Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

E.D. de Goede

Stabilization of a time integrator for the 3D shallow water equations by smoothing techniques

Department of Numerical Mathematics Report NM-R8916 August

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Stabilization of a Time Integrator for the 3D Shallow Water Equations by Smoothing Techniques

Erik D. de Goede
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009AB Amsterdam, The Netherlands

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 vertical terms are treated implicitly. The stability condition on the time step only depends on the horizontal mesh sizes. Therefore, in horizontal direction an explicit smoothing operator is added. Due to the smoothing, the maximally stable time step increases considerably, while the accuracy is hardly affected. Moreover, it turns out that the explicit smoothing operator is efficient on any type of computer. In particular, the smoothing operator is very efficient on vector and parallel computers.

1980 Mathematics Subject Classification: 65M10, 76D99.
Key Words & Phrases: three-dimensional shallow water equations, method of lines, time integrators, smoothing, stability, vector and parallel computers.
Note: This work was supported by the Ministry of Transport and Public Works (RWS).
Note: This report will be submitted for publication elsewhere.

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 S to some vector **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 S is given by $\mathbf{G} = S\mathbf{F}$, where

$$\begin{aligned} G_1 &= F_1 \\ G_i &= \frac{1}{4}(F_{i-1} + 2F_i + F_{i+1}) , \quad i=2,\dots,m-1, \\ G_m &= F_m \end{aligned} \tag{1.1}$$

with F_i and G_i denoting the components of the vectors **F** and **G**, respectively.
In this report we will apply a smoothing technique to improve the stability of a semi-implicit time integrator that has been developed for the linearized three-dimensional shallow water equations (SWEs) [3]. In this method the vertical terms are treated implicitly. It will be shown that the time integrator can be considered as a method in which an implicit smoothing operator already appears. For this method we are faced with a C.F.L. 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 an explicit smoothing operator to make the stability condition, due to the horizontal mesh sizes, less restrictive. Both the implicit and the explicit smoothing may be

Report NM-R8916
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

interpreted as a preconditioning of the right-hand side of the semi-discrete shallow water equations. It will be shown that the maximally stable time step increases considerably when the explicit smoothing operator is applied. The time step for the stabilized time integrator is now dictated by accuracy considerations, which is also the case for implicit methods. Moreover, the stabilized time integrator can be computed efficiently, as will be shown in the experiments.
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 [9]. For an overview of various smoothing techniques we refer to [4].
In Section 2 the theory is presented. 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. In Section 5 we discuss 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. 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

$$\frac{\partial \mathbf{w}}{\partial t} = L\mathbf{w}(t,\mathbf{x}) + \mathbf{c}(t,\mathbf{x}), \tag{2.1}$$

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 semi-discretized into a system of ordinary differential equations (ODEs) of the form

$$\frac{d\mathbf{W}}{dt} = J\,\mathbf{W}(t) + \mathbf{C}(t), \tag{2.2}$$

with 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 semi-discretization. We shall always assume that this system is stable in the sense that the eigenvalues of J are in the nonpositive half plane. In Section 3 we shall see that the linearized 3D shallow water equations can be semi-discretized into this form.
If the system (2.2) is integrated by an *explicit* time integrator, then its maximally stable time step is limited due to the usually extremely large magnitude of the spectral radius of 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 semi-discretization (2.2), or some part of it, by a *smoothing matrix* S. Thus, we replace (2.2) either by

$$\frac{d\mathbf{W}}{dt} = S\,\{J\,\mathbf{W}(t) + \mathbf{C}(t)\}, \tag{2.3a}$$

or by

$$\frac{d\mathbf{W}}{dt} = SJ\,\mathbf{W}(t) + \mathbf{C}(t). \tag{2.3b}$$

In (2.3b) a part of the right-hand side is smoothed. The semi-discretization (2.3a) 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 semi-discretization (2.2) is also a 'smooth' grid function, so that it may be premultiplied by the smoothing matrix S without much loss of accuracy.
The maximally stable time step may increase considerably when the explicit time integrator is applied to (2.3) instead of to (2.2). To achieve that the condition of SJ is better than that of J, the matrix S

