Electronic Transactions on Numerical Analysis.
Volume 15, pp. 18-28, 2003.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

AN ADDITIVE SCHWARZ PRECONDITIONER FOR THE SPECTRAL ELEMENT OCEAN MODEL FORMULATION OF THE SHALLOW WATER EQUATIONS *

CRAIG C. DOUGLAS†, GUNDOLF HAASE‡, AND MOHAMED ISKANDARANI§

Abstract. We discretize the shallow water equations with an Adams-Bashford scheme combined with the Crank-Nicholson scheme for the time derivatives and spectral elements for the discretization in space. The resulting coupled system of equations will be reduced to a Schur complement system with a special structure of the Schur complement. This system can be solved with a preconditioned conjugate gradients, where the matrix-vector product is only implicitly given. We derive an overlapping block preconditioner based on additive Schwarz methods for preconditioning the reduced system.

Keywords: Shallow water equations, h-p finite elements, adaptive grids, multigrid, parallel computing, conjugate gradients, additive Schwarz preconditioner.

AMS subject classifications. 68W10, 65Y05, 47N40, 76D33

1. Introduction. The shallow water equations are good approximations to the equations of fluid motion whenever the fluid's density is homogeneous, and its depth is much smaller than a characteristic horizontal distance. These equations are often used to model the circulation in coastal areas and in shallow bodies of water. Their virtue is that they reduce the complicated set of the 3D equations to 2D, but are still capable of representing a large part of the dynamics. The shallow water equations also arise frequently in the solution of the 3D primitive hydrostatic equations if the top surface of the fluid is free to move. The presence of the free surface allows the propagation of gravity waves at the speed of $\sqrt{gh}$. The gravity wave speed can greatly exceed the advective velocity of the fluid in the deep part of the ocean and results in a very restrictive CFL limit in order to maintain stability in 3D ocean models [7, 8].

In order to mitigate the cost of ocean simulations, modelers often split the dynamics into a barotropic depth-integrated part (external mode), and a baroclinic part (internal modes). The barotropic equations are akin to the shallow water equations and govern the evolution of the gravity waves. The gravity wave speed stability restriction on the baroclinic part is removed and the 3D baroclinic equations can be integrated explicitly using a large time step. The barotropic equations on the other hand are either integrated explicitly using small time steps that respect the CFL condition for the gravity waves, or implicitly using a stable time-differencing scheme [8].

Implicit integration results in a system of equations that has to be solved at each time step. A direct solver can perform adequately only if the number of unknowns is small. For large problems, however, memory limitations preclude the use of direct solvers and iterative solvers are required. The choice of a robust and efficient solver will determine the cost effectiveness of any implicit method.

* This research has been partially supported by NSF grants DMS-9707040, ACR-9721388, ACR-9814651, CCR-9902022, and CCR-9988165, and National Computational Science Alliance grant OCE980001 (utilizing the University of Illinois Origin 2000 and the University of New Mexico Los Lobos systems). Received May 23, 2001. Accepted for publication October 19, 2001. Recommended by Jan Mandel.

†University of Kentucky, Department of Computer Science, 325 McVey Hall-CCS, Lexington, KY 40506-0045, USA. douglas@ccs.uky.edu. Also, Yale University, Department of Computer Science, P.O. Box 208285, New Haven, CT 06520-8285, USA. douglas-craig@cs.yale.edu.

‡Johannes Kepler University of Linz, Institute of Analysis and Computational Mathematics, Altenberger Str. 69, A–4040 Linz, Austria. ghaase@numa.uni-linz.ac.at.

§University of Miami, Rosenstiel School of Marine and Atmospheric Science, 4600 Rickenbacker Causeway, Miami, FL 33149-1098, USA. MIskandarani@rsmas.miami.edu.

