is set as

$$h(x,t) = 0.1\sin(x-t). \tag{76}$$

Although there is no interface in this case, the mappings (10)–(12) can be readily formed (for $Z \geqslant 0$ only) based on the moving boundary $h(x,t)$. Consequently, all parts except the interfacial conditions, in our numerical formulation can be readily applied. Hence, the mapped equations, the Navier–Stokes solver and the formulation as a boundary value problem in Z will be thoroughly tested, whereas 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 (74).

We conduct this test (as well as next one) on a 2.4 GHz Xeon dual-processor workstation. We perform the computation for $\rho = 1.0,\ \mu = 0.313$ (corresponding to a Reynolds number of 100) and run the codes until $\tau = 0.4$. We use 32 points in the X-direction so that the errors associated with ΔX is negligible. Let N be the number of time steps and J the number of points in the Z-direction. We make consecutive runs with N and J doubled each time, and the results are presented in Table 1. Here $E(u,N,J)$ 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 $R(u,N,J)$ denotes the quantity

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

which indicates the order of accuracy for u. 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 second-order convergence in both $\Delta\tau$ and ΔZ for the velocity and pressure.

3.2. Two-phase linear flow test

Due to the presence of the nonlinear interfacial conditions (6)–(8), an analytical form of solutions is not available for general two-phase flows. Nevertheless, there are exact solutions for the corresponding linearized problem. If we use a linearized solution as the initial condition and set the amplitude of the interface h small enough, then the influence of the nonlinear terms in both the governing equations and interfacial conditions becomes unimportant since they are in the order of $O(h^2)$. Consequently, we expect the solution of our nonlinear problem will be very close to that of the linear one. Hence, in this example we use a linearized solution as the reference to test the accuracy. At the same time, the numerical treatment of the interfacial conditions will be tested at least in the linear level.

Table 1
One-phase flow test results. N and J are the numbers of time steps and spatial points, respectively. $E(\cdot,N,J)$ denotes the L_2-norm of the errors and $R(\cdot,N,J)$ indicates the order of convergence.

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)	1.512×10^{-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)

Solutions for the linear motion of interfacial flows are available in [10], and most recently in [42]. They take the form of

$$\begin{pmatrix} u_k \\ w_k \\ P_k \\ h_k \end{pmatrix} = \begin{pmatrix} u_k^0 \\ w_k^0 \\ P_k^0 \\ h_k^0 \end{pmatrix} e^{\sigma(k)t}, \tag{77}$$

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

$$\begin{aligned} &\left[\rho^{(1)}\sqrt{\nu^{(1)}}\left(\Omega^{(1)}+\sqrt{\nu^{(1)}}k\right)+\rho^{(2)}\sqrt{\nu^{(2)}}\left(\Omega^{(2)}+\sqrt{\nu^{(2)}}k\right)\right]\times[(\rho^{(2)}-\rho^{(1)})gk+\gamma k^3+(\rho^{(2)}+\rho^{(1)})\sigma^2(k)] \\ &\quad+4\left(\rho^{(1)}\sqrt{\nu^{(1)}}\Omega^{(1)}+\rho^{(2)}\nu^{(2)}k\right)\left(\rho^{(2)}\sqrt{\nu^{(2)}}\Omega^{(2)}+\rho^{(1)}\nu^{(1)}k\right)\sigma(k)k=0, \end{aligned} \tag{78}$$

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 numerically find the roots of the nonlinear equation (78). Once $\sigma(k)$ is determined, the initial values $\left\{u_k^0, w_k^0, P_k^0, h_k^0\right\}$ are determined as follows.

In the upper domain:

$$\begin{aligned} w_k^0 &= A\exp(-|k|z)+B\exp\left(-\frac{\Omega^{(1)}}{\sqrt{\nu^{(1)}}}z\right), \\ u_k^0 &= -\frac{i|k|}{k}A\exp(-|k|z)-\frac{i}{k}\frac{\Omega^{(1)}}{\sqrt{\nu^{(1)}}}B\exp\left(-\frac{\Omega^{(1)}}{\sqrt{\nu^{(1)}}}z\right), \\ P_k^0 &= \frac{\rho^{(1)}\sigma(k)}{|k|}A\exp(-|k|z). \end{aligned} \tag{79}$$

In the lower domain:

$$\begin{aligned} w_k^0 &= C\exp(|k|z)+D\exp\left(\frac{\Omega^{(2)}}{\sqrt{\nu^{(2)}}}z\right), \\ u_k^0 &= \frac{i|k|}{k}C\exp(|k|z)+\frac{i}{k}\frac{\Omega^{(2)}}{\sqrt{\nu^{(2)}}}D\exp\left(\frac{\Omega^{(2)}}{\sqrt{\nu^{(2)}}}z\right), \\ P_k^0 &= -\frac{\rho^{(2)}\sigma(k)}{|k|}C\exp(|k|z). \end{aligned} \tag{80}$$

For the interface:

$$h_k^0=\frac{a}{2}, \tag{81}$$

where a is a small real constant that specifies the initial wave amplitude. The coefficients $A,\ B,\ C,\ D$ are determined by a and $\sigma(k)$, and their values can be found in [42].

In this test, we pick $k=1$, $a=0.01$, and consider an air–water system with the physical parameters: $\rho^{(1)}=0.0012\ \text{g/cm}^3$, $\mu^{(1)}=1.8\times10^{-4}\ \text{g/(cm s)}$, $\rho^{(2)}=1.0\ \text{g/cm}^3$, $\mu^{(2)}=1.1\times10^{-2}\ \text{g/(cm s)}$, which correspond to a Reynolds number of 2846. The surface tension is set to zero in this test. The domain of computation is chosen to be a rectangle

$$\{(x,z)|0\leqslant x\leqslant 2\pi,\ -H\leqslant z\leqslant H\} \tag{82}$$

with $H=1$. The initial conditions and the boundary values at the two ends ($Z=\pm H$) are taken from the linear solution. We advance the solution until $\tau=0.4$, using (fixed) 32 points in the X-direction, $2J$ points in the Z-direction and N steps in time. Again we keep doubling N and J to check the error pattern, and the results are shown in Table 2 where the quantities E and R have the same meaning as defined before. We observe bigger numerical errors than those in the first example due to the fact that we are using an approximate linear solution as the reference. Another reason is that the viscosity is much smaller (or,

Table 2
Test results for the two-phase linear flow *without* surface tension. N and J are the numbers of time steps and spatial points, respectively. $E(\cdot,N,J)$ denotes the L_2-norm of the errors and $R(\cdot,N,J)$ indicates the order of convergence.