should strongly damp the high frequencies (stiff components) in the Fourier expansion of the vector **JW**, so that the spectral radius of SJ is substantially less than that of J. One may consider the equations (2.3) as 'smoothed' or 'preconditioned' semi-discretizations of the original equation (2.1).
We emphasize that, in this report, the right-hand side function is smoothed, instead of the grid function **W**(t) itself. The latter type of smoothing is often used. However, it may only be applied, without considerable loss of accuracy, if **W**(t) itself is a 'smooth' grid function for a fixed value of t. This is in general not the case. An example of this latter type of smoothing is the well-known Lax-Wendroff method [7].
To characterize the effect of right-hand side smoothing on the accuracy of the initial semi-discretization (2.2), we introduce the *order of consistency of smoothing matrices.* Let Δ be the mesh size, then the smoothing matrix S is said to be consistent of order p if $S=I+O(\Delta^p)$ as Δ tends to zero. Hence, S converges to the identity matrix 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

$$\frac{d\mathbf{W}}{dt} = \mathbf{F}(t,\mathbf{W}(t)), \tag{2.2'}$$

by replacing it by the smoothed system

$$\frac{d\mathbf{W}}{dt} = S\,\mathbf{F}(t,\mathbf{W}(t)). \tag{2.3'}$$

Summarizing, the smoothing matrix S should satisfy the following requirements:

(A) S is consistent of order $p \geq 1$
(B) the smoothed system is again stable
(C) the spectral radius of SJ is considerably smaller than that of J
(D) the application of the matrix 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 (simple) explicit methods, provided that the right-hand side function of the system of ODEs (2.2) is premultiplied by a smoothing matrix S such that the condition of the right-hand side function improves 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 matrices that are to a large degree independent of the right-hand side function will be discussed in Section 2.2.

2.1. Smoothing matrices based on operator splitting

Smoothing matrices 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 semi-discretization (2.2), which can be described by

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \tau\,\{J\mathbf{W}^n + \mathbf{C}^n\}, \tag{2.4}$$

where $\mathbf{W}^n$ denotes an approximation to $\mathbf{W}(n\tau)$ and τ is the time step. Let us split the matrix J into

$$J = J_1 + J_2,$$

and let us replace the forward Euler method (2.4) by the splitting method

$$\mathbf{W}^{n+1} - \tau J_2 \mathbf{W}^{n+1} = \mathbf{W}^n + \tau\,\{J_1 \mathbf{W}^n + \mathbf{C}^n\},$$

or, equivalently,

$$\mathbf{W}^{n+1} = (I - \tau J_2)^{-1}\,\{(I + \tau J_1)\mathbf{W}^n + \tau\,\mathbf{C}^n\}\,. \tag{2.5}$$

This method can be rewritten to

$$W^{n+1} = W^n + \tau\, S\,\{J\, W^n + C^n\}. \tag{2.6}$$

with

$$S = (I - \tau\, J_2)^{-1}\,. \tag{2.7}$$

The splitting method (2.6)-(2.7) may be interpreted as the forward Euler method applied to the system of ODEs (2.3a), which is a 'smoothed' version of the initial semi-discretization (2.2), with a smoothing matrix S defined by (2.7). By an appropriate choice of the matrix J_2, this splitting method has much better stability characteristics than the forward Euler method (2.4). For example, the choices $J_2=J$ or $J_2=J/2$ lead to the A-stable methods of Laasonen (backward Euler) and Crank-Nicolson (trapezoidal rule), respectively. An other possibility is to choose 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 [2] and Sielecki [8]. In fact, the method developed in [3] for the linearized shallow water equations may be interpreted as a combination of the Crank-Nicolson method and the approach of Sielecki and Fisher. 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 matrices for general vector functions

The smoothing matrices considered in the previous subsection strongly depend on the specific form of the right-hand side function. In this subsection, we summarize the main properties of the family of smoothing matrices developed in [4,6]. These matrices 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 (2.3'). We will again assume that the eigenvalues of the Jacobian matrix $J:=\partial F/\partial W$ are in the nonpositive half plane.

The smoothing matrix S will be chosen of the form $S=P(D)$, where D is a difference matrix and the smoothing function $P(z)$ is a polynomial. Firstly, we discuss the choice of the matrix D. In our *theoretical* considerations we assume that D is equal to the Jacobian J, normalized by its spectral radius, i.e.,

$$D = \frac{J}{\rho(J)}\,. \tag{2.8}$$

We emphasize that in *practice*, it is generally not attractive to choose D according to (2.8), and we shall employ some cheap approximation to the normalized Jacobian matrix. If D is defined according to (2.8), then the eigenvalues of $SJ=P(D)J$ are given by $\rho(J)zP(z)$, where z runs through the spectrum of D.