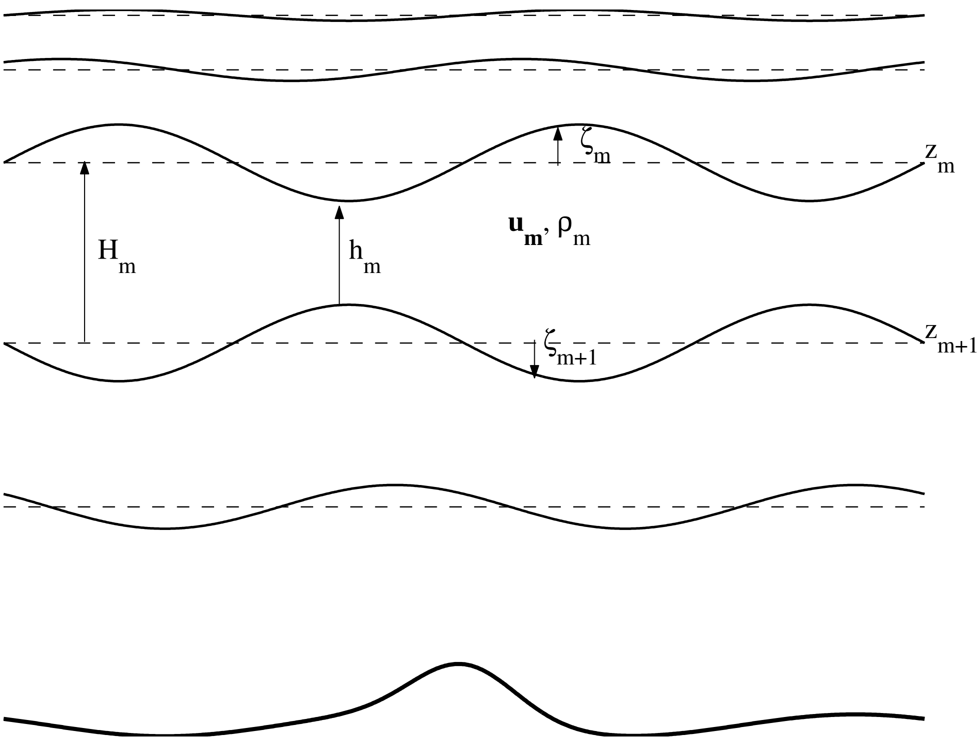

FIGURE 1.1. *Sketch of a multi-layered configurations. There are M-layers numbered from top to bottom. The m-th layer has associated with it the following properties: the layer thickness at rest H_m, the displacement of the upper interface ζ_m and of the lower interface is ζ_{m+1}. The vertical cordinates of the upper and lower interfaces are z_m and z_{m+1}, respectively. The active thickness of the layer is $h_m = z_m - z_{m+1} = H_m + \zeta_m - \zeta_{m+1}$. The total resting depth is $h = \sum_m H_m$.*

We present an alternate time-differencing scheme that treats implicitly solely the terms giving rise to the gravity waves. An Uzawa-like scheme is used to separate the velocity and pressure unknowns. The system of equations for the pressure is symmetric and definite allowing the use of a cheap and robust solver like conjugate gradients (CG). The system matrix therein is never stored explicitly so that only the operation matrix times vector is available. CG was originally accelerated by a diagonal preconditioner that has been replaced with a block diagonal preconditioner based on Additive Schwarz Methods (ASM). The numerical tests show good acceleration of CG with this new preconditioner.

The improved solver has been incorporated into the isopycnal[1] Spectral Element Ocean Model [7, 8]. The novel feature of this model is the combination of isopycnal coordinates in the vertical and spectral element discretization in the horizontal. The benefits of the spectral element discretization include: geometric flexibility, dual h-p paths to convergence, low numerical dispersion and dissipation errors, and dense computational kernels leading to extremely good parallel scalability. We refer the reader to [10, 12, 13, 14] for general information on spectral elements, and to [2, 3, 4, 7, 8, 9, 15, 16] for spectral element applications in geophysical modeling.

Isopycnal models, often referred to as layered models, divide the water column into constant density layers as shown in Fig. 1.1. This division is physically motivated by the fact

[1] An isopycnal is a constant density surface .

ETNA
Kent State University
etna@mcs.kent.edu

that most oceanic currents flow along isopycnal surfaces. Mathematically, it amounts to a mapping from the physical vertical coordinate z to a density coordinate system. The rational for a layered model include: ease of development (since it can be achieved by vertically stacking a set of shallow water models), minimization of cross-isopycnal diffusion, elimination of pressure gradient errors, representation of baroclinic processes, and cost savings over a fully three-dimensional formulation. We refer the reader to [1] and [7] for a general discussion of the pros and cons of isopycnal coordinates. Processes amenable to investigation with the layered model include the wind-driven circulation, eddy generation, and (in part) flow/topography interaction.

2. Derivation of the numerical model. The shallow water equations can be written in the vector form:

$$\vec{u}_t + g\nabla\zeta = \vec{F} \tag{2.1}$$

$$\zeta_t + \nabla \cdot [(h+\zeta)\vec{u}] = Q \tag{2.2}$$

where $\vec{u} = (u, v)$ is the velocity vector, ζ is the sea surface displacement (which, because of hydrostaticity, also stands for the pressure), g is the gravitational acceleration, h is the resting depth of the fluid, Q is a mass source/sink term. Finally, $\vec{F} = (f^x, f^y)$ is a generalized forcing term for the momentum equations that includes the Coriolis force, non-linear advection, viscous dissipation, and wind forcing. Appropriate boundary conditions must be provided to complete the system. For simplicity, we assume no-slip boundary conditions.

