INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 25, 1229–1240 (1997)

A MULTIGRID SEMI-IMPLICIT FINITE DIFFERENCE METHOD FOR THE TWO-DIMENSIONAL SHALLOW WATER EQUATIONS

R. M. SPITALERI* AND L. CORINALDESI

Istituto per le Applicazioni del Calcolo-CNR, Viale del Policlinico 137, I-00161 Roma, Italy

SUMMARY

A multigrid semi-implicit finite difference method is presented to solve the two-dimensional shallow water equations which describe the behaviour of basin water under the influence of the Coriolis force, atmospheric pressure gradients and tides. The semi-implicit finite difference method discretizes implicitly both the gradient of the water elevation in the momentum equations and the velocity divergence in the continuity equations and explicitly the convective terms using an Eulerian–Lagrangian approach. At each time step we apply the multigrid computation to solve the resulting linear, symmetric, pentadiagonal system of discrete equations. The multigrid algorithm, defined on staggered grids, provides accelerated convergence histories. We numerically simulate the water circulation in a closed rectangular basin, centrally crossed by a deeper channel. Moreover, simulation of the circulation in San Pablo Bay shows the high flexibility and applicability of this method to concrete problems. Visualizations of the computed variables, water depth and velocity, are shown by figures. Displays of convergence histories show promising multigrid acceleration. © 1997 John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **25**: 1229–1240 (1997)

No. of Figures: 12. No. of Tables: 0. No. of References: 6.

KEY WORDS: shallow water equations; semi-implicit finite difference discretization; multigrid computation; staggered grid; Gauss–Siedel relaxation

1. INTRODUCTION

Multigrid computation has been successfully applied in several research fields and has been widely investigated for performance evaluation. We could briefly say that multigrid methods solve a system of discrete equations on a given grid by defining interactions with a hierarchy of auxiliary grids and applying appropriate information transfer operators. Besides providing multigrid acceleration, this computational strategy appears to have several other capabilities, e.g. handling non-linear problems and using adaptive grid refinement.[1]

We have already coupled multigrid computation with numerical grid generation methods[2] and now we apply this powerful computational technique to the solution of the shallow water equations discretized by a semi-implicit finite difference discretization method. We deal with the numerical approach proposed and tested by Casulli.[3] In this approach the continuous shallow water equations are discretized in such a way that a linear, symmetric, pentadiagonal system for the shallow water elevation is defined on a staggered grid and has to be solved. For this step of the whole solution

* Correspondence to: R. M. Spitaleri, Istituto per le Applicazioni del Calcolo-CNR, Viale del Policlinico 137, I-00161 Roma, Italy.

CCC 0271–2091/97/111229–12 $17.50

Received December 1995
Revised November 1996

procedure, multigrid computation seems to be able to solve this system and provide accelerated convergence histories.

This paper presents the specific multigrid algorithm for the solution of the semi-implicit discretization of the two-dimensional shallow water equations which has been defined, implemented and tested with promising results. In order to have a self-contained presentation, we specify the governing equations and their semi-implicit discretization, along with the explicit discretization of the convective terms by an Eulerian–Lagrangian approach.

Then we introduce generality in the multigrid computation and present the multigrid components of the specific algorithm we tested. Since the variables are defined on staggered grid points, we assume slightly more complex multigrid transfer operators.

We applied the multigrid algorithm to simulate water circulation in artificial and natural basins:

(a) two closed rectangular basins, centrally crossed by a deeper channel,[3] which differ only in their physical extent
(b) San Pablo Bay.

Therefore we describe experimental numerical results which show the capability of the defined algorithm to solve large simulation problems with satisfactory effectiveness. The algorithm provides convergence histories with increasing acceleration depending on the grid number and, when compared with one-grid algorithms, confirms multigrid computation as a promising numerical strategy.

2. GOVERNING EQUATIONS

The two-dimensional shallow water equations are quasi-linear hyperbolic partial differential equations and can be written in the form[3,4]

$$\begin{aligned} &\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+g\frac{\partial z}{\partial x}=-\gamma u+fv,\\ &\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+g\frac{\partial z}{\partial y}=-\gamma v-fu,\\ &\frac{\partial z}{\partial t}+\frac{\partial[(h+z)u]}{\partial x}+\frac{\partial[(h+z)v]}{\partial y}=0, \end{aligned} \tag{1}$$

where $u(x,y,t)$ and $v(x,y,t)$ are the depth-averaged velocity components in the x- and y-direction respectively, $z(x,y,t)$ is the water surface elevation and $h(x,y)$ is the water depth, both measured from the undisturbed water surface, g is the constant gravitational acceleration, γ is the bottom friction coefficient and $f=2\omega\sin\varphi$ is the coefficient of the Coriolis force included in the model. Figure 1 shows the total water depth defined as $H(x,y,t)=h(x,y)+z(x,y,t)$.

Computational limitations of explicit numerical methods for (1) are known, while a fully implicit discretization often becomes very heavy since it leads to the simultaneous solution of a large number of coupled non-linear equations.

