INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 12, 475–490 (1991)

STABILIZATION OF A TIME INTEGRATOR FOR THE 3D SHALLOW WATER EQUATIONS BY SMOOTHING TECHNIQUES

E. D. DE GOEDE

Centre for Mathematics and Computer Science, Amsterdam, The Netherlands

SUMMARY

A smoothing technique is applied to improve the stability of a semi-implicit time integrator for the three-dimensional shallow water equations. In this method the terms involving the vertical direction are treated implicitly. The stability condition on the time step depends only on the horizontal mesh sizes; therefore in the horizontal direction a smoothing operator is added. Owing to the smoothing, the maximally stable time step increases considerably while the accuracy is hardly affected. Moreover, it turns out that the smoothing operator is efficient on vector and parallel computers.

KEY WORDS 3D shallow water equations Method of lines Time integrators Smoothing Stability Vector and parallel computers

1. INTRODUCTION

In numerical analysis we distinguish explicit and implicit time integrators for partial differential equations. It is well known that implicit methods are in general stable for any time step but cannot exploit the facilities of vector and parallel computers as well as explicit methods do. On the other hand, explicit methods impose a severe restriction on the time step and therefore the time step is not dictated by accuracy considerations. To improve the stability of explicit methods we will use smoothing techniques.

Smoothing techniques are frequently applied in numerical methods. Usually the smoothing technique consists in applying a matrix $\mathbf{S}$ to some vector $\mathbf{F}$. The aim is to reduce the magnitude of the high frequencies occurring in the Fourier expansion of the vector to be smoothed without affecting the lower frequencies too much. A simple example of an $m \times m$ smoothing matrix $\mathbf{S}$ is given by $\mathbf{G}=\mathbf{SF}$, where

$$G_1 = F_1, \qquad G_i = \tfrac{1}{4}(F_{i-1} + 2F_i + F_{i+1}), \quad i = 2, \ldots, m-1, \qquad G_m = F_m, \tag{1}$$

with F_i and G_i denoting the components of the vectors $\mathbf{F}$ and $\mathbf{G}$ respectively.

In this paper our starting point is the semi-implicit time integrator that has been developed for the linearized three-dimensional shallow water equations (SWEs).[1] In this method only the vertical terms are treated implicitly. For this method we are faced with a CFL stability condition that depends on the horizontal mesh sizes Δx and Δy. For small values of Δx and Δy this time step restriction may be more severe than necessary for accuracy considerations. Therefore we will add a smoothing operator in the horizontal direction to make the stability condition due to the horizontal mesh sizes less restrictive.

0271–2091/91/050475–16$08.00

Received August 1989
Revised April 1990

The above-mentioned time integrator can be considered as a method in which an implicit smoothing operator already in the vertical direction. The smoothing in both horizontal and vertical directions may be interpreted as a preconditioning of the right-hand side of the semidiscrete shallow water equations. It will be shown that the maximally stable time step increases considerably when the smoothing operator in the horizontal direction is applied. The time step for the stabilized time integrator is now dictated by accuracy considerations, as it applies to implicit methods. Moreover, the stabilized time integrator can be implemented efficiently, as will be shown in the experiments. The efficiency of this method will be tested on various domains. In the experiments we will use a rectangular domain representing the North Sea and an irregular domain representing the IJsselmeer. The IJsselmeer is the largest lake in The Netherlands.

The technique of stabilizing explicit time integrators by right-hand-side smoothing has been applied by Wubs for the numerical solution of the two-dimensional shallow water equations.[2] An overview of various smoothing techniques has been presented by Van der Houwen.[3]

Section 2 provides the theory for the smoothing. In Section 3 we describe the semi-implicit time integrator for the shallow water equations. In Section 4 the smoothing is applied to stabilize this time integrator. Section 5 is devoted to the implementation of the smoothing matrices. Finally, in Section 6 we show by a number of experiments that applying smoothing operators leads to a considerable reduction of the computation time while the accuracy remains acceptable. The numerical solution was compared with a solution computed with a very small time step on the domain used in the experiments. This reference solution may therefore be considered as an almost exact solution on this domain. The reduction of the computation time is more or less independent of the domain. When the solution tends to a steady state we even obtain a reduction factor of about 10.

2. RIGHT-HAND-SIDE SMOOTHING

Consider the partial differential equation

$$\partial \mathbf{w}/\partial t = L\mathbf{w}(t, \mathbf{x}) + \mathbf{c}(t, \mathbf{x}), \tag{2}$$

where L is a *linear* differential operator with respect to the space variable $\mathbf{x}$ and $\mathbf{c}$ is a given function. This equation, together with its boundary conditions, can be semidiscretized into a system of ordinary differential equations (ODEs) of the form

$$\mathrm{d}\mathbf{W}/\mathrm{d}t = \mathbf{J}\mathbf{W}(t) + \mathbf{C}(t), \tag{3}$$

with $\mathbf{J}$ the Jacobian matrix, $\mathbf{C}$ an approximation to $\mathbf{c}$ and $\mathbf{W}$ an approximation to $\mathbf{w}$ at the grid points used for the semidiscretization. We shall always assume that this system is stable in the sense that the eigenvalues of $\mathbf{J}$ are in the non-positive half-plane. In Section 3 we shall see that the linearized 3D shallow water equations can be semidiscretized into this form.

If the system (3) is integrated by an *explicit* time integrator, then its maximally stable time step is limited owing to the usually extremely large magnitude of the spectral radius of $\mathbf{J}$. Therefore the time step has to be unrealistically small in order to achieve stability. This restriction is a drawback if the variation of the solution in time is so small that accuracy considerations would allow a larger time step. To obtain a better-conditioned right-hand-side function we premultiply the right-hand side of the original semidiscretization (3), or some part of it, by a *smoothing operator* $\mathbf{S}$. Thus we replace (3) either by

$$\mathrm{d}\mathbf{W}/\mathrm{d}t = \mathbf{S}[\mathbf{J}\mathbf{W}(t) + \mathbf{C}(t)] \tag{4a}$$

or by

$$\mathrm{d}\mathbf{W}/\mathrm{d}t = \mathbf{S}\mathbf{J}\mathbf{W}(t) + \mathbf{C}(t). \tag{4b}$$

In (4b) a part of the right-hand side is smoothed. The semidiscretization (4a) is particularly attractive in problems where it is known that the time derivative of the exact solution, i.e. $\partial \mathbf{w}/\partial t$, is a smooth function of the space variable $\mathbf{x}$ (e.g. in problems where a steady state is to be approximated). In such cases the right-hand-side function of the semidiscretization (3) is also a 'smooth' grid function, so that it may be premultiplied by the smoothing operator $\mathbf{S}$ without much loss of accuracy.