The variational form of the above equations in a Cartesian coordinate system is:

$$\int_A \phi u_t dA + \int_A g\zeta_x \phi dA = \int_A f^x \phi dA, \tag{2.3}$$

$$\int_A \phi v_t dA + \int_A g\zeta_y \phi dA = \int_A f^y \phi dA, \tag{2.4}$$

$$\int_A \psi\zeta_t dA - \int_A (\psi_x u + \psi_y v)(h+\zeta) dA = \int_A Q\psi dA. \tag{2.5}$$

where ϕ and ψ denote the test functions for velocity and pressure space. Notice that the divergence operator in the continuity equation has been integrated by parts. The resulting boundary integral represents the amount of fluid injected into the domain. In our case it is identically zero because of the no-slip boundary condition. The integration by parts is important for the solution procedure as it turns the pressure Schur complement into a symmetric positive definite matrix.

The terms responsible for the gravity wave speed are the pressure gradient term in the momentum equation, g$\nabla\zeta$, and the divergence term in the continuity equation $\nabla\cdot(h\vec{u})$. These terms must be integrated implicitly in order to avoid severe time step restrictions. We adopt a semi-implicit integration procedure whereby the gravity terms are discretized with a Crank-Nicholson scheme, and the remaining terms with a third order Adams Bashforth scheme.

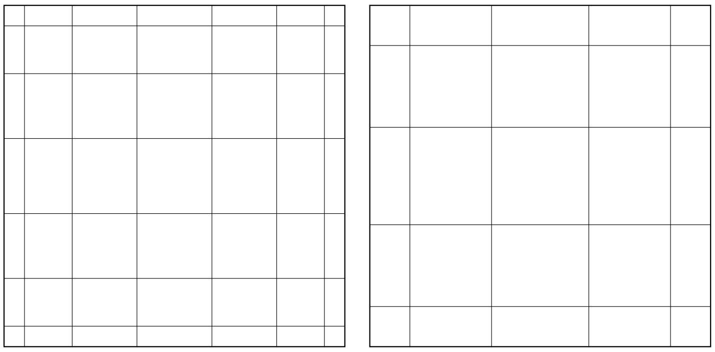

FIGURE 2.1. *Gauss-Lobatto points for the velocity (left) and pressure (right) unknowns in a typical spectral element.*

Letting $\tau = \Delta t$, the time discretized equations become:

$$\begin{aligned}
\frac{1}{\tau}\int_A u^{n+1}\phi dA + \frac{1}{2}\int_A g\zeta_x^{n+1}\phi dA &= \frac{1}{\tau}\int_A u^{n}\phi dA - \frac{1}{2}\int_A g\zeta_x^{n}\phi dA \\
&\quad + \sum_{p=0}^{2}\alpha_p \int_A f^{x(n-p)}\phi dA, \\
\frac{1}{\tau}\int_A v^{n+1}\phi dA + \frac{1}{2}\int_A g\zeta_y^{n+1}\phi dA &= \frac{1}{\tau}\int_A v^{n}\phi dA - \frac{1}{2}\int_A g\zeta_y^{n}\phi dA \\
&\quad + \sum_{p=0}^{2}\alpha_p \int_A f^{y(n-p)}\phi dA, \\
\frac{1}{\tau}\int_A \zeta^{n+1}\psi dA - \frac{1}{2}\int_A \nabla\psi\cdot h\vec{u}^{n+1} dA &= \frac{1}{\tau}\int_A \zeta^{n}\psi dA + \frac{1}{2}\int_A \nabla\psi\cdot h\vec{u}^{n} dA \\
&\quad + \sum_{p=0}^{2}\alpha_p \left(\int_A Q^{n-p}\psi + \nabla\psi\cdot\vec{u}^{n-p}\zeta^{n-p}\right) dA.
\end{aligned}$$

For spatial discretization, we employ the spectral element method. The computational domain is divided into elements where the unknowns are interpolated with Legendre cardinal functions collocated at the Gauss-Lobatto points (see Fig. 2.1), so that for each element we can write

$$\begin{aligned}
\vec{u}(\xi,\eta) &= \sum_{k,l=1}^{N^v} \vec{u}_{k,l}\,\mu_k^v(\xi)\mu_l^v(\eta), \\
\zeta(\xi,\eta) &= \sum_{i,j=1}^{N^p} \zeta_{i,j}\,\mu_i^p(\xi)\mu_j^p(\eta)\ \ ,
\end{aligned}$$

with μ_k^v and μ_i^p as local 1D basis functions for velocity and pressure, respectively. We use a lower order polynomial for the pressure in order to suppress spurious pressure modes [8]. Hence, $N^p = N^v - 2$.

The Galerkin formulation reduces the variational equations in each time step to a system of algebraic equation after substituting the above formula. By the assembly procedure and

ETNA
Kent State University
etna@mcs.kent.edu

with obvious notation for a, b and c, the system of equations can be written as:

$$\begin{aligned} \frac{1}{\tau} M^v u + \frac{1}{2} g G^x \zeta &= a, \\ \frac{1}{\tau} M^v v + \frac{1}{2} g G^y \zeta &= b, \\ -\frac{1}{2} D^x u - \frac{1}{2} D^y v + \frac{1}{\tau} M^p \zeta &= c. \end{aligned} \tag{2.6}$$

where M^v, and M^p are the mass matrices for the velocity and pressure unknowns, G^x and G^y are the discrete gradient operators in the (x, y) directions, and D^x and D^y are the components of the discrete gradient operator.