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)

the Reynolds number is much higher) in the current case, so that higher resolution is required to achieve good accuracy. Nevertheless, the results in Table 2 clearly indicate that second-order convergence for all the flow variables and the interface profile is approached as the resolution is refined.

We note that in the test presented above, the surface tension is ignored (i.e., $\gamma = 0$). A similar test can be performed which is based on the same set-up as above, but includes the physical surface tension (with $\gamma = 74\ \mathrm{g/s^2}$) to demonstrate the capability of our method. To calculate the mean curvature κ (see Eq. (9)), the surface derivatives h_x and h_{xx} are evaluated by the discrete Fourier transform. The test results are shown in Table 3. We have found that due to the addition of the surface tension, smaller Δt is needed to maintain good accuracy. Nonetheless, the results again confirm that fully second-order convergence is achieved.

3.3. Parallelization

One advantage of the numerical algorithm described in Section 2 is that it can be easily adapted to parallel computer architectures. The details are presented below.

Suppose the domain of computation is a rectangle and there are $2K$ points in the X-direction and $2J$ 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 processor performs the Fourier transform along the X-direction, calculates the right-hand side vectors R_k (see Eq. (55)), and prepares the data for the boundary value problem. Then we switch to column-wise striped partitioning, by way of an all-to-all communication, to solve the boundary value problem (66). 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 major overhead in this parallel algorithm is the switch between row-wise and column-wise partitioning.

A test of the performance of the parallelization by using MPI [21] 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 where generally good efficiency is observed. The drop in performance for 32 processors is associated with the fact $K = 32$, as communication costs are beginning to be important.

3.4. Viscous Stokes waves simulation

Now we apply the validated algorithm to simulate Stokes waves [13,33,34,36,38] in the presence of viscosity. In what follows we neglect the surface tension. We consider a two-fluid system in a frame moving with the phase speed c and use the expansion formula from [38] to obtain the initial condition. The wave profile h can be expanded in a dimensionless form by a Fourier cosine series

$$h = \sum_{k=1}^{\infty} A_k(A)\cos kx, \tag{83}$$

where A is the amplitude parameter and each coefficient A_k is depending on A. The first five Fourier coefficients are given as, to the order of $O(A^5)$,

Table 3
Test results for the two-phase linear flow *with* surface tension. N and J are the numbers of time steps and spatial points, respectively. $E(\cdot,N,J)$ denotes the L_2-norm of the errors and $R(\cdot,N,J)$ indicates the order of convergence.

N	J	$E(u,N,J)$ $(R(u,N,J))$	$E(w,N,J)$ $(R(w,N,J))$	$E(P,N,J)$ $(R(P,N,J))$	$E(h,N,J)$ $(R(h,N,J))$
160	80	3.082×10^{-3} (–)	1.710×10^{-3} (–)	5.943×10^{-2} (–)	1.037×10^{-4} (–)
320	160	7.556×10^{-4} (2.02)	4.164×10^{-4} (2.03)	1.459×10^{-2} (2.02)	2.528×10^{-5} (2.03)
640	320	1.410×10^{-4} (2.31)	8.108×10^{-5} (2.27)	2.780×10^{-3} (2.29)	4.759×10^{-6} (2.30)

Table 4
Performance of parallelization by using MPI. The speedup is defined as $S(n) = T(1)/T(n)$ and the efficiency is $E(n) = S(n)/n$.

No. of processors n	1	2	4	8	16	32
CPU time $T(n)$ (in s)	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

$$
\begin{aligned}
A_1 &= A,\\
A_2 &= \frac{1}{2}\frac{\rho^{(2)}-\rho^{(1)}}{\rho^{(2)}+\rho^{(1)}}\left(1+\frac{17(\rho^{(2)})^2-38\rho^{(2)}\rho^{(1)}+17(\rho^{(1)})^2}{12(\rho^{(2)}+\rho^{(1)})^2}A^2\right)A^2,\\
A_3 &= \frac{3(\rho^{(2)})^2-10\rho^{(2)}\rho^{(1)}+3(\rho^{(1)})^2}{8(\rho^{(2)}+\rho^{(1)})^2}A^3\\
&\quad+\frac{459(\rho^{(2)})^4-2468(\rho^{(2)})^3\rho^{(1)}+4130(\rho^{(2)})^2(\rho^{(1)})^2-2468\rho^{(2)}(\rho^{(1)})^3+459(\rho^{(1)})^4}{384(\rho^{(2)}+\rho^{(1)})^4}A^5,\\
A_4 &= \frac{(\rho^{(2)}-\rho^{(1)})((\rho^{(2)})^2-6\rho^{(2)}\rho^{(1)}+(\rho^{(1)})^2)}{3(\rho^{(2)}+\rho^{(1)})^3}A^4,\\
A_5 &= \frac{125(\rho^{(2)})^4-1516(\rho^{(2)})^3\rho^{(1)}+3118(\rho^{(2)})^2(\rho^{(1)})^2-1516\rho^{(2)}(\rho^{(1)})^3+125(\rho^{(1)})^4}{384(\rho^{(2)}+\rho^{(1)})^4}A^5,
\end{aligned}
\tag{84}
$$

and the phase speed c is

$$
c^2=\frac{\rho^{(2)}-\rho^{(1)}}{\rho^{(2)}+\rho^{(1)}}\left(1+\frac{(\rho^{(2)})^2+(\rho^{(1)})^2}{(\rho^{(2)}+\rho^{(1)})^2}A^2+\frac{(\rho^{(2)}-\rho^{(1)})^2(5(\rho^{(2)})^2-14\rho^{(2)}\rho^{(1)}+5(\rho^{(1)})^2)}{4(\rho^{(2)}+\rho^{(1)})^4}A^4\right). \tag{85}
$$

Tsuji and Nagata [38] were able to give the expression for the Fourier coefficients of the stream functions in both fluids up to the fifth-order. Consequently, the velocity and pressure can be calculated from the stream functions. These solutions are used as the initial values in our codes. Then we turn on the viscosity in both fluids and start the computation. Our main purpose is to have a deep understanding of the decay pattern for the wave amplitude due to viscous damping. To that end, we consider three choices for the amplitude parameter

(1) a small value $A = 0.01$;
(2) a moderate value $A = 0.1$;
(3) a relatively big value $A = 0.2$.