The maximally stable time step may increase considerably when the explicit time integrator is applied to (4) instead of to (3). To achieve that the condition of $\mathbf{SJ}$ is better than that of $\mathbf{J}$, the operator $\mathbf{S}$ should strongly damp the high frequencies (stiff components) in the Fourier expansion of the vector $\mathbf{JW}$, so that the spectral radius of $\mathbf{SJ}$ is substantially less than that of $\mathbf{J}$. One may consider the equations (4) as 'smoothed' or 'preconditioned' semidiscretizations of the original equation (2).

We emphasize that in this paper the right-hand-side function is smoothed instead of the grid function $\mathbf{W}(t)$ itself. The latter type of smoothing is often used. However, it may only be applied without considerable loss of accuracy if $\mathbf{W}(t)$ itself is a 'smooth' grid function for a fixed value of t. This is generally not the case. An example of this latter type of smoothing is the well-known Lax–Wendroff method.[4]

To characterize the effect of right-hand-side smoothing on the accuracy of the initial semidiscretization (3), we introduce the *order of consistency of smoothing operators*. Let Δ be the mesh size; then the smoothing operator $\mathbf{S}$ is said to be consistent of order p if $\mathbf{S}=\mathbf{I}+O(\Delta^p)$ as Δ tends to zero. Hence $\mathbf{S}$ converges to the identity operator $\mathbf{I}$ if the grid is refined.

We remark that the application of right-hand-side smoothing is not restricted to linear ODEs. Right-hand-side smoothing can also be applied to more general systems of the form

$$\mathrm{d}\mathbf{W}/\mathrm{d}t=\mathbf{F}(t, \mathbf{W}(t)) \tag{3'}$$

by replacing it by the smoothed system

$$\mathrm{d}\mathbf{W}/\mathrm{d}t=\mathbf{SF}(t, \mathbf{W}(t)). \tag{4'}$$

Summarizing, the smoothing operator $\mathbf{S}$ should satisfy the following requirements:

(A) $\mathbf{S}$ is consistent of order $p \geqslant 1$.
(B) The smoothed system is again stable.
(C) The spectral radius of $\mathbf{SJ}$ is considerably smaller than that of $\mathbf{J}$.
(D) The application of the operator $\mathbf{S}$ does not require much computational effort.

In the following subsections it will be shown that, instead of looking for highly stable integration methods, one may equally well apply methods in which the right-hand-side function of the system of ODEs (3) is premultiplied by a smoothing operator $\mathbf{S}$ such that the magnitude of the spectral radius associated with the right-hand-side function reduces considerably. We distinguish smoothing that is dependent on and smoothing that is largely independent of the right-hand-side function. The former type of smoothing is based on operator splitting and will be discussed in Section 2.1. Smoothing operators that are to a large degree independent of the right-hand-side function will be discussed in Section 2.2.

2.1. Smoothing operators based on operator splitting

Smoothing operators based on operator splitting are suggested by considering splitting methods developed for the time integration of partial differential equations. Our starting point is the forward Euler method applied to the semidiscretization (3), which can be described by

$$\mathbf{W}^{n+1}=\mathbf{W}^n+\tau(\mathbf{JW}^n+\mathbf{C}^n), \tag{5}$$

where τ is the time step and $\mathbf{W}^n$ and $\mathbf{C}^n$ denote approximations to $\mathbf{W}(n\tau)$ and $\mathbf{C}(n\tau)$ respectively. Let us split the matrix $\mathbf{J}$ into

$$\mathbf{J}=\mathbf{J}_1+\mathbf{J}_2$$

and let us replace the forward Euler method (5) by the splitting method

$$\mathbf{W}^{n+1}-\tau\mathbf{J}_2\mathbf{W}^{n+1}=\mathbf{W}^n+\tau(\mathbf{J}_1\mathbf{W}^n+\mathbf{C}^n)$$

or, equivalently,

$$\mathbf{W}^{n+1}=(\mathbf{I}-\tau\mathbf{J}_2)^{-1}[(\mathbf{I}+\tau\mathbf{J}_1)\mathbf{W}^n+\tau\mathbf{C}^n]. \tag{6}$$

This method can be rewritten as

$$\mathbf{W}^{n+1}=\mathbf{W}^n+\tau\mathbf{S}(\mathbf{J}\mathbf{W}^n+\mathbf{C}^n), \tag{7}$$

with

$$\mathbf{S}=(\mathbf{I}-\tau\mathbf{J}_2)^{-1}. \tag{8}$$

The splitting method (7)–(8) may be interpreted as the forward Euler method applied to the system of ODEs (4a), which is a 'smoothed' version of the initial semidiscretization (3), with a smoothing operator $\mathbf{S}$ defined by (8). By an appropriate choice of the matrix $\mathbf{J}_2$, this splitting method has much better stability characteristics than the forward Euler method (5). For example, the choices $\mathbf{J}_2=\mathbf{J}$ and $\mathbf{J}_2=\mathbf{J}/2$ lead to the A-stable methods of Laasonen (backward Euler) and Crank–Nicolson (trapezoidal rule) respectively. Another possibility is to choose $\mathbf{J}_2$ equal to a lower (or upper) triangular matrix. For the two-dimensional shallow water equations such an approach has been followed by Fischer[5] and Sielecki.[6] In fact, the method developed by De Goede[1] for the linearized shallow water equations may be interpreted as a combination of the Crank–Nicolson method and the approach of Sielecki and Fischer. In that paper it was shown that the stability of the resulting numerical method improves considerably whereas the computations can be performed efficiently.

2.2. Smoothing operators for general vector functions

The smoothing operators considered in the previous subsection depend strongly on the specific form of the right-hand-side function. In this subsection we summarize the main properties of the family of smoothing operators.[3,7] These operators are largely independent of the particular form of the vector function to which they are applied and therefore we shall present the results for the general equation (4′). We will again assume that the eigenvalues of the Jacobian matrix $\mathbf{J}:=\partial\mathbf{F}/\partial\mathbf{W}$ are in the non-positive half-plane.