We will solve (1) by using a spatially *staggered* mesh and discretizing the governing equations with a *semi-implicit technique*.[3] The convective terms will be discretized by using an accurate Eulerian–Lagrangian approach which provides an unconditionally stable algorithm allowing larger steps to be used.[3] The computational process can be summarized in the following terms. At each time step: we first derive an $n\times m$ linear pentadiagonal system in just one unknown, the new water surface elevation, which is symmetric and positive definite; thus we apply a *multigrid method* to solve this

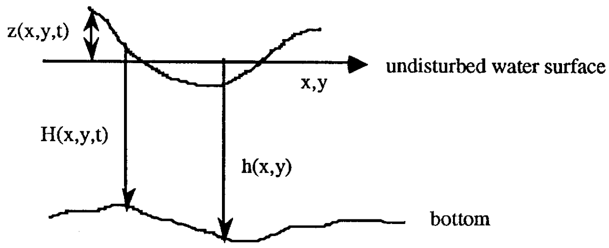

Figure 1. Co-ordinate system for shallow water equations

discrete system which can be solved uniquely; then the fluid velocity is obtained explicitly from the discretized momentum equations.

3. SEMI-IMPLICIT NUMERICAL METHOD

In this section we will derive a semi-implicit discretization of (1). The convective terms will be discretized explicitly; the friction coefficient γ and the Coriolis force will be evaluated explicitly in the momentum equations; the total water depth $H(x, y, t)$ will be taken explicitly in the continuity equations. All other terms of the system will be discretized implicitly.

As shown in Figure 2, the finite difference mesh used to discretize (1) consists of rectangular cells of length Δx and width Δy. These cells are numbered at their centre by indices i and j, with i counting the columns in the x-direction and j counting the rows in the y-direction. Since the water depth $h(x, y)$ is assumed to be known everywhere, the water elevation z is the system variable which is defined at each cell centre. The horizontal component u of the velocity is defined at the centre of each vertical side, while the vertical component v is defined at the centre of each horizontal side. Thus we are dealing with a staggered grid.

We obtain the following semi-implicit discretization of (1) at time $t_k = k\Delta t$:

$$
\begin{aligned}
u^{k+1}_{i+1/2,j} &= Fu^k_{i+1/2,j} - g\frac{\Delta t}{\Delta x}(z^{k+1}_{i+1,j} - z^{k+1}_{i,j}) - \Delta t\gamma^k_{i+1/2,j}u^{k+1}_{i+1/2,j} - f\Delta tFv^k_{i,j+1/2},\\
v^{k+1}_{i,j+1/2} &= Fv^k_{i,j+1/2} - g\frac{\Delta t}{\Delta y}(z^{k+1}_{i,j+1} - z^{k+1}_{i,j}) - \Delta t\gamma^k_{i,j+1/2}v^{k+1}_{i,j+1/2} + f\Delta tFu^k_{i+1/2,j},\\
z^{k+1}_{i,j} &= z^k_{i,j} - \frac{\Delta t}{\Delta x}[(\bar{z}^k_{i+1/2,j} + h_{i+1/2,j})u^{k+1}_{i+1/2,j} - (\bar{z}^k_{i-1/2,j} + h_{i-1/2,j})u^{k+1}_{i-1/2,j}]\\
&\quad - \frac{\Delta t}{\Delta y}[(\bar{z}^k_{i,j+1/2} + h_{i,j+1/2})v^{k+1}_{i,j+1/2} - (\bar{z}^k_{i,j-1/2} + h_{i,j-1/2})v^{k+1}_{i,j-1/2}],
\end{aligned}
\tag{2}
$$

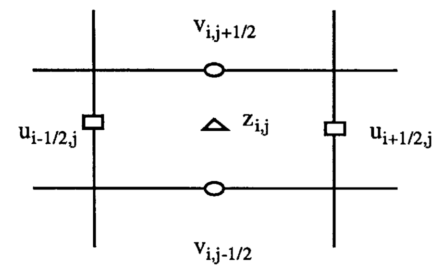

Figure 2. Staggered grid

where $u^k_{i\pm 1/2,j}$ and $v^k_{i,j\pm 1/2}$ are the depth-averaged velocity components and $z^k_{i,j}$ is the water surface elevation. We immediately write (2) in the form

$$\begin{aligned}
(1+\gamma^k_{i+1/2,j}\Delta t)u^{k+1}_{i+1/2,j} &= Fu^k_{i+1/2,j} - g\frac{\Delta t}{\Delta x}(z^{k+1}_{i+1,j} - z^{k+1}_{i,j}) - f\Delta t F v^k_{i,j+1/2},\\
(1+\gamma^k_{i,j+1/2}\Delta t)v^{k+1}_{i,j+1/2} &= Fv^k_{i,j+1/2} - g\frac{\Delta t}{\Delta y}(z^{k+1}_{i,j+1} - z^{k+1}_{i,j}) + f\Delta t F u^k_{i+1/2,j},\\
z^{k+1}_{i,j} &= z^k_{i,j} - \frac{\Delta t}{\Delta x}[(\bar{z}^k_{i+1/2,j} + h_{i+1/2,j})u^{k+1}_{i+1/2,j} - (\bar{z}^k_{i-1/2,j} + h_{i-1/2,j})u^{k+1}_{i-1/2,j}]\\
&\quad - \frac{\Delta t}{\Delta y}[(\bar{z}^k_{i,j+1/2} + h_{i,j+1/2})v^{k+1}_{i,j+1/2} - (\bar{z}^k_{i,j-1/2} + h_{i,j-1/2})v^{k+1}_{i,j-1/2}],
\end{aligned} \tag{3}$$