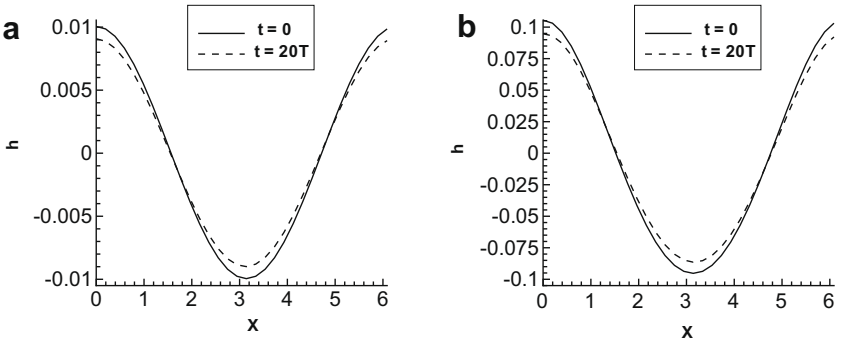

Fig. 1. The interface profiles from the numerical simulation of viscous Stokes waves at $t = 0$ and $t = 20T$, where T is one wave period, with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8\times10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1\times10^{-2}$ and two choices for the amplitude parameter A: (a)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

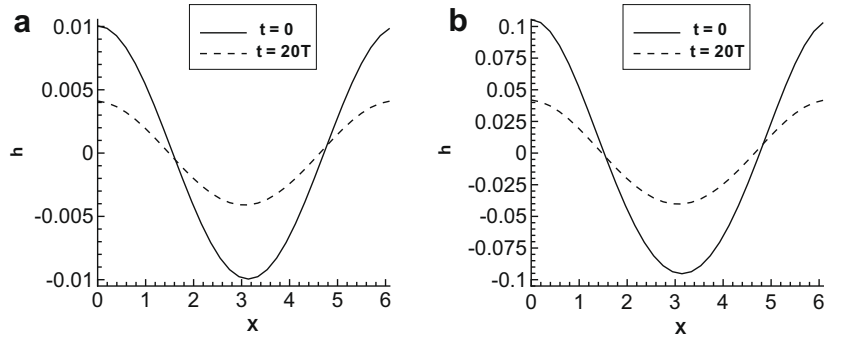

Fig. 2. The interface profiles from the numerical simulation of viscous Stokes waves at $t = 0$ and $t = 20T$, where T is one wave period, with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8\times10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1\times10^{-1}$ and two choices for the amplitude parameter A: (a) $A = 0.01$; (b) $A = 0.1$.

We also consider two choices for the viscosities:

(1) The typical air–water case with $\rho^{(1)} = 0.0012,\ \mu^{(1)} = 1.8 \times 10^{-4},\ \rho^{(2)} = 1.0,\ \mu^{(2)} = 1.1 \times 10^{-2}$.
(2) An artificial case where the densities are the same with, but the viscosities are 10 times bigger than, the air–water case. Specifically, $\rho^{(1)} = 0.0012,\ \mu^{(1)} = 1.8 \times 10^{-3},\ \rho^{(2)} = 1.0,\ \mu^{(2)} = 1.1 \times 10^{-1}$.

The spatial domain of computation is the same as defined in (82) but with H large enough so that it reasonably represents two layers of infinite thickness, which is the case considered in most previous work. We perform the computation from $\tau = 0$ until $\tau = 20T$, where T is one wave period. Figs. 1 and 2 show the wave profiles at $\tau = 0$ and $\tau = 20T$ for $A = 0.01, 0.1$ and the

Table 5
Decay rates in the air–water case for three choices of the amplitude parameter A. Results indicate that the decay rate for the mode A_k is approximately k times that for A_1. This pattern is distinct from the linear case.

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 6
Decay rates in the case with 10 times bigger viscosities, for three choices of the amplitude parameter A. Results again show that the decay rate for the mode A_k is approximately k times that for A_1, a pattern distinct from the linear prediction.