The integrals resulting from the spatial discretization are evaluated using Gauss-Lobatto quadrature of the same order as the interpolation polynomial. This results in a diagonal matrix for both velocity and pressure. At the element level (before assembly), the mass matrices have the following entries:

$$\begin{aligned} M^v_{ij,kl} &= \int_{-1}^{1} \int_{-1}^{1} \mu^v_i(\xi) \mu^v_j(\eta) \mu^v_k(\xi) \mu^v_l(\eta) |J| d\xi d\eta \\ &\approx \delta_{ik} \delta_{jl} \omega^v_k \omega^v_l |J|^v_{kl}, \end{aligned}$$

Here ω^v_k denotes the weight for the integration on the velocity points in an element, and $J^v_{kl} = J(v(x_k, y_l))$ is the Jacobian of the mapping between (x, y) space and the (ξ, η) computational space. The same holds for the pressure mass matrix

$$M^p_{ij,kl} = \delta_{ik} \delta_{jl} \omega^p_k \omega^p_l |J|^p_{kl}.$$

The evaluation of the other integrals is more involved since the corresponding matrices are not diagonal. After algebraic manipulation, we obtain

$$\int_A \zeta_x \phi dA = \sum_{k,l} G^x_{ij,kl} \zeta_{kl}, \text{ and } \int_A \zeta_y \phi dA = \sum_{k,l} G^y_{ij,kl} \zeta_{kl},$$

where

$$G^x_{ij,kl} \equiv \mu^{p\prime}_k(\xi^v_i) \mu^p_l(\eta^v_j) \omega^v_i \omega^v_j |J|^v_{ij} \; \xi_x|^v_{ij} + \mu^p_k(\xi^v_i) \mu^{p\prime}_l(\eta^v_j) \omega^v_i \omega^v_j |J|^v_{ij} \; \eta_x|^v_{ij}$$

and

$$G^y_{ij,kl} \equiv \mu^{p\prime}_k(\xi^v_i) \mu^p_l(\eta^v_j) \omega^v_i \omega^v_j |J|^v_{ij} \; \xi_y|^v_{ij} + \mu^p_k(\xi^v_i) \mu^{p\prime}_l(\eta^v_j) \omega^v_i \omega^v_j |J|^v_{ij} \; \eta_y|^v_{ij} \,.$$

The factors ξ_x, ξ_y, η_x, and η_y are metrics of the elemental mapping. They are evaluated on the velocity grid points (ξ^v_i, η^v_j). Note that we have used the higher order quadrature on the velocity grid to evaluate the discrete gradient operator.

For the divergence terms and recalling that (i, j) correspond to velocity points and (k, l) to pressure points, we have

$$D^x_{ij,kl} = h_{kl} G^x_{kl,ij} \text{ and } D^y_{ij,kl} = h_{kl} G^y_{kl,ij}.$$

where, thanks to the continuity of h_{kl}, the relation $D^s = (G^s)^T H$ holds at a global level (H is a global and diagonal matrix whose entries are the resting depths at the collocation points.).

ETNA
Kent State University
etna@mcs.kent.edu

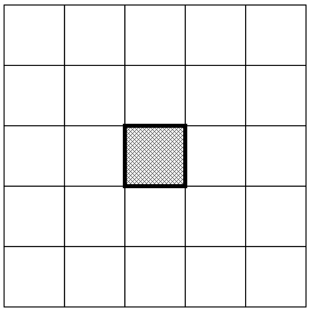

FIGURE 3.1. *Elements that have non-zero matrix entries with nodes of the marked element.*

3. Solution of the reduced system. By substitution and using the fact that $D^s = (G^s)^T H$ and that $M^v = M^u$ and H are diagonal matrices, the system (2.6) can easily be reduced to a symmetric system of equations in ζ since

$$S\zeta := \left[\frac{M^p}{\tau} + \frac{\tau g}{4}\left((G^x)^T H(M^v)^{-1}G^x + (G^y)^T H(M^v)^{-1}G^y\right)\right]\zeta = f \tag{3.1}$$

with the Schur complement S and the right hand side

$$f := c - \frac{\tau}{2}\left[(G^x)^T H(M^v)^{-1}a + (G^y)^T H(M^v)^{-1}b\right].$$

Matrix S is only available via the matrix-vector operation $S \cdot w$. An explicit storing of S is very memory consuming since there exist many entries from nodes of an element to nodes in elements two layers away (see Fig. 3.1). Therefore, S is never stored explicitly. Additionally, the implemented implicit matrix-vector multiplication is as fast as conventional multiplication since it takes advantage of the local tensor product structure of the matrix.

