Ocean Modeling

John Dukowicz, Richard Smith and Robert Malone

The Parallel Ocean Program (POP) was developed at Los Alamos National Laboratory (LANL) under the sponsorship of the Department of Energy's CHAMMP program, which brought massively parallel computers to the realm of climate modeling. POP is a descendant of the Bryan-Cox-Semtner (BCS) class of models (Bryan, 1969; Cox, 1975; Semtner, 1974). A number of improvements to the standard BCS model have been developed and incorporated in POP. Although originally motivated by the adaptation of POP for massively parallel computers, in particular the Connection Machine (CM-5), many of these changes improved not only its computational performance but the model's physical representation of the ocean as well. The most significant of these improvements are summarized below. Details can be found in articles by Smith, *et al.* (1992); Dukowicz, *et al.* (1993); Dukowicz and Smith (1994); and Smith, *et al.* (1995).

The Bryan-Cox-Semtner ocean model is a three-dimensional model in Eulerian coordinates (latitude, longitude, and depth). The incompressible Navier-Stokes equations and equations for the transport of temperature and salinity, along with a turbulent eddy viscosity and diffusivity, are solved subject to the hydrostatic and Boussinesq approximations. As originally formulated, the model includes a rigid-lid approximation (zero vertical velocity at the ocean surface) to eliminate fast surface waves. The presence of such waves would require use of a very short time step in numerical simulations and hence greatly increase the computational cost. The equations of motion are split into two parts: a set of two-dimensional "barotropic" equations describing the vertically averaged flow, and a set of three-dimensional "baroclinic" equations describing temperature, salinity, and deviation of the horizontal velocity components from the vertically averaged flow. (The vertical velocity component is determined from the constraint of mass conservation.) The barotropic equations contain the fast surface waves and separate them from the rest of the model.

The baroclinic equations are solved explicitly; that is, their solution involves a simple forward time-stepping scheme, which is well suited to parallel computing. On the other hand, the barotropic equations (two-dimensional sparse-matrix equations linking nearest-neighbor grid points) are solved implicitly; that is, they are solved at each time step by iteration.

For historical reasons, the barotropic equations in the Bryan-Cox-Semtner model are formulated in terms of a streamfunction. Such a formulation requires solving an additional equation for each island, an equation that links all points around the island. This was not a problem when limited computing power would permit only very coarse resolution ($\geq 5°$ in latitude and longitude), because only continent-size landmasses could be resolved. As the model was pushed to higher resolution, not only were there many additional equations to solve, but each equation required "gather-scatter" memory accesses on each solver iteration. This was costly even on machines with fast memory access like Cray parallel-vector-processor computers. To reduce the number of equations to solve, it was common practice to submerge islands, connect them to nearby continents with artificial land bridges, or merge an island chain into a single mass without gaps. The first modification created artificial gaps, permitting increased flow, while the latter two closed channels that should exist. For example, in the pioneering work of Semtner and Chervin (1988, 1992), of the eighty islands resolvable at the horizontal resolution employed (0.5 degrees latitude and longitude), all but the three largest “islands” (Antarctica, Australia and New Zealand) were eliminated by artificial changes in the bottom topography. Even then, the barotropic part of the code consumed about one-third of the total computing time when the model was executed on a Cray. On distributed-memory parallel computers, these added equations became even more costly because, on every iteration, each required gathering data from a (possibly large) set of processors to do a summation around each island. When the model was executed on a Connection Machine, about two-thirds of the total computing time was spent on the barotropic part.

Surface-pressure formulation of barotropic mode

The above considerations led us to focus our efforts on speeding up the barotropic part of the code. We developed and implemented two new numerical formulations of the barotropic equations, both of which involve a surface-pressure field rather than a streamfunction. The surface-pressure formulations have several advantages over the streamfunction formulation and are more efficient on both distributed-memory parallel and shared-memory vector computers.

