J Eng Math (2007) 58:149–156
DOI 10.1007/s10665-006-9106-6

ORIGINAL PAPER

A note on time integrators in water-wave simulations

Didier Clamond · Dorian Fructus · John Grue

Received: 6 February 2006/Accepted: 12 October 2006 / Published online: 28 November 2006

Abstract The importance of a suitable temporal integrator for fully nonlinear simulations of surface gravity waves is emphasized. Via numerical examples, it is demonstrated that constant-step procedures are inefficient. This relates to the practice of energy-conserving symplectic integration, assuming constant time steps, and is compared to direct numerical simulations using Runge–Kutta integrators with variable time-step control. It is concluded that the latter with automatic variable time-step control is the more efficient, and should be applied. The practice and efficiency of a stabilization procedure for the time-step selection is described and illustrated. An important point of the method is that the linear part of the prognostic equations is integrated analytically, which means that this part is obtained to machine presicion for any (large) time step (time interval). The evolution and instabilities of highly nonlinear water waves in three dimensions are exemplified through an accurate and efficient time-integration procedure.

Keywords Accurate time-integration · Direct numerical simulations · Nonlinear water waves · Symplectic method · Variable time-step control

1 Introduction

Publications on strongly nonlinear numerical wave tanks mainly concern the efficient solution of the Laplace equation (see [1,2] for reviews of the different methods). This is because the inversion of the Laplace equation represents the most demanding part of the computations. Little attention has been paid

We are grateful to Professor J. N. Newman for his longstanding and pioneering contributions to the research field of marine hydrodynamics. His analytical and numerical works on fundamental and industrial problems in relation to water waves and their interaction with floating bodies have inspired us and many fellow scientist world wide over long time.

D. Clamond
Institutt For Energiteknikk, Postboks 40, 2027 Kjeller, Norway
e-mail: didier.clamond@ife.no

D. Fructus · J. Grue (✉)
Mechanics Division, Department of Mathematics, University of Oslo, P.O. Box 1053 Blindern, 0316 Oslo, Norway
e-mail: johng@math.uio.no

D. Fructus
e-mail: dorianf@math.uio.no

to the time marching of the wave field, however. Simple, constant-step procedures are generally employed using Runge–Kutta, Adams–Bashford or Taylor-series expansion methods. However it is well known in the theory of numerical integration of differential equations that constant-time-step methods are inefficient, [3,4]. This is also true for simulations of the motion of surface waves which will be emphasized in this note. Our aim is to illustrate that the temporal integrator is almost as important as the Laplace-equation solver and should not be overlooked.

We investigate the motion of surface gravity waves in deep water. Let $\boldsymbol{x} = (x_1, x_2)$ denote horizontal Cartesian coordinates, y the upward vertical coordinate and t the time. Let $y = 0$ and $y = \eta(\boldsymbol{x}, t)$ be the equations of the free-surface elevation at rest (the still water level) and in motion, respectively. We assume that the flow is described by potential theory and let ϕ denote the velocity potential. The horizontal fluid velocity is then obtained by $\boldsymbol{u} = (u_1, u_2) = \nabla\phi$, where ∇ denotes the horizontal gradient, and the vertical velocity by $v = \phi_y$. We denote with 'tildes' the quantities at the free surface, e.g. $\tilde{\phi}(\boldsymbol{x}, t) = \phi(\boldsymbol{x}, y = \eta(\boldsymbol{x}, t), t)$. At the free surface, the pressure is zero and the effect of surface tension is neglected. The kinematic and dynamic conditions at the surface can be conveniently written [5] as follows;

$$\partial\eta/\partial t \; - \; V = 0, \tag{1}$$

$$\partial\tilde{\phi}/\partial t \; + \; g\eta + \frac{|\nabla\tilde{\phi}| - V^2 - 2V\nabla\eta\cdot\nabla\tilde{\phi} + |\nabla\eta\times\nabla\tilde{\phi}|^2}{2(1 + |\nabla\eta|^2)} = 0, \tag{2}$$

where the equations are evaluated at $y = \eta$. Further, $V = \partial\phi/\partial n[1 + |\nabla\eta|^2]^{1/2}$ denotes the scaled normal velocity at the free surface, where n denotes the unit normal pointing out of the fluid, and g denotes the acceleration of gravity.