where $\bar{z}^k_{i\pm 1/2,j}$ and $\bar{z}^k_{i,j\pm 1/2}$ are averaged values, from the closest scalar grid points, computed by the expressions

$$\bar{z}^k_{i\pm 1/2,j} = \tfrac{1}{2}(z^k_{i,j} + z^k_{i\pm 1,j}), \qquad \bar{z}^k_{i,j\pm 1/2} = \tfrac{1}{2}(z^k_{i,j} + z^k_{i,j\pm 1})$$

and F is an explicit non-linear finite difference operator which defines the spatial discretization of the convective terms $u_t + uu_x + vu_y$ and $v_t + uv_x + vv_y$. The specific form of F will be introduced in the following section.

Given any F, system (3), which is linear in the unknowns $u^{k+1}_{i+1/2,j}$, $v^{k+1}_{i,j+1/2}$ and $z^{k+1}_{i,j}$, has to be solved at each time step recursively from assigned initial data. Since this requires most of the computational time, we reduce this system to a smaller one in which $z^{k+1}_{i,j}$ are the only unknowns.[3]

By inserting the expressions for $u^{k+1}_{i+1/2,j}$ and $v^{k+1}_{i,j+1/2}$ obtained from the first two equations into the third equation of system (3), we have

$$\begin{aligned}
z^{k+1}_{i,j} &- g\frac{\Delta t^2}{\Delta x^2}\left(\frac{\bar{z}^k_{i+1/2,j} + h_{i+1/2,j}}{1+\gamma^k_{i+1/2,j}\Delta t}(z^{k+1}_{i+1,j} - z^{k+1}_{i,j}) - \frac{\bar{z}^k_{i-1/2,j} + h_{i-1,2,j}}{1+\gamma^k_{i-1/2,j}\Delta t}(z^{k+1}_{i,j} - z^{k+1}_{i-1,j})\right)\\
&- g\frac{\Delta t^2}{\Delta y^2}\left(\frac{\bar{z}^k_{i,j+1/2} + h_{i,j+1/2}}{1+\gamma^k_{i,j+1/2}\Delta t}(z^{k+1}_{i,j+1} - z^{k+1}_{i,j}) - \frac{\bar{z}^k_{i,j-1/2} + h_{i,j-1/2}}{1+\gamma^k_{i,j-1/2}\Delta t}(z^{k+1}_{i,j} - z^{k+1}_{i,j-1})\right)\\
&= z^k_{i,j} - \frac{\Delta t}{\Delta x}\left(\frac{\bar{z}^k_{i+1/2,j} + h_{i+1/2,j}}{1+\gamma^k_{i+1/2,j}\Delta t}Fu^k_{i+1/2,j} - \frac{\bar{z}^k_{i-1/2,j} + h_{i-1/2,j}}{1+\gamma^k_{i-1/2,j}\Delta t}Fu^k_{i-1/2,j}\right)\\
&\quad - \frac{\Delta t}{\Delta y}\left(\frac{\bar{z}^k_{i,j+1/2} + h_{i,j+1/2}}{1+\gamma^k_{i,j+1/2}\Delta t}Fv^k_{i,j+1/2} - \frac{\bar{z}^k_{i,j-1/2} + h_{i,j-1/2}}{1+\gamma^k_{i,j-1/2}\Delta t}Fv^k_{i,j-1/2}\right)\\
&\quad + f\frac{\Delta t^2}{\Delta x}\left(\frac{\bar{z}^k_{i+1/2,j} + h_{i+1/2,j}}{1+\gamma^k_{i+1/2,j}\Delta t}Fv^k_{i,j+1/2} + \frac{\bar{z}^k_{i-1/2,j} + h_{i-1/2,j}}{1+\gamma^k_{i-1/2,j}\Delta t}Fv^k_{i-1,j+1/2}\right)\\
&\quad - f\frac{\Delta t^2}{\Delta y}\left(\frac{\bar{z}^k_{i,j+1/2} + h_{i,j+1/2}}{1+\gamma^k_{i,j+1/2}\Delta t}Fu^k_{i+1/2,j} + \frac{\bar{z}^k_{i,j-1/2} + h_{i,j-1/2}}{1+\gamma^k_{i,j-1/2}\Delta t}Fu^k_{i+1/2,j-1}\right).
\end{aligned} \tag{4}$$

Let us assume in system (4) that

$$Au(i,j) = g\left(\frac{\Delta t}{\Delta x}\right)^2 \frac{\bar{z}^k_{i+1/2,j} + h_{i+1/2,j}}{1+\gamma^k_{i+1/2,j}\Delta t},$$

$$Av(i,j) = g\left(\frac{\Delta t}{\Delta y}\right)^2 \frac{\bar{z}^k_{i,j+1/2} + h_{i,j+1/2}}{1+\gamma^k_{i,j+1/2}\Delta t},$$

