MASSIVELY PARALLEL IMPLEMENTATION OF A HIGH ORDER DOMAIN DECOMPOSITION EQUATORIAL OCEAN MODEL

Hong Ma
Brookhaven National Laboratory, Upton, NY 11973
hm@bnl.gov 516 344-4138

Joseph W. McCaffrey
Naval Research Laboratory, Stennis Space Center, MS 39529
mccaffrey@nrlssc.navy.mil 601 688-5053

Steve Piacsek
piacsek@nrlssc.navy.mil 601 688-5316

Abstract

The present work is about the algorithms and parallel constructs of a spectral element equatorial ocean model. It shows that high order domain decomposition ocean models can be efficiently implemented on massively parallel architectures, such as the Connection Machine Model CM5. The optimized computational efficiency of the parallel spectral element ocean model comes not only from the exponential convergence of the numerical solution, but also from the work-intensive, medium-grained, geometry-based data parallelism. The data parallelism is created to efficiently implement the spectral element ocean model on the distributed-memory massively parallel computer, which minimizes communication among processing nodes. Computational complexity analysis is given for the parallel algorithm of the spectral element ocean model, and the model's parallel performance on the CM5 is evaluated. Lastly, results from a simulation of wind-driven circulation in low-latitude Atlantic ocean are described.

1 Introduction

The spectral element method is a combination of both the spectral and the finite element methods. The spectral element method is also called the “p-type finite element” method, or the h-p type weighted residual method. Like the spectral method, it uses high order polynomials as trial functions, but like the finite element method, it decomposes the computational domain into many elements and defines local trial functions. The hybrid character of the spectral element method enables it to overcome the shortcomings of both the spectral method and the finite element method but still retain their advantages. Since the trial functions of the spectral element method are local, it can handle complex geometry easily. On the other hand, it is still a high order weighted residual method, so the exponential convergence rate is achieved as the degree of the polynomials in each element is increased. The main difference between the spectral element method and the spectral multi-domain method is that the $C^0$ and $C^1$ boundary conditions at the interface of the elements have to be explicitly enforced by the spectral multi-domain method. The spectral element method, by contrast, uses the variational principle to guarantee $C^0$ and $C^1$ (weakly) continuity at the
interface, which results in a much simpler and more natural approach than the nonvariational method; therefore, parallel algorithms can be conveniently implemented. In the past decade or so, research on the spectral element method has made important progress in perfecting this state-of-the-art numerical method [4,13,14]. More recently, the spectral element method has shown encouraging potential in oceanic applications [5,6,7,8,9,10,11].

The present work is about the implementation and results of a massively parallel, spectral element, high-resolution, three-dimensional equatorial ocean model which, in particular, is capable of resolving both the horizontal and the vertical structures of the low-latitude western boundary processes. The current version of the model is driven solely by wind stress and ignores the dynamical effects of stratification. This model is designed to study the effect of wind in the formation and variation of important meso-to-small scale equatorial ocean phenomenon, such as eddies, low-latitude western boundary currents, and vertically alternating equatorial zonal jets. The high efficiency of this model is based on an optimized coupling between the numerical algorithm and the computer architecture (algorithm-architecture). In addition to its exponential convergence rate, the model’s performance is further enhanced by the spectral element tensor-product factorization and spectral element parallelism.

2 Governing Equations

The model equations include three dimensional time dependent primitive equations with hydrostatic approximation [1]. The vertical Coriolis force term in the zonal momentum equation, which is usually omitted in general circulation models, is kept for the reason that it could become important at the equator.

