Journal of Computational Physics 228 (2009) 7313–7332

Contents lists available at ScienceDirect

Journal of Computational Physics

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

On the generation and maintenance of waves and turbulence in simulations of free-surface turbulence

Xin Guo, Lian Shen *

Department of Civil Engineering, Johns Hopkins University, Baltimore, MD 21218, USA

ARTICLE INFO

Article history:
Received 15 December 2008
Received in revised form 22 June 2009
Accepted 23 June 2009
Available online 1 July 2009

Keywords:
Free-surface flow
Wave
Turbulence

ABSTRACT

One of the challenges in numerical simulation of wave–turbulence interaction is the precise setup and maintenance of wave and turbulence fields. In this paper, we investigate techniques for the generation and suppression of specific surface wave modes, the generation of turbulence in an inhomogeneous physical domain with a wavy boundary-fitted grid, and the generation and maintenance of waves and turbulence during the complex wave–turbulence interaction process. We apply surface pressure to generate and suppress waves. Based on the solution of linearized Cauchy–Poisson problem, we derive three pressure expressions, which lead to a δ-function method, a time-segment method, and a gradual method. Numerical experiments show that these methods generate waves as specified and eliminate spurious waves effectively. The nonlinear wave effect is accounted for with a time-relaxation method. For turbulence generation, we extend the linear forcing method to an inhomogeneous physical domain with a curvilinear computational grid. Effects of force distribution and computational grid distortion are examined. For wave–turbulence interaction, we develop an algorithm to instantaneously identify specific progressive and standing waves. To precisely control the wave amplitude in a complex turbulent flow field, we further develop an energy controlling method. Finally, a simulation example of wave–turbulence interaction is presented. Results show that turbulence has unique features in the presence of waves. Velocity fluctuations are found to be strongly dependent on the wave phase; variations of these fluctuations are explained by the pressure–strain correlation associated with the wave-induced strain field.

1. Introduction

Interaction between water surface waves and turbulence is a challenging problem. As a wave passes a turbulence field, the turbulence scatters and dissipates the wave. At the same time, orbital motion and surface drift of the wave distort the turbulence. Because of the significance of the problem in basic scientific research and because of its importance in engineering and geophysical applications, there is a critical need for a fundamental understanding of wave–turbulence interaction.

This paper addresses the computational issues of creation and maintenance of wave and turbulence in numerical simulations of free-surface turbulence. For a mechanistic study, it is essential to set up the wave and turbulence fields precisely. For example, the turbulence-to-wave time ratio needs to be specified for the effect of rapid and slow distortions to be investigated. The wave field should not be deteriorated by spurious wave modes such as standing waves, which are often caused by the use of periodic boundary condition in the simulation. Moreover, for turbulence statistics to converge, the flow should

* Corresponding author.
E-mail address: LianShen@jhu.edu (L. Shen).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.06.030

be in equilibrium statistically in the simulation study; that is, proper energy input should be provided to balance dissipation, without violating the dynamics of wave and turbulence.

To resolve the above issues, in this paper we address three key challenges in simulation of free-surface flows. The first is how to generate waves as specified and how to suppress unwanted waves. Wave simulations reported in the literature often began with a prescribed wave solution as the initial condition. However, because of the complexity of nonlinear free-surface boundary conditions, an exact solution usually cannot be obtained beforehand. The simulations had to construct the initial fields based on simple wave theories. As a result, the discrepancy between the true wave dynamics and the approximations of the initial condition often resulted in spurious standing waves. The standing wave can be caused by the pressure associated with the vortical portion of the flow, or the wave nonlinearity. Dommermuth developed novel numerical procedures to suppress spurious standing waves by reducing the initial impulse of pressure [6] and by using a relaxation method that accounts for wave nonlinearity gradually [7].

In this study, we extend Dommermuth's approach to the general case. Based on the solution of the Cauchy–Poisson problem for pressure-generated waves, we develop three methods of applying surface pressure forcing to generate waves precisely as specified and to suppress unwanted waves efficiently. For the complexities associated with wave nonlinearity as well as viscosity, we use the time-relaxation method so that spurious disturbance is minimized.

Our second challenge is about the generation of statistically steady isotropic turbulence under the wave. The isotropic turbulence helps the investigation of the fundamental mechanism of wave–turbulence interaction without the complication of shear flow and decaying turbulence effects. In previous simulations of free-surface turbulence without wind effects, three methods have often been used to generate turbulence: (i) production at a bottom boundary layer [15,17,10,1]; (ii) shear instability in the bulk flow [5,26,25]; and (iii) decaying turbulence using another turbulent flow as initial input [23,31]. In the first method, the advantage of turbulence being produced at a nonslip bottom is that it can directly correspond to open channel flows, and that the flow can be quasi-steady. The disadvantage is that the channel bottom is sometimes undesirable, e.g. in the study of a flow with a deep bottom. To remove the flow anisotropy generated at the bottom of the open channel, the simulation needs a sufficiently large Reynolds number [20], which is beyond the computational capability of direct numerical simulation (DNS). With the second method, the computational bottom can be free-slip, minimizing the bottom effect. The disadvantage is that the mean shear flow is unsteady. As a result, many ensemble runs are needed for turbulence statistics to converge. The third method starts with an initially isotropic turbulent flow, suddenly inserts a free surface, and then allows the turbulence to decay. Turbulence decays fast in this case, which makes flow statistics difficult to converge, often necessitating a large number of ensemble runs.

In this study, we seek a method that can generate isotropic turbulence continuously in a wave field. For flows that are isotropic and homogeneous in all three directions, one can generate turbulence by injecting energy at low wavenumbers [3,27] or by applying a stochastic external force [9]; both are carried out in the spectral space of computation. This method is, however, not suitable for wave–turbulence interaction, which has to be simulated in an inhomogeneous physical space that is nonperiodic in the vertical direction. Recently, Lundgren [19] developed a linear forcing method, in which a force proportional to the velocity fluctuation is applied directly in the physical space. In the current paper, we extend this method to the complex setting of a wave field, in which the flow is inhomogeneous and the computational grid is curved. The effect of nonuniform forcing and computational grid distortion on the properties of the turbulence generated is studied.

Our third challenge is about the generation and suppression of waves in a broadband wave system with turbulence, and the generation and maintenance of turbulence in the multi-scale wave environment. It is essential to identify different wave components in order to apply different treatments to them. Since the turbulence intermittently excites the surface to generate waves, the wave treatment needs to be continuous in time. It is also critical that in the process of generating and maintaining turbulence, the wave field is not distorted.

In this study, we first develop a method to identify progressive and standing waves for each specific modes based on instantaneous surface elevation and velocity. We then apply surface pressure forcing to generate and suppress different wave modes. To fine-tune and maintain the target wave mode the study focuses on, we further develop an energy controlling method to keep its amplitude quasi-stationary in the simulation. Finally, we present an example of wave–turbulence interaction simulation, in which important flow features are identified and explained.

This paper is organized as follows: Section 2 introduces two numerical codes used in this paper. Section 3 discusses wave generation and maintenance, followed by Section 4 that discusses turbulence generation and maintenance. Section 5 addresses the implementation of the techniques for wave and turbulence setup, together with some special treatments for the complex environment of wave–turbulence interaction. Finally, Section 6 presents conclusions.

2. Numerical schemes

The methods for wave and turbulence generation developed in this study are not limited to any specific simulation code. In this paper, to show numerical results, we use two simulation tools as examples. The first is a potential flow based high-order spectral (HOS) method, which is highly efficient and accurate, and we use it to show wave response to surface pressure. The second is a direct numerical simulation (DNS) of Navier–Stokes equations with viscous nonlinear free-surface boundary conditions, which is used to show results of turbulent flows.

2.1. High-order spectral method

When the wave is described by potential flow theory, the governing equation for the velocity potential Φ is

$$\nabla^2\Phi = \frac{\partial^2\Phi}{\partial^2 x} + \frac{\partial^2\Phi}{\partial^2 y} + \frac{\partial^2\Phi}{\partial^2 z} = 0. \tag{1}$$

Here x and y are horizontal coordinates and z-axis points upward with $z = 0$ the mean water level. In HOS, evolution of η and surface potential $\Phi^s \equiv \Phi|_{z=\eta}$ [35] are simulated. Using Φ^s, we write fully nonlinear free-surface kinematic and dynamic boundary conditions as

$$\eta_t + \nabla_s\eta\cdot\nabla_s\Phi^s - (1+\nabla_s\eta\cdot\nabla_s\eta)\Phi_z(x,y,\eta,t) = 0, \tag{2}$$

$$\Phi^s_t + \frac{\eta}{Fr^2} + \frac{1}{2}\nabla_s\Phi^s\cdot\nabla_s\Phi^s - \frac{1}{2}(1+\nabla_s\eta\cdot\nabla_s\eta)\Phi^2_z(x,y,\eta,t) = -P_a, \tag{3}$$

where $\nabla_s = \partial/\partial x + \partial/\partial y$, η is the surface elevation, P_a is atmospheric pressure at the surface. Here and hereafter, all of the variables are normalized by characteristic velocity scale U and length scale L. The Froude number is $Fr = U/(gL)^{1/2}$, with g the gravity acceleration. Pressure is normalized by ρU^2, with ρ the water density. In Eq. (3), a viscous term $D_\Phi = 2c_g\varepsilon\Phi^s$ [33] can be added to the left-hand side to represent dissipation, with c_g the group velocity of the carrier wave and ε the dissipation rate.

To solve Eqs. (2) and (3), we express the surface potential as a perturbation series of the velocity potential and perform Taylor series expansion about the mean water level:

$$\Phi^s(\vec{x},t) = \sum_{m=1}^{M}\sum_{l=0}^{M-m}\frac{\eta^l}{l!}\left.\frac{\partial^l\Phi^{(m)}}{\partial z^l}\right|_{z=0}. \tag{4}$$

Here $()^m$ denotes a quantity of $\mathcal{O}(\epsilon^m)$, with $\epsilon = ka$ a small value that represents the wave steepness. Here k is wavenumber and a is wave amplitude.

By applying the pseudo-spectral method using Fourier series and mode-coupling treatment, we integrate Eqs. (2) and (3) in time. In this work, a fourth-order Runge–Kutta scheme is used. Details of HOS formulation and its implementation are provided in [8] and Chapter 15 of [21].

2.2. Direct numerical simulation with nonlinear free-surface boundary conditions

When the wave and turbulent flow are described as viscous Newtonian fluid, the governing equations are the Navier–Stokes equations

$$\frac{\partial\vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u} = -\nabla p + \frac{1}{Re}\nabla^2\vec{u}, \tag{5}$$

and the continuity equation

$$\nabla\cdot\vec{u} = 0. \tag{6}$$

Here p is the dynamic pressure normalized by ρU^2; $Re = UL/\nu$ is the Reynolds number, with ν the kinematic viscosity.

At the surface, the kinematic free-surface boundary condition is

$$\frac{\partial\eta}{\partial t} + u\frac{\partial\eta}{\partial x} + v\frac{\partial\eta}{\partial y} - w = 0, \quad \text{at } z = \eta. \tag{7}$$

The dynamic boundary conditions are

$$\begin{cases} \vec{t}_1\cdot[\sigma]\cdot\vec{n}^T = 0, \\ \vec{t}_2\cdot[\sigma]\cdot\vec{n}^T = 0, \\ \vec{n}\cdot[\sigma]\cdot\vec{n} = -P_a. \end{cases} \tag{8}$$

Here the stress tensor $[\sigma]$ and the direction vectors $\vec{n}, \vec{t}_1$, and $\vec{t}_2$ are, respectively:

$$[\sigma] = \begin{bmatrix} -P + \frac{2}{Re}\frac{\partial u}{\partial x} & \frac{1}{Re}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & \frac{1}{Re}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \frac{1}{Re}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & -P + \frac{2}{Re}\frac{\partial v}{\partial y} & \frac{1}{Re}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \frac{1}{Re}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) & \frac{1}{Re}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) & -P + \frac{2}{Re}\frac{\partial w}{\partial z} \end{bmatrix}, \tag{9}$$

$$\begin{cases} \vec{n} = \frac{(-\eta_x,-\eta_y,1)}{\sqrt{\eta_x^2+\eta_y^2+1}}, \\ \vec{t}_1 = \frac{(1,0,\eta_x)}{\sqrt{\eta_x^2+1}}, \\ \vec{t}_2 = \frac{(0,1,\eta_y)}{\sqrt{\eta_y^2+1}}, \end{cases} \tag{10}$$

with $P = p - z/Fr^2$. In Eq. (8), $\vec{n}^T$ denotes the transpose of $\vec{n}$. To solve the governing equations (5) and (6) with free-surface boundary conditions (7) and (8), we use a boundary-fitted grid system. We apply an algebraic mapping to transform the irregular Cartesian space (x,y,z,t) confined by the wave surface to a rectangular computational domain $(\xi,\psi,\varsigma,\tau)$ as shown in Fig. 1. The transformation is

$$\tau = t, \quad \xi = x, \quad \psi = y, \quad \varsigma = \frac{z+\overline{H}}{\eta+\overline{H}}, \tag{11}$$

where $\overline{H}$ is the distance from the mean surface level to the computational domain bottom. Based on the chain rule, we express Eqs. (5)–(10) in term of $(\xi,\psi,\varsigma,\tau)$. For space discretization, in the horizontal directions, we use a pseudo-spectral method with Fourier series; in the vertical direction, we use a second-order finite-difference scheme on a staggered grid [11], which is clustered towards the top boundary. A Crank–Nicholson scheme is used for the viscous terms. An Adams–Bashforth scheme is used for the convection terms. The momentum equations are advanced in time by a fractional-step method with the pressure solved by a Poisson equation to satisfy the continuity equation [14]. For the evolution of the free surface, Eq. (7) is advanced in time with a second-order Runge–Kutta scheme. Numerical details of DNS and its validation are provided in [34].

