JOURNAL OF COMPUTATIONAL PHYSICS 104, 347–360 (1993)

Performance of the Capacitance Matrix Method for Solving Helmholtz-Type Equations in Ocean Modelling

E. BLAYO AND C. LE PROVOST

Institut de Mécanique de Grenoble, BP 53X, 38041 Grenoble Cedex, France

Received August 1, 1990; revised March 31, 1992

A Capacitance Matrix Method has been implemented in a quasi-geostrophic eddy-resolving general circulation model, for applications to real oceanic basins. This method allows the use of very fast direct solvers to treat problems on irregular regions. The aim of the present study is to test the performance of this method in terms of CPU requirements and accuracy. © 1993 Academic Press, Inc.

1. INTRODUCTION

Scientists have been modelling the atmosphere for 40 years, and have now achieved an acceptable level of realism. Similar progress in oceanography has been slowed down by several factors, among them the fact that (i) the horizontal scales of the energetic features, the mesoscale eddies, measure only a few hundreds of kilometers; (ii) the typical time scales of the oceanic circulations are years to decades for the dynamic aspects, and decades to centuries for the thermodynamic processes; and (iii) the oceans are confined in basins with quasi-meridional boundaries in contrast with the spherical continuity of the atmosphere.

As in other scientific domains, progress in numerical ocean modelling is thus closely linked to the speed of computers. This limitation has provided a strong stimulus for the development of methods to improve the efficiency of the numerical technique used in terms of cpu requirements.

Several classes of ocean models include the solution of Helmholtz equations. In the Primitive Equation Ocean General Circulation Models (PEOGCM) developed since the late 1960s (Bryan [3]), the rigid lid approximation, which was introduced to allow larger time steps by removing the very fast external gravity waves, leads to the solution of an elliptic equation for the two dimensional barotropic mass transport streamfunction. The Quasi-Geostrophic Eddy resolving General Circulation Models (QGEGCM), which have been intensively used to investigate the physics of the ocean mesoscale eddies and their crucial importance for the global circulation (Holland [11]), basically rely upon the solution of Helmholtz problems.

Usually, in PEOGCM, the techniques used to solve the finite difference Helmholtz equation are the Sucessive Over Relaxation method (SOR) or the Preconditioned Conjugate Gradient (PCG), which are rather time consuming. On the other hand, fast solvers have been used for QGEGCM restricted to rectangular basins for process studies. This is, for example, the case of Hockney's FACR (Fourier Analysis—Cyclic Reduction) method, the efficiency of which has been displayed in many applications. However, these fast solvers can be used only on rectangular basins.

The Capacitance Matrix Method (so called because of its first use by Hockney [10] for a potential calculation problem) allows one to extend the applicability of these very efficient solvers to irregular computational domains. Many different formulations of this method (further denoted by CMM) have been proposed: Cummins and Mysak [5] use an analytical presentation, Buzbee *et al.* [4] and Golub and Meurant [8] describe its matricial aspect, while Proskurowski and Widlund [19] show the close relationship between this method and the classical potential theory. Here, we will present a finite difference formulation of the CMM implemented in a QGEGC multilayer model. After a description of the CPU and memory space requirements of this method, we will compare its accuracy with that of the FACR method. We will demonstrate that these levels of accuracy are very similar, and then show the significant speed-up of the CMM over classical iterative methods.

2. DESCRIPTION OF THE NUMERICAL MODEL

Our model is of the type which was first presented by Holland [11]. For midlatitude oceanic circulations, equations may be derived from the Navier–Stokes equations as

0021-9991/93 $5.00
Copyright © 1993 by Academic Press, Inc.
All rights of reproduction in any form reserved.

a function of the Rossby number $\varepsilon = U/f_0 L$, where U is a horizontal velocity scale (a few cm/s), L is a horizontal length scale (100 to 1000 km), and f_0 is the Coriolis parameter at the mean latitude of the basin. The β-plane approximation is used, so that the Coriolis parameter is written as $f(y) \simeq f_0 + \beta y$, where β is constant. At the order ε, one obtains the quasi-geostrophic equations, which can be written (Pedlosky [18]) as the conservation of potential vorticity,

$$\frac{D}{Dt}\left[\nabla^2\psi + \frac{\partial}{\partial z}\left(\frac{1}{S}\frac{\partial\psi}{\partial z}\right) + \beta y\right] = F + T, \tag{1}$$

where ψ is the 3-dimensional streamfunction,

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \frac{\partial\psi}{\partial x}\frac{\partial}{\partial y} - \frac{\partial\psi}{\partial y}\frac{\partial}{\partial x} = \frac{\partial}{\partial t} + J(\psi, .),$$

∇^2 is the horizontal Laplacian operator, and $S(z) = (g/f_0^2)((d\ln\rho)/dz)$ represents the vertical variation of density (g is the gravitational acceleration, and ρ is the density). The model is driven by the vertical component of the wind stress curl, T, and dissipation F is included through a bottom friction and a horizontal friction.

The vertical discretisation divides the ocean into N layers of constant densities $\rho_1, ..., \rho_N$. At rest, the thicknesses of the layers are $H_1, ..., H_N$ (see Fig. 1).

Equation (1) may be written

$$\frac{\partial}{\partial t}[\nabla^2\Psi - A\Psi] = B \tag{2}$$

with

$$\Psi = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_N \end{pmatrix}$$

the streamfunctions in the different layers

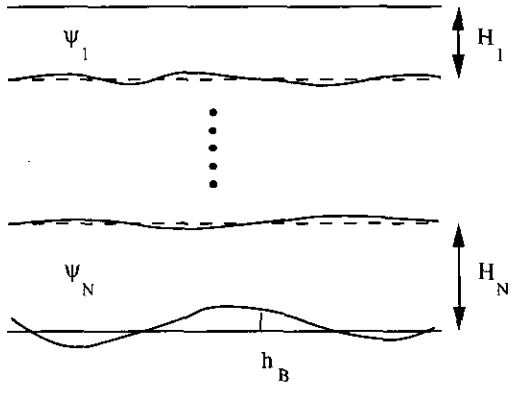

FIG. 1. Stratification of the model.

- A the tridiagonal matrix whose nonzero row elements are

$$\frac{-f_0^2}{H_k g'_{k-1/2}}, \quad \left(\frac{f_0^2}{H_k g'_{k-1/2}} + \frac{f_0^2}{H_k g'_{k+1/2}}\right), \quad \frac{-f_0^2}{H_k g'_{k+1/2}}$$

with $g'_{k+1/2} = g(\rho_{k+1} - \rho_k)/\rho_0$ for $0 < k < N$, and $g'_{1/2} = g'_{N+1/2} = \infty$ (ρ_0 is a reference density)

- $$B = \begin{pmatrix} \vdots \\ J\left(\nabla^2\psi_k + \beta y + \dfrac{f_0}{H_k}(h_{k+1/2} - h_{k-1/2}), \psi_k\right) + F_k + T_k \\ \vdots \end{pmatrix}$$

with $h_{k+1/2} = f_0(\psi_{k+1} - \psi_k)/g'_{k+1/2}$ for $0 < k < N$, $h_{1/2} = 0$ and $h_{N+1/2} = h_B$ (height of the bottom topography).

A may be diagonalized (Bengtsson and Temperton [2]): $A = P^{-1}DP$ where D is a diagonal matrix of positive eigenvalues. The first element of D is 0, because A is singular. Equation (2) becomes

$$\frac{\partial}{\partial t}[\nabla^2(PP^{-1}\Psi) - P\,DP^{-1}\psi] = PP^{-1}B,$$

i.e., the decoupled system

$$\frac{\partial}{\partial t}[\nabla^2\Phi - D\Phi] = P^{-1}B, \tag{3}$$

where $\Phi = P^{-1}\Psi$. The first eigenvector of Φ, corresponding to the eigenvalue 0, is called the barotropic mode, and the other eigenvectors are called baroclinic modes.

The numerical integration of the system (3) is performed using a finite difference method with the same space step in both horizontal directions. The time scheme is a "leap-frog" (second-order accurate scheme). The occasional insertion of a step made by a two-level scheme (Euler scheme) suppresses the development of a computational mode in the solution. For a detailed discussion of the leap-frog properties, see, for example, Mesinger and Arakawa [16, Chap. 2].