$$Az(i,j) = 1 + Au(i,j) + Au(i-1,j) + Av(i,j) + Av(i,j-1),$$

$$\begin{aligned} b(i,j) = z^k_{i,j} &- \frac{\Delta t}{\Delta x}\left(\frac{\bar{z}^k_{i+1/2,j} + h_{i+1/2,j}}{1+\gamma^k_{i+1/2,j}\Delta t}(Fu^k_{i+1/2,j} - f\Delta t Fv^k_{i,j+1/2}) \right. \\ &\left. - \frac{\bar{z}^k_{i-1/2,j} + h_{i-1/2,j}}{1+\gamma^k_{i-1/2,j}\Delta t}(Fu^k_{i-1/2,j} - f\Delta t Fv^k_{i-1,j+1/2})\right) \\ &- \frac{\Delta t}{\Delta y}\left(\frac{\bar{z}^k_{i,j+1/2} + h_{i,j+1/2}}{1+\gamma^k_{i,j+1/2}\Delta t}(Fv^k_{i,j+1/2} + f\Delta t Fu^k_{i+1/2,j}) \right. \\ &\left. - \frac{\bar{z}^k_{i,j-1/2} + h_{i,j-1/2}}{1+\gamma^k_{i,j-1/2}\Delta t}(Fv^k_{i,j-1/2} + f\Delta t Fu^k_{i+1/2,j-1})\right). \end{aligned}$$

By applying rotated lexicographic ordering to the unknowns $z^{k+1}_{i,j}$, we obtain the following linear pentadiagonal system of equations:

$$-Au(i-1,j)z^{k+1}_{i-1,j} - Av(i,j-1)z^{k+1}_{i,j-1} + Az(i,j)z^{k+1}_{i,j} - Av(i,j)z^{k+1}_{i,j+1} - Au(i,j)z^{k+1}_{i+1,j} = b(i,j). \quad (5)$$

The assumptions $(\bar{z}^k + h)_{i\pm1/2,j} > 0$ and $(\bar{z}^k + h)_{i,j\pm1/2} > 0$ lead system (5) to be symmetric and strictly diagonally dominant, with positive elements on the main diagonal and negative ones elsewhere.[3] That is, the system is positive definite with a unique solution which can be very efficiently solved by a multigrid algorithm.[1]

4. CONVECTIVE TERM DISCRETIZATION

Let us use an Eulerian–Lagrangian approach to discretize the convective terms and obtain an explicit form for F which is relatively accurate and unconditionally stable.[3]

Let us rewrite the convective terms as Lagrangian derivatives

$$\frac{d\omega}{dt} = \frac{\partial\omega}{\partial t} + u\frac{\partial\omega}{\partial x} + v\frac{\partial\omega}{\partial y}, \quad (6)$$

where the substantive derivative d/dt indicates that the temporal rate of change is calculated along streamlines defined by

$$\frac{dx}{dt} = u, \qquad \frac{dy}{dt} = v. \quad (7)$$

Let us denote by a and b the Courant numbers

$$a = u\frac{\Delta t}{\Delta x}, \qquad b = v\frac{\Delta t}{\Delta y}.$$

We derive from (6) the correct physical expression for $F\omega^k_{i,j}$ in the form

$$F\omega^k_{i,j} = \omega^k_{i-a,j-b}.$$

Since in general a and b are not integers, $(i-a, j-b)$ is not a grid point. Thus, in dealing with Eulerian–Lagrangian methods which use a generalization of the interpolation concept, i.e. they define interpolation over three or more mesh points either including or not including point (i, j), we approximate $\omega^k_{i-a,j-b}$ with the following bilinear interpolation over the four surrounding mesh points:[3]

$$F\omega^k_{i,j} = (1-p)[(1-q)\omega^k_{i-n,j-m} + q\omega^k_{i-n,j-m-1}] + p[(1-q)\omega^k_{i-n-1,j-m} + q\omega^k_{i-n-1,j-m-1}], \quad (8)$$

where $a = n + p$ and $b = m + q$, with n and m integers and $p, q \in [0, 1)$.

Having the correct values of a and b means solving the ordinary differential equations (7). Let us assume that the values of u and v, known at time level t_k, do not vary over a time step. At each mesh point (i, j), equations (7) are integrated numerically backward from t_{k+1} to t_k by using the Euler method. By dividing the time step into N equal parts of length $\tau = \Delta t/N$, we have the discrete system

$$\begin{aligned} x^{s-1} &= x^s - \tau u^k(x^s, y^s), \qquad & x^N &= x_i, \\ y^{s-1} &= y^s - \tau v^k(x^s, y^s), \qquad & y^N &= y_j, \qquad s = N, N-1, \ldots, 2, 1, \end{aligned}$$

where $u^k(x^s, y^s)$ and $v^k(x^s, y^s)$ are provided by an interpolation operator analogous to (8). Then, at (x_i, y_j), a and b are given by

$$a = \frac{x_i - x^0}{\Delta x}, \qquad b = \frac{y_j - y^0}{\Delta y}.$$

In this way the streamlines, which in general are not straight lines, are better approximated. We recall that the amplification factor of the operator F expressed by (8), with u and v constant, is[3]

$$\begin{aligned} f = {} & [\cos(n\alpha) - \mathrm{I}\sin(n\alpha)][\cos(m\beta) - \mathrm{I}\sin(m\beta)][1 - p + p\cos(\alpha) - \mathrm{I}p\sin(\alpha)] \\ & \times [1 - q + q\cos(\beta) - \mathrm{I}q\sin(\beta)], \end{aligned}$$

and since $0 \leqslant p < 1$ and $0 \leqslant q < 1$, it results that $|f| \leqslant 1$ identically for every α and β and with no restriction on the time step. Thus applying the Eulerian–Lagrangian approach to discretize the convective terms leads to unconditionally stable resulting difference equations (3). Just a restriction on the time subdivision τ must be imposed to approximate streamlines not crossing the solid boundaries.[3]

