JOURNAL OF COMPUTATIONAL PHYSICS **141,** 93–111 (1998)
ARTICLE NO. CP985898

On the Schwarz Alternating Method for Oceanic Models on Parallel Computers

L. Debreu and E. Blayo

Project IDOPT, Laboratoire de Modélisation et Calcul, BP 53X, 38041 Grenoble Cedex, France
E-mail: Laurent.Debreu@imag.fr

Received January 29, 1997; revised December 8, 1997

In this paper we examine the application of a domain decomposition method to the solution of Poisson and Helmholtz equations often involved in the oceanic models. By studying a theoretical convergence rate of the method in the case of two subdomains (which can be generalized to some extent to the N-subdomains case), we can link some numerical properties of the method to physical features of the model ocean circulation. In particular, the peculiar role of the barotropic mode with respect to baroclinic modes is discussed. The tuning of the different parameters involved in a practical implementation of this Schwarz alternating method is also addressed for the case of a quasi-geostrophic oceanic model. Some CPU-timings are presented. On 16 processors of a CRAY T3D, the parallel code runs as fast as the sequential version does on a CRAY C90.

Key Words: domain decomposition methods; oceanography; Helmholtz equations.

1. INTRODUCTION

To achieve an acceptable level of realism, ocean modelling requires huge computational resources, both in terms of CPU and storage. This follows from the fact that the dynamical equations, which are similar to those in meteorology, must be integrated with a very high horizontal resolution (typically 10 km) to explicitly resolve the mesoscale eddies, which influence on the large-scale circulation is fundamental. Moreover, the typical time scales are 10 times longer for the ocean than the atmospheric ones and are as long as decades to centuries for thermodynamical processes. These spatial and temporal scales explain that ocean models are systematically pushing the power of available supercomputers to the limit. The development in the 1980s of vector computers led to a spectacular increase in the size of the problems that became tractable by ocean modellers. Moreover, there were generally no need for dramatic changes in the numerical codes to ensure an efficient use of such machines. The present tendency in the evolution of supercomputers is directed towards the development of massively parallel machines having hundreds, or even thousands, of processors and with

0021-9991/98 $25.00

a distributed memory. The power of these machines will almost certainly reach a teraflop (10^{12} floating point operations per second) in the forthcoming years, making possible the long-term integration of high resolution global ocean models (see, for example, [18] for further details).

However, the adaptation of present sequential numerical codes on these new machines is generally far from straightforward. To ensure an efficient use of parallel computers, two main constraints must guide this adaptation: the load-balancing (all the processors must execute approximatively the same amount of work) and the ratio communication over computation (communication times must be as low as possible with regard to computation). There are classically two main approaches to implement a code on parallel computers: direct parallelization and domain decomposition methods. In a direct parallelization approach, the structure of the original sequential code is not modified. The code is decomposed into a succession of elementary tasks, each task being executed in parallel using the power of the numerous processors. This can be done explicitly by using message passing instructions, or even implicitly on certain machines, which offers the facility of a virtual central memory, managed by the system and the compiler. So this approach requires relatively few changes of the code. However, global operations in the sequential algorithm often lead to global communications in the parallel code (for example, in a matrix product). Hence, direct parallelization generally does not provide a small ratio communication/computation and thus often yields relatively poor results. On the other hand, domain decomposition methods (DDM) avoid those global operations by creating local problems quasi-independent one to each other. The original problem is divided into identical local problems of smaller size, which are treated in parallel by the different processors. Most communications involved are local, between processors that are assigned to subdomains with common boundaries. Moreover, the load balancing can be adjusted by the choice of the sizes of the different subdomains (e.g., of equal sizes if the amount of work is uniform over the whole domain). Several mathematical and physical potential advantages of DDM can also be cited: possibility for local mesh refinements, improvement of the problem conditioning by reducing the problem scale, approximation of discontinuous coefficients by constant coefficients in each subdomain, etc. Ocean models are naturally well suited for running on massively parallel computers, since most of the computations are local. As a matter of fact, the numerical schemes for the spatial differential operators only require the knowledge of the state variables in the immediate vicinity of the computation point. Thus the amount of communications will be weak, provided the fact that the domain decomposition is 2D and not 3D, i.e. that the domain is divided horizontally into subdomains containing the whole vertical dimension (this comes from the fact that the vertical scale is much smaller than the horizontal scale and, thus, that a 3D decomposition would involve a lot of communications to compute the vertical schemes). The time discretization scheme is also of importance for parallelization purposes. Explicit time schemes are of course the simplest to manage for parallel computers, since no solution method is necessary and, hence, no communications, but the use of such scheme requires short time steps [3, 12]. On the other hand, implicit time schemes allow greater time steps, but imply the solution of elliptic equations at each time step [6, 9, 19]. Both approaches are equally used in the panel of general ocean circulation models presently developed and exploited by the oceanographic community.

When the solution of elliptic equations is necessary, the parallel versions of these numerical models generally make use of parallel versions of classical iterative solvers, like preconditioned conjugate gradient. To our knowledge, very few efforts have been made in the context of ocean modelling towards the use of solvers specifically designed for domain decomposition problems. We can only cite Guyon *et al.* [9], who use a dual Schur complement method to solve the equation for the barotropic streamfunction. In this context, this paper presents an application of the Schwarz alternating method to the solution of Poisson and Helmholtz equations often encountered in ocean circulation models. The method and its different possible parallel versions are presented in Section 2. Their theoretical convergence properties are discussed in Section 3, and some physical interpretations are given. Then, we present in Section 4 some numerical tests. Different strategies for the choice of the partitioning of the domain and of the first guess for the DDM are discussed, and the stability of the method with respect to the turbulence of the flow is addressed. Finally, timing results are given for implementations on both a CRAY T3D and a IBM SP1.

2. THE SCHWARZ ALTERNATING METHOD

The main idea of a domain decomposition method is simply to decompose the original problem on the domain Ω into several identical problems of smaller sizes on subdomains Ω_i. The problem is then to identify the conditions on the internal boundaries. There exist two main classes of DDM: the nonoverlapping methods, and the overlapping methods (subdomains have common zones). For a review of the DDMs, see, for example, [5].

The Schwarz algorithm, which is probably the more classical DDM, is an overlapping method. We will give here a brief description of this method. See [13] for further details.

Let Ω be a bounded open set of R^N $(N \leq 3)$. We consider a decomposition of Ω into M subdomains Ω_i with overlapping:

$$\bar{\Omega} = \bigcup_{i=1}^{M} \bar{\Omega}_i.$$

The internal boundaries are denoted by γ_{ij}, defined by $\gamma_{ij} = \partial\Omega_i \cap \Omega_j$. An example of such a decomposition into three subdomains is given in Fig. 1. We want to solve the problem

$$\begin{aligned} Lu &= f \quad \text{in } \Omega \\ u &= g \quad \text{on } \partial\Omega, \end{aligned}$$

where L is a linear second-order operator defined on R^N.