The solution of the Laplace equation, resulting from conservation of mass of the incompressible flow, is obtained exactly by means of a Green function, i.e.,

$$\int_S \frac{1}{r}\frac{\partial\phi'}{\partial n'}\mathrm{d}S' = 2\pi\tilde{\phi} + \int_S \tilde{\phi}'\frac{\partial}{\partial n}\frac{1}{r}\mathrm{d}S', \tag{3}$$

where $\tilde{\phi} = \tilde{\phi}(\boldsymbol{x}, t)$, $\tilde{\phi}' = \tilde{\phi}(\boldsymbol{x}', t)$ and r is the distance between the source point and the field point (at the free surface) and is expressed by $r = [R^2 + (y' - y)^2]^{1/2}$ where $R = |\boldsymbol{x}' - \boldsymbol{x}|$. We introduce the difference in the surface elevation at the two points $\boldsymbol{x}$ and $\boldsymbol{x}'$ divided by the horizontal distance, i.e., $D = (\eta' - \eta)/R$, and note that $D \to \partial\eta/\partial R$ when $R \to 0$ and $D \to 0$ when $R \to \infty$. The reorganized integral equation (3) may be put on the form

$$\int\!\!\int_{-\infty}^{\infty} \frac{V'}{R}\,\mathrm{d}\boldsymbol{x}' = 2\pi\,\tilde{\phi} + \int\!\!\int_{-\infty}^{\infty} (\eta' - \eta)\nabla'\tilde{\phi}'\cdot\nabla'\frac{1}{R}\,\mathrm{d}\boldsymbol{x}' \; + \; 2\pi\,T(\tilde{\phi}) \; + \; 2\pi\,N(V), \tag{4}$$

where the integration is over the (horizontal) $\boldsymbol{x}'$-coordinate; the functions $T(\tilde{\phi})$ and $N(V)$ are given below, and we have used that $\mathrm{d}S' = [1 + |\nabla\eta'|^2]^{1/2}\mathrm{d}\boldsymbol{x}'$. Equation 4 is inverted by means of a Fourier transform [5,6]. The solution is obtained by a rapidly converging iterative procedure where the leading terms are explicit:

$$\mathcal{F}(V^{(1)}) = k\,\mathcal{F}(\tilde{\phi}), \tag{5}$$

$$\mathcal{F}(V^{(2)}) = \mathcal{F}(V^{(1)}) - k\mathcal{F}(\eta V^{(1)}) - \mathrm{i}\boldsymbol{k}\cdot\mathcal{F}(\eta\nabla\tilde{\phi}), \tag{6}$$

where $\mathcal{F}$ denotes Fourier transform, $\boldsymbol{k} = (k_1, k_2)$ the wavenumber vector and $k = |\boldsymbol{k}|$. The full solution is obtained by solving iteratively the relation (one iteration is usually sufficient)

$$\mathcal{F}(V) = \mathcal{F}(V^{(2)}) + k\mathcal{F}[N(V) + T(\tilde{\phi})], \qquad \text{where} \tag{7}$$

$$N(V) = \frac{1}{2\pi}\int\!\!\int_{-\infty}^{\infty} \frac{V'}{R}\,[\,1 - (1 + D^2)^{-1/2}\,]\,\mathrm{d}\boldsymbol{x}', \tag{8}$$

$$T(\tilde{\phi}) = \frac{1}{2\pi}\int\!\!\int_{-\infty}^{\infty} \tilde{\phi}'\,[1 - (1 + D^2\,)^{-3/2}]\,\nabla'\cdot\left[\,(\eta' - \eta)\,\nabla'\frac{1}{R}\,\right]\,\mathrm{d}\boldsymbol{x}'. \tag{9}$$

Generalizations including the effect of a horizontal sea bed, overturning surface, surface tension, generating pressure and absorbing boundaries are given in [5–8]. The claim of the present paper applies identically to all these generalizations which are not considered here for the sake of simplicity.

