Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational Physics 205 (2005) 665–685

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

An efficient model for three-dimensional surface wave simulations Part I: Free space problems

Dorian Fructus, Didier Clamond, John Grue *, Øyvind Kristiansen

Mechanics Division, Department of Mathematics, University of Oslo, P.O. Box 1053, Blindern, 0316 Oslo, Norway

Received 1 July 2004; received in revised form 7 July 2004; accepted 29 November 2004
Available online 22 January 2005

Abstract

An efficient numerical scheme for simulations of fully nonlinear non-breaking surface water waves in 3D is presented. The water depth is either shallow, finite or infinite. The method is based on a fast, rapidly converging, iterative algorithm to compute the Dirichlet to Neumann operator. This is evaluated by expanding the operator as a sum of global convolution terms and local integrals with kernels that decay quickly in space. The global terms are computed very quickly via FFT. The local terms are evaluated by numerical integration. Analytical integration of the linear part of the prognostic equations in Fourier space is obtained to machine precision. The remaining nonlinear components are integrated forward in time using an RK-scheme combined with a special step size control technique. This yields a very stable and accurate time marching procedure. Zeros-padding in the spectral space represents the anti-aliasing strategy. The method requires no smoothing. Illustration through examples show that the total energy is well conserved during the numerical simulations. The scheme is stable and accurate, even for very long time simulations of very steep wave events. The scheme is easily parallelizable. It propagates for example a Stokes wave of slope 0.2985 with a phase shift error of about 0.3° after 1000 periods of propagation.

Keywords: Surface waves; Fully nonlinear; Three-dimensional; Boundary integral formulation; Pseudo-spectral method; Rapid Dirichlet to Neumann operator

* Corresponding author. Tel.: +47 2285 5839; fax: +47 2285 4349.
E-mail addresses: dorianf@math.uio.no (D. Fructus), didier@math.uio.no (D. Clamond), johng@math.uio.no (J. Grue), oyvinkri@math.uio.no (Øyvind Kristiansen).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.11.027

1. Introduction

The rapidly increasing computer power enables computations that are significantly more advanced than until recently were possible. There are problems that are so demanding that they will not be solved in the near future by only increasing the hardware capabilities, however. To solve such types of problems, it is also necessary to improve the numerical algorithms reducing substantially the amount of calculations. Direct simulations of three-dimensional surface water waves is one example. Wave propagation over long time in large domains is a very demanding numerical task, that cannot be tackled by the brute force approach. This paper describes a novel efficient numerical method for fully nonlinear three-dimensional surface wave simulations. The method is very efficient from a computational point of view and is suitable for simulations using parallel computers.

Computations of very steep ocean waves represents an important challenge. The evolution of long wave fields and the action of instability mechanisms have been focused in several studies following the pioneering works of Benjamin and Feir [2] and Zakharov [40]. Reviews of the advances can be found in [15,39]. Recent studies focus on instabilities and the formation of very big waves at sea [12,16,22,27,33]. The effect of nonlinear four-wave interactions on the formation of freak waves was studied by Janssen [21] using the NLS and the Zakharov equations. How a wave spectrum of a sea relaxes toward stationary state was recently studied by Dysthe et al. [13] using extended NLS equations in two horizontal dimensions.

Various approaches have been derived to solve exactly or approximately the surface wave problems (see [9,15,34] for reviews). These methods can be divided into three categories: the simplified equations, the high-order spectral methods and the boundary integral formulations. The oldest and simplest models are based on equations as, e.g. the nonlinear Schrödinger equation or extensions of this equation [12,33]. These equations are easily and quickly solved numerically, and for some of them even analytically. The ranges of validity are limited and they cannot be used for steep waves or long time simulations, however. High-order spectral methods are based on Taylor expansions [8,11,37]. They are computationally efficient when the series converge, meaning that the waves must not be too steep. For high-order approximations, the methods involve high-order derivatives and nonlinearities. The schemes then become inaccurate and numerically unstable. For highly nonlinear waves these methods do not converge. In practice, few terms are used. If rewritten in the Fourier space, these equations, with low-order expansions, include the Zakharov equation [40]. Other methods for three-dimensional water wave simulations also exist (see [1,19]).

The request for an improved understanding of the formation of very big waves motivated us to initiate a fully nonlinear modeling of the phenomenon in three dimensions. It is important that the method is rapid and the time integration accurate. This is fundamental, since realistic wave analysis typically is carried out for large physical domains and long time. A fully nonlinear modeling was requested by the recent ISSC-report [20], for example.

Our computational strategy is based on an integral equation formulation and extensive use of Fourier transform. This solves the Laplace equation in the fluid domain, expressing the normal velocity at the free surface in terms of the potential at the free surface and the wave elevation (the Dirichlet to Neumann operator). The dominant part of the solution is global and is obtained by fast Fourier transform (FFT). The remaining part is highly nonlinear and have integrals with kernels that decay quickly in the space coordinate. The iterative method is so rapidly convergent that one iteration is sufficient for most practical applications. In such cases the method is explicit. The method follows the mathematical formulae that were derived in Clamond and Grue [6] for an infinitely deep fluid in three dimensions and generalized to single and two layer motion with finite depths in three dimensions in Grue [17]. So far, the method was neither implemented nor tested out in practical (3D) computations.

This paper communicates important extensions of the previous publications by Clamond and Grue [6] and by Grue [17] on three main points. These include, firstly, an enhanced subdivision between the global and local contributions to the inversion of the Laplace equation solver. We find that in the three-dimensional computational scheme, it is highly useful to include one more term in the global Fourier inversion of the integral equation, as compared to the two-dimensional procedure. Further, the contributions resulting from the imaging through the sea floor are enhanced, as compared to the initial formulae given in Grue [17]. The added terms in the global evaluation of the Dirichlet to Neumann operator are evaluated by FFT which significantly speeds up the performance of the scheme in 3D. The additions to the Fourier inversion part share the beauty of analyticity which is ideal for potential manipulations.

Secondly, how to perform a highly accurate time integration of the three-dimensional problem is addressed here for the first time. This is a nontrivial extension of the time integrator in Clamond and Grue in two dimensions. The conservative form of the equations differs from the two-dimensional case, see Eqs. (20) and (21) below. Further, another decomposition of the dynamical boundary condition is employed in 3D, where the formula also includes new terms as compared to 2D. This substantially modifies the computational strategy.

Finally, the implementation of the method is tested and documented more systematically than what has been previously done. The detailed implementation of the de-aliasing technique for the products in three dimensions is highlighted. The accuracy is mainly governed by two parameters: Tol and Iter. The former sets the tolerance of the time-integrator scheme and is actively used in the step-size control. The parameter Iter determines the number of iterations in the inversion of the Laplace equation solver. Numerical experiments with Stokes waves show that using Iter = 1 and Tol = 10^{-8}, i.e., in practice an explicit scheme, leads to highly accurate results, with an error in the wave phase of 75° after 1000 periods of propagation, conserving the energy, however. This phase error is returned to zero upon reversal of the time stepping. Further tests, using an enhanced resolution, document that using Iter = 3 and Tol = 10^{-8} reduces the error in the wave phase to 0.3° after 1000 periods of propagation, returning the wave phase to 0° upon reversal of the time stepping. The same result is obtained using Iter = 5 and Tol = 10^{-8} meaning that only very few iterations are required still keeping a highly accurate and energy conserving scheme.

This paper provides a complete description of the mathematical method and its numerical implementation in three dimensions. The performance of the method is illustrated in several examples. The description is divided into this part I, focussing on wave motion in an infinite domain (using periodic boundary conditions). A complementary part II [5] describes a fully nonlinear wave generation procedure and a novel efficient wave damping strategy, with zero reflection.

This paper (part I) is organized as follows: Section 2 of the paper describes the prognostic equations (obtained from the kinematic and dynamic boundary conditions) and the integral equation solving the Laplace equation in the fluid domain. Reorganization and inversion of the integral equation using Fourier transform leads to a rapid method for the normal velocity at the free surface. Section 3 describes the numerical algorithms, the anti-aliasing technique, how to compute the global contributions from the integral equation and how to evaluate the local contributions that are fast-decaying in space. The time stepping method is further described. Section 4 describes the numerical tests, including integration of a progressive Stokes wave 10^3 periods forward in time, and return, a progressive solitary wave, and highly nonlinear evolution of crescent wave patterns. Finally, Section 5 is concluding remarks.

2. Mathematical formulation

We consider three-dimensional irrotational wave motion at the surface of a homogeneous incompressible fluid over a horizontal impermeable bottom.

2.1. Equations in the physical space

Let $\boldsymbol{x} = (x_1,x_2)$ be the horizontal Cartesian coordinates, y the upward vertical coordinate and t the time. $y = \eta(\boldsymbol{x},t)$ determines the surface elevation (with $y = 0$ the level at rest) and $y = -h$ the level of the impermeable horizontal bottom. Let $\vec{v} = (\boldsymbol{u}, v)$ be the velocity field, where $\boldsymbol{u} = (u_1,u_2)$ and v are the horizontal and vertical velocities, so that $\vec{v} = \operatorname{grad}\phi$, $\boldsymbol{u} = \nabla\phi$ and $v = \phi_y$, where ϕ denotes the velocity potential and ∇ the horizontal gradient. We denote with 'tildes' the quantities at the free surface, e.g. $\tilde{\phi}(\boldsymbol{x},t) = \phi(\boldsymbol{x}, y = \eta(\boldsymbol{x},t), t)$. Note that $\tilde{\boldsymbol{u}} = \widetilde{\nabla\phi}$ differs from $\nabla\tilde{\phi} = \tilde{\boldsymbol{u}} + \tilde{v}\nabla\eta$. At the free surface, $\tilde{\boldsymbol{u}}$ and $\tilde{v}$ are expressed by:

$$\tilde{\boldsymbol{u}} = \frac{\nabla\tilde{\phi} - V\nabla\eta + \left(\nabla\eta \times \nabla\tilde{\phi}\right) \times \nabla\eta}{1 + |\nabla\eta|^2}, \quad \tilde{v} = \frac{V + \nabla\eta \cdot \nabla\tilde{\phi}}{1 + |\nabla\eta|^2}, \tag{1}$$

where $V = \partial\phi/\partial n\sqrt{1 + |\nabla\eta|^2}$ and $\vec{n}$ denotes the outward normal of unit length at the free surface.

At the free surface, the pressure p (per unit mass) is either zero, or balances the effect of a surface tension (if taken into account), a generating pressure $\tilde{p}_{\mathrm{G}}$ (i.e., a pneumatic wavemaker) and a dissipative pressure $\tilde{p}_{\mathrm{D}}$ (i.e., a wave absorber). The free surface is generally material. For immaterial surfaces, one can also consider a prescribed vertical velocity $\tilde{v}_{\mathrm{D}}$ at the surface ($\tilde{v}_{\mathrm{D}} = 0$ for a material surface). The velocity $\tilde{v}_{\mathrm{D}}$ can be used to introduce a damping effect in order to absorb waves (similar to the damping pressure $\tilde{p}_{\mathrm{D}}$). Efficient procedures of wave generation and wave damping are addressed in the accompanying paper, Part II [5].