In the case of only two subdomains, the original form of the Schwarz method is the following alternating algorithm:

- **Initialization**. Choose a first guess u_2^0 on Ω_2
- **Iteration**. For $n \geq 1$, find u_i^n in $H^1(\Omega_i)(i = 1, 2)$ such that

$$\begin{array}{llll} Lu_1^n = f & \text{in } \Omega_1, \quad u_1^n = u_2^{n-1} & \text{on } \gamma_{12}, \quad u_1^n = g & \text{on } \partial\Omega_1 \cap \partial\Omega; \\ Lu_2^n = f & \text{in } \Omega_2, \quad u_2^n = u_1^n & \text{on } \gamma_{21}, \quad u_2^n = g & \text{on } \partial\Omega_2 \cap \partial\Omega. \end{array}$$

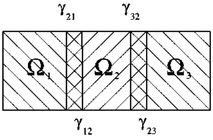


FIG. 1. Decomposition into three subdomains with full overlap.

Under this form, this algorithm presents no intrinsic parallelism. To generalize it to M subdomains with a possible parallelization, two ways are possible.

The first one, known as the "additive" Schwarz method, consists of the following algorithm, in which the computation of the approximate solution at iteration n only depends of the approximate solution at iteration $n-1$:

- **Initialization**. Choose a first guess u_i^0 on each subdomain Ω_i
- **Iteration**. For $n \geq 1$ (with u_i^{n-1} known), find u_i^n in $H^1(\Omega_i)$ such that

$$\begin{aligned} Lu_i^n &= f && \text{in } \Omega_i \\ u_i^n &= g && \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i^n &= u_j^{n-1} && \text{on } \gamma_{ij} \quad \text{for } j \in 1 \cdots M \text{ such that } \Omega_j \cap \Omega_i \neq \emptyset. \end{aligned}$$

The convergence of this algorithm can be demonstrated under the assumption that $\Omega_i \cap \Omega_j \cap \Omega_k = \emptyset \ \forall (i \neq j \neq k)$. However, the overlapping conditions can be less restrictive [14]. Moreover, the presence of the overlapping areas implicitly ensures the continuity of u and its derivatives between the different subdomains. This algorithm will be denoted by Schwarz-1. Another possible parallel implementation of the Schwarz method is to conserve the alternating nature of the original algorithm and to introduce the parallelization via a red/black numbering of the M subdomains; each subdomain is associated to one color, for example, Ω_i is red if i is even and black if i is odd, in such a way that two subdomains with common boundaries have different colors:

- **Initialization**. Choose a first guess u_i^0 on each black subdomain Ω_i
- **Iteration**. For $n \geq 1$ (with u_i^{n-1} known), find u_i^n in $H^1(\Omega_i)$ such that for Ω_i red

$$\begin{aligned} Lu_i^n &= f && \text{in } \Omega_i \\ u_i^n &= g && \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i^n &= u_j^{n-1} && \text{on } \gamma_{ij} \quad \text{for } j \in 1 \cdots M \text{ such that } \Omega_j \cap \Omega_i \neq \emptyset; \end{aligned}$$

then for Ω_i black

$$\begin{aligned} Lu_i^n &= f && \text{in } \Omega_i \\ u_i^n &= g && \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i^n &= u_j^{n} && \text{on } \gamma_{ij} \quad \text{for } j \in 1 \cdots M \text{ such that } \Omega_j \cap \Omega_i \neq \emptyset. \end{aligned}$$

This algorithm will be denoted by Schwarz-2. The link between these two algorithms is close to the link between the Jacobi and Gauss–Seidel methods for the solution of linear systems (see, for example, [4]). So, in the same way than SOR is a natural extension of the Jacobi and Gauss–Seidel algorithms, a possible extension of the previous Schwarz methods is:

- **Initialization**. Choose a first guess u_i^0 on each subdomain Ω_i
- **Iteration**. For $n \geq 1$, find u_i^n in $H^1(\Omega_i)$ such that for Ω_i red

$$\begin{aligned} Lu_i^n &= f && \text{in } \Omega_i \\ u_i^n &= g && \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i^n &= u_j^{n-2} + \omega\big(u_j^{n-1} - u_j^{n-2}\big) && \text{on } \gamma_{ij} \quad \text{for } j \in 1 \cdots M \text{ such that } \Omega_j \cap \Omega_i \neq \emptyset; \end{aligned}$$

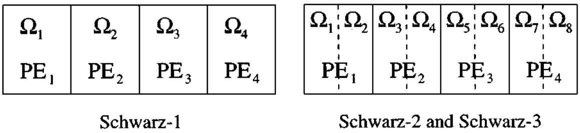

Ω_1 PE$_1$
Ω_2 PE$_2$
Ω_3 PE$_3$
Ω_4 PE$_4$
Schwarz-1
Ω_1 Ω_2 Ω_3 Ω_4 Ω_5 Ω_6 Ω_7 Ω_8
PE$_1$ PE$_2$ PE$_3$ PE$_4$
Schwarz-2 and Schwarz-3

FIG. 2. Assignment subdomains—processors in the case of four subdomains (overlap not represented).

then for Ω_i black

$$\begin{aligned} Lu_i^n &= f && \text{in } \Omega_i \\ u_i^n &= g && \text{on } \partial\Omega_i \cap \partial\Omega \\ u_i^n &= u_j^{n-1} + \omega\big(u_j^n - u_j^{n-1}\big) && \text{on } \gamma_{ij} \quad \text{for } j \in 1 \cdots M \text{ such that } \Omega_j \cap \Omega_i \neq \emptyset. \end{aligned}$$

This algorithm will be denoted by Schwarz-3.

Before addressing the convergence properties of these methods, let us discuss briefly their practical implementation. The usual way to code the Schwarz method is:

- initialize data on the processors
- 1. exchange of the boundary values with the neighbours
- 2. solve the local problem
- 3. compute a stopping criterion in parallel
- 4. go to 1 or continue

The Schwarz-1 algorithm is explicitly parallel and can be easily implemented by assigning a subdomain per processor. On the other hand, Schwarz-2 and Schwarz-3 algorithms require more subdomains than processors to be fully parallel. A usual implementation consists in defining twice more subdomains than processors, and assigning one red and one black subdomain to each processor. For example, in the case of four processors, Fig. 2 presents the decomposition for the different Schwarz algorithms.