2 The temporal scheme

Extracting the linear parts of (1)–(2) and taking their Fourier transforms, we may write the prognostic equations as

$$\vec{Y}_t + \mathbb{A}\,\vec{Y} = \vec{N}, \tag{10}$$

with

$$\vec{Y} = \begin{pmatrix} k\,\mathcal{F}(\eta) \\ \frac{k\omega}{g}\,\mathcal{F}(\tilde{\phi}) \end{pmatrix}, \quad \mathbb{A} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}, \quad \vec{N} = \begin{pmatrix} k\,\mathcal{F}\left\{V - V^{(1)}\right\} \\ \frac{k\omega}{2g}\,\mathcal{F}\left\{\tilde{v}\,V - \tilde{\boldsymbol{u}}\cdot\nabla\tilde{\phi}\right\} \end{pmatrix} \tag{11}$$

where $\omega = \sqrt{gk}$ and $\mathcal{F}(V^{(1)}) = k\mathcal{F}(\tilde{\phi})$ is obtained from (5). Equation 10 forms a conservative system of equations for the pair of dimensionless dependent variables $k\mathcal{F}(\eta)$ and $k\omega\mathcal{F}(\tilde{\phi})/g$. The left-hand side of (10) includes all the linear terms and the right-hand side all the nonlinear ones. This equation must be supplemented with the initial condition $\vec{Y} = \vec{Y}_0$ for $t = t_0$.

To avoid (linear) instabilities and thus a slow scheme, it is advantageous to integrate the linear part analytically. This is easily done with the change of variables

$$\vec{Y}(\boldsymbol{k},t) \;=\; \exp[\mathbb{A}(t_0 - t)]\,\vec{Z}(\boldsymbol{k},t), \quad \vec{Z}(\boldsymbol{k},t) \;=\; \exp[\mathbb{A}(t - t_0)]\,\vec{Y}(\boldsymbol{k},t), \tag{12}$$

where

$$\exp(\mathbb{A}T) \;=\; \begin{bmatrix} \cos\omega T & -\sin\omega T \\ \sin\omega T & \cos\omega T \end{bmatrix}. \tag{13}$$

This yields the equation

$$\vec{Z}_t = \exp[\mathbb{A}(t - t_0)]\,\vec{N}, \quad \vec{Z}(\boldsymbol{k},t_0) = \vec{Y}(\boldsymbol{k},t_0). \tag{14}$$

After transformation, the temporal scheme is necessarily unconditionally linearly stable and exact. This is very important for the robustness, speed and accuracy. Depending on the physics involved and the numerical parameters (e.g., discretization), we have observed that the transformation (12) allows a time step that is 10–100 times larger than without the transformation. This substantially improves the performance of the numerical model.

The modified equation (14) is solved with a six-stages fifth-order Runge–Kutta scheme with an embedded fourth-order scheme for the time-step selection [9]. (For the two-dimensional version of the code [6] we implemented an eight-order Runge–Kutta scheme.) We choose a Runge–Kutta scheme (RKS) for the following reasons:

- In the integration procedure, an RKS 'looks forward' to update the variables while a multi-step method (MSM) (e.g., Adam–Bashford) uses past information. This sounds more consistent for stiff problems.
- Although the computation of one time step requires more evaluations than an MSM, the step that is allowed is larger, with the consequence that the overall computation is reduced.
- The stability of RKS increases with the order while it decreases for MSM.
- RKS are auto-restart (computations can be stopped and restarted at any time without changing the result).
- In general, the MSM requires more storage than the RKS of the same order.