5. MULTIGRID COMPUTATION

At each time step t_{k+1} let us rewrite the linear system (5) in the form

$$L^{k+1}_M Z^{k+1}_M = F^{k+1}_M \quad \text{on } \mathrm{G}_M, \quad (9)$$

where G_M is the staggered grid, with spatial mesh sizes Δx and Δy, which covers the physical domain. To obtain a fast solution of system (9) via the multigrid method, we add to G_M a sequence of coarser uniform grids $\mathrm{G}_0, \mathrm{G}_1, \ldots, \mathrm{G}_{M-1}$ provided by the widely used coarsening method called mesh size doubling: $\Delta x_l = 2\Delta x_{l+1}$ and $\Delta y_l = 2\Delta y_{l+1}$, where $l = 0, 1, \ldots, M-1$ is the *level number*.[1,5,6]

If the approximation z_M of the solution Z^{k+1}_M is available, the *correction* $V^{k+1}_M = Z^{k+1}_M - z_M$ satisfies the correction equation

$$L^{k+1}_M V^{k+1}_M = \tilde{F}^{k+1}_M \quad \text{on } \mathrm{G}_M,$$

where $\tilde{F}^{k+1}_M = F^{k+1}_M - L^{k+1}_M z_M$ is called the *residual*.

We can see immediately that solving the correction equation or equation (9) is completely equivalent. V_M^{k+1} can be well approximated by using the coarser grid G_{M-1} if this correction is 'smooth', i.e. if its high-frequency components are small compared with its low-frequency components. This can be achieved very efficiently by suitable relaxation methods. The same thing is true also for the other grids, then, in the general step of the multigrid cycle.[1]

Let us determine the approximate solution $(V_a)_l^{k+1}$ of the problem

$$L_l^{k+1} V_l^{k+1} = \tilde{F}_l^{k+1} \quad \text{on } G_l,$$

called problem G_l, for $l = 0, \ldots, M-1$, where $\tilde{F}_l^{k+1} = I_{l+1}^l(\tilde{F}_{l+1}^{k+1} - L_{l+1}^{k+1} V_{l+1}^{k+1})$ and I_{l+1}^l is an appropriate fine-to-coarse transfer operator or *restriction*.

Finally let us determine a new approximation

$$\tilde{V}_{l+1}^{k+1} = V_{l+1}^{k+1} + I_l^{l+1}(V_a)_l^{k+1},$$

where I_l^{l+1} is an appropriate coarse-to-fine transfer operator or *prolongation*.

The linear *multigrid* algorithm (LMG) is recursively defined on the basis of a *two-grid* method. One iteration of the $(l+1, l)$ two-grid method computing $\hat{z}_{l+1}^{k+1}$ from z_{l+1}^{k+1} is composed of the following steps.[5,6]

1. *Pre-smoothing*. Compute $\bar{z}_{l+1}^{k+1}$ by applying υ_1 ($\geqslant 0$) sweeps of a given relaxation method with starting guess z_{l+1}^{k+1}:

$$\bar{z}_{l+1}^{k+1} = \text{Relax}^{\upsilon_1}(z_{l+1}^{k+1}, L_{l+1}^{k+1}, F_{l+1}^{k+1}).$$

2. *Coarse-grid correction*
   (a) computation of the residual $\tilde{F}_{l+1}^{k+1} = F_{l+1}^{k+1} - L_{l+1}^{k+1}\bar{z}_{l+1}^{k+1}$
   (b) restriction of the residual: $\tilde{F}_l^{k+1} = I_{l+1}^l \tilde{F}_{l+1}^{k+1}$
   (c) computation of the exact solution of $L_l^{k+1} V_l^{k+1} = \tilde{F}_l^{k+1}$
   (d) interpolation of the correction: $V_{l+1}^{k+1} = I_l^{l+1} V_l^{k+1}$
   (e) computation of the correct approximation: $\tilde{z}_{l+1}^{k+1} = z_{l+1}^{k+1} + V_{l+1}^{k+1}$.

3. *Post-smoothing*. Compute $\hat{z}_{l+1}^{k+1}$ by applying υ_2 ($\geqslant 0$) sweeps of the given relaxation method with starting guess $\tilde{z}_{l+1}^{k+1}$:

$$\hat{z}_{l+1}^{k+1} = \text{Relax}^{\upsilon_2}(\tilde{z}_{l+1}^{k+1}, L_{l+1}^{k+1}, F_{l+1}^{k+1}).$$

In this description, 'Relax' stands for a relaxation procedure which has suitable error-smoothing properties; υ_1 smoothing steps are performed before and υ_2 smoothing steps are performed after the coarse-grid correction.

