INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 6, 587–591 (1986)

ON PARALLEL COMPUTATION OF THREE-DIMENSIONAL WIND-DRIVEN CIRCULATION

BOGUSŁAW JACKOWSKI

Institute of Hydroengineering, Polish Academy of Sciences, ul. Cystersów 11, 80-953 Gdańsk, Poland.

PAWEŁ PACZKOWSKI

Institute of Mathematics, Gdańsk University, ul. Wita Stwosza 57, 80-952 Gdańsk, Poland

INTRODUCTION

A significant speed-up in time-consuming computations of wind-driven circulations can be achieved by introducing parallelism into the numerical method, if possible. Such a speed-up is particularly needed in the case of three-dimensional models. This paper is intended as a case study, illustrating that even quite standard computational techniques (such as the finite element method) may lead, in a natural way, to parallel algorithms.

The idea of the model of wind-driven circulations that we consider goes back to Ekman,[1] Welander[2] and their successors. Originally, the calculations described here were made on a sequential machine[3] using the formulation of the model proposed by Kolodko *et al.*[4] Later on, however, an observation was made that the computations can be carried out in parallel. Now the authors believe that the parallel formulation of the algorithm is even more natural than the sequential one: it reflects the parallelism of the physical phenomenon and is closely related to the discrete formulation of the model. The proposed approach seems to be general enough to be applied to other models which possess a similar mathematical description (an evolutionary form of the governing equations). In particular, many of the existing non-stationary models of three-dimensional wind-driven circulations can be treated in this way.

We start with the formulation of the governing equations. The discrete form of the equations obtained by application of the finite element method is shown to be convenient for the parallel treatment and the parallel algorithm is formulated. Finally, its complexity is briefly discussed.

FORMULATION OF THE MODEL

The situation we consider is shown in Figure 1.

The effect of wind action on the behaviour of the water velocity field and the surface denivellation is assumed to have the following mathematical description:

Governing equations

$$\partial_t \mathbf{u} = \partial_z(\nu \partial_z \mathbf{u}) + g\nabla h \qquad \text{(momentum equation)}, \tag{1}$$

$$\partial_t h = \nabla \cdot \mathbf{U} \qquad \text{(continuity equation)}, \tag{2}$$

0271–2091/86/090587–05$05.00
© 1986 by John Wiley & Sons, Ltd.
Received 28 June 1985
Revised 1 March 1986

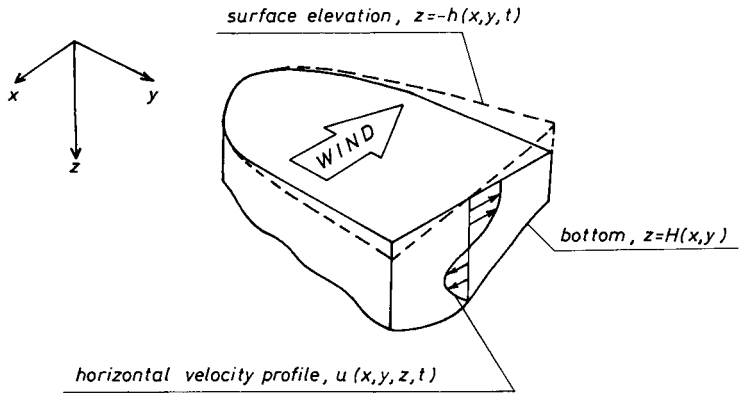

Figure 1. General situation

Boundary conditions

$$z = 0: \quad \nu \partial_z \mathbf{u} = -\mathbf{s},$$
$$z = H: \quad \mathbf{u} = \mathbf{0},$$
$$\text{at shore: } \mathbf{U}_{\text{normal}} = 0,$$

where

$\partial_x, \partial_y, \partial_z, \partial_t$	partial differential operators,
$\nabla = (\partial_x, \partial_y)$	two dimensional gradient operator,
$h(x, y)$	denivellation of the surface,
$\mathbf{u}(x, y, z, t)$	$= (u^x, u^y)$ horizontal velocity,
g	gravity acceleration,
$\nu(x, y, z, t)$	semi-empirical coefficient of turbulent viscosity,
$\mathbf{s}(x, y, t)$	semi-empirical function, representing the wind action on the water surface,
$\mathbf{U}(x, y, t)$	$= \int_0^H \mathbf{u}(x, y, z, t)\, dz$ horizontal transport,
$H(x, y)$	depth ($\lvert h \rvert \ll H$).