Of course, the relevance of these considerations are problem-dependent. They were found to be important for accurate simulations of fully nonlinear surface waves.

The embedded fourth-order scheme is used to estimate the size of the next time step. Hence, the scheme adapts itself to the step according to the physics involved. This is a very important feature because no unnecessary computations are carried out when they are not needed (infinitesimal waves). When rapid events occur, they are properly resolved, however (e.g., freak waves). Therefore, the scheme is more stable and accumulates fewer round-off errors than the corresponding constant-time-step procedure, often in a spectacular way.

It happens that the estimated step is too large for the tolerance required and must therefore be rejected. The time step is then reduced and the computations restarted. This is time consuming if it happens too often, and has consequences for the stability too. To limit the number of rejected steps and increase the stability of the procedure, we use a stabilization technique for stiff problems, the 'PI step size control' [4]. For stiff problems, it allows a reduced cost (by reducing the number of rejected steps) together with a stabilization of the overall procedure. This point is further illustrated through the examples in the next section.

The 'PI step size control' is implemented as follows: at the begining, the solution Z_{i-1} at $t = t_{i-1}$, $i > 1$ is known, and the user chooses then Δt_i and a parameter Tol. The algorithm consists in evaluating a next, optimal time step, Δt_{opt} by the following procedure:

```
(1) Compute $\tilde{Z}_i$ at $t = t_{i-1} + \Delta t_i$, from the fifth-order Runge–Kutta scheme.
(2) Compute $\hat{Z}_i$ obtained from the embedded fourth-order scheme.
(3) Compute the local relative error $\mathrm{err}_i = ||\hat{Z}_i - \tilde{Z}_i||/\mathrm{Tol}$.
(4) Compute $\Delta t_{\mathrm{opt}} = \Delta t_i \times (\mathrm{err}_i^{-\alpha} \mathrm{err}_{i-1}^{\beta})$
(5) IF ($\mathrm{err}_i < 1$ ) THEN
            $t_i = t_{i-1} + \Delta t_i$
            $Z_i = \tilde{Z}_i$
            $i = i + 1$
    ELSE
            $\Delta t_i = \Delta t_{\mathrm{opt}}$
    END
(6) GOTO (1)
```

In the above α and β are constants defining the 'PI step size control' method. If $\alpha = 1/5$ and $\beta = 0$, the standard step-size adaptive method is recovered. Here we use $\alpha = 0.7/5$, and $\beta = 0.4/5$. This choice was made in view of numerical tests; see [4].

3 Numerical examples

The efficiency of the present time integrator is illustrated with computations in two and three dimensions. In the first example, a wave packet, homogeneous in the transverse direction, is considered. Following [10], the initial condition is obtained from an exact Stokes wave, with wavenumber $\boldsymbol{k}_0 = (k_0, 0)$, half total wave height or amplitude a and wave period T_0. The surface elevation and the tangential velocity at the surface are then multiplied by the bell function $\mathrm{sech}[\epsilon\sqrt{2}ak_0^2(x_1 - x_{1,0})]$. Here we consider the case $\epsilon = 0.25$ and $ak_0 = 0.10$. (The case $\epsilon = 1$ corresponds to an envelope soliton that is a solution of the nonlinear Schrödinger equation.) The computational domain is 32 Stokes wavelengths in the propagation direction, and the discretization is 32 nodes per wavelength in both directions. Figure 1 displays the free-surface