The kinematic and dynamic conditions at the surface can be conveniently written:

$$\eta_t - V = \tilde{v}_{\mathrm{D}}, \tag{2}$$

$$\tilde{\phi}_t + g\eta + \frac{1}{2}\tilde{\boldsymbol{u}} \cdot \nabla\tilde{\phi} - \frac{1}{2}\tilde{v}V = \sigma\nabla \cdot \left[\frac{\nabla\eta}{\sqrt{1 + |\nabla\eta|^2}}\right] - \tilde{p}_{\mathrm{G}} - \tilde{p}_{\mathrm{D}}, \tag{3}$$

where g being the acceleration of gravity and σ the (constant) surface tension coefficient. The generating pressure $\tilde{p}_{\mathrm{G}}$, the damping pressure $\tilde{p}_{\mathrm{D}}$ and velocity $\tilde{v}_{\mathrm{D}}$ are not used in Part I. They will be defined in Part II of the paper. In the numerical examples below we shall take $\tilde{p}_{\mathrm{G}} = \tilde{p}_{\mathrm{D}} = \tilde{v}_{\mathrm{D}} = \sigma = 0$.

The solution of the Laplace equation (resulting from incompressibility and irrotationality) provides a relation between the (unknown) normal velocity at the surface, defined through V below Eq. (1), and the (known) quantities $\tilde{\phi}$ and η at the surface (and their spacial derivatives). This relation defines an operator which produces the Neumann data V from the Dirichlet data $\tilde{\phi}$. This operator is often referred to as the Dirichlet to Neumann operator.

The Laplace equation, together with the bottom impermeability, is solved exactly by means of a Green function and the method of images ([6, Section 6]; [17, Section 6]), i.e.

$$\int_S \left(\frac{1}{\tilde{r}} + \frac{1}{\tilde{r}_{\mathrm{B}}}\right)\frac{\partial\phi'}{\partial n'}\mathrm{d}S' = 2\pi\tilde{\phi} + \int_S \tilde{\phi}'\frac{\partial}{\partial n}\left(\frac{1}{\tilde{r}} + \frac{1}{\tilde{r}_{\mathrm{B}}}\right)\mathrm{d}S', \tag{4}$$

where $\tilde{\phi} = \tilde{\phi}(\boldsymbol{x},t)$, $\tilde{\phi}' = \tilde{\phi}(\boldsymbol{x}',t)$, $\tilde{r}^2 = R^2 + (y' - y)^2$ and $\tilde{r}_{\mathrm{B}}^2 = R^2 + (y' + y + 2h)^2$.

Here, $R = |\boldsymbol{R}|$ denotes the horizontal distance between the field point and the source point, where $\boldsymbol{R} = \boldsymbol{x}' - \boldsymbol{x}$. In (4) S denotes the instantaneous free surface of the fluid.

In the paper [6], Clamond and Grue introduced the variable $D = [\eta(\boldsymbol{x}',t) - \eta(\boldsymbol{x},t)]/R = [\eta' - \eta]/R$. Note that $D \sim R^{-1}$ as $R \to \infty$ and $D \to \partial\eta/\partial R$ as $R \to 0$. Correspondingly, a variable D_{B} is introduced by $D_{\mathrm{B}} = [\eta' + \eta]/R_{\mathrm{B}}$ where $R_{\mathrm{B}} = \sqrt{R^2 + 4h^2}$. For non-overturning surfaces with $\mathrm{d}S' = \sqrt{1 + |\nabla'\eta'|^2}\mathrm{d}x_1'\mathrm{d}x_2'$, the integral equation may be expressed in the form

$$
\begin{aligned}
&\int \frac{V'}{(1+D^2)^{1/2}} \frac{\mathrm{d}\boldsymbol{x}'}{R} + \int \frac{V'}{(1+4hD_\mathrm{B}R_\mathrm{B}^{-1}+D_\mathrm{B}^2)^{1/2}} \frac{\mathrm{d}\boldsymbol{x}'}{R_\mathrm{B}} \\
&\quad = 2\pi\tilde{\phi} + \int \frac{\tilde{\phi}'(\boldsymbol{R}\cdot\nabla'\eta' - \eta' + \eta)}{(1+D^2)^{3/2}} \frac{\mathrm{d}\boldsymbol{x}'}{R^3} + \int \frac{\tilde{\phi}'(\boldsymbol{R}\cdot\nabla'\eta' - \eta' - \eta - 2h)}{(1+4hD_\mathrm{B}R_\mathrm{B}^{-1}+D_\mathrm{B}^2)^{3/2}} \frac{\mathrm{d}\boldsymbol{x}'}{R_\mathrm{B}^3},
\end{aligned} \tag{5}
$$

where the following convenient brief notation is used:

$$
\int \bullet \mathrm{d}\boldsymbol{x}' \equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \bullet \mathrm{d}x_1'\mathrm{d}x_2'. \tag{6}
$$

2.2. Reformulation of the boundary integrals

The direct numerical computation of V from the boundary integrals in (5) is demanding and not suitable for large domains and long time simulations. A more suitable formulation is derived rewriting the lowest-order nonlinear terms as convolutions, the remaining integrals having rapidly decaying kernels allowing truncated integrations [6]. For simplicity, we present here the deep water case only ($h = \infty$). The corresponding expansion for finite depth is given in Appendix A.

Following [6] we first exploit that

$$
\frac{\boldsymbol{R}\cdot\nabla'\eta'}{R^3} - \frac{\eta'-\eta}{R^3} = -\nabla'\cdot\left[(\eta'-\eta)\nabla'\frac{1}{R}\right]. \tag{7}
$$

Then, by application of Gauss theorem, we may partially rewrite the corresponding integral in (5). The modified and reorganized version of the equation reads (with $h = \infty$, see [6])

$$
\begin{aligned}
\int V'R^{-1}\mathrm{d}\boldsymbol{x}' = 2\pi\tilde{\phi} &+ \int (\eta'-\eta)\nabla'\tilde{\phi}'\cdot\nabla'R^{-1}\mathrm{d}\boldsymbol{x}' + \int V'R^{-1}\left[1-(1+D^2)^{-1/2}\right]\mathrm{d}\boldsymbol{x}' \\
&+ \int \tilde{\phi}'\left[1-(1+D^2)^{-3/2}\right]\nabla'\cdot\left[(\eta'-\eta)\nabla'R^{-1}\right]\mathrm{d}\boldsymbol{x}'.
\end{aligned} \tag{8}
$$

A decomposition $V = V_1 + V_2 + V_3 + V_4$ is introduced, where:

$$
\int V_1'R^{-1}\mathrm{d}\boldsymbol{x}' = 2\pi\tilde{\phi}, \tag{9}
$$

$$
\int V_2'R^{-1}\mathrm{d}\boldsymbol{x}' = \int (\eta'-\eta)\nabla'\tilde{\phi}'\cdot\nabla'R^{-1}\mathrm{d}\boldsymbol{x}', \tag{10}
$$

$$
\int V_3'R^{-1}\mathrm{d}\boldsymbol{x}' = \int \tilde{\phi}'\left[1-(1+D^2)^{-3/2}\right]\nabla'\cdot\left[(\eta'-\eta)\nabla'R^{-1}\right]\mathrm{d}\boldsymbol{x}', \tag{11}
$$

$$
\int V_4'R^{-1}\mathrm{d}\boldsymbol{x}' = \int V'R^{-1}\left[1-(1+D^2)^{-1/2}\right]\mathrm{d}\boldsymbol{x}'. \tag{12}
$$

Fourier transform is then applied to invert the equations (for the unknowns V_1, V_2, V_3 and V_4). For the left hand sides of (9)–(12) we get

$$
\mathscr{F}\left\{\int V_j'R^{-1}\mathrm{d}\boldsymbol{x}'\right\} = 2\pi k^{-1}\int V_j'\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}'}\mathrm{d}\boldsymbol{x}' = 2\pi k^{-1}\hat{V}_j, \tag{13}
$$

where the Fourier transform $\mathscr{F}$ is defined as:

$$\hat{\phi}(\boldsymbol{k},t) = \mathscr{F}\left\{\tilde{\phi}\right\} = \int \tilde{\phi}(\boldsymbol{x},t)\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}\mathrm{d}\boldsymbol{x},$$

$$\tilde{\phi}(\boldsymbol{x},t) = \mathscr{F}^{-1}\left\{\hat{\phi}\right\} = \frac{1}{4\pi^2}\int \hat{\phi}(\boldsymbol{k},t)\mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}\mathrm{d}\boldsymbol{k},$$

where $\boldsymbol{k} = (k_1,k_2)$ denotes the wavenumber, and $k = |\boldsymbol{k}|$. We have exploited that $\mathscr{F}\{R^{-1}\} = 2\pi k^{-1}\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}'}$. The transformed Eq. (9) becomes $\hat{V}_1 = k\hat{\phi}$ and, similarly, for the other terms we obtain ([6, Section 6]):

$$\hat{V}_2 = -k\mathscr{F}\{\eta V_1\} - \mathrm{i}\boldsymbol{k}\cdot\mathscr{F}\left\{\eta\nabla\tilde{\phi}\right\}, \tag{14}$$

$$2\pi\hat{V}_3 = k\mathscr{F}\left\{\int \tilde{\phi}'\left[1-(1+D^2)^{-3/2}\right]\nabla'\cdot\left[(\eta'-\eta)\nabla' R^{-1}\right]\mathrm{d}\boldsymbol{x}'\right\}, \tag{15}$$

$$2\pi\hat{V}_4 = k\mathscr{F}\left\{\int V'R^{-1}\left[1-(1+D^2)^{-1/2}\right]\mathrm{d}\boldsymbol{x}'\right\}. \tag{16}$$

In finite depth computations, a corresponding set of equations results from (A.2), see Appendix A.

2.3. Global and local integration

In [6] only the quadratic terms ($\hat{V}_2$) were written as convolutions. We push here this expansion further in writing also a part of the cubic term as convolutions ($\hat{V}_4$). This allows us to write the integral (16) as a sum of convolutions, which are computed via Fourier transform, plus an integral with a kernel that decays (even more) quickly

$$2\pi\hat{V}_4 = k\mathscr{F}\left\{\int V'R^{-1}\frac{1}{2}D^2\mathrm{d}\boldsymbol{x}'\right\} + k\mathscr{F}\left\{\int V'R^{-1}\left[1-\frac{1}{2}D^2-(1+D^2)^{-1/2}\right]\mathrm{d}\boldsymbol{x}'\right\}. \tag{17}$$

The former integral is easily obtained by convolutions, i.e.

$$k\mathscr{F}\left\{\int V'R^{-1}\frac{1}{2}D^2\mathrm{d}\boldsymbol{x}'\right\} = -\pi k\mathscr{F}\{\eta^2\mathscr{F}^{-1}\{k\mathscr{F}\{V\}\} - 2\eta\mathscr{F}^{-1}\{k\mathscr{F}\{\eta V\}\} + \mathscr{F}^{-1}\{k\mathscr{F}\{\eta^2 V\}\}\}.$$

The kernels of the inner integrals of (15) and of the second term on the right hand side of (17) decay like R^{-4} and R^{-5}, respectively. These integrals may be evaluated over a very limited region of the $\boldsymbol{x}$-plane, still keeping high accuracy.