3. Wave generation and suppression

In this study, we generate and suppress waves by applying pressure at the free surface. We discuss linear wave case first, for which we obtain the relationship between the pressure and surface response by solving the linearized Cauchy–Poisson problem. The details of the solution are given in [21]. Here we outline the solution for two-dimensional deep water wave, whereas the extension to three dimension and finite water depth is straightforward.

Given initial condition $\eta(x,0)$ and $\Phi(x,z,0)$, one can use Fourier–Laplace transform to obtain [21]:

$$\begin{aligned} \eta(x,t) = {} & \frac{1}{2\pi}\int_{-\infty}^{+\infty} dk \exp[ikx]\tilde{\eta}(k,0)\cos[\omega t] + \frac{1}{2\pi}\int_{-\infty}^{+\infty} dk \exp[ikx]\widetilde{\Phi}(k,0,0)Fr^2(\omega\sin[\omega t] - \delta(t)) \\ & - \frac{1}{2\pi}\int_{-\infty}^{+\infty} dk \exp[ikx]\int_0^t \widetilde{P}_a(k,\tau)Fr^2\omega\sin[\omega(t-\tau)]d\tau, \end{aligned} \tag{12}$$

where the tilde denotes the Fourier mode, and the wave frequency is $\omega = \sqrt{k/Fr^2}$.

Eq. (12) shows that the evolution of surface elevation is controlled by initial surface elevation, initial velocity potential, and history of surface pressure applied. On the right-hand side of the equation, the first term shows that an initial surface elevation leads to a standing wave that has the same amplitude. The second term shows that an initial velocity potential generates a standing wave, of which the amplitude is $\widetilde{\Phi}(k,0,0)\omega Fr^2$. The third term shows that when a pressure is applied on a surface, the surface responds to the pressure input. Therefore, the surface can be controlled by different pressure inputs. We can solve the inverse problem to find the pressure that leads to a specific surface elevation.

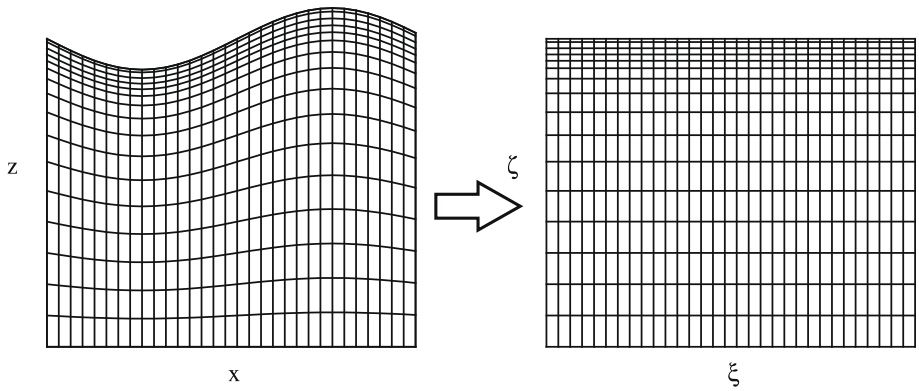

Fig. 1. Algebraic mapping to transform the irregular Cartesian space (x,y,z,t) confined by the wave surface to a rectangular computational domain $(\xi,\psi,\varsigma,\tau)$.

3.1. Standing wave

Spurious standing waves are often encountered in simulations. To suppress them, one can generate additional standing waves that have the same magnitude but with opposite phase. Therefore, we first study how to generate a standing wave as specified. Based on the Cauchy–Poisson problem solution, we develop three methods as discussed below.

3.1.1. δ-Function method

If we want a standing wave to be

$$\eta = a(t)\cos[kx] = a_0 \sin[\omega t]\cos[kx], \tag{13}$$

from Eq. (12), we can solve the pressure as

$$P_a = -\frac{a_0}{Fr^2\omega}\delta(t)\cos[kx]. \tag{14}$$

Here $\delta(t)$ is Dirac's delta function. In our simulation, a mollified $\delta(t)$-function is used:

$$\delta(t) = \begin{cases} \frac{1}{2\Delta}\left(1+\cos\left[\frac{\pi}{\Delta}t\right]\right) & -\Delta < t < \Delta, \\ 0 & \text{otherwise}, \end{cases} \tag{15}$$

where Δ is a parameter that determines the width of numerical smoothing in time. The lower limit of Δ depends on the timestep dt in the simulation, for the delta function approximation to be adequately represented. The upper limit of Δ depends on the wave period T, for the pressure (14) to generate the wave (13) sufficiently fast and accurately. Our tests show that satisfying results can be obtained if $4dt \leqslant \Delta \leqslant 10\%T$, which is not difficult to satisfy in wave–turbulence interaction simulation because the timestep is usually small for fine turbulence structures to be resolved. Therefore, the δ-function method needs only a short duration in the simulation for it to take effect.

Fig. 2(a) shows the standing wave generated by this δ-function method using the HOS and DNS codes. Results show reasonable agreement among the numerical results and Eq. (13).

3.1.2. Time-segment method

If we want to generate a standing wave within a time period as

$$\eta = a(t)\cos[kx] = \begin{cases} \frac{1}{2}a_0 \sin\left[\omega t - \frac{\pi}{n}\right](1-\cos[n\omega t])\cos[kx] & t \leqslant \frac{\pi}{n\omega}, \\ a_0 \sin\left[\omega t - \frac{\pi}{n}\right]\cos[kx] & t > \frac{\pi}{n\omega}, \end{cases} \tag{16}$$

from Eq. (12), we can solve the pressure as

$$P_a = \begin{cases} -\frac{n}{4}\frac{a_0}{Fr^2}\cos[kx] & \left[(n+2)\sin\left[(n+1)\omega t - \frac{\pi}{n}\right]\right. \\ & \left. -(n-2)\sin\left[(n-1)\omega t + \frac{\pi}{n}\right]\right] \quad t \leqslant \frac{\pi}{n\omega}, \\ 0 & t > \frac{\pi}{n\omega}. \end{cases} \tag{17}$$

In the above equations, n is a parameter.

Fig. 2(b) shows the evolution of the amplitude of the standing wave generated by this time-segment method using HOS and DNS codes. The numerical results agree with the theoretical prediction (16). The standing wave is generated completely when the pressure is taken off at $t = \pi/(n\omega)$. At $t = \pi/(n\omega)$, the fluid velocity reaches its maximum value, the surface elevation is zero, and the pressure acting on the surface becomes zero. Therefore, the pressure is taken off smoothly. The transition does not generate an additional standing wave.

3.1.3. Gradual method

If we want to generate a standing wave gradually according to

$$\eta = a_0(1-\exp[-\beta t])\sin[\omega t]\cos[kx], \tag{18}$$

from Eq. (12), we can solve the pressure as

$$P_a = a_0\frac{\beta}{\omega^2 Fr^2}\exp[-\beta t](-2\omega\cos[\omega t] + \beta\sin[\omega t])\cos[kx]. \tag{19}$$

In the above, β is a parameter controlling the speed of wave generation.

By using this method, we generate the standing wave gradually. The pressure acting on the surface has a profile of a standing wave. It has a phase difference from the standing wave we want. The amplitude of the pressure decreases gradually. The decay rate is the same as the increase rate of the standing wave. As shown in Fig. 2(c), the standing wave is generated faster when β has a larger value. If we consider the generation process of the standing wave to be complete when 99% of the energy has been input, we can calculate the time needed for this process as $5.3/\beta$, which is independent of the wave period.

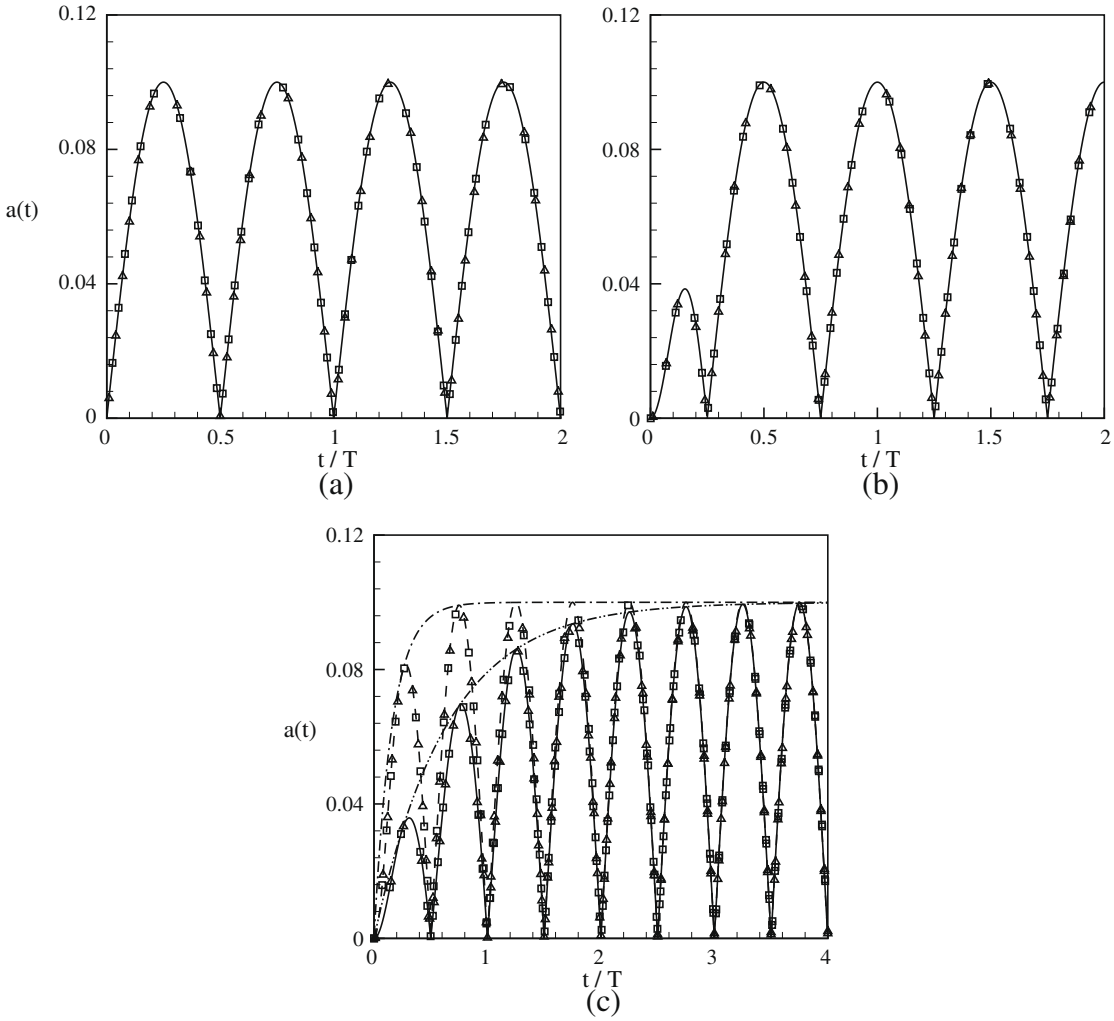

Fig. 2. History of the amplitude of the standing wave generated by different methods: (a) δ-function method; (b) time-segment method ($n = 2$); and (c) gradual method ($\beta = 0.25$ and $\beta = 1$). △ denotes DNS results and □ denotes HOS results. In both (a) and (b), — denotes the theoretical prediction. In (c), — and – – – denote the theoretical prediction of the standing wave amplitude for $\beta = 0.25$ and $\beta = 1$, respectively; – ·· – ·· – and – · – · – denote the envelope of wave amplitude for $\beta = 0.25$ and $\beta = 1$, respectively. In all the figures, $a_0 = 0.1$, $k = 1$, and $Fr^2 = 1$.