The first new formulation recasts the barotropic equations in terms of a surface-pressure field but retains the rigid-lid approximation. The surface pressure represents the pressure that would have to be applied to the surface of the ocean to keep it flat (as if capped by a rigid lid). The barotropic equations must still be solved implicitly, but the boundary conditions are simpler and much easier to implement. Furthermore, islands require no additional equations, and therefore any number of islands can be included in the grid at no extra computational cost. Perhaps most importantly, the surface-pressure, rigid-lid formulation, unlike the streamfunction, rigid-lid formulation, exhibits no convergence problems due to steep gradients in the bottom topography. The matrix operator in the surface-pressure formulation is proportional to the depth field H, whereas the matrix operator in the streamfunction formulation is proportional to 1/H. Therefore, the latter matrix operator is much more sensitive than the former to rapid variations in the depth of waters over the edges of continental shelves or submerged mountain ranges. In such situations, the depth may change from several thousand meters to a few tens of meters within a few grid points. Because such a rapidly varying operator may prevent convergence to a solution, steep gradients were removed from the streamfunction formulation by smoothing the depth field (which also had the then-desirable effect of eliminating many islands). The surface-pressure formulation, on the other hand, converges even in the presence of steep depth gradients. Artificial smoothing of the depth field can significantly affect the accuracy of a numerical simulation of the interaction of a strong current with bottom topography. For example, the detailed course and dynamics of the Antarctic Circumpolar Current (the strongest ocean current in terms of total volume transport) is greatly influenced by its interaction with bottom topography.

As we worked with the surface-pressure, rigid-lid model, we noticed a problem in shallow isolated bays such as the Sea of Japan. In principle, we should have been able to infer the elevation of the ocean surface (relative to the mean elevation) from the predicted surface pressure. We found, however, that the surface heights so inferred were quite different from those expected due to inflow or outflow from the bays. Removing the rigid lid solved that problem, but of course it also brought back the unwanted and unneeded surface waves. We were able to overcome that new difficulty by treating the terms responsible for the surface waves implicitly, which artificially slows the waves, whereas the rigid-lid approximation artificially speeds up the waves to infinite velocity. (Either departure from reality is acceptable: Climate modeling does not require an accurate representation of the waves because they have little effect on the ocean circulation.)

Free-surface formulation

Those considerations led us next to abandon the rigid-lid approximation in favor of a free-surface formulation. The surface pressure is then proportional to the mass of water above a reference level near the surface. The benefits of the surface-pressure, free-surface model are greater physical realism and faster convergence of the barotropic solver. In particular, the revised barotropic part of the code, including eighty islands, is many times faster than the original, which included only three islands (when both are implemented on the 0.5º grid). In addition, the surface pressure is now a prognostic variable that can be compared to global satellite observations of sea-surface elevation to validate the model, and satellite data can now be assimilated into the model to improve short-term prediction of near-surface ocean conditions.

None of our revisions, of course, changed the fact that the large matrix equation in the barotropic solver must be solved implicitly. We chose to use conjugate-gradient methods for that purpose because they are both effective and easily adapted to parallel computing. Conjugate-gradient methods are most effective when the matrix is symmetric. Unfortunately, the presence of Coriolis terms (terms associated with the rotation of the earth) in the barotropic equations makes the matrix non-symmetric. By using an approximate factorization method to split off the Coriolis terms, we retained the accuracy of the time-discretization of the Coriolis terms and produced a symmetric matrix to which a standard conjugate-gradient method may be applied. We also developed a new preconditioning method for use on massively parallel computers that is very effective at accelerating the convergence of the conjugate-gradient solution. The method exploits the idea of a local approximate inverse to find a symmetric preconditioning matrix. Calculating the pre-conditioner is relatively expensive but only needs to be done once for a given computational grid.

Pressure-averaging

Elimination of the extra equations for islands and the associated gather-scatter memory operations greatly reduced the cost of solving the barotropic equations. Further savings can be obtained by implementing “pressure averaging”, a well known technique in atmospheric modeling for increasing the time step (Brown and Campana, 1978). After the temperature and salinity have been updated to time-step n+1 in the baroclinic routines, the density ρ^{n+1} and pressure p^{n+1} can be computed. By calculating the pressure gradient with a linear combination of p at three time-levels (n-1, n, and n+1), it is possible to increase the time-step by as much as a factor of two. However, at first this doubling was not obtained because something else was limiting the time step. Analysis of factors constraining the time step revealed that it was being limited by horizontal diffusion at high latitudes, as described next.