3. CONVERGENCE PROPERTIES IN THE CONTEXT OF THE HELMHOLTZ EQUATION

As mentioned previously, in the context of ocean modelling, the equations to be solved numerically at each time step mainly consists of Poisson and Helmholtz equations. As a matter of fact, when they are not totally explicit, the primitives equations (PE) models require the solution of a Poisson equation for the barotropic mode. And similarly, quasi-geostrophic (QG) models require the solution of both Poisson and Helmholtz equations, since the system to solve at each time step is of the form:

$$\begin{aligned} \Delta u &= f && \text{for the barotropic mode} \\ \Delta u - \lambda^2 u &= g && \text{for each baroclinic mode,} \end{aligned}$$

where $\lambda = 1/R_d$, R_d being the Rossby radius of deformation corresponding to the baroclinic mode.

As an example, the derivation of the QG equations leading to these Helmholtz equations will be described in Section 4.

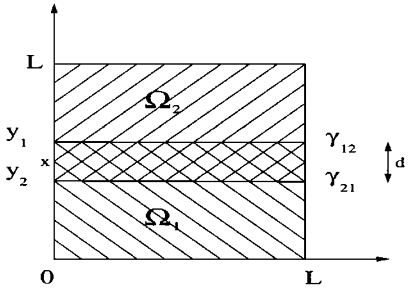


FIG. 3. Decomposition into two subdomains.

3.1. *Convergence Results for the Helmholtz Equation*

In the case of the Helmholtz equations, we can determine the theoretical factor of convergence of the Schwarz method. For sake of simplicity we consider a decomposition of a $L \times L$ domain into two subdomains as in Fig. 3 (more details can be found in [7]). Let

- u^* be the solution of the problem
- $e_1^n = u_1^n - u^*$ be the error on the subdomain Ω_1 at iteration n, and
- $e_2^n = u_2^n - u^*$ be the error on the subdomain Ω_2 at iteration n.

ALGORITHM SCHWARZ-1. e_1^n et e_2^n are the solutions of the system:

$$\begin{aligned} \Delta e_1^n - \lambda^2 e_1^n &= 0 && \text{in } \Omega_1 \\ e_1^n &= 0 && \text{on } \partial\Omega_1 \cap \partial\Omega \\ e_1^n &= e_2^{n-1} && \text{on } \gamma_{12}; \end{aligned}$$

$$\begin{aligned} \Delta e_2^n - \lambda^2 e_2^n &= 0 && \text{in } \Omega_2 \\ e_2^n &= 0 && \text{on } \partial\Omega_2 \cap \partial\Omega \\ e_2^n &= e_1^{n-1} && \text{on } \gamma_{21}. \end{aligned}$$

By a separate variables method we determine solutions of those systems:

$$e_1^n(x,y) = \sum_{m=1}^{\infty} a_m^n \sin \frac{m\pi}{L} x \frac{\sinh\sqrt{\lambda^2 + \frac{m^2\pi^2}{L^2}}y}{\sinh\sqrt{\lambda^2 + \frac{m^2\pi^2}{L^2}}y_1}$$

$$e_2^n(x,y) = \sum_{m=1}^{\infty} b_m^n \sin \frac{m\pi}{L} x \frac{\sinh\sqrt{\lambda^2 + \frac{m^2\pi^2}{L^2}}(L-y)}{\sinh\sqrt{\lambda^2 + \frac{m^2\pi^2}{L^2}}(L-y_2)}.$$

Then the boundary conditions yield

$$a_m^n = \frac{\sinh\sqrt{\lambda^2 + \frac{m^2\pi^2}{L^2}}y_2 \ \sinh\sqrt{\lambda^2 + \frac{m^2\pi^2}{L^2}}(L-y_1)}{\sinh\sqrt{\lambda^2 + \frac{m^2\pi^2}{L^2}}y_1 \ \sinh\sqrt{\lambda^2 + \frac{m^2\pi^2}{L^2}}(L-y_2)} a_m^{n-2}$$

and the same relation for b_m^n. A study of the ratio a_m^n / a_m^{n-2} shows that its value is maximum for $m = 1$. If we denote by μ this value we have

$$\begin{aligned}\left|e_1^n(x,y)\right| &\le \sum_{m=1}^{\infty}\left|a_m^n\right| = \sum_m \left|a_m^{n-2}\frac{a_m^n}{a_m^{n-2}}\right| \\ &\le \max\left(\left|\frac{a_m^n}{a_m^{n-2}}\right|\right)\sum_m\left|a_m^{n-2}\right| \\ &\le \mu\sum_m\left|a_m^{n-2}\right| \\ &\le \mu^{E(\frac{n}{2})}\sum_m\left|a_m^0\right| \left(\sum_m\left|a_m^1\right| \text{ if } n \text{ is even}\right).\end{aligned}$$

It follows that the theoretical convergence rate of the Schwarz-1 method is

$$\rho_1(y_1,y_2) = \sqrt{\mu} = \sqrt{\frac{\sinh\beta y_2\ \sinh\beta(L-y_1)}{\sinh\beta y_1\ \sinh\beta(L-y_2)}},$$

where $\beta = \sqrt{\lambda^2 + \pi^2/L^2}$.

ALGORITHM SCHWARZ-2. The same study leads to the following convergence rate:

$$\rho_2(y_1,y_2) = \frac{\sinh\beta y_2\ \sinh\beta(L-y_1)}{\sinh\beta y_1\ \sinh\beta(L-y_2)} = \mu.$$

ALGORITHM SCHWARZ-3. In [8], Evans and Kang computed the convergence rate of the algorithm Schwarz-3: if $\omega_{\text{opt}} = \left(3/\sqrt{\rho_2}\right)\cos[(s+4\pi)/3]$, where $s = \text{arc}\cos(-\sqrt{\mu}\,)$ then

$$\rho_3(y_1,y_2) = \frac{3(\omega_{\text{opt}}-1)}{\omega_{\text{opt}}}.$$

3.2. *Discussion*

In this section we intend to study the evolution of the convergence factor as a function of the partitioning of the domain. In the following discussion the algorithm Schwarz-1 is used, but all three algorithms have similar convergence properties with regard to the studied parameters.

1. *Size of subdomains.* Let $x = (y_1 + y_2)/2$ the location of the middle of the overlapping area and $d = y_1 - y_2$ the overlapping length as in Fig. 3. For a fixed value of d, we intend to find an optimal value for x. Figure 4 shows the evolution of the convergence rate ρ_1 as a function of x, in the case of the barotropic mode ($\lambda = 0$) and of a baroclinic mode ($\lambda = (12.10^3\ \text{m})^{-1}$). Our domain is a square of size $L = [4.10^6\ \text{m}]$. For each case, d is fixed to its optimal value: 9×10^5 m for the barotropic mode and 2×10^5 m for the baroclinic mode (this choice will be discussed in Section 4.2).