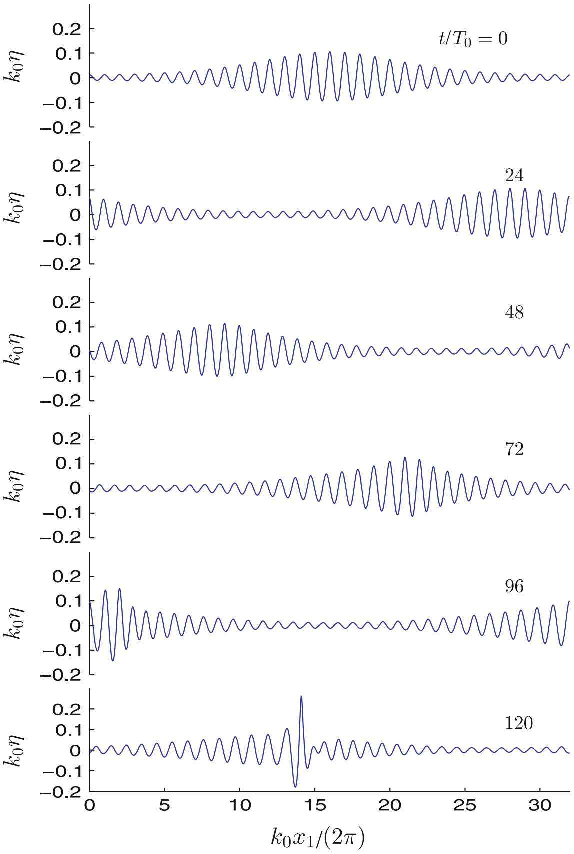

Fig. 1 Surface elevation at $t = \{0; 24; 48; 72; 96; 120\}T_0$. An initially long wave packet (upper plot) will split into a number of shorter wave packets, each with the form of an envelope soliton. Strong, local interaction takes place during the split-up process (lower plot)

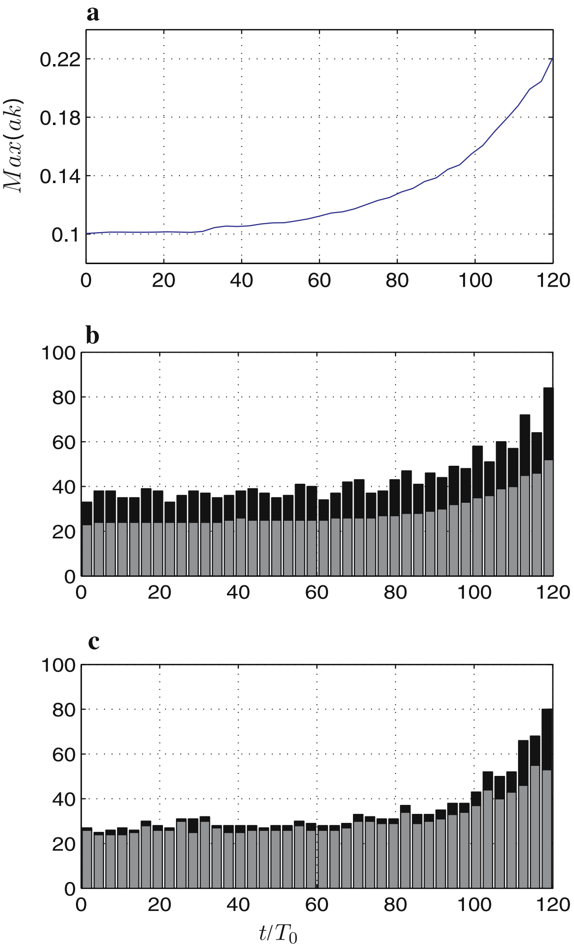

Fig. 2 **(a)** Maximum waveslope during evolution. **(b)** Time integration without and **(c)** with stabilizing 'PI step sized control'. The whole time series is divided into 40 sub-intervals. The number of accepted (gray) and rejected (black) steps are displayed for each interval

elevation at several times during the simulation. The surface becomes steeper and steeper as the time evolves, and a 'freak' wave event eventually occurs (at $t/T_0 \approx 120$). Figure 2a displays the local maximum steepness ak observed during the simulation. The total simulation time is further divided into 40 intervals of equal length. For each time interval the number of accepted and rejected steps is counted using the time integrator described in Sect. 2 with and without the 'PI step size control' stabilization technique. The comparison is presented in Fig. 2b, c.