At each time step, we must then solve N Helmholtz equations

$$\nabla^2\Phi^{n+1} - D\Phi^{n+1} = 2\Delta t(P^{-1}B)^{(n-1,n)} + \nabla^2\Phi^{n-1} - D\Phi^{n-1},$$

where the superscripts denote time levels, and Δt is the time increment. The superscript $(n-1,n)$ over $P^{-1}B$ indicates that, to ensure numerical stability, the dissipative terms

(bottom and horizontal friction) are computed at time level $n-1$ and the other terms at time level n.

The Laplacian operator is evaluated with the standard five-point scheme, while the Arakawa [1] procedure is used to compute the Jacobian operator.

3. THE CAPACITANCE MATRIX METHOD

In this section, L is a linear operator and the dot superscript indicates the interior of a domain (e.g., $\dot{\Omega} = \Omega - \partial\Omega$).

We must find the function u such that $Lu = f$ on a domain $\dot{\Omega}$, with the boundary condition $u = b$, a given function on $\partial\Omega$ (see Fig. 2). Let us assume that an efficient algorithm able to solve this type of equation on a rectangular domain is at our disposal. We define a rectangular region Ω_1 in which Ω is embedded, and denote by $\partial\Omega_1$ its boundary. We also define $\partial\Omega'$ by $\partial\Omega' = \partial\Omega \cap \dot{\Omega}_1$. In a grid representation, the elements of $\partial\Omega'$ are called "irregular points".

We can then define f_1 on Ω_1 and b_1 on $\partial\Omega_1$ by

$$f_1 = \begin{cases} f & \text{on } \dot{\Omega}, \\ 0 & \text{elsewhere;} \end{cases} \quad \text{and} \quad b_1 = \begin{cases} b & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ 0 & \text{elsewhere.} \end{cases}$$

Let U_1 be the solution of

$$\begin{aligned} LU_1 &= f_1 \qquad \text{on } \dot{\Omega}_1 \\ U_1 &= b_1 \qquad \text{on } \partial\Omega_1. \end{aligned}$$

We can compute U_1 because this is a problem on a rectangular domain.

For each irregular point I_i, $i = 1, ..., M$, we can also compute the Green's function G_i defined by

$$\begin{aligned} LG_i &= \delta(I_i) \qquad \text{on } \dot{\Omega}_1 \\ G_i &= 0 \qquad \text{on } \partial\Omega_1, \end{aligned}$$

where $\delta(I_i)$ is the Dirac function the value of which is 1 for I_i and 0 anywhere else.

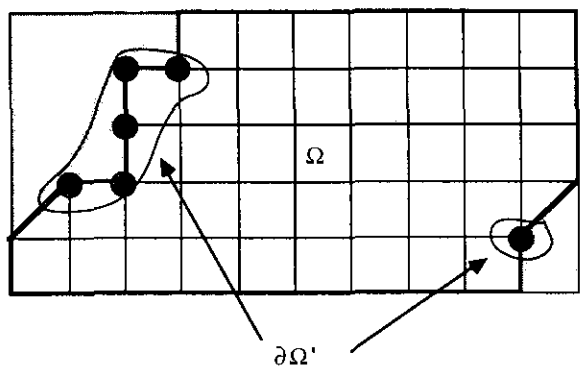

FIG. 2. Domain definition.

Any function U_2 defined by

$$U_2 = U_1 + \sum_{i=1}^{M} \alpha_i G_i$$

is then a solution of:

$$\begin{aligned} LU_2 &= \begin{cases} \alpha_i \text{ on } I_i, & i = 1, ..., M, \text{ since } f_1(I_i) = 0, \\ f_1 \text{ on } \dot{\Omega}_1 - \partial\Omega'; \end{cases} \\ U_2 &= b_1 \text{ on } \partial\Omega_1. \end{aligned}$$

To solve the initial problem, we only have to choose correctly the values of $\alpha_1, ..., \alpha_M$, so that the system $U_2(I_i) = b(I_i)$, $i = 1, ..., M$, is satisfied. This can be written with matrix notation:

$$\begin{pmatrix} G_1(I_1) & \cdots & G_M(I_1) \\ \vdots & \vdots & \vdots \\ G_1(I_M) & \cdots & G_M(I_M) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_M \end{pmatrix} = \begin{pmatrix} b(I_1) - U_1(I_1) \\ \vdots \\ b(I_M) - U_1(I_M) \end{pmatrix}$$

The inverse of the left matrix G is called the *capacitance matrix* (henceforth denoted by C).

Then, we can give an algorithm for the CMM:

- *Preliminary phase*
  For $i = 1$ to M:

$$\text{Solve} \begin{cases} LG_i = \delta(I_i) & \text{on } \dot{\Omega}_1, \\ G_i = 0 & \text{on } \partial\Omega_1. \end{cases}$$

  If possible, store G_i on Ω_1, else only store $G_i(I_1), ..., G_i(I_M)$.
  Compute C
- *At each time step*

$$\text{Solve} \begin{cases} LU_1 = f_1 & \text{on } \dot{\Omega}_1 \\ U_1 = b_1 & \text{on } \partial\Omega_1 \end{cases}$$

  (Note that f_1 and b_1 are time dependent functions).

Compute

$$\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_M \end{pmatrix} = C \begin{pmatrix} b(I_1) - U_1(I_1) \\ \vdots \\ b(I_M) - U_1(I_M) \end{pmatrix}.$$

Compute $U_2 = U_1 + \alpha_1 G_1 + \cdots \alpha_M G_M$ if the Green's functions have been stored, otherwise solve

$$\begin{aligned} LU_2 &= \begin{cases} \alpha_i \text{ on } L_i & i = 1, ..., M \\ f_1 \text{ on } \dot{\Omega}_1 - \partial\Omega' \end{cases} \\ U_2 &= b_1 \text{ on } \partial\Omega_1. \end{aligned}$$

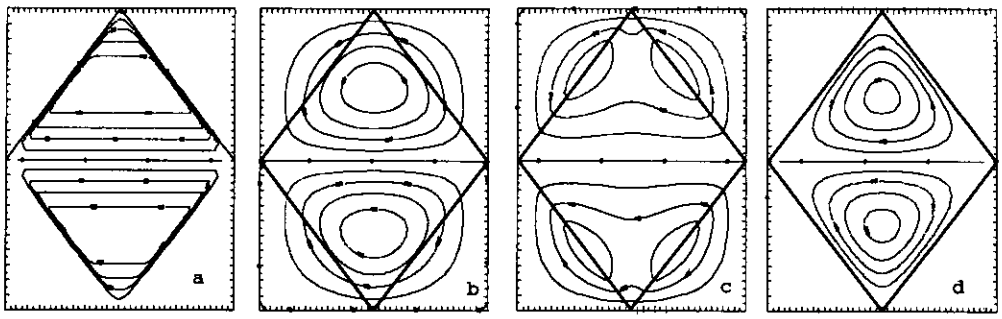

FIG. 3. The different steps of the Capacitance Matrix Method: (a) forcing, (b) solution U_1 on the rectangular domain Ω_1, (c) Green's functions contribution, and (d) final solution U_2 on the rhombus domain Ω.

This algorithm is illustrated in Fig. 3, for the case of a rhombus domain, with a forcing function $f(x, y) = \sin 2\pi y/L_y$, where L_y is the y-dimension of the domain. As previously described, two different implementations are possible, either with the storage of the Green's functions and only one call of the solver, or with no storage of the Green's functions but two calls of the solver. The first implementation is, for obvious computation time considerations, the more attractive. Unfortunately, in a realistic oceanic context, a basin will be modelled with a 10-km horizontal resolution, and six or eight levels on the vertical. The number of irregular points is then approximately between 500 to 1000 (for each level), while the storage of *one* Green's function would require about 1.5 megawords of memory.