Note, that system (3.1) is similar to a discretization of the equation

$$-\frac{g}{4}\nabla^T\left(h(x)/m(x)\nabla\zeta(x)\right) + \frac{1}{\tau^2}\zeta(x) = \frac{1}{\tau}f(x), \qquad x \in \Omega. \tag{3.2}$$

The matrix $S_{N\times N}$ is obviously symmetric and positive definite so that system (3.1) can be solved using a preconditioned CG. A simple diagonal preconditioner of S is defined using the available matrix-vector multiplication

$$\{C_{ii}\}_{i=1}^N := \{(S\cdot 1)_i\}_{i=1}^N, \tag{3.3}$$

i.e., the diagonal preconditioner C is the lumped Schur complement.

4. Additive Schwarz preconditioner. A better preconditioner for S from (3.1) can be derived as an Additive Schwarz Method (ASM) preconditioner. We will denote the restriction from a global vector to the local components of an element r by $\mathcal{A}_r : \mathbb{R}^N \mapsto \mathbb{R}^{N_r}$, $r = 1, \ldots, \text{nelem}$.

The original matrix S can no longer be expressed in terms of element matrices since these matrices are no longer locally on the element because of the long connections between nodes of all surrounding elements as shown in Fig. 3.1. If we consider only local element contributions in the generation of local matrices $\widehat{S}_r$, i.e., we restrict the support of the transformed

ETNA
Kent State University
etna@mcs.kent.edu

pressure basis functions to the element, then we can derive a matrix

$$
\begin{aligned}
(4.1)\quad \widehat{S} &= \sum_{r=1}^{\text{nelem}} \mathcal{A}_r^T \widehat{S}_r \mathcal{A}_r \\
&= \sum_{r=1}^{\text{nelem}} \mathcal{A}_r^T \left[\frac{M_r^p}{\tau} + \frac{\tau g}{4} \left((G_r^x)^T H (M_r^v)^{-1} G_r^x + (G_r^y)^T H (M_r^v)^{-1} G_r^y \right) \right] \mathcal{A}_r.
\end{aligned}
$$

The local matrices are similar to equation (3.2) in the domain Ω_r with homogeneous Neumann conditions on the element boundary $\partial\Omega_r$. This provides two ways of calculating the $\widehat{S}$ matrices. First, we can derive $\widehat{S}$ as stiffness matrix of the local problem (3.2). Second, we can use a modified matrix times vector routine that operates only locally in the element. Applying this routine to a vector $[0, \ldots, 0, 1_j, 0, \ldots, 0]^T$ results in column entries $S_{r,i,j}$ for all local rows i. This second version is a more straightforward approach, but consumes more CPU time than the first one.

We realized the second version. Besides being much easier to implement, it is only a once a run expense. Since a normal run is a very large number of time steps (usually tens of thousands of time steps), the added cost over the first version is negligible over an entire simulation.

The matrices $\widehat{S}_r$ really represent the local properties of S and it is no surprise that $S \cdot 1 \equiv \widehat{S} \cdot 1$ holds, i.e., $\widehat{S}$ is some sort of a blockwise lumped S. All matrices $\widehat{S}_r$ are non-singular as long as τ does not tend to infinity. Therefore we can derive an ASM preconditioner for $\widehat{S}$ and also for S defined by the inverse

$$
(4.2)\qquad \widehat{C}^{-1} = W^{-1} \sum_{r=1}^{\text{nelem}} \mathcal{A}_r^T \widehat{S}_r^{-1} \mathcal{A}_r.
$$

Matrix $\widehat{C}^{-1}$ is not block diagonal but its blocks overlap only in rows/columns shared by more than one element. This requires in (4.2) the diagonal weight matrix $W = \sum_{r=1}^{\text{nelem}} \mathcal{A}_r^T \mathcal{A}_r$ containing the number of elements a node belongs to, i.e., these are the weights needed for a partition of unity. An abstract analysis of preconditioners similar to $\widehat{C}$ can be found in [5, 17] and the references therein.

The parallelization of the preconditioner $\widehat{C}^{-1}$ is straightforward and requires only one next neighbor communication per application [6].

5. Numerical results. We tested the new solver on a two sample problems. Our focus is comparing the new preconditioner $\widehat{C}$ from (4.2) with the old preconditioner C from (3.3). The preconditioned CG stops when the relative error measured in the $\| \cdot \|_{SC^{-1}S}$ norm, respective in the $\| \cdot \|_{S\widehat{C}^{-1}S}$ norm, is reduced by 10^{-4}. We note that the solution of the shallow water equations consummed a large portion of the model CPU time and improvements in the model performance translates into a substantial reduction in the overall cost.

In our tests, we integrate the isopycnal spectral element ocean model for 200 time steps, which is enough to get a good feel for how the simulation will run. A complete simulation can involve millions of time steps to complete a 14-16 year study, though the simulations are usually run a few tens of thousands of time steps at a time. The model is configured with 5 layers in both cases and is forced with observed winds stresses obtained from the National Center for Environmental Prediction. Seventh and fifth order spectral elements are used for the velocity and pressure, respectively.

