CNRS - Université Pierre et Marie Curie - Université Versailles-Saint-Quentin
CEA - ORSTOM - Ecole Normale Supérieure - Ecole Polytechnique

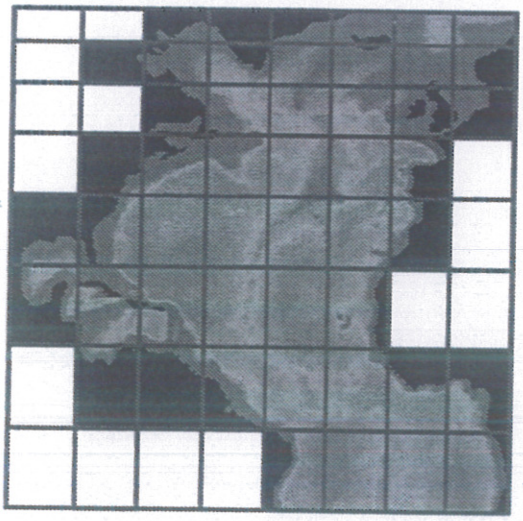
Institut Pierre Simon Laplace
des Sciences de l'Environnement Global

Notes du Pôle de Modélisation

Domain Decomposition Method as a Nutshell for Massively Parallel Ocean Modelling with the OPA Model

Marc Guyon (1), Gurvan Madec (2), François-Xavier Roux (3), Christophe Herbaut (2), Maurice Imbard (2), and Philippe Fraunie (4)

(1) CETIIS, Aix-en-Provence
(2) LODYC, France (3) ONERA, Châtillon
(4) LSEET, La Garde



Janvier 1999 **Note n° 12**

I P S L

CNRS - Université Pierre et Marie Curie - Université Versailles-Saint-Quentin
CEA -CNES - ORSTOM - Ecole Normale Supérieure - Ecole Polytechnique

Institut Pierre Simon Laplace
des Sciences de l'Environnement Global

CETP - LMD - LODYC - LPCM - LSCE - SA

Université Pierre-et-Marie-Curie
B 102 - T15-E5 - 4, Place Jussieu
75252 Paris Cedex 05 (France)
Tél : (33) 01 44 27 39 83
Fax : (33) 01 44 27 37 76

Université Versailles-Saint-Quentin
College Vauban, 47 Boulevard Vauban
78047 Guyancourt Cedex (France)
Tél : (33) 01 39 25 58 17
Fax : (33) 01 39 25 58 22

Domain Decomposition Method as a Nutshell for Massively Parallel Ocean Modelling with the OPA Model

Marc Guyon (1), Gurvan Madec (2), François-Xavier Roux (3), Christophe Herbaut (2), Maurice Imbard (2), and Philippe Fraunie (4)

(1) CETIIS, Aix-en-Provence
(2) LODYC, France (3) ONERA, Châtillon
(4) LSEET, La Garde

Ce papier présente les modifications qui ont été apportées à OPA, le modèle de circulation générale océanique du LODYC, pour l'adapter aux calculateurs à mémoire distribuée. La parallélisation du modèle est basée sur une méthode de décomposition de domaine. Le domaine d'étude est découpé en sous-domaines dont les dépendances locales sont levées par l'introduction d'un opérateur de condition aux limites latérales. Cet opérateur exploite une stratégie de découpage en crayons verticaux et de communication par frontière recouvrante. Deux approches sont proposées pour résoudre l'équation elliptique associée au terme de gradient de pression de surface : un gradient conjugué préconditioné parallèle et un solveur basé sur la méthode FETI. Il est montré que le solveur FETI présente de meilleures propriétés de convergence qui le rendent particulièrement attractif pour des domaines comportant un grand nombre de points de grille. Le comportement du modèle parallèle est illustré à travers un jeu de cas test académiques de taille croissante et une application réelle de modélisation de la Méditérannée occidentale.

Janvier 1999
Note n° 12

Directeur de publication : Gérard Mégie, Directeur de l'IPSL
Demande de n° ISSN en cours

Domain Decomposition Method as a Nutshell for Massively Parallel Ocean Modelling with the OPA Model

MARC GUYON [a], GURVAN MADEC [b],
FRANÇOIS-XAVIER ROUX [c], CHRISTOPHE HERBAUT [b],
MAURICE IMBARD [b], and PHILIPPE FRAUNIE [d]

[a] CETIIS, Aix Métropole, Bat. D, 30 av. Malacrida, 13100 Aix-en-Provence

[b] LODYC, UMR 121, CNRS-UPMC-ORSTOM, Université Pierre et Marie Curie,
Case 100, 4, place Jussieu, 75252 Paris Cedex 05, France

[c] ONERA, BP 72, 9322 Châtillon Cedex, France

[d] LSEET, Université de Toulon et du Var, BP 132, 83957 La Garde Cedex, France

ABSTRACT

This paper presents the adaptation of OPA, the LODYC ocean general circulation model, to distributed memory computers. The parallelization is based on Domain Decomposition Methods. The local dependencies problem is solved by a lateral boundary operator that exploits a pencil splitting and an overlapping strategy. Two different algorithms have been implemented to solve the elliptic equation associated to surface pressure gradient contribution: a parallel preconditioned conjugate gradient and a new solver based on a Dual Schur Complement method (the FETI method). The FETI solver is shown to have better properties of convergence, especially for large model size. Parallel model validation and performances are assessed on increasing size academic problems and on a real life application of the western Mediterranean sea circulation.

[illegible]ACT

This pa[illegible] p[illegible] the adaptation [illegible] OP[illegible] the [illegible]DYC ocean general circulation model to distributed memory computers. The [illegible] is based on Domain decomposition methods. The total dependencies on [illegible] is [illegible] by a set of boundary operator that [illegible] splitting and an overlapping strategy. Two different algorithms have been implemented to solve the elliptic [illegible] to surface pressure [illegible] parallel preconditioned conjugate gradient and a new solver based on a Schur complement [illegible] method). The [illegible] solver is shown to have [illegible] properties of convergence [illegible] ally [illegible] are [illegible] Parallel model [illegible] and [illegible] are [illegible] on [illegible] academic [illegible] and [illegible] Meridional [illegible]

1. Introduction

The study of the ocean and its influence on the global climate system require to investigate more and more precisely physical processes described by a sophisticated set of equations based on Navier-Stokes equations plus some physical approximations and parameterizations: the Primitive Equation. These processes are characterized by a broad range of spatial and temporal scales which encompass the relevant dynamics and involve longer integrations at finer resolution. One of the key to investigate a new physics and to continue to do useful climate research is to design numerical models that can utilise state of art high performance computers. As computer technology advances into the age of massively parallel processors, the new generation of OGCM has to offer an efficient parallel tool to exploit memory and computing resources of distributed architectures and previous limitations regarding domain size, simulation times and sensitivity testing will be much less daunting than before [1]. Developers need to elaborate an efficient parallel strategy and adapt the software structure to benefit from these new generation of computers. At the beginning of nineteenth, the Single Instruction Multiple Data flow execution model (SIMD) [2] seemed to be the parallel solution in Earth Sciences field. To obtain an efficient parallel model, Wolters [3] has developed a parallel implementation of the Hirlam model on a SIMD MasPar MP-1 system, like Smith *et al.* [4] for the GFDL Model [5] on a CM-2 Connection Machine, and Vittard *et al.* [6] for the OPA model [7] on the same platform. Today, SIMD execution model seems too poor in CFD context, at each time step, each processor has to realise the same operation on different data, and requires high degree of parallelism to exploit efficiently a very large number of elementary processors. So parallelization of implicit operator, that couples all the points of a considered domain, like the surface pressure gradient (hereafter referred to as SPG) in ocean field has been a critical point. The Multiple Instruction Multiple Data flow execution model (MIMD) [2] appears to be more appropriate to answer to needs of the CFD scientists [8, 9]. For computer development and for numerical experiment our target architecture is the MIMD distributed machines. These machines use a large number of standard processors and memory is physically distributed among the processors to constitute nodes. These nodes are connected together, with a topology : tree, 2D grid, 3D torus, hypercube...

In the context of a community ocean model, a right methodology has to preserve the readability of the FORTRAN code by non computer specialists for future numerical developments to investigate new processes and the portability to remain machine independent.

The domain decomposition methods are known to be very effective as parallelization methods of complex scientifics models, they consist in splitting the large computation domain of a numerical experiment into several smaller sub-domains and in solving the set of equations by addressing "independent" local problems. Such an approach reduces the elapsed time, minimizes the computer code modifications that preserves its properties (physics parameterizations implementation, modular coding style, vectorization, ...) and ensures software portability (the computer could be a devoted architecture or a workstation network...). In addition, an original mathematical methods [10] can be considered to improve the numerical behaviour in order to save computer time. As the local problems are solved by using the same code, it has been noticed that even if our methodology is very general and well suited to exploit coupled models on MIMD platform, we focus on the parallelization of OPA model by using a SPMD (Single Process Multiple Data Flow) execution model.

The message passing programming model, that considers the Massively Parallel Processors (MPP) machine is equivalent to a multi-machine and each machine has its own processor unit and its local memory, defines a natural framework to exploit a Domain Decomposition strategy and a distributed algorithm efficient on a distributed architecture. Developers respect the initial structure of the FORTRAN code , but they have to organise local and global information transfers to manage the distributed memory. This is why error detection may be time consuming and development cost can increase then the data-parallel strategy presents the advantage to hide interprocessor communication from the programmer, but the code has to be rewritten by adding directives [11, 12]. Neverthless, we will see in Section 3 that messages passing operations can be also hidden from the scientists in specialised subroutine that encapsulates the calls to standard or vendor proprietary communication library.

After introducing the OPA model and it's main numerical characteristics in Section 2, the different parallelization choices and their implementation are discussed in Section 3. The strategies adopted for the elliptic equation associated with the surface pressure gradient are presented in Section 4. Real life and academic experiments are assessed in Section 5 to validate the massively parallel code resulting.

2. The OPA model

2.1. Model equations

The ocean is a fluid which can be described by the Navier-Stokes equations plus the following additional hypothesis which are made from scale considerations : (a) spherical Earth approximation; (b) thin-shell approximation; (c) turbulent closure hypothesis; (d) Boussinesq hypothesis; (e) hydrostatic hypothesis; (f) Incompressibility hypothesis. In oceanography, we refer to this set of equations as Primitive Equations (hereafter PE).

It is useful to solve PE in various curvilinear coordinate systems. Indeed, in many ocean circulation problems, the flow field has regions of enhanced dynamics (i.e. western boundary currents, equatorial currents, or oceanic fronts) which extend in a relatively small fraction of the physical domain. Added efficiency can be earned by improving resolution in such regions.

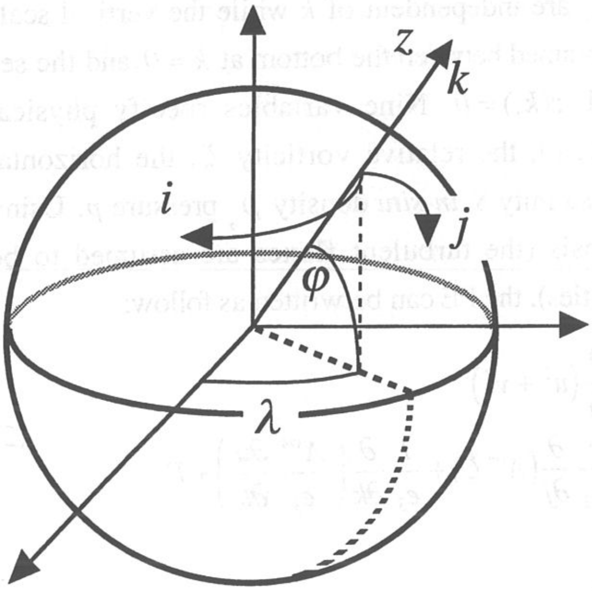


Figure 1 - The geographical coordinate system (λ,φ,z) and the survilinear coordinate system (i,j,k) on the sphere.

The common geographical coordinate system has a singular point at the North Pole which cannot be easily treated in a global model without filtering. One solution consists in introducing an appropriate coordinate transformation which shifts the singular point on land [13]. An efficient way of introducing an appropriate coordinate transform consists in using tensorial formalism. Such a formalism is suited to any multi-dimensional curvilinear coordinate system. Ocean modellers mainly use three-dimensional orthogonal grids on the sphere, with conservation of the local vertical. Herein we will give the simplified equations for this particular case.