The smoothing operator $\mathbf{S}$ will be chosen of the form $\mathbf{S}=P(\mathbf{D})$, where $\mathbf{D}$ is a difference matrix and the smoothing function $P(z)$ is a polynomial or a rational function yielding explicit or implicit smoothing operators, respectively. First we discuss the choice of the matrix $\mathbf{D}$. In our *theoretical* considerations we assume that $\mathbf{D}$ is equal to the Jacobian $\mathbf{J}$ normalized by its spectral radius, i.e.

$$\mathbf{D}=\mathbf{J}/\rho(\mathbf{J}). \tag{9}$$

We emphasize that in *practice* it is generally not attractive to choose $\mathbf{D}$ according to (9) and we shall employ some cheap approximation to the normalized Jacobian matrix. If $\mathbf{D}$ is defined according to (9) then the eigenvalues of $\mathbf{SJ}=P(\mathbf{D})\mathbf{J}$ are given by $\rho(\mathbf{J})zP(z)$, where z runs through the spectrum of $\mathbf{D}$.

2.2.1. Explicit smoothing operators. In the case of explicit smoothing we are looking for a polynomial $P(z)$ such that the magnitude of $zP(z)$ is sufficiently small with $P(0)=1$ and z either in $[-1,0]$ or in $[-\mathrm{i},\mathrm{i}]$. Moreover, the polynomial $P(z)$ will be chosen such that $zP(z)$ remains in the non-positive half-plane. It was shown[3,7] that polynomials of the form

$$P(z)=\frac{U_{2k}[\sqrt{(1+z^2)}]}{2k+1}, \qquad U_{2k}(x):=\frac{\sin[(2k+1)\cos^{-1}(x)]}{\sin[\cos^{-1}(x)]}, \tag{10}$$

minimize the magnitude of $zP(z)$ on the purely imaginary interval $[-\mathrm{i},\mathrm{i}]$. However, if z has negative real parts then it may happen that $\mathrm{Re}\{zP(z)\}>0$, causing unstable behaviour. Since we shall apply smoothing to vector functions whose Jacobian matrices possess eigenvalues with small negative real parts (caused by the vertical diffusion and the bottom friction in the SWEs), we require that $\mathrm{Re}(zP(z))\leqslant 0$ for all values with $\mathrm{Re}\{z\}\leqslant 0$ (see condition (B)). For this case the following theorem defines a family of nearly optimum polynomials.[3,7]

Theorem 1

Let $\mathbf{D}$ be defined by (9) and let $\mathbf{S}=P(\mathbf{D})$ with $P(z)$ defined by

$$P(z)=\frac{T_k(1+2z^2)-1}{2k^2z^2}, \qquad T_k(x)=\cos[k\cos^{-1}(x)]. \tag{11}$$

Then the following assertions hold:

(a) If $\mathrm{Re}\{z\}\leqslant 0$ then $\mathrm{Re}\{zP(z)\}\leqslant 0$.
(b) If z is purely imaginary then $zP(z)$ is again purely imaginary and for sufficiently large k its maximum is approximately $2/\pi k$. (12)

Proof. For a proof of (a) we refer to References 3 and 7.
(b) We have to find the maximum of $|zP(z)|$ on $[-\mathrm{i},\mathrm{i}]$ or, equivalently,

$$\max\left|[T_k(1+2z^2)-1]\frac{1}{2k^2z}\right|. \tag{13}$$

The range of $1+2z^2$ in (13) is $[-1,1]$. On this interval the Chebyshev polynomial $T_k(1+2z^2)$ satisfies the 'so-called' equal ripple property,[8] which means that it alternatingly assumes equal maximum and minimum values. Because of the factor $1/2k^2z$, let us now assume that the value in (13) can be approximated at the smallest value of $|z|$ for which $T_k(1+2z^2)$ reaches its minimum. Thus we require that

$$T_k(1+2z^2)=\cos[k\cos^{-1}(1+2z^2)]=-1$$

for $|z|$ as small as possible, which yields

$$z=\pm\mathrm{i}\frac{\sqrt{[1-\cos(\pi/k)]}}{\sqrt{2}}.$$

For these values of z we obtain that (13) is bounded by

$$\frac{\sqrt{2}}{k^2\sqrt{[1-\cos(\pi/k)]}}\approx\frac{2}{\pi k} \tag{14}$$

for k sufficiently large. For many values of k we verified numerically that the reduction factor is close to $2/\pi k$. Therefore we conclude that the approximation applied in this theorem is justified. □

An extremely efficient implementation of the smoothing operator of Theorem 1 can be obtained by using the following factorization theorem (see also Section 5), which justifies the application of these smoothing operators.

Theorem 2

Let the matrix $\mathbf{D}$ be defined by (9), let $\mathbf{S}=P(\mathbf{D})$ with $P(z)$ defined by (11), let the factor matrices $\mathbf{F}_j$ be generated by

$$\mathbf{F}_1=\mathbf{I}+\mathbf{D}^2, \qquad \mathbf{F}_{j+1}=(\mathbf{I}-2\mathbf{F}_j)^2, \quad j>0,$$

and let $k=2^q$. Then $\mathbf{S}$ can be factorized by

$$\mathbf{S}=\mathbf{F}_q\mathbf{F}_{q-1}\cdots\mathbf{F}_1. \tag{15}$$

Proof. For a proof of Theorem 2 we refer to References 3 and 7. □

2.2.2. Implicit smoothing operators. In this subsection we will discuss implicit smoothing, i.e. if $\mathbf{F}$ is the vector to be smoothed then the smoothed vector $\mathbf{G}$ is obtained by solving $\mathbf{G}=\mathbf{S}^{-1}\mathbf{F}$, where $\mathbf{S}^{-1}$ is a smoothing operator (see (1)). Implicit smoothing has been applied in References 2 and 9.

Theorem 3

Let $\mathbf{S}^{-1}=P(\mathbf{D})$ with $\mathbf{D}$ a difference matrix and $P(z)$ defined by

$$P(z)=\frac{1}{1-\frac{1}{4}\alpha^2 z^2}, \quad \alpha>0. \tag{16}$$

Then the following assertion holds: if z is purely imaginary then

$$|zP(z)|\leqslant 1/\alpha.$$

Proof. This follows immediately from elementary analysis. □

As mentioned before, in practice we shall choose $\mathbf{D}$ equal to some cheap approximation of the normalized Jacobian which satisfies condition (B). In choosing a difference matrix $\mathbf{D}$ the boundary conditions have to be incorporated in $\mathbf{D}$. This is important to preserve conservation of mass. In this paper we shall choose smoothing operators of the form

$$\mathbf{D}^2=\frac{1}{4}\begin{pmatrix} 0 & & & & & & 0 \\ 1 & -2 & 1 & & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & & 1 & -2 & 1 \\ 0 & & & & & & 0 \end{pmatrix}. \tag{17}$$

The implicit smoothing operator described in Theorem 3 with $\mathbf{D}^2$ as in (17) results in the solution of a tridiagonal system. Therefore this implicit smoothing operator does not require much computational effort. In practice the value of α in (16) depends on the time step and on the mesh sizes.