Our first example consisted of calculating the wind-driven circulation in the North Atlantic Basin (NAB08). The grid is relatively small and has coarse resolution: it has 792

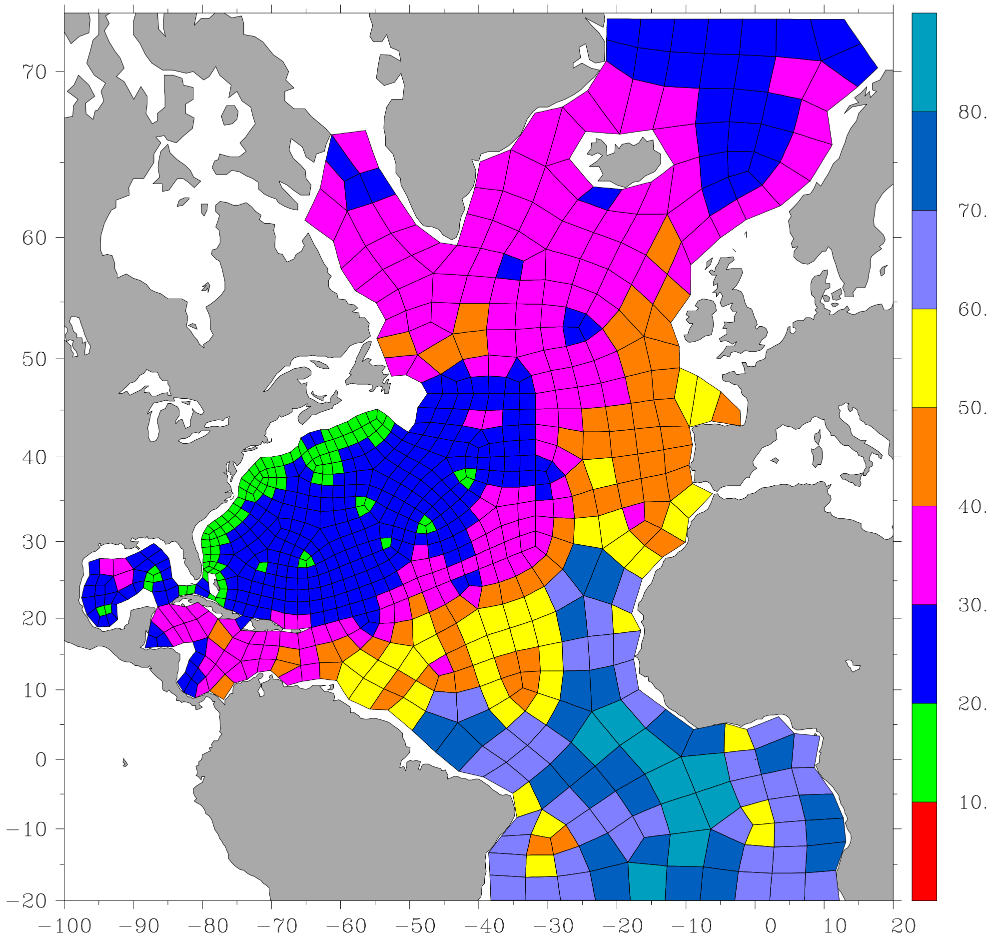

FIGURE 5.1. *North Atlantic grid (NAB08) with surface mesh spacing (km) color coded.*

	t_{Uzawa} Seconds	Acceleration
Old: C	334	1.0
New: $\widehat{C}$	134	2.8

TABLE 5.1
North Atlantic example on one processor.

elements, 39610 velocity nodes, and 20372 pressure nodes. The surface grid is presented in Fig. 5.1. The elements have been refined in the Gulf Stream region to improve the (smaller scale) dynamical features of the Gulf Stream.

The use of the new preconditioner reduced the number of CG iterations from 80-85 down to 17-19. The application of $\widehat{C}$ is more expensive than the old one preconditioner C and therefore the gain in performance will be less than 4. We performed our test on an ORIGIN 3800 with 400MHz MIPS R14000 with 8MB L2-cache and 512MB main memory per processor. A test on a single processor resulted in the CPU times contained in Table 5.1.