Therefore, the only usable version of the algorithm is the second one. Compared with the solver on a rectangular domain, it requires extra memory space to store the N capacitance matrices (one for each level). Note here that they are symmetric, and the storage space needed is consequently equal to $N \times M \times (M+1)/2$ space units. One must also generate some auxilliary arrays relative to the irregular boundaries, whose size is negligible. The computation time is much more affected by this method, and two extra costs must be considered. In the preliminary phase, the computation of the Green's functions and the capacitance matrices requires a cpu time equivalent to $N \times M$ calls to the solver on a rectangular domain, and N inversions of an $M \times M$ matrix. On the basis of the discretization parameters given before, with a model time increment of the order of one hour, this cpu time can be evaluated as being equivalent to about 3 to 6 months (model time) of simulation: this represents only a small fraction of extra cost, if one remembers that typical experiments in our oceanic context have to be carried out over about 20 years to reach a statistically steady state. On the other hand, at each time step, the solution of the equation requires about twice the time it would require on a rectangular domain, because two calls to the rectangular domain solver become necessary. This increases the *total* computation time of our code by about 50%.

4. CMM'S ACCURACY

Before using the CMM in our general circulation model for physical investigations, we wanted to be sure that the computation remains accurate, and to compare this accuracy with the one obtained when using the solver in its optimal conditions, i.e., on rectangular domains. We propose here an analytical approach to this problem, and some numerical experiments on test functions and in an oceanic context.

4.1. *Analytical Approach*

In the following, we will denote the computed values with a tilde, and the exact numerical values without a tilde.

In one time step, the second stage of the CMM is the solution of the problem

$$\begin{aligned} LU_2 &= \begin{cases} \alpha_i \text{ on } I_i, & i = 1, ..., M, \\ f_1 \text{ on } \mathring{\Omega}_1 - \partial\Omega'; \end{cases} \\ U_2 &= b_1 \text{ on } \partial\Omega_1, \end{aligned} \tag{4}$$

but the numerical solver computes $\tilde{U}_2$, the numerical solution of

$$\begin{aligned} L\tilde{U}_2 &= \begin{cases} \tilde{\alpha}_i \text{ on } I_i, & i = 1, ..., M, \\ f_1 \text{ on } \mathring{\Omega}_1 - \partial\Omega'; \end{cases} \\ \tilde{U}_2 &= b_1 \text{ on } \partial\Omega_1. \end{aligned} \tag{5}$$

Consequently, the error $\tilde{U}_2 - U_2$ has two different origins: the error e_s of the numerical solver on the rectangular domain Ω_1, and the propagation of the error $\tilde{\alpha}_i - \alpha_i$ $(i = 1, ..., M)$ provided by the first stage of the CMM.

While the perturbation on the right-hand side of Eq. (4) is nonzero only on a few grid points (M), one can assume that, if this perturbation $\tilde{\alpha}_i - \alpha_i$ $(i = 1, ..., M)$ is small enough, it will induce only a small error on U_2.

Let us try to estimate $\|\tilde{\alpha}-\alpha\|$, where

$$\tilde{\alpha}=\begin{pmatrix}\tilde{\alpha}_1\\ \vdots\\ \tilde{\alpha}_M\end{pmatrix},\ \alpha=\begin{pmatrix}\alpha_1\\ \vdots\\ \alpha_M\end{pmatrix}$$

and $\|\cdot\|$ is a matrix norm. The computed matrix $\tilde{G}$ is equal to $G+e_G$, where e_G is the error due to the solver on a rectangular domain. We have then

$$\tilde{C}=(G+e_G)^{-1}+e_I,$$

where e_I is the error induced by the inversion method.

We can then apply to G the easily verified equality:

$$(A+\Delta A)^{-1}=A^{-1}-(A+\Delta A)^{-1}\,\Delta A\,A^{-1} \tag{6}$$

where ΔA is a perturbation on a matrix A.

This implies $(G+e_G)^{-1}=C-(G+e_G)^{-1}e_GC$. So $\|(G+e_G)^{-1}-C\|=\|(G+e_G)^{-1}e_GC\|\leqslant\|(G+e_G)^{-1}\|\times\|e_G\|\,\|C\|$.

Relation (6) also involves

$$\frac{\|(G+e_G)^{-1}\|}{\|C\|}=\|I-(G+e_G)^{-1}e_G\|.$$

Since $\|(G+e_G)^{-1}e_G\|\ll 1$ in our computations, we can use the approximation $\|(G+e_G)^{-1}\|\simeq\|C\|$ to obtain

$$\|(G+e_G)^{-1}-C\|\leqslant\|C\|^2\,\|e_G\|$$

Since

$$\alpha=C\begin{pmatrix}b(I_1)-U_1(I_1)\\ \vdots\\ b(I_M)-U_1(I_M)\end{pmatrix}$$

and

$$\tilde{\alpha}=\tilde{C}\begin{pmatrix}b(I_1)-U_1(I_1)-e_1(I_1)\\ \vdots\\ b(I_M)-U_1(I_M)-e_1(I_M)\end{pmatrix},$$

where e_1 is the error on U_1 induced by the solver on the rectangular domain, we have

$$\tilde{\alpha}-\alpha\simeq(e_C+e_I)\begin{pmatrix}b(I_1)-U_1(I_1)\\ \vdots\\ b(I_M)-U_1(I_M)\end{pmatrix}-C\begin{pmatrix}e_1(I_1)\\ \vdots\\ e_1(I_M)\end{pmatrix},$$

where $e_C=(G+e_G)^{-1}-C$, and if we assume that the second order term

$$e_C\begin{pmatrix}e_1(I_1)\\ \vdots\\ e_1(I_M)\end{pmatrix}$$

is negligible.

This calculation leads to

$$\|\tilde{\alpha}-\alpha\|\leqslant(\|C\|^2\,\|e_G\|+\|e_I\|)\times\left\|\begin{matrix}b(I_1)-U_1(I_1)\\ \vdots\\ b(I_M)-U_1(I_M)\end{matrix}\right\|+\|C\|\left\|\begin{matrix}e_1(I_1)\\ \vdots\\ e_1(I_M)\end{matrix}\right\|.$$

If N_x and N_y are the numbers of grid points of Ω_1 in the x and y directions, we can note that $\|e_G\|$ and

$$\left\|\begin{matrix}e_1(I_1)\\ \vdots\\ e_1(I_M)\end{matrix}\right\|$$

are, respectively, of the order of $M/\sqrt{N_x\times N_y}\,\|e_s\|$ and $\sqrt{M}/\sqrt{N_x\times N_y}\,\|e_s\|$ (if we use, for example, the Frobenius norm $\|A\|=\sqrt{\sum_i\sum_j a_{ij}^2}$), where e_s is the error of the numerical solver on the rectangular domain.

Then, we have

$$\|\tilde{\alpha}-\alpha\|\leqslant\left(\frac{M}{\sqrt{N_x\times N_y}}\,\|C^2\|\,\|e_s\|+\|e_I\|\right)\times\left\|\begin{matrix}b(I_1)-U_1(I_1)\\ \vdots\\ b(I_M)-U_1(I_M)\end{matrix}\right\|+\frac{\sqrt{M}}{\sqrt{N_x\times N_y}}\,\|C\|\,\|e_s\|.$$

To ensure the smallness of $\|\tilde{\alpha}-\alpha\|$, an obvious condition is that $\|e_I\|\ll\|e_C\|$, i.e., that the inversion method used to compute C be very accurate. Then

$$\|\tilde{\alpha}-\alpha\|\leqslant\frac{\sqrt{M}\,\|C\|}{\sqrt{N_x\times N_y}}\times\left(1+\sqrt{M}\,\|C\|\left\|\begin{matrix}b(I_1)-U_1(I_1)\\ \vdots\\ b(I_M)-U_1(I_M)\end{matrix}\right\|\right)\|e_s\|.$$

It is easily demonstrated (see, for instance, [13]) that, if Δy is a perturbation of the right-hand side of the equation $Ax=y$, the perturbation Δx of the solution satisfies

$$\frac{\|\Delta x\|}{\|x\|}\leqslant\mathrm{Cond}(A)\,\frac{\|\Delta y\|}{\|y\|},$$

where $\mathrm{Cond}(A)$ is the condition number of A, defined by $\|A\|\,\|A^{-1}\|$. So the perturbation e_2 of U_2 due to the propagation of the error $\tilde{\alpha}_i - \alpha_i$ $(i = 1, ..., M)$ satisfies

$$\frac{\|e_2\|}{\|U_2\|} \leqslant \mathrm{Cond}(L)\,\frac{\|\tilde{\alpha}-\alpha\|}{\|f_\alpha\|},$$

where f_α is the right-hand side of equation (4). So

$$\|e_2\| \leqslant \frac{\sqrt{M}\,\|C\|}{\sqrt{N_x \times N_y}} \times \left(1 + \sqrt{M}\,\|C\| \left\| \begin{matrix} b(I_1) - U_1(I_1) \\ \vdots \\ b(I_M) - U_1(I_M) \end{matrix} \right\| \right) \times \frac{\|U_2\|}{\|f_\alpha\|}\,\mathrm{Cond}(L)\,\|e_s\|.$$

We have numerically computed this upper bound in some realistic cases. L was a Poisson or Helmholtz operator, and the solver on the rectangular domain was an FACR method. The worst empirical result that we obtained on a CRAY-2 computer (64 bits precision) was

$$\|e_2\| \leqslant 2\ 10^{-7}\,Cond(L)\,\|e_s\|.$$

This coefficient is of course pessimistic, because we have always considered that the errors were maximum, and were always added without ever cancelling each other. In any case, although $\mathrm{Cond}(L)$ is a very large number, this upper bound seems small enough to ensure the smallness of e_2. For example, the condition number of the Laplacian operator discretised on a regular mesh with $N_x = N_y = 200$ to 300 (typical size in realistic oceanic simulations) is of the order of 10^5 to 10^6.