The polynomial $P(z)$ will be chosen such that $zP(z)$ remains in the nonpositive half plane. We are now looking for a polynomial such that the magnitude of $zP(z)$ is sufficiently small. It was shown in [4,6] that polynomials of the form

$$P(z) = \frac{U_{2k}\left(\sqrt{1 + z^2}\right)}{2k + 1}, \quad U_{2k}(x) := \frac{\sin((2k+1)\arccos(x))}{\sin(\arccos(x))}, \tag{2.9}$$

minimize the magnitude of $zP(z)$ on the purely imaginary interval $[-i,i]$. 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 $\mathrm{Re}\,\{zP(z)\} \le 0$ for all values with $\mathrm{Re}\,\{z\} \le 0$ (see condition (B)). For this case, the following theorem defines a family of nearly optimal polynomials [4,6]:

Theorem 2.1. *Let* D *be defined by* (2.8), *let* S=P(D) *with* P(z) *defined by*

$$P(z) = \frac{T_k(1+2z^2) - 1}{2k^2 z^2} \; , \; T_k(x) = \cos(k \arccos(x)) \; , \tag{2.10}$$

Then the following assertions hold:
(a) *If* Re $\{z\} \leq 0$ *then* Re $\{zP(z)\} \leq 0$.
(b) *If* z *is purely imaginary, then* $zP(z)$ *is again purely imaginary, and, for sufficiently large* k, *bounded by*

$$\frac{2}{\pi k}. \tag{2.11}$$

Proof. for a proof of (a) we refer to [4,6].
(b): We have to find the maximum of $|zP(z)|$ on $[-i,i]$, or, equivalently,

$$\max \left| T_k(1+2z^2) \frac{1}{2k^2 z} \right| . \tag{2.12}$$

The range of $\{1+2z^2\}$ in (2.12) is $[-1,1]$. On this interval the Chebyshev polynomial $T_k(1+2z^2)$ satisfies the 'so-called' equal ripple property [5], which means that it alternatingly assumes equal maximum and minimum values. Due to the factor $1/(2k^2z)$, let us now assume that the value in (2.12) can be approximated at the smallest value of $|z|$ for which $T_k(1+2z^2)$ reaches its minimum. Thus, we require that

$$T_k(1+2z^2) = \cos(k \arccos(1+2z^2) = -1 \; ,$$

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

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

For this value of z, we obtain that (2.12) is bounded by

$$\frac{\sqrt{2}}{k^2\sqrt{1-\cos(\pi/k)}} \approx \frac{2}{\pi k} , \tag{2.13}$$

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 2.1 can be obtained by using the following factorization theorem (see also Section 5), which justifies the application of these smoothing matrices:

Theorem 2.2. *Let the matrix* D *be defined by* (2.8), *let* $S=P(D)$ *with* $P(z)$ *defined by* (2.10), *let the factor matrices* F_j *be generated by*

$$F_1 = I + D^2, \quad F_{j+1} = (I - 2F_j)^2 \; , \; j > 0 \; ,$$

and let $k = 2^q$. *Then,* S *can be factorized by*

$$S = F_q F_{q-1} \cdots F_1 \; . \; [] \tag{2.14}$$

For a proof of Theorem 2.2 we refer to [4,6].
As mentioned before, in practice we shall choose D equal to some cheap approximation of the normalized Jacobian, which satisfies condition (B). In choosing a difference matrix D the boundary conditions have

to be incorporated in D. This is important to preserve conservation of mass. In this report we shall choose

$$D^2 = \frac{1}{4}\begin{pmatrix} 0 & & & & & 0 \\ 1 & -2 & 1 & & & \\ & & \cdot & \cdot & \cdot & \\ & & & 1 & -2 & 1 \\ 0 & & & & & 0 \end{pmatrix}. \tag{2.15}$$

Let us now discuss the order of consistency of the smoothing operator with D defined by (2.15). We assume that D and P(z) satisfy the conditions

$$D = O(\Delta^s) \text{ as } \Delta \to 0, \quad P(z) = 1 + O(z^r) \text{ as } z \to 0, \tag{2.16}$$

where Δ denotes the mesh size and r and s are positive integers. Hence, S is consistent of order p=rs. For example, the smoothing matrix defined by (1.1) can be generated by $P(z)=1+z^2$ with D defined by (2.15) and is second-order consistent (s=2, r=1). When P(z) is defined by (2.10) and D by (2.15), then it can be easily verified that S is also second-order consistent.
Summarizing, if we choose the matrix D defined by (2.15), then the smoothing matrix S=P(D), with P(z) the polynomial (2.10), improves the condition of the right-hand side function considerably, whereas the spectrum associated with the smoothed right-hand side function remains in the nonpositive half plane. For this choice of D, the smoothing matrix S is independent of the right-hand side function.
In Section 4 we shall use both a smoothing matrix based on operator splitting and a smoothing of general vector functions based on Theorem 2.1.