Figure 3 shows the computational scheme. Indeed, in the multigrid method the linear coarse-grid equation is not solved exactly, but approximately by several multigrid steps using still coarser grids. Thus the computation of the exact solution V_l^{k+1} is replaced by an approximate solution.[5]

$$z_{l+1}^{k+1} \xrightarrow[\text{Relax}^{\upsilon_1}]{} \bar{z}_{l+1}^{k+1} \rightarrow \tilde{F}_{l+1}^{k+1} = F_{l+1}^{k+1} - L_{l+1}^{k+1}\bar{z}_{l+1}^{k+1} \quad V_{l+1}^{k+1} \rightarrow \tilde{z}_{l+1}^{k+1} = z_{l+1}^{k+1} + V_{l+1}^{k+1} \xrightarrow[\text{Relax}^{\upsilon_2}]{} \hat{z}_{l+1}^{k+1}$$

$$\downarrow I_{l+1}^{l} \qquad\qquad \uparrow I_l^{l+1}$$

$$\tilde{F}_l^{k+1} \longrightarrow L_l^{k+1} V_l^{k+1} = \tilde{F}_l^{k+1}$$

Figure 3. Linear $(l+1, l)$ two-grid method

6. MULTIGRID ALGORITHM ON STAGGERED GRID

Let us note that using staggered grids makes the structure of the multigrid method slightly more complicated. Since there is no correspondence among points on each grid pair, appropriate transfer operators have to be used. For non-staggered grids a natural definition of the restriction is the 'straight injection', i.e. the value at each node on the coarser level comes directly from the corresponding node of the finer level. For staggered grids we have to compute a new value at the fictitious node on the finer level, located in correspondence to each node on the coarser level, and then transfer this value to the coarser-grid node (Figure 4). For prolongation of the correction a weighted linear interpolation appears to be appropriate (Figure 5).

The multigrid algorithmic components and the related algorithms we use are as follows:

(a) coarser-grid construction (hierarchy of grids)→standard coarsening
(b) type of grids→staggered grids
(c) relaxation for error smoothing (Relax$^{\nu_1}$, Relax$^{\nu_2}$)→Gauss–Seidel relaxation, lexicographic ordering
(d) restriction (fine-to-coarse-grid transfer operator, I_{l+1}^{l})→local averaging (Figure 4)
(e) prolongation (coarse-to-fine-grid transfer operator, I_{l}^{l+1})→weighted interpolation (Figure 5)
(f) multigrid cycling (grid changing)→V-, W-cycles.

We apply Gauss–Seidel relaxation since wide experience shows that successive displacement (Gauss–Seidel, Gauss–Seidel–Newton, etc.) schemes are superior to simultaneous displacement (Jacobi, Jacobi–Newton, etc.) ones.[1,6] Furthermore, for successive displacement schemes the order in which the equations are relaxed has an important effect on the smoothing factors. Orderings widely used are lexicographic, red–black and more general pattern relaxation orderings which are similar to red–black but with different and possibly more colours.[5,6] In this multigrid application we tested the Gauss–Seidel relaxation algorithm with lexicographic ordering.

The evaluation of the algorithmic performance has been obtained by the use of convergence histories, which represent values of approximation errors depending on the number of computing iterations.

Summary of a calculation step

Beginning from the initial data, the water circulation is advanced by a series of time steps, each of length Δt. At each time $t_{k+1} = (k+1)\Delta t$ we have to

(a) compute the Lagrangian convective terms by expression (8)
(b) determine the water elevation field $z_{i,j}^{k+1}$ by solving system (5) by the defined linear multigrid algorithm

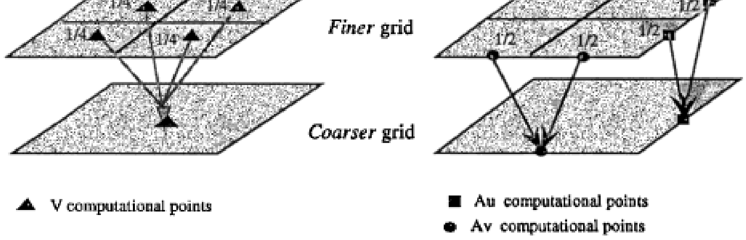

Figure 4. Fine-to-coarse-grid restriction

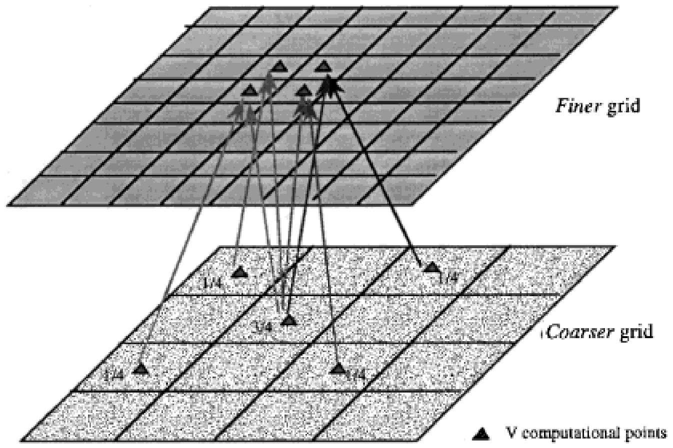

Figure 5. Coarse-to-fine-grid prolongation

(c) update the water velocity components $u^{k+1}_{i\pm1/2,j}$ and $v^{k+1}_{i,j\pm1/2}$ by an explicit calculation with the first two equations of system (3).