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}

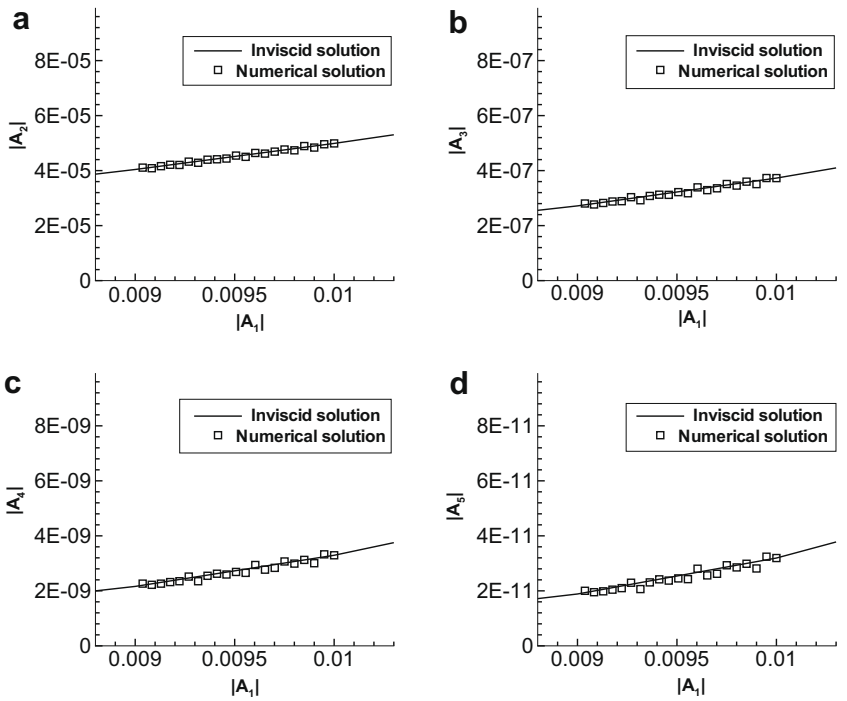

Fig. 3. Comparison between the inviscid solution and the numerical viscous solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4},\ \rho^{(2)} = 1.0,\ \mu^{(2)} = 1.1 \times 10^{-2}$ and the amplitude parameter $A = 0.01$. The numerical solution is displayed from $\tau = 0$ 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 $|A_1|$.

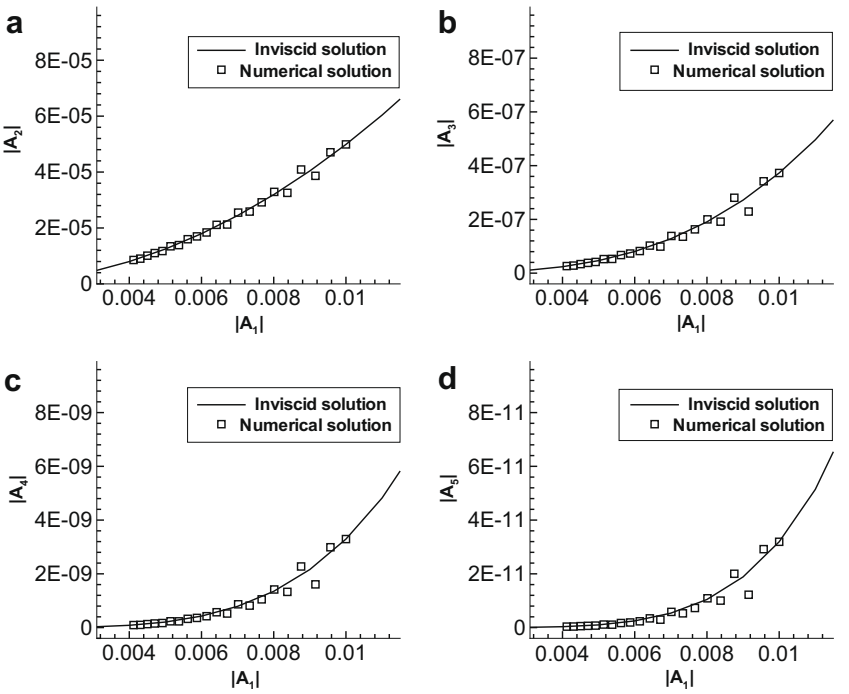

Fig. 4. Comparison between the inviscid solution and the numerical viscous solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$ and the amplitude parameter $A = 0.01$. The numerical solution is displayed from $\tau = 0$ 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 $|A_1|$.

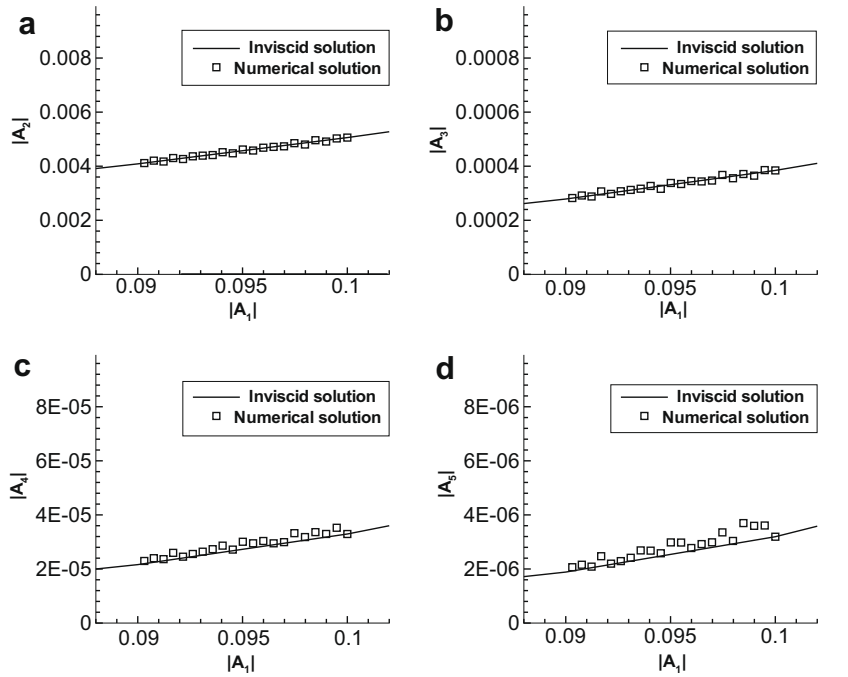

Fig. 5. Comparison between the inviscid solution and the numerical viscous solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$ and the amplitude parameter $A = 0.1$. The numerical solution is displayed from $\tau = 0$ 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 $|A_1|$.