\[
\frac{du}{dt} - (2\Omega + \frac{u}{r \cdot cos\phi})(v \cdot sin\phi - w \cdot cos\phi) = -\frac{1}{\rho \cdot r \cdot cos\phi \frac{\partial p}{\partial \lambda}} + A_H \Delta u + A_v \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \frac{\partial}{\partial r}) u
\]

\[
\frac{dv}{dt} + \frac{wv}{r} + (2\Omega + \frac{u}{r \cdot cos\phi}) u \cdot sin\phi = -\frac{1}{\rho \cdot r \frac{\partial p}{\partial \phi}} + A_H \Delta v + A_v \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \frac{\partial}{\partial r}) v
\]

\[
\frac{1}{r \cdot cos\phi} \frac{\partial u}{\partial \lambda} + \frac{1}{r \cdot cos\phi} \frac{\partial (v \cdot cos\phi)}{\partial \phi} + \frac{\partial w}{\partial r} = 0
\]

where

\[
\frac{d}{dt} = \frac{\partial}{\partial t} + \frac{u}{r \cdot cos\phi} \frac{\partial}{\partial \lambda} + \frac{v}{r} \frac{\partial}{\partial \phi} + \frac{w}{r} \frac{\partial}{\partial r}
\]
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
\[ \Delta = \frac{1}{r^2 \cos^2 \phi} \frac{\partial^2}{\partial \lambda^2} + \frac{1}{r^2 \cos \phi} \frac{\partial}{\partial \phi} (\cos \phi \frac{\partial}{\partial \phi}) \]  

and \( \lambda \) is longitude, \( \phi \) latitude, and \( r \) radial distance; \( u, v, \) and \( w \) are the velocity components in the \( \lambda, \phi, \) and \( r \) directions, respectively; \( p \) is the pressure term. \( A_H \) and \( A_v \) are the eddy viscosity coefficients in the horizontal and vertical directions, respectively.

For convenience, the original spherical coordinates \((\lambda, \phi, r)\) are mapped onto another coordinates system \((x, y, z)\), and relationship between these two coordinate systems is defined as:

\[
\begin{align*}
x &= r_0 \cos \phi \\
y &= r_0 \phi \\
z &= r - r_0
\end{align*}
\]

where \( r_0 \) is the radial distance of the sea level.

No-normal flow, no-slide boundary conditions are applied to all lateral boundaries and to the ocean floor. At the sea-surface, we assume the rigid-lid boundary condition. The present model is solely driven by wind stress:

\[
\begin{align*}
\rho_0 A_v \frac{\partial u}{\partial z} |_{z=0} &= \tau_x \\
\rho_0 A_v \frac{\partial v}{\partial z} |_{z=0} &= \tau_y
\end{align*}
\]

where \( \tau_x \) and \( \tau_y \) are surface wind stress components.

3 Spectral Element Discretization and Solvers

The basis sets used in the present work are as follows:

\[ \psi^{e}_{lnn}(\xi, \eta, \zeta) = h_l(\xi) h_m(\eta) h_n(\zeta) \quad l, m, n \in \{0, 1, \ldots N\}^3 \]

where \( h_i(s) \) are the Gauss-Lobatto-Legendre polynomials.

If we use a single subscript, \( q (q \in \{1, 2, \ldots, (N + 1)^3\}) \), the mapping between a macroelement, \( \Omega_e \), and its phase domain, \( \tilde{\Omega}_e \), can be expressed as:

\[ x = x_q \psi^e_q(\xi) \]

Where \( x \in \Omega^e \) and \( \xi \in \tilde{\Omega}_e \).

Let solution \( u \) at time \( n\Delta t \) on each subdomain \( \Omega^e \) be expanded as:

\[ u^e(x, n\Delta t) = u^e_q(n\Delta t) \psi^e_q[\xi(x)] \]

3
where \( f_q(t) \) is the value of function \( f \) at the collocation point \( x_q \in \Omega^e \) at time \( t \).

By using the same variational procedures as those in [6,7], i.e., all the integrations are evaluated by the Gauss-Lobatto quadrature scheme, which is an exact formula for \((2N-1)^{th}\) order polynomials, the spatially discretized formulae for the primitive equations can be obtained. In particular, the isoparametric spectral element discretization formulae for the horizontal momentum equations of the present primitive equation model are virtually identical as those in [6, 7]. One advantage of using the Gauss-Lobatto-Legendre polynomials as basis functions is that we only have to deal with one set of grid points for both interpolating the solutions and evaluating the integrals.

We choose the isoparametric spectral element discretization scheme [6, 7], namely, using nonstaggered grids, for the present numerical model. The nonstaggered formulation avoids spurious pressure modes as staggered schemes do, and, at the same time, has the advantage that pressure is continuous across boundaries of the spectral elements. Only one set grids is required for both interpolation and quadrature, hence simplifying operations.