From the numerical results in Sections 3.1.1–3.1.3, we conclude that DNS results agree with the HOS results for wave simulation. Hereafter, due to space limitation, for pure-wave cases, we present only HOS results to save space.

3.1.4. Dommermuth's method

In numerical simulation of free-surface viscous flows, improper initialization generates high-frequency standing waves due to the impulse of the pressure associated with vortical motions [6]. The standing wave can be shown from an example given by [6] as follows. When the surface elevation and fluid velocity are initially zero, a pressure $P = P_0 \cos[kx]$ is applied on the surface, with P_0 a constant. From Eq. (12), a surface elevation is generated as

$$\eta = -Fr^2 P_0(\cos[\omega t] - 1)\cos[kx], \tag{20}$$

which consists of a mean hydrostatic component and a standing wave component.

Dommermuth [6] introduced a special treatment by smoothing the surface pressure transition with $P_a = P(1 - \exp[-t^n/\tau^n])$, where n is a parameter and τ is the adjustment period. The amplitude of the standing wave is found to reduce by two orders of magnitude.

We next compare Dommermuth's method to the three methods introduced in Sections 3.1.1–3.1.3. To suppress the standing wave in Eq. (20), we generate a standing wave with the same magnitude but opposite phase using the δ-function method, time-segment method, and gradual method. Fig. 3(a) shows that all of the methods can reduce the amplitude of the standing wave to a small value. The reason that the three new methods do not eliminate the standing wave completely is

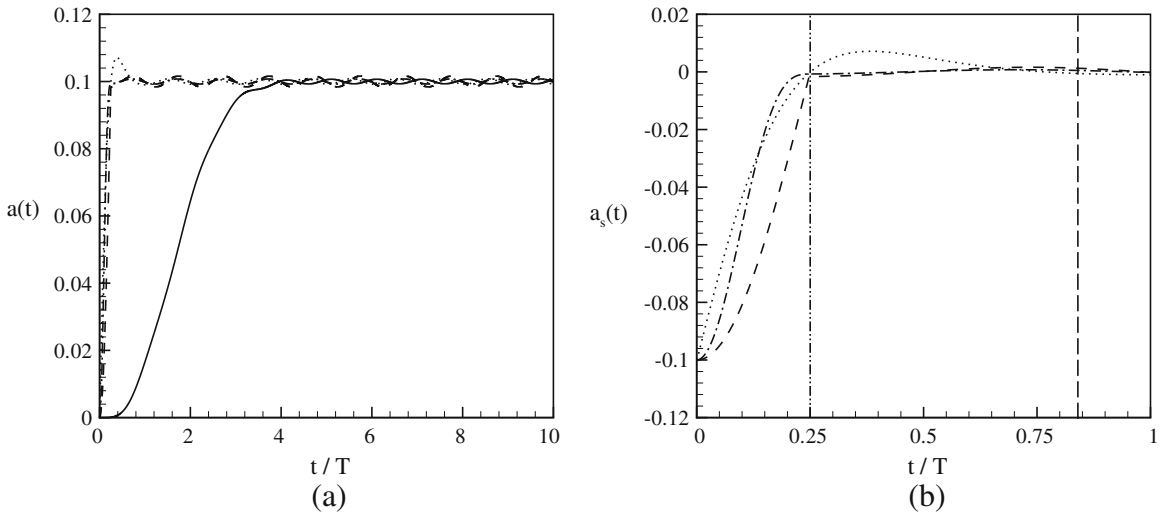

Fig. 3. (a) History of the amplitude of the surface deformation where the unwanted standing wave is suppressed by different methods. —, Dommermuth's method ($n = 2$ and $\tau = 2T$). (b) Detailed history of the amplitude of the unwanted standing wave. – ·· – ·· – denotes $t = \pi/(n\omega)$ with $n = 2$, and — — denotes $t = 5.3/\beta$ with $\beta = 1$. In both (a) and (b), – – –, δ-function method; – · – · –, time-segment method ($n = 2$); · · · · · ·, gradual method ($\beta = 1$).$P_0 = 0.1, k = 1$, and $Fr^2 = 1$; the results are obtained with HOS.

because they are derived based on initially calm surface, while in the present problem the surface is not calm due to the action of P. As a result, the surface motion caused by P leads to extra energy input by the pressure. Fig. 3(a) shows that the residual standing waves in the three new methods have about the same magnitude as that of Dommermuth's method. But the latter needs a longer time (small τ leads to large standing wave amplitude and Dommermuth [6] pointed out τ should be greater than wave period). Fig. 3(b) shows the detailed history of the amplitude of the standing wave (shown in Eq. (20)) suppressed by our methods. The δ-function method suppresses the standing wave rapidly once it is applied at $t = T/4$, whereas the time-segment method and the gradual method, which are applied at $t = 0$, need $\pi/(n\omega)$ and $5.3/\beta$, respectively.

3.2. Progressive wave

In this section, we discuss how to generate a progressive wave as specified. The approach is similar to the standing wave case, but with additional issues addressed below.

3.2.1. δ-Function method

A progressive wave can be decomposed into two standing waves:

$$\eta = a_0 \sin[\omega t + kx] = a_0 \sin[\omega t]\cos[kx] + a_0 \cos[\omega t]\sin[kx].$$

The two standing waves can be generated by the methods introduced in Section 3.1. Therefore, we construct a progressive wave by using the δ-function method introduced in Section 3.1.1 twice, with the pressure given as

$$P_a = -\frac{a_0}{Fr^2\omega}\delta(t)\cos[kx] + \frac{a_0}{Fr^2\omega}\delta\left(t - \frac{\pi}{2\omega}\right)\sin[kx]. \tag{21}$$

Fig. 4(a) shows the numerical result of this method. We note that at $t = T/4$, the velocity associated with the first standing wave is zero. Because there is no vertical motion of the surface at that instance, the addition of the pressure to generate the second standing wave does not input extra energy. This is an advantage of the δ-function method. The other two methods introduced in Section 3.1, the time-segment method and the gradual method, both need a period to input energy. As a result, the pressure to generate one standing wave inputs extra energy to the other standing wave, which produces spurious waves besides the progressive wave we want. Therefore, we use only the δ-function method to generate the two standing waves (i.e. the two impulses in Eq. (21)).

We remark that from Eq. (12), one can also generate the progressive wave with a pressure as

$$P_a = -\frac{a_0}{Fr^2\omega}\delta(t)\cos[kx] - \frac{a_0}{Fr^2\omega^2}\frac{d\delta(t)}{dt}\sin[kx]. \tag{22}$$

In this case, the two standing waves are generated at the same time. However, this method requires the implementation of a mollified $d\delta(t)/dt$ function, which we found difficult. Therefore, we use Eq. (21) only.

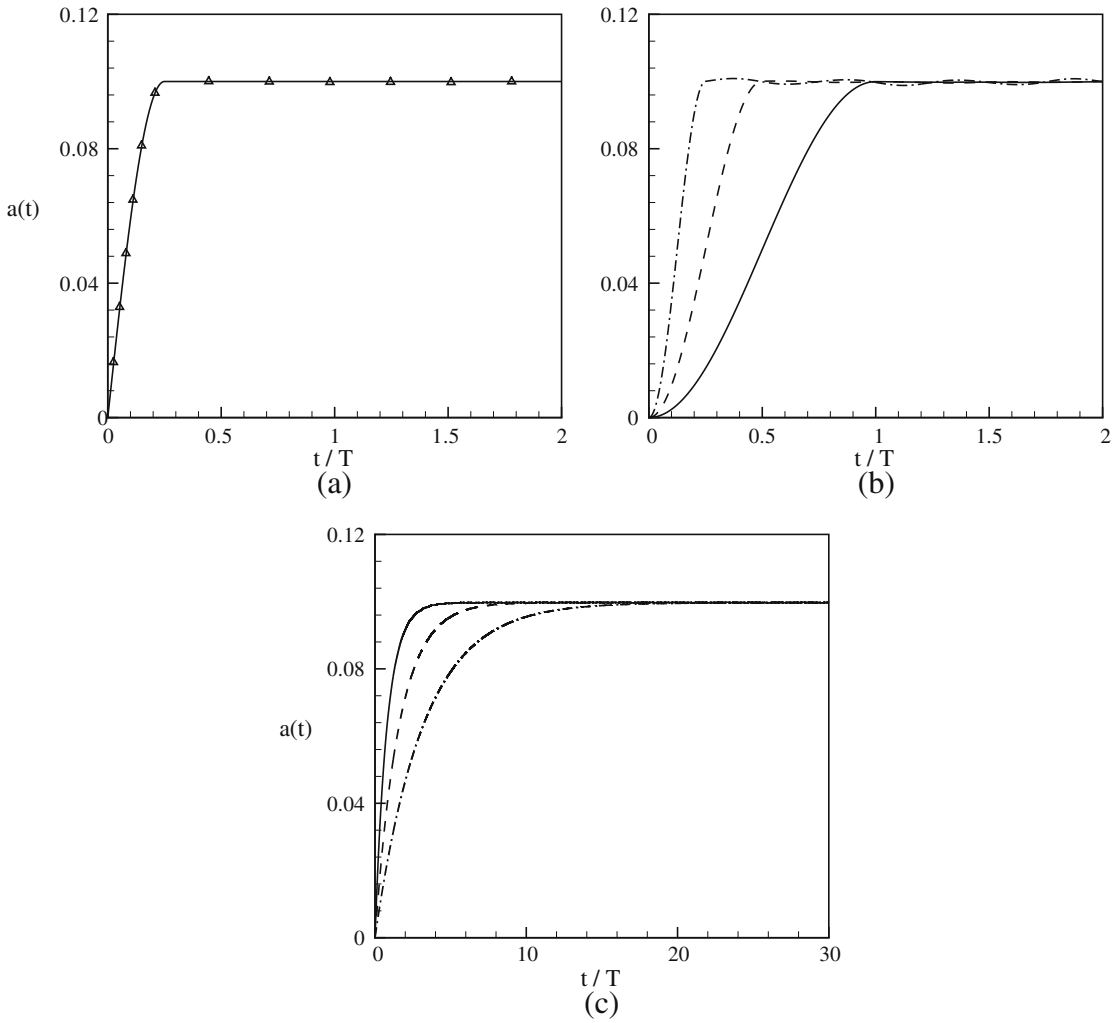

Fig. 4. (a) History of the amplitude of the standing wave generated by δ-function method. — denotes the theoretical prediction and $\triangle$ denotes HOS results. (b) History of the amplitude of the standing wave generated by time-segment method from HOS with values of n in Eq. (25) of: —, $n = 0.5$; – – –, n = 1; – · – · –, $n = 2$. (c) History of the amplitude of the standing wave generated by gradual method from HOS with β values in Eq. (26) of: —, $\beta = 0.2$; – – –, $\beta = 0.1$; – · – · –, $\beta = 0.05$.$a_0 = 0.1, k = 1$, and $Fr^2 = 1$.

3.2.2. Time-segment method

Besides decomposing the progressive wave into two standing waves and generating them separately, one can also generate a progressive wave directly. Similar to Section 3.1.2, we can apply a pressure according to

$$P_a = \begin{cases} -\frac{n}{4}\frac{a_0}{Fr^2} & [(n+2)\sin[kx+(n+1)\omega t] \\ & -(n-2)\sin[(n-1)\omega t - kx]] \quad t \leqslant \frac{\pi}{n\omega}, \\ 0 & t > \frac{\pi}{n\omega}. \end{cases} \tag{23}$$

The pressure during $t \in (0, \pi/n\omega)$ generates a progressive wave as

$$\eta = \frac{1}{2}a_0 \sin[kx+\omega t](1-\cos[n\omega t]). \tag{24}$$

At $t = \pi/(n\omega)$, the above equation becomes $\eta = a_0 \sin[kx+\omega t]$, which is the progressive wave we want. However, different from the standing wave case discussed in Section 3.1.2, the pressure given by Eq. (23) is nonzero at $t = \pi/(n\omega)$. As a result, the sudden taking off of the pressure at $t = \pi/(n\omega)$ generates a spurious standing wave. Nevertheless, we find that this standing wave has much smaller magnitude than the dominant progressive wave, especially when n is small. A numerical example is shown in Fig. 4(b).