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	undisturbed depth of water
t	time
u,v	velocity components in x- and y-direction
x,y,σ	a left-handed set of coordinates
W_f	wind stress
ρ	density
ζ	elevation above undisturbed depth
φ	angle between wind direction and the positive x-axis.

The simplified, linearized three-dimensional test model in sigma coordinates [3] is described by

$$\frac{\partial u}{\partial t} = fv - g\frac{\partial \zeta}{\partial x} + \frac{1}{\rho}\frac{1}{h^2}\frac{\partial\left(A^\sigma \frac{\partial u}{\partial \sigma}\right)}{\partial \sigma} \tag{3.1}$$

$$\frac{\partial v}{\partial t} = -fu - g\frac{\partial \zeta}{\partial y} + \frac{1}{\rho}\frac{1}{h^2}\frac{\partial\left(A^\sigma \frac{\partial v}{\partial \sigma}\right)}{\partial \sigma} \tag{3.2}$$

$$\frac{\partial \zeta}{\partial t} = -\frac{\partial}{\partial x}\left(h\int_0^1 u\,d\sigma\right) - \frac{\partial}{\partial y}\left(h\int_0^1 v\,d\sigma\right), \tag{3.3}$$

with boundaries

$0 \le x \le L$
$0 \le y \le B$
$1 \ge \sigma \ge 0$.

Thus, the domain is a rectangular basin. Due to the sigma transformation in the vertical, the domain is constant in time. We have the closed boundary conditions

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

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

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

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

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

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 (3.1)-(3.3) the computational domain is covered by an $nx \cdot ny \cdot ns$ rectangular staggered grid (see [3]). For the approximation of the spatial derivatives, second-order central finite differences are used in both the horizontal and vertical direction.
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,
$\Lambda_{\sigma\sigma}$ is a tridiagonal matrix approximating the vertical diffusion term,

Θ_1 is an (nx·ny·ns)·(nx·ny) matrix (a row of ns diagonal matrices of order $(nx \cdot ny)^2$ with $\Delta\sigma_k$ on the diagonal of the kth submatrix),
Θ_2 is an (nx·ny)·(nx·ny·ns) matrix (a column of ns identity matrices of order $(nx \cdot ny)^2$),
F is a four-diagonal matrix (due to the grid staggering) of order $(nx \cdot ny \cdot ns)^2$, approximating the Coriolis term,
D_x and D_y are bidiagonal matrices (one diagonal and one lower diagonal) of order $(nx \cdot ny)^2$, approximating the differential operators $\partial/\partial x$ and $\partial/\partial y$, respectively,
E_x and E_y are bidiagonal matrices (one diagonal and one upper diagonal) with $E_x = -D_x^T$ and $E_y = -D_y^T$.
The matrices D_x and E_x differ because of the grid staggering.
Now, the semi-discretized system can be written in the form

$$\frac{d}{dt}\mathbf{W} = \mathbf{F}(\mathbf{W}) = (A + B)\,\mathbf{W} + \mathbf{C}\,, \tag{3.4}$$

with

$$\mathbf{W} = \begin{pmatrix} \mathbf{U} \\ \mathbf{V} \\ \mathbf{Z} \end{pmatrix},\ A = \begin{pmatrix} \Lambda_{\sigma\sigma} & 0 & 0 \\ -F & \Lambda_{\sigma\sigma} & 0 \\ -\Theta_1 hE_x & -\Theta_1 hD_y & 0 \end{pmatrix},\ B = \begin{pmatrix} 0 & F & -\Theta_2 gD_x \\ 0 & 0 & -\Theta_2 gE_y \\ 0 & 0 & 0 \end{pmatrix},\ \text{and}\ \mathbf{C} = \begin{pmatrix} \mathbf{F}_u \\ \mathbf{F}_v \\ 0 \end{pmatrix}. \tag{3.5}$$

The reason for this splitting will become clear in the next sections. Vector **C** contains the components of the wind stress. Note that the integrals in (3.3) are approximated by $\Theta_1\mathbf{U}$ and $\Theta_1\mathbf{V}$, respectively.