7. APPLICATIONS AND NUMERICAL RESULTS

We applied the multigrid algorithm to a set of simulation problems of water circulation in artificial and natural basins and we obtained both satisfactory solutions and promising performance evaluation results. Plates 1, 2 and 3, 4 show results of two problems of water circulation simulation. We display, at each vizualization time step t_{vis}, two-dimensional basins with the total water depth and the velocity field represented by colour and a typical arrow plot respectively. In Plates 1–4 the light-to-dark-blue scale corresponds to increasing water depth.

Performance evaluation has been carried out on the basis of logarithmic values of the residual (*log.err*). By assuming the computational work in one sweep over the finest grid as one work unit (WU), the error behaviour has been evaluated in terms of either WUs or multigrid cycles (*iter*). Multigrid V- and W-cycles have been carried out. A few convergence histories are shown in Figures 6–8 which are related to the V(2, 1)-cycle with different numbers of grids (*ngrid*). They show examples of error behaviour which persists through all the time steps.

Rectangular basins

Let us consider a closed rectangular basin of constant depth $h = 1$ m whose length in the x-direction is 6000 m and whose width in the y-direction is 3300 m. The basin is crossed centrally along the x-direction by a deeper channel which is closed at the right end and open at the left end. The channel width is 300 m and its depth, measured from the undisturbed water surface, is $h = 6$ m. At the basin boundaries the normal velocity is set equal to zero everywhere except at the left end of the channel, where a tide of 12 h period and 0·4 m amplitude is specified. We divided the flow domain into 48×24 finite difference cells of equal sides $\Delta x = \Delta y = 150$ m. We assumed that the tidal circulation begins with all water masses at rest.

From the numerical results provided by the multigrid semi-implicit finite different method, we can note first that this method allows the tide to be correctly simulated. Results are visualized in Plates 1 and 2, which show symmetric water surface elevation and velocity coherent with physical symmetry.

Besides visualizations of the computed solution, we also visualized convergence histories of the method, from which we can derive several observations. From Figure 6 it appears that the multigrid algorithms are better than the one-grid algorithm. However, it also seems that adding the fourth grid is of no advantage for this problem, since the $ngrid = 4$ curve substantially overlaps the $ngrid = 3$ curve. Therefore we considered the same test problem but physically larger, 50,100 m × 26,100 m, and we used up to five grids. From the convergence histories analysed for all the time steps, one of which is shown in Figure 7, we derived the following results.

1. All the multigrid algorithms are faster than the one-grid solver.
2. The method speed-up increases on passing from one to five grids, so adding both the fourth and fifth grids becomes useful.
3. The convergence rate turns out optimal, and almost unchanged, on passing from two to five grids up to a precision of $0{\cdot}5 \times 10^{-2}$, and beyond this value it progressively improves with the increment of the level number.
4. The multigrid algorithm appears to be particularly convenient for physically large problems.

We also compared the multigrid method with the one-grid method using the preconditioned conjugate gradient method (PCGM),[5,6] which is a very efficient technique for solving the pentadiagonal system (5). As we can see in Figure 8, the multigrid algorithm turns out to be the best up to a precision of about $0{\cdot}5 \times 10^{-4}$, where the two convergence histories meet each other, and very fast up to $0{\cdot}5 \times 10^{-2}$. These results, besides being very satisfactory, show that promising new algorithmic combinations could be able to improve the obtained interesting performances.