Let (i, j, k) be a set of orthogonal curvilinear coordinates on the sphere associated with a set of unit orthogonal vectors **(i,j,k)** linked to the Earth such that **k** is the local upward vector (Fig. 1). A geographical position is defined by the latitude $\varphi(i, j)$, the longitude $\lambda(i, j)$ and the distance from the centre of the earth $a + z(k)$ where a is the earth radius and z the altitude above a reference sea level (Fig. 1). Using hypothesis (b), the local deformation of the curvilinear coordinates system is given by three scale factors :

$$
\begin{aligned}
e_1 &= a\left[\left(\frac{\partial\lambda}{\partial i}\cos\varphi\right)^2 + \left(\frac{\partial\varphi}{\partial i}\right)^2\right]^{1/2} \\
e_2 &= a\left[\left(\frac{\partial\lambda}{\partial j}\cos\varphi\right)^2 + \left(\frac{\partial\varphi}{\partial j}\right)^2\right]^{1/2} \qquad (1) \\
e_3 &= \left(\frac{\partial z}{\partial k}\right)
\end{aligned}
$$

The resulting horizontal scale factors e_1 and e_2 are independent of k while the vertical scale factor is a single function of k. The ocean is contained between the bottom at $k=0$ and the sea surface at $k=k_o$ where $z(0)=-H(i,j)$ and $z(k_o)=0$. Nine variables specify physical condition of the ocean: the velocity $\mathbf{U}=(u,v,w)$, the relative vorticity ζ, the horizontal divergence χ, the potential temperature T, the salinity S, *in-situ* density $\rho_{,}$ pressure p. Using the most standard turbulent closure hypothesis (the turbulent fluxes are assumed to be proportional to the gradient of large scale quantities), the PE can be written as follow:

$$\begin{aligned}\frac{\partial u}{\partial t} &= +(\zeta+f)v-\frac{1}{e_3}w\frac{\partial u}{\partial k}-\frac{1}{2e_1}\frac{\partial}{\partial i}(u^2+v^2)\\ &\quad -\frac{1}{\rho_o e_1}\frac{\partial p}{\partial i}+\frac{1}{e_1}\frac{\partial}{\partial i}(A^{lm}\chi)-\frac{1}{e_2}\frac{\partial}{\partial j}(A^{lm}\zeta)+\frac{1}{e_3}\frac{\partial}{\partial k}\left(\frac{A^{vm}}{e_3}\frac{\partial u}{\partial k}\right)+F_u\end{aligned} \tag{2}$$

$$\begin{aligned}\frac{\partial v}{\partial t} &= -(\zeta+f)u-\frac{1}{e_3}w\frac{\partial v}{\partial k}-\frac{1}{2e_2}\frac{\partial}{\partial j}(u^2+v^2)\\ &\quad -\frac{1}{\rho_o e_2}\frac{\partial p}{\partial j}+\frac{1}{e_2}\frac{\partial}{\partial j}(A^{lm}\chi)+\frac{1}{e_1}\frac{\partial}{\partial i}(A^{lm}\zeta)+\frac{1}{e_3}\frac{\partial}{\partial k}\left(\frac{A^{vm}}{e_3}\frac{\partial v}{\partial k}\right)+F_v\end{aligned} \tag{3}$$

$$\frac{\partial p}{\partial k}=-\rho\, g\, e_3 \tag{4}$$

$$\frac{\partial w}{\partial k}=-\chi\, e_3 \tag{5}$$

$$\begin{aligned}\frac{\partial T}{\partial t} &= -\frac{1}{e_1e_2}\left[\frac{\partial}{\partial i}(e_2 Tu)+\frac{\partial}{\partial j}(e_1 Tv)\right]-\frac{1}{e_3}\frac{\partial}{\partial k}(Tw)\\ &\quad +\frac{1}{e_1e_2}\left[\frac{\partial}{\partial i}\left(A^{lT}\frac{e_2}{e_1}\frac{\partial T}{\partial i}\right)+\frac{\partial}{\partial j}\left(A^{lT}\frac{e_1}{e_2}\frac{\partial T}{\partial j}\right)\right]+\frac{1}{e_3}\frac{\partial}{\partial k}\left(\frac{A^{vT}}{e_3}\frac{\partial T}{\partial k}\right)+F_T\end{aligned} \tag{6}$$

$$\begin{aligned}\frac{\partial S}{\partial t} &= -\frac{1}{e_1e_2}\left[\frac{\partial}{\partial i}(e_2 Su)+\frac{\partial}{\partial j}(e_1 Sv)\right]-\frac{1}{e_3}\frac{\partial}{\partial k}(Sw)\\ &\quad +\frac{1}{e_1e_2}\left[\frac{\partial}{\partial i}\left(A^{lT}\frac{e_2}{e_1}\frac{\partial S}{\partial i}\right)+\frac{\partial}{\partial j}\left(A^{lT}\frac{e_1}{e_2}\frac{\partial S}{\partial j}\right)\right]+\frac{1}{e_3}\frac{\partial}{\partial k}\left(\frac{A^{vT}}{e_3}\frac{\partial S}{\partial k}\right)+F_S\end{aligned} \tag{7}$$

$$\rho=\rho(T,S,z) \tag{8}$$

where ρ_o is a reference density, p is the pressure, f is the Coriolis acceleration ($f=2\,\Omega\sin\varphi(i,j)$ where W is the Earth angular velocity), g is the acceleration of gravity, A^{lm}, A^{vm}, A^{lT} and A^{vT} are the lateral and vertical eddy viscosity and diffusivity coefficients, F_u, F_v, F_T and F_S are the surface forcing fluxes of momentum heat and salt prescribed from observations or from an atmospheric model, and the relative vorticity ζ and the horizontal divergence χ are given by:

$$\zeta = \frac{1}{e_1 e_2}\left[\frac{\partial(e_2 v)}{\partial i} - \frac{\partial(e_1 u)}{\partial j}\right] \tag{9}$$

$$\chi = \frac{1}{e_1 e_2}\left[\frac{\partial(e_2 u)}{\partial i} + \frac{\partial(e_1 v)}{\partial j}\right] \tag{10}$$

Equation (8) represents the non-linear state equation, usually chosen as the UNESCO equation of state as derived by Jackett and McDougall [14]. It gives in situ density as a function of potential temperature, salinity and depth as pressure in decibars is approximated in (8) by depth in meters.

In addition to Eqs. (2) to (8), a parameterization of convection must be added. Indeed, due to the hydrostatic hypothesis, convective processes have been removed from the initial Navier-Stokes equations. They can be parameterized in three different ways in the OPA model. First, a non-penetrative convective adjustment algorithm can be used which mixes downwards instantaneously the statically unstable portion of the water column [15]. Second, the vertical eddy coefficients can be assigned to be very large (a typical value is $1\, m^2 s^{-1}$) in regions where the stratification is unstable. Last, a 1.5 turbulent closure scheme can be used. This scheme has been embedded in OPA by Blanke and Delecluse [16] to represent the various vertical turbulent mixing regime encountered in the ocean. It can also deal with statically unstable density profile in a way similar to the second method as it produces large vertical eddy coefficients in case of static unstabilities [17].

The ocean configuration is defined by specifying the depth field $H(i,j)$. Lateral boundaries may be either closed or periodic. Otherwise, arbitrary specifications of the coastline, bottom topography, and connectedness are allowed. The boundary conditions are that there is neither flow nor flux of heat or salt across solid boundaries. In particular, at the ocean bottom, the flow is required to be parallel to the slope, i.e.,

$$w = -\frac{u}{e_1}\frac{\partial H}{\partial i} - \frac{v}{e_2}\frac{\partial H}{\partial j} \quad \text{at } k = 0 \tag{11}$$

Additional momentum boundary conditions are needed for the boundary layers. Usually, a no-slip condition ($u = v = 0$) is imposed on lateral walls while the momentum fluxes are either zero or specified by a linear or quadratic law at the ocean bottom.

The surface kinematics boundary condition depends on the way the surface pressure gradient (SPG) is evaluated. Indeed, the total pressure at a given depth z is composed of a surface pressure p_s at a reference geopotential surface ($z = 0$) and a hydrostatic pressure p_h:

$p(i,j,z,t) = p_s(i,j,t) + p_h(i,j,z,t)$. The latter is computed by integrating (4), but the former requires a more specific treatment. Two strategies can be considered: (*i*) the introduction of a new variable *h*, the free-surface elevation (link to p_s by: $p_s = \rho g \eta$), for which a prognostic equation can be established and solved ; (*ii*) the assumption that the ocean surface is a rigid lid, on which the pressure (or its horizontal gradient) can be diagnosed. When the former strategy is used, a solution of the free-surface elevation consists in the excitation of external gravity waves. The flow is barotropic and the surface moves up and down with gravity as the restoring force. The phase speed of such waves is high (some hundreds of metres per second) so that the time step would have to be very short if they were present in the model. Several methods exist to overcome this problem, sub-cycling or implicit time scheme [18, 19], or introduction of a force which selectively damps external gravity waves that are resolved spatially but not temporally [20]. The latter strategy filters these waves as the rigid lid approximation implies $\eta = 0$, i.e. the sea surface is the surface $z = 0$. This well-known approximation increases the surface wave speed to infinity and modifies certain other long-wave dynamics (e.g. barotropic Rossby or planetary waves). In OPA, the two strategies can be used [20]. As they both leads to a similar algorithm (inversion of an elliptic equation), we only detail here the strategy (*ii*).

Assuming that the ocean surface is a rigid lid on which a pressure p_s is exerted, the surface kinematics boundary condition reduces to $w = 0$ at the sea surface. From (5) and (11), it can be shown that the vertically integrated flow $H\overline{\mathbf{U}}_h$ is nondivergent (where the overbar indicates a vertical average over the whole water column). Thus, $H\overline{\mathbf{U}}_h$ can be derived from a volume transport streamfunction ψ :

$$\overline{\mathbf{U}}_h = \frac{1}{H}(\mathbf{k}\times\nabla\psi) = \left(-\frac{1}{H e_2}\frac{\partial\psi}{\partial j}\,,\,+\frac{1}{H e_1}\frac{\partial\psi}{\partial i}\right) \qquad (12)$$

As p_s does not depend on depth, its horizontal gradient is obtained by forming the vertical average of (2) and (3):

$$\frac{1}{\rho_o}\nabla_h p_s = \overline{\mathbf{M}} - \frac{\partial\overline{\mathbf{U}}_h}{\partial t} = \left(\overline{M_u} + \frac{1}{H e_2}\frac{\partial}{\partial j}\left(\frac{\partial\psi}{\partial t}\right),\, \overline{M_v} - \frac{1}{H e_1}\frac{\partial}{\partial i}\left(\frac{\partial\psi}{\partial t}\right)\right) \qquad (13)$$

Here $\mathbf{M} = (M_u, M_v)$ represents the collected contributions of the Coriolis, hydrostatic pressure gradient, non-linear and viscous terms in (2) and (3). The time derivative of ψ is the solution of an elliptic equation (14) which is obtained from the vertical component of the curl of (13) :

$$\frac{\partial}{\partial i}\left[\frac{e_2}{H e_1}\frac{\partial}{\partial i}\left(\frac{\partial\psi}{\partial t}\right)\right] + \frac{\partial}{\partial j}\left[\frac{e_1}{H e_2}\frac{\partial}{\partial j}\left(\frac{\partial\psi}{\partial t}\right)\right] = \frac{\partial}{\partial i}\left(e_2\overline{M_v}\right) - \frac{\partial}{\partial j}\left(e_1\overline{M_u}\right) \qquad (14)$$

Using the proper boundary conditions, (14) can be solved to find $\partial\psi/\partial t$ and thus using (13) the horizontal surface pressure gradient. Nevertheless, a difficulty lies in the determination of the boundary condition on $\partial\psi/\partial t$. The boundary condition on velocity is that there is no flow normal to a solid wall, i.e. the coastlines are streamlines. Therefore (14) is solved with the following Dirichlet boundary condition: $\partial\psi/\partial t$ is constant along each coastline of the same continent or of the same island. When all the coastlines are connected (there are no islands), the constant value of $\partial\psi/\partial t$ along the coast can be arbitrarily chosen to be zero. When islands are present in the domain, the value of the barotropic streamfunction will generally be different for each island and for the continent, and will vary with respect to time. So, the boundary condition is: $\psi = 0$ along the continent and $\psi = \mu_n$ along island n ($1 \le n \le Q$), where Q is the number of islands present in the domain and μ_n is a time dependent variable. As (14) is linear, its solution $\partial\psi/\partial t$ can be decomposed as follows :

$$\frac{\partial\psi}{\partial t} = \frac{\partial\psi_o}{\partial t} + \sum_{n=1}^{n=Q} \frac{\partial\mu_n}{\partial t}\psi_n \tag{15}$$

where $\partial\psi_o/\partial t$ is the solution of (14) with $\partial\psi_o/\partial t = 0$ along all the coastlines, and where ψ_n are the solution of (14) without the time derivative, with the right-hand side equal to 0, and with $\psi_n = 1$ along the island n and $\psi_n = 0$ along the other boundaries. Using this decomposition and evaluating the circulation of the time derivative of the vertical average velocity field along a closed contour around each island, an equation for $\partial\mu_n/\partial t$ is obtain:

$$\left[\oint_n \frac{1}{H}[\mathbf{k}\times\nabla\psi_m]\cdot\mathbf{dl}\right]_{\substack{1\le m\le Q\\ 1\le n\le Q}} \left(\frac{\partial\mu_n}{\partial t}\right)_{1\le n\le Q} = \left(\oint_n \left[\overline{\mathbf{M}} - \frac{1}{H}\left[\mathbf{k}\times\nabla\left(\frac{\partial\psi_o}{\partial t}\right)\right]\right]\cdot\mathbf{dl}\right)_{1\le n\le Q} \tag{16}$$

The left hand side of (16) is a product of a small size matrix by a vector $(\partial\mu_n/\partial t)_{1\le n\le Q}$. The matrix is independent of time and can be calculated and inverted once.