For the baroclinic mode, Fig. 4 shows that the location of the artificial boundary does not have a real influence on the convergence factor. This can be interpreted by examining the relative sizes of L and $R_d = 1/\lambda$. As a matter of fact, for values of λ and L which satisfy the rule $\lambda L \gg \pi$ (i.e., $\pi R_d \ll L$), then β can be approximated by λ and ρ_1 by $\exp(-\lambda d)$, where d is the overlap. In other words, when the typical scale of baroclinic motions is

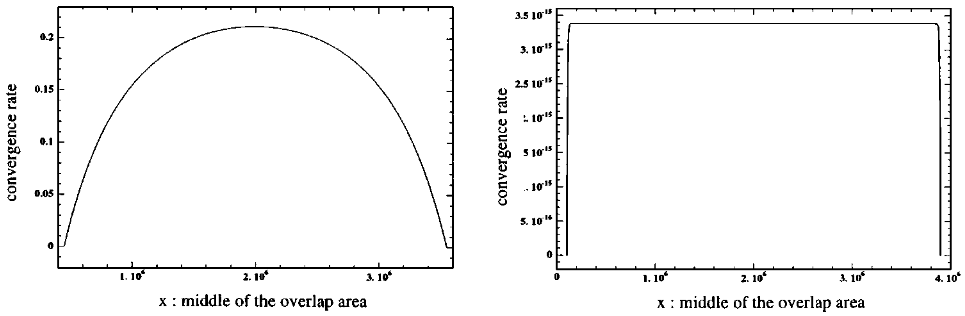


FIG. 4. Convergence rate ρ_1 as a function of x the location of the middle of the overlap area: (left) barotropic mode; (right) baroclinic mode.

small, compared to the ocean basin size (which is almost true in practical applications), the convergence rate of the Schwarz algorithm is nearly independent of the location of the overlap area but depends only of the size of the overlap.

On the other hand, when the relation $\lambda L \gg \pi$ is not satisfied, as it is the case for the barotropic mode where $\lambda = 0$, we can notice on Fig. 4a that the choice of two subdomains of equal sizes ($x = L/2$) leads to the worst convergence factor.

This remark can be useful during the implementation of a DDM on a sequential computer. This can be done, for example, to address such problems as increasing the precision of a numerical solution, or as defining differnt values of coefficients in each subdomain. In these cases, when it is possible, a decomposition into subdomains of different sizes would be more efficient than into subdomains in equal sizes.

However, processing with subdomains of different sizes is unacceptable for the programming on parallel machines. As a matter of fact, at each iteration a subdomain waits for its new boundary values from its neighbours. Then for evident reasons of load balancing DDMs strongly require that the different processors end their local work at the same time. So the subdomains will be chosen of equal sizes, even if this penalizes the convergence rate.

2. *Variation of the overlap.* This choice of $x = L/2$ being made, we can now study the evolution of the convergence rate ρ_1 as a function of d, the size of the overlap. Figure 5 shows the evolution of the convergence factor with the overlap for a decomposition into

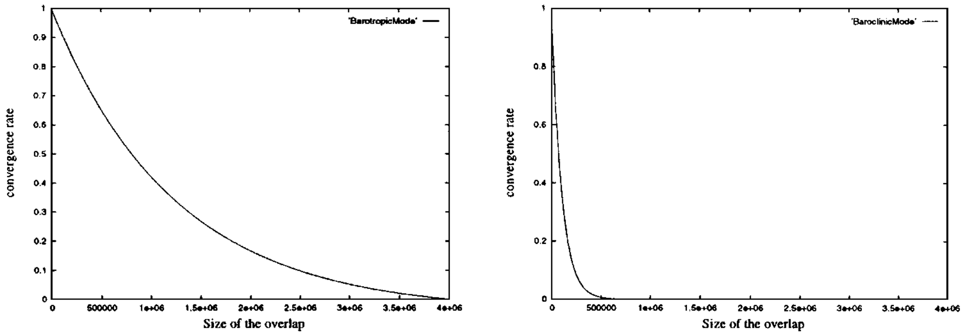


FIG. 5. Evolution of the convergence factor with the size of the overlap: (left) barotropic mode; (right) baroclinic mode.

two subdomains of equal sizes. A Taylor expansion of the convergence rate gives the slope at the origin: $-(\pi/L)\coth(\pi/2)$ for the barotropic mode and $-\lambda$ for the baroclinic mode (for this mode the curve is exponential). Using the values of L and λ mentioned previously, this yields to slopes equal to $-8,57\times 10^{-7}$ for the barotropic mode and $-8,33\times 10^{-5}$ for the baroclinic mode. This explains the differences between the two figures.

For the baroclinic mode, even a small size of overlap leads to a good convergence rate, and increasing the size of the overlap would only increase the amount computations without accelerating the convergence. On the other hand, for the barotropic mode a larger value of overlap is needed. For the practical choice of d, and optimal value can be found which balances the two contradictory effects associated with an increase of d: the increasing amount of computations at each iteration; and the decreasing number of iterations.

3. *Generalization to n subdomains.* For the case of n subdomains ($n\geq 3$) we cannot find a literal expression of the convergence factor. However, let us consider a decomposition into n subdomains along the Y-axis, i.e. $\Omega_i=[0,L]\times[a_i,b_i], i=1\cdots n,$ where $0=a_1<a_2<b_1<a_3<b_2<\cdots<a_{n-1}<b_{n-2}<a_n<b_{n-1}<b_n=L$.

Kang and Evans [11] proved that the convergence factor is governed by the spectral radius of the matrix

$$A=\begin{pmatrix} 0 & D_1^{-1}B_1 & 0 & \cdots & 0 \\ D_2^{-1}B_2' & 0 & D_2^{-1}B_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & D_{n-1}^{-1}B_{n-1} \\ 0 & \cdots & 0 & D_n^{-1}B_n' & 0 \end{pmatrix}$$

where

$$D_i=\begin{pmatrix} \sinh\beta a_i & \cosh\beta a_i \\ \sinh\beta b_i & \cosh\beta b_i \end{pmatrix}, \quad B_i=\begin{pmatrix} 0 & 0 \\ \sinh\beta b_i & \cosh\beta b_i, \end{pmatrix},$$

$$B_i'=\begin{pmatrix} \sinh\beta a_i & \cosh\beta a_i \\ 0 & 0 \end{pmatrix}.$$

Then in the case of n subdomains the convergence factor of the algorithm Schwarz-1 is the spectral radius of the matrix $A:\rho_{\text{Schwarz}-1}=\rho(A)=\max|\mu_i|$, where μ_i are the eigenvalues of A.