While V_1, V_2, and V_3 are determined by known functions at the free surface, V_4 is determined implicitly. The latter is computed iteratively using an iteration procedure that may be continued until a desired accuracy is achieved. The number of iterations is in the code determined by the parameter ITER (see Section 3.2 below). In many practical computations one iteration (ITER = 1) is sufficient, however, i.e., we replace V by $V_1 + V_2 + V_3$ in the right-hand side of (17). In this case the inversion procedure of the Laplace equation is explicit. The simulations presented in Section 4 below show that an enhanced accuracy is obtained if three iterations (using ITER = 3) are performed. This is important if the time simulations are very long or of high complexity. Comparisons using ITER = 1, 3 and 5 show that the two latter give the same accuracy, while ITER = 1 gives an excellent prediction, which can be refined, however.

The solution of the Laplace equation presented here can be viewed as a "hybrid" between the expansion method [8,11,37] and the boundary integral method [10,23], taking advantage of the features of both formulations: speed and accuracy.

2.4. Remarks

In cases when $D^2 < 1$, the term $1 - (1 + D^2)^{-1/2}$ has a convergent series expansion of the form

$$1 - \left(1 + D^2\right)^{-1/2} = \tfrac{1}{2}D^2 - \tfrac{3}{8}D^4 + \tfrac{5}{16}D^6 + \cdots \quad (D^2 < 1). \tag{18}$$

This infinite expansion can formally be used to rewrite the integral V_4 in the form of a sum of convolutions only. Hence, after functional iterations, one obtains an explicit expression for V which, after some analytical manipulations, reduces to the expansion of the Dirichlet–Neumann operator due to Craig and Sulem [8]. However, this method suffers of several drawbacks (see also comments in Section 2.5 below):

(i) The series converges for $D^2 < 1$ only. There are physical situations where $|D|$ is locally close to one for non-breaking waves.
(ii) Even if the series does converge, the convergence is very slow, except for very small $|D|$. A significant gain of accuracy can therefore be expected only at the expense of considerable computations.
(iii) Since calculations are performed with finite precision arithmetics, the gain expected from higher truncations is annihilated by a large accumulation of round-off errors. Thus, even for small $|D|$, the usefulness of higher-order expansions is limited [31].
(iv) The convolution terms involve high-order derivatives that increase with the order of truncation. Computations of high-order derivatives are numerically unstable. On the other hand, computations of integrals have natural smoothing effects. Note that in the two-dimensional version of the model as it is implemented in [6], only the quadratic terms ($\hat{V}_2$) can be rewritten as convolutions keeping the scheme stable. The method is therefore better conditioned in three than in two dimensions.
(v) Another inconvenience with the convolution terms is their ill-conditioning due to cancellations. This puts additional restrictions on the efficiency of the expansion methods.
(vi) Finally, it turns out that expansions methods are competitive only at low-order truncations, i.e., to simulate waves with small steepnesses over relatively short periods of time.

If the computation of V is limited to the convolution terms, our scheme is comparable to the classical Zakharov equation [40]. This is therefore a decently accurate approximation. Thus, very fast investigations can be performed computing $V_1 + V_2 + V_4$ only (V_4 being computed replacing V by $V_1 + V_2$ in the right hand side of (17) without the integral), and when interesting results are obtained, the computations are redone with the entire solution.

In summary, our main conclusion is that the term $1 - (1 + D^2)^{-1/2}$ is best divided into the sum

$$1 - \left(1 + D^2\right)^{-1/2} = \tfrac{1}{2}D^2 - \left[\left(1 + D^2\right)^{-1/2} - 1 + \tfrac{1}{2}D^2\right],$$

where the first term ($\frac{1}{2}D^2$) contributes to a global evaluation, and the second to a local, truncated integration. This division is found to be insensitive to the magnitude of $|D|$. If $|D|$ is moderately small, the second term is correspondingly small and may be neglected.

An alternative is to split the term into

$$\tfrac{1}{2}D^2 - \tfrac{3}{8}D^4 - \left[\left(1 + D^2\right)^{-1/2} - 1 + \tfrac{1}{2}D^2 - \tfrac{3}{8}D^4\right]$$

including also $(-3/8)D^4$ in the global evaluation, and the remaining part as an even stronger local, truncated integration. This strategy has been tested and found to lead to a numerically unstable procedure.

We have found that it is important to keep terms up to cubic appearance (due to the free surface contribution), but stop a further expansion of the Dirichlet to Neumann operator. With this, we derive a highly efficient anti-aliased scheme, see Section 3.1 below. For images through the bottom (finite depth contribution) higher order convolutions are still effective, because the Fourier transform typically then involves a factor e^{-2kh}, contributing to a strong reduction of the energy at high wavenumbers.

2.5. Further comments

While standard expansion methods suffer from significant ill-conditioning problems, alternative perturbative approaches for problems in 2D, leading to a stable evaluation of the Dirichlet to Neumann operator, are studied by Nicholls and Reitich [26]. A detailed stability analysis was recently presented by Hou and Zhang [19] for the point vortex method for 2D water waves, adding stabilizing terms to the boundary integral method that exactly cancels a destabilizing term produced by using a point vortex method. Such strategies were not required with the present, non-perturbative formulation.

3. Numerical algorithms

The computational domain is rectangular, periodic in both directions and discretized with constant steps $\Delta \boldsymbol{x} = (\Delta x_1,\Delta x_2)$ over $2\boldsymbol{N} = 2(N_1,N_2)$ nodes. $\pi/\Delta x_i$ is thus the highest wavenumber (the Nyquist frequency) in the ith direction.

Simulations of non-periodic problems are achieved embedding the physical domain into a larger periodic box and using absorbing boundary conditions described in the Part II [5].

All the linear operations (e.g. differentiation) are computed in Fourier space and all the nonlinear ones in physical space. The transfers between physical and spectral space are obtained by FFT algorithms. The computations of the nonlinear terms require special care in order to avoid aliasing. We emphasize that the anti-aliasing technique explained below is the only method used to avoid numerical instabilities: we do not use, e.g. smoothing interpolations, regridding, artificial numerical viscosity, etc.

3.1. Anti-aliasing technique

For the numerical stability, it is crucial to compute the nonlinear terms free of aliasing errors ([3,4], Section 3.2).

The product between two discrete functions in physical space corresponds to a circular convolution in Fourier space (the spectra are periodic). De-aliased computations can be performed extending the spectra by zeros padding ([4], Section 3.2). The spectra must be extended by (at least) a factor of three-half for quadratic nonlinearities. Similarly, extensions by a factor four-half is required for cubic nonlinearities, by a factor of five-half for quartic nonlinearities, etc. This method becomes expensive, in terms of computational time and memory, for high-order nonlinearities. Instead, we apply repeatedly the *four-half rule*, setting to zero the extra wavenumbers each time a double or triple product is performed.

Note that, for example, it is not exactly equivalent to compute f^4 with the five-half rule and to compute $f^2 \times f^2$ applying twice the three-half rule (or to compute $f^3 \times f$ applying first the four-half rule and then the three-half rule). However, the Fourier spectra of regular functions being exponentially decaying, the error induced by repeated applications of the three-half rule as $(f^2)^2$ is comparable to the round-off error, if the discretization of f is sufficiently refined.

3.2. Computation of V

The integrals involved in V_3 and V_4 are truncated for $\boldsymbol{x}' \in [\boldsymbol{x} - \boldsymbol{\lambda};\boldsymbol{x} + \boldsymbol{\lambda}]$, $\boldsymbol{\lambda}$ being given a priori. Since the kernels of the integrals are decaying rapidly $\boldsymbol{\lambda}$ can be taken much smaller than the size of the computational box. In practice $\boldsymbol{\lambda}$ is chosen as one characteristic wavelength (e.g. the peak frequency of a JONSWAP spectrum), which is sufficient for most applications. The effect of truncation was studied in detail in [6] for the two-dimensional case. In the present three-dimensional extension of the method, the kernels decay faster than their two-dimensional counterparts, justifying the truncation over one typical wavelength. Note that the truncation depends on the wave field characteristics and not on the size of the computational domain. This means that the computational time increases linearly with the domain size, and not quadratically (or even worse) as in other methods.

The implicit term in V_4 is computed applying simple functional iterations. The number of iterations used for its evaluation is Iter. One iteration is often sufficient in practice. Of course, the required number of iterations Iter depends on the phenomenon studied and on the desired accuracy.

The domain is discretized with constant steps and the integrals are computed using the trapezoidal rule. According to the Euler–MacLaurin formula, this provides a very accurate approximation. To avoid explicit treatment of apparent singularities at $\boldsymbol{x}' = \boldsymbol{x}$, the integrands are evaluated at the nodes $\boldsymbol{x}_j = (j + \frac{1}{2})\Delta\boldsymbol{x}$, and shifted back to the regular nodes via Fourier interpolations (requiring two FFTs).

Note that the computation of the regular integrals is, by essence, easily parallelizable. The implementation is done in Fortran 90. This allows the use of multiprocessor systems, either using shared memory parallel systems (SMP), dispatched memory parallel systems (DMP) or a combination of both (hybrid solution). This is exploited by dividing the physical domain where the integrals have to be computed into sub-domains. A given sub-domain is then resolved by a single node (which may itself be a SMP system if the machine used is of hybrid type). The computation of the integrals is the more demanding part. The execution time is therefore (roughly) divided by the number of processors available. The procedure is fast, and becomes particularly interesting when considering large domains where the use of hybrid machines can really be exploited.

3.3. Time stepping

Extracting the linear parts of (2) and (3) and taking their Fourier transforms, these equations are rewritten into the skew-symmetric form

$$\hat{F}_t + \check{A}\hat{F} + \hat{R} = \hat{N}, \tag{19}$$

where:

$$\hat{R} = \begin{pmatrix} 0 \\ \frac{k\omega}{\mathscr{g}}\hat{p}_{\mathrm{G}} \end{pmatrix}, \quad \hat{F} = \begin{pmatrix} k\hat{\eta} \\ \frac{k\omega}{\mathscr{g}}\hat{\phi} \end{pmatrix}, \quad \check{A} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}, \tag{20}$$

$$\hat{N} = \begin{pmatrix} k\mathscr{F}\{V - V_1 + \tilde{v}_{\mathrm{D}}\} \\ \frac{k\omega}{2\mathscr{g}}\mathscr{F}\left\{\tilde{v}V - \tilde{\boldsymbol{u}}\cdot\nabla\tilde{\phi} + 2\sigma\nabla\cdot\left[\frac{\left(\sqrt{1+|\nabla\eta|^2}-1\right)\nabla\eta}{\sqrt{1+|\nabla\eta|^2}}\right]\right\} - \frac{k\omega}{\mathscr{g}}\hat{p}_{\mathrm{D}} \end{pmatrix}, \tag{21}$$

and $\mathscr{g} = g + \sigma k^2$ and $\omega = \sqrt{\mathscr{g}k\tanh kh}$. Eq. (19) form a conservative system of equations (if $\tilde{p}_{\mathrm{G}} = \tilde{p}_{\mathrm{D}} = \tilde{v}_{\mathrm{D}} = 0$) for the pair of dimensionless dependent variables $k\hat{\eta}$ and $k\omega\hat{\phi}/\mathscr{g}$.