2.2. Numerical characteristics

The ocean mesh (i.e. the position of all the scalar and vector points) is defined by the transformation that gives (λ,φ,z) as a function of (i,j,k). Special attention has been given to the homogeneity of the solution in the three space directions. The arrangement of variables is the same in all directions (Fig. 2). It consists in cells centered on scalar points (T, S, p, ρ, χ) with vector points (u, v, w) defined in the centre of each face of the cells. This is the generalization to three dimensions of the well-known Arakawa- C grid [21]. The scale factors are defined as the local analytical value provided by (1). As a result, the mesh on which partial derivatives $\partial/\partial i$, $\partial/\partial j$, and $\partial/\partial k$ have to be evaluated is an uniform mesh which grid size is the unity [22]. The numerical techniques used to solve the model equations are based on the

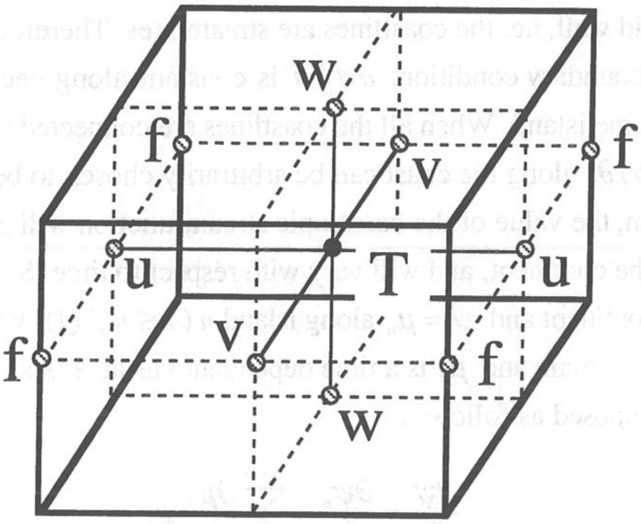


Figure 2 - C grids in Arakawa's classification.

traditional, centered second-order finite difference approximation. Note that the partial derivatives of scale factors are also computed by centered second-order finite difference approximation. This preserves the symmetry of the discrete set of equations and therefore allows to satisfy many of the continuous properties [7]. Further detail about the model and the numerical schemes used are given in the model reference manual [7].

For the non-diffusive processes, the time stepping is achieved with the well known leapfrog (three-level centered) scheme [21]. The computational noise associated with this scheme (a splitting of the solution on even and odd timesteps) is controlled through the use of an Asselin time filter (first designed by Robert [23] and more comprehensively studied by Asselin [24]). For diffusive processes, an Euler forward scheme is used. Unfortunately, the scheme is often unstable for vertical diffusive terms due to the relative strength of the vertical eddy coefficients compared to the vertical scale factors, and so, for these, an Euler backward scheme can be used. The latter scheme, as the non-penetrative convective adjustment algorithm, requires information over each entire individual water column.

The last algorithmic choice associated with the model equations concerns the way (14) is solved. With the second order finite difference approximation chosen, (14) leads to a matrix equation of the form:

$$\mathbf{E}\,\mathbf{x} = \mathbf{b} \tag{17}$$

where $\mathbf{E}$ is a positive-definite symmetric sparse matrix, and $\mathbf{x}$ and $\mathbf{b}$ are the vector representation of $\partial\psi/\partial t$ and of the right hand side of (14), respectively. Due to the huge size of the problem (currently more than 50,000 unknowns), (17) is solved using iterative methods such as a Successive Over-Relaxation (SOR) or Preconditioned Conjugate Gradient (PCG) algorithm. PCG method is further detailed in section 4. At this stage, the important point to note is that the resolution of (17) requires information over the whole horizontal domain.

The numerics briefly described above allows to identify the space dependencies associated with the resolution of the PE. Computing SPG requires a knowledge over the whole horizontal domain (horizontal implicit operators), the vertical physics (thereafter VP that includes parameterization of convective processes, 1.5 vertical turbulent closure, time stepping on the vertical diffusion terms) involves each ocean water column as a whole (vertical implicit operators), while for the remaining part of the model (hereafter PE') the computation remains local, depending only of the point and its very neighbours (fully explicit for all space operators).

3. Parallelization choices.

In domain decomposition methods field, it has been noticed [25] that an efficient parallelization strategy involves to split the global domain into subdomains by preserving the locality (the locality characterises the spatial extension of an operator, *e. g.* the number of neighbouring grid points involved in a computation) and exploiting a coarse granularity of the calculations, *i. e.* with tasks performing a number of operations one magnitude higher than the quantity of data to be recovered on the in or output of these tasks. Such a granularity ensures the scalability of the algorithm *i.e.* the whole CPU time continues to decrease as the number of processors increases. As the order of the operators used in the explicit finite-difference algorithms of OPA remain weak, calculations are local and depend only on the point and its neighbours. The dependencies problem is therefore equivalent to a non homogeneous boundary problem. To inverse an implicit operator as in (14), a large portion of the domain is involved: there is no locality of the calculations and thus no intrinsic parallelism. In the latter case, it can be useful to use specific numerical methods [10] to change the dependencies problem into an open boundary problem. In the case of OPA model, dependencies problem is different in PE', than in SPG or VP, so they have to be analysed and parallelized in a separate, but coherent, way [26, 27].

Granularity of parallel tasks for explicit algorithms of PE' is assumed to be function of the ratio of the number of inner points over the number of interface points. Convexity considerations prove cubic splitting presents a better granularity and scalability in the context of parallel computing applied to isotropic processes, but not in Geophysical Fluid Dynamics whereas the number of grid points of the vertical is one magnitude less than the latitude or longitude ones. In such a context, pencil splitting (Fig. 3) minimizes the communication cost

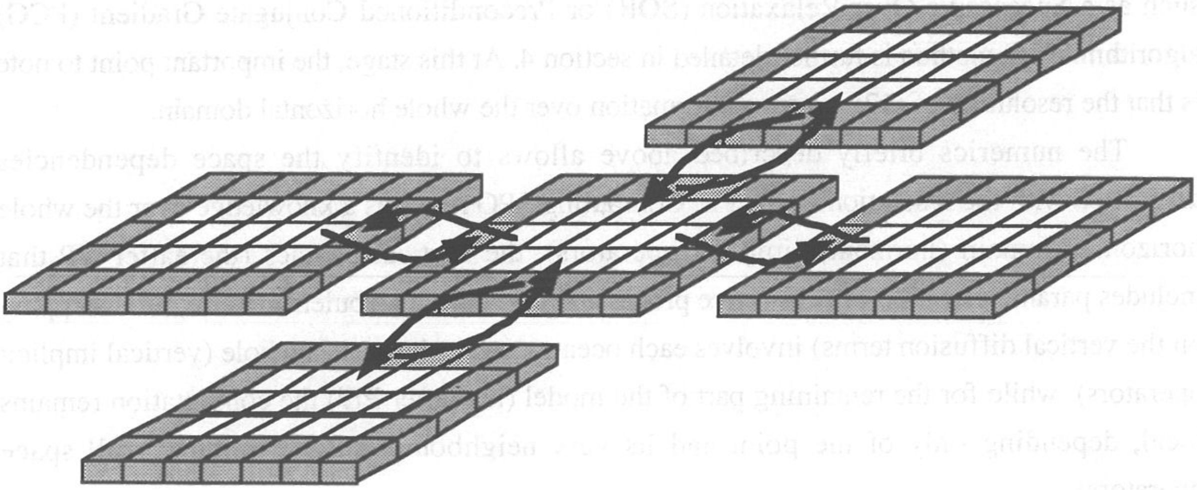


Figure 3 - Pencil splitting with the additional outer halos and the alternated direction communication strategy.

for PE' by reducing the number of interfaces from 6, for the cubic splitting, to 4. The pencil splitting ensures better granularity and scalability in massively parallel computing field than the slice splitting used by Clare and Stevens [28] or Andrich *et. al.* [29] to exploit multitasking technique on vector machines. Such a strategy is an elegant solution to solve the dependencies problem of VP and preserves the boundary conditions of the ocean on the surface and the bottom (11) and appears to be natural in Ocean [30] and in Atmosphere [9].

The OPA finite-difference algorithms are solved using an Arakawa-C grid : local operator are symmetric, but the FORTRAN expression becomes non-symmetric. So, the implementation of the local boundary conditions for the global operator is assumed to be the sum of local operators whose the local boundary conditions could be non-coherent. These features naturally lead to the easiest programming solution: a data sub-structuring with overlapping boundaries (Fig. 3). Each processor computes a subdomain. This subdomain owns inner points (they are part of the study domain) plus overlapping interface points in an additional outer halo (they are also inner points for neighbouring subdomains). During each time step, overlapping area stocks boundary conditions for computation on inner points of the subdomain. When the iteration ends, a synchronised communication phase starts. The overlapping area is declared to receive computed data sent from the processors that manage adjacent subdomains and *vice versa.* During such a communication phase, the interface matching conditions are related to usual Dirichlet boundary conditions.

As the information is located on the boundary of each subdomain, transfers concern only a subdomain and its neighbours. In such a case, the alternated direction communication mode is a very appropriated solution. Each processor communicates with its neighbours in one direction (east-west), then in the other one (north-south). These operations stay local and can be done in a parallel way. A hierarchical and centralized point of view where a responsible gets the whole information and then diffuse it to each subdomain is avoided and the MPP code scalability is preserved. In Earth Sciences context in general, and in our parallel system context in particular, memory appears to be a more critical resource than communication time. So, the size of the overlapping interface has been parameterized to be configured as a function of the order of numerical scheme. In OPA code, schemes are second order, so a one grid point overlapping interface and one communication phases per time step are sufficient to determinate most of local operators. For specifics operators that present large spatial extension, like isopycnal or biharmonic diffusion operators..., specifics communication phases have been implemented. This strategy optimises memory and provides a flexible framework for future evolution of OPA code.

This analysis leads to solve in the same way the dependency problem of each subdomains and the lateral boundary conditions of the global domain. Last but not least, a generic lateral boundary condition operator (LBC), that solve (Dirichlet, symmetric, periodic and bi-periodic) geographic boundary conditions and the dependences between subdomains, is defines and isolated in an unique subroutine. Such an approach preserves the readability of the OPA code for future developments and research actions (the message passing library is hidden from the scientist) and ensures portability and performance on different systems (the message passing library can be changed without difficulty). Moreover the work to maintain the code is minimizing because the initial code and the MPP one are similar modulo the communication library called in the LBC subroutine.

4. A domain decomposition solver to compute the surface pressure gradient.

As mentioned in section 3, the calculation of the SPG is a very specific part of the PE model. Indeed, whatever the hypothesis used to compute SPG (free-surface or rigid lid hypothesis), this term of the equations leads to a two dimensional horizontal problem that has to be solved with an iterative algorithm, either in time for subcycling time scheme [18], or in space for the other methods as these latter all lead to a huge elliptic equation solved by iterative methods (direct solvers [31] are not efficient for large numbers of grid points while fast Fourier transform methods [32] are not suitable for complex geometry of the ocean domain). So, whatever the choice done, one will have to perform a computation on horizontal fields typically a few hundred time by time step with a few communication phases for each. The ratio between computation and communication will therefore be a crucial point for SGP calculation and appear to stay strongly machine dependent. One looks for an algorithm that minimizes the communication phase and/or the number of iterations in order to obtain good granularity tasks to benefit from massively parallel computers.

In this section, we describe the two algorithms used to solve the elliptic equation (17). The former is an adaptation to massively parallel computer of the old preconditioned conjugate gradient (PCG) algorithm [33] while the later is a new development based on a Finite Elements Tearing and Interconnecting (FETI) method [34].

Note that since we consider the rigid lid version of the OPA model, we have also to compute the islands contribution given by (16). As it has to be done once by time step, this is not a critical point if the number of islands is small. Then each processor has to know the inverse of the small size matrix (16). To compute it during the initial part of simulation, a global communication is required to compute dot product operations associated with the circulations along each island that define the different coefficients of **A** (16). The coefficient of the time dependent vector **B**, defined by a circulation in (16), are computed in a same way at each time step. Moreover, the island contribution is no more required in free surface formulation [19, 20].

4.1. A parallel Preconditioned Conjugated Gradient.

The diagonal PCG algorithm used in OPA model may be described as follows [29, 33]. Given **M** the diagonal of **E**, $\mathbf{x}^0$, the starting point (usually extrapolated from the two previous time step calculations), and the gradient at the starting point $\mathbf{r}^0 = \mathbf{b} - \mathbf{E}\mathbf{x}^0$, choose an initial descent direction $\mathbf{d}^0 = \mathbf{M}^{-1}\mathbf{r}^0$, and an initial trial step length $\alpha_0 = \left\langle \mathbf{M}^{-1}\mathbf{r}^0, \mathbf{r}^0 \right\rangle$. Repeat the following steps until the norm of the gradient is reduced below some threshold (usually $\left\langle \mathbf{M}^{-1}\mathbf{r}^n, \mathbf{r}^n \right\rangle \le \varepsilon^2 \left\langle \mathbf{M}^{-1}\mathbf{b}, \mathbf{b} \right\rangle$, where ε is the precision that is required).

$$\beta_{n-1} = \left\langle \mathbf{E}\mathbf{d}^{n-1}, \mathbf{d}^{n-1} \right\rangle \tag{18a}$$

$$\mathbf{x}^n = \mathbf{x}^{n-1} + \left(\alpha_{n-1}/\beta_{n-1}\right)\mathbf{d}^{n-1} \tag{18b}$$

$$\mathbf{r}^n = \mathbf{r}^{n-1} - \left(\alpha_{n-1}/\beta_{n-1}\right)\mathbf{E}\mathbf{d}^{n-1} \tag{18c}$$

$$\alpha_n = \left\langle \mathbf{M}^{-1}\mathbf{r}^n, \mathbf{r}^n \right\rangle \tag{18d}$$

$$\mathbf{d}^n = \mathbf{M}^{-1}\mathbf{r}^n + \left(\alpha_n/\alpha_{n-1}\right)\mathbf{d}^{n-1} \tag{18e}$$

Where $\langle\ ,\ \rangle$ represents the canonical dot product.