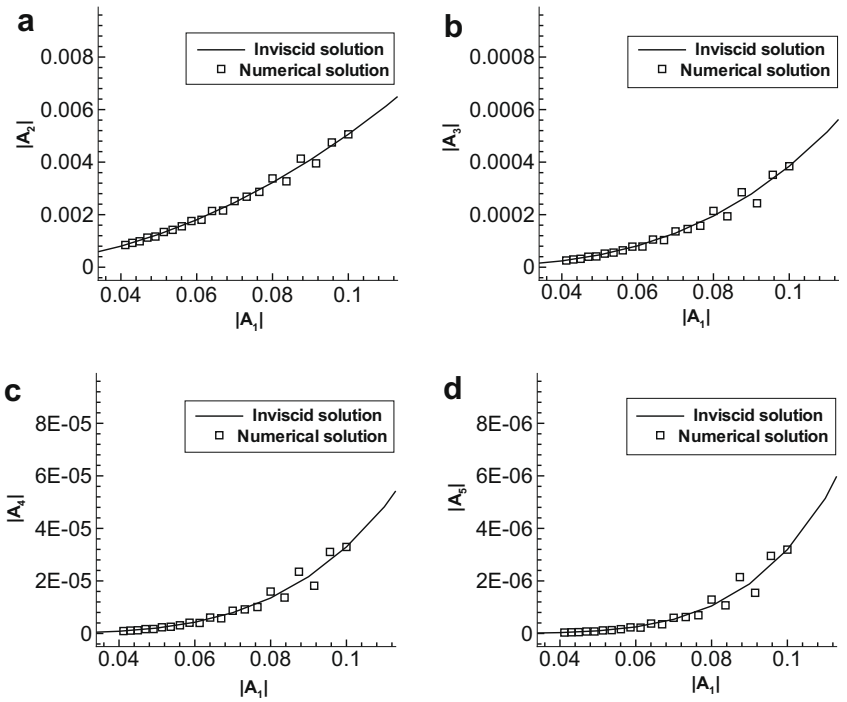

Fig. 6. Comparison between the inviscid solution and the numerical viscous solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$ and the amplitude parameter $A = 0.1$. The numerical solution is displayed from $\tau = 0$ 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 $|A_1|$.

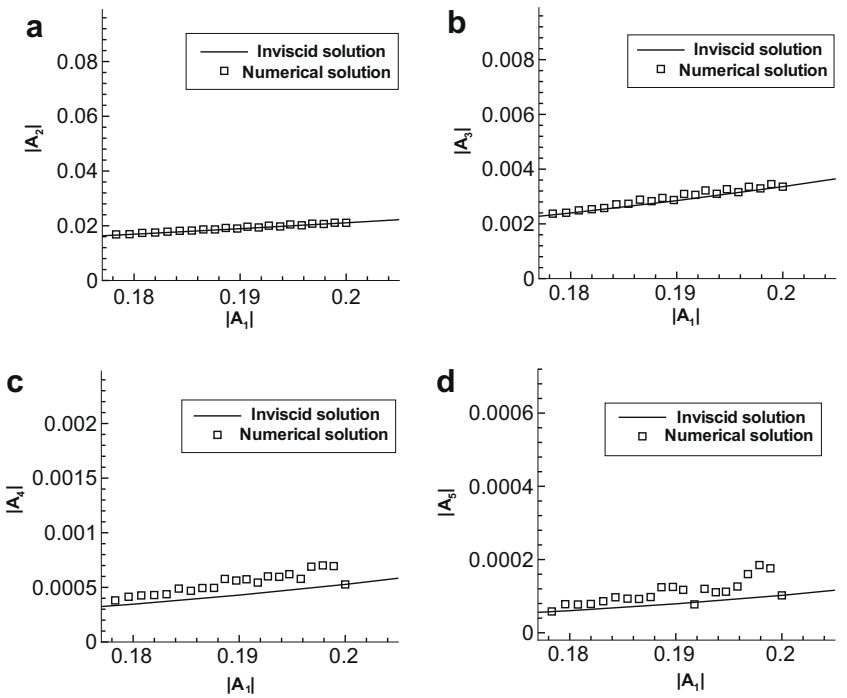

Fig. 7. Comparison between the inviscid solution and the numerical viscous solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$ and the amplitude parameter $A = 0.2$. The numerical solution is displayed from $\tau = 0$ 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 $|A_1|$.

two choices of the viscosities, respectively. One can clearly see the decay of the wave amplitude. In the case of bigger viscosities (Fig. 2), the wave decays faster than in the air–water case (Fig. 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 perform a quantitative study, we define the decay rate, $\hat{\sigma}(k)$, for each mode A_k through the relation

$$A_k(\tau) = A_k(0)e^{\hat{\sigma}(k)\tau}. \tag{86}$$

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

$$\hat{\sigma}(k) = \frac{\mathrm{Ln}(A_k(20T)) - \mathrm{Ln}(A_k(0))}{20T}, \tag{87}$$

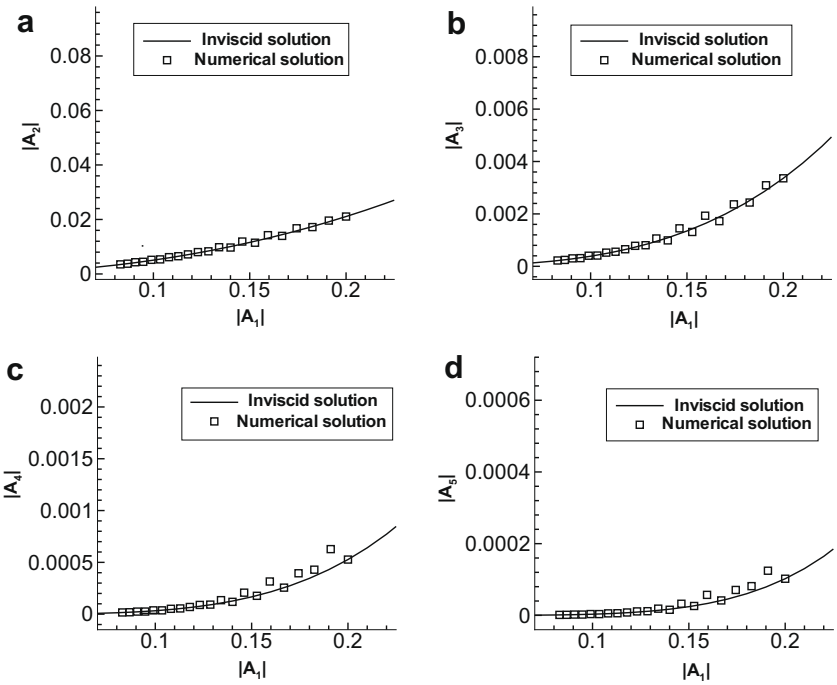

Fig. 8. Comparison between the inviscid solution and the numerical viscous solution of the Stokes wave with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-3}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-1}$ and the amplitude parameter $A = 0.2$. The numerical solution is displayed from $\tau = 0$ 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 $|A_1|$.