Let us now discuss the order of consistency of the smoothing operator $\mathbf{S}$ with $\mathbf{D}^2$ defined in (17). We assume that $\mathbf{D}^2$ and $P(z)$ satisfy the conditions

$$\mathbf{D}^2=O(\Delta^s) \quad \text{as } \Delta\to 0, \qquad P(z)=1+O(z^{2r}) \quad \text{as } z\to 0, \tag{18}$$

where Δ denotes the mesh size and r and s are positive integers. Hence $\mathbf{S}$ is consistent of order $p=rs$. For example, the smoothing matrix defined by (2) can be generated by $P(z)=1+z^2$ with $\mathbf{D}^2$ defined by (17) and is second-order-consistent ($s=2$, $r=1$). When $P(z)$ is defined by (11) and $\mathbf{D}^2$ by (17), it can be easily verified that $\mathbf{S}$ is also second-order-consistent.

Summarizing, if we choose the matrix $\mathbf{D}^2$ defined by (17) then the smoothing matrix $\mathbf{S}=P(\mathbf{D})$, with $P(z)$ the polynomial (11) or the rational function (16), reduces the magnitude of the spectrum associated with the right-hand-side function considerably whereas the spectrum remains in the non-positive half-plane. For this choice of $\mathbf{D}^2$ the smoothing matrix $\mathbf{S}$ is independent of the right-hand-side function.

In Section 4 we shall use both smoothing based on operator splitting and smoothing of general vector functions based on the theorems in Section 2.2.

3. MATHEMATICAL MODEL

In this section we will describe the mathematical model and the time integrator to which the smoothing will be applied. The following symbols are used:

A^σ	vertical diffusion coefficient
C	Chezy coefficient
f	Coriolis term
F_b	bottom stress in x-direction
F_s	surface stress in x-direction
g	acceleration due to gravity
G_b	bottom stress in y-direction
G_s	surface stress in y-direction
H	total depth ($=h+\zeta$)
h	undisturbed depth of water
t	time
u, v	velocity components in x- and y-direction
x, y, σ	a left-handed set of co-ordinates
W_f	wind stress
ρ	density
ζ	elevation above undisturbed depth
φ	angle between wind direction and positive x-axis.

We will use a three-dimensional model in sigma co-ordinates in which the advective terms have been omitted. This model is described by

$$\frac{\partial u}{\partial t}=fv-g\frac{\partial \zeta}{\partial x}+\frac{1}{\rho}\frac{1}{H^2}\frac{\partial(A^\sigma \partial u/\partial\sigma)}{\partial\sigma}, \tag{19}$$

$$\frac{\partial v}{\partial t}=-fu-g\frac{\partial \zeta}{\partial y}+\frac{1}{\rho}\frac{1}{H^2}\frac{\partial(A^\sigma \partial v/\partial\sigma)}{\partial\sigma}, \tag{20}$$

$$\frac{\partial \zeta}{\partial t}=-\frac{\partial}{\partial x}\left(H\int_0^1 u\,\mathrm{d}\sigma\right)-\frac{\partial}{\partial y}\left(H\int_0^1 v\,\mathrm{d}\sigma\right), \tag{21}$$

with boundaries

$$0\leqslant x\leqslant L, \qquad 0\leqslant y\leqslant B, \qquad 1\geqslant\sigma\geqslant 0.$$

Thus the domain is a rectangular basin. Owing to the sigma transformation in the vertical, the

domain is constant in time. We have the closed boundary conditions

$$u(0, y, \sigma, t)=0, \qquad u(L, y, \sigma, t)=0, \qquad v(x, 0, \sigma, t)=0, \qquad v(x, B, \sigma, t)=0.$$

The boundary conditions at the sea surface ($\sigma=0$) are given by

$$-\left(A^{\sigma}\frac{\partial u}{\partial \sigma}\right)_0 = HF_s, \qquad -\left(A^{\sigma}\frac{\partial v}{\partial \sigma}\right)_0 = HG_s,$$

and at the bottom ($\sigma=1$) by

$$-\left(A^{\sigma}\frac{\partial u}{\partial \sigma}\right)_1 = HF_b, \qquad -\left(A^{\sigma}\frac{\partial v}{\partial \sigma}\right)_1 = HG_b.$$

The bottom stress is parametrized using a linear law of bottom friction, which is of the form

$$F_b = g\rho u_d/C^2, \qquad G_b = g\rho v_d/C^2,$$

with u_d and v_d the components of the current at some depth near the bottom. The surface stresses are expressed as

$$F_s = W_f \cos\varphi, \qquad G_s = W_f \sin\varphi.$$

3.1. Space discretization

For the space discretization of the equations (19)–(21) the computational domain is covered by an $nx \times ny \times ns$ rectangular staggered grid.[1] For the approximation of the spatial derivatives, second-order central finite differences are used in both the horizontal and vertical directions. We use the following notation: **U**, **V** and **Z** are grid functions approximating u, v and ζ respectively. The **Z**-points are only specified at the sea surface. Furthermore: $\mathbf{\Lambda}_{\sigma\sigma}$ is a tridiagonal matrix approximating the vertical diffusion term; $\mathbf{\Theta}_1$ is an $(nx \times ny \times ns) \times (nx \times ny)$ matrix (a row of ns diagonal matrices of order $(nx \times ny)^2$ with $\Delta\sigma_k$ on the diagonal of the kth submatrix); $\mathbf{\Theta}_2$ is an $(nx \times ny) \times (nx \times ny \times ns)$ matrix (a column of ns identity matrices of order $(nx \times ny)^2$; **F** is a four-diagonal matrix (due to the grid staggering) of order $(nx \times ny \times ns)^2$, approximating the Coriolis term; $\mathbf{D}_x$ and $\mathbf{D}_y$ are bidiagonal matrices (one diagonal and one lower diagonal) of order $(nx \times ny)^2$, approximating the differential operators $\partial/\partial x$ and $\partial/\partial y$ respectively; $\mathbf{E}_x$ and $\mathbf{E}_y$ are bidiagonal matrices (one diagonal and one upper diagonal) with $\mathbf{E}_x = -\mathbf{D}_x^{\mathrm{T}}$ and $\mathbf{E}_y = -\mathbf{D}_y^{\mathrm{T}}$. The matrices $\mathbf{D}_x$ and $\mathbf{E}_x$ differ because of the grid staggering.