The parallelization of the PCG algorithm is rather straightforward. Indeed, since $\mathbf{M}^{-1}$ is a diagonal matrix, computing (18b) and (18e) only requires the knowledge of variables at the very grid point the computation is done. Communication phases are only required in the calculation of (18a), (18c) and (18d) due to the matrix-vector product $\mathbf{E}\mathbf{d}^{n-1}$ and the two dot products. $\mathbf{E}\mathbf{d}^{n-1}$ is computed using the same overlapping strategy as those used on PE' in order to solve the dependencies problem as an open Dirichlet boundary one. It requires an alternated direction communication phase to update $\mathbf{d}^{n}$ on the overlapping area just after (18c). The local dot products are computed in parallel on each interior subdomain and sum over the whole domain through a reduction-diffusion operation to generate the coefficients α_n and β_{n-1}.

At each iteration of PCG algorithm requires two matrix-vector products, two dot products, three linear operations and three communication phases. The PCG algorithm can be characterised by a weak granularity. For a given number of subdomains, computations have to be done on a very large size experiment to obtain a good parallel efficiency. Beare and Stevens [35] have proposed to improve parallel efficiency by adding extra halo to the overlapping interface applied to the explicit free surface computation, the same idea can be applied to SPG solved with PCG.

The number of PCG iterations can be reduced by using preconditioning methods based on local Approximate-Inverse matrix that preserve the locality of the equations (18d) to (18e) [4]. Nevertheless, the PCG-algorithm convergence slows down with increasing matrix size [36], *i.e.* in our case with the number of horizontal grid points. Moreover, PCG can have important difficulties to capture the solution, because the residual can oscillate and never reach the required value. Several studies, based on analysis on the eighenvalues spectrum of different operators, give numerical explanations to this phenomenon [37, 38]. The PCG algorithm captures, in a rather easy way, the modal components of solution corresponding to large values and the convergence speed of the PCG algorithm depends on the ratio $(K_{(\mathbf{E})}^{1/2} - 1)/(K_{(\mathbf{E})}^{1/2} + 1)$ where $K_{(\mathbf{E})}$ is the condition number, i.e. the ratio of the largest eighenvalue of $\mathbf{E}$ over the smallest one. The five points discretization of (14) leads to a matrix which coefficients depend on the aspect ratio of the horizontal mesh and of the bathymetry. For a real application, this often leads to a large $K_{(\mathbf{E})}$ so that the convergence speed can be very low. Moreover, the condition number depends on number of points in the domain and it's easy to verify that $K_{(\mathbf{E})}$ increases with the grid size when $\mathbf{E}$ is a Laplacian operator. So, the conjugate effects of these numerical characteristics and the accumulation of cut off penalise the convergence of the PCG-algorithm in the context of large size computations. This is why this strategy can be too time consuming and can be a very critical point to use parallel model for high resolution climatic investigations.

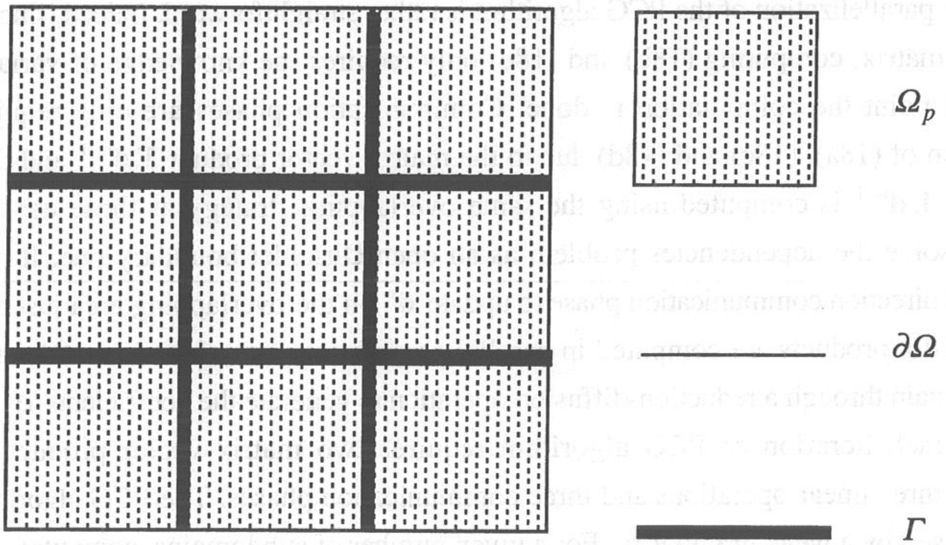


Figure 4 - N_p subdomains partitioning, $(\Omega_p)_{p=1,N_p}$, and the non overlapping interface Γ.

4.2. A domain decomposition solver based on the FETI method.

The most standard domain decomposition methods to solve an elliptic equation are the Schwarz algorithm [39], the Schur complement method [40] and the FETI method (Finite Elements Tearing and Interconnecting) also called the dual Schur complement method [34]. As the discretized SPG equation (17) is computed only on f-points of the Arakawa C-grid (Fig. 2), solving (17) by a non-overlapping method, *i. e.* the two latter, is easy to implement and present a better intrinsic parallelism than the Schwarz alternating procedure [10]. Navon and Cai [41] have used the Schur complement method to solve the shallow-water equations, nevertheless the FETI method is a non-conforming method where the different subdomains can have different discretizations: such an algorithm defines a natural field to implement a local zooming approach [42] adapted to solve heterogeneous physical phenomena of Earth Sciences [43]. So the local grid resolution could be adapted to observe consistent space scales for a reasonable computing cost: a good knowledge of ocean features will help us to choose interesting area. Last, we would like to insist on the point that the FETI approach is a very general approach to solve any elliptic equation associated with SPG.

The global domain Ω which boundary is $\partial\Omega$ is divided into a set of N_p subdomains $(\Omega_p)_{p=1,N_p}$ which boundaries are $\left(\partial\Omega_p\right)_{p=1,N_p}$, and a non-overlapping interface between the subdomains (Fig. 4):

$$\Gamma = \bigcup_{p=1,N_p} \partial\Omega_p - \partial\Omega$$

For purpose of simplification the elliptic equation (14) is rewritten as (19a) with Dirichlet boundary conditions (19b) :

$$E(x) = \nabla(\kappa \nabla(x)) = b \quad \text{on} \quad \Omega \tag{19a}$$

$$x = x^{\partial\Omega} \quad \text{on} \quad \partial\Omega \tag{19b}$$

Let us consider the set of the local second order elliptic operators $\left(E_p\right)_{p=1,N_p}$ associated to the SPG operator E on $(\Omega_p)_{p=1,N_p}$, the Dirichlet-boundary problem satisfied by the global field x associated to the source term b (19) is equivalent, in its variational form [34], to the set of N_p local problems with the continuity conditions, (20c) and (20d), on the interface Γ :

$$E_p\left(x_p\right) = b_p \quad \text{on} \quad \Omega_p \tag{20a}$$

$$x_p = x_p^{\partial\Omega} \quad \text{on} \quad \partial\Omega \cap \partial\Omega_p \tag{20b}$$

$$x_p = x_{p_i} \quad \text{on} \quad \partial\Omega_p \cap \partial\Omega_{p_i} \tag{20c}$$

$$\kappa_p \nabla\left(x_p\right) n_p + \kappa_{p_i} \nabla\left(x_{p_i}\right) n_{p_i} = 0 \ \text{on} \quad \partial\Omega_p \cap \partial\Omega_{p_i} \tag{20d}$$

where x_p, b_p, κ_p and $x_p^{\partial\Omega}$ are the values of x, b, κ and $x^{\partial\Omega}$ in the subdomain Ω_p, n_p is the outer normal vector of $\partial\Omega_p \cap \Gamma$. As $\left(\Omega_{p_i}\right)_{i=1,N_a^p}$ is the set of the N_a^p adjacent subdomains of Ω_p, x_{p_i} and κ_{p_i} are the values of x and κ in the subdomain Ω_{p_i}, n_{p_i} is the outer normal vector of $\partial\Omega_{p_i} \cap \Gamma$. Let us introduce the Lagrange multiplier associated to the continuity condition (20c) :

$$\kappa_p \nabla\left(x_p\right) n_p = s\,\lambda \quad \text{on} \quad \Gamma \cap \partial\Omega_p \tag{21}$$

where the factor $s = \pm 1$. The change of sign indicates that the outer normal derivatives of x_p and x_{p_i} are opposite each other.

The FETI methods consists in finding λ the value of the normal derivative of the fields x_p along the interfaces Γ for which the solution of the local boundary-value problems, (20a) and (20b), and that satisfy the matching condition (20c). Then, the global field x whose restriction x_p to each subdomain Ω_p is defined as the solution of the local Neumann problem is continuous and also the normal fluxes $\kappa_p \nabla\left(x_p\right) n_p$, this is therefore the solution of the global problem.

The centered second order Finite Differences procedure transforms the local boundary-value problem, (20a) and (20b), associated with the Neumann condition (21) and the matching condition (20c) into the algebraic system :

$$\mathbf{E_p}\,\mathbf{x_p} = \mathbf{b_p} - \mathbf{B_p^t}\,\mathbf{L} \tag{22a}$$

$$\sum_{p=1,N_p} \mathbf{B_p}\,\mathbf{x_p} = \mathbf{0} \tag{22b}$$

where $\mathbf{E_p}$, $\mathbf{x_p}$, $\mathbf{b_p}$ and $\mathbf{L}$ are respectively the local SPG matrix, the vectors associated with Finite Difference discretization of x_p, b_p and λ. $\mathbf{B_p}$ is a signed Boolean matrix which localizes a subdomain quantity to the interface Γ. The vector of the Lagrange multiplier $\mathbf{L}$ represents the interaction fluxes needed for gluing together the subdomains Ω_p along the interface. Although the global SPG matrix is symmetric positive definite, a set of N_n local matrices $\mathbf{E_p}$ may be indefinite when (20) has no Dirichlet condition on $\partial\Omega_p$. Thus, after renumbering the points of Ω_p, $\mathbf{E_p}$ can be partitioned as follow by considering

$$\mathbf{Range}(\mathbf{E_p}) \oplus \mathbf{Ker}(\mathbf{E_p}),$$

the solution of the associated local problems of the form (22a) may then be defined as the sum of two components (23a), one in $\mathbf{Range}(\mathbf{E_p})$ (23b) and the other in $\mathbf{Ker}(\mathbf{E_p})$

$$\mathbf{x_p} = \mathbf{E_p^+}\left(\mathbf{b_p} - \mathbf{B_p^t}\,\mathbf{L}\right) + \mathbf{K_p}\,\mathbf{c} \tag{23a}$$

$$\mathbf{K_p^t}\,\mathbf{b_p} - \mathbf{K_p^t}\,\mathbf{B_p^t}\,\mathbf{L} = \mathbf{0} \tag{23b}$$

where $\mathbf{E_p^+}$ is a symmetric generalised inverse, $\mathbf{K_p}$ is a basis of $\mathbf{Ker}(\mathbf{E_p})$ and the product $\mathbf{K_p}\,\mathbf{c}$ is a constant field. If we consider a $\mathbf{E_p}$ matrix that distinguishes :

$$\mathbf{E_p} = \begin{bmatrix} \mathbf{E_p^{11}} & \mathbf{E_p^{12}} \\ {}^{t}\mathbf{E_p^{12}} & \mathbf{E_p^{22}} \end{bmatrix} \quad \text{with} \quad rank(\mathbf{E_p}) = rank(\mathbf{E_p^{11}}) \tag{24a}$$

a generalized inverse $\mathbf{E_p^+}$ and a $\mathbf{K}_p$ matrix can be defined, respectively, as

$$\mathbf{E_p^+} = \begin{bmatrix} \mathbf{E_p^{11^{-1}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{K_p} = \begin{bmatrix} -\mathbf{E_p^{11^{-1}}}\,\mathbf{E_p^{12}} \\ \mathbf{I} \end{bmatrix} \tag{24b}$$

More details on computing the generalized inverse and null-space of a large-scale matrix can be found in [44]. If $\mathbf{E_p}$ is regular then :

$$\mathbf{E_p^+} = \mathbf{E_p^{-1}} \quad \text{and} \quad \mathbf{K_p} = \mathbf{0}$$

We can now substitute the value of $\mathbf{x_p}$ given by (23a) in (22b) to lead, with the condition (23b) to matrix equation satisfied by $\mathbf{L}$ and the dual Schur complement operator $\mathbf{D}$:

$$\begin{bmatrix} \mathbf{D} & -\mathbf{G} \\ -{}^{t}\mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{L} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{b^d} \\ -\mathbf{b^c} \end{bmatrix} \tag{25a}$$

$$\mathbf{D} = \sum_{p=1,N_p} \mathbf{B_p E_p^+ B_p^t} \text{ and } \mathbf{G} = \left[\mathbf{B_p K_p}\right]_{p=1,N_n} \tag{25b}$$

$$\mathbf{b^d} = \sum_{p=1,N_p} \mathbf{B_p E_p^+ b_p} \text{ and } \mathbf{b^c} = \left[\mathbf{K_p^t b_p}\right]_{p=1,N_n} \tag{25c}$$

This decomposition can be interpreted as a multi-level scheme where the Lagrange multiplier is split into two components associated to the vectors $\mathbf{c}$ which collects a local constant value within a subdomain and $\mathbf{L}$ which accounts for the local fluctuations within each subdomain [8]. Therefore, the problem to be solved is presented as a constrained maximization problem [34], the form of which is given using matrix-notations by :

$$\underset{\lambda}{Maximize} \Im(\lambda) = -\frac{1}{2}\lambda^t D\lambda + \lambda^t\left(b^d + Gc\right) \tag{26a}$$

$$Subject\, to\, G^t \lambda = b^c \tag{26b}$$

This constrained quadratic problem is then solved by means of a Preconditioned Conjugate Projected Gradient algorithm [45], hereafter PCPG. If $\mathbf{L^0}$ is chosen to satisfy the constraint equation $\mathbf{G^t L^0 = b^c}$, then for $n > 0$, $\mathbf{L^n}$ must satisfy $\mathbf{G^t L^n = 0}$ and a suitable projector is an operator that projects $\mathbf{L^n}$ onto the null space of $\mathbf{G^t}$. Let the projector $\mathbf{P}$ be defined by the following expression :

$$\mathbf{P} = \mathbf{I} - \mathbf{G}\left(\mathbf{G^t G}\right)^{-1}\mathbf{G^t} \tag{27}$$