The discretized incompressibility condition and the hydrostatic condition have the following format:

\[
[D^e][w] = [g]
\]

where \([D^e]\) is the matrix generated by applying variational procedures to the vertical differentiation operator; \([w]\) is the vector representing the unknown at the collocation points, and \([g]\) is a vector whose components are known.

To obtain the solution for the vertical velocity, \( w \), we need to solve a matrix problem of the above format. It can be done by using either matrix iteration methods or direct matrix inversion. The latter is especially efficient when the vertical grains of the spectral element mesh are parallel to the z axis, since the dimension of the matrix to be inverted is the same as the number of levels in the vertical direction.

The time marching scheme for the hyperbolic equations of the present model is the 3\(^{rd}\) order Adams-Bashforth scheme. This scheme has proven to be efficient in high Reynolds number, high resolution simulations, especially in a massively parallel computing environment [7]. In fact, except in the upper range of eddy viscosity (diffusion) for oceanic applications, it is likely to be more efficient to use a fully-explicit scheme because it results in diagonal stiffness matrices for the hyperbolic equations.

A preconditioned conjugate gradient iterative solver is used in the present model to solve the elliptic equation \([A][x] = [b]\) associated with the pressure term, which has the following algorithm

\[
[x_0] = initial\; guess; \quad [r_0] = [b] - [A][x_0]; \quad [q_0] = [P^{-1}][r_0]; \quad [s_0] = [q_0];
\]

\[
\alpha_m = [r_m] \cdot [s_m] /[q_m] \cdot \{[A][q_m]\}; \quad [x_{m+1}] = [x_m] + \alpha_m[q_m];
\]

\[
[r_{m+1}] = [r_m] - \alpha_m[A][q_m]; \quad [s_{m+1}] = [P^{-1}][r_{m+1}];
\]


\[ \beta_m = [r_{m+1}] \cdot [s_{m+1}] / [r_m] \cdot [s_m]; \quad [q_{m+1}] = [s_{m+1}] + \beta_m [q_m] \quad (15) \]

where \( P \) is the Jacobi preconditioner.

Evaluation of the matrix-vector product, \([A][u]\), constitutes the operation count kernel of the spectral element iteration solver. Inherited from the properties of the spectral method, the number of operations for the matrix-vector product \([A][u]\) in the present spectral element model would be proportional to \( KN^{2d} \) (\( d \) is the number of spatial dimensions) if a simple-minded algorithm were applied. Although the problem is less severe for the spectral element method than it is for the spectral method, it still hinders the efficiency of the 3-D model. Fortunately, the partial summation algorithm, which was first proposed by Orszag (1980) for the spectral method, can drastically reduce the cost of solving the matrix-vector product problem.

Assume \([A]\) is the global stiffness matrix for the laplacian operator. The straightforward spectral element formulation gives the tensor product form

\[
[A][u](q, r, s) = \sum_{k=1}^{K} \sum_{l=0}^{N} \sum_{m=0}^{N} A^k(q, r, s, l, m, n) u^k(l, m, n)
\]

\[ (q, r, s \in \{0, \ldots, N\})^3 \quad (16) \]

where \([A][u](q, r, s)\) are elements of \([A][u]\); \(A^k(q, r, s, l, m, n)\) are elements of the local stiffness matrix \([A^k]\) and \(\sum'\) is the direct stiffness matrix summation. The evaluation of \([A][u]\), therefore, requires \(O(KN^6)\) operations.

The present work uses the partial summation algorithm at the elemental level to evaluate \([A][u]\) more efficiently. \(A^k(q, r, s, l, m, n)\) can be evaluated in the form of

\[
A^k(q, r, s, l, m, n) = H^z(l, q) H^y(m, r) H^z(n, s) \quad (17)
\]

where \(H^p(i, j)\) are functions of \(i\) and \(j\).