Now the semidiscretized system can be written in the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{W} = \mathbf{F}(\mathbf{W}) = (\mathbf{A}+\mathbf{B})\mathbf{W} + \mathbf{C}, \tag{22}$$

with

$$\mathbf{W}=\begin{pmatrix}\mathbf{U}\\ \mathbf{V}\\ \mathbf{Z}\end{pmatrix}, \quad \mathbf{A}=\begin{pmatrix}\mathbf{\Lambda}_{\sigma\sigma} & 0 & 0\\ -\mathbf{F} & \mathbf{\Lambda}_{\sigma\sigma} & 0\\ -\mathbf{\Theta}_1 H\mathbf{E}_x & -\mathbf{\Theta}_1 H\mathbf{D}_y & 0\end{pmatrix}, \quad \mathbf{B}=\begin{pmatrix}0 & \mathbf{F} & -\mathbf{\Theta}_2 g\mathbf{D}_x\\ 0 & 0 & -\mathbf{\Theta}_2 g\mathbf{E}_y\\ 0 & 0 & 0\end{pmatrix}, \quad \mathbf{C}=\begin{pmatrix}\mathbf{F}_u\\ \mathbf{F}_v\\ 0\end{pmatrix}. \tag{23}$$

The reason for this splitting will become clear in the next sections. The vector **C** contains the components of the wind stress. Note that the integrals in (21) are approximated by $\mathbf{\Theta}_1\mathbf{U}$ and $\mathbf{\Theta}_1\mathbf{V}$ respectively.

3.2. Time integration

We start with the following time integrator for (22)–(23):[3]

$$\begin{pmatrix} \mathbf{I}-\tau\boldsymbol{\Lambda}_{\sigma\sigma} & 0 & 0 \\ \tau\mathbf{F} & \mathbf{I}-\tau\boldsymbol{\Lambda}_{\sigma\sigma} & 0 \\ \tau\boldsymbol{\Theta}_1 H\mathbf{E}_x & \tau\boldsymbol{\Theta}_1 H\mathbf{D}_y & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{U}^{n+1} \\ \mathbf{V}^{n+1} \\ \mathbf{Z}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \tau\mathbf{F} & -\tau\boldsymbol{\Theta}_2 g\mathbf{D}_x \\ 0 & \mathbf{I} & -\tau\boldsymbol{\Theta}_2 g\mathbf{E}_y \\ 0 & 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{U}^n \\ \mathbf{V}^n \\ \mathbf{Z}^n \end{pmatrix} + \tau \begin{pmatrix} \mathbf{F}_u^n \\ \mathbf{F}_v^n \\ 0 \end{pmatrix}$$

or, equivalently,

$$(\mathbf{I}-\tau\mathbf{A})\mathbf{W}^{n+1} = (\mathbf{I}+\tau\mathbf{B})\mathbf{W}^n + \tau\mathbf{C}^n.$$

This method can be written in the form

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \tau(\mathbf{I}-\tau\mathbf{A})^{-1}\mathbf{F}(\mathbf{W}^n). \tag{24}$$

In terms of (7)–(8) we have that $\mathbf{S} = (\mathbf{I}-\tau\mathbf{A})^{-1}$. Thus this time integrator can be considered as a method in which the right-hand-side function is preconditioned by the implicit smoothing operator $(\mathbf{I}-\tau\mathbf{A})^{-1}$. It can easily be seen that the components are calculated sequentially (first **U**, then **V** and finally **Z**). This is advantageous for both the stability and storage requirements. For the two-dimensional shallow water equations a similar approach has been followed by e.g. Fischer[5] and Sielecki.[6] The time step restriction for method (24) is given by[1]

$$\tau < \frac{1}{\sqrt{(gH)}} \frac{1}{\sqrt{[1/(\Delta x)^2 + 1/(\Delta y)^2]}} \sqrt{\left(1 + \frac{\tau}{2(\Delta\sigma)^2} \frac{A^\sigma}{\rho H^2}\right)}, \tag{25}$$

where Δx, Δy and $\Delta\sigma$ denote the mesh sizes. We remark that the time step in (25) hardly depends on the vertical mesh size $\Delta\sigma$. However, the condition imposed by the horizontal mesh sizes is still rather restrictive. Therefore we will add a smoothing operator in the horizontal direction. This smoothing operator will be described in the next section.

4. SMOOTHING

In this section the stability of method (24) will be improved by a smoothing of general vector functions (see Section 2.2).

First we consider method (24). This method can be written in the form

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \tau[\mathbf{I}-\tau(\mathbf{A}_1+\mathbf{A}_2)]^{-1}\mathbf{F}(\mathbf{W}^n), \tag{26}$$

where $\mathbf{A}_1 + \mathbf{A}_2 = \mathbf{A}$ (see (23)), with

$$\mathbf{A}_1 = \begin{pmatrix} \boldsymbol{\Lambda}_{\sigma\sigma} & 0 & 0 \\ 0 & \boldsymbol{\Lambda}_{\sigma\sigma} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 0 \\ -\mathbf{F} & 0 & 0 \\ -\boldsymbol{\Theta}_1 H\mathbf{E}_x & -\boldsymbol{\Theta}_1 H\mathbf{D}_y & 0 \end{pmatrix}.$$

Method (26) may therefore be interpreted as the forward Euler method in which the right-hand-side function has been smoothed by the matrix $[\mathbf{I}-\tau(\mathbf{A}_1+\mathbf{A}_2)]^{-1}$. The vertical terms are treated implicitly because the matrix $\mathbf{A}_1$ contains the discretization of the vertical diffusion term. The stability condition for this time integrator hardly depends on the vertical mesh size $\Delta\sigma$. However, the condition imposed by the horizontal mesh sizes is still rather restrictive (see (25)). The horizontal terms are treated partly implicitly ($\mathbf{A}_2\mathbf{W}^{n+1}$ in (26)) and partly explicitly ($\mathbf{B}\mathbf{W}^n$ in (24)). Hence we add another preconditioning of the right-hand-side function, i.e. a smoothing of general vector functions described in Section 2.2.