Using this value of 10^6 in the last inequality, one obtains

$$\|e_2\| \leqslant 0.2\,\|e_s\|.$$

Then, if we add the systematic error of the solver on the rectangular domain, we obtain an inequality concerning the *total error of the solution*:

$$\|\tilde{U}_2 - U_2\| \leqslant 1.2\,\|e_s\|.$$

This result suggests that a solver, when its use is extended from problems on rectangular domains to problems on irregular regions by using the CMM, maintains the same accuracy, provided that the inversion method used to compute the capacitance matrix is very accurate.

4.2. *Some Simple Numerical Tests*

Although there is no way to compare the bahaviour of the CMM and the basic FACR solver in exactly the same configuration, we performed some simple tests with close computational configurations on the Poisson equation example (the FACR algorithm combines a decomposition into Fourier series of the unknowns with a cyclic odd–even reduction. For a description of this method, see, for instance, Hockney [10] or Swarztrauber [21]).

For different values of N, we considered a square $N \times N$ domain Ω embedded in a larger domain Ω_1. Then, we successively generated $2N-1$ "true" solutions of the form

$$u_k(i,j) = \sin\frac{2\pi k i}{N}\sin\frac{2\pi k j}{N} \quad \text{on } \mathring{\Omega}, \qquad u_k = 0 \quad \text{on } \partial\Omega, \qquad k = 1, ..., 2N-1 \tag{7}$$

and computed $f_k = \Delta u_k$ using double-precision, to minimize round-off error. We then used the FACR method on Ω and the CMM on Ω_1 to solve the Poisson equations $\Delta\tilde{u}_k = f_k$, and compared the computed solutions $\tilde{u}_k$ to the original ones u_k (see Fig. 4). The mean error (in absolute value) $1/N^2 \sum_{i,j} |\tilde{u}_k - u_k|$ is shown in Fig. 5.

We can note that:

- for both methods, the error globally increases with the forcing wavelength kL/N (where L is the basin length);
- for forcing wavelengths less than $L/2$, the two methods lead to very similar errors;
- for forcing wavelengths greater than $L/2$, the CMM can be either better or worse than the FACR method, depending on N and on the forcing wavelength. We found no rule available to predict which method is best for given N and k.

Globally, our numerous computations (more than 500) seem to respect the relations

$$\frac{1}{2} \leqslant \frac{e_{\mathrm{cmm}}}{e_{\mathrm{facr}}} \leqslant 2 \quad \text{for forcing wavelengths less than } L/2$$

$$\frac{1}{5} \leqslant \frac{e_{\mathrm{cmm}}}{e_{\mathrm{facr}}} \leqslant 5 \quad \text{for forcing wavelengths greater than } L/2.$$

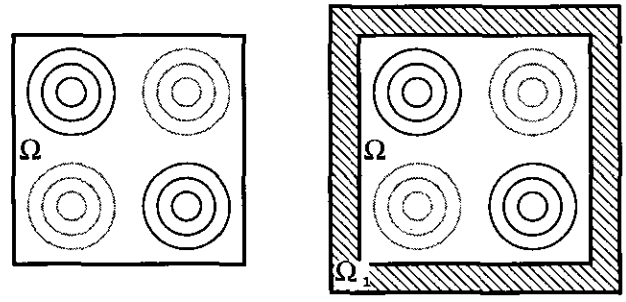

FIG. 4. Computation Configuration.

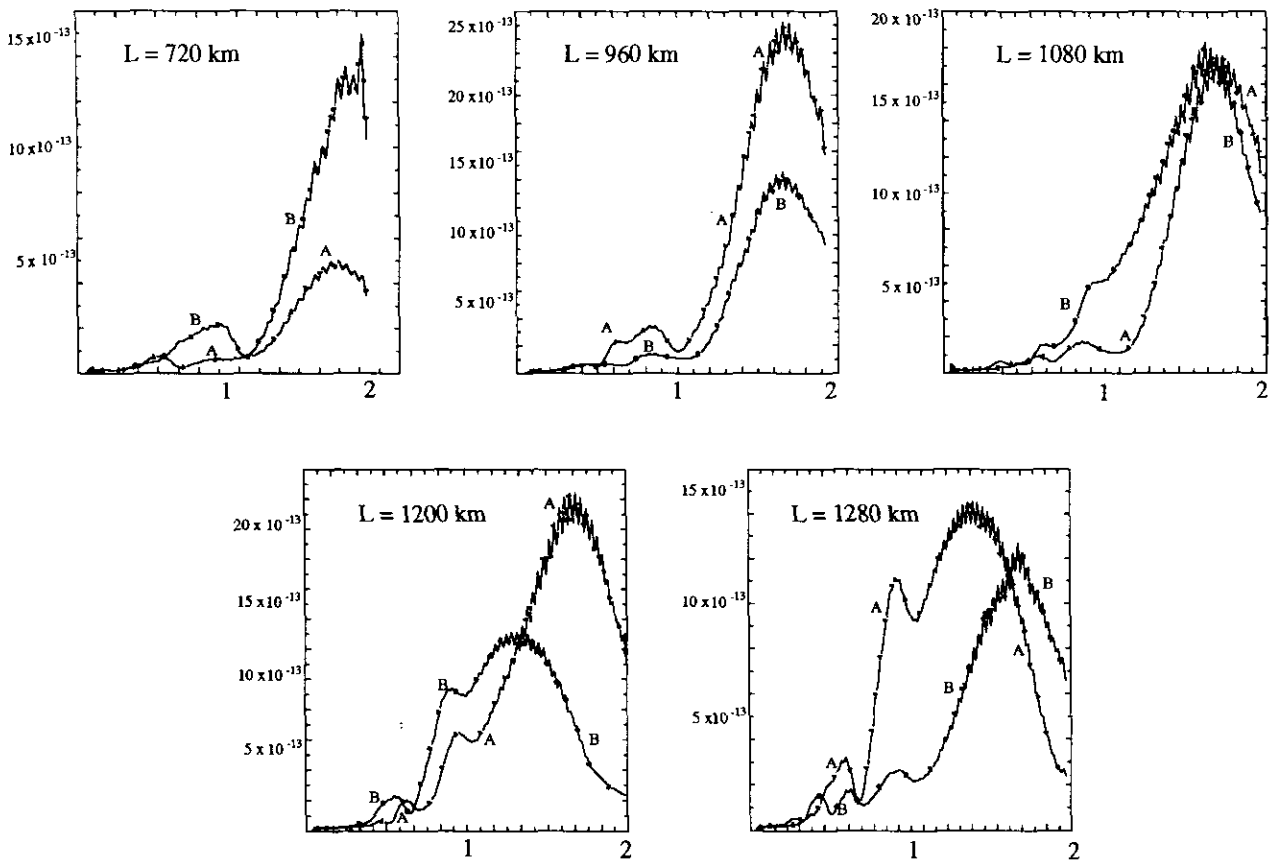

FIG. 5. Mean error (in absolute value) for the numerical solutions of Poisson's equation on an $L \times L$ square domain, computed for different values of L with the FACR method (curves A) or with the CMM method (curves B). The forcing functions u_k are defined by (7). The x-axis represents the ratio k/N.