We have also derived a pressure solution that gives exact transition of the surface elevation as

$$\eta = \begin{cases} \frac{1}{2}a_0 \sin[kx+\omega t](1-\cos[n\omega t]) & t \leqslant \frac{\pi}{n\omega}, \\ a_0 \sin[kx+\omega t] & t > \frac{\pi}{n\omega}. \end{cases} \tag{25}$$

However, we found the pressure form very complicated, which we do not recommend for numerical implementation.

3.2.3. Gradual method

If we want to gradually generate a progressive wave as

$$\eta = a_0(1-\exp[-\beta t])\sin[kx+\omega t], \tag{26}$$

from Eq. (12), we can solve the pressure as

$$P_a = \frac{a_0\beta}{\omega^2 Fr^2}[\exp[-\beta t](\beta \sin[kx+\omega t] - 2\omega\cos[kx+\omega t]) + \delta(t)\sin[kx]]. \tag{27}$$

The numerical example plotted in Fig. 4(c) shows that the progressive wave can be generated smoothly with this gradual method. Similar to the standing wave case discussed in Section 3.1.3, it needs a period of $5.3/\beta$ for 99% of the energy to be input.

Finally, we remark that the three methods introduced above are based on the linear wave solution of Eq. (12). For nonlinear waves, there exist higher wavenumber modes. While in principle one can adjust each wave mode if their amplitude and phase can be obtained beforehand, this approach is difficult to implement in practice. In next section, we discuss a more robust method, which introduces the wave nonlinearity gradually to the simulation with a time-relaxation method.

3.3. Nonlinear effect of waves

As illustrated by [7], in nonlinear simulation of a progressive wave, spurious standing waves of high frequency may be generated if a linear wave solution is used directly as the initial condition. In this study, we first apply the three linear wave-based methods of Section 3.2 to generate the wave from a calm water to investigate their suitability for nonlinear effects. We then implement a time-relaxation method to improve the performance.

Fig. 5(a) shows the history of the modal amplitude of the first four harmonics of the progressive wave. With the δ-function method and the time-segment method, oscillations exist for all of the four modes (for the first mode, the oscillation is not obvious in the plot due to the log-scale used, which is consistent with [7]), indicating the presence of standing waves. In the case of the δ-function method, the magnitude of the standing waves is significant, because the pressure impulse is equivalent to an initial linear velocity potential. The imbalance between the dynamics of nonlinear wave and the initial condition of the linear wave results in spurious standing waves, in a way similar to that in [7]. When the time-segment method is used, the wave generation process becomes smoother. Consequently, the spurious standing waves become smaller. In the case of the gradual method, the flow field has sufficient time for transition, and the spurious standing waves are negligible as shown in Fig. 5(a).

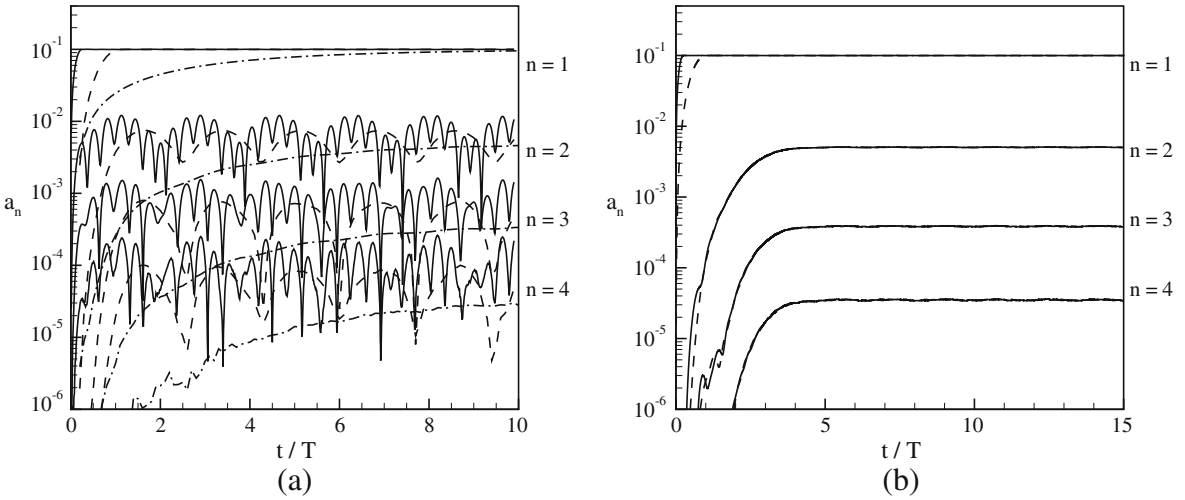

Fig. 5. Amplitude of the first four modes of the progressive waves generated by different methods: (a) δ-function method, time-segment method, and gradual method without nonlinear relaxation treatment; (b) δ-function method and time-segment method with nonlinear time-relaxation treatment. In both (a) and (b), —, δ-function method; – – –, time-segment method; – · – · –, gradual method. $a_0 = 0.1, k = 1$, and $Fr^2 = 1$; the results are obtained with HOS.

Results shown above indicate that nonlinear effect may produce erroneous results if the δ-function method and the time-segment method are used in wave generation. To reduce this error, we employ a time relaxation strategy introduced by Dommermuth [7]. In this approach, the nonlinearity is accounted for gradually in the simulation. With the HOS formulation, the free-surface boundary conditions, Eqs. (2) and (3), are modified as

$$\begin{cases} \eta_t - w^{(1)} = \left[-w^{(1)} - \nabla_s \Phi^s \cdot \nabla_s \eta + (1 + \nabla_s \eta \cdot \nabla_s \eta) w^s\right] \cdot \left(1 - \exp\left[-\left(\frac{t}{\tau}\right)^n\right]\right), \\ \Phi^s_t + \frac{\eta}{Fr^2} = -P_a + \left[-\frac{1}{2}\nabla_s \Phi^s \cdot \nabla_s \Phi^s + \frac{1}{2}(1 + \nabla_s \eta \cdot \nabla_s \eta)(w^s)^2\right] \cdot \left(1 - \exp\left[-\left(\frac{t}{\tau}\right)^n\right]\right). \end{cases} \tag{28}$$

Here $^{(1)}$ denotes the linear wave solution component.

In the case of DNS, the kinematic boundary condition (7) and the dynamic boundary condition associated with normal stress (8) are rewritten as

$$\begin{cases} w^{(1)} - \eta_t = (w^{(1)} - w + \eta_x u + \eta_y v)\left(1 - \exp\left[-\left(\frac{t}{\tau}\right)^n\right]\right), \\ \frac{\eta}{Fr^2} + P_a - p = -\frac{2}{Re} f \cdot \left(1 - \exp\left[-\left(\frac{t}{\tau}\right)^n\right]\right), \end{cases} \tag{29}$$

where

$$f = \frac{\eta_x^2 u_x + \eta_x \eta_y (v_x + u_y) + \eta_y^2 v_y - \eta_x (u_z + w_x) - \eta_y (v_z + w_y) + w_z}{1 + \eta_x^2 + \eta_y^2}.$$

As shown in [7], when the nonlinear wave dynamics are included in the simulation gradually, the imbalance with an initial condition based on linear wave theory can be reduced. In this study, we extend this adjustment method to the generation of finite-amplitude waves with a pressure variation obtained from linear wave solution (12). The nonlinear free-surface boundary conditions are originally used directly in the simulation. Here we revise the surface conditions, as shown in Eqs. (28) and (29), for them to transit smoothly from the linearized free-surface boundary conditions to the nonlinear ones. The dynamics of wave nonlinearity is fully accounted for in the end of the transition. Our numerical test shows that this method produces satisfying results. The standing waves observed in Fig. 5(a) are significantly suppressed in Fig. 5(b) after the relaxation method is implemented. Finally, we remark that in the DNS, the relaxation method can also smooth the initial imbalance between the solution of Eq. (12) and the viscous effect at the free surface, which is another source of initial spurious standing wave in the simulation.

3.4. Wave amplitude maintenance

In previous sections, we discuss how to generate and suppress waves in numerical simulation. After a wave is set up, an important issue in the simulation is to maintain its amplitude so that converged statistics of wave–turbulence interaction can be obtained in a quasi-steady setting. In this section, we discuss the use of surface pressure to balance viscous dissipation in wave simulation. More complex case associated with interaction of turbulence with multiple wave components is discussed in Section 5.

We consider the following viscous linear wave solution [16]:

$$\begin{cases} \eta(x,t) = a_0 \exp\left[-\frac{2k^2}{Re} t\right] \sin[kx + \omega t], \\ \Phi(x,z,t) = \frac{a\omega}{k} \exp\left[kz - \frac{2k^2}{Re} t\right] \cos[kx + \omega t], \\ U(x,z,t) = \frac{2k\beta}{Re} a \exp\left[\beta z - \frac{2k^2}{Re} t\right] (\sin[kx + \beta z + \omega t] - \cos[kx + \beta z + \omega t]), \\ W(x,z,t) = \frac{2k^2}{Re} a \exp\left[\beta z - \frac{2k^2}{Re} t\right] \sin[kx + \beta z + \omega t]. \end{cases} \tag{30}$$

Here $\beta = \sqrt{Re \cdot \omega/2}$ and $\omega = \sqrt{k/Fr^2}$. Potential components of the velocity are represented through $\Phi(x,z,t)$, and viscous components are denoted by $U(x,z,t)$ and $W(x,z,t)$. The viscous effects are accounted for directly in DNS. In HOS, an equivalent dissipation [33] is included as discussed in Section 2.1.

As shown in Eq. (30), the wave decays due to viscous dissipation. To maintain the wave, we supply the same amount of energy with a surface pressure. The pressure has amplitude of P_0 and a phase difference of $\pi/2$ with respect to the wave. As a result, the energy input rate is

$$-\int_{S_0} \eta_t P_a dS = P_0 a_0 \omega \pi. \tag{31}$$

The rate of viscous dissipation in the wave is

$$\varepsilon = -\frac{1}{Re} \int_V \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \frac{\partial u_i}{\partial x_j} dV. \tag{32}$$

To obtain the dissipation rate, we can either calculate Eq. (32) numerically or solve it analytically by using Eq. (30). With the latter approach (which is the same as performing Taylor series expansion of the wave amplitude in time), together with Eq. (31), we obtain the pressure magnitude as

$$P_0 = a_0 \frac{4k^2}{ReFr^2\omega}. \tag{33}$$

Fig. 6 shows numerical results. For the freely decaying case, both DNS and HOS yield decay rates that agree with the analytical solution (30). When the surface pressure is applied, the wave is well maintained. Because the pressure with the progressive wave profile is applied suddenly at $t = 0$, a standing wave is generated. This standing wave usually does not cause concerns in application, because its amplitude is small for the high Re used in wave–turbulence simulations. In the present case, the amplitude of the standing wave is two orders of magnitude smaller than that of the progressive wave. We also find that if Re decreases, although the initial magnitude of the standing wave increases, the smaller Re value leads to larger viscous dissipation of the standing wave for it to attenuate fast.

4. Turbulence generation

In this section, we generate turbulence by using the linear forcing method introduced by Lundgren [19]. The application in the wave environment requires the turbulence to be generated in an inhomogeneous domain with a curvilinear computational grid. We extend the linear forcing method by investigating effects of spacial variation of the forcing and distortion of computational meshes.