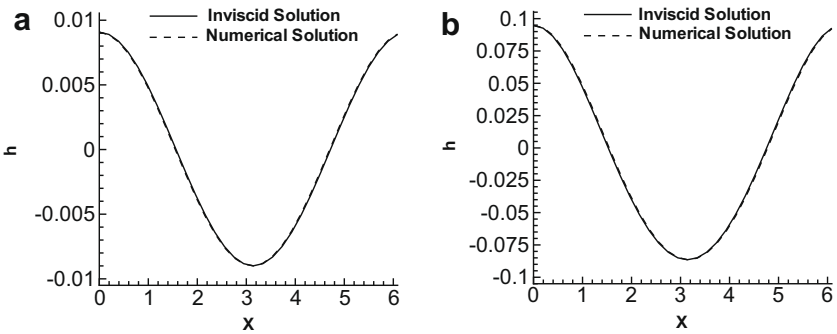

Fig. 9. Comparison between the inviscid solution and the numerical viscous solution for the profiles of Stokes waves with $\rho^{(1)} = 0.0012$, $\mu^{(1)} = 1.8 \times 10^{-4}$, $\rho^{(2)} = 1.0$, $\mu^{(2)} = 1.1 \times 10^{-2}$. (a) The numerical solution starts from $A = 0.01$ and is plotted at $t = 20T$, 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 $A \doteq 0.09031$.

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 Eq. (78). The comparison is made in dimensionless units and the results are shown in Tables 5 and 6, 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 is that the decay rate $\hat{\sigma}(k)$ for the kth mode is approximately $\hat{\sigma}(k) = k\hat{\sigma}(1)$, at least for $k = 1, 2, \ldots, 5$. 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 5, the decay rate of the fourth mode when $A = 0.2$. There are several possibilities for this discrepancy, but we will delay discussion of it until after we view the results from a different perspective.

Tables 5 and 6 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 (84) we know the analytic relationship between these modes. For the inviscid case, the

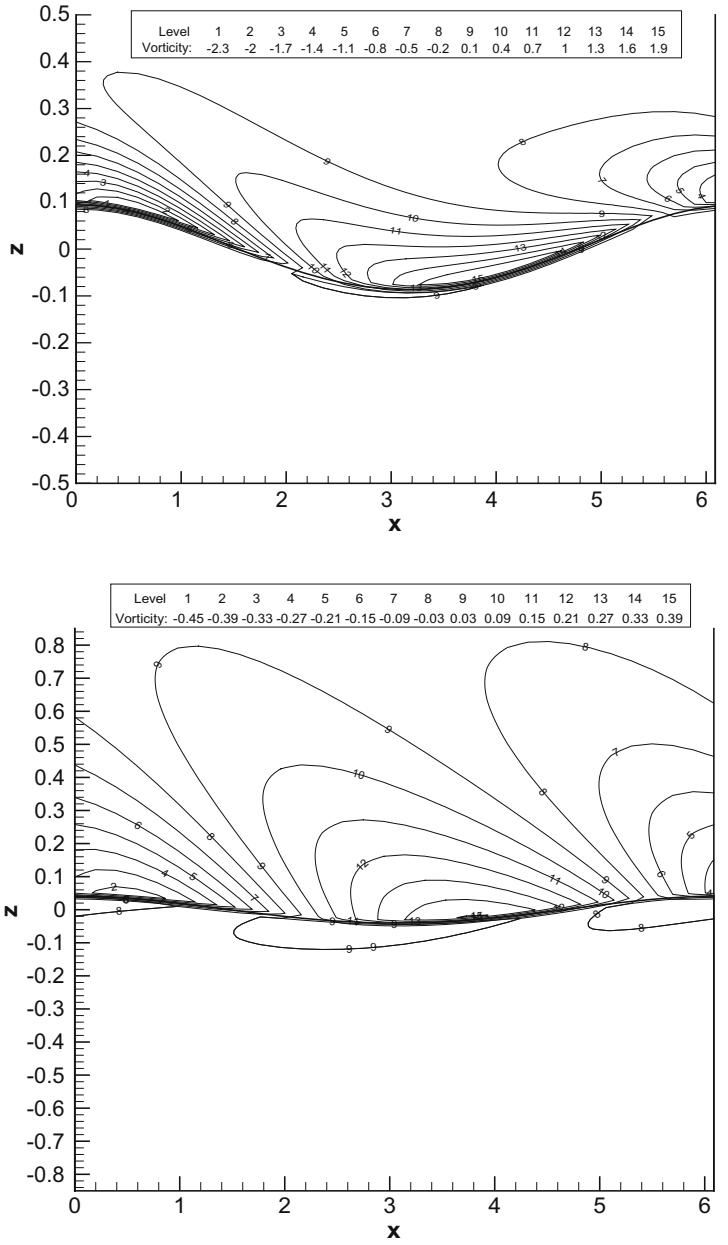

Fig. 10. The vorticity contours with the amplitude parameter $A = 0.1$ for the two choices of viscosities: (a) $\rho^{(1)} = 0.0012,\ \mu^{(1)} = 1.8 \times 10^{-4},\ \rho^{(2)} = 1.0,\ \mu^{(2)} = 1.1 \times 10^{-2}$; (b) $\rho^{(1)} = 0.0012,\ \mu^{(1)} = 1.8 \times 10^{-3},\ \rho^{(2)} = 1.0,\ \mu^{(2)}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

results in (84) suggest one way to view the family of Stokes waves is to consider the curves $|A_k|(|A_1|)$. 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 (84) 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 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 Figs. 3–8, 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 $\tau = 20T$. Figs. 3, 5 and 7 give the results in the air–water case for $A = 0.01, 0.1, 0.2$, respectively. Figs. 4, 6 and 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. These results agree with the asymptotic expansion for viscous Stokes waves proposed in [40].