Therefore, the auxiliary matrices \([B]\) and \([C]\) can be constructed in the following way:

\[
B(l, m, s) = \sum_{n=0}^{N} u^k(l, m, n) H^z(n, s) \quad (18)
\]

\[
C(l, r, s) = \sum_{m=0}^{N} B(l, m, s) H^y(m, r) \quad (19)
\]

Then \([A^k][u^k]\) can be evaluated by

\[
[A^k][u^k](q, r, s) = \sum_{l=0}^{N} C(l, r, s) H^z(l, q) \quad q, r, s \in \{0, 1, \ldots, N\}^3 \quad (20)
\]
Now, the total operation count in computing $B$, $C$, and $[A][u]$ is $O(KN^4)$; this is $N^2$ times less expensive than using the straightforward algorithm to evaluate the matrix-vector product.

The greatly reduced operation count resulting from the partial summation algorithm is not the only advantage of the spectral element model in evaluating the matrix-vector product; it also has a much smaller storage requirement. In fact, the storage required to evaluate $[A][u]$ for a 3-D model would be $O(KN^3)$ instead of $O(K^2N^6)$ because the stiffness matrix elements are "computed on the fly" rather than stored in the memory.

4 Parallel Implementation

The spectral element primitive equation ocean model is parallelized to run efficiently on the Connection Machine Model CM5. In order to avoid unnecessary communication among processing nodes, which is of first order importance in a parallel implementation on a distributed memory, massively parallel architecture, a data mapping scheme was created so that all the information related to a given spectral element is collected in the memory of a single processor. Prior to assembling the global stiffness-matrices, only data related to a given spectral element are used to create the local matrices of that spectral element. At this stage, all computations are carried out at the local level, therefore, there is no communication among neighboring processors while assembling local (elemental) matrices. On the Connection Machine model CM-5, we pursue data parallelism by designing the layout of the arrays of the spectral element model in such a way that the axes along the number of elements are assigned as parallel dimensions, and those along intra-element degrees of freedom as serial ones, is no communication among neighboring processors during elemental level computations, and they are performed concurrently across all virtual processors.

The computational kernel of solving the discretized primitive equations model is calculating matrix-vector products. We split the procedure of calculating matrix-vector products into two steps, each of which admits concurrency. At the first step, the matrix-vector products are carried out at the elemental level with, for example, the elemental Laplacian and mass matrices:

$$r^k(i) = \sum_{q=1}^{(N+1)^d} A^k(i, q) u^k(q) \quad i \in \{1, 2, \ldots, (N+1)^d\}, \quad k \in \{1, 2, \ldots, K\} \quad (21)$$

$$s^k(i) = B^k(i) u^k(i) \quad i \in \{1, 2, \ldots, (N+1)^d\}, \quad k \in \{1, 2, \ldots, K\} \quad (22)$$

After applying tensor-product factorization, the computational complexities to evaluate (21) and (22) would be $C_1 KN^{d+1}/Q$ and $C_2 KN^d/Q$, respectively, where $Q$ is the number of physical processors involved. On the Connection Machine systems, parallel data structure allows (22) to be performed in an array operation, which means that
thousands of simultaneous multiplications are made across all the array elements. Hence, $C_2$ is a small number. Consequently, diagonal preconditioning is especially efficient in the data parallel environment: it does not require direct stiffness summation, and only local computation is involved. Iteration counts can be reduced by twenty to thirty percent with about a one percent increase in cost. The processing nodes on the CM-5 model are equipped with powerful vector-processing units that can further reduce the cost of elemental level computation. These vector-processing units are most efficient when the order of the spectral elements is high.

The second step is to carry out direct stiffness summation, $\sum_{k=1}^{K} \cdot$, in which contributions from local nodes that share the same physical coordinates are first accumulated, and then assigned back to those local nodes. In a serial spectral element model, this procedure can be accomplished by using global and local index systems, and is automatically done as the matrix computation is made for each spectral element. In the parallel spectral element model, however, it is more efficient to use a separate step for the direct stiffness summation. Since each spectral element has at least one edge (two-dimensional case) or one surface (three-dimensional case) that is shared by a neighboring element, the direct stiffness summation can be carried out simultaneously along these edges or surfaces enabling structured message exchange, i.e., edge-based message exchange for two-dimensional problems, and surface-based message exchange for three-dimensional ones. Since this kind of information exchange takes place along the linkages of the “macro-element-skeleton”, it can be easily synchronized for all elements in the entire domain. The work per processor that is required in this procedure is $C_3 d K N^{d-1}/Q$. The structured message exchange mostly avoids explicit short messages, and it considerably improves the parallel efficiency of the spectral element model.