The temporal resolution of such a system is very stiff for large ω, and the time step must therefore be very small. To avoid this problem, it is advantageous to introduce the change of dependent function (integrating factor [3]):

$$\hat{F}(\boldsymbol{k},t) = \exp[\check{A}(t_0 - t)]\hat{G}(\boldsymbol{k},t) - \int_{t_0}^{t}\exp[\check{A}(t' - t)]\hat{R}(\boldsymbol{k},t')\,\mathrm{d}t', \tag{22}$$

$$\hat{G}(\boldsymbol{k},t) = \exp[\check{A}(t - t_0)]\hat{F}(\boldsymbol{k},t) + \int_{t_0}^{t}\exp[\check{A}(t' - t_0)]\hat{R}(\boldsymbol{k},t')\,\mathrm{d}t', \tag{23}$$

which yields the equation

$$\hat{G}_t = \exp[\check{A}(t - t_0)]\hat{N}, \quad \hat{G}(\boldsymbol{k}, t_0) = \hat{F}(\boldsymbol{k}, t_0). \tag{24}$$

The nonlinear system (24) is then solved via a six-stages fifth-order Runge–Kutta scheme with an embedded fourth-order scheme for the time stepping control (Hairer and Wanner [18]). Since water wave problems are numerically very stiff, a stabilization of the time step is achieved by a special "PI step size control" (see [18, IV Section 2]). Numerical tests have shown that such a stabilization technique yields a substantial improvement to the method. We denote by Tol the tolerance required for the time integrator scheme.

Due to the analytic integration of the linear part, the scheme is unconditionally linearly stable and exact. For the temporal integration of the nonlinear remaining equation, many integrators are possible (including multiple steps, Taylor expansions, nonlinear, symplectic and hybrid schemes). We have found that the scheme above represents a very good compromise between: ease of implementation, speed, memory requirement, stability, accuracy and conservation of invariants.

We note that if we now neglect the right hand side of (24) (no dissipation nor nonlinearities, $\hat{N} = 0$), $\hat{G}(\boldsymbol{k}, t) = \hat{F}(\boldsymbol{k}, t_0)$. This leads to the following linear solution:

$$\begin{aligned} \hat{\eta}(\boldsymbol{k}, t) &= \hat{\eta}_0 \cos \omega(t - t_0) + \frac{\omega}{g} \hat{\phi}_0 \sin \omega(t - t_0) - \frac{\omega}{g} \int_{t_0}^{t} \hat{p}_{\mathrm{G}}(\boldsymbol{k}, t') \sin \omega(t - t')\, \mathrm{d}t', \\ \hat{\phi}(\boldsymbol{k}, t) &= \hat{\phi}_0 \cos \omega(t - t_0) - \frac{g}{\omega} \hat{\eta}_0 \sin \omega(t - t_0) - \int_{t_0}^{t} \hat{p}_{\mathrm{G}}(\boldsymbol{k}, t') \cos \omega(t - t')\, \mathrm{d}t', \end{aligned} \tag{25}$$

where $\hat{\eta}_0 = \hat{\eta}(\boldsymbol{k}, t_0), \hat{\phi}_0 = \hat{\phi}(\boldsymbol{k}, t_0)$. Once the term $\hat{p}_{\mathrm{G}}(\boldsymbol{k}, t)$ is defined, one can explicitly express analytically the linear solution (some explicit solutions for different functions $\hat{p}_{\mathrm{G}}(\boldsymbol{k}, t)$ are given in the part II [5]).

4. Numerical tests

We have chosen three different test cases relevant to typical problems of surface gravity waves. We present first two sets of simulations of long-crested progressive waves in deep and shallow water (i.e., a Stokes wave and a solitary wave). We also illustrate the method with a purely unsteady three-dimensional application (horse shoe) for which accurate experimental data are available. Our primary goal is to illustrate the method and to show that the procedure is fast, robust and very accurate.

The accuracy of the simulations are governed by the two parameters: Tol and Iter, where the former is used in the step-size control, and the latter determines the number of iterations in the inversion of the Laplace equation solver.

4.1. Progressive Stokes wave

We first consider the propagation in deep water of an exact Stokes wave of steepness $ak = 0.2985$ (a the half wave height and k the wavenumber). The initial condition (i.e., at $t = 0$) for the free surface elevation η and the potential at the surface $\tilde{\phi}$ are obtained from Fenton's exact numerical solver [14]. The computational domain is quadratic and includes two wavelengths in each horizontal direction. It is discretized over 64×64 collocation points, meaning that the first seven harmonics of the Stokes wave are resolved.

In an attempt to evaluate the accuracy of both the temporal integrator scheme and of the evaluation procedure of V, we assess the problem of reversibility of the solution. We study the evolution of the phase shift $\Delta\varphi$ and of the total energy (defined by $2E = \int(\eta_t\tilde{\phi} + g\eta^2)\mathrm{d}\boldsymbol{x}$) in a long time forth-and-back simulation. The forward simulation is carried out up to $t = 10^3 T_0$, where T_0 is the period of the Stokes wave. Then, the time is reversed and the backward simulation is performed down to the initial time $t = 0$. Simulations are done with different values of the tolerance parameter TOL and the parameter ITER.

Fig. 1 presents the results of these simulations. In Fig. 1(a), the free surface elevation is plotted at $t = 0$, $t = 10^3 T_0$ and $t = 0$ (after backward simulation), computed with TOL = 10^{-8} and ITER = 3. The effect of the parameters TOL and ITER on the convergence of the model are depicted in Fig. 1(b) and (c), respectively. The evolution of the total energy (with TOL = 10^{-8} and ITER = 3) is shown in Fig. 1(d). The results in Fig. 1 motivate for the following comments:

(i) The method converges quickly with the number of iterations, ITER. Although there are substantial differences (after long time) between one and three iterations, more than three iterations do not provide further improvements. For very accurate long time simulations it is therefore sufficient to employ ITER = 3. For short simulations, one iteration is sufficient, most of the time, however.

(ii) The reversibility of the scheme increases as the tolerance parameter TOL decreases. With TOL = 10^{-8} (at least for this particular long time forward–backward simulation) the initial condition is perfectly recovered (Fig. 1(a) and (b)). This means that the errors due to the temporal integrator, to the evaluation of the Green integral and the accumulation of round-off errors are insignificant.

(iii) The total energy is relatively well conserved (Fig. 1(d)) and, more importantly, no increasing or decreasing trend is observed. The absence of a trend is crucial for the reversibility of the scheme. The integrator used is an explicit Runge–Kutta scheme, and one may expect the total energy to increase during the simulation, however. Such a global trend is observed in long time simulations if TOL = 10^{-6} is used (results not shown here). Setting TOL = 10^{-8} reduces an increase in the total energy error to a very low value. Even after 2000 periods of propagation, we do not observe any trend. On the other hand, usual smoothing techniques or filtering decreases the total energy, often in a dramatic way. Since neither smoothing nor filtering is used here, the total energy is allowed to be conserved. With no smoothing techniques used, one could then expect high frequencies of the numerical solution to blow up. Indeed the long time simulations for large waves show that the scheme is highly stable.

(iv) The phase shift error obtained from the numerical solution is only of about 18° after 1000 periods of propagation (Fig. 1(a) and (b)). This demonstrates the efficiency of the evaluation procedure of V.

To further check the convergence of the method, we perform a series of simulations where we fix TOL = 10^{-8} and ITER = 3, varying the number of collocation nodes in both directions. All the simulations are carried out from $t = 0$ up to $t = 10^3 T_0$. We evaluate, during each simulation, the relative error in the effective period (or alternatively the small phase shift) as compared to the period of the Stokes wave T_0. The relative error (in percents) is $\varepsilon_{\text{err}} = 100|T - T_0|/T_0$ where $T = 2\pi/\omega$ denotes the effective period. Here ω is evaluated numerically from the phase shift by $\omega t = \omega_0 t + \Delta\varphi$ where $\Delta\varphi$ is given in radians at $t = mT_0$ and where $\omega_0 = 2\pi/T_0$. Table 1 presents the relative error for several resolutions. It can be seen from these examples that the method converges quickly with increasing number of nodes in both directions. For the simulations corresponding to a purely two-dimensional problem, the strong contribution to an enhanced accuracy due to a refinement in the lateral direction, increasing the value of N_2, may seem surprising. This is due to the fact that the evaluation of the local integrals are performed in both directions, however. It can be noted that the simulations with a discretization of 64 × 128 collocation points lead to a very small relative error in the period (see Table 1). The phase shift error obtained in this case is only of about one third of a degree after 1000 periods of propagation.

4.2. Progressive solitary wave

To further illustrate the efficiency of the method, we consider as well a problem in shallow water. A solitary wave is propagated along the numerical tank in a forward time simulation. Then the simulation is