3.2. Time integration

In [3] the following time integrator for (3.4)-(3.5) has been developed:

$$\begin{pmatrix} I-\tau\Lambda_{\sigma\sigma} & 0 & 0 \\ \tau F & I-\tau\Lambda_{\sigma\sigma} & 0 \\ \tau\Theta_1 hE_x & \tau\Theta_1 hD_y & I \end{pmatrix} \begin{pmatrix} \mathbf{U}^{n+1} \\ \mathbf{V}^{n+1} \\ \mathbf{Z}^{n+1} \end{pmatrix} = \begin{pmatrix} I & \tau F & -\tau\Theta_2 gD_x \\ 0 & I & -\tau\Theta_2 gE_y \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} \mathbf{U}^{n} \\ \mathbf{V}^{n} \\ \mathbf{Z}^{n} \end{pmatrix} + \tau \begin{pmatrix} \mathbf{F}_u^n \\ \mathbf{F}_v^n \\ 0 \end{pmatrix},$$

or, equivalently,

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

This method can be written in the form

$$\mathbf{W}^{n+1} = (I - \tau A)^{-1}\,\{ (I + \tau B)\mathbf{W}^n + \tau\,\mathbf{C}^n \}\,. \tag{3.6}$$

In terms of formula (2.5), we have that $J_1 = B$ and $J_2 = A$. Thus, this time integrator can be considered as a method in which the right-hand side function is preconditioned by the implicit smoothing operator $(I - \tau A)^{-1}$. It can easily be seen that the components are calculated sequentially (firstly **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 [2] and Sielecki [8]. The time step restriction for method (3.6) is given by [3]

$$\tau < \frac{1}{\sqrt{gh}}\,\frac{1}{\sqrt{\dfrac{1}{(\Delta x)^2} + \dfrac{1}{(\Delta y)^2}}}\,\sqrt{1 + \frac{\tau}{2(\Delta\sigma)^2}\,\frac{A^\sigma}{\rho h^2}}\,, \tag{3.7}$$

where Δx, Δy and $\Delta\sigma$ denote the mesh sizes. We remark that the time step in (3.7) hardly depends on the vertical mesh size $\Delta\sigma$.

4. Smoothing

In this section the stability of method (3.6) will be improved by a smoothing of general vector functions (cf. Section 2.2).

At first, we consider method (3.6). This method can be rewritten to

$$\mathbf{W}^{n+1} = \mathbf{W}^{n} + \tau\,(I - \tau A)^{-1}\,\{\,\mathbf{F}(\mathbf{W}^{n})\,\}\,, \tag{4.1}$$

and may therefore be interpreted as the forward Euler method in which the right-hand side function has been smoothed by the matrix $(I - \tau A)^{-1}$. The vertical terms are treated implicitly, because matrix A contains the discretizations 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 (3.7)). Hence, we add another preconditioning of the right-hand side function, viz. a smoothing of general vector functions based on Theorem 2.1.
The right-hand side function of the **U**-component only contains derivatives in x-direction and will therefore be smoothed in 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 2.2) is only feasible in one-dimensional cases. Therefore, we apply one-dimensional smoothing in x- and y-direction, successively.
In the x-direction the smoothing matrix has the simple structure

$$\begin{pmatrix} S_U & & \\ & 0 & \\ & & S_Z \end{pmatrix},$$

where S_U and S_Z denote the smoothing matrices for the right-hand side function of the **U**- and **Z**-component, respectively. Here, $S_U=P(D_1)$ and $S_Z=P(D_2)$ with P(z) defined by (2.10) and

$$D_1 = (1/4)\begin{pmatrix} 0 & & & & & 0 \\ 1 & -2 & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & 1 & -2 & 1 \\ 0 & & & & & 0 \end{pmatrix}, \quad \text{and} \quad D_2 = (1/4)\begin{pmatrix} -1 & 1 & & & & 0 \\ 1 & -2 & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & 1 & -2 & 1 \\ 0 & & & & -1 & 1 \end{pmatrix}. \tag{4.2}$$