4.1. Inhomogeneous random forcing

Lundgren [19] introduced a scheme called linear forcing method to generate turbulence in physical space. In this method, a force proportional to the velocity fluctuation $\vec{u}'$ is applied to the Navier–Stokes equations:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u} = -\nabla p + \frac{1}{Re}\nabla^2\vec{u} + b\vec{u}', \tag{34}$$

where b is called body force parameter. The forcing, however, needs to vanish near the free surface to avoid generation of spurious interfacial phenomena. Therefore, we set b as: $b = b_0 f[z_c]$. Here z_c called central distance is the vertical distance to the center of the computational domain. The term $f[z_c]$ called body force shape function has unit value in the bulk region. It is damped gradually over a damping region, and becomes zero outside, where we call free region. In this study, we set $f[z_c]$ as

$$f[z_c] = \begin{cases} 1 & z_c \leqslant l_b, \quad \text{bulk region}, \\ \frac{1}{2}\left(1 - \cos\left[\frac{\pi}{l_d}(z_c - l_b - l_d)\right]\right) & z_c \leqslant l_b + l_d, \quad \text{damping region}, \\ 0 & z_c > l_b + l_d, \quad \text{free region}, \end{cases} \tag{35}$$

where l_b is half of the vertical length of the bulk region and l_d is the length of the damping region.

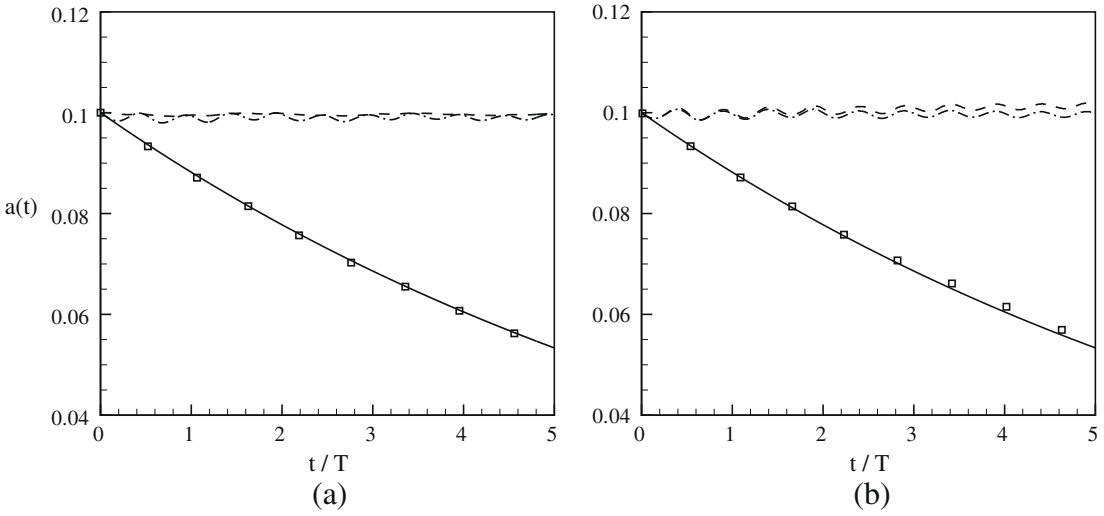

Fig. 6. History of the amplitude of freely decaying and maintained progressive waves. (a) HOS result. (b) DNS result. In both (a) and (b), —, theoretical prediction for freely decaying wave (30); □, freely decaying wave by simulation; – – –, wave maintained with the dissipation rate ε calculated analytically (33); – · – · –, wave maintained with ε calculated numerically. $a_0 = 0.1, k = 1, Fr^2 = 1$, and $Re = 100$.

Since the turbulence is generated inhomogeneously, there is a need to study the flow properties as a function of z_c. For the bulk flow, based on the results of Rosales and Meneveau [24], the integral length scale l is about 19% of the computational domain size L_d. We estimate the root-mean-square of velocity fluctuation u'_{rms}, dissipation rate ε, and Taylor scale λ [29] as

$$\begin{cases} u'_{rms} = 0.57bL_d, \\ \varepsilon = 0.97b^3L_d^2, \\ \lambda = \sqrt{\frac{15u'^2_{rms}}{Re\varepsilon}} = 2.24\sqrt{\frac{1}{bRe}}. \end{cases} \tag{36}$$

With different values of the controlling parameter b, flow specifics of the computation cases considered in this paper are given in Table 1.

In this study, we set $l_b = 1.5\pi$ to ensure that the bulk region is deep enough to serve as a reservoir of isotropic turbulence. To investigate effect of damping region length, we consider a wide damping region with $l_d = \pi/2$ and a narrow damping region with $l_d = \pi/10$.

Fig. 7 shows profiles of λ and u'_{rms} obtained from our DNS for Case 1 listed in Table 1. The results shown in this section are normalized by the values at the center of the bulk region (denoted by the subscript "c"). Fig. 7 shows that λ in the bulk region is almost a constant and is affected little by l_d. In the damping and free regions, λ increases, with the increase rates about the same between the two l_d cases. For u'_{rms}, it is constant at the center of the bulk region. In the case of $l_d = \pi/2$, it starts to decay at a distance of $\pi/2$ from the boundary of the bulk region. As a result, there is a $(2\pi)^3$ cubic in the bulk region with isotropic turbulence. In the case of $l_d = \pi/10$, u'_{rms} starts to decay at a distance of π from the boundary of the bulk region. Therefore, in the bulk region, the velocity fluctuation is affected by the damping region length. However, in the damping and free regions, u'_{rms} decays at about the same rate for the two l_d cases.

We next study variations of turbulence properties in the free region, where the free surface is located. This region is of critical importance to the dynamics of wave–turbulence interaction. In some sense, the approach of generating turbulence in the bulk region and letting the turbulence transport to the free region resembles the use of oscillating grid for turbulence generation in experiments. Thompson and Turner [30] and Hopfinger and Toly [13] found that $l \propto z_v$ and $u'_{rms} \propto z_v^{-1}$. Here z_v is defined as the distance from a "virtual origin", which is typically obtained by extrapolation of l for it to be zero. One can further obtain that $\lambda \propto z_v$. In this paper, we focus on λ instead of l, because the former is easier to obtain a smooth value numerically.

We fit the normalized λ and u'_{rms} as

$$\begin{cases} \frac{\lambda}{\lambda_c} = B_\lambda z_v, \\ \frac{u'_{rms}}{u'_{rms,c}} = \frac{B_u}{z_v}. \end{cases} \tag{37}$$

Table 2 shows the curve fitting results. It is shown that the slopes of the curve fitting, B_λ and B_u, do not change much for different l_d. However, the insensitivity in the coefficients does not necessarily indicate the suitability of the curve fitting itself. Examination of Fig. 7 shows that the approximation of oscillating grid turbulence works well for the Taylor length scale, but works poorly for the velocity fluctuation. On the other hand, we find that a better fit can be obtained with the exponent function:

$$\begin{cases} \frac{\lambda}{\lambda_c} = \exp[C_\lambda + D_\lambda z_c], \\ \frac{u'^2_{rms}}{u'^2_{rms,c}} = \exp[C_u + D_u z_c]. \end{cases} \tag{38}$$

The values of $C_\lambda, D_\lambda, C_u$, and D_u are listed in Table 2.

We next examine the anisotropic property of the turbulence. Following [18], we calculate the anisotropy tensor a_{ij} as

$$a_{ij} = \frac{\overline{u'_i u'_j} - \frac{1}{3}\delta_{ij}\overline{u'_k u'_k}}{\overline{u'_k u'_k}}, \tag{39}$$

where the overline ($^-$) denotes the horizontal average value. The invariants of the anisotropy tensor a_{ij} are

$$I_1 = a_{kk}, \quad I_2 = -\frac{1}{2}a_{ij}a_{ji}, \quad I_3 = \det[a_{ij}]. \tag{40}$$

Table 1
Parameters of the turbulence cases considered in this paper.

Case	b_0	Re	L_d	u'_{rms}	l	λ	$Re_\lambda = u'_{rms}\lambda/\nu$
1	0.1	1000	2π	0.358	1.19	0.224	80
2	0.25	20	2π	0.895	1.19	1.00	18

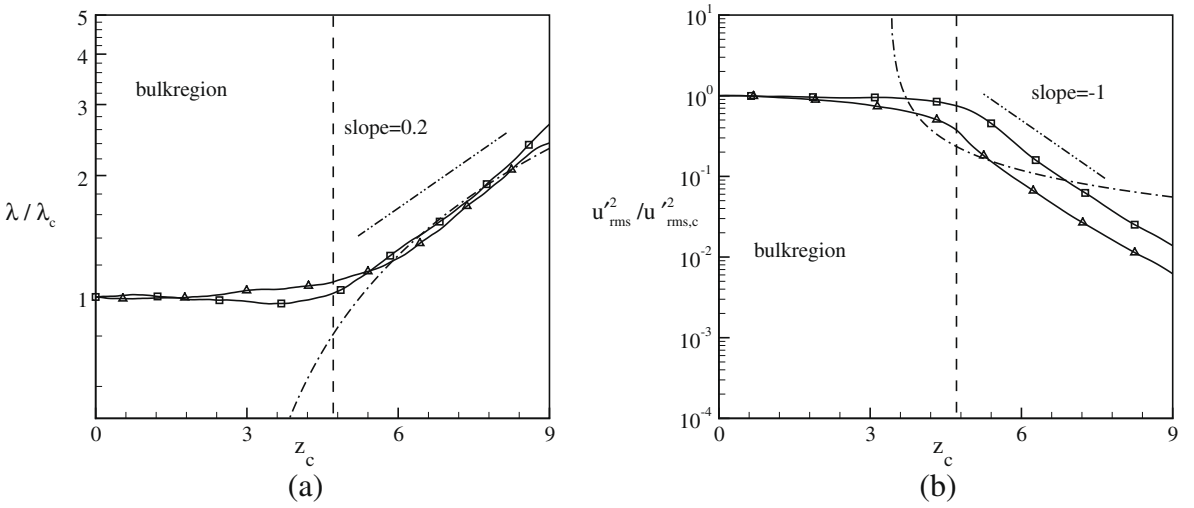

Fig. 7. Profiles of (a) λ/λ_c and (b) $u'^2_{rms}/u'^2_{rms,c}$. Numerical results: $\triangle, l_d = \pi/10; \square, _d = \pi/2$. Curve fitting: – · – · –, grid turbulence form from Eq. (37); – ·· – ·· –, exponent function from Eq. (38). The vertical dashed line denotes the boundary of the bulk region.

Table 2
Parameters of curve fitting in grid turbulence form from Eq. (37) and exponent function from Eq. (38).

l_d	B_λ	B_u	C_λ	D_λ	C_u	D_u
$\pi/2$	0.4245	0.3114	−1.0124	0.2085	4.3016	−0.9681
$\pi/10$	0.4030	0.3572	−0.9631	0.2050	3.2196	−0.9468

In isotropic turbulence, $I_2 = I_3 = 0$. For our test cases, we found that in the bulk, damping, and free regions, the maximum values of $|I_2|$ are, respectively, $3.3\times10^{-3}, 2.7\times10^{-3}$, and 1.4×10^{-2}; while those of $|I_3|$ are $2.5\times10^{-5}, 6.8\times10^{-6}$, and 7.8×10^{-5}, respectively. Therefore, despite the inhomogeneity in the problem, the turbulence remains isotropic at all the locations. This is not surprising because the body force is applied in all the directions in the same way.

4.2. Effect of computational grid distortion

In simulations involving surface waves, a curvilinear grid fitted to the wave boundary is often used. For example, our DNS shown in Section 2.2 maps the irregular wave-following physical space to a rectangular computation domain (Fig. 1). For simplicity, the linear forcing method is implemented in the $(\xi,\psi,\varsigma,\tau)$ space. Viewed in the physical space, the force is now applied over a wavy domain with distorted computational meshes. In this section, we investigate whether such forcing can still generate the same turbulence as in the rectangular domain case.

We consider as a canonical problem DNS in a domain bounded by wavy top and bottom boundaries with their locations given as

$$\begin{cases} z_{top} = -a_0 \sin[kx], \\ z_{bot} = -h - a_0 \sin[kx]. \end{cases}$$

Here $a_0 = 0.25$ and $k = 1; h = 5\pi$ is the height of the computational domain. The computational domain has a horizontal dimension of $2\pi\times2\pi$. Parameters for the turbulence are the same as Case 2 in Table 1.

In applying the body force as in Section 4.1, we define the central distance as $z_c = |z - (z_{top} + z_{bot})/2|$. Since the top and bottom boundaries are not flat, the force distribution is also wavy in the physical space. For comparison, we also consider another case, in which the body force is applied over a rectangular domain.