With parallel prefix of the CM Fortran, MATMUL and SUM, an inner product can be executed completely in parallel. Its computational complexity is $C_4 K N^d/Q$. Due to the high level of concurrency afforded by the parallel prefixes, $C_4$ is a small number.

In high Reynolds number case, the discretized horizontal momentum equations only requires a direct method to solve. The computation kernel here is the evaluation of the advection term where concurrency can be achieved at different levels of the computation. We first evaluate the shears of velocities at all nodal points

$$\frac{\partial u^n_e(l)}{\partial x_j} = \sum_{s=1}^{(N+1)^d} u^n_m (s) \frac{\partial u^n_e}{\partial x_j}$$

$$l \in \{1,2,...,(N+1)^d\}, \quad m, j \in \{1,2,...,d\}^2, \quad e \in \{1,2,...,K\} \quad (23)$$

This operation is executed concurrently across all virtual processors. With the partial summation method, the computational complexity for (23) is $C_5 K (N+1)^{d+1}/Q$. The discretized advection term also can be written as
Therefore, once the shears of velocities are obtained, the remaining operation to evaluate the advection terms is the same as that of (22). Hence, the total computational complexity of (24) is \( C_2 K (N + 1)^d / Q + C_5 K (N + 1)^{d+1} / Q \).

As spectral elements are of high-order, most of the costly operations are at the elemental level, and they are executed concurrently. The spectral granularity at the elemental level can take full advantage of the computing power that the latest processing units provide. The structured message exchange, combined with parallel prefix, makes inter-element communication a lower-order rather than a higher-order cost, compared to that of elemental level computation. This communication cost should be much smaller than that of the h-type finite element model, partially because many fewer redundant nodal values, shared by more than one element, have to be stored.
number of spectral elements becomes so large that the memory in the CM5 partition is saturated. The excellent scalability of the model recovers when the size of the CM5 partition is increased.

5 Results From An Equatorial Atlantic Experiment

In this numerical experiment, the model is driven solely by surface wind stress, and the model ocean is an idealized rectangular basin. It is designed to study the effect of wind in the formation and variation of important equatorial ocean phenomena which are meso-scale at least in one spatial dimension, such as low-latitude western boundary currents, eddies, and vertically/meridionally alternating low-latitude zonal jets. The assigned values for the horizontal and vertical eddy viscosity coefficients for the current simulation are $2.4 \times 10^6 \text{ cm}^2 \text{s}^{-1}$ and $30 \text{ cm}^2 \text{s}^{-1}$, respectively. Since the present spectral element model has adequate resolution to resolve meso-scale eddies, we were able to use a horizontal eddy viscosity coefficient which is an order of magnitude smaller than what was typically used in OGCMs to allow the meso-scale processes to be modeled more realistically.

The lengths of the edges of the macro spectral elements which were used to discretize the present model measured 3.5° in the meridional direction, and 90 m vertically. A strip of refined spectral elements is embedded in the region west of 34° W where the zonal length of the macro spectral elements measured 1.5°; elsewhere it measured 6°. This discretization strategy is based on the fact that short equatorial waves are confined near to the western boundary region. The seventh order Gauss-Lobatto-Legendre polynomials were used to construct the basis functions within each macro spectral elements. Based on the rule-of-thumb that 3.5 interpolation points per wavelength would be required for the spectral element model to resolve a wave-like solution with $O(1\%)$ numerical error, the present spectral element ocean model configuration can adequately resolve waves of wave-lengths 150 km in the meridional direction, 50 m in the vertical direction, and 70 km in the zonal direction (300 km if east of 34° W).