Figure 6 shows the evolution of the convergence factor as a function of the number of subdomains for the barotropic mode ($\lambda=0$). For this figure, a_i and b_i were chosen in order to obtain domains of equal size, and the length of the overlap is a quarter of the length of each subdomain. As can be seen, the convergence rate seems to strongly suffer of an increase of the number of subdomains. The choice of the first guess of the iteration will thus be of crucial importance (see Section 4.2).

3.3. *Conclusion on the Schwarz Algorithms*

This theoretical study let us conclude that the optimal overlap must be seen as a function of λ, i.e. as a function of the Rossby radius of deformation R_d. The properties of the method strongly differ between the barotropic mode and the baroclinic modes. We have

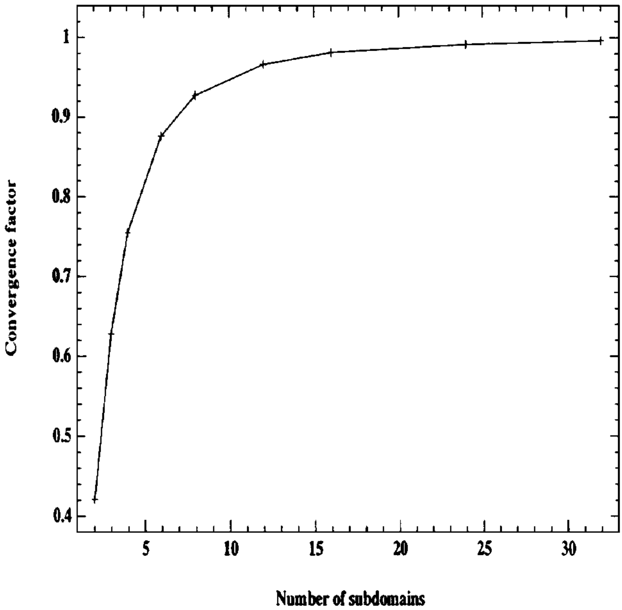


FIG. 6. Evolution of the convergence factor with the number of subdomains.

already tried to give a physical interpretation of this behaviour, but the next section will clearly establish the link between the physical properties of the ocean model and the optimal overlap, demonstrating the peculiar-role of the barotropic mode.

From a theoretical point of view, the Schwarz-3 algorithm seems to produce the best convergence factor for an appropriate value of ω.

4. NUMERICAL EXPERIMENTS

The results presented in this section were obtained with the algorithm Schwarz-1. However, all the remarks are also relevant to the other algorithms, and timing results are given for all the algorithms.

4.1. *The Numerical Model*

The model used in the present study is a multilayered quasi-geostrophic (QG) model, as first proposed by Holland [10]. For midlatitude oceanic circulations, QG equations may be derived from the Navier–Stokes equations, via an expansion as a function of the Rossby number $\varepsilon = U/f_0 L$, where U is a horizontal velocity scale (a few centimeters per second), L is a horizontal 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 Coriolis parameter is written as $f(y) = f_0 + \beta y$, where β is constant. At the order ε, one obtains the quasi-geostrophic equations, which can be written as the conservation of potential vorticity [17],

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

where Ψ is the three-dimensional streamfunction, D/Dt is the Lagrangian derivative,

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

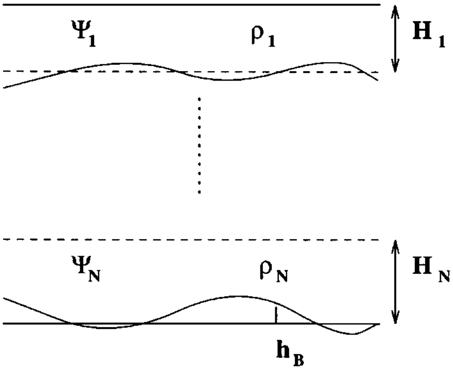


FIG. 7. Stratification of the model.

∇^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 discretization divides the ocean into N layers of constant densities $\rho_1, \ldots, \rho_N$. At rest, the thicknesses of the layers are $H_1, \ldots, H_N$ (see Fig. 7). 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
- A is a tridiagonal matrix which nonzero 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)

-

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

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), $T_1 \neq 0$, $T_k = 0$, $k > 1$ (wind stress curl).

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

$$\frac{\partial}{\partial t}[\nabla^2 PP^{-1}\Psi - PDP^{-1}\Psi] = PP^{-1}B;$$

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 others 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). This scheme has the disadvantage to develop a computational mode in the solution resulting in an unstable method. The occasional insertion of a step made by a two-level scheme (Euler scheme) suppresses the development of such a mode. For a detailed discussion of the leap-frog properties, see, for example, [16].

The leap-frog scheme leads to the system:

$$\nabla^2\Phi^{n+1} - D\Phi^{n+1} = 2\Delta t(P^{-1}B)^{(n-1,n)} + \nabla^2\Phi^{n-1} - D\Phi^{n-1}, \tag{4}$$

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 procedure [1] is used to compute the Jacobian operator.

At each time step, we must then solve N Helmholtz equations which unknowns are the modes:

$$\Delta\Phi_i^{n+1} - \lambda_i\Phi_i^{n+1} = C_i^{n,n-1}, \quad 1 \le i \le N.$$

The first equation ($i = 1$) corresponding to the barotropic mode is a Poisson equation ($\lambda_1 = 0$).

The different tests we realized were based on the following configuration of the ocean model:

- a rectangular domain of 4000 km × 4000 km, discretized with a resolution of 10 km.
- a vertical discretization of the domain into three layers, which thicknesses at rest are respectively 300, 700, and 4000 m. The density stratification is chosen to correspond roughly to the stratification of the North Atlantic ocean. The three values of the Helmholtz coefficients corresponding to the three modes are respectively $\lambda_1 = 0$ (i.e., Poisson equation for the barotropic mode), $\lambda_2 = 0.445 \times 10^{-4}$ and $\lambda_3 = 0.883 \times 10^{-4}$ m^{-1} for the baroclinic modes (i.e., Rossby radii of deformation $R_1 = 22.5$ km and $R_2 = 12$ km).

The ocean bottom is flat, and the wind is chosen in order to produce a two-gyre circulation, with an intense median jet (see Fig. 8).

Each test corresponds to a solution computed by the direct sequential solver. This solution is considered as the true solution and allows us to determine the corresponding right-hand side of the equation for the parallel programs. Then we can compare the solution given by the parallel algorithm with the original solution.

For each test case, we made several experiments using several fields issued from simulations. Then each result presented here is an average over those experiments.