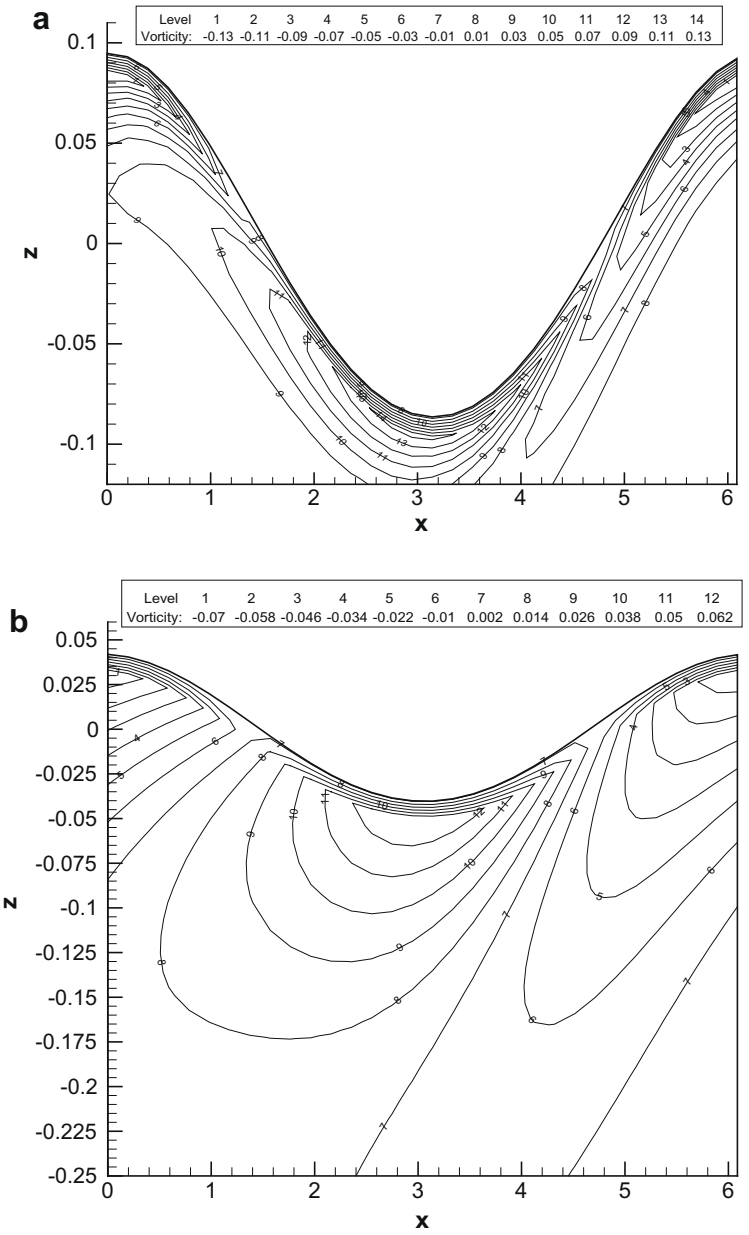

Fig. 11. The vorticity contours in the lower fluid with the amplitude parameter $A = 0.1$ for the two choices of viscosities: (a) $\rho^{(1)} = 0.0012,\ \mu^{(1)} = 1.8 \times 10^{-4},\quad \rho^{(2)} = 1.0,\ \mu^{(2)} = 1.1 \times 10^{-2}$; (b) $\rho^{(1)} = 0.0012,\ \mu^{(1)} = 1.8 \times 10^{-3},\ \rho^{(2)} = 1.0,\ \mu^{(2)} = 1.1 \times 10^{-1}$.

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 discontinuous 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 Fig. 8) where the modes decay much faster and the numerical solution and the inviscid solution show pretty good agreement when $\tau \geqslant 10T$.

One more evidence is provided in Fig. 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 (83) to obtain an inviscid solution. The numerical solutions and the inviscid solutions are plotted for both cases in Fig. 9 and we find excellent agreement between them.

We mention again that surface tension is not considered in our numerical study of Stokes waves. In the presence of capillary effects, waves will typically develop sharp corners [44,45] which clearly breaks the pattern of Stokes waves. However, such capillary effects are significant only for very short waves (with large wave numbers). For waves with small or moderate wave numbers, gravity and viscosity dominate and surface tension can be reasonably neglected [40,44].

Finally we plot the vorticity contours in Fig. 10 for $A = 0.1$ and with the two choices of viscosities. We see that the viscous boundary layers are well resolved. 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 domain for both cases and present the enlarged pictures in Fig. 11.

4. Conclusions

We have developed a numerical algorithm which ensures an accurate representation of the viscous interfacial motion, and which enables us to perform direct numerical simulation to Navier–Stokes equations with moving interfaces. This method allows us to treat viscosity jumps and large density ratios (about 1000 to 1 in the water–air case) without introducing unnecessary numerical smoothing. It is thus capable of capturing the very thin boundary layers at an evolving interface in slightly viscous fluids. The method achieves spectral accuracy in the horizontal direction, and fully second-order accuracy in the time marching and the vertical direction, for both the velocity and pressure. One important feature of this algorithm is that it can be easily adapted to parallel computing, making large-scale simulations possible.

There are several possible ways to make refinements for this algorithm. For example, more delicate mapping techniques [20] can be applied if it is necessary to consider situations where the interfaces fail to be single-valued, though this is not a concern in our current study focused on viscous effects. Higher-order time-marching methods [2] and boundary value problem solvers [3] can be employed to further improve the accuracy if needed. Although we have restricted our attention to two-dimensional flows in this paper, there is no inherent difficulty in adding to our numerical formulation another horizontal direction, say Y, especially if periodicity is assumed. In that case, the Fourier transform will be applied in both X and Y directions, which still leads to an ODE system in the form of (55) that can be similarly solved by identifying growing and decaying solutions.

With the algorithm presented in this paper, Stokes waves can be followed sufficiently in time for an accurate study of the viscous effects on wave motion. Our numerical results are consistent with the asymptotic expansion for viscous Stokes waves proposed in [40]. Our future plan is to apply this algorithm to numerically study the generation of water waves (such as sea surface waves) by wind forcing, and investigate the role of viscosity in the processes of wind–water interaction.