As $\mathbf{Range(G) = Ker\left(G^t\right)}$ the local constants contributions are treated implicitly in this PCPG algorithm since they are associated with the projector $\mathbf{P}$, and the Lagrange multipliers contributions are treated iteratively. The solutions $\mathbf{(L,c)}$ has to satisfy (25) and are not interesting by themselves, any combination of these variables that satisfy equilibrium (25) is acceptable. Note that it has been demonstrated in [34], for the category of problems treated herein, the G matrix has full column rank and therefore this projector exists. The constrained maximization problem given in (26) is then equivalent to the linear system :

$$\mathbf{P}\left(\mathbf{D}\mathbf{L}-\mathbf{b}^{\mathbf{d}}\right)=\mathbf{0} \tag{28a}$$

$$\mathbf{G}^{\mathbf{t}}\mathbf{L}=\mathbf{b}^{\mathbf{c}} \tag{28b}$$

In summary the Preconditioned Conjugate Projected Gradient algorithm (hereafter PCPG) for solving the interface problem arising from the FETI method goes as follow: given $\mathbf{M}^{-1}$ an approximation of $\mathbf{D}^{-1}$, $\mathbf{P}$ the projector on to the null space of $\mathbf{G}^{\mathbf{t}}$ (Eqs. (24b), (25b) and (27)), $\mathbf{L}^{\mathbf{0}}$ the starting point ($\mathbf{L}^{\mathbf{0}}=\mathbf{G}\left(\mathbf{G}^{\mathbf{t}}\mathbf{G}\right)^{-1}\mathbf{b}^{\mathbf{c}}$ is chosen to satisfy $\mathbf{G}^{\mathbf{t}}\mathbf{L}^{\mathbf{0}}=\mathbf{b}^{\mathbf{c}}$ (25a)), the gradient at the starting point $\mathbf{r}^{\mathbf{0}}=\mathbf{b}^{\mathbf{d}}-\mathbf{D}\mathbf{L}^{\mathbf{0}}$ and the projected gradient $\tilde{\mathbf{r}}^{\mathbf{0}}=\mathbf{P}\mathbf{r}^{\mathbf{0}}$, choose an initial projected descent direction $\mathbf{d}^{\mathbf{0}}=\mathbf{P}\mathbf{M}^{-1}\tilde{\mathbf{r}}^{\mathbf{0}}$, and an initial trial step length $\alpha_0=\left\langle\mathbf{d}^{\mathbf{0}},\tilde{\mathbf{r}}^{\mathbf{0}}\right\rangle$. Repeat the following steps until the norm of the gradient is reduced below some threshold (we have chosen $\left\langle\mathbf{M}^{-1}\tilde{\mathbf{r}}^{\mathbf{n}},\mathbf{M}^{-1}\tilde{\mathbf{r}}^{\mathbf{n}}\right\rangle\leq\varepsilon^2\left\langle\mathbf{b},\mathbf{b}\right\rangle$, where ε is the precision that is required for the problem (28) and can be different to the precision required for the problem (17)).

$$\beta_{n-1}=\left\langle\mathbf{D}\mathbf{d}^{\mathbf{n-1}},\mathbf{d}^{\mathbf{n-1}}\right\rangle \tag{29a}$$

$$\mathbf{L}^{\mathbf{n}}=\mathbf{L}^{\mathbf{n-1}}+\left(\alpha_{n-1}/\beta_{n-1}\right)\mathbf{d}^{\mathbf{n-1}} \tag{29b}$$

$$\mathbf{r}^{\mathbf{n}}=\mathbf{r}^{\mathbf{n-1}}-\left(\alpha_{n-1}/\beta_{n-1}\right)\mathbf{D}\mathbf{d}^{\mathbf{n-1}} \tag{29c}$$

Project
$$\tilde{\mathbf{r}}^{\mathbf{n}}=\mathbf{P}\mathbf{r}^{\mathbf{n}} \tag{29d}$$

Precondition
$$\mathbf{m}^{\mathbf{n}}=\mathbf{M}^{-1}\tilde{\mathbf{r}}^{\mathbf{n}} \tag{29e}$$

Re-Project
$$\tilde{\mathbf{m}}^{\mathbf{n}}=\mathbf{P}\mathbf{m}^{\mathbf{n}} \tag{29f}$$

$$\alpha_n=\left\langle\tilde{\mathbf{m}}^{\mathbf{n}},\tilde{\mathbf{r}}^{\mathbf{n}}\right\rangle \tag{29g}$$

$$\mathbf{d}^{\mathbf{n}}=\tilde{\mathbf{m}}^{\mathbf{n}}+\left(\alpha_n/\alpha_{n-1}\right)\mathbf{d}^{\mathbf{n-1}} \tag{29h}$$

The first component of the solution (23a) is estimated from the equation (29b) :

$$\mathbf{x}_{+}^{\mathbf{n}}=\mathbf{E}_{\mathbf{p}}^{+}\left(\mathbf{b}_{\mathbf{p}}-\mathbf{B}_{\mathbf{p}}^{\mathbf{t}}\mathbf{L}^{\mathbf{n}}\right) \tag{29i}$$

As the projector $\mathbf{P}$ is solving the local constants contributions Gc in (26), the estimation of the local constants contributions in $\mathbf{x}^{\mathbf{n}}$ is obtained by considering the definition (27):

$$\mathbf{r}^{\mathbf{n}}-\tilde{\mathbf{r}}^{\mathbf{n}}=\mathbf{G}\left(\mathbf{G}^{\mathbf{t}}\mathbf{G}\right)^{-1}\mathbf{G}^{\mathbf{t}}\mathbf{r}^{\mathbf{n}}=-\mathbf{G}\mathbf{c}^{\mathbf{n}} \tag{29j}$$

Then, the estimation of the solution arises from (23a):

$$\mathbf{x}^{\mathbf{n}} = \mathbf{x}^{\mathbf{n}}_{+} - \mathbf{K}_{\mathbf{p}}\left(\mathbf{G}^{\mathbf{t}}\,\mathbf{G}\right)^{-1}\mathbf{G}^{\mathbf{t}}\,\mathbf{r}^{\mathbf{n}} \tag{29k}$$

The matrix **D** is a general matrix, it is not feasible to explicitly assemble it. This implies that a direct method cannot be used to solve the above interface problem (28). An efficient algorithm to solve (28) is the PCG method because once $\left(\mathbf{E}^{+}_{\mathbf{p}}\right)_{p=1,N_p}$ have been factored, matrix-vector products of the form **DL** can be efficiently performed using only forward and backward substitutions. Moreover, the size of this problem is equal to the sum of the number of points of the interface of each subdomain $\partial\Omega_p \cap \Gamma$, so this is a small size problem in comparizon of number of points over all the domain of simulation, the size of the problem (17). Furthermore, **D** is a first order differential operator, it presents a better intrinsic condition number than a second order differential operator as **E**, then the numerical behaviour of a PCG algorithm to solve (28) is better than to solve (17). An other interest of this methods consists in the decomposition of the matrix **D** and the right hand dual member $\mathbf{b}^{\mathbf{d}}$ in local matrices and local right hand members : this approach leads to a naturally distributed parallel algorithm based on fast local algebraic operations. As the size of the local problem (22a) is equal to the total number of points of the domain divided by the number of subdomains, (22a) can be solved by a direct method as LU factorization. This approach avoids the difficulties arising from the convergence of all the local problems in parallel and fully non-dependent way. As, the criterion convergence is apply to the dual equation (28) high precision is required to estimate $\mathbf{x}_{\mathbf{p}}$, this is an other advantage of a direct solver. It has been noticed that these method leads to a naturally distributed parallel algorithm where each subdomain is allocated to a different processor. This approach seems to be more efficient than the parallelization of the PCG algorithm because the parallel solution of these local problems leads to a parallelism with a coarse granularity.

In order to prevent convergence difficulties, different strategies have been proposed to ensure robustness and to speed up the convergence of the interface problem. As an approximation of the inverse of **D** (25b), a preconditioner matrix based on the local matrices $\mathbf{E}^{\mathbf{11}}_{\mathbf{p}}$ has been implemented :

$$\mathbf{M}^{-1} = \sum_{p=1,N_p} \mathbf{B}_{\mathbf{p}}\,\mathbf{E}^{\mathbf{11}}_{\mathbf{p}}\,{}^{\mathbf{t}}\mathbf{B}_{\mathbf{p}} \tag{30}$$

A Reconjugation method is applied to enforce, at iteration *n*, the direction vector of the Conjugated gradient to be conjugated to the *n-1* previous ones to avoid the propagation of numerical errors due to finite precision arithmetic, so the equation of the vector direction $\mathbf{d}^{\mathbf{n}}$ (29h) is changed into the conjugation relation [46] :

$$\mathbf{d^n} = \tilde{\mathbf{m}}^\mathbf{n} + \sum_{i=1,n} \left(\alpha_i / \alpha_{i-1}\right) \mathbf{d^{i-1}} \tag{31}$$

Moreover, in order to optimise the behaviour of the PCPG algorithm associated to FETI solver during time integration, a Krylov initialisation-correction procedure [47] has been developed. To solve the linear system equations satisfied by the dual operator (28), PCPG algorithm consists in computing a set of direction vectors that are conjugated with respect to the dot product associated with the matrix $\mathbf{D}$. The descent vectors generated $\left(\mathbf{d^i}\right)_{\mathbf{i=1,n}}$ constitute a bases of the Krylov space of vectors $\mathbf{H^n} = span\left(\mathbf{d^0}, \mathbf{d^1}, \ldots, \mathbf{d^n}\right)$. Let note n_d^{it} the number of PCPG iterations required to solve (28) at the time step it of an ocean simulation, the set of conjugate vectors $\left(\mathbf{d^i}\right)_{i=1,N_d^{N_{it}}}$, where $N_d^{N_{it}} = \sum_{it=1,N_{it}} n_d^{it}$, is built by accumulating the direction vectors computed for the solution of all previous problems (those have been solved at time step 1 to N_{it}). If a second hand member $\mathbf{b^d}$ and a first guess $\mathbf{L^0}$ are given, then the element $\mathbf{L^0_{opt}}$ of $\mathbf{L^0} + \mathbf{H}^{N_d^{N_{it}}}$ that optimises the initial residual can easily be computed and at each time step N_{it}, the iteration of the conjugate gradient algorithm can start from the optimal starting point:

$$\mathbf{L^0_{opt}} = \mathbf{L^0} - \sum_{i=1,N_d^{N_{it}}} \frac{\left\langle \mathbf{\tilde{r}^0}, \mathbf{d^i} \right\rangle}{\left\langle \mathbf{D d^i}, \mathbf{d^i} \right\rangle} \mathbf{d^i} \tag{32}$$

Mathematical proprieties, technical difficulties and a precise version of the FETI algorithm are detailed in [34]. It has been noticed that our implementation is really vectorized and uses BLAS subroutines [48].

5. Testing the parallel strategy.

A variety of test problems has been studied to determine the computational character of the parallel model by evaluating algorithmic and numerical solutions and the scalability of the code in the context of large size computations. The tests included : (p_1) realistic simulation of the western Mediterranean basin; (p_2) scaling test case based on academic ocean boxes. In (p_1), the model configuration designed by Herbaut *et al.* [49] to study the western Mediterranean Sea is used to validate the physical behaviour of the parallel model with a well-know experiment in the context of real life simulation. The computational behaviour of each parts of the parallel model have been considered to propose a performance model to describe the speed up of the parallel code versus the number of subdomains. A set of academic test cases has been designed

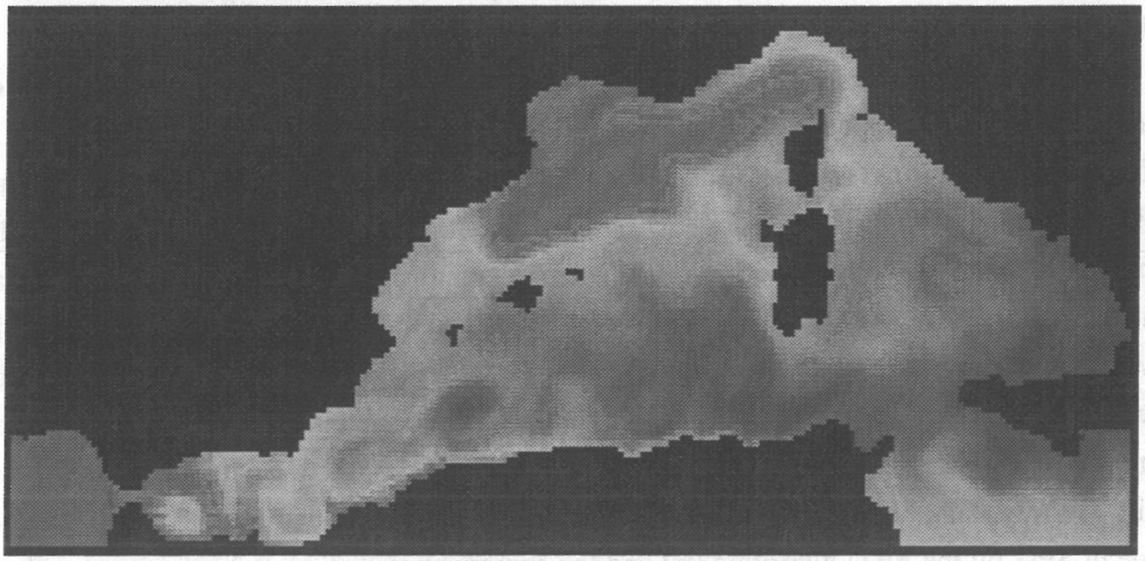


Figure 5 - Sea Surface Temperature of the western Mediterranean after 2 years of time integration on a Paragon system.

in (p_2) to illustrate the different numerical behaviour and scalability of the PCG and FETI algorithms when the domain size increases.