This result confirms the findings of Section 4.1, which indicated that the accuracies of the two methods were of the same order. Furthermore, the observed differences will probably be less in a realistic oceanic context, because the natural forcing is a combination of several wavelengths.

4.3. *Some Tests in an Oceanic Context*

To confirm the hypothesis that the FACR and CMM methods have similar accuracy levels in "realistic" simulations, two experiments were performed, using our numerical model described in Section 2, with two different ranges of parameters, corresponding to different types of model responses. We simulated the circulations in a square oceanic basin, a very schematic domain indeed, but widely used in the literature for 10 years to investigate the physical properties of the wind-driven ocean circulation (cf. Holland [11]). For each computation, we compared the solutions obtained with the two different configurations described in the previous section: the FACR method applied to Ω and CMM to Ω_1 (see Fig. 4). We will see that the numerical error provided by the CMM was shown to be larger in one experiment that the one generated by the FACR method, but smaller in the other experiment.

The initial state was rest, and the computation was performed until statistically steady circulation was reached. The ocean was divided into $N = 3$ vertical layers, with respective thicknesses 300, 700, and 4000 m. The reduced gravities g' were chosen as $g'_{3/2} = 0.036$ and $g'_{5/2} = 0.016\ \mathrm{m \cdot s^{-2}}$, which correspond to internal Rossby radius of deformation $R_1 = 44.8$ and $R_2 = 25.3$ km (R_1 and R_2 are the reciprocal of the square root of the eigenvalues corresponding to the baroclinic modes, see Section 2, and are length scales of the baroclinic motions). This stratification corresponds approximately to the mean midlatitude ocean stratification in the North Atlantic (Emery, Lee, and Maagard [6]). The bottom dissipation is modelled by the expression $C_b \nabla^2 \psi_N$, with $C_b = 10^{-7}\ \mathrm{s^{-1}}$ (this is a frequently used value: Holland [11], Cummins and Mysak [5]). The lateral dissipation parameterization in each layer may be $A_2 \nabla^4 \psi$ or $A_4 \nabla^6 \psi$. The boundary conditions may be slip ($\partial \psi / \partial n = 0$) or no-slip ($\partial \mathbf{V} / \partial n = 0$), where n is the vector locally normal to the boundary and $\mathbf{V}$ is the horizontal velocity vector. The wind forcing is sinusoidal:

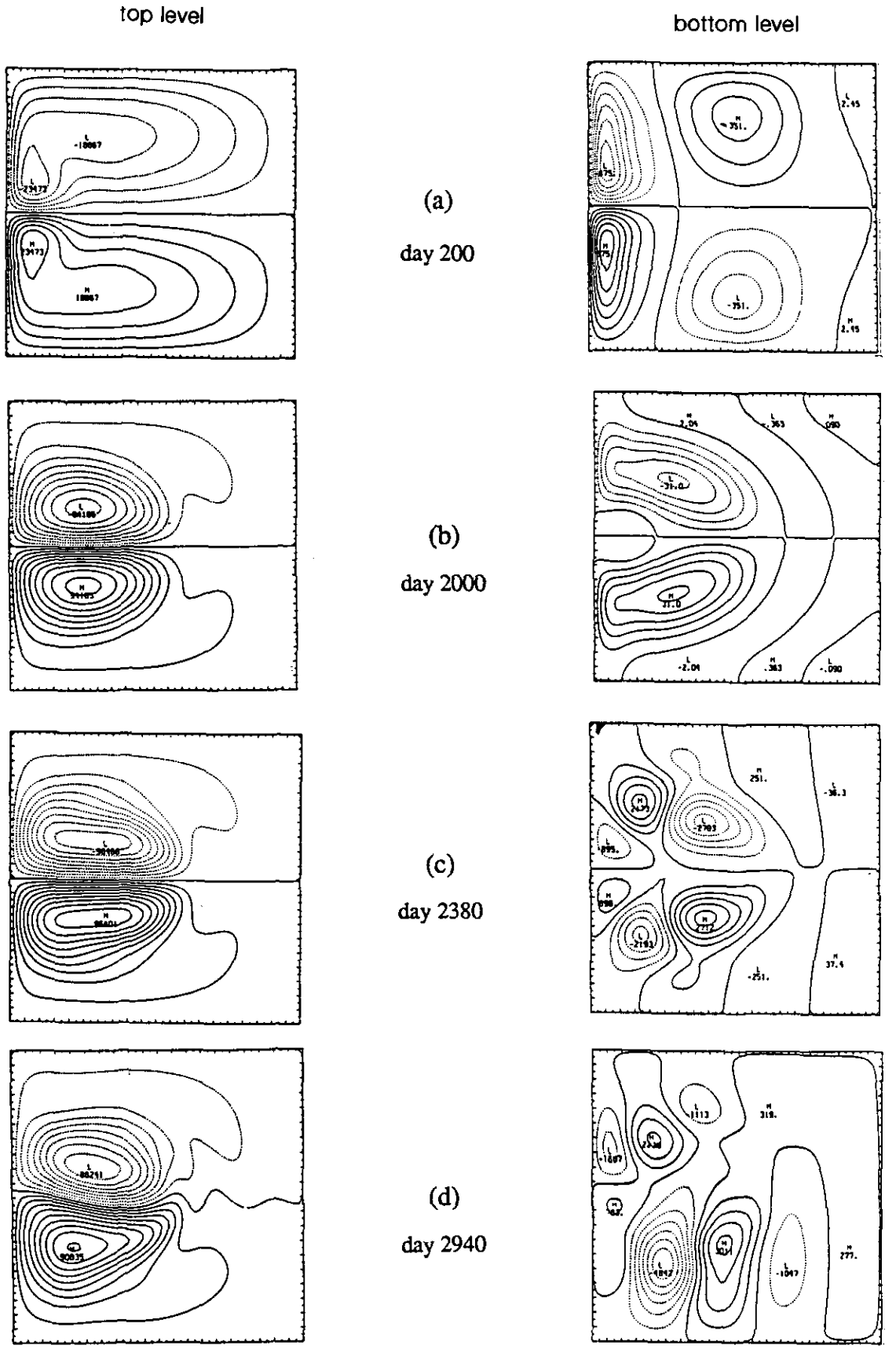

FIG. 6. Instantaneous streamfunction patterns of the top and bottom levels for FRD experiment (using FACR). (a) Formation of the inertial jet and eastward propagation of Rossby waves. (b) The jet is stable and symmetric, the Rossby waves are damped. (c) The jet is still stable and symmetric, but the westward return flow is destabilized. One can observe a slight asymmetry in the numerical values of the bottom streamfunctions. (d) Destabilization of the jet, and total asymmetry of the bottom layer.

$T(y) = ((2\pi\tau_0)/L)\sin(2\pi y)/L$, and leads to a double gyre classical circulation. The value of τ_0 was chosen so that the western boundary current, of typical width $\delta_I = 40$ km (Le Provost and Verron [14]), is sufficiently resolved by three points within the boundary layer. The parameter values are given in Table I.