Acknowledgments

We thank the two anonymous referees for their valuable suggestions to improve the paper. This work was partially supported by NSF Grant No. DMS-0112759.

References

[1] D.A. Anderson, J.C. Tannehill, R.H. Pletcher, Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Corporation, 1984.
[2] U. Ascher, S. Ruuth, B. Wetton, Implicit–explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (1995) 797–823.
[3] U. Ascher, R. Matteij, R. Russell, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, Prentice Hall, New Jersey, 1988.
[4] G.R. Baker, K.M. Berger, J.T. Johnson, 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] G.R. Baker, D.I. Meiron, S.A. Orszag, Generalized vortex methods for free surface flow problems, J. Fluid Mech. 123 (1982) 477–501.
[6] G.K. Batchelor, An Introduction to Fluid Dynamics, Cambridge University Press, 1967.
[7] J.B. Bell, P. Colly, H.M. Gall, A second order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (1989) 257–283.
[8] J. Billingham, A.C. King, Wave Motion, Cambridge University Press, 2000.
[9] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.
[10] S. Chandrasekhar, Hydrodynamic and Hydromagnetic Stability, Clarendon Press, Oxford, 1961.
[11] S.C. De, Contribution to the theory of Stokes waves, Proc. Cambridge Philos. Soc. 51 (1955) 713–736.
[12] W. E, J.-G. Liu, Projection method I: convergence and numerical boundary layers, SIAM J. Numer. Anal. 32 (1995) 1017–1057.
[13] J.D. Fenton, A fifth-order Stokes theory for steady waves, J. Waterw. Port Coast. Ocean Eng. 111 (2) (1985) 216–234.
[14] J. Glimm, O. McBryan, R. Menikoff, D. Sharp, Front tracking applied to Rayleigh–Taylor instability, SIAM J. Sci. Stat. Comput. 7 (1986) 230–251.
[15] P.M. Gresho, Incompressible fluid dynamics: some fundamental formulation issues, Annu. Rev. Fluid Mech. 23 (1991) 413–453.
[16] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluid 8 (1965) 2182–2189.
[17] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[18] J.Y. Holyer, Large amplitude progressive interfacial waves, J. Fluid Mech. 93 (1979) 433–448.
[19] K. Ito, M.C. Lai, Z. Li, An augmented approach for Stokes equations with a discontinuous viscosity and singular forces, Comput. Fluid 36 (2007) 622–635.
[20] P. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, 1994.
[21] V. Kumar, A. Grama, A. Gupta, G. Karypis, Introduction to Parallel Computing: Design and Analysis of Algorithms, The Benjamin/Cummings Publishing Company, 1994.
[22] R.J. LeVeque, Z. Li, Immersed interface method for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (1997) 709–735.
[23] M.T. Levi Civita, Determination rigoureuse des ondes permanentes d'ampleur finie, Math. Ann. 93 (1925) 264–314.
[24] Z. Li, K. Ito, The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains, SIAM Front. Appl. Math. 33 (2006).
[25] H. Lomax, T.H. Pulliam, D.W. Zingg, Fundamentals of Computational Fluid Dynamics, Springer, 2001.
[26] M.S. Longuet-Higgins, E.D. Cokelet, The deformation of steep surface waves on water. I: A numerical method of computation, Proc. R. Soc. Lond. A 95 (1976) 1–26.
[27] W.F. Noh, P.R. Woodward, SLIC (simple line interface calculation), in: Proc. Fifth Int. Conf. Fluid Dyn., vol. 59, Lect. Notes Phys., Springer-Verlag, Berlin, 1976, pp. 330–340.
[28] S.A. Orszag, Numerical simulation of incompressible flows within simple boundaries. I: Galerkin (spectral) representations, Stud. Appl. Math. 50 (1971) 293–327.
[29] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[30] R. Peyret, Spectral Methods for Incompressible flow, Springer, 2002.
[31] R.D. Richtmyer, K.W. Morton, Difference Method for Initial Value Problems, John Wiley and Sons, 1967.
[32] R. Scardovelli, S. Zaleski, Direct numerical simulation of free surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[33] L.W. Schwartz, Computer extension and analytic continuation of Stokes' expansion for gravity waves, J. Fluid Mech. 62 (1974) 553–578.
[34] L.W. Schwartz, J.D. Fenton, Strongly nonlinear waves, Annu. Rev. Fluid Mech. 14 (1982) 39–60.
[35] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 2000.
[36] G.G. Stokes, On the theory of oscillatory waves, Trans. Cambridge Philos. Soc. 8 (1847) 441–455.
[37] N. Takada, M. Misawa, A. Tomiyama, A phase-field method for interface-tracking simulation of two-phase flows, Math. Comput. Simul. 72 (2006) 220–226.
[38] Y. Tsuji, Y. Nagata, Stokes' expansion of internal deep water waves to the fifth order, J. Ocean. Soc. Jpn. 29 (1973) 61–69.
[39] T. Vinje, P. Brevig, Numerical simulation of breaking waves, Adv. Water Resour. 4 (1981) 77–82.
[40] J. Wang, An asymptotic expansion for Stokes waves with viscosity, Fluid Dynam. Res. 40 (2008) 155–161.
[41] J. Wang, Computation of 2D Navier–Stokes equations with moving interfaces by using GMRES, Int. J. Numer. Method Fluid 54 (2007) 333–352.
[42] J. Wang, On the accuracy of a numerical method for linear interfacial motion, Appl. Math. Comput. 188 (2007) 1733–1741.
[43] J.E. Welch, F.H. Harlow, J.P. Shannon, B.J. Daly, The MAC method, Los Alamos Scientific Laboratory Report LA-3425, Los Alamos, NM, 1966.
[44] G.B. Whitham, Linear and Nonlinear Waves, John Wiley and Sons, 1974.
[45] Y. Yang, G. Tryggvason, Dissipation of energy by finite-amplitude surface waves, Comput. Fluid 27 (1998) 829–845.