The improvement needs no further comment. To obtain the same result with a constant-time-step procedure, one has to use the smallest time step of the overall simulation that leads to the maximum acceptable error. Since the stiffness of the problem increases with increasing waveslope, the maximum acceptable time step decreases. This leads to a constant number of steps of around a hundred per time interval in the present example.

Further we consider a purely three-dimensional example. Here a plane exact Stokes wave train with wavenumber $\boldsymbol{k}_0 = (k_0, 0)$ and waveslope $ak_0 = 0.3$ is perturbed by the most unstable class II mode $\boldsymbol{k}_{\text{pert}} = (1.5, 1.33)k_0$. The subsequent surface elevation is presented in Fig. 3 at several times of the

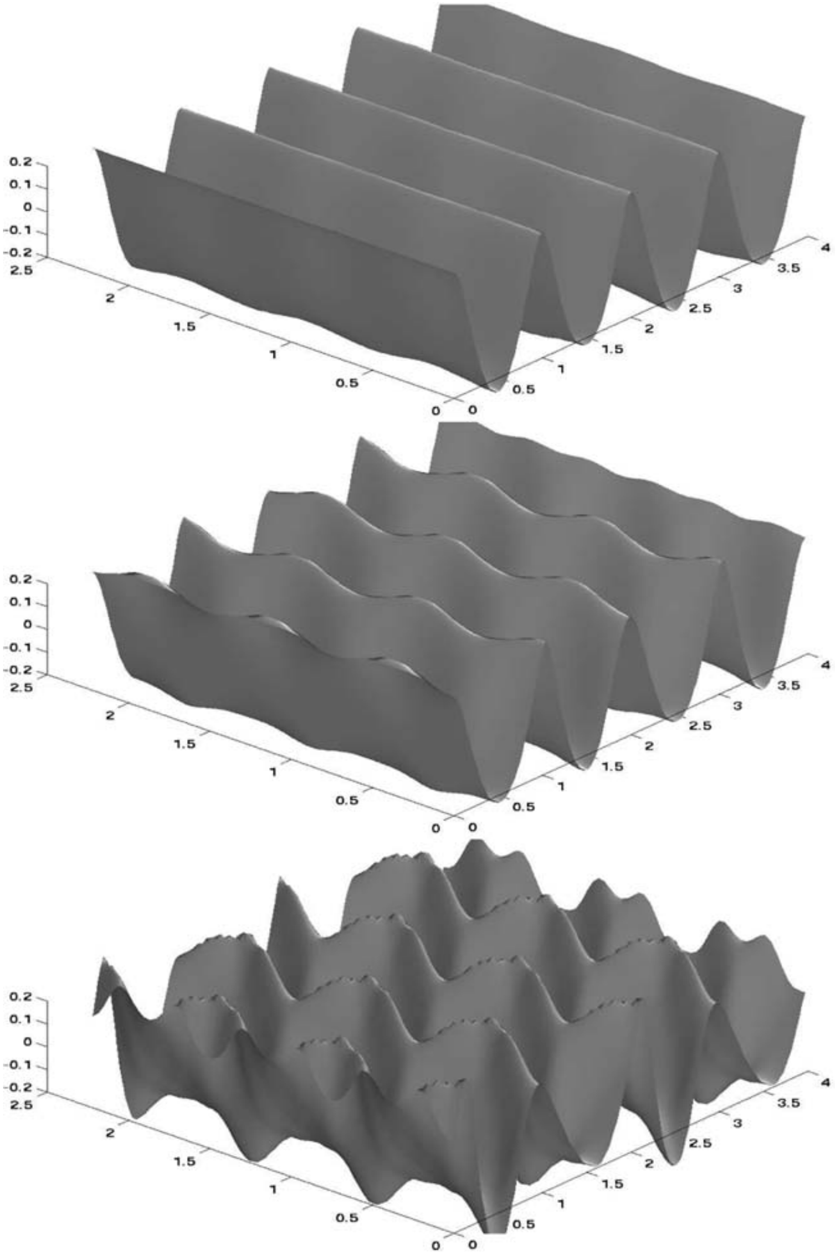

Fig. 3 Stokes wave train perturbed with class II instability. Free-surface elevation at $t/T_0 = 0, 9, 18$. Vertical coordinate: $k_0\eta$, horizontal coordinates: $k_0x_1/(2\pi)$, $k_0x_2/(2\pi)$