Such simulations of oceanic circulations are typically divided into several phases. Basically, the water masses are put in motion by the wind force acting at the surface. After a short initial phase where the entire layer of water is set in motion, the circulations are accelerated mainly in the upper layers. Transient Rossby waves are generated which propagate westward, due to the local gradient of the Coriolis force. This leads to an intensification of the current along the western boundary. The important inertial character of the flows (in agreement with reality) results in the for-

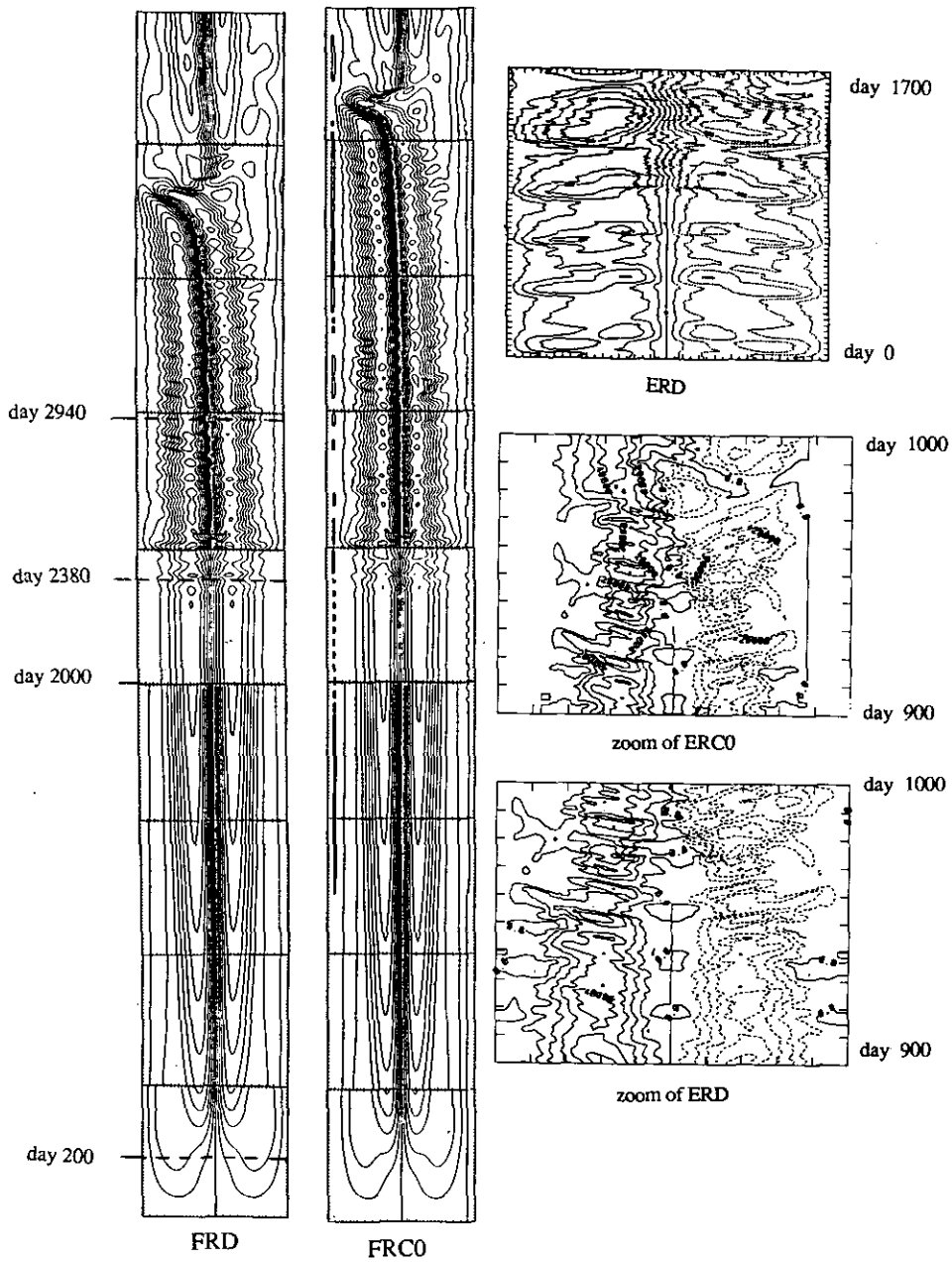

FIG. 7. Time history diagram of the streamfunction in the upper layer along the section $x_c = 0.25L$ for the different experiments. Asymmetry in the computed values appears later in the ERD simulation (using FACR) than in the ERC0 simulation (using CMM), but earlier in the FRD simulation (using FACR) than in the FRC0 simulation (using CMM). On the left: we have pointed out on the y-axis of the FRD and FRC0 diagrams the instants corresponding to the snapshots of Fig. 6. (The apparent discontinuities in the FRD and FRC0 diagrams are only due to changes of the contour interval.) On the right: time history diagram for the total ERD simulation, and zooms around the instant of destabilization of the jet for ERD and ERC0 simulations.

TABLE I

Parameters of the Experiments

	L	Δx	Computation configuration	τ_0	Lateral friction	Boundary condition
ERD	720 km	20 km	FACR	$2.67 \times 10^{-5} N \cdot m^{-2}$	$A_4 = 10^9 m^4 s^{-1}$	Slip
ERC0	—	—	CMM	—	—	—
FRD	1280 km	—	FACR	$4 \times 10^{-5} N \cdot m^{-2}$	$A_2 = 10^3 m^2 s^{-2}$	—
FRC0	—	—	CMM	—	—	—

mation of an inertial eastward jet, along the zero wind stress curl isoline. These features are clearly illustrated in Fig. 6a, where the instantaneous streamfunctions of the upper and lower layers are displayed for the FRD experiment: the western boundary layers, the mid intertial jet, very strong in the upper layer, but also existing in the lower layer, and the presence of the Rossby waves easily observable in the lower layer (on the graphs, H indicates high, and L indicates low). The transport and the penetration of the jet into the basin increase with time in the upper layers, while the Rossby waves and the circulation in the lower layers are damped, and strong intertial recirculations take place on the two sides of the jet (see Fig. 6b at time $t = 2000$ days). Then the western return flows destabilize: signature of the meanders clearly appears on the lower layer through intense highs and lows which drift westward with the upper layer return flows (see Fig. 6c). At time $t = 2380$ days, it should be noted that the solutions are still antisymmetrical, although a slight asymmetry can be observed in the numerical values of the bottom layer streamfunctions. These instabilities develop and finally destabilize the jet itself which starts slow oscillation toward the south (Fig. 6d). This meandering finally leads to the formation of eddies, which then drift toward the west and are damped in the western boundary layer. After each eddy pinching, the jet comes back to a quasi-symmetrical picture.

While the model equations and the wind forcing are symmetric with respect to the line $y = L/2$ (L is the basin length), the asymmetry in the computational solution is due only to the numerical error of the solver. So, the more accurate the solver, the later the asymmetry will appear.

A way to display these destabilisation events is to draw the time history of the streamfunction along the longitude $x = 0.25L$ and to observe the symmetry of the solution. This is illustrated in Fig. 7. On the FRD picture, the dates of the different sequences displayed on Fig. 6 are indicated, which help to understand the content of Fig. 7. At time 220 days, the extrema of the streamfunctions at $x_c = 0.25L$ are still situated at $y = 0.25L$ and $y = 0.75L$. These extrema move toward the center of the jet ($y = 0.5L$), when it overcomes the abscissa x_c, and the isolines increase in number with the increasing intensity of the flow, until instabilities appear on the two sides of the jet ($t = 2380$ days). Then, the jet itself starts wiggling and undergoes a southward meander. It then returns to the center of the domain.

These two methods are of comparable accuracy. If we refer to the time necessary for the jet to be destabilized in the experiments, as an index of the impact of computational inaccuracy on the solution, we observe that this destabilization occurs 20 days later in ERD (using FACR) than in ERC0 (using CMM), and about 200 days later in FRC0 (using CMM) than in FRD (using FACR). This is proof that there is no systematic advantage of one method over the other. The FACR method is better in the first experiment, the CMM in the second case.

Statistics computed from these experiments show that the global dynamic is unchanged, and that mean quantities computed over the stationary phase are almost the same, with the two solvers. The same physical processes are present in corresponding simulations, but not exactly at the same time steps because of the different propagation of the errors of the two solvers.

These results are in agreement with our previous predictions, and confirm the similarity in accuracy of the two methods in realistic applications.

4.4. *Grid Representation of the Boundary*

An unavoidable problem when implementing the CMM in a numerical model is the choice of the grid representation

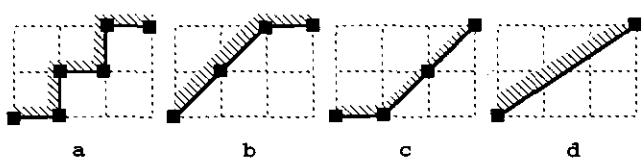

FIG. 8. Grid representations of a boundary.

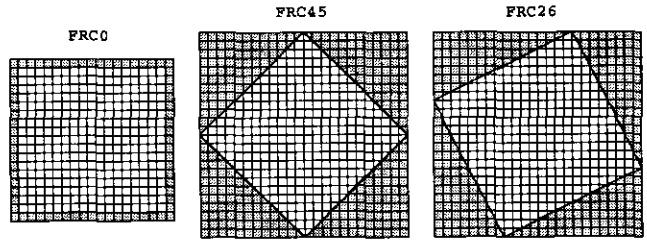

FIG. 9. FRC0, FRC45, and FRC26 simulations.

of the irregular boundaries. In fact, as illustrated in Fig. 8, there are several ways to discretize the same portion of boundary.