Latitudinal scaling of horizontal diffusion

Horizontal mixing by unresolved turbulence is commonly parameterized by either Laplacian (∇^2) or biharmonic (∇^4) diffusion terms. These operators scale as Δx^{-m}, with m=2 or 4. Here $\Delta x = a*\Delta\lambda*\cos \varphi$, where φ and λ are latitude and longitude, respectively, and 'a' is the radius of the earth. Because $\cos \varphi \rightarrow 0$ at the poles, these diffusion terms become very large at high latitude. Although horizontal diffusion parameterizations are intended to mimic the effects of unresolved turbulence, their essential purpose is to dissipate energy at scales near the grid resolution. Consequently, they can be arbitrarily re-scaled as long as they give sufficient dissipation to prevent the build-up of computational noise at small spatial scales. The diffusion term ($\nabla^m T$) only needs to be big enough at all latitudes to balance the advection term ($U \bullet \nabla T$) in the transport equation for tracer T. The advection term scales as Δx^{-1}, so scaling of the horizontal diffusion coefficient by $(\cos \varphi)^n$ was introduced, where n=m-1 (n=1 for Laplacian mixing and n=3 for biharmonic mixing). This scaling prevents horizontal diffusion from limiting the time step severely at high latitudes, yet keeps diffusion large enough to maintain numerical stability.

Once this scaling was introduced and the associated time step constraint was removed, the doubling of the time step with the "pressure-averaging" method was attained. Taken together, the improved numerical stability of the surface-pressure formulations, the $(\cos \varphi)^3$ tapering of the biharmonic diffusion coefficient, and pressure-averaging permitted the time step to be increased by about a factor of four compared to the best calculations at that time (Semtner and Chervin, 1988, 1992). They used a time step of 15 minutes when running a standard BCS model at 0.5º resolution. With POP, it was possible to run with a 30-minute time step at 0.28º resolution, an improvement of a factor of four over the 7.5-minute time step expected by extrapolating Semtner's experience.

Designed for parallel computers

The code is written in Fortran90 and can be run on a variety of parallel and serial computer architectures. It uses domain decomposition in latitude and longitude, combined with MPI for inter-processor communications on distributed memory machines. SHMEM is also available on machines that support it (SGI Origin 2000 and Cray T3E).

General orthogonal coordinates and the "displaced-pole" grid

Because the code is written in Fortran90, it was relatively easy to reformulate and discretize the equations of motion to allow the use of any locally orthogonal horizontal grid without a major re-write of the code (Smith, *et al.*, 1995). This generalization provides alternatives to the standard latitude-longitude grid with its singularity at the North Pole. In particular, a "displaced-pole" grid was developed, in which the singularity arising from convergence of meridians at the North Pole is moved into an adjacent landmass such as North America, Greenland, or Russia. This leaves a smooth, singularity-free grid in the Arctic Ocean, which is important for the modeling of sea ice.

That grid joins smoothly at the equator with a standard Mercator grid in the Southern Hemisphere. If the singularity is moved to Greenland, distortion relative to the standard grid is minimized but the smallness of ocean cells just off the coast of Greenland may restrict the time step excessively. Placing the singularity in either Greenland or North America increases the resolution in the Gulf Stream and the northern seas; the Gulf Stream transports warm salty water into the northern seas, where deep water is formed by wintertime convection. Both the transport and convection are important aspects of the global thermohaline circulation that need to be as well resolved as possible.

The displaced-pole grid has proven to be one of the most popular features of POP, especially in fully coupled atmosphere-ocean-sea ice models. The Los Alamos sea ice model (CICE) also supports the displaced-pole grid, so no interpolation is needed between POP and CICE. A package based on conservative remapping techniques, the Spherical Coordinate Remapping and Interpolation Package (SCRIP), has been developed (Jones, 1999) that transforms state variables and fluxes between any pair of orthogonal grids on the sphere. SCRIP handles the transformations between the atmospheric model grid and the displaced-pole grid used by POP and CICE.

Many of the improvements first introduced in POP have been adopted in other models, even for use on parallel-vector machines.