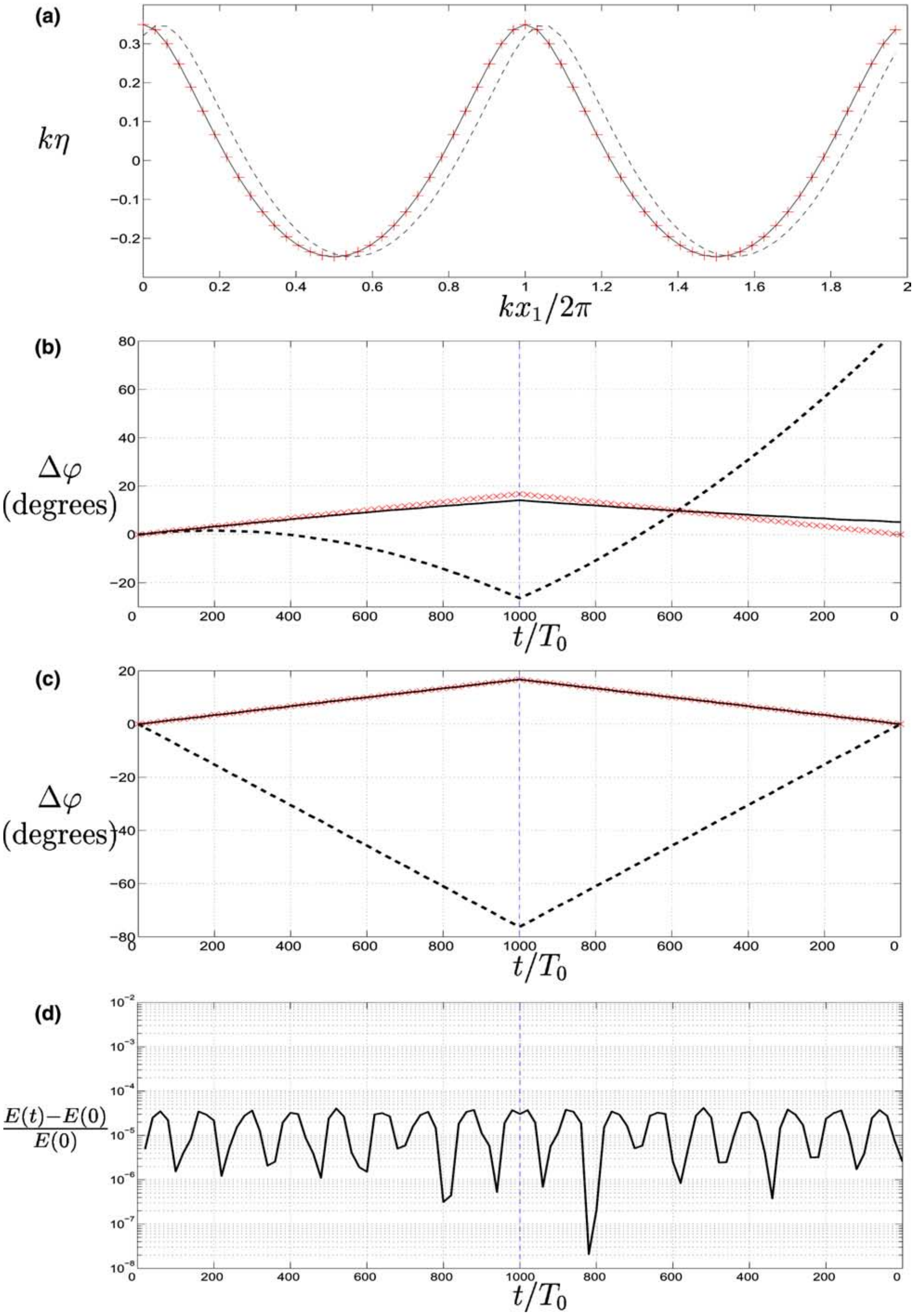

Fig. 1. Forward–backward simulation of a Stokes wave ($ak = 0.2985$) discretized over 64×64 collocation points. (a) Surface at: (—) $t = 0$, (– –) $t = 10^3 T_0$, (+) $t = 0$ (after backward integration in time); (Tol $= 10^{-8}$, Iter $= 3$). (b) Evolution of phase error $\Delta\varphi$ with Iter $= 3$ and (– –) Tol $= 10^{-6}$, (—) Tol $= 10^{-7}$, (×) Tol $= 10^{-8}$. (c) Evolution of the phase error $\Delta\varphi$ with Tol $= 10^{-8}$ and (– –) Iter $= 1$, (—) Iter $= 3$, (×) Iter $= 5$. (d) Relative error of the total energy (Tol $= 10^{-8}$, Iter $= 3$).

Table 1
Relative error (percent) of the wave period $\varepsilon_{\text{err}} = 100|T - T_0|/T_0$ computed for a Stokes wave $ak = 0.2985$

$N_2 + N_2$	$N_1 + N_1$			
	16	32	64	256
16	4.1958	0.1509	0.0234	0.0238
32	4.2035	0.1650	0.0115	0.0116
64	4.2044	0.1752	0.0046	0.0058
128	4.2044	0.1747	0.0001	0.0001

The computational domain includes two wavelengths in each direction and is discretized over $2N_1 \times 2N_2$ collocation points. T_0 corresponds to the exact wave period computed from [14]. The simulated wave period T is evaluated from the obtained phase shift after $1000T_0$ of propagation. Using Iter = 3 and Tol = 10^{-8}. A relative error of 0.0001% corresponds to about 0.3° phase shift after 1000 periods of propagation.

reversed, as in the previous example. For illustration, we consider a solitary wave with $a/h = 0.6$ (a denotes the maximal excursion of the wave, h the water depth). The initial condition is obtained using an algorithm by Tanaka [32] which computes the exact solitary wave solution. The solitary wave is discretized over (4096×32) collocation points, with $\Delta x_1/h = 0.2$ and $\Delta x_2/h = 0.3$. Fig. 2 presents the time evolution of the free surface elevation while the Fig. 3 compares the elevation at $t = 0$, $t = 250\sqrt{h/g}$ and after backward integration in time. Here the numerical parameters have been taken as $\boldsymbol{\lambda}/h = (10,10)$, Tol = 10^{-7} and Iter = 3. The initial condition is recovered after backward integration in time, similarly to the previous example (Fig. 3).

These two examples illustrate the efficiency of the method. Considering two-dimensional exact solutions in infinite depth and in shallow water, we show that the time integration scheme is very efficient while the obtained solution is very accurate. Owing to the use of fast algorithm for the evaluation of Discrete Fourier Transform, the scheme is very fast while the anti-aliasing technique allows us to avoid the use of smoothing. This leads to computations where, even on very long time simulations, the total energy is conserved. The solution is accurate and the computation is stable.

4.3. Crescent wave patterns

While the physics in the previous examples were purely two-dimensional, we now illustrate the potential of the method computing three-dimensional instabilities of Stokes waves. We study the evolution of large amplitude Stokes waves perturbed by a small initial three-dimensional perturbation. We focus on the emergence of crescent wave patterns due to class II instabilities (following McLean's description [24]). The evolution of such class II instabilities have been the topic of many studies. Experimental investigations [29,25] as well as theoretical [28] and numerical ones [38] provide a good understanding of the underlying phenomenon responsible for the formation of crescent wave patterns. Two satellites of wavevectors $\boldsymbol{k}_a$ and $\boldsymbol{k}_b$ and respective frequencies ω_a and ω_b, may get into resonance with the fundamental of the Stokes wave, $\boldsymbol{k}_0 = (1,0)k_0$ and frequency ω_0, in a quintet interaction if the following resonance condition is fulfilled:

$$\begin{cases} \boldsymbol{k}_a + \boldsymbol{k}_b = 3\boldsymbol{k}_0, \\ \omega_a + \omega_b = 3\omega_0. \end{cases} \tag{26}$$

Let us consider an initial perturbation with dominant wavevectors $\boldsymbol{k}_a = (1 + p,q)k_0$ and $\boldsymbol{k}_b = (2 - p,-q)k_0$ where p and q are arbitrary real numbers. If (p,q) lies in the class II instability zone, the quintet interaction (26) leads to the formation of crescent wave patterns. The most unstable perturbation of class II corresponds to $p = 1/2$. The corresponding pattern, known as horse-shoe pattern, is phase-locked to the unperturbed Stokes waves. Many observations of such patterns have been reported while another interesting three-dimensional pattern was generated experimentally by Collard and Caulliez [7]. This pattern, named

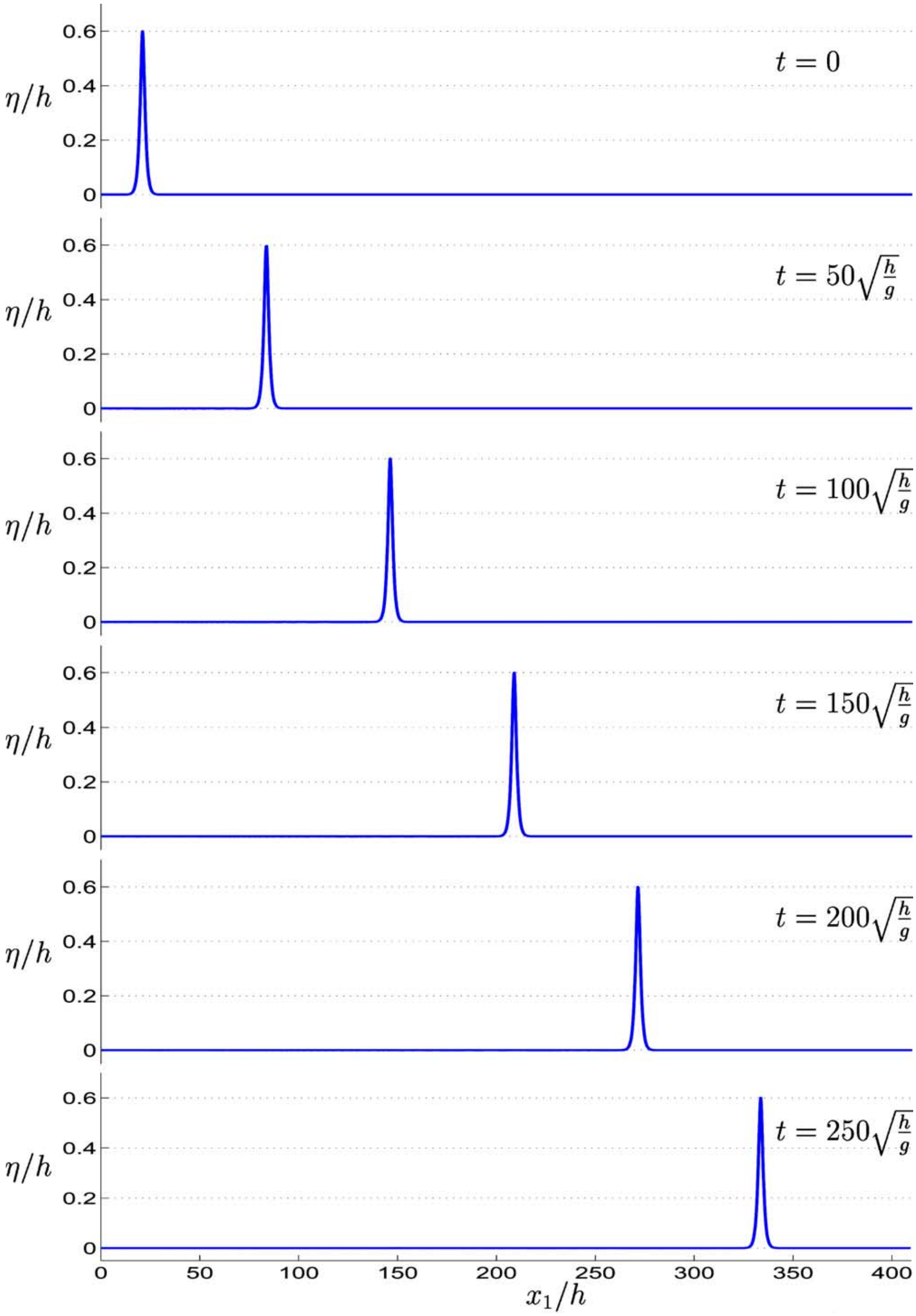

Fig. 2. Propagation of a solitary wave $a/h = 0.6$ with Tol = 10^{-7} and Iter = 3. Elevation at $t = \{0; 50; 100; 150; 200; 250\}\sqrt{h/g}$.