The results will therefore depend on where the grid is put and also on the way the boundary conditions are discretized. We tried to evaluate the consequences of the different possible choices by performing two numerical simulations, with the same parameters as in the FRD and FRC0 simulations described above, but with other computation configurations. The basin was turned 45° to the left in the first simulation (FRC45), and about 26° (arctan 0.5) to the left in the second simulation (FRC26) (see Fig. 9). In this simple case of linear boundaries, the curvature of the coastlines is zero, and this leads to the boundary conditions: $\psi = 0$ and $\Delta\psi = 0$. They are taken into account in the computation of the laplacian and jacobian operators on the grid points located near the boundaries. The numerical schemes for both operators on those grid points are five-point schemes, of the second order of accuracy for each simulation.

Despite this fact, we can state, without giving a detailed description of these simulations, that our experiments lead to the conclusion that the introduction of oblique segments with an orientation other than 45° induces a faster propagation of the computational error. The reason for this may be that the resolution on the coastlines is worse in these cases (there are less gridpoints on the boundary). So, a good way to discretize the physical boundary of the domain is to use horizontal, vertical and 45° oblique segments.

5. COMPARISON OF CPU PERFORMANCE WITH OTHER METHODS

The other methods often used in ocean modelling for solving Helmholtz equations are Successive-Over-Relaxation (SOR) and Preconditioned Conjugate Gradient (PCG)—with different possible preconditioning techniques. Several authors have compared the efficiency of these methods and of FACR. Numerous numerical tests have been performed, on different computers (vectorized or not), and it is well known that the FACR method is much more efficient in terms of CPU time than the other methods for the solution in rectangular domains (see, for instance, the recent paper of Fujishiro *et al.* [7]).

This result has been already pointed out by Buzbee *et al.* [4] in the case of a domain with a complex geometry. They found that the CMM applied with a FACR solver was more accurate and was at least five times faster than the iterative methods SOR, SLOR (successive line overrelaxation), and ADI (Peaceman-Rachford alternating direction implicit iteration). This paper is somewhat old, however, and improvements in numerical and computational techniques led us to retest the efficiency of the CMM. Madec *et al.* [15] published the results of a comparison between SOR and PCG, used to solve the Poisson equation in a general ocean circulation model. This paper shows that the PCG (using diagonal preconditioner) is typically ten times faster than SOR in "academic" cases, and twice as fast in real cases.

We performed similar tests on the same CRAY-2 vector computer, and compared our results with the curve of CPU time versus accuracy obtained by Madec. The domain size was 110×73 gridpoints. The accuracy of our solutions was of the order of 10^{-8}, and the CPU time required for the computation of a solution was 0.06 s, while Madec's PCG needed 0.7 s (CPU ratio $\simeq 11.5$) to reach the same precision, and 0.2 s (CPU ratio $\simeq 3$) to reach an accuracy of only 10^{-3}.

These tests confirm the efficiency of the CMM, when used together with a FACR solver, compared with classical iterative solvers, for solving Helmholtz-type equations encountered in ocean modelling.

Another recent and very efficient technique is the multigrid method. Theoretically, it is the fastest method, because the asymptotic number of operations required for a solution is proportional to the number of gridpoints (Stüben and Trottenberg [20]). However, for computations in a realistic oceanic context, difficulties appear, due to the different representations on the different grids of the very irregular shape of the boundaries. This point was illustrated by Jensen [12], who obtained somewhat disappointing CPU time results when introducing realistic coastlines in his ocean model based on multigrid technique. He attributed these difficulties to the method of handling the irregular boundaries on coarse grids. An alternative to this problem, if the coastlines are smooth enough, is to use orthogonal curvilinear horizontal coordinates (Ohring [17], Haidvogel *et al.* [9]). The multigrid solver is then used as in a rectangular domain, and its efficiency is maximum. One can then expect a substantial improvement in terms of CPU time with regard to the CMM. This is precisely the way a new oceanic primitive equations model SPEM (Haidvogel *et al.* [9]) was initially conceived. But, due to the very complex shape of the oceanic coastlines, this model is now evolving toward a solution still based on a multigrid solver with boundary-fitted coordinates, but associated with a CMM to deal with the irregular boundaries.

6. A NOTE ABOUT PREDICTING THE COMPUTATIONAL ERROR OF THE MODEL

The simulations described in Section 4.3 enabled us to compare the evolutions of two computed solutions of the same problem. We will now use these observations to model and predict the evolution in the precision of a computed solution compared with the unknown *exact* numerical solu-

tion. A study of our code led us to model the statistical distribution of the error induced by the solver with a Gaussian centered distribution $\mathcal{N}(0, \sigma^2)$ (σ^2 is the variance), and to model the effect of all other calculations with an amplification factor δ, whose value must be only slightly larger than 1. If we assume that the initial solution is exact, the error after one time step will follow the distribution $\mathcal{N}(0, \sigma^2)$. After the second time step, the error's distribution will be $\mathcal{N}(0, \sigma^2) + \delta\mathcal{N}(0, \sigma^2) = \mathcal{N}(0, (1+\delta^2)\sigma^2)$.

Continuing with these calculations, we see that the error after i time steps will be distributed according to $\mathcal{N}(0, (1+\delta^2+\cdots+\delta^{2(i-1)})\sigma^2) = \mathcal{N}(0, (\delta^{2i}-1)/(\delta-1)\sigma^2)$.

We will denote by σ^2_{facr} and σ^2_{cmm} respectively, the variances of the error after one time step for the model in a square domain Ω and the model in a larger domain Ω_1. According to our forecast, the difference between the solutions of the two models after i time steps should be as follows:

$$\mathcal{N}\left(0, \frac{\delta^{2i}-1}{\delta-1}\sigma^2_{\text{facr}}\right) + \mathcal{N}\left(0, \frac{\delta^{2i}-1}{\delta-1}\sigma^2_{\text{cmm}}\right) = \mathcal{N}\left(0, \frac{\delta^{2i}-1}{\delta-1}(\sigma^2_{\text{facr}} + \sigma^2_{\text{cmm}})\right).$$

Let us define e_i to be the mean difference in absolute value after i times steps, and $\varepsilon_{\text{facr}}$ and ε_{cmm} the mean errors in absolute value after one step for the models on Ω and Ω_1. We must have

$$e_i = \sqrt{\frac{\delta^{2i}-1}{\delta-1}(\varepsilon^2_{\text{facr}} + \varepsilon^2_{\text{cmm}})},$$

i.e., $\log e_i \simeq i \log \delta - 0.5 \log(\delta-1) + 0.5 \log(\varepsilon^2_{\text{facr}} + \varepsilon^2_{\text{cmm}})$, assuming that $\delta^{2i} \gg 1$ for large values of i. Then, the curve $\log e_i = f(i)$ must be linear.

For each pair of FRD–FRC0 and ERD–ERC0 experiments, we choose an initial field for the stream functions, performed the integration of the model equations in the two computation configurations during a few thousands of time steps, and computed the corresponding curves $\log e_i = f(i)$ (Fig. 10). As can be seen, the agreement with our forecast seems quite good. Because of the size of the domains and the different inertial characteristics of the flows, the typical time scales of the two experiments are quite different, and this explains the difference between the propagation time of the computational error in the two cases. Using the advection time scale $T = L/U$, where U is the mean velocity in the middle of the jet, and estimating U from the experiments as $0.65\ \text{m} \cdot \text{s}^{-1}$ for FRD–FRC0 and as $1.15\ \text{m} \cdot \text{s}^{-1}$ for ERD–ERC0, one obtains respectively $T = 273$ time steps and $T = 87$ time steps. Thus, by non-dimensionalizing the time scale of Fig. 10 with these values, one obtains very similar pictures for both cases.

While the exponential increase of the computational error is nothing new, we are able in these cases to *quantitatively* evaluate the error.

In fact, for any particular simulation, one can easily evaluate $\varepsilon_{\text{facr}}$ or ε_{cmm}, by performing tests identical to those presented in Section 4.2 and only replacing the test functions u_k by solutions computed during the simulation.