Comparison of numerical results between the two cases is shown in Fig. 8. Fig. 8(a) shows profiles of u'_{rms}. Inside the bulk region, u'_{rms} has about the same value for the two cases. Around the damping region, u'_{rms} with the curvilinear force is slightly smaller than that with the regular force. This can be explained by the distribution of the body force near the boundary of the bulk region. Fig. 8(b) shows the profiles of the horizontally averaged body force parameter $\bar{b}$. Around the damping region, $\bar{b}$ with the curvilinear force is slightly smaller than that with the regular force. As a result, turbulence in the curvilinear force case decays faster there compared to the regular force case. The difference between the two cases, however, becomes negligibly small in the free region, where the free surface is located. Therefore, we conclude that the difference caused by the curvilinear computational grid is small, and the linear forcing method can be applied directly on a wave-following computational grid.

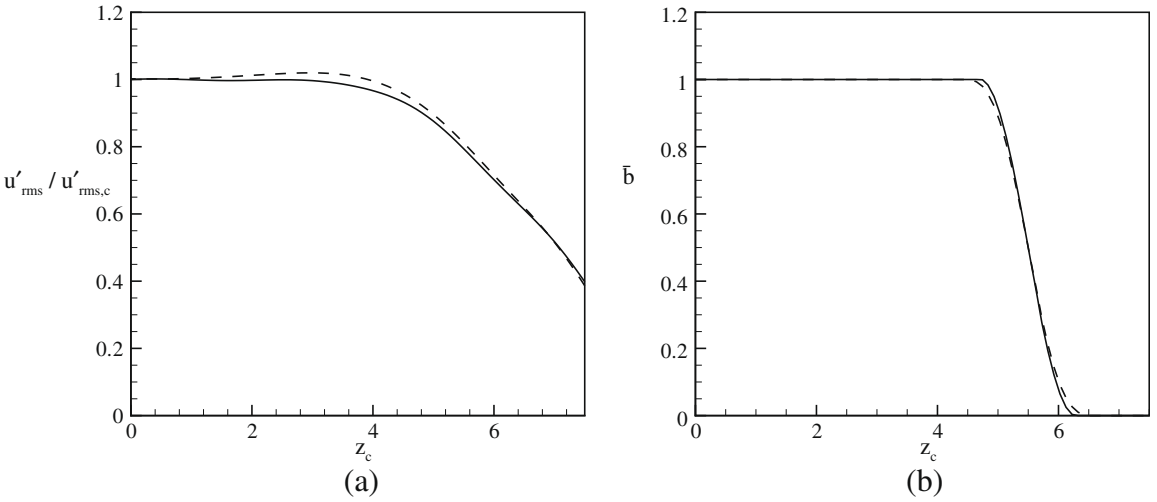

Fig. 8. (a) Profiles of horizontal velocity fluctuation; (b) Profiles of the horizontally averaged body force parameter. In both (a) and (b), —, body force applied on curvilinear wave boundary-fitted grid; – – –, body force applied on rectangular Cartesian grid.

5. Wave–turbulence interaction

To simulate wave–turbulence interaction, we use the methods introduced in Sections 3 and 4 to generate wave and turbulence, respectively. The numerical procedure, however, is not simply putting the aforementioned wave and turbulence generators together. The interaction between wave and turbulence introduces additional complexities to the problem, which require special numerical treatments. In particular, the turbulence keeps exciting the surface to generate waves with a broadband spectrum. There is a critical need to identify different wave modes efficiently during the simulation. Based on the wave components identified, we can then maintain the waves of interest and suppress unwanted waves, all done in a continuous manner throughout the simulation. For wave generation, we also need to consider the presence of other waves prior to and during the wave generation processes.

5.1. Standing wave and progressive wave identification

In the simulation of wave–turbulence interaction, due to the presence of multiple wave components, it is essential to identify the components that the study focuses on, to maintain them during the simulation, and to suppress the unwanted ones. Due to the excitation of turbulent flow from below and wave–turbulence interaction, new waves of different wavelengths are generated continuously. Meanwhile, existing waves are dissipated at different rates for different wave modes. As a result, the method for identifying wave modes in wave–turbulence interaction needs to be highly efficient based on instantaneous surface information. Strictly speaking, the reconstruction of a nonlinear wave field requires all of the essential nonlinear wave interaction processes to be considered together with an optimization procedure [32]. The state-of-the-art of such an approach, however, is currently limited to potential flow. For the much more complex case of wave–turbulence interaction studied here, we focus on efficiency of the identification algorithm by using instantaneous surface elevation and velocity data obtained from the DNS.

The surface wave is decomposed into different Fourier modes. For each mode, we further decompose it into a progress wave and a standing wave, denoted by subscripts "*p*" and "*s*," respectively:

$$\begin{cases} \eta_{p,n} = a_{p,n} \sin[k_n x + \omega_n t + \alpha_n], \\ \eta_{s,n} = a_{s,n} \sin[k_n x + \beta_n] \cos[\omega_n t + \theta_n]. \end{cases} \tag{41}$$

In the above equations, n denotes the wave mode. The wave amplitudes ($a_{p,n}$ and $a_{s,n}$) and phases (α_n, β_n, and θ_n) need to be determined. In this study, we obtain their values based on instantaneous surface elevation and vertical velocity, which can be expressed in terms of Fourier modes as

$$\begin{cases} \eta = \sum_{n=0}^{\infty} (A_n \cos[k_n x] + B_n \sin[k_n x]), \\ w = \sum_{n=0}^{\infty} (C_n \cos[k_n x] + D_n \sin[k_n x]). \end{cases} \tag{42}$$

For wave mode n, linear wave theory with Eqs. (41) and (42) yields

$$\begin{cases} a_{s,n}\sin[\beta_n]\cos[\omega_n t+\theta_n]+a_{p,n}\sin[\omega_n t+\alpha_n]=A_n,\\ a_{s,n}\cos[\beta_n]\cos[\omega_n t+\theta_n]+a_{p,n}\cos[\omega_n t+\alpha_n]=B_n,\\ -\omega_n a_{s,n}\sin[\beta_n]\sin[\omega_n t+\theta_n]+\omega_n a_{p,n}\cos[\omega_n t+\alpha_n]=C_n,\\ -\omega_n a_{s,n}\cos[\beta_n]\sin[\omega_n t+\theta_n]-\omega_n a_{p,n}\sin[\omega_n t+\alpha_n]=D_n. \end{cases} \tag{43}$$

By solving Eq. (43), we obtain the standing wave amplitude as

$$a_{s,n}=\frac{\sqrt{(\omega_n A_n+D_n)^2+(\omega_n B_n+C_n)^2}}{\omega_n}. \tag{44}$$

The value of $\beta_n-\theta_n-\omega_n t$ is determined by the following equations:

$$\begin{cases} \sin[\beta_n-\theta_n-\omega_n t]=\frac{\omega_n A_n+D_n}{\omega_n a_{s,n}},\\ \cos[\beta_n-\theta_n-\omega_n t]=\frac{\omega_n B_n-C_n}{\omega_n a_{s,n}}. \end{cases} \tag{45}$$

We rewrite the surface elevation as

$$\begin{aligned} A_n\cos[k_n x]+B_n\sin[k_n x]=a_{p,n}\sin[k_n x+\omega_n t+\alpha_n]+\frac{a_{s,n}}{2}\sin[k_n x+\omega_n t+\theta_n+\beta_n]+\frac{a_{s,n}}{2}(\cos[\beta_n-\theta_n-\omega_n t]\sin[k_n x]\\ +\sin[\beta_n-\theta_n-\omega_n t]\cos[k_n x]). \end{aligned} \tag{46}$$

From Eq. (46), we obtain

$$a_{p,n}^2+a_{s,n}a_{p,n}\cos[\beta_n+\theta_n-\alpha_n]+\frac{a_{s,n}^2}{4}=E_n^2+F_n^2, \tag{47}$$

$$\begin{cases} E_n=A_n-\frac{a_{s,n}}{2}\sin[\beta_n-\theta_n-\omega_n t],\\ F_n=B_n-\frac{a_{s,n}}{2}\cos[\beta_n-\theta_n-\omega_n t]. \end{cases} \tag{48}$$

In Eq. (47), there are two unknowns, $a_{p,n}$ and $\cos[\beta_n+\theta_n-\alpha_n]$, to be determined. Therefore, there is no unique solution for them. This is as expected because there is no unique way in the decomposition into a progressive wave and a standing wave based on instantaneous surface information. This can be seen from the fact that the five unknowns in question ($a_{p,n}, a_{s,n}, \alpha_n, \beta_n$, and θ_n) are under-determined with the four equations (43).

To obtain reasonable values for the unknowns, we assume that the progressive wave has the maximum amplitude $a_{p,n}$. This assumption is made based on the fact that in our study of wave–turbulence interaction, the progressive wave dominates and is the focus of the investigation. Our goal is to maintain the amplitude of the dominant progressive wave during the turbulence–wave interaction simulation. From our experience, we found that this assumption leads to a well maintained dominant wave, with the external disturbance associated with the pressure input kept minimum. With this assumption, we have

$$\cos[\beta_n+\theta_n-\alpha_n]=-1. \tag{49}$$

Therefore, Eq. (47) is solved as

$$a_{p,n}=\frac{a_{s,n}}{2}+\sqrt{E_n^2+F_n^2}. \tag{50}$$

By substituting Eqs. (44), (49), and (50) into Eq. (46), we determine $\omega_n t+\alpha_n$ with the following equations:

$$\begin{cases} \sin[\omega_n t+\alpha_n]=\frac{2E_n}{2a_{p,n}-a_{s,n}},\\ \cos[\omega_n t+\alpha_n]=\frac{2F_n}{2a_{p,n}-a_{s,n}}. \end{cases} \tag{51}$$

After $\omega_n t+\alpha_n$ is solved, we obtain the progressive wave $\eta_{p,n}$ according to Eq. (41).

5.2. Standing wave suppression

In Section 3.1, we introduce methods for the suppression of standing waves by applying surface pressure. The complication of free-surface turbulence is that the surface is excited by turbulence all the time. As a result, standing waves are generated continuously. Here, we focus on the δ-function method since the surface pressure is applied over a much shorter period than the other two methods.

Even for the δ-function method, there exists the issue of when to apply the surface pressure. As shown in Section 3.1.1, the δ-function method should be applied when the surface is flat to generate a standing wave to cancel the existing one. At a time instance during the simulation, however, the standing wave identified using the method in Section 5.1 is

$$\eta_s=a_s\sin[\omega t+\alpha]\sin[kx+\beta]. \tag{52}$$

In most cases, $\sin[\omega t + \alpha]$ is nonzero at the instance, so the surface is not flat. Waiting for the standing wave to reach the phase of being flat makes the algorithm inefficient. Moreover, in this waiting period new standing wave may be generated by the turbulence, so that the perfect timing is difficult to obtain in practice.

If the pressure is applied immediately, it generates another standing wave $\eta_\delta = -a_s \sin[\omega t] \sin[kx + \beta]$. Therefore, the combined surface elevation is

$$\eta_s + \eta_\delta = a_s(2\sin[\alpha])\cos[\omega t]\sin[kx + \beta]. \tag{53}$$

Eq. (53) shows that a new standing wave is generated, of which the amplitude is $|2\sin[\alpha]|a_s$. When the phase of the original standing wave is within the ranges of $0 \leqslant \alpha < \pi/6, 5\pi/6 < \alpha < 7\pi/6$, or $11\pi/6 < \alpha < 2\pi$, we have $|2\sin[\alpha]|a_s < a_s$. As a result, the new standing wave is smaller than the original one. Therefore, the strategy we take is to use the instantaneous surface information to obtain the value of α first. If α falls into the above ranges, we apply the δ-function method immediately to reduce the amplitude of the standing wave. We carry out this operation continuously in our simulation, and we find that the spurious standing wave can be controlled effectively (confirmed by our test with the standing wave suppression turned on and off, the results not shown here due to space limitation).

5.3. Turbulence–wave interaction

Finally, we apply the methods developed in the proceeding sections to set up a DNS run for the interaction of statistically steady isotropic turbulence with a well maintained progressive wave. For the DNS, the numerical method introduced in Section 2.2 is used. The parameters of Case 1 in Table 1 are used to generate turbulence. The parameters of the dominant progressive wave are $Fr^2 = 0.01, a_p = 0.1$, and $k = 1$. From Table 1 and Eq. (38), near the surface, the maximum turbulence-to-mean-flow time ratio is $S_{max}q/\varepsilon = a_p k\omega u_i^2/(2\varepsilon) = 13.88 > 10$. Therefore, this example belongs to the rapid distortion case [28,4].