For the sake of simplicity Coriolis force is neglected here, but it can be easily introduced into the model, if needed. The model can also be adjusted to the situation in which h is comparable to H, without spoiling the feasibility of the parallel treatment.

DISCRETIZATION OF THE MODEL

The finite element method seems to be suitable for spatial discretization of the model, since shapes of water reservoirs are usually very irregular. The grid is constructed as follows: the surface is triangulated and a uniform vertical structure of the grid is imposed in the interior of the reservoir. This leads to the partition of the reservoir into prisms, as shown in Figure 2.

In order to obtain a simple discrete model the simplest shape functions (linear over edges) and the node-based integration scheme over the elements are applied. An explicit finite difference scheme is used for time discretization.

Let q denote the number of surface nodes and p the number of layers. The discrete quantities to be computed are h_j, the surface denivellation at the jth node, and $\mathbf{u}_j = [\mathbf{u}_{j1}, \ldots, \mathbf{u}_{jp}]$, the velocity

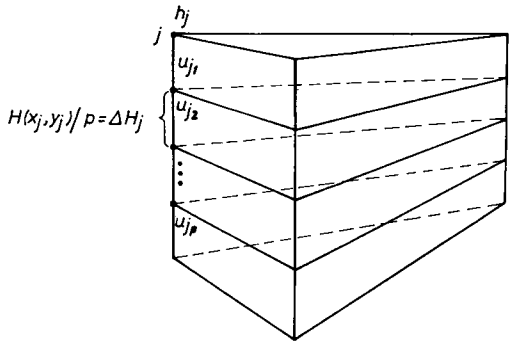

Figure 2. Partition into the finite elements

profile below the jth surface node, for $j = 1, \ldots, q$, where, in turn, $\mathbf{u}_{jk}$ are two-dimensional vectors of horizontal velocities $\mathbf{u}_{jk} = (u^x_{jk}, u^y_{jk})$.

The resulting discrete model has the following form (the details of the derivation of these formulae are omitted here since they are not relevant to the problem of parallelism): for $j = 1, \ldots, q$

$$\mathbf{M}_j(\mathbf{u}_j^{n+1} - \mathbf{u}_j^n)/\delta = \mathbf{Q}_j^n \mathbf{u}_j^n + \sum_{k \in E_j} \mathbf{R}_{jk} h_k^n + \mathbf{f}_j^n, \tag{3}$$

$$N_j(h_j^{n+1} - h_j^n)/\delta = \sum_{k \in E_j} \mathbf{R}_{kj} \cdot \mathbf{u}_k^n, \tag{4}$$

where

δ	the time step,
E_j	the set of surface nodes belonging to the same surface element as node j, i.e. the nodes that are neighbours of node j, including node j itself,
$\mathbf{f}_j^n$	the p-dimensional, time dependent vector corresponding to wind extortion (i.e. to the boundary condition on the surface),
$\mathbf{M}_j, N_j$	a diagonal $p \times p$ matrix and a scalar, respectively, (both determined by the sizes of the elements that are neighbours of the jth vertical),
$\mathbf{Q}_j^n$	a time dependent tridiagonal $p \times p$ matrix, corresponding to the operator $\partial_z(v\partial_z)$ in equation (1),
$\mathbf{R}_{jk}, \mathbf{R}_{kj}$	p-dimensional vectors, corresponding to the operator ∇,
n	the number of the current time step.

It is particularly important for the formulation of the parallel algorithm that the surface denivellation and the velocity profile at a given vertical depend only on the surface denivellations and the velocity profiles at the neighbouring verticals.

PARALLEL ALGORITHM

The set of equations (3), (4) could be treated parallely in a general way as, for example, in References 5 or 6. In this particular case, however, we can propose an algorithm which seems to reflect better the evolution of the physical process.