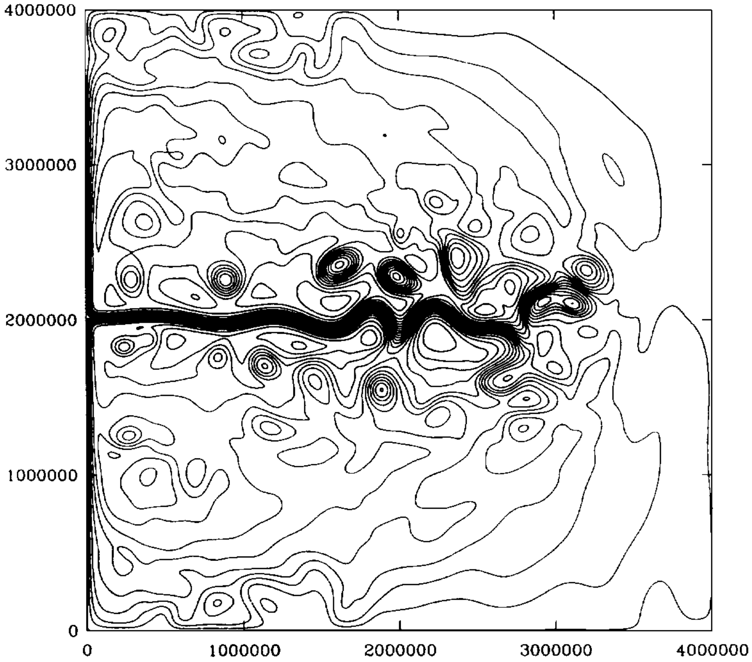


FIG. 8. Example of instantaneous field for the Barotropic mode.

The stopping criterion for the Schwarz method is

$$\sup_{i \le j \le M} \frac{\left\| u_j^n - u_j^{n-1} \right\|_2}{\left\| u_j^{n-1} \right\|_2} < \epsilon,$$

where M is the number of subdomains and $\| \ \|_2$ denotes the euclidian norm. ϵ is fixed to 10^{-5} for the tests. This choice is discussed further.

For the iterative Schwarz method, results are given with three different first guess u_j^0:

- a first guess 0, where $u_j^0 = 0$,
- a first guess 1, where $u_j^0(t+1) = u_j^*(t)$,
- a first guess 2, where $u_j^0(t+1) = 2u_j^*(t) - u_j^*(t-1)$ (i.e., a first-order approximation),

where $u_j^*(t)$ is the solution at the time t.

An example of the instantaneous field is given in Fig. 8 for the barotropic mode. As can be seen, the circulation is turbulent, and numerous rings are generated by the instabilities of the eastward jet. From the numerical point of view, the efficiency of the domain decomposition methods strongly depends on the solver used on each subdomain. In this model, we use a fast direct solver, combining fast Fourier transforms and cyclic reductions (FACR), to solve the Poisson and Helmholtz equations. This solver performs its Fourier transforms in the X direction. Thus the longer the dimension in the X direction, the faster the solver. This will have an influence on the choice of the partitioning.

In the next subsection we will illustrate the behaviour of the Schwarz method for this model and will particularly focus on the four following points: particular role of the barotropic mode, existence of an optimal partitioning, influence of an increase of the turbulence, and the error of the solver.

4.2. *Tuning of the Method Parameters*

As mentioned previously, the Schwarz algorithm depends on several parameters, like the overlap, the first guess, and the partitioning. In this section we try to fix those parameters.

Overlap. The optimal overlaps found for the three different modes were 900 km for the barotropic mode, 300 km for the first baroclinic mode, and 200 km for the second baroclinic mode. Those values correspond to the smallest times of execution. We recall that when the size of the overlap increases, the number of iterations decreases but the amount of work on each subdomain grows. So this optimal value corresponds to a minimum of time and not to a minimum number of iterations number.

All the results presented further in this paper were obtained with these overlaps. We can observe that the optimal overlap for the barotropic mode is much more important than for the baroclinic modes, which is consistent with the theoretical results presented previously (Fig. 5). For the barotropic mode the optimal overlap is nearly 900 km, which leads to an additional work of about 90%. For the baroclinic modes the additional work is about 20%. From a physical point of view, this result illustrates the fact that the convergence rate of the method is good as soon as the size of the overlap is of the order of the typical scale of motion ($2\pi R_d$ for the baroclinic modes, L for the barotropic mode).

Choice of the first guess. Now we focus on the influence of the first guess on the convergence properties. Table 1 shows the number of iterations required for the solution of each mode as a function of the first guess. These results totally agree with the interpretation of the theoretical factor of convergence we made in Section 3.2. The number of iterations for the baroclinic modes are much less than for the barotropic mode.

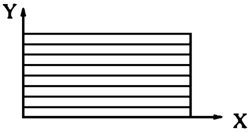
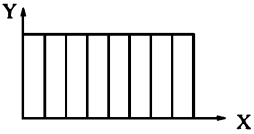
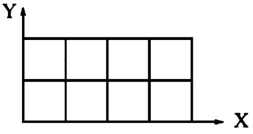
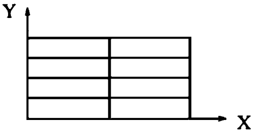
As suggested in Section 3.2, we see that the choice of the first guess has a strong influence on the results for the barotropic mode. In particular, the use of the first guess 2 (first-order approximation) can lead to noticeably better results than the first guess 1. Since most numerical models store two time steps in memory, the use of this first guess 2 is straightforward and can lead to an efficient implementation of the DDM.

Partitioning. Additional tests were performed, corresponding to different partitionings of the ocean basin into eight subdomains. The results for the barotropic mode and for the first guess 2 are displayed in Table 2. We indicate the number of iterations to converge and the corresponding relative error, which is given by $\|u - u^*\|_2/\|u^*\|_2$ on Ω. (For sake of simplicity the figures do not represent the overlaps.) It appears that a domain decomposition in the two directions leads to more iterations that in the one direction decomposition case, due to the increase of the overlap and to our solver. Then we decide to used a decomposition in one direction.

TABLE 1
Number of Iterations as a Function of the Mode and the First Guess

Mode	First guess 0	First guess 1	First guess 2
0	22	10	7
1	3	3	2
2	3	3	2

TABLE 2
Number of Iterations and Error for Different Partitionings

Partitioning	Iterations number	Relative error
	10	4.265e-5
	9	4.373e-5
	15	3.499e-5
	18	8.167e-5

Moreover, as mentioned previously the larger the length in the X direction, the faster the solver. So even if the DDM converges approximatively in the same number of iterations for the X or Y decompositions we prefer using a decomposition in the Y direction.

More than finding this optimal partitioning for this particular model, the important point here lies in the fact that the choice of the domain partitioning appears as a key factor for the efficiency of the method. It is linked to the different sources of anisotropy, in the numerical solver (different treatment of the X- and Y-axis), in the computer (vectorization, for instance), and in the physics of the problem.