High-resolution simulations enabled by POP

Massively parallel computers are ideally suited to high-resolution modeling of the oceans. “Meso-scale” eddies in the oceans are 50-100 km in size, roughly ten times smaller than their atmospheric analogues: high and low pressure and frontal systems. Thus, ocean models need to have finer grids than atmospheric models. Cost rises rapidly as resolution is increased: doubling the horizontal resolution increases the cost by an order of magnitude when the reduction in time step and a modest increase in vertical resolution are taken into account. At the time POP was being developed, the state of the art in high-resolution global modeling was the work of Semtner and Chervin (1988, 1992) at 0.5°. They were using a model with the standard rigid-lid streamfunction formulation, smoothed bottom topography with only three “islands”, bi-harmonic diffusion and no pressure averaging; the model time step was 15 minutes at 0.5° resolution. With POP running on the CM-5, it was possible to double the resolution to 0.28°, use unsmoothed bottom topography, include all 112 resolvable islands, and run with a time step of 30 minutes. Although many aspects of the 0.28° global simulations were improved compared to the earlier simulations, quantitative comparisons of sea-surface height variability predicted by POP with measurements from the TOPEX/Poseidon satellite altimeter showed that the model variability was still a factor of two low. This meant that the mesoscale eddy spectrum was still not adequately resolved. Limitations in computing power made it impractical to go to higher resolution at the global scale, however it was feasible to go to 0.1° in the Atlantic Ocean basin only. That calculation had about the same number of horizontal grid points (992 x 1280) as the global 0.28° calculation (1280 x 896) but 40 depth levels rather than 20. The time step had to be reduced by the resolution ratio (2.8) to 10 minutes, so 3 times as many time steps were needed to integrate the model for a decade of simulated time. With twice as many depth levels, the 0.1° calculation was 6 times more expensive than a similar length

0.28° run. Four months of almost dedicated time on 512 processors of the LANL/ACL CM-5 were needed to complete the calculation. Many important aspects of the Atlantic circulation were accurately captured for the first time, including good quantitative agreement between POP and TOPEX/Poseidon. The results are so impressive that the international oceanographic community is eagerly awaiting a global simulation at the same scale (0.1°). This has been impossible until this year (1999), when the ACL took delivery of a 2048 processor SGI Origin 2000 system with a peak rating of 1 TF. Benchmark tests of POP indicate that roughly six months of non-stop computing on 512 processors will be required to extend the 0.1° simulation to the global scale. The grid will have 3600 points in longitude, 2400 in latitude, and 40 depth levels, a total 3.5×10^8 grid cells. With a 10-minute time-step, nearly a million time steps will be needed.

References

J. A. Brown and K. A. Campana, 1978. An economical time-differencing system for numerical weather prediction. *Mon. Weather Rev*., **106**, 1125-1136.

K. Bryan, 1969. A numerical method for the study of the circulation of the world ocean. *Journal of Computational Physics,* **4**, 347-376.

M. D. Cox, 1975. A baroclinic numerical model of the world ocean: Preliminary results. In *Numerical Models of Ocean Circulation*, pp. 107-118, National Academy of Sciences, Washington, D.C.

J. K. Dukowicz and R. D. Smith, 1994. Implicit free-surface method for the Bryan-Cox-Semtner ocean model. *J. Geophys. Res.*, **99**, 7991-8014.

J. K. Dukowicz, R. D. Smith, and R. C. Malone, 1993. A reformulation and implementation of the Bryan-Cox-Semtner ocean model on the Connection Machine. *Journal of Atmospheric and Oceanic Technology,* **10**, 195-208.

P. W. Jones, 1999. First- and second-order conservative remapping schemes for grids in spherical coordinates. *Monthly Weather Review*, **127**, 2204-2210.

A. J. Semtner, 1974. An oceanic general circulation model with bottom topography. Tech. Rep. #9, 99pp., Dept. of Meteorology, Univ. of California, Los Angeles.

A. J. Semtner and R. M. Chervin, 1988. *A* simulation of the global ocean circulation with resolved eddies. *Journal of Geophysical Research,* **93**, 15502-15522.

R. D. Smith, J. K. Dukowicz, and R. C. Malone, 1992. Parallel ocean general circulation modeling. *Physica D,* **60**, 8-61.

R. D. Smith, S. Kortas, and B. Meltz, 1995. Curvilinear coordinates for global ocean models. Los Alamos National Laboratory Report, LA-UR-95-1146, 38 pp.