5.1. Modelling the western Mediterranean circulation on a massively parallel machine

The Western Mediterranean Sea (Fig. 5) is a pertinent example of the basin study complexity from physics and computer sciences point of view. It can be view as a small size ocean where most of phenomena that have effects on global ocean circulation are present: deep convection, coastal current instabilities, straits dynamics, interaction between large scale circulation and coastal circulation... The basin extends to an area situated from 8.5° West to 16.5° East in longitude and from 34.8° to 44.8° North in latitude. The grid size is 1/8° in longitude and 1/10° in latitude which roughly correspond to a 10 km grid size in both directions. The number of vertical levels is 31 whose thickness varies from 6 meters at the surface to 400 meters at 4000 meters. The resulting model mesh has $200 \times 100 \times 31 = 620{,}000$ grid points. The topography has been designed from a realistic bathymetry at 5' of resolution. Only five islands have been conserved at the considered resolution : Ibiza, Majorca, Minorca, Corsica and Sardinia. The Straits of Gibraltar and Sicily are respectively connected to a schematic Atlantic ocean and Eastern Mediterranean sea filled with waters whose temperature and salinity are restored to a climatology [50]. Surface boundary conditions (wind stress, heat and fresh water fluxes) are prescribed from the daily output of the Peridot forecast model of Meteo France [51].

The computer time used for one month circulation is 53 minutes with a 50-processors Paragon system, that is equivalent to a Cray C90 system. The executable code occupies 22 Mbytes per processor. The access file is a very stress point with several reading phases (daily forcing) and several writing phases (3D output files). (Fig. 6) presents the evolution of I/O time versus the number of subdomains during a one month simulation. The different fields are gathered by a few number of writer/reader processors and the parallelism of the Parallel Files System of the Paragon is managed by only six I/O processors, these bottlenecks explain the weak parallelism of I/O phases whereas the number of subdomains is high. Neverthless, the time used for disk acces phases (td) remains negligible compared to the whole simulation time. (Fig. 7) shows that, for a number of subdomains is varying between 30 and 90, the calculation time of SPG by the FETI algorithm (tf) and the communication time (tc) associated with the remaining code are negligible compared to the calculation time of PE' and VP (tr) that can be identified to the whole simulation time (approximation 1). As the pencil splitting preserves the dependencies along the vertical associated with VP, the calculation cost of the space implicit part of these operators can be neglected compared to the calculation cost of the 3D explicit operators of PE' (approximation 2). Then on a N_p-processors machine, the computer time t^{N_p} is supposed to be proportional to the local domain size:

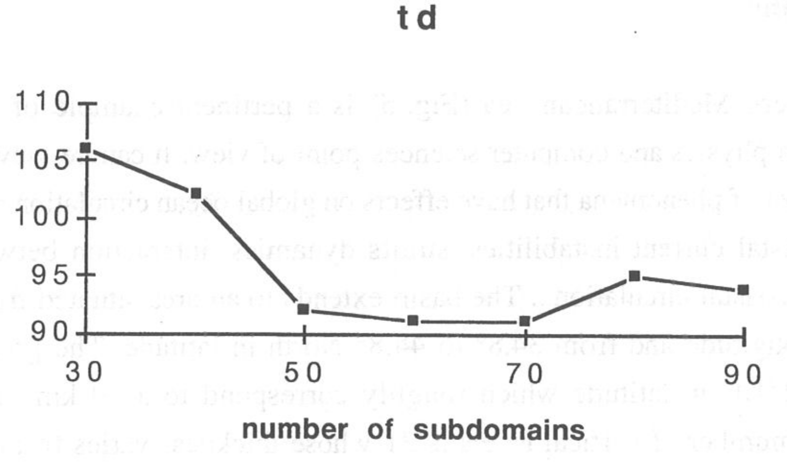


Figure 6 - Evolution of I/O time versus the number of processors for a monthly time integration of the western Mediteranean model of Herbaut et al. on a Paragon system.

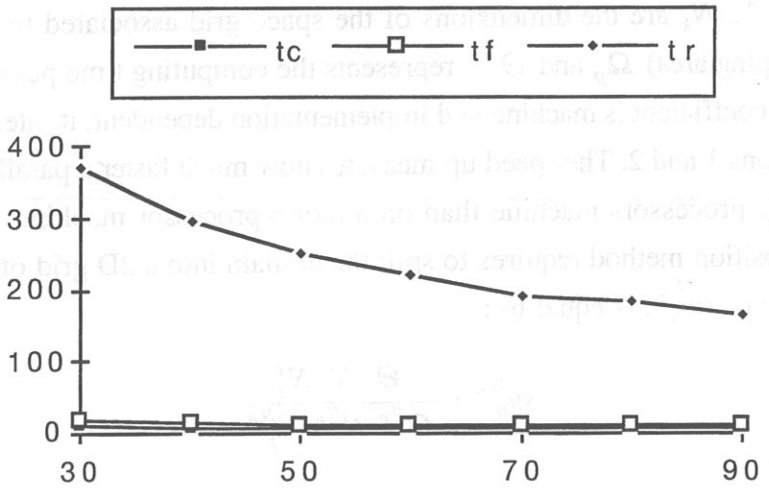


Figure 7 - Different calculation and computation times versus the number of subdomain for one hundred time steps integration of the western Mediteranean model of Herbaut et al. on a Paragon system.

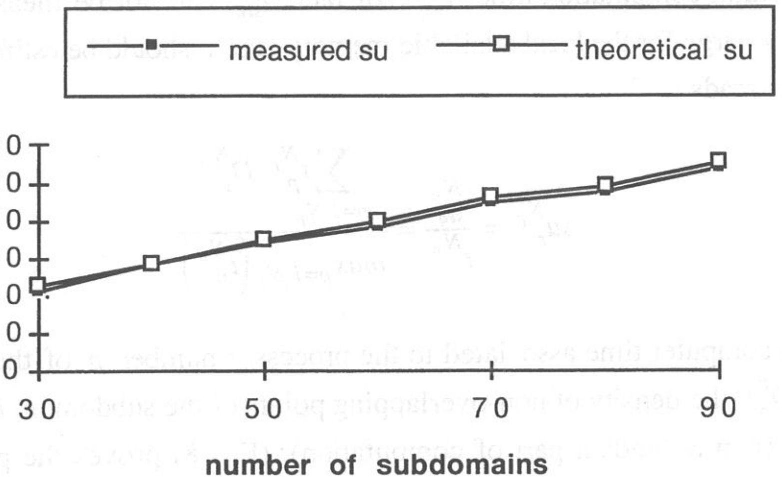


Figure 8 - Measured and theoretical speed up versus the number of subdomains for one hundred time steps integration of the western Mediteranean model of Herbaut et al. on a Paragon system.

$$t^{N_p} = \Theta^{N_p} N_i^{N_p} N_j^{N_p} N_k \tag{33}$$

where $N_i^{N_p}$, $N_j^{N_p}$, N_k are the dimensions of the space grid associated to a N_p-subdomain (with the overlapping area) Ω_p and Θ^{N_p} represents the computing time per grid points of this subdomain. This coefficient is machine and implementation dependent, it integrates imbalances plus approximations 1 and 2. The speed up measures how much faster a parallel computation is performed on N_p-processors machine than on a mono-processor machine. As the proposed domain decomposition method requires to split the domain into a 2D grid of subdomains, the theoretical speed up, $su_{th}^{N_p}$, is equal to :

$$su_{th}^{N_p} = \frac{\Theta^1 N_i^1 N_j^1}{\Theta^{N_p} N_i^{N_p} N_j^{N_p}} \tag{34}$$

Assuming a perfect parallelism (approximation 3) plus approximation 1 and approximation 2, the Θ coefficient remains independent of the number of subdomains and (34) can be approximated by :

$$su_{th}^{N_p} = \frac{N_i^1 N_j^1}{N_i^{N_p} N_j^{N_p}} \tag{35}$$

This ratio has to be compared to the real speed up that compare the elapsed time for the parallel algorithm on a "N_p-processors machine" for a N_p-subdomain experiment, t^{N_p}, with the sequential algorithm computation time, $t_{seq}^{N_p}$. In fact, $t_{seq}^{N_p}$ can not be measured because the experiment is too large for the local available memory, and it should be estimated. In this case, the real speed up reads :

$$su_r^{N_p} = \frac{t_{seq}^{N_p}}{t^{N_p}} = \frac{\sum\limits_{p=1,N_p} t_p^{N_p} D_p^{N_p}}{max_{p=1,N_p}\left(t_p^{N_p}\right)} \tag{36}$$

with $t_p^{N_p}$ is the computer time associated to the processor number p of the "N_p-processors machine" and $D_p^{N_p}$ the density of non-overlapping points of the subdomain number p in case of a $\mathrm{N_p}$-mesh (non redundant part of computation). (Fig. 8) proves the perfect agreement between the real speed up and the theoretical speed up that validates the three approximations and illustrates the quality of the implementation of the parallelization choices and the FETI solver.

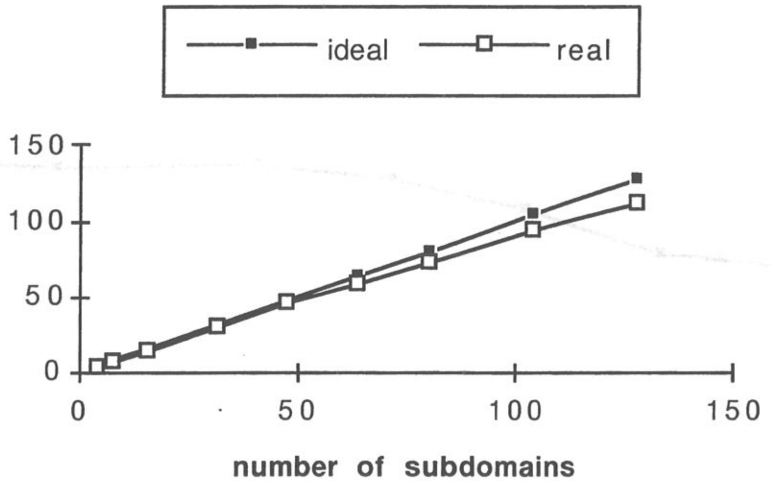


Figure 9 - Speed up of the OPA code excepted the surface pressure gradient operator versus the number of subdomains for one hundred time steps integration on a Cray T3E system.

5.2 An academic large scale experiment

An elementary subdomain is defined which nearly requires the whole local memory of each processor of the T3E machine, its size, $60 \times 60 \times 31$ grid points, occupies 90 Megabytes. Each processor solves the same model on its own subdomain defined from the initial elementary subdomain. Such a test with a fixed local granularity allow to consider the scalability of the OPA algorithms instead of the scalability of a numerical experiment governing by a variable granularity. When the application is fully parallel, the (theoretical) scaled speed up is nearly linear due to the halo effect (that is represented by $D_p^{N_p}$ coefficient in (36)). To measure the real scaled speed up $ssu_r^{N_p}$, it's not possible to run scaled problems on a mono-processor machine because they don't fit in the memory, so this quantity is estimated by the equation (36). (Fig. 9) confirms the scalability of the algorithm associated with VP and PE', then whereas the horizontal domain size is increasing, OPA becomes more and more time consuming with parallel PCG than with FETI (Fig. 10). As explained in section 4 and shown by (Fig. 11), the PCG numerical behaviour is degrading the scalability of OPA code in comparizon with FETI one whereas the number of subdomains increases and SPG computer cost becomes preponderant. As an example, (Fig. 12-a) illustrates how, for the 32-subdomains test case, the number of iterations of the optimized FETI decreases very fast versus time and after 10 time steps, less than 7 iterations per time step are required to capture the solution.

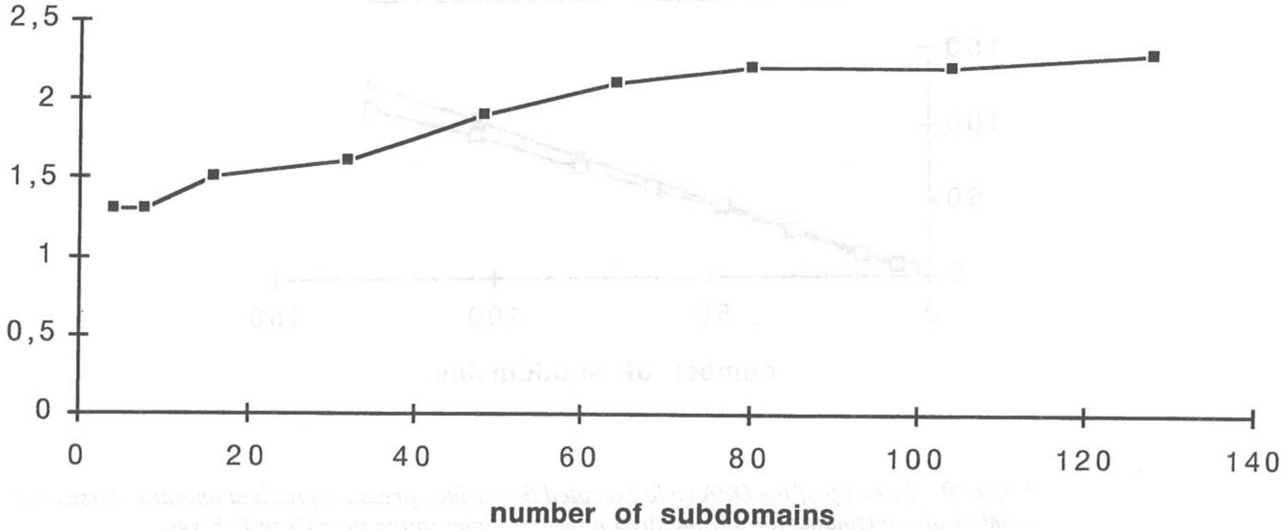


Figure 10 - Ratio of time simulation of OPA with PCG over time simulation of OPA with FETI versus the number of subdomains for one hundred time steps integration on a Cray T3E system.