We first generate isotropic turbulence underneath the free surface with the extended linear forcing method introduced in Section 4.1. Based on the instantaneous surface data, the progressive and standing wave components are identified by the algorithm introduced in Section 5.1. The standing waves are suppressed by the δ-function method discussed in Section 5.2. For the target dominant progressive wave, we use the methods introduced in Section 3.2 to increase its amplitude to $a = 0.1$, in which process the time-relaxation method introduced in Section 3.3 is implemented to account for nonlinear wave effect.

We note that for wave generation, the methods of Section 3.2 were developed for initially calm water, while in the present case there already exist surface waves excited by the turbulence. As a result of the interaction of the wave with the surface pressure applied, there exists deviation of the surface wave amplitude from the target value, for which adjustment is required. In this study, we use an energy controlling method to fine-tune the wave amplitude at the final stage of wave generation. The energy controlling method is developed based on the method of inputting energy to balance the viscous dissipation introduced in Section 3.4. For the energy input rate, instead of setting it to be equal to viscous dissipation, we prescribe it in such way that the amplitude of the progressive wave approaches the target value smoothly. We have found that the following form of amplitude evolution works fine in our numerical test:

$$a_p = \begin{cases} a_{p,0} + \frac{1}{2}(a_{p,t} - a_{p,0})[1 - \cos[n\omega(t - t_s)]] & 0 \leqslant t - t_s \leqslant \frac{\pi}{n\omega}, \\ a_{p,t} & t - t_s > \frac{\pi}{n\omega}. \end{cases} \tag{54}$$

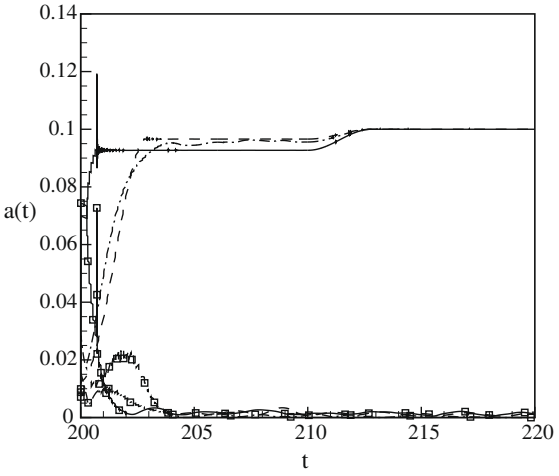

Fig. 9. History of the amplitude of progressive (lines) and standing (lines with symbols) waves. The progressive wave is generated by: —, δ-function method; – – –, time-segment method; and – · – · –, gradual method, and further adjusted by the energy controlling method. The standing wave is suppressed by the δ-function method.

Here $a_{p,0}$ and $a_{p,t}$ are, respectively, the wave amplitude values before and after the energy controlling method is applied; n is a parameter, for which a value of 2 is used here; and t_s is the time that the energy controlling method is applied.

With the above approach, the target progressive wave is set up and maintained precisely as shown in Fig. 9. (In the generation process, the spike at $t = 200 + T/4$ is caused by the wave identification associated with the second pressure impulse in the δ-function method; this spike does not affect the result afterwards as shown in Fig. 9.) Throughout the simulation, the amplitude of the target progressive wave has less than 0.1% of fluctuation. Meanwhile, standing waves are suppressed effectively, which have magnitude less than 1% of the progressive wave.

We also note that for turbulence generation in the wave field, the linear forcing should be applied only to the turbulence to assure that the wave is not distorted by the force. For this purpose, we apply Helmholtz's theorem to decompose the velocity into irrotational and rotational components [6] as $\vec{u} = \nabla\Phi + \vec{U}$. Here Φ is a velocity potential, which corresponds to the wave motion, and $\vec{U}$ is a vortical flow field, which corresponds to the turbulence.

With the above methods, we finally set up a well defined and maintained wave and turbulence field in the DNS, which enables us to perform a mechanistic study. We show an example of the effect of the wave-generated strain on turbulence velocity fluctuations. In this study, to illustrate the dependence of turbulence statistics on wave phases, we use the phase average defined as

$$\langle a\rangle(x,z) = \frac{1}{L_y}\frac{1}{T}\int_{L_y}\int_{T} a(x-ct,y,z)\,dt\,dy. \tag{55}$$

With the phase average, a variable a can be decomposed into a wave-coherent component $\langle a\rangle$ and a turbulence component $a' = a - \langle a\rangle$.

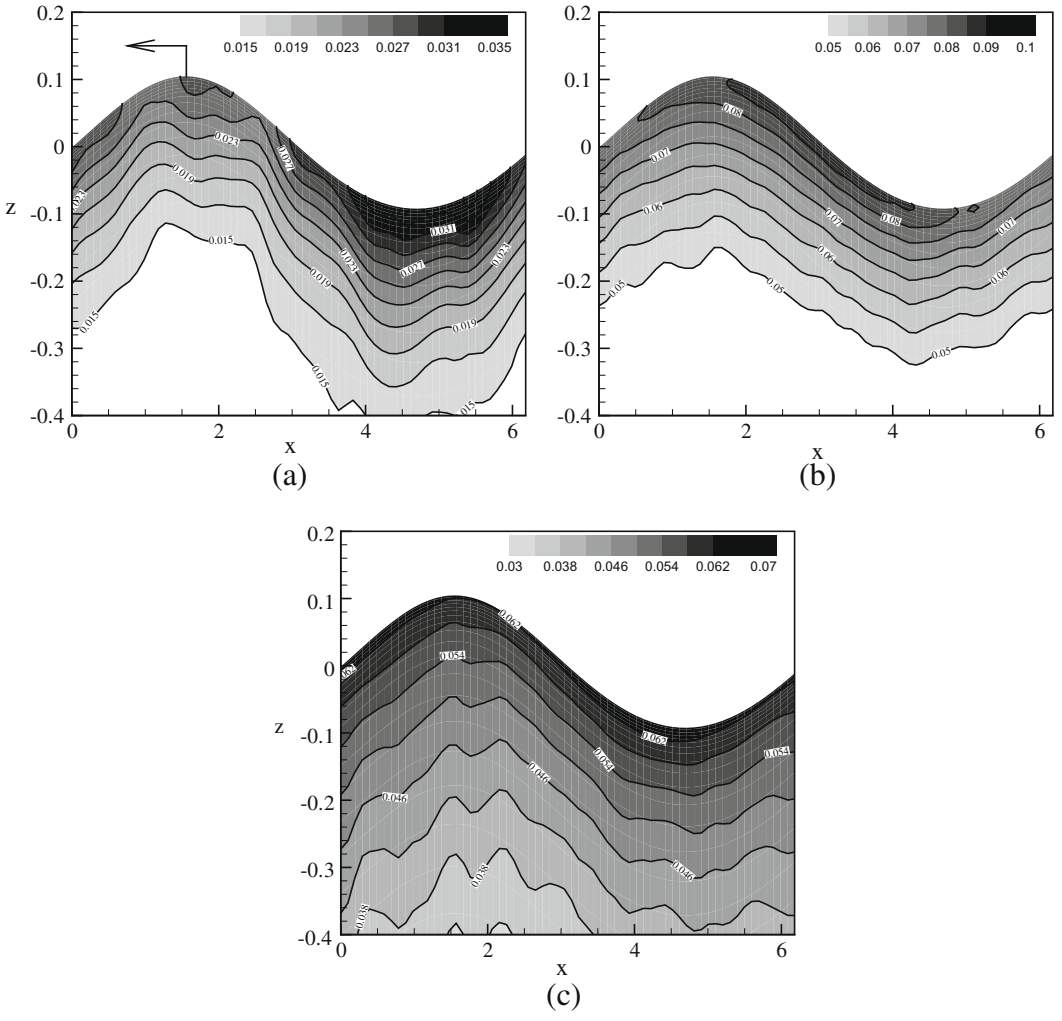

Fig. 10. Contours of (a) $\langle u'^2\rangle$, (b) $\langle v'^2\rangle$, and (c) $\langle w'^2\rangle$. Arrow shown in (a) denotes the wave propagating direction.

Fig. 10 shows the contours of $\langle u'^2\rangle, \langle v'^2\rangle$, and $\langle w'^2\rangle$. The streamwise component $\langle u'^2\rangle$ has its maximum value under the wave trough, with the minimum located under the wave crest. The spanwise component $\langle v'^2\rangle$ is also strongly phase dependent, with the largest value under the backward face of the wave [12]. We note that $\langle v'^2\rangle$ is much larger than $\langle u'^2\rangle$. For $\langle w'^2\rangle$, the variation in the streamwise direction is relatively small compared to that of $\langle u'^2\rangle$ and $\langle v'^2\rangle$. Nevertheless, the phase dependence is still evident.

The spatial variation of $\langle u'^2\rangle, \langle v'^2\rangle$, and $\langle w'^2\rangle$ shown above can be explained by the pressure–strain correlation $\langle p'\partial u_i'/\partial x_i\rangle$ (without summation over i), which measures the energy transfer among different velocity components (for free-surface turbulence applications, see for instance [17,22,23,26]). Fig. 11 plots the results. Fig. 11(a) shows that $\langle p'\partial u'/\partial x\rangle$ is positive under the backward face of the wave. As a result, $\langle u'^2\rangle$ gains energy when the backward face of the wave passes and reaches its maximum value under the trough of the wave. After the wave trough has passed by, $\langle p'\partial u'/\partial x\rangle$ becomes negative. The $\langle u'^2\rangle$ loses energy and reaches its minimum value under the forward face of the crest. Therefore, the distribution of $\langle u'^2\rangle$ is affected by the accumulative effect of the wave passage. Recall that the maximum turbulence-to-mean-flow time ratio is 13.88 in the current case. Therefore, the wave motion is relatively rapid and the turbulence reacts to the passage of different wave phases.

In the spanwise direction, $\langle p'\partial v'/\partial y\rangle$ is also strongly phase dependent as shown in Fig. 11(b). It is positive under the forward face of the wave and the wave crest, and is negative under the trough. Due to the similar accumulative effect as in the $\langle u'^2\rangle$ case, $\langle v'^2\rangle$ reaches its maximum value at the backward face of the wave after the positive $\langle p'\partial v'/\partial y\rangle$ region passes by. Afterwards, $\langle v'^2\rangle$ starts to decrease because $\langle p'\partial v'/\partial y\rangle$ is negative, and reaches its minimum value under the forward face of the wave. The area and the strength of positive $\langle p'\partial v'/\partial y\rangle$ is much larger than those of the negative one. As a result, $\langle v'^2\rangle$ gains energy.

We notice that the distribution of $\langle p'\partial w'/\partial z\rangle$ is similar to $\langle p'\partial u'/\partial x\rangle$, but with the sign reversed, indicating energy exchange between the streamwise and vertical velocity components. When the backward face of the wave passes by, energy is transferred from $\langle w'^2\rangle$ to $\langle u'^2\rangle$. Locally, the surface acts like an inclined plate moving downward. The blockage effect results