For example, in the FRD and FRC0 simulations, we obtained $\varepsilon_{\text{facr}} \simeq 810^{-9}$ and $\varepsilon_{\text{cmm}} \simeq 510^{-9}$. The mean error in absolute value after i time steps can then be evaluated for the FRC0 simulation by $\sqrt{(1.0016^{2i}-1)/0.0016}\, 5 \times 10^{-9}$. So, if we define the value 10^{-2} as an acceptable limit for precision, the error remains within this limit until:

$$-2 \geqslant i \log 1.0016 - 0.5 \log 0.0016 + \log 5 \times 10^{-9}, \quad \text{i.e.,} \quad i \leqslant 7000.$$

Consequently, with the FRC0 simulation, we can expect any simulated process which lasts less than 7000 time steps to be well solved by the numerical code.

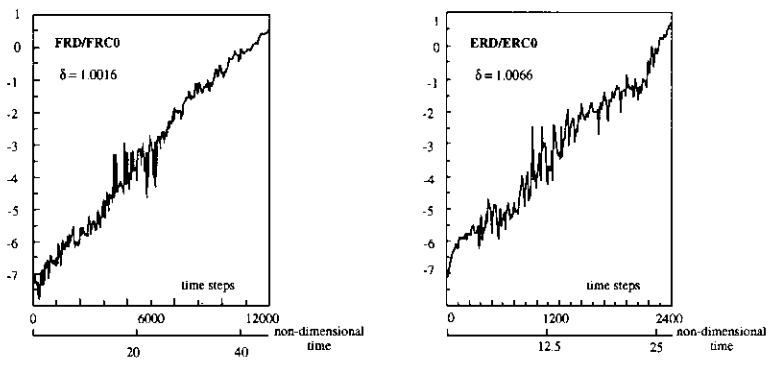

FIG. 10. Curves $\log e_i = f(i)$, where e_i is the mean difference, in absolute value, after i time steps between the two simulations ERD-ERC0 and FRD-FRC0; $\log \delta$ was evaluated by the least-squares method.

7. CONCLUSIONS

A Capacitance Matrix Method was implemented in a QGEGCM for applications to real oceanic basins. This method makes it possible to extend to irregular non rectangular domains the use of direct solvers of Helmholtz-type equations, which are particularly efficient in terms of CPU requiremets but restricted to rectangular domains. The oceanic domain Ω is embedded in a rectangular domain Ω_1 where Hockney's FACR method is applied to obtain a first guess at the solution at each time step. A correction term is then computed for each mode, using the capacitance matrices, and a second application of the FACR method leads to the exact solution on Ω. Provided that the geometry and the Helmholtz constant D do not change, the capacitance matrices need be determined for the N modes (layers) only once, and then stored.

Compared to the solver on a rectangular domain, the CMM requires extra memory space to store the N capacitance matrices: if M is the number of irredular grid points describing the coastlines, this storage space is of $N \times M \times (M+1)/2$ space units. The computation time requirements are increased by two extra costs: the computation of the Green's functions and the capacitance matrices, at the beginning of the computation, and solving twice on the rectangular domain Ω_1 at each time step. Typically, for oceanic numerical experiments which must be conducted over about 20 years, the total computation time is 50% greater than the time which would be necessary to solve the problem on the rectangular Ω_1 domain.

Despite the complexification of the computations, it has been analytically guessed, and experimentally verified, that the CMM does not introduce unacceptable inaccuracies. We observed on elementary numerical tests that (i) for both methods (direct FACR and CMM) the error globally increases with the wavelength of the forcing terms; (ii) for forcing wavelengths less than $L/2$ (L is the typical size of the basin), the two methods lead to very similar errors; and (iii) for larger forcing wavelengths, the CMM can be either better or worse than the FACR method, depending on the number of discretization points of the domain N, and the forcing wavelength k; it was impossible to find a rule to predict which method is the best for given N and k. However, it is of very little importance, since forcing with wavelength larger than L is physically of no interest.

Tests performed in an oceanic context have confirmed these conclusions: there is no systematic tendency for one method to be more accurate in terms of computational noise than the other. The major result of these investigations is that, globally, the dynamic processes typical of unsteady flows are reproduced in a similar way with both kinds of methods: mean statistics computed on long enough periods of unsteady simulations are the same, and the physical processes characteristic of these experiments are present in both simulations, but not exactly at the same time steps because of the difference in error propagation in the two solvers.

Some tests in realistic oceanic cases have showed that the CMM used together with a FACR solver is a much faster method than SOR or PCG.

Finally, starting from the hypothesis of a Gaussian centered curve for the statistical distribution of the error induced by the solver, and an amplification factor close to 1 for the effects of the other calculations at each time step, it was demonstrated that some estimates of the typical error propagation times can be obtained on the basis of elementary tests (as described in Section 4.2). For the kind of experiments relevant to the ocean, it was demonstrated that, starting from the same initial field, any simulated process of less than 20 nondimensional time steps is resolved identically (within 10^{-2} precision) using either method.

The Capacitance Matrix Method coupled with a FACR method thus appears to be a very efficient and reliable method for solving Helmholtz type equations in the context of ocean circulation modelling over domains with irregular coastlines.

ACKNOWLEDGMENTS

This work has been supported by a grant from the Institut Français de Recherche pour l'Exploitation de la Mer. Calculations were carried out using the numerical facilities of the Centre de Calcul Vectoriel pour la Recherche in Palaiseau.

REFERENCES

1. A. Arakawa, *J. Comput. Phys.* **1**, 119 (1966).
2. L. Bengtsson and C. Temperton, in *Numerical Methods used in Atmospheric Models*, Vol. II. Garp Publication Series 17, WMO/ICSU Joint Organizing Committee (1979).
3. K. Bryan, *J. Comput. Phys.* **4**, 347 (1969).
4. B. L. Buzbee, F. W. Dorr, J. A. George, and G. H. Golub, *SIAM J. Numer. Anal.* **8**, 722 (1971).
5. P. Cummins and L. Mysak, *J. Phys. Oceanogr.* **18**, 1261 (1988).
6. W. J. Emery, W. G. Lee, and L. Magaard, *J. Phys. Oceanogr.* **14**, 294 (1984).
7. I. Fujishiro, Y. Ikebe, A. Harashima, and M. Watanabe, *Comput. Fluids* **17**, 419 (1989).
8. G. H. Golub and G. Meurant, *Résolution numérique des grands systèmes linéaires* (Eyrolles, Paris, 1983).
9. D. B. Haidvogel, J. L. Wilkin, and R. Young, *J. Comput. Phys.* **94**, 151 (1991).
10. R. W. Hockney, *Math. Comput. Phys.* **9**, 135 (1970).
11. W. R. Holland, *J. Phys. Oceanogr.* **8**, 363 (1978).
12. T. G. Jensen, in *Advanced Physical Oceanographic Numerical Modelling*, edited by J. J. O'Brien NATO ASI Series, Vol. 186 (1986).
13. P. Lascaux and R. Theodor, *Analyse numérique matricielle appliquée à l'art de l'ingénieur* (Masson, Paris, 1986).
14. C. Le Provost and J. Verron, *Dyn. Atmos. Oceans* **11**, 175 (1987).

15. G. Madec, C. Rahier, and M. Chartier, *Ocean Modell.* **78**, 1 (1988).
16. F. Mesinger and A. Arakawa, in *Numerical Methods Used in Atmospheric Models*, Vol. I. Garp Publication series 17, WMO/ICSU Joint Organizing Committee (1976).
17. S. Ohring, *J. Comput. Phys.* **50**, 307 (1983).
18. J. Pedlosky, *Geophysical Fluid Dynamics* (Springer-Verlag, New York/Heidelberg/Berlin, 1979).
19. W. Proskurowski and O. Widlund, *Math. Comput.* **30**, 433 (1976).
20. K. Stüben and U. Trottenberg, in *Lecture Notes in Mathematics*, Vol. 960, edited by A. Dold and B. Eckmann (Springer, Berlin/Heidelberg/New York, 1982).
21. P. N. Swartrauber, *SIAM Rev.* **19**, 490 (1977).
22. C. Temperton, *J. Comput. Phys.* **31**, 1 (1979).