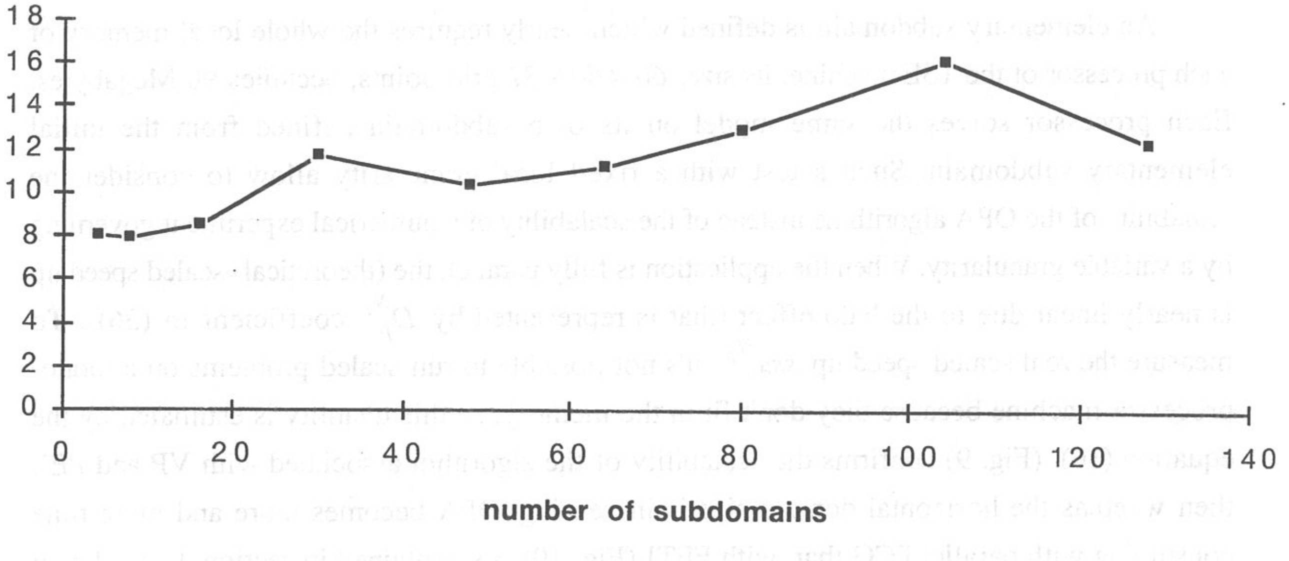


Figure 11 - Ratio of time simulation of SPG with PCG over time simulation of SPG with FETI versus the number of subdomains for one hundred time steps integration on a Cray T3E system.

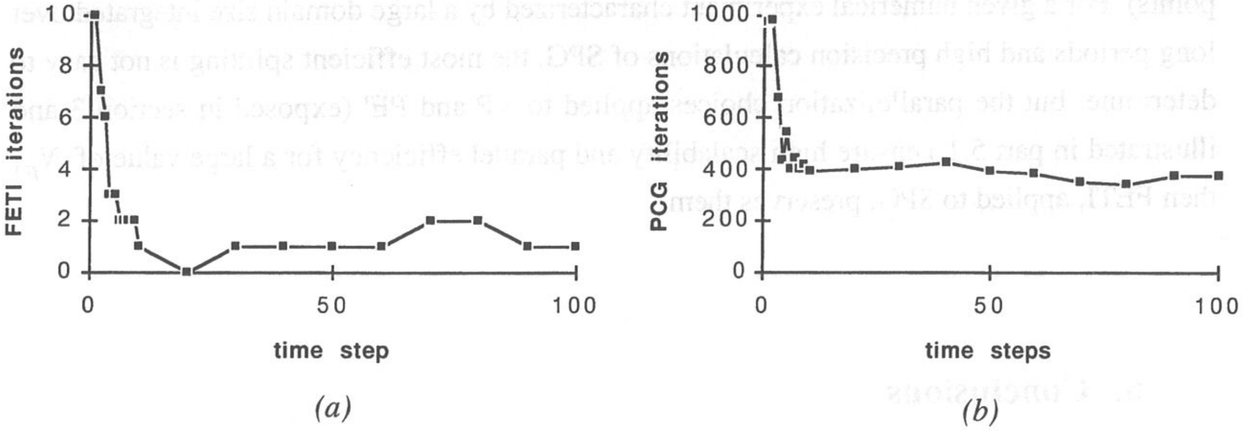


Figure 12 - Number of solver iterations versus time for a 32-subdomains test case on a Cray T3E system: (a) FETI solver, (b) PCG solver

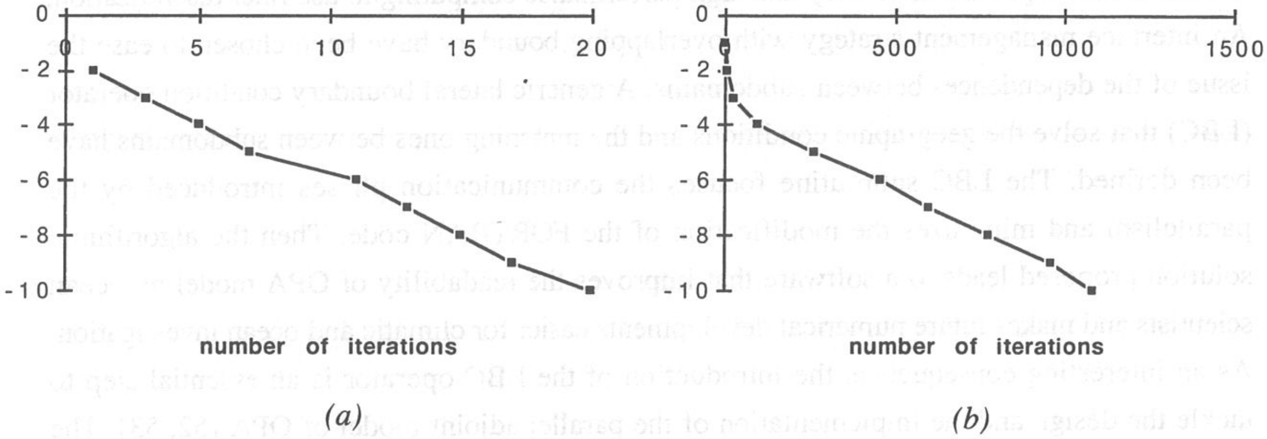


Figure 13 - Number of iterations versus the relative precision, $log_{10}(e)$, for a 32-subdomains test case on a Cray T3E system: (a) FETI solver, (b) PCG solver.

On the contrary, the number of iterations required for PCG convergence stays constant and large (400 iterations) during time integration (Fig. 12-b). Last but not least, high resolution simulation involves large size experiments over long periods and high precision ε on the calculation of the SPG. The curve shows a very regular and efficient convergence behaviour for FETI (Fig. 13-a) and the number of iterations stays small, then PCG number of iterations increases very fast with the precision (Fig. 13-b). Obviously, an iteration of FETI is more time consuming than a PCG one, this cost depends on the ratio defined by the whole experiment domain size over local subdomains size then, to simplify $\left(FETI\left(N_p\right)\right)_{N_p=2,\ldots}$ can be described as a set of methods associated with N_p-subdomains and two limit cases : an exact inverse solver (N_p is equal to one) and the PCG solver (N_p is equal to the number of horizontal grid points). For a given numerical experiment characterized by a large domain size integrated over long periods and high precision calculations of SPG, the most efficient splitting is not easy to determine, but the parallelization choices applied to VP and PE' (exposed in section 3 and illustrated in part 5.1.) ensure high scalability and parallel efficiency for a large value of N_p, then FETI, applied to SPG, preserves them.

6. Conclusions

The ideas underlying domain decomposition methods have been integrated in the OPA model release 8. The domain has been divided into subdomains by a pencil splitting that respects physical proprieties, the vertical boundary condition and the algorithmic scalability needed to obtain parallel efficiency and high performance computing to use finer discretization. An interface management strategy with overlapping boundary have been chosen to ease the issue of the dependences between subdomains. A generic lateral boundary condition operator (LBC) that solve the geographic conditions and the matching ones between subdomains have been defined. The LBC subroutine focuses the communication phases introduced by the parallelism and minimizes the modification of the FORTRAN code. Then the algorithmic solution proposed leads to a software that improves the readability of OPA model by ocean scientists and makes future numerical developments easier for climatic and ocean investigation. As an interesting consequence, the introduction of the LBC operator is an essential step to tackle the design and the implementation of the parallel adjoint model of OPA [52, 53]. The portability of the code on multi-processors and mono-processor platform is preserved: developments have been done on an Intel Paragon computer, then a real life version of the code have run on a large set of different platforms such as Cray T3D, Cray T3E, IBM SP2 and Fujitsu VPP ones with different message passing communication libraries (NX, SHMEM, PVM, MPI). The parallel version of the conjugated gradient algorithm based on this domain decomposition strategy allows to solve the surface pressure gradient contribution in the parallel processing field, this solution presents the advantage to be easy implemented. On the other side, the FETI method exploits largest granularity computations and presents a more robust numerical behaviour. The presented Domain Decomposition approach can be described as a nutshell strategy of the initial code plus the implementation of a fully optimised Dual Schur Complement Method to solve the elliptic equation associated with the surface pressure gradient [54]. Efficiency tests prove the total scalability and illustrate the implementation quality.

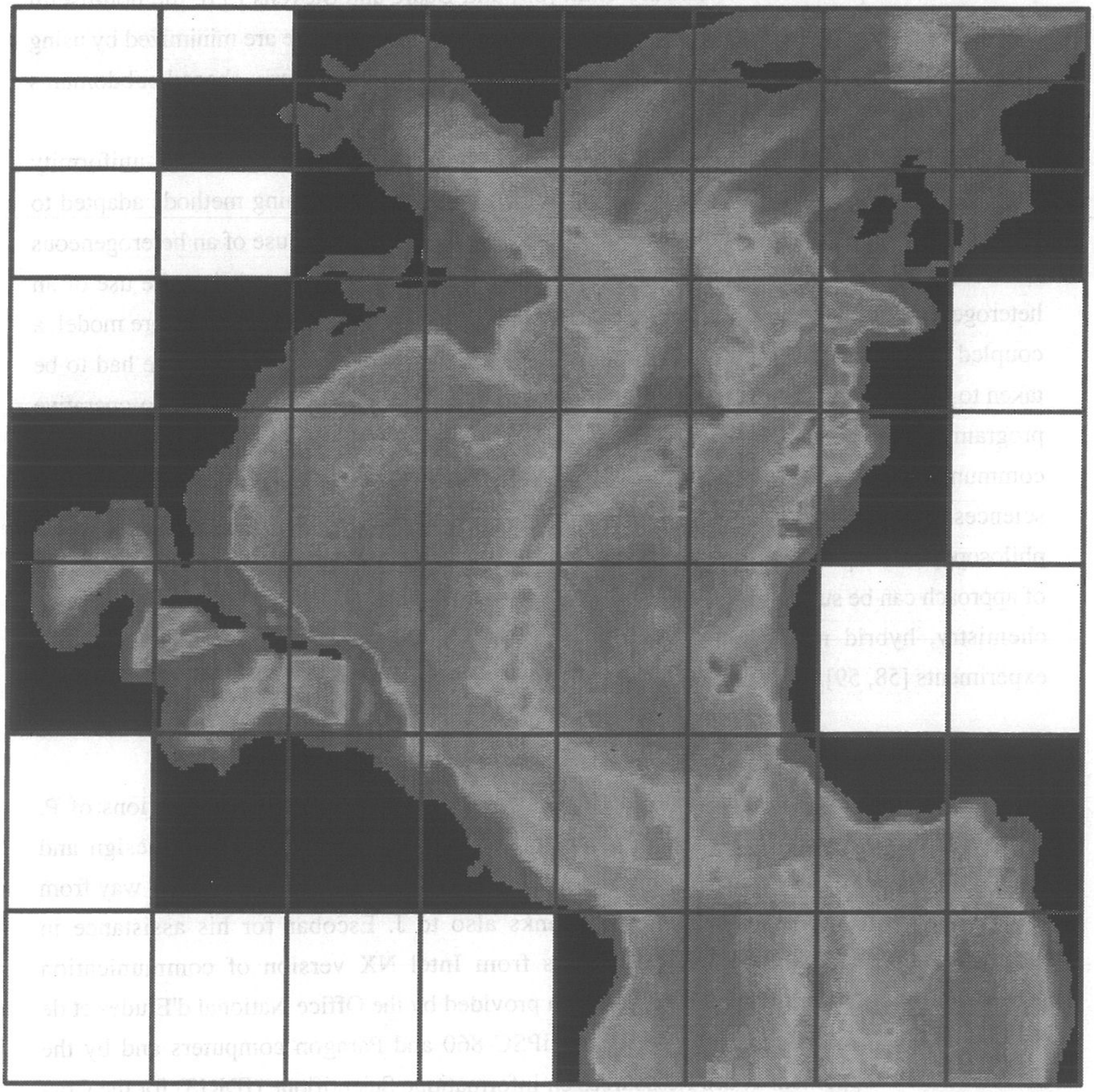


Figure 14 - Optimization of the subdomain splitting for a North-Atlantic experiment. The 15 Squares in white represent the sub-domains which have been illiminated as they are land-only sub-domains.