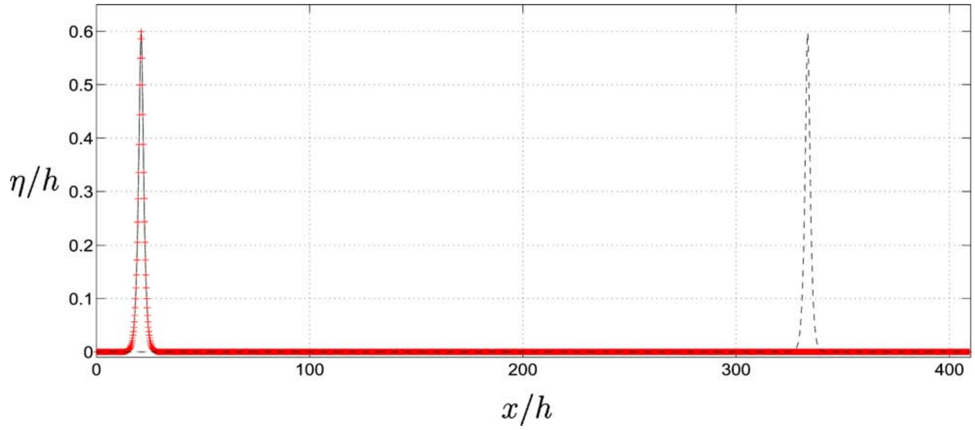

Fig. 3. Same as Fig. 2 in a simulation back and forth in time. Surface elevation at: – $t = 0$, – – $t = 250\sqrt{h/g}$, + $t = 0$ (after backward integration in time).

oscillating horse-shoe pattern, corresponds to the development of the unstable mode at $p = 0$. The pattern was in the wave tank experiments caused by a perturbation that fits with the finite width of the tank.

In order to reproduce numerically such patterns with the present model, we compute first an exact numerical solution of plane Stokes waves of steepness $ak_0 = 0.2985$ [14]. A small perturbation η' of the wave surface, taking the form $\eta' = \varepsilon a \sin[(1 + p)k_0 x_1]\cos(qk_0 x_2)$, is added to the Stokes wave train. Here, ε is a small number, making the amplitude of the initial perturbation field a fraction of the Stokes waves. For each simulation, four periods of Stokes waves in the longitudinal direction and three periods of the perturbation in the transversal direction are resolved by (128×64) nodes. The values of $(p,|q|)$ are chosen as (0.5,1.25) in the first simulation and as (0,1.32) and (1,1.32) in the second one. These choices correspond to unstable modes in the (p,q)-plane, according to the analysis of the Class II instability [24]. The corresponding wave vectors involved in (26) are then $\boldsymbol{k}_0 = (1,0)k_0$, $\boldsymbol{k}_1 = (1.5,1.25)k_0$ and $\boldsymbol{k}_1^* = (1.5, -1.25)k_0$ in the first simulation. Two quintets are involved in the second simulation: $\boldsymbol{k}_0, \boldsymbol{k}_2 = (1, 1.32)k_0$ and $\boldsymbol{k}_3^* = (2, -1.32)$ for the first quintet and $\boldsymbol{k}_0$, $\boldsymbol{k}_3 = (2,1.32)$, $\boldsymbol{k}_2^* = (1, -1.32)k_0$ for the second one. The growth of the crescent formed waves is then computed up to the level when breaking occurs in the simulations, corresponding to a saturation of the perturbations in real conditions like in a wave laboratory. We compare the simulated wave fields prior to breaking with observations that are available. It is noted that the present simulations differ from the physical laboratory experiments on a very important point: in the experiments the resonant quintets are balanced by the effect of breaking, preventing a further growth of the perturbation waves, contributing to a steady state. The latter is not achieved by a conservative modeling.

The evolution of the modal energy for the wave vectors involved in the two simulations are presented in Fig. 4.

Fig. 4(a) presents the results from the first simulation. The dynamics leads to the formation of horse-shoe patterns through the quintet interaction $\boldsymbol{k}_1 + \boldsymbol{k}_1^* = 3\boldsymbol{k}_0$.

Fig. 4(b) is concerned with the results from the second simulation. The dynamics leads to the formation of oscillating horse-shoe patterns. Here the two quintet interactions are $\boldsymbol{k}_2 + \boldsymbol{k}_3^* = 3\boldsymbol{k}_0$ and $\boldsymbol{k}_3 + \boldsymbol{k}_2^* = 3\boldsymbol{k}_0$. The continuous growth is accompanied by an oscillatory energy transfer between the modes (due to the resonant interaction $\boldsymbol{k}_2 + \boldsymbol{k}_3^* = \boldsymbol{k}_3 + \boldsymbol{k}_2^*$).

Fig. 5(a) and (b) presents the free surface elevation corresponding to horse-shoe and oscillating horse-shoe patterns, respectively, while Fig. 5(c) and (d) presents the corresponding wave frequency spectra. The superharmonic frequency peaks that appear in Fig. 5(c) and (d) confirms the second condition for

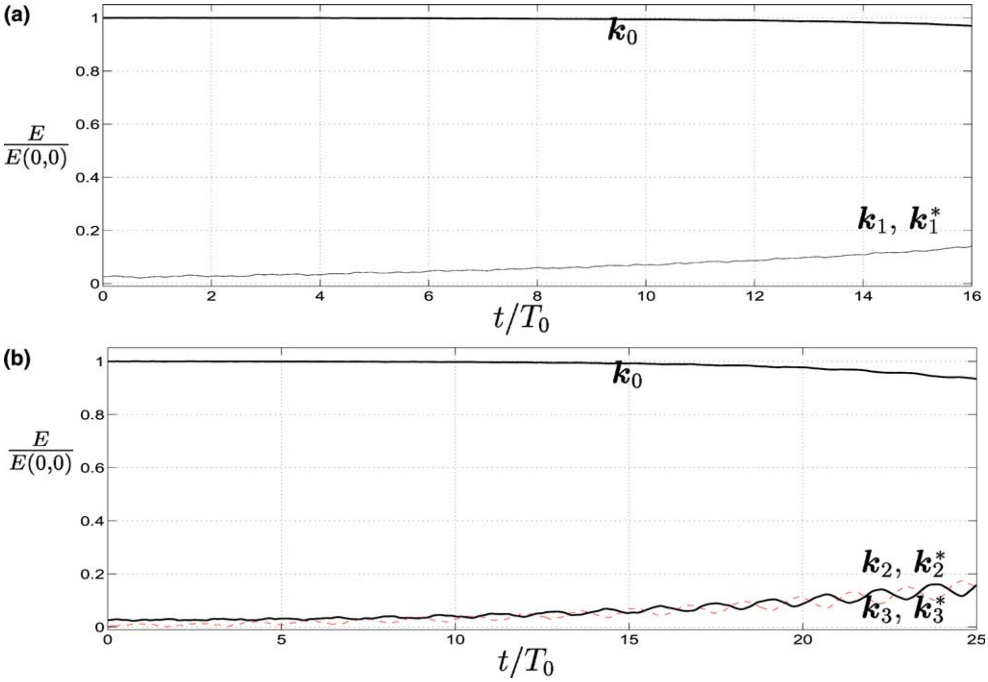

Fig. 4. Evolution of spectral energy for the main modes. (a) $\boldsymbol{k}_0 = (1,0)$, $\boldsymbol{k}_1 = (1.5,1.25)$, $\boldsymbol{k}_1^* = (1.5,-1.25)$. The initial wave field is a Stokes wave with $(ak)_0 = 0.3$ and the initial perturbations were set at $\boldsymbol{k}_1$ and $\boldsymbol{k}_1^*$ with $\varepsilon = 0.025$. (b) $\boldsymbol{k}_0 = (1,0)$, $\boldsymbol{k}_2 = (1,1.32)$, $\boldsymbol{k}_2^* = (1,-1.32)$, $\boldsymbol{k}_3 = (2,1.32)$, $\boldsymbol{k}_3^* = (2,-1.32)$. The initial wave field is a Stokes wave with $(ak)_0 = 0.3$ and the initial perturbations were set at $\boldsymbol{k}_2$ and $\boldsymbol{k}_2^*$ with $\varepsilon = 0.025$.

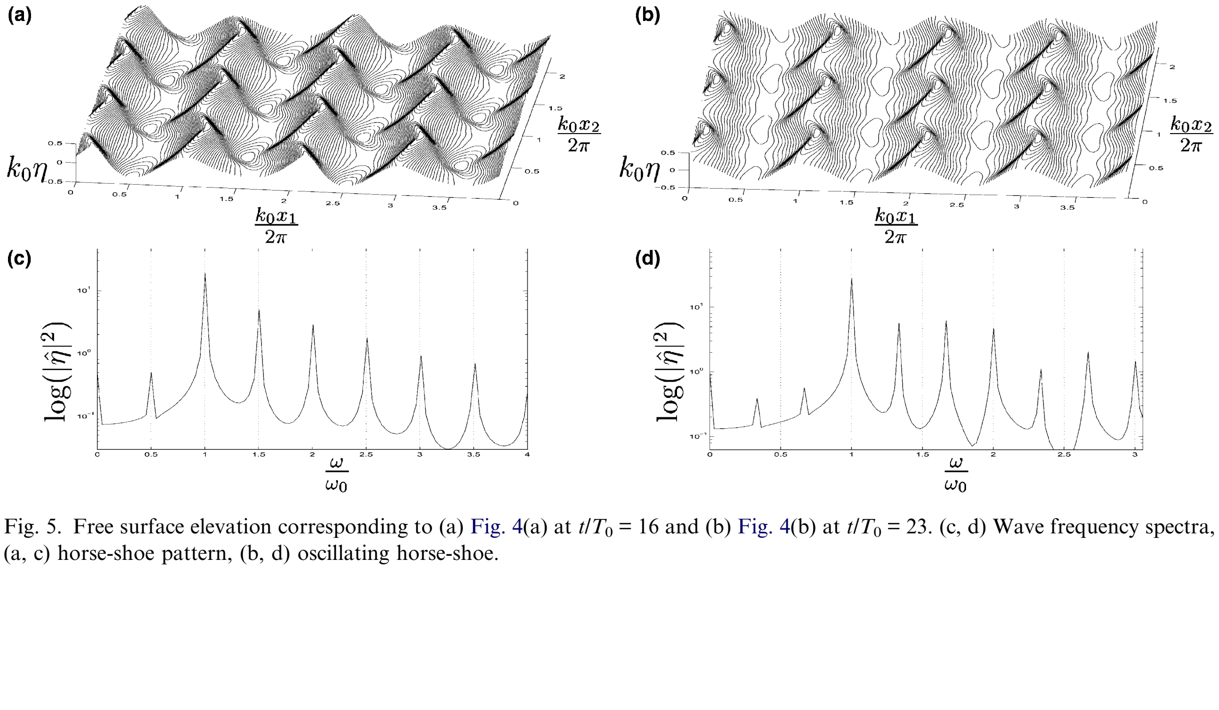

Fig. 5. Free surface elevation corresponding to (a) Fig. 4(a) at $t/T_0 = 16$ and (b) Fig. 4(b) at $t/T_0 = 23$. (c, d) Wave frequency spectra, (a, c) horse-shoe pattern, (b, d) oscillating horse-shoe.

the quintet interaction (26). While the peaks corresponding to ω_0,$2\omega_0$,... are due to the basic Stokes wave, additional ones at $1.5\omega_0$,... (Fig. 5(c)) are due to $\boldsymbol{k}_1$ and $\boldsymbol{k}_1^*$ while the peaks at $4/3\omega_0$,$5/3\omega_0$,... (Fig. 5(d)) correspond to $\boldsymbol{k}_2, \boldsymbol{k}_3^*, \boldsymbol{k}_3$ and $\boldsymbol{k}_2^*$.