simulation. Horseshoe patterns eventually develop and lead to breaking of the waves. Using the stabilizing technique the efficiency significantly improves the performance of the time integrator, as is illustrated in Fig. 4.

4 Remarks on symplectic temporal integrators

The mechanical system of ocean-surface wave motion (1)–(2) can be recast in a Hamiltonian form [11]. This has the advantage that the Hamiltonian of the system is preserved, with the generalized coordinates

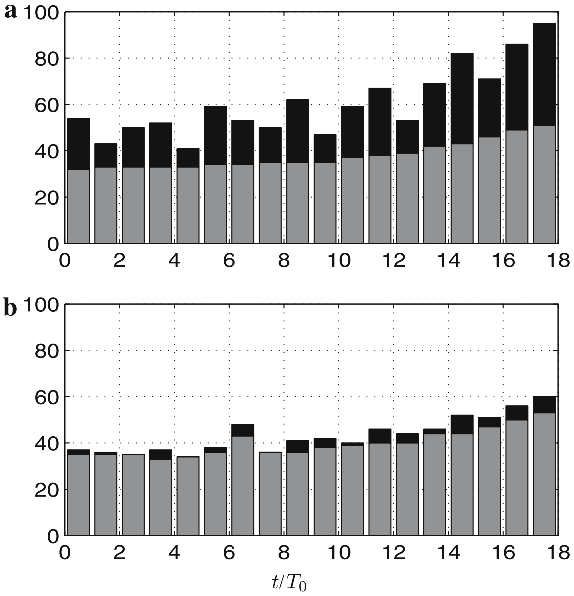

Fig. 4 **(a)** Time integration without and **(b)** with stabilizing 'PI step size control'. The whole time series is divided into 18 sub-intervals. The number of accepted (gray) and rejected (black) steps are displayed for each interval

evolving in a *symplectic* manner. The use of symplectic integrators [12] makes sense only if the spatial scheme does not involve smoothing or other artificial stabilization techniques. This is precisely the case for our scheme. We have spent much effort in trying to implement a time-stepping method that is symplectic. Based on intensive numerical experimentations (and not on mathematical investigations), we have observed that symplectic temporal integrators are inefficient for surface waves for the following reasons:

(i) General symplectic integrators are constant step schemes, i.e., symplecticity is destroyed if the time step is not constant. As we have mentioned and illustrated above, schemes with constant time steps are inefficient for large-scale simulations.

(ii) There are some recent investigations on symplectic schemes with variable steps [13]). However, these schemes are very involved and require more iterations than the corresponding scheme with constant step (see next point).

(iii) General symplecting integrators are implicit schemes and, therefore, their use requires iterations. It is unrealistic in practice to iterate until machine precision is reached, and the iterations are stopped when a given tolerance (say 10^{-8}) has been obtained. Thus, the scheme can only be approximately symplectic (except for a few peculiar equations for which *ad hoc* explicit symplectic schemes exist).

(iv) Simple functional iterations converge too slowly to be used in practice. Therefore, symplectic schemes often use Newton's method with an incomplete Jacobian to improve the convergence. This procedure induces non-polynomial nonlinearities and the aliasing is greatly increased, often leading to dramatic instabilities and numerical blow-up. Hence, more is lost on the spacial scheme than what is gained on the temporal one.

(v) Our implementations of the implicit Runge–Kutta schemes based on Gauss–Legendre quadratures, for example, where found to be very slow and highly unstable, often leading to numerical blow-up after three to five time steps only.