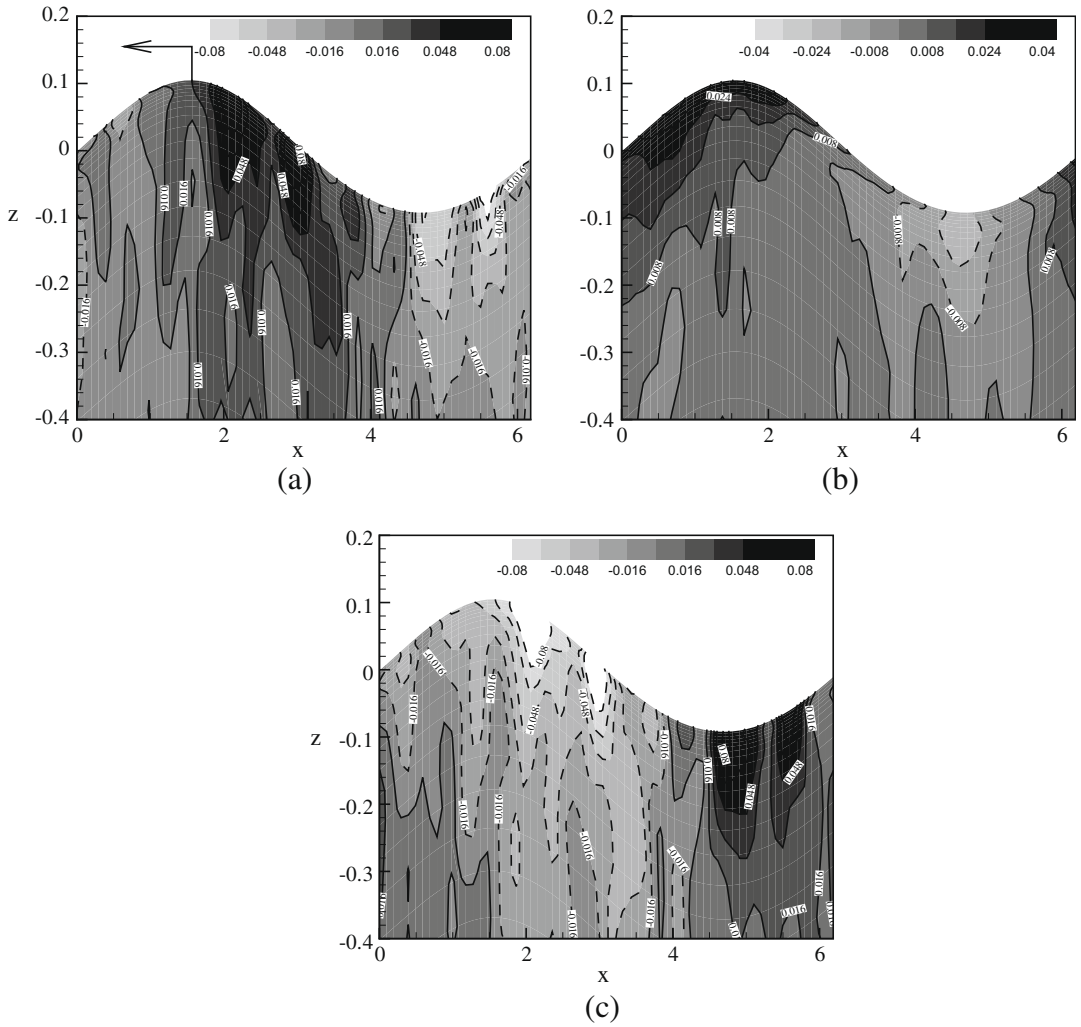

Fig. 11. Contours of (a) $\langle p'\partial u'/\partial x\rangle$, (b) $\langle p'\partial v'/\partial y\rangle$, and (c) $\langle p'\partial w'/\partial z\rangle$. Arrow shown in (a) denotes the wave propagating direction.

in an increase in streamwise motion (the increase in spanwise motion is much smaller because the flow configuration is uniform in the spanwise direction). After the trough has passed by, the opposite happens. Therefore, there exist substantial splat and antisplat interactions between the streamwise and vertical velocity components. However, the net effect on each over the entire wave length is small. As shown in Fig. 11(b), on average it is the spanwise velocity component that gains most of the energy.

6. Conclusions

In this paper, we investigate numerical methods for the setup of turbulence and surface waves in simulation of free-surface turbulence. Our object is to obtain the wave and turbulence fields precisely as specified to facilitate a mechanistic study of wave–turbulence interaction.

For the wave field, we report an approach that generates and maintains the wave components our study focuses on; this approach effectively suppresses unwanted waves, especially the spurious standing waves that have been found to be a numerical artifact. Since simulation of wave–turbulence interaction is performed with a dynamically evolving surface, an acceptable method must set up the wave field naturally without violating the wave dynamics. In this study, we introduce three methods of applying surface pressure to generate waves. Using a mollified δ-function, we generate a wave rapidly in the simulation. Alternatively, applying either a time-segment method or a gradual method, we generate a wave over a time period. We also account for the nonlinear wave effect with a time-relaxation method. The extensive numerical experiments conducted in this study show that the three methods perform well.

For the fundamental physics of wave–turbulence interaction to be investigated, it is desirable that the turbulence field in the bulk flow has simple and well defined properties. In this study, we develop a method that directly generates isotropic turbulence in the physical space of the complex wave field by extending the linear forcing method. Because the turbulence forcing must be separated from free surface processes, we apply the force in the bulk flow only. Properties of the turbulence under this inhomogeneous forcing are documented. Because the wave simulation is often performed in a wave fitted domain, we investigate effect of computational grid distortion on turbulence. Because the wave and turbulence coexist in the flow, we decompose the two and force the turbulence part only. After the above numerical treatments are implemented, we find that the quality of the isotropic turbulence generated is good. Based on these results, we conclude that the extended linear forcing method is suitable for the generation of isotropic turbulence in a wave field.

To simultaneously apply the wave and turbulence generation methods in a simulation of wave–turbulence interaction, we address several additional issues caused by wave–turbulence coupling. Because the turbulence generates and distorts waves continuously, we first develop a method to effectively identify different types of wave modes, based on which specific numerical treatments are implemented for different purposes. We modify the standing wave suppression method so that it is more effective in the turbulence field when surface disturbance is continuous. For the generation of the target dominant wave, we implement the three methods developed in this study and further develop an energy controlling method for its fine-tuning and maintenance. Using the above techniques, we find that the target dominant wave can be set up precisely, while spurious standing waves can be suppressed effectively.

We remark that the method of applying surface pressure to generate and maintain waves corresponds to the dynamics of wind input to surface gravity waves in nature. In oceans, surface waves are mainly generated by wind action through surface pressure. The energy is then redistributed among different wave components through nonlinear wave interaction, and is finally dissipated through wave breaking and wave–turbulence interaction [2]. Direct simulation of these processes is far beyond the current numerical capability, and existing ocean wave models rely heavily on parameterizations. The method developed in this paper provides a useful tool to set up wind-wave field. For example, in nature the wind-wave growth depends on the amplitude of surface pressure and the phase difference between pressure and wave form. It typically requires several hundreds of wave periods for the wave to fully develop. The method proposed in this study provides an efficient approach for the setup of the wave field within several wave periods to make numerical study affordable in terms of computational cost, while faithfully capturing the essential dynamics of the problem. In addition, the wave maintenance scheme developed in this paper establishes a framework for the mechanistic study of wave–turbulence interaction in equilibrium state of the wave field. Without such numerical treatment, direct simulation of the aforementioned processes of wind input, nonlinear interaction, and wave dissipation would be a formidable task in numerical study.

Finally, we present an example of DNS of turbulence under rapid distortion by a progressive wave. Results show that turbulence is greatly affected by wave motion in the near-surface region. Fluctuations of velocity are highly dependent on the wave phase. Energy redistribution among different velocity components is explained by the wave's straining effect through pressure–strain correlation. History of wave passage is found to play an important role in the process. These results illustrate that wave–turbulence interaction possesses many unique features with profound physics. The numerical methods developed in this study will enable us to perform a thorough investigation of the underlying mechanisms, which is a subject of our on-going research.

Acknowledgment

This research is supported by Office of Naval Research.

References

[1] V. Borue, S.A. Orszag, I. Staroselsky, Interaction of surface waves with turbulence: direct numerical simulations of turbulent open-channel flow, J. Fluid Mech. 286 (1995) 1.
[2] L. Cavaleri, J.-H.G.M. Alves, F. Ardhuin, A. Babanin, M. Banner, K. Belibassakis, M. Benoit, M. Donelan, J. Groeneweg, T.H.C. Herbers, P. Hwang, P.A.E.M. Janssen, T. Janssen, I.V. Lavrenov, R. Magne, J. Monbaliu, M. Onorato, V. Polnikov, D. Resio, W.E. Rogers, A. Sheremet, J. McKee Smith, H.L. Tolman, G. van Vledder, J. Wolf, I. Young, Wave modelling – the state of the art, Prog. Oceanogr. 75 (2007) 4.
[3] J.R. Chasnov, Simulation of the Kolmogorov inertial subrange using an improved subgrid model, Phys. Fluids A 3 (1991) 188.
[4] J. Chen, C. Meneveau, J. Katz, Scale interactions of turbulence subjected to a straining-relaxation-destraining cycle, J. Fluid Mech. 562 (2006) 123.
[5] A.A. Dimas, G.S. Triantafyllou, Nonlinear interaction of shear flow with a free surface, J. Fluid Mech. 260 (1994) 211.
[6] D.G. Dommermuth, The initialization of vortical free-surface flows, J. Fluids Eng. 116 (1994) 95.
[7] D.G. Dommermuth, The initialization of nonlinear waves using an adjustment scheme, Wave Motion 32 (2000) 307.
[8] D.G. Dommermuth, D.K.P. Yue, A high-order spectral method for the study of nonlinear gravity waves, J. Fluid Mech. 184 (1987) 267.
[9] V. Eswaran, S.B. Pope, An examination of forcing in direct numerical simulations of turbulence, Comput. Fluids. 16 (1988) 257.
[10] R.A. Handler, Jr. T.F. Swean, R.I. Leighton, J.D. Swearingen, Length scales and the energy balance for turbulence near a free surface, AIAA Paper 31 (1993) 1998.
[11] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182.
[12] B.R. Hodges, R.L. Street, On simulation of turbulent nonlinear free-surface flows, J. Comput. Phys. 151 (1999) 425.
[13] E.J. Hopfinger, J.A. Toly, Spatially decaying turbulence and its relation to mixing across density interfaces, J. Fluid Mech. 78 (1976) 155.
[14] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308.
[15] K. Lam, S. Banerjee, Investigation of turbulent flow bounded by a wall a free surface, in: E. Michaelides, M.P. (Eds.), Sharma Fundamentals of Gas–Liquid flows, ASME, 1988, p. 29.
[16] H. Lamb, Hydrodynamics, Dove, 1932.
[17] R.I. Leighton, T.F. Swean Jr., R.A. Handler, J.D. Swearingen, Interaction of vorticity with a free surface in turbulent open channel flow, AIAA Paper 91-0236, 1991.
[18] J.L. Lumley, G.R. Newman, The return to isotropy of homogeneous turbulence, J. Fluid Mech. 82 (1977) 161.
[19] T.S. Lundgren, Linearly forced isotropic turbulence, in: Annual Research Briefs, Center for Turbulence Research, Standford, vol. 461, 2003.
[20] J. Magnaudet, High-Reynolds-number turbulence in a shear-free boundary layer: revisiting the Hunt–Graham theory, J. Fluid Mech. 484 (2003) 167.
[21] C.C. Mei, M. Stiassnie, D.K.P. Yue, Theory and Applications of Ocean Surface Waves Part 1: Linear Aspects, World Scientific, 2005.
[22] Y. Pan, S. Banerjee, A numerical study of free-surface turbulence in channel flow, Phys. Fluids 7 (1995) 1649.
[23] B. Perot, P. Moin, Shear-free turbulent boundary layers Part. 1. Physical insights into near-wall turbulence, J. Fluid Mech. 295 (1995) 199.
[24] C. Rosales, C. Meneveau, Linear forcing in numerical simulations of isotropic turbulence: physical space implementations and convergence properties, Phys. Fluids. 17 (2005) 095106.
[25] L. Shen, G.S. Triantafyllou, D.K.P. Yue, Mixing of a passive scalar near a free surface, Phys. Fluids 13 (2001) 913.
[26] L. Shen, X. Zhang, D.K.P. Yue, G.S. Triantafyllou, The surface layer for free-surface turbulent flows, J. Fluid Mech. 386 (1999) 167.
[27] N.P. Sullivan, S. Mahalingam, R.M. Kerr, Deterministic forcing of homogeneous isotropic turbulence, Phys. Fluids 6 (1994) 1612.
[28] M.A.C. Teixeira, S.E. Belcher, On the distortion of turbulence by a progressive surface wave, J. Fluid Mech. 458 (2002) 229.
[29] H. Tennekes, J.L. Lumley, A First Course in Turbulence, The MIT Press, 1972.
[30] S.M. Thompson, J.S. Turner, Mixing across an interface due to turbulence generated by an oscillating grid, J. Fluid Mech. 67 (1975) 349.
[31] D.T. Walker, R.I. Leighton, L.O. Garza-Rios, Shear-free turbulence near a flat free surface, J. Fluid Mech. 320 (1996) 19.
[32] G.Y. Wu, Direct simulation and deterministic prediction of large-scale nonlinear ocean wave-field, Ph.D. Thesis, MIT, 2004.
[33] G.Y. Wu, Y.M. Liu, D.K.P. Yue, A note on stabilizing the Benjamin–Feir instability, J. Fluid Mech. 556 (2006) 45.
[34] D. Yang, L. Shen, A numerical method for direct simulation of viscous and turbulent free-surface flows, in preparation.
[35] V.E. Zakharov, Stability of periodic waves of finite amplitude on the surface of a deep fluid, J. Appl. Mech. Tech. Phys. (Eng. Transl.) 9 (1968) 86.