For the sake of validation of the present results, a direct comparison with the experiments on the steady horse shoes are performed. Su [29] used five parameters to describe the waves (Fig. 6). Table 2 presents Su's results compared to the ones we obtained from the simulations just before breaking. Our results has a maximum relative difference from the observations of less than one percent for four of the parameters and six percent for the remaining parameter. For comparison, the results from the numerical simulation of Xue et al. [38] have been added in Table 2. Their simulations differ from ours, however, in the sense that the initial level of disturbance was much higher in their simulations (they used $\varepsilon = 0.16$) and that they considered slightly steeper initial Stokes waves ($ak = 0.33$) in their simulations. The agreement between the experimental results and the present numerical predictions is gratifying. This means that the final stage of the computed steady horse-shoe pattern corresponds to the experimentally observed one, both in amplitude and wave length.

Secondly, we perform comparisons with experimental results of Collard and Caulliez [7] for the oscillating horse-shoe patterns. We consider first the experimental observations of the oscillatory horse-shoe pattern, at the stage where saturation is reached. The log-plot of the energy density shows a ratio between the perturbation modes and the fundamental being ~1.5/3.5 [7]. The corresponding result for the computations is ~2/3.5, see the Fig. 5(d). This means that the simulations of the oscillatory horse-shoe patterns are carried out up to a level corresponding to the experimental observations where a further growth of the waves is limited by wave breaking.

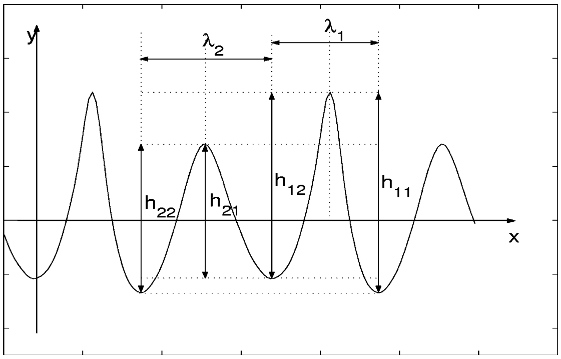

Fig. 6. Definition sketch of Su's geometrical parameters for the steady horse-shoe pattern. Wave elevation along propagation direction, through main (every second) maximum perturbation peak (see Fig. 5(a)).

Table 2
Comparison of characteristic crescent wave geometrical parameters for which quasi-steady state is assumed to be reached

	λ_1/λ_2	h_{11}/h_{12}	h_{21}/h_{22}	h_{11}/h_{21}	$S_{\max}$
Su [29]	1.28	1.10	0.88	1.66	0.65
Present model (Fig. 5(a))	1.28	1.11	0.88	1.56	0.66
Xue et al. [35] $t = t_1$	1.05	1.18	0.81	1.49	0.45
Xue et al. [35] $t = t_2$	1.07	1.09	0.88	1.64	0.64
Xue et al. [35] $t = t_3$	1.03	1.18	0.80	1.63	0.69
Xue et al. [35] $t = t_4$	1.17	1.26	0.75	1.60	0.51

$S_{\max} = \mathrm{Max}(|\eta_x|)$ denotes the maximum slope.

For the results concerning the steady horse shoes, the computations were performed on a 1.3 GHz Pentium 4M processor (laptop). The resolution was (128 × 64) nodes and the numerical parameters were taken as Tol = 10^{-7} and Iter = 3. The total computational time for each of these two simulations was ≈5500 s.

5. Concluding remarks

We have developed a numerical method to simulate fully nonlinear non-overturning free-surface waves. The water depth is either finite or infinite. While other methods exist to solve this problem, they suffer from several drawbacks. Numerical stability of the solution is usually a crucial point and is, in practice, a very limiting factor. One has often to use smoothing techniques to avoid numerical blow-up of the solution. Physically, this means that global parameters such as the total energy are not well conserved during simulations of wave field evolution. Moreover, numerous methods use expansion techniques about the free surface level at rest. This implies limitation to small to moderate local steepnesses of the solution. Finally, the time execution is as well a very limiting factor, making large scale simulations unrealistic.

The strategy we present here is based on an integral equation formulation and extensive use of Fourier transform. At each step in the time integrator scheme, the Dirichlet to Neumann operator (solution of the Laplace equation) is evaluated by expanding the operator through a highly truncated sum of convolution terms, plus some integrals, representing the remainder in a perturbation expansion, with kernels that decay quickly in space. The global convolution terms are computed very quickly via FFTs. The local contributions, due to the integrals, are evaluated by numerical integration over limited regions in the horizontal plane. Moreover, the computation of the integrals (both global FFTs and local integrations) is, by essence, easily parallelizable. This allows implementation on hybrid architectures and leads to very fast evaluation of the local terms.

The Dirichlet to Neumann operator is evaluated by an iterative procedure that is rapidly converging. In many applications one iteration is sufficient. In this case the inversion procedure of the Laplace equation is explicit. The number of iterations is in the code governed by the parameter Iter. Very accurate predictions are obtained using Iter = 3. This is important for long time simulations in large tanks. There is no gain in using a higher value of Iter.

In order to limit the aliasing errors, the spectra are extended using zeros-padding. This rather expensive procedure cannot be carried out indefinitely, however. It is here implemented for products that are up to the fourth power. Extraction of more convolution terms from the integrals over the horizontal plane is theoretically possible. This leads to even faster decaying contributions of the remaining integrals. We find that such a procedure is accompanied by increasing numerical instability, however. This is due to aliasing errors from higher order nonlinear terms which are difficult to tackle in practical computations.

Analytical integration of the linear part of the prognostic equations means that this part of the solution is obtained to machine precision. The remaining nonlinear part is integrated using a highly accurate time integrator combined with a special step size control technique. This yields a very stable and accurate time integration scheme. We illustrate in three examples that the method converges quickly and that the errors due to the temporal integrator, to the evaluation of the Green integral and the accumulation of round-off errors are insignificant. The method can cope with the description of very steep waves and requires no smoothing. The total energy is perfectly conserved during the numerical simulations and the scheme is shown to be stable and accurate, even for very long time simulations of very steep wave events.

Owing to the development of new technologies, including the democratization of 64 bits machines and multi-processor architectures, the size of the numerical tank appears no longer as a drastic limitation. Large computations in both time and space of fully nonlinear water waves are now possible. Simulations of evolution of wave spectra or freak wave generation using the present model are realistic in the near future.

The present model limits itself by the periodic wave tank. For many physical applications one has to consider a tank of infinite length. If one wants to simulate an infinite channel the present model has to be modified to include wave damping at the border of the tank. Fully nonlinear wave generation and wave absorption are described in the accompanying paper, part II [5].

Acknowledgements

This work was conducted under the Strategic University Programme 'Modeling of currents and wave for sea structures' and the BmatA-programme 'Computational methods for stratified flows involving internal waves', both funded by the Research Council of Norway.

Appendix A. Boundary integrals for the bottom

As for the surface integrals, the kernels of the contributions due to the bottom in (5) are partially expanded around $D_B = 0$ (i.e., around $\eta = 0$). We introduce $R_B{}^2 = R^2 + (2h)^2$, and the part of the kernel which is a polynomial in D_B reduces to convolutions and can be computed via Fourier transforms.

Thus, applying the Fourier transform of (5) and using the relation

$$\mathscr{F}\{R_B^{-1}\} = 2\pi e_h k^{-1} \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}'}, \quad e_h \equiv \mathrm{e}^{-2kh}, \tag{A.1}$$

one obtains after some algebra (see [17, Section 6.1])

$$\begin{aligned}\mathscr{F}(V) = {} & k\tanh(kh)\mathscr{F}(\tilde{\phi}) - k\tanh(kh)\mathscr{F}\{\eta V_1\} - \mathrm{i}\boldsymbol{k}\cdot\mathscr{F}\{\eta\nabla\tilde{\phi}\} + kC_h[e_h\mathscr{F}(\eta(V - V_1)) \\ & + \mathscr{F}(\eta\mathscr{F}^{-1}[e_h\mathscr{F}(V - V_1)])] + kC_h[\mathscr{F}(T(\tilde{\phi}) + T_1(\tilde{\phi})) + \mathscr{F}(N(V) + N_1(V))],\end{aligned} \tag{A.2}$$

where $V_1 = \mathscr{F}^{-1}[k\tanh(kh)\mathscr{F}(\tilde{\phi})]$, $C_h \equiv 1/(1 + e_h)$ and

$$T(\tilde{\phi}) = \frac{1}{2\pi}\int \tilde{\phi}'[1 - (1 + D^2)^{-3/2}]\nabla'\cdot\left[(\eta' - \eta)\nabla'\frac{1}{R}\right]\mathrm{d}\boldsymbol{x}', \tag{A.3}$$

$$N(V) = \frac{1}{2\pi}\int \frac{V'}{R}[1 - (1 + D^2)^{-1/2}]\mathrm{d}\boldsymbol{x}', \tag{A.4}$$

$$\begin{aligned}N(V) = {} & \frac{1}{2\pi}\int \frac{V'}{R}[1 - (1 + D^2)^{-1/2} - D^2/2]\mathrm{d}\boldsymbol{x}' - \frac{k}{2}\mathscr{F}(\eta^2 V) + \frac{1}{2}\mathscr{F}(2\eta\mathscr{F}^{-1}[k\mathscr{F}(\eta V)]) \\ & - \frac{1}{2}\mathscr{F}(\eta^2\mathscr{F}^{-1}[k\mathscr{F}(V)]),\end{aligned} \tag{A.5}$$

$$T_1(\tilde{\phi}) = -\frac{1}{2\pi}\int \tilde{\phi}'\frac{12h^2(\eta' + \eta)}{R_B^5}\mathrm{d}\boldsymbol{x}' + \frac{1}{2\pi}\int \tilde{\phi}'[\mathbf{R}\cdot\nabla'\eta' - (\eta' + \eta) - 2h]\left[\frac{1}{r_1^3} - \frac{1}{R_B^3}\right]\mathrm{d}\boldsymbol{x}', \tag{A.6}$$