(vi) Runge–Kutta schemes, for example, are linear and exact for all polynomials up to a certain order. We tried the exponential integrator of Hochbruck et al. [14] and found it very slow and unstable. We have also tried to implement our own (nonlinear) implicit schemes based on rational (Padé) approximations. Though these schemes were designed to be A-stable (see [4] for the definitions of various stability definitions), they were found disastrously slow and unstable.

(vii) Finally, we note that by stability one should understand here ‘practical’ stability. This means that, if a scheme is theoretically stable for time steps smaller than, say, one-fiftieth of a characteristic period, it is considered practically unstable because such small steps are useless for realistic simulations, due to slowness and accumulation of round-off errors.

5 Conclusion

The importance of a suitable temporal integrator for simulations of water waves is discussed in this paper. If the waves are linear, the time integration can be performed in one time step, with the result evaluated to machine precision. When the waves are nonlinear, the linear part of the time stepping should be performed analytically, and the remaining, genuinely nonlinear and relatively smaller part of the prognostic equations, integrated in time by a method that assumes a variable time-step. Our experience is that constant-step methods are inefficient and one should *always* prefer an auto-adaptive method instead. Such schemes are easily implemented and provide very substantial improvements.

With regard to symplectic integrators, we conclude that these, in general, are not yet competitive in practice for the stiff problem of surface waves. A significant breakthrough in symplectic integration must take place before such a scheme can be used in realistic numerical wave tanks. For the time being, a well-chosen explicit scheme, including improvements — such as the integrating factor to remove part of the stiffness — is *much* more efficient in terms of speed, stability and accuracy. We have illustrated this claim in this paper, through the computation of instabilities of highly nonlinear water waves in three dimensions.

References

1. Dias F, Bridges TJ (2006) The numerical computation of freely propagating time-dependent irrotational water waves. Fluid Dyn Res 38(12):803–830
2. Fenton JD (1999) Numerical methods for nonlinear waves. Advances in Coastal and Ocean Engineering, vol 5, pp 241–324. World Scientific
3. Hairer E, Nørsett SP, Wanner G (1993) Solving Ordinary differential equations I. Nonstiff problems. Series in Comp Math 8, Springer-Verlag
4. Hairer E, Wanner G (1996) Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems. Series Comp Math, vol 14, Springer-Verlag
5. Grue J (2002) On four highly nonlinear phenomena in wave theory and marine hydrodynamics. Appl Ocean Res 24:261–274
6. Clamond D, Grue J (2001) A fast method for fully nonlinear water wave computations. J Fluid Mech 447:337–355
7. Clamond D, Fructus D, Grue J, Kristiansen Ø (2005) An efficient model for three-dimensional surface wave simulations. Part II: Generation and absorption. J Comp Phys 205:686–705
8. Fructus D, Clamond D, Grue J, Kristiansen Ø (2005) An efficient model for three-dimensional surface wave simulations. Part I: Free space problems. J Comp Phys 205:665–685
9. Dormand JR, Prince PJ (1980) A family of embedded Runge–Kutta formulae. J Comp Appl Math 6:19–26
10. Clamond D, Grue J (2002) Interaction between envelope solitons as a model for freak wave formation. Part I: Long time interaction. C R Mecanique 330:575–580
11. Zakharov VE (1968) Stability of periodic wave of finite amplitude on the surface of a deep fluid. J Appl Mech Phys Engl Transl 2:190–198
12. Hairer E, Lubich C, Wanner G (2002) Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations. Series Comp Math 31, Springer-Verlag
13. Hairer E, Söderling G (2005) Explicit, time reversible, adaptive step size control. SIAM J Sci Comp 26(6):1838–1851
14. Hochbruck M, Lubich C, Selhofer H (1998) Exponential integrators for large systems of differential equations. SIAM J Scient Comp 19(5):1552–1574