The second example (NEP08) involves a larger grid with a greater disparity in element sizes; see Fig. 5.2. The focus of this simulation is the Northern Pacific Ocean, with particular

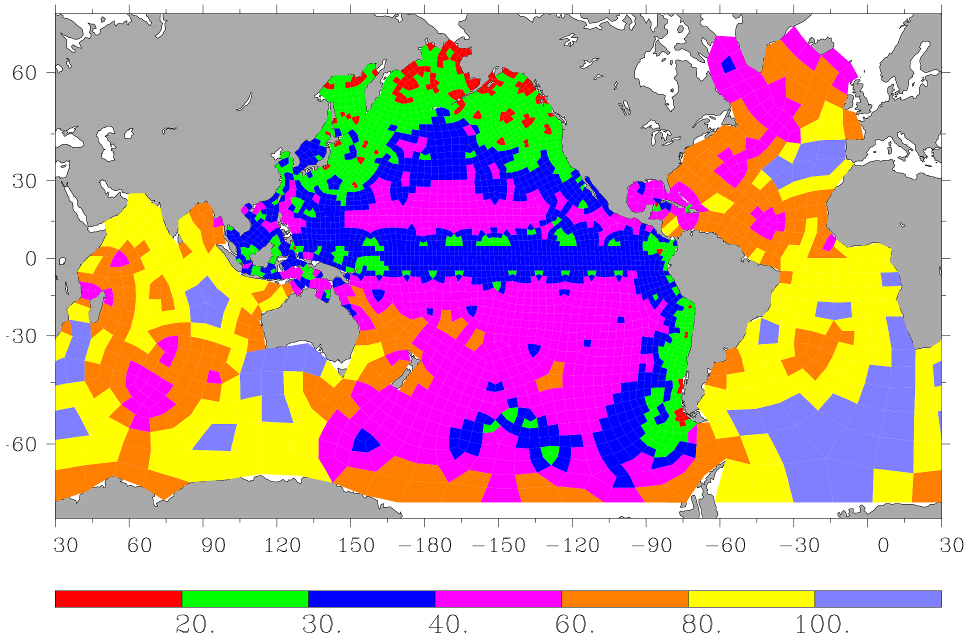

FIGURE 5.2. *Northeast Pacific grid (NEP08) with surface mesh spacing (km) color coded.*

Processors:	2	4	6	8	12	16	32	48
SGI Origin 3800								
Old: C	410.3	212.2	140.4	98.2	72.7	55.3	29.5	26.5
New: $\widehat{C}$	146.1	88.4	45.1	43.1	20.1	17.5	12.1	8.9
Acceleration	2.8	2.4	3.1	2.3	3.6	3.2	2.4	3.0
SGI Origin 2000								
Old: C	940.2	378.6	230.4	164.4	110.7	85.2	47.0	40.0
New: $\widehat{C}$	306.4	141.5	89.0	64.8	46.0	28.6	17.4	11.6
Acceleration	3.1	2.7	2.6	2.5	2.4	3.0	2.7	3.4

TABLE 5.2
North East Pacific example, in wall clock CPU seconds.

interest in the Gulf of Alaska, and the equatorial and northern coastal wave guide. We have included the global ocean in our simulation to avoid open boundaries. The grid emphasizes our areas of interests which are discretized with small elements with a typical grid resolution ranging from 20 km in the Gulf of Alaska to about 35 km in the wave guides. The element sizes are increased gradually away from the Pacific Ocean in order to reduce the cost of the global simulation. The grid contains 3552 elements, 176377 velocity nodes, and 90459 pressure nodes.

In this example, we are interested in the parallel performance of the two preconditioners. The domain is partitioned into p subdomains, where $p \in \{2, 4, 6, 8, 12, 16, 32, 48\}$. One subdomain is used per processor. We used METIS [11] to calculate the subdomains (see Fig 5.3 for the case of $p = 4$). METIS is only used once per domain and per the number of processors as a preprocessing step. Hence, its cost is zero in a simulation. The results are contained in Table 5.2.