$$\begin{aligned}T_1(\tilde{\phi}) = &-\frac{1}{2\pi}\int \tilde{\phi}'\left(\frac{280h^4}{R_{\mathrm{B}}^2} - 30h^2\right)\frac{(\eta'+\eta)^3}{R_{\mathrm{B}}^7}\mathrm{d}\boldsymbol{x}' + \frac{1}{2\pi}\int \tilde{\phi}'[\mathbf{R}\cdot\nabla'\eta' - (\eta'+\eta) - 2h]\\ &\cdot\left[\frac{1}{r_1^3} - \frac{1}{R_{\mathrm{B}}^3} + \frac{6h(\eta'+\eta)}{R_{\mathrm{B}}^5} - \left(\frac{30h^2}{R_{\mathrm{B}}^2} - \frac{3}{2}\right)\cdot\frac{(\eta'-\eta)^2}{R_{\mathrm{B}}^5}\right]\mathrm{d}\boldsymbol{x}' + \frac{1}{2}e_h\mathrm{i}\mathbf{k}\cdot\mathscr{F}(\eta^2\nabla\tilde{\phi})\\ &+\frac{1}{2}\mathscr{F}(2\eta\mathscr{F}^{-1}[e_h\mathrm{i}\mathbf{k}\cdot\mathscr{F}(\eta\nabla\tilde{\phi})]) - \frac{1}{2}\mathscr{F}(\eta^2\mathscr{F}^{-1}[e_hk^2\mathscr{F}(\tilde{\phi})]) - \frac{1}{6}e_hk\mathrm{i}\mathbf{k}\cdot\mathscr{F}(\eta^3\nabla\tilde{\phi})\\ &-\frac{1}{6}\mathscr{F}(3\eta\mathscr{F}^{-1}[e_hk\mathrm{i}\mathbf{k}\cdot\mathscr{F}(\eta^2\nabla\tilde{\phi})]) - \frac{1}{6}\mathscr{F}(3\eta^2\mathscr{F}^{-1}[e_hk\mathrm{i}\mathbf{k}\cdot\mathscr{F}(\eta\nabla\tilde{\phi})])\\ &+\frac{1}{6}\mathscr{F}(\eta^3\mathscr{F}^{-1}[e_hk^3\mathscr{F}(\tilde{\phi})]),\end{aligned} \tag{A.7}$$

$$N_1(V) = \frac{1}{2\pi}\int V'\left(\frac{1}{R_{\mathrm{B}}} - \frac{2h(\eta'+\eta)}{R_{\mathrm{B}}^3} - \frac{1}{r_1}\right)\mathrm{d}\boldsymbol{x}', \tag{A.8}$$

$$\begin{aligned}N_1(V) = &\frac{1}{2\pi}\int V'\left(\frac{1}{R_{\mathrm{B}}} - \frac{2h(\eta'+\eta)}{R_{\mathrm{B}}^3} - \frac{1}{r_1}\right)\mathrm{d}\boldsymbol{x}' + \frac{1}{2\pi}\\ &\times\int V'\left(\left(\frac{6h^2}{R_{\mathrm{B}}^2} - \frac{1}{2}\right)\frac{(\eta'+\eta)^2}{R_{\mathrm{B}}^3} - \left(\frac{20h^3}{R_{\mathrm{B}}^2} - 3h\right)\frac{(\eta'+\eta)^3}{R_{\mathrm{B}}^5}\right)\mathrm{d}\boldsymbol{x}' - \frac{k}{2}e_h\mathscr{F}(\eta^2V)\\ &-\frac{1}{2}\mathscr{F}(2\eta\mathscr{F}^{-1}[ke_h\mathscr{F}(\eta V)]) - \frac{1}{2}\mathscr{F}(\eta^2\mathscr{F}^{-1}[ke_h\mathscr{F}(V)]) + \frac{1}{6}e_hk^2\mathscr{F}(\eta^3V)\\ &+\frac{1}{6}\mathscr{F}(3\eta\mathscr{F}^{-1}[e_hk^2\mathscr{F}(\eta^2V)]) + \frac{1}{6}\mathscr{F}(3\eta^2\mathscr{F}^{-1}[e_hk^2\mathscr{F}(\eta V)]) + \frac{1}{6}\mathscr{F}(\eta^3\mathscr{F}^{-1}[e_hk^2\mathscr{F}(V)])\end{aligned} \tag{A.9}$$

Eqs. A.2,A.3,A.4,A.6 and (A.8) are equal to the single layer special case of the transient interface equations in Grue [17]. Eqs. (A.5), (A.7) and (A.9) are results of further expansions, and it is these versions which are implemented. The surface integral involving $\tilde{\phi}$ can be expanded in even orders in nonlinearity. If expanded to sixth order, the fourth order convolution term is unstable. Therefore, this integral is given above to fourth order. The surface integral involving V can be expanded in odd orders. The cubic convolution terms is usually stable, hence this integral is given above as fifth order. Each of the bottom integrals are expanded in both even and odd orders in nonlinearity (i.e., 1, 2, 3,...). Their fourth order convolution terms are found to be stable, hence the integrals are given above as fifth order. The bottom integrals could probably be expanded even further without the convolution terms becoming unstable.

The factor $e_h \equiv e^{-2kh}$ which appears in most convolution terms is numerically fortunate since it acts like a low-pass filter, allowing even the quartic (and probably higher order) convolutions to be unhampered by high-frequency noise.

References

[1] J.T. Beale, A convergent boundary integral method for three-dimensional water waves, Math. Comp. 70 (235) (2001) 977–1029.

[2] T.B. Benjamin, J.E. Feir, The disintegration of wave trains on deep water, J. Fluid Mech. 27 (1967) 417–430.

[3] J.P. Boyd, Chebyshev and Fourier Spectral Methods, Dover, New York, 2004.

[4] C. Canuto, M.Y. Haussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics. Springer Series in Computational Physics, Springer-Verlag, Berlin, 1987.

[5] D. Clamond, D. Fructus, J. Grue, Ø. Kristiansen, An efficient model for three-dimensional surface wave simulations. Part II: Generation and absorption, J. Comp. Phys. (in press).

[6] D. Clamond, J. Grue, A fast method for fully nonlinear water wave computations, J. Fluid Mech. 447 (2001) 337–355.

[7] F. Collard, G. Caulliez, Oscillating crescent-shaped water wave patterns, Phys. Fluid Lett. 11 (1999) 3195–3197.

[8] W. Craig, C. Sulem, Numerical simulation of gravity waves, J. Comp. Phys. 108 (1993) 73–83.

[9] F. Dias, C. Kharif, Nonlinear gravity and capillary-gravity waves, Annu. Rev. Fluid Mech. 31 (1999) 301–346.

[10] J.W. Dold, D.H. Peregrine, Water-wave modulation, in: Proceedings of the 20th International Conference on Coastal Engineering, American Society of Civil Engineers, Taipei, 10–14 November 1986, pp. 163–175.

[11] D. Dommermuth, D.K.P. Yue, A high-order spectral method for the study of nonlinear gravity waves, J. Fluid Mech. 184 (1987) 267–288.

[12] K.B. Dysthe, Note on a modification to the nonlinear Schrödinger equation for application to deep water, Proc. R. Soc. Lond. A 369 (1979) 105–114.

[13] K.B. Dysthe, K. Trulsen, H.E. Krogstad, H. Socquet-Juglard, Evolution of a narrow-band spectrum of random surface gravity waves, J. Fluid Mech. 478 (2003) 1–10.

[14] J.D. Fenton, The numerical solution of steady water wave problems, Comp. Geosci. 14 (3) (1988) 357–368.

[15] J.D. Fenton, Numerical methods for nonlinear wavesAdvances in Coastal and Ocean Engineering, vol. 5, World Scientific, Singapore, 1999, pp. 241—324.

[16] M. Francius, C. Kharif, On the disappearance of the lowest-order instability for steep gravity waves in finite depth, Phys. Fluids 15 (2003) 2445–2448.

[17] J. Grue, On four highly nonlinear phenomena in wave theory and marine hydrodynamics, Appl. Ocean Res. 24 (2002) 261–274.

[18] E. Hairer, G. Wanner, Solving ordinary differential equations II. Stiff and differential-algebraic problemsSpringer Series in Computational Mathematics, vol. 14, Springer-Verlag, Berlin, 1991.

[19] T.Y. Hou, P. Zhang, Convergence of a boundary integral method for 3D water waves, Discrete Contin. Dyn. Syst. Ser. B 2 (1) (2002) 1–34.

[20] ISSC 2000, Report of the Environment Committee, in: 14th International Ship and Offshore Structures Congress, Nagasaki, Japan.

[21] P.A.E.M. Janssen, Nonlinear four-wave interactions and freak waves, J. Phys. Oceangr. 33 (2003) 863–884.

[22] C. Kharif, E. Pelinovsky, T. Talipova, A. Slunyaev, Focusing of nonlinear wave groups in deep water, JETP Lett. 73 (2001) 170–175.

[23] 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 350 (1976) 1–26.

[24] J.W. McLean, Instability of finite amplitude water waves, J. Fluid Mech. 114 (1982) 315–330.

[25] W.K. Melville, The instability and breaking of deep-water waves, J. Fluid Mech. 115 (1982) 165–185.

[26] D.P. Nicholls, F. Reitich, Stability of high-order perturbative methods for the computation of Dirichlet–Neumann operators, J. Comput. Phys. 170 (1) (2001) 276–298.

[27] M. Onorato, A.R. Osborne, M. Serio, Extreme wave events in directional, random oceanic sea states, Phys. Fluid 14 (2002) 25–28.

[28] V.I. Shrira, S.I. Badulin, C. Kharif, A model of water wave 'horse-shoe' patterns, J. Fluid Mech. 318 (1996) 375–404.

[29] M.Y. Su, Three-dimensional deep-water waves. Part 1. Experimental measurement of skew and symetric wave patterns, J. Fluid Mech. 124 (1982) 73–108.

[30] C. Skandrani, C. Kharif, J. Poitevin, Nonlinear evolution of water surface waves: the frequency down-shift phenomenon, Comtemp. Math. 200 (1996) 157–171.

[31] M. Tanaka, The stability of solitary waves, Phys. Fluids 29 (1986) 650–655.

[32] K. Trulsen, I. Kliakhadler, K.B. Dysthe, M.G. Velarde, On weakly nonlinear modulation of waves on deep water, Phys. Fluids 12 (10) (2000) 2432–2437.

[33] W.T. Tsai, D.K.P. Yue, Computation of nonlinear free-surface flows, Annu. Rev. Fluid Mech. 28 (1996) 249–278.

[34] B.J. West, K.A. Brueckner, R.S. Janda, D.M. Milder, R.L. Milton, A new numerical method for surface hydrodynamics, J. Geophys. Res. 92 (11) (1987) 11803–11824.

[35] M. Xue, H. Xü, Y. Liu, D.K.P. Yue, Computations of fully nonlinear three-dimensional wave–wave and wave–body interactions. Part 1. Dynamics of steep three-dimensional waves, J. Fluid Mech. 438 (2001) 11–39.

[36] H.C. Yuen, B.M. Lake, Nonlinear dynamics of deep-water gravity waves, Adv. Appl. Mech. 22 (1982) 67–229.

[37] V.E. Zakharov, Stability of periodic wave of finite amplitude on the surface of a deep fluid, J. Appl. Mech. Phys., Engl. Transl. 2 (1968) 190–198.