The computations can be done by p concurrent processes. With each vertical j a process Π_j is associated which has to compute the velocity profile $\mathbf{u}_j^{n+1}$ and the surface denivellation h_j^{n+1} for consecutive time steps $n = 0, 1, 2, \ldots$ We assume that processes corresponding to the

neighbouring verticals can communicate with each other and that the data associated with the jth vertical ($\mathbf{M}_j, N_j, \mathbf{Q}_j^n, \mathbf{u}_j^n$ $\mathbf{f}_j^n$, h_j^n and $\mathbf{R}_{jk}$, for $k \in E_j$) are local for the process Π_j.

Algorithm

Every process $\Pi_j, j = 1, \ldots, p$, proceeds as follows:

(i) Π_j communicates with each of the processes Π_k for $k \in E_j$.
Π_j sends the values of h_j^n and $\mathbf{R}_{jk} \cdot \mathbf{u}_j^n$ to Π_k and, in return, receives the values of h_k^n and $\mathbf{R}_{kj} \cdot \mathbf{u}_k^n$ computed by Π_k.

(ii) Π_j computes $\mathbf{u}_j^{n+1}$ and h_j^{n+1} according to equations (3) and (4), using the information received in step (i).

The applied communication pattern enables the processors to perform the computations with a possibly small amount of local data. Therefore the processes require only a small local memory, the more so as the vectors $\mathbf{R}_{jk}$ and $\mathbf{f}_j^n$ can be, in fact, represented by single numbers.

COMPLEXITY OF THE ALGORITHM

Under the assumption that the processes proceed at the same speed, $O(q) + C$ can be taken as the measure of complexity of each time step, where C is the cost of communication. In the case of optimal scheduling of information exchange C is $O(1)$. On the other hand $O(pq)$ arithmetical operations are necessary to compute $\mathbf{u}_j^{n+1}, h_j^{n+1}$ (for $j = 1, \ldots, q$) sequentially from the equations (3) and (4). This means that the speed-up obtained by the use of parallelism is by a factor p. Since p is the number of concurrently working processes one may argue that the exploitation of parallelism is near to optimal.

REMARKS

It is worthy of note that the concurrency of the presented algorithm is not due to a clever manipulation of data. On the contrary, as has already been mentioned, the algorithm simply reflects the parallelism of the physical phenomenon. Therefore this approach can be regarded as even more natural than the traditional, sequential one.

The presented description of the algorithm is general enough to be adjusted to different specific models of concurrent computations. Instead of explicit communication, global common resources can be used by synchronous processes. The computations can also be carried out by a network of processors with direct communication between processors corresponding to neighbouring verticals. If the interconnection pattern of processors is fixed (like in ILLIAC IV) and does not provide direct connections between processors corresponding to neighbouring verticals, more complex broadcast strategies should be used. This, however, might increase the amount of time spent on communication.

Though parallel computers are not yet in everyday use, access to such machines is becoming easier. Therefore the authors believe that the parallelism in numerical methods is important not only as a theoretically interesting feature but also as an approach which is more and more of practical importance.

REFERENCES

1. V. M. Ekman, 'Über die horizontalzirkulation winderzeugter Meeresströmungen', *Ark. Math. Astr. Och. Fys.*, **17**, (1923).
2. P. Welander, 'Wind action on a shallow sea: some generalizations of Ekman's theory', *Tellus*, **9**, 45–59 (1957).
3. B. Jackowski, 'Computation of wind-driven circulations in shallow lakes', *Proc. 4th International Conference on Finite Elements in Water Resources*, Hannover, 1982, pp. 7.3–7.12.
4. J. Kołodko, B. Jackowski and E. Wojtowicz, *Technical Report* C_1*-3/1982*, Institute of Hydroengineering of PAS, Gdansk (in Polish).
5. D. Heller, 'A survey of parallel algorithms in numerical linear algebra', *SIAM Review*, **20**, 740 (1978).
6. Y. Wallach, *Alternating Sequential/Parallel Processing*, LNCS 127, Springer-Verlag, Berlin, Heidelberg, New York, 1982.