In the y-direction the smoothing matrix has a similar simple structure. Note that D_1 and D_2 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). Therefore, the values in the first and last row are computed in advance.
Summarizing, the time integration method can be written in the form

$$\mathbf{W}^{n+1} = \mathbf{W}^{n} + \tau\,(I - \tau A)^{-1}\,S\,\{\,\mathbf{F}(\mathbf{W}^{n})\,\}\,. \tag{4.3}$$

with the matrix A defined in (3.5) and the smoothing matrix S defined in Theorem 2.1. The stability condition for method (4.3) reads (cf. (2.13) and (3.7))

$$\tau < \frac{\pi k}{2}\,\frac{1}{\sqrt{gh}}\,\frac{1}{\sqrt{\dfrac{1}{(\Delta x)^2} + \dfrac{1}{(\Delta y)^2}}}\,\sqrt{1 + \frac{\tau}{2(\Delta\sigma)^2}\,\frac{A^{\sigma}}{\rho h^2}}\,. \tag{4.4}$$

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 $(I - \tau A)^{-1}$ and S (see (4.3)). For the **U**- and **V**-component, the smoothing matrix $(I - \tau A)^{-1}$ requires the solution of nx·ny tridiagonal systems of order ns, which can be computed efficiently [3]. The smoothing matrix S can be computed in various ways. The most efficient implementation is based on the factorization property presented in Theorem 2.2. If the factor matrices of (2.14) are computed in advance, then the evaluation of P(D) only requires q (= $^2\log(k)$) matrix-vector operations.
E.g., applying Theorem 2.2 for matrix D_1 (see (4.2)), we find the factor matrices

$$F_1 = (1/4)\begin{pmatrix} 4 & & & & & 0 \\ 1 & 2 & 1 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 2 & 1 \\ 0 & & & & & 4 \end{pmatrix}, \qquad F_2 = (1/4)\begin{pmatrix} 4 & & & & & & 0 \\ 2 & 1 & 0 & 1 & & & \\ 1 & 0 & 2 & 0 & 1 & & \\ & 1 & 0 & 2 & 0 & 1 & \\ & & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \text{ etc.}$$

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

6. Numerical experiments

In this section we show for a test problem [1,3] the effects of smoothing on the stability and on the accuracy. In this test problem the water is initially at rest and the motion in the basin is generated by a wind stress. The closed rectangular basin has dimensions representing the North Sea. Thus, a wind driven circulation is gradually developed. We carry out two experiments, one with a constant wind stress and one with a time-dependent wind stress.
The following parameter values are used in the experiments :

$L = 400$ km
$\Delta x = 10$ km
$B = 800$ km
$\Delta y = 10$ km
$\Delta\sigma = 0.25$
$f = 0.44/3600 = 1.22$ e-4
$g = 9.81$ [m/s^2]
$h = 65$ m
$A^\sigma = 0.065/\rho$
$\rho = 1$
$\varphi = 90$ [deg.] (= northerly wind) .

For the time integration we use the method (4.3). In the experiments we vary the number of smoothing factors in the factorized smoothing operator (see (2.14)). The computations have been performed on a grid with nx = 41, ny = 81 and ns = 4. 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.
We integrated over a period of seven days with the constant wind stress

$$W_f = 1.5 \text{ [kg m/s}^2] . \tag{6.1}$$

At that time, the steady state has already been reached. To represent the results, we use the following notation:

q	: number of smoothing factors
NT	: number of time steps
$TOT_{VP/S}$	: total computation time
$SMO_{VP/S}$	: computation time for the smoothing operator
ζ	: water elevation at the south-west corner of the basin.

The indices VP or S indicate Vector and Parallel optimization or Scalar optimization. Thus, the experiment is carried out on one processor if the scalar optimization is used only. We experimentally determined the maximally stable time step for each value of q. In Table 6.1 the results are listed for these time steps. The water elevation reaches its maximum at the south-west corner of the basin.

Table 6.1 : Test problem with a constant wind stress.

q	τ (sec.)	NT	TOT_{VP} (sec.)	SMO_{VP} (sec.)	TOT_S (sec.)	SMO_S (sec.)	ζ (m.)
0	280	2160	369.4	0.0	2781.2	0.0	1.086
1	850	712	166.7	17.1	1410.5	234.2	1.086
2	1800	336	87.1	16.4	780.8	223.6	1.085
3	3600	168	47.2	11.9	449.2	170.7	1.081
4	7200	84	26.0	8.3	256.8	117.3	1.075
5	14400	42	14.0	5.2	145.9	76.6	1.092

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, small errors occur. This is due to the fact that for these values of q the steady state has not been reached yet. 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.