ETNA
Kent State University
etna@mcs.kent.edu

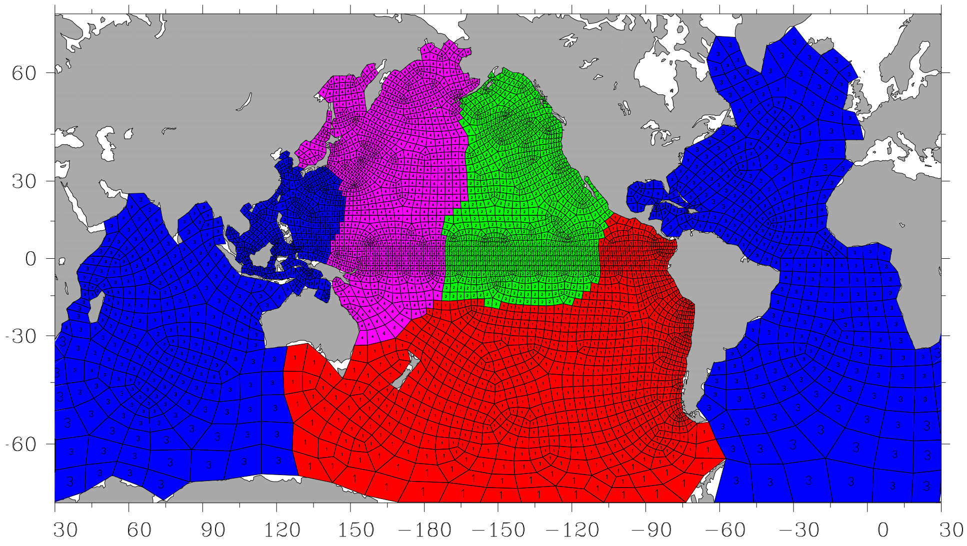

FIGURE 5.3. *Northeast Pacific surface grid (NEP08) with 4 subdomains.*

6. Conclusions. The reduction of the coupled system (2.6) to the reduced system (3.1) cannot provide us with an explicit matrix and so the local approximation of the matrix of the reduced system (4.1) is the only way to derive a good preconditioner for CG. The preconditioner setup costs approximately 10 Uzawa iterations but this can be neglected with respect to the overall solution time, especially when we have to solve (2.6) for tens of thousands to millions of time steps.

The new preconditioner $\widehat{C}^{-1}$ from (4.2) reduces the number of iterations in the Uzawa algorithm by a factor of 4 and the wall clock CPU time by a factor of 3 in the presented examples. Trying to find new preconditioners that provide improvements with respect to the time discretization is part of our future research plan.

REFERENCES

[1] R. BLECK, *Ocean modeling in isopynic coordinates*, in Ocean Modeling and Parameterization, Kluwer Academic Publishers, Dordrecht, Boston, London, pp. 423–448.

[2] E. N. CURCHITSER, D. HAIDVOGEL, AND M. ISKANDARANI, *On the transient adjustment of a mid-latitude abyssal ocean basin with realistic geometry: The constant depth limit*, Dynamics of Atmosphere and Oceans, 29 (1999), p. 147.

[3] ———, *On the transient adjustment of a mid-latitude abyssal ocean basin with realistic geometry and bathymetry*, JPO, 31 (2001), pp. 725–745.

[4] E. N. CURCHITSER, M. ISKANDARANI, AND D. B. HAIDVOGEL, *A spectral element solution of the shallow water equations on multiprocessor computers*, J. Atmos. Ocean. Techn., 15 (1998), pp. 510–521.

[5] M. GRIEBEL AND P. OSWALD, *On the abstract theory of additive and multiplicative Schwarz algorithms*, Numer. Math., 70 (1995), pp. 163–180.

[6] G. HAASE, *New matrix-by-vector multiplications based on a non-overlapping domain decomposition data distribution*, in Proceedings of the Euro-Par97, vol. 1300 of Lecture Notes in Computer Sciences, Heidelberg, 1997, Springer, pp. 726–733.

[7] D. B. HAIDVODEL AND A. BECKMANN, *Numerical Ocean Circulation Modeling*, vol. 2 of Environmental Science and Management, Imperial College Press, London, 1999.

[8] M. ISKANDARANI, D. B. HAIDVOGEL, AND J. P. BOYD, *A staggered spectral element model with application to the oceanic shallow water equations*, International Journal for Numerical Methods in Fluids, 20 (1995), pp. 393–414.

[9] M. ISKANDARANI, D. B. HAIDVOGEL, AND J. LEVIN, *A three-dimensional spectral element model for the solution of the hydrostatic primitive equations*, Journal of Computational Physics, (2001). submitted.

[10] G. E. KARNIADAKIS AND S. SHERWIN, *Spectral/hp Element Methods for CFD*, Oxford University Press, 1999.

[11] G. KARYPIS, *METIS serial graph partitioning and matrix ordering software*. In URL http://www-users.cs.umn.edu/∼karypis/metis/metis/main.shtml, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA.

[12] Y. MADAY, D. MEIRON, A. PATERA, AND E. RONQUIST, *Analysis of iterative methods for the steady and unsteady stokes problem: application to spectral element discretizations*, SIAM Journal of Scientific and Statistical Computing, 14 (1993), p. 310.

[13] E. RONQUIST, *Optimal Spectral Element Methods for the Unsteady Three-Dimensional Incompressible Navier-Stokes Equations*, PhD thesis, Massachusetts Institute of Technology, Cambridge MA, 1988.

[14] ———, *A domain decomposition method for elliptic boundary value problems: Application to unsteady incompressible fluid flow*, in Proceedings of the Fifth Conference on Domain Decomposition for Partial Diffential Equations, Norfolk, Virginia, 1991, pp. 545–557.

[15] M. TAYLOR, J. TRIBBIA, AND M. ISKANDARANI, *The spectral element method for the shallow water equations on the sphere*, Journal of Computational Physics, 130 (1997), pp. 92–108.

[16] B. WINGATE AND J. P. BOYD, *Spectral element methods on triangles for geophysical fluid dynamics problems*, in Proceedings of the Third International Conference On Spectral and High-Order Methods, Houston, Texas, 1996, Houston Journal of Mathematics, pp. 305–314.

[17] J. XU, *Iterative methods by space decomposition and subspace correction: A unifying approach*, SIAM Review, 34 (1992), pp. 581–613.