4.3. *Influence of the Turbulence*

Another point of interest in the context of ocean modelling is the stability of the method with regard to an increase of the turbulence. As a matter of fact, the large scale low resolution ocean models exhibit only relatively low level of turbulence, while eddy resolving models can be highly turbulent. That is why we compared the performance of the method for three different cases corresponding to three levels of turbulence (denoted cases 1, 2, 3). The three corresponding Reynolds numbers are $R_e = 700$, 2000, and 6000, respectively. The results for the three modes and for the first guess 2 are summarized in Tables 3 and 4 in

TABLE 3
Iterations Number

Case	Mode 0	Mode 1	Mode 2
1	7	2	2
2	8	2	2
3	8	3	3

TABLE 4
Relative Error

Case	Mode 0	Mode 1	Mode 2
1	6.026e-5	1.135e-5	7.94e-6
2	7.229e-5	1.279e-5	9.470e-6
3	1.056e-4	1.123e-4	8.433e-6

terms of number of iterations and relative error. The important point here is that the number of iterations remains stable as the turbulence increases. However, we can observe a small increase of the relative error, which remains in agreement with the error associated with the direct sequential solver.

4.4. *Accuracy of the Method*

In this section we compare the errors given by the Schwarz method and by the sequential solver. We first study the evolution of the error with an increase of the number of subdomains. Table 5 summarizes the results obtained for a decomposition in the Y direction. It appears that increasing the number of subdomains produces a decrease of the relative error, yielding a greater accuracy than with the direct solver. Indeed, as all other direct methods, the direct fast Fourier transform method can lead to poor precision results when the size of the domain becomes too large. Here domain decomposition appears as a way to guarantee good accuracy using the same fast direct method on all subdomains.

In our case the relative error given by the direct solver is small enough for our study. So we have now to find a stopping criterion for the DDM which leads to the same precision as the direct solver does (in order to compare timing results obtained for the same precision). Figure 9 shows an example of the evolution of the stopping criterion value $\sup_{i\leq j\leq M}(\|u_j^n - u_j^{n-1}\|_2/\|u_j^{n-1}\|_2)$ and of the corresponding relative error. For this test, the precision obtained by the DDM is better than the precision of the sequential solver after 15 iterations. This corresponds to a stopping criterion of $\epsilon = 2, 5.10^{-4}$. In the following, we will use $\epsilon = 10^{-4}$, which ensures a good precision. Note that this test was made with four subdomains. However, the first remark of this paragraph leads to the conclusion that for the same value of ϵ the precision is improved when using more subdomains.

4.5. *Timing Results*

Code implementation. In order to implement the code on a parallel machine we used the standard language MPI (message passing interface; see [15]). This language is a library of communication routines that allows the exchange of messages between processors.

TABLE 5
Evolution Error—Number of Subdomains

Error of the direct solver: 8.715e-5	
Number of subdomains	Relative error
2	1.085e-4
4	6.026e-5
8	4.265e-5
16	2.152e-5

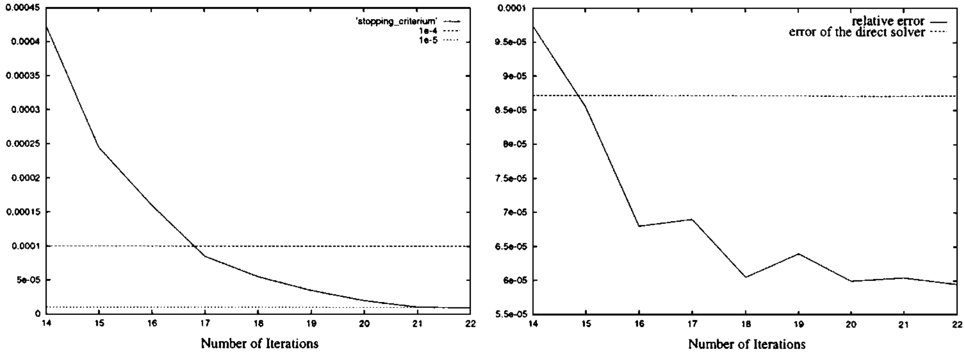


FIG. 9. Evolution of the value of the stopping criterion (left) and the relative error (right).

The Schwarz algorithms were integrated in the quasi-geostrophic model. So the Helmholtz equations are solved at each time step using the Schwarz alternating method. The remaining part of the code (mainly the right hand side and the time stepping) can be computed totally independently on each subdomain and does not require any communications. The model is integrated during 1000 time steps. We implemented this code on the CRAY T3D (128 processors) and the IBM SP1 (32 processors). These two massively parallel machines are actually the most commonly used in the scientific community. The T3D benefits of a very fast communication network due to its topology (a 3D torus) and the global addressing of its memory. On the other hand, the communication times are more costly on the SP1 (even using the IBM switch). However, the SP1 takes advantage of its processors (IBM RS6000) which are much faster that the relatively slow processors of the T3D (DIGITAL DEC alpha).

The practical implementation of the Schwarz alternating method is very straightforward. The algorithm consists of two main points. First, we need to exchange the boundary conditions between subdomains. This is done by packing all the elements of a boundary in an array and by sending this array to the processors handling the neighboring subdomain (using the routines **MPI_Send** and **MPI_Recv**). The second point is the computation of the residual error. We start by computing the relative error locally on each subdomain and then the maximum is computed over all the subdomains. MPI provides a very efficient procedure for this **(MPI_Allreduce**). This routine takes in argument the local norm on each subdomain and then, called by each processor with the parameter **MPI_MAX**, it returns *to each processor* the maximum of all the norms. An equivalent routine does not exist in PVM (where the maximum is returned to only *one selected processor*) and we take it as a significant drawback of PVM.

Timing results. The evolution of the parallel time on the CRAY T3D and the IBM SP1 are displayed in Fig. 10. It appears clearly that the CRAY T3D and the IBM SP1 have similar behaviour with regard to an increase of the number of subdomains. Algorithms Schwarz-2 and Schwarz-3 are more efficient than algorithm Schwarz-1, which is in agreement with the results found in Section 3. For more than 16 processors, the timings do not significantly decrease anymore. The reason is that the size of the subdomains and, therefore, the ratio computation over communication become too small. A second reason is that we know from the theoretical study that the convergence properties of the algorithm decrease when the number of subdomains grows.