The right-hand-side function of the U-component only contains derivatives in the x-direction and will therefore be smoothed in the x-direction only. Similarly, the V-component is only smoothed in the y-direction. The Z-component is smoothed in both directions. However, the smoothing of the right-hand-side function in two directions is complicated. The precomputation of the *cheap* factor matrices (see Theorem 2) is only feasible in one-dimensional cases. Therefore we apply one-dimensional smoothing in the x- and y-direction successively.

In the x-direction the smoothing matrix has the simple structure

$$\begin{pmatrix} \mathbf{S}_u & & \\ & 0 & \\ & & \mathbf{S}_z \end{pmatrix},$$

where $\mathbf{S}_u$ and $\mathbf{S}_z$ denote the smoothing matrices for the right-hand-side function of the U- and Z-component, respectively. Here $\mathbf{S}_u = P(\mathbf{D}_u)$ and $\mathbf{S}_z = P(\mathbf{D}_z)$ with $P(z)$ defined by (11) and

$$\mathbf{D}_u^2 = \frac{1}{4}\begin{pmatrix} 0 & & & & & & 0 \\ 1 & -2 & 1 & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & 1 & -2 & 1 & \\ 0 & & & & & & 0 \end{pmatrix}, \qquad \mathbf{D}_z^2 = \frac{1}{4}\begin{pmatrix} -1 & 1 & & & & & 0 \\ 1 & -2 & 1 & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & 1 & -2 & 1 & \\ 0 & & & & & -1 & 1 \end{pmatrix}. \tag{27}$$

In the y-direction the smoothing matrix has a similar simple structure. Note that $\mathbf{D}_u$ and $\mathbf{D}_z$ only differ in the first and last row, which is due to the grid staggering and to the boundary conditions. The number of different boundary conditions is very limited (open or closed boundaries, u- or ζ-boundaries). The smoothing matrices, including the values in the first and last row, are therefore computed in advance.

The time integration method in which the smoothing based on general vector functions has been added can be written in the form

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \tau[\mathbf{I} - \tau(\mathbf{A}_1 + \mathbf{S}\mathbf{A}_2)]^{-1}\mathbf{S}\mathbf{F}(\mathbf{W}^n), \tag{28}$$

with the matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ defined in (26) and the smoothing operator $\mathbf{S}$ defined in Theorem 1. The smoothing operator $\mathbf{S}$ appears twice in (28). The first operator $\mathbf{S}$ is a result of the fact that the components of $\mathbf{W}$ are computed sequentially. The second operator $\mathbf{S}$ in (28) is clearly a smoothing of the right-hand-side function. In cases where the solution becomes stationary (thus $\mathbf{F}(\mathbf{W}) = \mathbf{0}$) it is evident that methods (26) and (28) obtain the same stationary solution.

The stability condition for method (28) reads (see (14) and (25))

$$\tau < \frac{\pi k}{2} \frac{1}{\sqrt{(gH)}} \frac{1}{\sqrt{[1/(\Delta x)^2 + 1/(\Delta y)^2]}} \sqrt{\left(1 + \frac{\tau}{2(\Delta\sigma)^2} \frac{A^\sigma}{\rho H^2}\right)}. \tag{29}$$

Hence the gain factor obtained by the smoothing of general vector functions is $\pi k/2$.

5. IMPLEMENTATION OF THE SMOOTHING MATRICES

In this section we discuss the implementation of the smoothing matrices $(\mathbf{I} - \tau\mathbf{A})^{-1}$ and $\mathbf{S}$ (see (28)). For the U- and V-component the smoothing matrix $(\mathbf{I} - \tau\mathbf{A})^{-1}$ requires the solution of $nx \times ny$ tridiagonal systems of order ns, which can be computed efficiently.[1] The smoothing operator $\mathbf{S}$ can be computed in various ways. The most efficient implementation is based on the

factorization property presented in Theorem 2. If the factor matrices of (15) are computed in advance, then the evaluation of $P(\mathbf{D})$ only requires q $(=2\log(k))$ matrix–vector operations.

For example, applying Theorem 2 for the matrix $\mathbf{D}_u^2$ (see (27)) we find the factor matrices

$$\mathbf{F}_1=\frac{1}{4}\begin{pmatrix} 4 & & & & & & 0 \\ 1 & 2 & 1 & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & 1 & 2 & 1 & \\ 0 & & & & & & 4 \end{pmatrix}, \qquad \mathbf{F}_2=\frac{1}{4}\begin{pmatrix} 4 & & & & & & 0 \\ 2 & 1 & 0 & 1 & & & \\ 1 & 0 & 2 & 0 & 1 & & \\ & 1 & 0 & 2 & 0 & 1 & \\ & & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \qquad \text{etc.}$$

Evidently, the matrix–vector multiplications with these essentially three-diagonal factor matrices are extremely cheap, especially on vector computers. For example, on the CDC CYBER 205 the operations can be performed in two linked triad instructions (except near the boundaries).

Since the smoothing operator $\mathbf{S}$ is applied in the x- and y-direction successively, it consists of a sequence of one-dimensional operators. Therefore the smoothing operator can be implemented on irregular domains too. However, the bandwidth of the factor matrices $\mathbf{F}_q$ is 2^q+1. In practice, the value of q is at most five. In experiments with irregular domains it might happen that there are not enough grid points in the x- or y-direction. In this case we apply the implicit smoothing operator defined in Theorem 3, with $\mathbf{D}^2$ as in (27), instead of the explicit smoothing operator. Thus the application of the smoothing operator is hardly complicated when the domain is irregular.

The implicit smoothing operator requires the solution of a small tridiagonal system with a dimension of at most 2^q. For the solution of the tridiagonal systems we use the Gaussian elimination method. Since these systems are small and the implicit smoothing is only applied in narrow regions (where the number of grid points is less than or equal to 2^q), the computation time for the sequential Gaussian elimination method is very limited also on vector and parallel computers.

6. NUMERICAL EXPERIMENTS

In this section we show for a number of test problems[1, 10] the effects of smoothing on the stability and on the accuracy. In the test problems the water is initially at rest and the motion in the basin is generated by a wind stress. Thus a wind-driven circulation is gradually developed. We carry out two experiments with a constant wind stress and one with a time-dependent wind stress. In the experiments with a constant wind stress we use a rectangular basin with dimensions representative of the North Sea and an irregular basin representing the IJsselmeer. The IJsselmeer is the largest lake in The Netherlands.