Figures 2a and 3a show that beneath the surface layer, there are bands of westward currents centered around 10°N and 9°S, which supply the equatorward western boundary undercurrents. Compared with the observed structure of the intermediate layer currents [2], these model subsurface westward currents correspond to the North and the South Equatorial Currents (NEC and SEC), respectively. In the winter season, the model North Equatorial Undercurrent (NEUC) at 3°N is fed from the north by the southward western boundary undercurrent, supplied by the NEC. In summer season, the southward western boundary undercurrent does not reach as far south as in winter, and it veers into the North Equatorial Countercurrent (NECC) centered at 7°N. In the upper part of the undercurrent layer (~ 100 m deep), in all seasons, the SEC feeds the model North Brazil Current (NBC) which in turn feeds the South Equatorial Undercurrent (SEUC) near 3°S and the EUC. The model NBC also feeds the NEUC in summer, in contrast to the northerly supply of the NEUC in winter. An interesting phenomenon is that while the model NEUC is a permanent current
feature, its supply in the western boundary region alternates between a southerly and northerly one depending on the season. From Figures 2a and 3a, we note that outside the western boundary region, the EUC provides input to the NEUC and the SEUC through poleward meridional flows on both of its sides. The present model reproduces the retroflection of the NBC near 4°N in summer. In the upper undercurrent layer, the NBC curves back to the NEUC (Figure 3a).

Figure 2. Model currents in January. (a) 100 m. (b) 200 m.

Figure 3. Model currents in July. (a) 100 m. (b) 200 m.
In the lower undercurrent layer (~200 m), the eastward model NEUC and SEUC diminish, and the model Equatorial Undercurrent (EUC) is now flanked by two westward off-equatorial undercurrents located at 1.5°N and at 1.5°S (Figures 2b and 3b). In winter, the westward off-equatorial current north of the equator diminishes east of 35°W (Figure 2b). Also, in summer at this vertical level, the equatorward (southward) off-shore countercurrent, which is part of the NBC retroflection, extends all the way to the equator where it joins the EUC. At 100 m depth, however, this countercurrent merges instead with the NEUC (Figure 3a). The model NBC does not retroreflect in winter. In the lower part of the upper-layer (below 200 m), however, there is an offshore anticyclonic gyre at the same latitude where the NBC retroflects in summer (Figure 2b). The temporal and spatial features of the model upper ocean currents described in this section closely resemble those of their counterparts observed in the tropical Atlantic, which are summarized in [2].

It is interesting that the branching of the model NBC is strongly layered. In the near-surface layer, the NBC feeds the NECC between 6°N and 9°N in winter, and between 4°N and 7°N in summer. Below the near-surface layer, the model NBC originates south of the equator. In the layer between 60 m and 80 m depth, the NBC branches off mostly to the SEUC near 4°S; in the layer between 150 m and 250 m, it branches mostly to the EUC; and in the intermediate layer between these two, it supplies both the SEUC and the EUC. In summer season, at depths below 400 m, the model NBC does not branch off to eastward interior flows until it reaches near 4°N where it retroreflects into an equatorward countercurrent which eventually joins the model EUC. This layered separation pattern of the NBC was also reported in water mass studies, e.g., [3,12].

The fact that the present model reproduced all major features of the currents in the upper couple hundred meters of the tropical Atlantic Ocean suggests that the wind effect here is the deterministic mechanism of current formation and variations. More details of the present numerical simulation are described in [11].

6 Conclusions

The present work shows that high order domain decomposition ocean models can be efficiently implemented on massively parallel architectures, such as the Connection Machine Model CM5. The optimized computational efficiency of the parallel spectral element ocean model comes not only from the exponential convergence of the numerical solution, but also from the work-intensive, medium-grained, geometry-based data parallelism. The data parallelism is created to efficiently implement the spectral element ocean model on the distributed-memory massively parallel computer, which minimizes communication among processing nodes and results in a highly scalable performance. The same advantage of the nonstaggered grid formulation was found in the present parallel, three dimensional, spectral element ocean model as in the earlier spectral element shallow water equation model [6,7].
Acknowledgement

This work was supported in part by DoD Common High Performance Computing Software Support Initiative (CHSSI) and by the U. S. Department of Energy under Contract No. DE-AC02-98CH10886.

References