Moreover, we can notice that the CRAY T3D and the IBM SP1 both lead approximately to the same timings. Despite its relatively slow processors, the T3D gives timings as good as

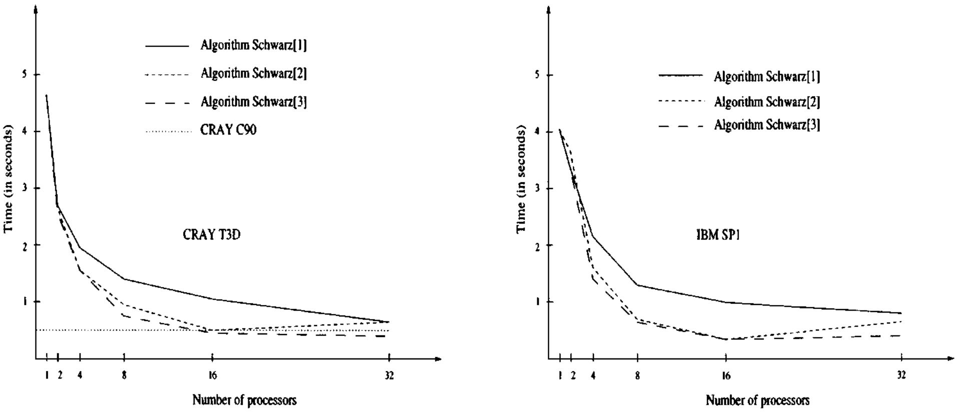


FIG. 10. Timings (left) CRAY T3D; (right) IBM SP1.

those given by the IBM SP1 due to its efficient communication network. On 16 processors, the code runs on the T3D as fast as on the CRAY C90, which is a good result for a parallel application.

5. CONCLUSION

In this paper we examine the application of the Schwarz domain decomposition method to the solution of Poisson and Helmholtz equations in the context of ocean modelling. A theoretical study highlights the peculiar role played by the barotropic mode. Due to the smaller typical length scales of baroclinic modes, the convergence of the algorithm is achieved faster for these modes than for the barotropic mode. On the numerical point of view, we have discussed the practical choice of the different parameters involved in the implementation of the method. Our tests showed that the convergence properties of the method are not altered by an increase of the turbulence, which was not true for a nonoverlapping method we also tested (not discussed in this paper). Moreover, it is shown that DDMs allow the solution of large problems with good accuracy, while global methods exhibit a loss of accuracy when the size of the problem increases. Timing results show that this method can be efficiently incorporated in an oceanic model for use on parallel computers. This model is presently used on the CRAY T3D for simulations of the Antarctic circumpolar current with a 1/6° horizontal resolution.

ACKNOWLEDGMENTS

The calculations were made possible by the computer facilities at the Centre Grenoblois de Calcul Vectoriel (CEA) and at the Institut du Developpement et des Ressources en Informatique Scientifique (IDRIS-CNRS). IDOPT is a joint CNRS/University of Grenoble/Institut National Polytechnique de Grenoble/INRIA project.

REFERENCES

1. A. Arakawa, Computational design for long term integration of the equations of fluid motions, *J. Comput. Phys.* **1**, 119 (1966).
2. L. Bengtsson and C. Temperton, Difference approximations to quasi geostrophic models, in *Numerical Methods used in Atmospheric Models*, Vol. II, p.338. (WMO/ICSU Joint Organizing Committee, London, 1979).[GARP Publ. Ser. 17]

3. R. Bleck, R. Rooth, D. Hu, and L. Smith, Salinity-driven thermocline transients in a wind- and thermohaline-forced isopycnic model of the north Atlantic, *J. Phys. Oceanogr.* **22**, 1486 (1992).
4. T. F. Chan and D. Goovaerts, On the relationship between overlapping and nonoverlapping domain decomposition methods, *SIAM J. Matrix Anal. Appl.* **13**, 663 (1992).
5. C. G. Douglas and M. B. Douglas, MGNet Bibliography, in mgnet/bib/mgnet.bib, on anonymous ftp server casper.ca.yale.edu, Yale University, Department of Computer Science, New Haven, CT, 1996. [Last modified on October 22, 1996]
6. J. K. Dukowicz and R. D. Smith, Implicit free-surface method for the Bryan–Cox–Semtner ocean model, *J. Geophys. Res.* **99**, 7991 (1994).
7. D. J. Evans, S. Jianping, K. Lishan, and C. Yuping, The convergence rate of the Schwarz alternating procedure (II)-for Two Dimensional Problems. *Int. Jour. Comp. Math.* **20**, 325 (1986).
8. D. J. Evans and K. Lishan, New domain decomposition strategies for elliptic partial differential equations, in *Proc. of the Second International Symposium on Domain Decomposition Methods for Partial Differential Equations*, p. 173 (SIAM, Philadelphia, 1989).
9. M. Guyon, M. Chartier, F.-X. Roux, and P. Fraunie, First considerations about modelling the ocean general circulation on MIMD machines by domain decomposition method, in *Proceedings of the NATO Advanced Research Workshop on High Performance Computing in the Geosciences, les Houches, France, 21–25 June 1993*, edited by F.-X. Le Dimet (Kluwer Academic, Dordrecht, 1995). [NATO ASI Series C462]
10. W. R. Holland, The role of mesoscale eddies in the general circulation of the ocean. Numerical experiments using a wind-driven quasi-geostrophic model, *J. Phys. Oceanogr.* **8**, 363 (1978).
11. K. Lishan and D. J. Evans, The convergence rate of the Schwarz alternating procedure (V)-for more than two subdomains. *Int. Jour. Comp. Math.* **23**, 295 (1988).
12. P. D. Killworth, D. Stainforth, D. J. Webb, and S. M. Paterson, The development of a free-surface Bryan–Cox–Semtner ocean model, *J. Phys. Oceanogr.* **21**, 1333 (1991).
13. P. L. Lions, On the Schwarz alternating method I, in *Proc. of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, p. 1 (SIAM, Philadelphia, 1988).
14. P. L. Lions, On the Schwarz alternating method II, Stochastic interpretation and order properties, in *Proc. of the Second International Symposium on Domain Decomposition Methods for Partial Differential Equations*, p. 47 (SIAM, Philadelphia, 1989).
15. MPI, MPI: A message-passing interface standard, in /pub/mpi /mpi-report.ps.Z on anonymous ftp server ftp.mcs.anl.gov. [228 pages]
16. F. Mesinger and A. Arakawa, Time differencing schemes, *Numerical Methods Used in Atmospheric Models*, Vol. II. p. 9 (WMO/ICSU Joint Organizing Committee, 1976). [GARP Publ. Ser. 17]
17. J. Pedlosky, *Geophysical Fluid Dynamics* (Springer-Verlag, Berlin, 1987).
18. A. J. Semtner, Modelling ocean circulation, *Science* **269**, 1379 (1995).
19. R. D. Smith, J. K. Dukowicz, and R. C. Malone, Parallel ocean general circulation model, *Physica D* **60**, 38 (1992).