The following parameter values are used in all experiments:

$f=0{\cdot}44/3600$ $(=1{\cdot}22\times10^{-4})$

$g=9{\cdot}81\ \mathrm{m\,s^{-2}}$

$A^{\sigma}=0{\cdot}065/\rho$

$\rho=1025\ \mathrm{kg\,m^{-3}}$

$\varphi=45^{\circ}$ (north-eastern wind).

For the time integration we use method (28). In the experiments we vary the number of smoothing

factors in the factorized smoothing operator (see (15)) to investigate the effects of smoothing on the stability and on the accuracy. The experiments have been carried out on an ALLIANT FX/4. This mini-supercomputer consists of four vector processors. On such a computer we can investigate the effect of the smoothing on both vector–parallel computers and scalar computers.

To represent the results we use the following notation:

q	number of smoothing factors
ERROR-ζ	maximal global error for the water elevation compared with a reference solution measured at the endpoint $t=T$
$\mathrm{TOT}_{\mathrm{VP/S}}$	total computation time
$\mathrm{SMO}_{\mathrm{VP/S}}$	computation time for the smoothing operator.

The indices VP and S indicate vector–parallel optimization and scalar optimization respectively. Thus the experiment is carried out on one processor if only the scalar optimization is used. At the end of the integration process the numerical solution for the ζ-component was compared with a reference solution computed on the same grid with $\tau=30$ s. The reference solution may be considered as an almost exact solution of our semidiscretized system (22). Thus the accuracy results listed in this section represent the error due to the time integration. We experimentally determined the maximally stable time step for each value of q. These time steps are in agreement with (29) (see Table II).

In the first two experiments we used a rectangular basin of 400 by 800 km with $\Delta x=10$ km, $\Delta y=10$ km, $\Delta\sigma=0{\cdot}25$ and $h=65$ m. Thus the computations have been performed on a grid with $nx=41$, $ny=81$ and $ns=4$.

In the first experiment we integrated over a period of five days with the constant wind stress

$$W_{\mathrm{f}}=1{\cdot}5\ \mathrm{kg\,m\,s^{-2}}. \tag{30}$$

At that time the steady state has already been reached.

In this experiment the maximal value for the water elevation is about 1·07 m. The results show that the time integration can be performed with much larger time steps when the smoothing technique is applied. In this experiment, in which the solution becomes stationary, the accuracy is hardly reduced by the smoothing procedure. Only for large q do some errors occur. This is due to the fact that for these values of q the steady state has not yet been reached. If the time integration is performed over a longer period we obtain the same results for large values of q as for the case $q=0$. This is in agreement with the theory that a stationary solution should be independent of the number of smoothing factors (see Section 4).

In Table II we list the gain factors of the maximally stable time steps compared with the case $q=0$ ($\tau_{\max}\approx 277$ s) and we compare them with the theoretical gain factors. Moreover, we list the gain factors in computation times.

Table I. Test problem with a constant wind stress

q	τ (s)	ERROR-ζ (m)	$\mathrm{TOT}_{\mathrm{VP}}$ (s)	$\mathrm{SMO}_{\mathrm{VP}}$ (s)	$\mathrm{TOT}_{\mathrm{S}}$ (s)	$\mathrm{SMO}_{\mathrm{S}}$ (s)
0	270	0·001	345·0	0·0	2600·1	0·0
1	800	0·002	173·1	20·5	1463·8	243·1
2	1800	0·008	85·6	17·6	767·3	219·7
3	3600	0·022	46·9	12·8	446·3	169·6
4	7200	0·055	25·9	8·7	255·8	116·8
5	14400	0·152	13·7	5·1	142·8	75·0

The theoretical gain factor $2^{q-1}\pi$ (see (29) with $k=2^q$) is in agreement with the experimental results. The results show a significant reduction in computation time, especially when the vector–parallel optimization is used. The overhead due to the smoothing operator is less than a factor of two even for large values of q. In the case of the vector–parallel optimization the computation time is reduced by about a factor of three due to the vectorization and also by a factor of three due to the parallel optimization.

It is interesting to investigate the effect of smoothing when the solution of a test problem does not become stationary. Therefore in the second experiment we introduce a time-dependent wind stress (see (30))

$$W_f = 1{\cdot}5\left[1 + 0{\cdot}5\sin\left(\frac{2\pi t}{24\times 3600}\right)\right]. \tag{31}$$

Now we have a periodically varying (north-eastern) wind with a period of 24 h. We integrated over a period of five days. At that time the solution is almost periodic. In the case without smoothing in the horizontal we obtained the following maximal water elevations at the south-west corner of the basin:

$$\zeta = 2{\cdot}106 \text{ m} \quad \text{at } t = 7{\cdot}3 + iP \text{ h},$$

with period $P=24$ h and i a positive integer. When smoothing is applied we observe that about the same maximal and minimal water elevations are reached as in the case without smoothing in the horizontal. It seems that the smoothing operator hardly introduces a dissipation error. However, some errors in the phase of the periodic solution appear. In Table III we list the maximal global error in the numerical solution for the water elevation measured at the endpoint $T=120$ h compared with a reference solution computed with $\tau=30$ s.

The results show that the error due to the smoothing operator is even smaller than the error due to the larger time steps. For example, in the case $q=2$ the error due to the larger time steps (i.e. 0·029 m) is larger than the error due to the smoothing (i.e. $\leqslant$0·023 m). Thus, if a fully implicit method had been used, the accuracy would also decrease for large time steps.

Table II. Gain factors

	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$
Theoretically ($=2^{q-1}\pi$ for $q>0$)	3·1	6·3	12·6	25·1	50·3
Experimentally (see Table I)	2·9	6·5	13·0	25·9	51·9
In computation time (VP)	2·0	4·0	7·4	13·3	25·2
In computation time (S)	1·8	3·4	5·8	10·2	18·2

Table III. Test problem with a time-dependent wind stress

q	τ (s)	ERROR-ζ (m)	τ (s)	ERROR-ζ (m)
0			270	0·008
1	270	0·008	720	0·014
2	270	0·023	1800	0·052
3	270	0·067	3600	0·139
4	270	0·193	7200	0·463

In the third experiment we investigate the efficiency of the smoothing operators on an irregular basin, i.e. the geometry of the IJsselmeer. Figure 1 shows the geometry of the IJsselmeer used in this experiment. We chose $\Delta x = \Delta y = 1{\cdot}0$ km and $h = 6{\cdot}5$ m. The IJsselmeer was represented by about 1100 grid points in the horizontal direction. The vertical representation was made by five layers of the same depth. We integrated over a period of one day with the same constant wind stress as in the first experiment (see (30)). At the endpoint $T = 24$ h we compared the numerical solution for the ζ-component with a reference solution computed with $\tau = 10$ s. Without smoothing the maximally stable time step is about 87 s. In Table IV we list the results.