In Table 6.2 we give the gain factors of the maximally stable time steps compared with the case q = 0, and we compare them with the theoretical gain factors. Moreover, we list the gain factors in computation times.

Table 6.2 : 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 6.1)	3.0	6.4	12.8	25.7	51.4
in computation time (VP)	2.2	4.2	7.8	14.2	26.4
in computation time (S)	2.0	3.6	6.2	10.8	19.1

The theoretical reduction factor $2^{q-1}\pi$ (cf. (4.4) with $k=2^q$) is in agreement with the experimental results. The results show a significant reduction in computation time, especially when the vector and

parallel optimization is used. The overhead due to the smoothing operator is less than a factor two, even for large values of q. In the case of the vector and parallel optimization, the computation time is reduced by about a factor 3 due to the vectorization, and also by a factor 3 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, we introduce a time-dependent wind stress (cf. (6.1))

$$W_f = 1.5 * \left(1 + 0.5 * \sin \frac{2\pi t}{24 * 3600} \right). \tag{6.2}$$

Now, we have a periodic varying (northerly) wind with a period of 24 hours. We integrated over a period of 10 days. The solution was periodic. In the case without smoothing in the horizontal we obtained the following maximal and minimal water elevations at the south-west corner of the basin:

ζ = 2.323 m at t = 7 + iP hours
ζ = -0.154 m at t = 19 + iP hours,

with period P = 24 hours and i a positive integer. When smoothing is applied we observe that the maximal and minimal water elevations are reached at the same points in time as in the case without smoothing in the horizontal. It seems that the smoothing operator does not introduce a dispersion error. However, some errors in the amplitude of the periodic solution appear. In Table 6.3 we list the maximal error in the numerical solution for the water elevation measured over the whole field at T = 223 hours compared with the case q = 0.

Table 6.3 : Test problem with a time-dependent wind stress.

q	τ (sec.)	error (m.)	τ (sec.)	error (m.)
0	280	0.0		
1	280	0.004	850	0.020
2	280	0.025	1800	0.063
3	280	0.088	3600	0.179
4	280	0.235	7200	0.484

The results show that the error due to the smoothing operator is even smaller than the error due to the larger time steps. E.g., in the case q = 2 the error due to the larger time steps (viz. 0.038 m.) is larger than the error due to the smoothing (viz. 0.025 m.). Thus, when a fully implicit method would have been used, the accuracy would also decrease for large time steps.

7. Conclusions

In this report 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 [3]. 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 vertical direction may be very small, explicit smoothing can not 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 used explicit smoothing. It turned out that this approach is efficient, especially on vector and parallel computers.
Due to the explicit smoothing, 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 with about a factor 12 (in the case q=3), while the accuracy was still acceptable. In this case the overhead in computation time due to the explicit smoothing was only about 35%. Moreover, the error due to the large time steps was even larger than the error introduced by the smoothing. Thus, also for fully implicit methods the accuracy would decrease for such large time steps.

8. References

[1] 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, *Intern. J. Numer. Methods Fluids*, **5** (1985), pp. 405-425.

[2] G. FISHER, Ein numerisch verfahren zur errechnung von windstau und gezeiten in randmeeren, *Tellus*, **11** (1959), pp. 60-76.

[3] E.D. DE GOEDE, *Finite difference methods for the three-dimensional hydrodynamic equations*, Report NM-R8813, CWI, Amsterdam, 1988.

[4] 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 1987: NUMDIFF-4), Teubner-Texte zur Mathematik 104, BSB B.G. Teubner Verlaggesellschaft, Leipzig, pp. 205-215.

[5] P.J. VAN DER HOUWEN, *Construction of integration formulas for initial value problems*, INorth-Holland, Amsterdam, 1977.

[6] 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** (1988), pp. 3-10.

[7] R.D. RICHTMYER AND K.W. MORTON, *Difference methods for initial value problems*, Interscience Publishers, Wiley, New York, London, 1967.

[8] A. SIELECKI, An energy conserving difference scheme for storm surge equations, *Monthly Weather Review*, **96** (1968), pp. 150-156.

[9] F.W. WUBS, Stabilization of explicit methods for hyperbolic partial differential equations, *Internat. J. Numer. Methods Fluids*, **6** (1986), pp. 641-657.