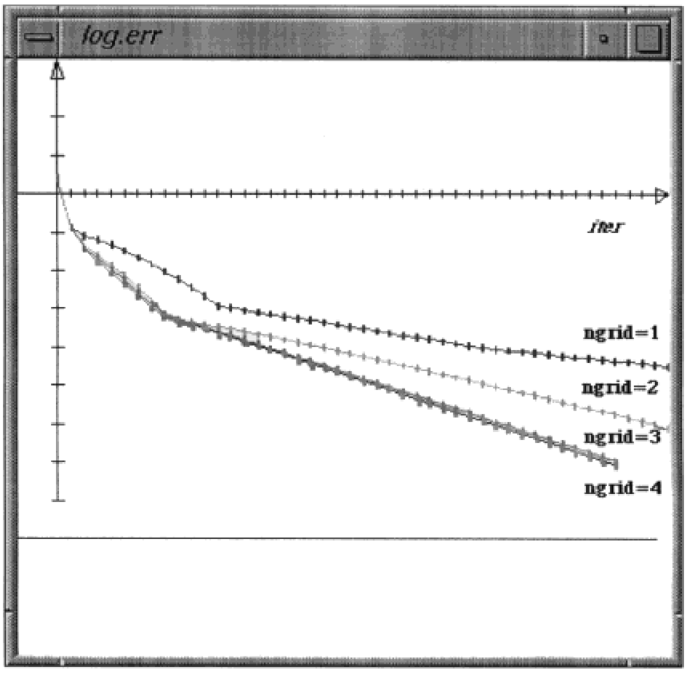

Figure 6. Small rectangular basin: convergence history at $t_k = 1$ h

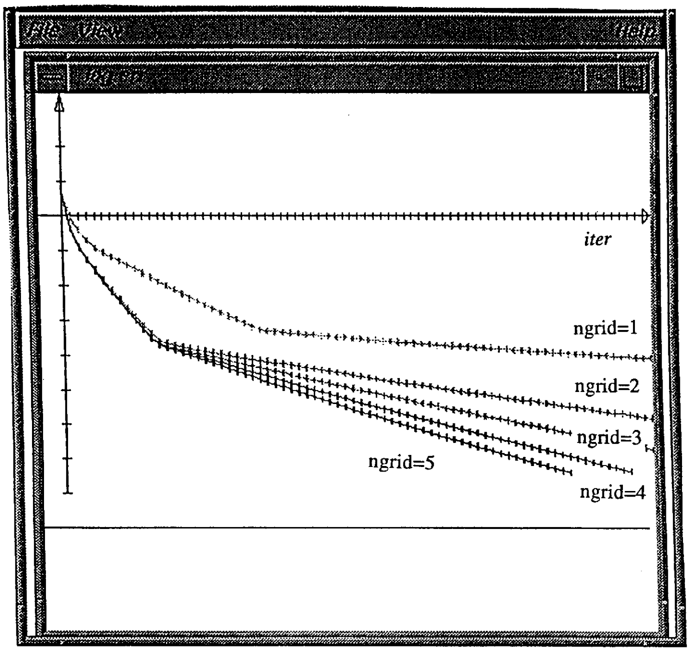

Figure 7. Large rectangular basin: convergence history at $t_k = 1\,\mathrm{h}$

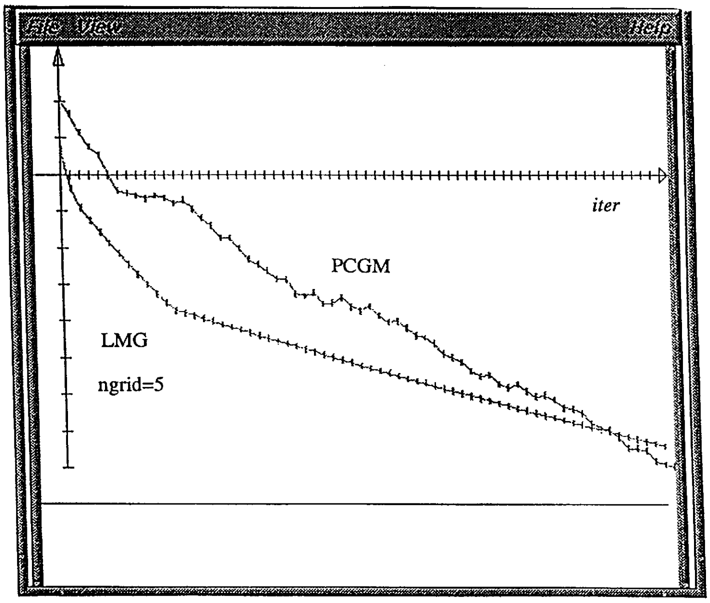

Figure 8. Large rectangular basin: convergence history at $t_k = 1\,\mathrm{h}$

San Pablo Bay

We have also applied the multigrid algorithm for the simulation of the real complex water circulation in San Pablo Bay. We solve this problem on a grid of 46×46 cells of equal sides $\Delta x = \Delta y = 500$ m with time step $\Delta t = 0{\cdot}25$ h. At the bay boundaries two tides of equal period, 12 h, but with different amplitude are specified at the cells covering the open bay sides, while the normal velocity is set equal to zero elsewhere.

Plates 3 and 4 show the resulting velocity field and water surface elevation obtained at $t_k = 6$ and 12 h. These results also confirm the effectiveness and flexibility of the present multigrid semi-implicit finite difference method.

8. CONCLUSIONS

A multigrid semi-implicit finite difference algorithm on staggered grids has been presented for the two-dimensional shallow water equations. By applying this method, we obtained interesting experimental results which allow the following observations to be derived.

1. The method provides significant multigrid acceleration with respect to the same one-grid algorithm.
2. It appears to be particularly convenient for physically large problems.
3. The multigrid acceleration turns out optimal, and almost unchanged, up to a rather interesting precision, and beyond that it increases with the number of grids.
4. The multigrid algorithm seems to be significantly faster than the fast preconditioned conjugate gradient one-grid method.

These results, besides being very satisfactory, show that promising new algorithmic combinations (e.g. suitable coupling of the multigrid calculation with the preconditioned conjugate gradient method) could be able to improve the obtained interesting performances.

ACKNOWLEDGEMENT

The authors are grateful to Professor V. Casulli for useful comments and suggestions.

REFERENCES

1. A. Brandt, 'Multi-level adaptive solutions to boundary-value problems', *Math. Comput.*, **31**, 333–354 (1977).
2. R. M. Spitaleri, 'A multigrid method for grid generation with line-spacing control', *Appl. Math. Comput.* **40**, 125–134 (1990).
3. V. Casulli, 'Semi-implicit finite difference methods for the two dimensional shallow water equations', *J. Comput. Phys.*, **86**, 56–74 (1990).
4. J. Häuser, H. G. Paap, D. Eppel and S. Sengupta, 'Solution of the shallow water equations', *Int. J. Numer. Methods Fluids*, **6**, 530–539 (1986).
5. W. Hackbush and U. Trottenberg (eds), *Multigrid Methods*, LNM Vol. 960, Springer, Berlin, 1982.
6. W. Hackbusch, *Multi-Grid Methods and Applications*, Springer, Berlin 1985.