In this experiment the maximal value for the water elevation is about 0·79 m. The results in this experiment are comparable with the results on a rectangular domain (see Table I). The accuracy

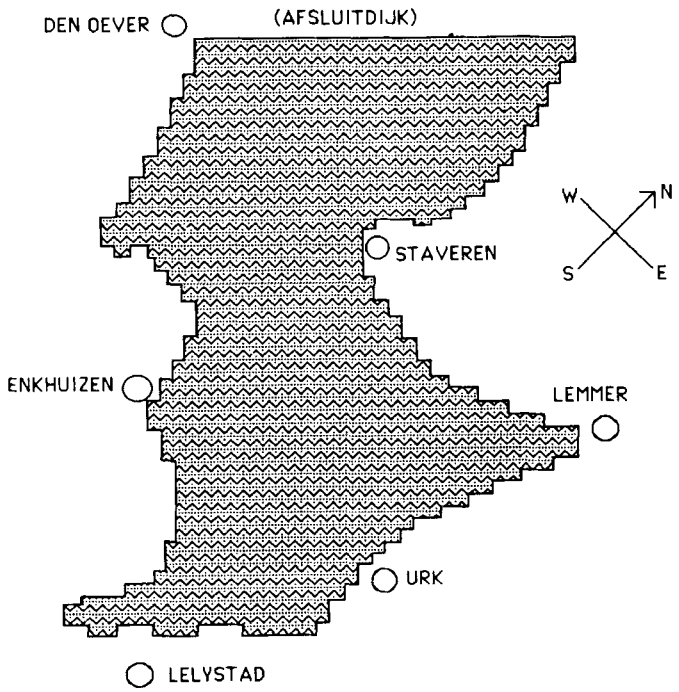

Figure 1. The geometry of the IJsselmeer

Table IV. IJsselmeer problem with a constant wind stress

q	τ (s)	ERROR-ζ (m)	TOT_V (s)	SMO_{VP} (s)
0	80	0·000	200·3	0·0
1	80	0·001	280·4	22·1
2	80	0·009	305·4	48·9
3	80	0·021	339·4	77·5
1	270	0·001	86·0	6·6
2	600	0·009	44·4	9·4
3	1200	0·030	24·8	6·9

is hardly reduced by the smoothing procedure. Moreover, the overhead due to the smoothing operator is even less than in the experiment with a rectangular domain. This is due to the fact that the smoothing matrix has been computed on the irregular domain representing the Ijsselmeer, whereas the computations that do not involve the smoothing have been performed on a surrounding rectangular domain. On vector–parallel computers this is in general an efficient approach because direct addressing can be used in most cases. The efficiency depends on the number of dummy grid points in the surrounding rectangle compared with the number of grid points in the irregular (physical) domain. However, regardless of the implementation used, it may be concluded that the smoothing operator can be implemented efficiently on both regular and irregular domains.

7. CONCLUSIONS

In this paper we applied right-hand-side smoothing to improve the stability of a time integrator for the linearized 3D shallow water equations. We started with the semi-implicit time integrator developed in Reference 1. It turned out that this method may be considered as a method in which the right-hand-side function is premultiplied by an implicit smoothing operator. The vertical terms were treated implicitly. Since the number of points in the vertical direction may be very small, explicit smoothing cannot be applied. Moreover, the stability condition imposed by the vertical terms is often the most restrictive one. Therefore we preferred an implicit treatment of the vertical terms.

In the horizontal direction we can choose between explicit and implicit smoothing of vector functions. In this paper we applied explicit smoothing whenever possible. Only in cases where explicit smoothing could not be applied (i.e. in narrow regions) did we use implicit smoothing. It turned out that this approach is efficient, especially on vector–parallel computers.

Owing to the smoothing in the horizontal direction, the maximally stable time step increased considerably while the accuracy decreased only slightly. In our wind-driven test problems the maximally stable time step increased by a factor of more than 10 (in the case $q=3$) while the accuracy was still acceptable. In this case the overhead in computation time due to the smoothing was only about 30%. Moreover, the error due to the large time steps was more or less comparable with the error introduced by the smoothing. Thus also for fully implicit methods the accuracy would decrease for such large time steps.

ACKNOWLEDGEMENT

This work was supported by the Dutch Ministry of Transport and Public Works (Rijkswaterstaat).

REFERENCES

1. E. D. de Goede, 'Finite difference methods for the three-dimensional hydrodynamic equations', *Report NM-R8813*, CWI, Amsterdam, 1988.
2. F. W. Wubs, 'Stabilization of explicit methods for hyperbolic partial differential equations', *Int. j. numer. methods fluids*, **6**, 641–657 (1986).
3. P. J. van der Houwen, 'Stabilization of explicit difference schemes by smoothing techniques', in K. Strehmel (ed.), *Numerical Treatment of Differential Equations (Proc. Fourth Seminar Halle: NUMDIFF-4), Teubner-Texte zur Mathematik 104*, B. G. Teubner, Leipzig, pp. 205–215, 1987.
4. R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial Value Problems*, Wiley–Interscience, New York/London, 1967.
5. G. Fisher, 'Ein numerisch verfahren zur errechnung von windstau und gezeiten in randmeeren', *Tellus*, **11**, 60–76 (1959).

6. A. Sielecki, 'An energy conserving difference scheme for storm surge equations', *Mon. Weather Rev.*, **96,** 150–156 (1968).
7. P. J. van der Houwen, C. Boon and F. W. Wubs, 'Analysis of smoothing matrices for the preconditioning of elliptic difference equations', *Z. Angew. Math. Mech.*, **68,** 3, 10 (1988).
8. P. J. van der Houwen, *Construction of Integration Formulas for Initial Value Problems*, North-Holland Publishing Company, Amsterdam, 1977.
9. A. Jameson, 'The evolution of computational methods in aerodynamics', *J. Appl. Mech.*, **50,** 1052–1076 (1983).
10. A. M. Davies, 'Application of the DuFort–Frankel and Saul'ev methods with time splitting to the formulation of a three dimensional hydrodynamic sea model', *Int. j. numer. methods fluids*, **5,** 405–425 (1985).