It has been noticed the initial code was quite well vectorized: a performance of 500 Mflops was obtained on a C90 processor whose peak performance is 1 Gflop. So, such a code have an interesting behaviour on a parallel-vector platform to benefit from parallel efficiency of the Domain Decomposition implementation, when tasks granularity is preserved, and vector potential of the processor. As Sawdey *et al.* [55] and Beare and Stevens [35], the non-useful points of the land domain due to Finite Differences and vector technique are minimized by using a very general scheme that it allows the ocean to be split into irregularly-shaped subdomains (Fig. 14).

Last but not least, domain decomposition approach allows to fit with the non-uniformity of the physical processes by providing a natural framework for zooming methods adapted to space or time scales consistent with local physics phenomena [43], the use of an heterogeneous model based on different parameterizations fit to local physics characteristics, the use of an heterogeneous model to solve an heterogeneous problem (coastal model with offshore model, a coupled ocean -sea ice- atmosphere model). It has been noticed that a special care had to be taken to specify consistent boundary conditions. For such an ambition, the implicit co-operative programming model has to organize future modelling with independent kernels that are able to communicate instead of a monolithic architecture integrating different models [56]. In Earth sciences field, this modularity and task separation concepts have defined the development philosophy of coupling tools such as OASIS software [57]. So, it has also shown that this kind of approach can be suited for very different types of coupling (paleoclimate studies, stratopheric chemistry, hybrid modelling...) on top of more traditional ocean-atmosphere coupled experiments [58, 59].

ACKNOWLEDGEMENTS: We would like to acknowledge the important contributions of P. Delecluse and C. Lévy to the development of this program. The conceptual design and implementation of the Domain Decomposition strategy has benefited in a significant way from discussions with J. Ryan and P. Leca. Thanks also to J. Escobar for his assistance in implementing PVM and SHMEM solutions from Intel NX version of communication procedures. Support for computations has been provided by the Office National d'Etudes et de Recherches Aérospatiales (ONERA) for the iPSC-860 and Paragon computers and by the Institut du Développement et des Ressources en Informatique Scientifique (IDRIS) for the Cray T3E. Funding has been provided by the Direction des Recherches Etudes et Techniques (DRET, n° 92/356) and the French Association Nationale de la Recherche Technique (ANRT, n° 204/92).

REFERENCES:

1. J. Jr. Semtner and R. M. Chervin, J. Geophys. Res. **97**, 5493 (1992).
2. M. J. Flynn, IEEE Trans. Comput. **C-21**, 948 (1972).
3. L. Wolters, in the Proceedings, Seventh International Workshop on the Use of Supercomputers in Theoretical Science, 18-19 June 1992.
4. R. D. Smith, J. K. Dukowicz and R. C. Malone, Physica D **60**, 38 (1992).
5. K. Bryan, J. Comput. Phys. **4**, 347 (1969).
6. F. Vittard and O. Marty, in the Proceedings, Parallel CFD'93 Conference, Paris, France, 1993.
7. G. Madec, P. Delecluse, M. Imbard and C. Levy, "OPA version 8.1, Ocean General Circulation Model, reference manual", Notes du Pôle de modélisation,N°11, 91pp., Laboratoire d'Océanographie Dynamique et de Climatologie, Paris, France (1998).
8. P.F. Fischer and E. M. Ronquist, Comput. Methods Appl. Mech. Engrg., 1994.
9. S. R. M.Barros, D. Dent, L. Isaksen and G. Robinson, in the Proceedings, Sixth ECMWF Worksh. on the use of Parallel Processors in Meteorology, November 21-25 Reading, UK, 1994.
10. F. -X. Roux, La Recherche Aérospatiale **1**, 37 (1990).
11. R.Bleck, M. O'Keefe and A. Sawdey, "A comparison of data-parallel and message-passing versions of the Miami Isopycnic Coordinate Ocean Model (MICOM)", in Parallel Computing. Special Issues on Applications. Climate and weather modeling. Volume 21, Number 10, 1995.
12. J. Marshall, A. Adcroft, C. Hill, L. Perelman and C.Heisey, J. Geophys. Res. **102**, 5753 (1997).
13. G. Madec and M. Imbard, Climate Dyn. **12**, 381 (1996).
14. D. R. Jackett and T. J. Mc Dougall, J. Atm. Ocean. Tech. **12**, 381 (1995).
15. G Madec, M. Chartier, P. Delecluse and M. Crépon, J. of Phys. Oceanogr. **21**, No 9 (1991)
16. B. Blanke and P. Delecluse, J. Phys. Oceanogr. **23** (1993)
17. M. Lévy, L. mémery and G. Madec, J. Mar. Syst., in press (1997).
18. P. D. Killworth, D. Stainforth, D. J. Webb and M. Paterson, J. of Phys. Oceanogr. **21**, 1333 (1991).
19. J. K. Dukowicz and D. Smith, J. of Geophys. Research. **99**, 79911(1994).
20. G. Roulet and G. Madec, "A variable volume formulation conserving salt content for a level OGCM : a fully nonlinear free surface", in preparation (1998).
21. F. Mesinger and A. Arakawa, GARP Publication series, No 17, World Meteorological Organisation, Geneva, 64 p (1976)
22. O. Marty, G. Madec and P. Delecluse, J. of Geophys. Research. **97**, 12, 763-12, 766 (1992)
23. A. J. Robert, J. Meteor. Soc. Japan **44**, 237 (1966).
24. R. Asselin, Mon. Wea. Rev. **100**, 487 (1972).
25. M. Guyon, F. -X. Roux, M. Chartier and P. Fraunie, in the Proceedings,Workshop "High Performance Computing in the Geosciences", June 21-25 Les Houches, France, 1993.
26. M. Guyon, F. -X. Roux, M. Chartier and P. Fraunie, Ocean Modelling **103** (1994)
27. M. Guyon, F. -X. Roux, M. Chartier and P. Fraunie, Ocean Modelling **105** (1994)
28. A. R. Clare and D. P. Stevens, Future Generation Computer Systems **9**, 11 (1993).
29. P. Andrich, G. Madec and D. L'Hostis, in the Proceedings, International Conference on Supercomputing, July 4-8, St-Malo, France, 1988.
30. D. J. Webb, Computers & Geosciences **22**, 569 (1996).
31. B. L. Buzbee, F. W. Dorr, J. A. George and G. H. Golub, SIAM, J. Numer. Anal. **8**, 722 (1971)
32. R. B. Wilhelson and J. H. Ericken, J. Comp. Phys., **25**, 319 (1977)
33. G. Madec, C. Rahier and M. Chartier, Ocean Modelling **78**, 1-6 (1988)
34. C. Farhat and F. -X. Roux, "Implicit parallel processing in structural mechanics", Computational mechanics advances, Vol 2, N° 1, June 1994.
35. M. I. Beare and D. P. Stevens, Annales Geophysicae **15**, 1369 (1997).
36. G. H. Golub and G. A. Meurant, Résolution de grands systèmes linéaires, Eyrolles, Paris (1983)
37. A. Van der Sluis and H. A. Van der Vorst, Numer. Math. **48**, 543 (1986).
38. O Axelsson and G. Lindskog, Numer. Math. **48**,. 499 (1986).
39. P. L. Lions, in the Proceedings, First Int. Symp. on Domain Decomposition Methods for Partial Differential Equations, January 7-9, Paris, France, 1987 (SIAM, Philadelphia, 1988).
40. P. E. Bjordstad and O. B. Wildlund, SIAM Journal of Numerical Analysis **23** (1986).
41. I.M. Navon and Y. Cai, Computer Methods in Applied Mechanics and Engineering (1993).
42. C. Bernardi, Y. Maday and A. T. Patera, A new nonconforming approach to domain decomposition: the mortar element method, Internal Rep. R 89027, Laboratoire d'Analyse Numérique, Universite Paris VI, France (1991)
43. M. Laugier, P. Angot and L. Mortier, Int. J. for Numer. Meths in Fluids, Vol 23, 1163-1195 (1996)

44. C. Farhat and M. Gérardin, Int. J. Numer. Meths. Engrs., Vol. 41, 675-696 (1998)

45. P. E. Gill and W. Murray, Numerical methods for constrained optimization, Academic Press, London (1974)

46. F. -X. Roux, Compte Rendu de l'Académie des Sciences **308**, série I, 193, Paris, France (1989).

47. F. -X. Roux and D. Tromeur-Dervout, in the Proceedings, 7th International Conference on Domain Decomposition Methods, 1995.

48. J. J. Dongarra, J. Du Croz, S. Hammarling and R. J. Hanson, Extended Set of FORTRANBasic Linear Algebra Subprograms, ACM Trans. Math. Soft. **14**, pp. 1-17 (1988)

49. C. Herbaut, F. Martel and M. Crépon, , J. Phys. Oceanogr.**27**, 2126 (1997).

50. S. Levitus, Climatological Atlas of the World Ocean. NOAA Prof. Paper, 13, U.S. Govt. Printing Office, 173 pp (1982)

51. M. Imbard and A. Joly, in the Proceedings, WMO/IUGG NWP Symposium, Tokyo (1986).

52. E. Greiner, S. Arnault and A. Morlière, Prog. Oceanogr., 41, 141-202 (1998)

53. E. Greiner, S. Arnault and A. Morlière, Prog. Oceanogr., 41, 203-247 (1998)

54. M. Guyon, "Contribution à la modéllisation océanique sur architectures parallèles par une approche de décomposition de domaine", Ph.D. Thesis, Université Aix-Marseille II, 202 pp (1995).

55. A. Sawdey, M. O'Keefe, R. Bleck and R. W. Numrich, in the Proceedings, Sixth ECMWF Worksh. on the use of Parallel Processing in Meteorology, Eds. G. Hoffman and N. Kreitz, World Scientific, London, pp 523-548 (1995)

56. P. Leca and F. -X. Roux, La Recherche Aérospatiale **4**, 257 (1996).

57. L. Terray, The OASIS coupler user guide, version 2.1. Tech. Rep. TR/CMGC/96-46, CERFACS (1996).

58. E. Guilyardi and G. Madec, Climate Dynamics, **13**, 149-165 (1997)

59. Barthelet, P., S. Bony, P. Braconnot, A. Braum, D. Cariolle, E. Cohen-Solal, J.-L. Dufresne, P. Delecluse, M. Déqué, L. Fairhead, M.-A. Filiberti, M. Forichon, J.-Y. Grandpeix, E. Guilyardi, M.-N. Houssais, M. Imbard, H. Le Treut, C. Lévy, Z. X. Li, G. Madec, P. Marquet, O. Marti, S. Planton, L. Terray, O. Thual, S, Valcke, *C. R. Acad. Sci Paris*, 326, 677-684 (1998)

Déjà paru :

1 : **Janvier 1998** Agnès Ducharne, Katia Laval and Jan Polcher, *Sensitivity of the hydrological cycle to the parameterization of soil hydrology in a GCM*

2 : **Janvier 1998** Marina Lévy, Laurent Mémery and Jean-Michel André , *Simulation of primary production and export fluxes in the Northwestern Mediterranean Sea*

3 : **Février 1998** Valérie Masson, Sylvie Joussaume, Sophie Pinot and Gilles Ramstein, *Impact of parameterizations on simulated winter mid-Holocene and Last Glacial Maximum climatic changes in the Northern Hemisphere*

4 : **Mars 1998** Jérôme Vialard et Pascale Delecluse, *An OGCM Study for the TOGA Decade. Part I: Role of Salinity in the Physics of the Western Pacific Fresh Pool, Part II: Barrier layer formation and variability*

5 : **Avril 1998** O. Aumont, J. C. Orr, P. Monfray, and G. Madec, *Nutrient trapping in the equatorial Pacific: The ocean circulation solution*

6 : **Mai 1998** Emmanuelle Cohen-Solal and Hervé Le Treut, *Long term climate drift of a coupled surface ocean-atmosphere model : role of ocean heat transport and cloud radiative feedbacks*

7 : **Juin 1998** Marina Lévy, Laurent Mémery and Gurvan Madec, *Combined Effects of Mesoscale Processes and Atmospheric High-Frequency Variability on the Spring Bloom in the MEDOC Area*

8 : **Septembre 1998** Carine Laurent, Hervé Le Treut, Zhao-Xin Li, Laurent Fairhead and Jean-Louis Dufresne, *The influence of resolution in simulating inter-annual and inter-decadal variability in a coupled ocean-atmosphere GCM, with emphasis over the North Atlantic.*

9 : **Octobre 1998** Francis Codron, Augustin Vintzileos and Robert Sadourny, *An Improved Interpolation Scheme between an Atmospheric Model and Underlying Surface Grids near Orography and Ocean Boundaries.*

10 : **Novembre 1998** Z.X. Li and A.F. Carril, *Transient properties of atmospheric circulation in two reanalysis datasets.*

11 : **Décembre 1998** Gurvan Madec, Pascale Delecluse, Maurice Imbard and Claire Lévy, *OPA8.1 ocean general circulation model reference manual.*

Plus de détails sont disponibles sur Internet :
http://www.ipsl.jussieu.fr/modelisation/liste-notes.html.