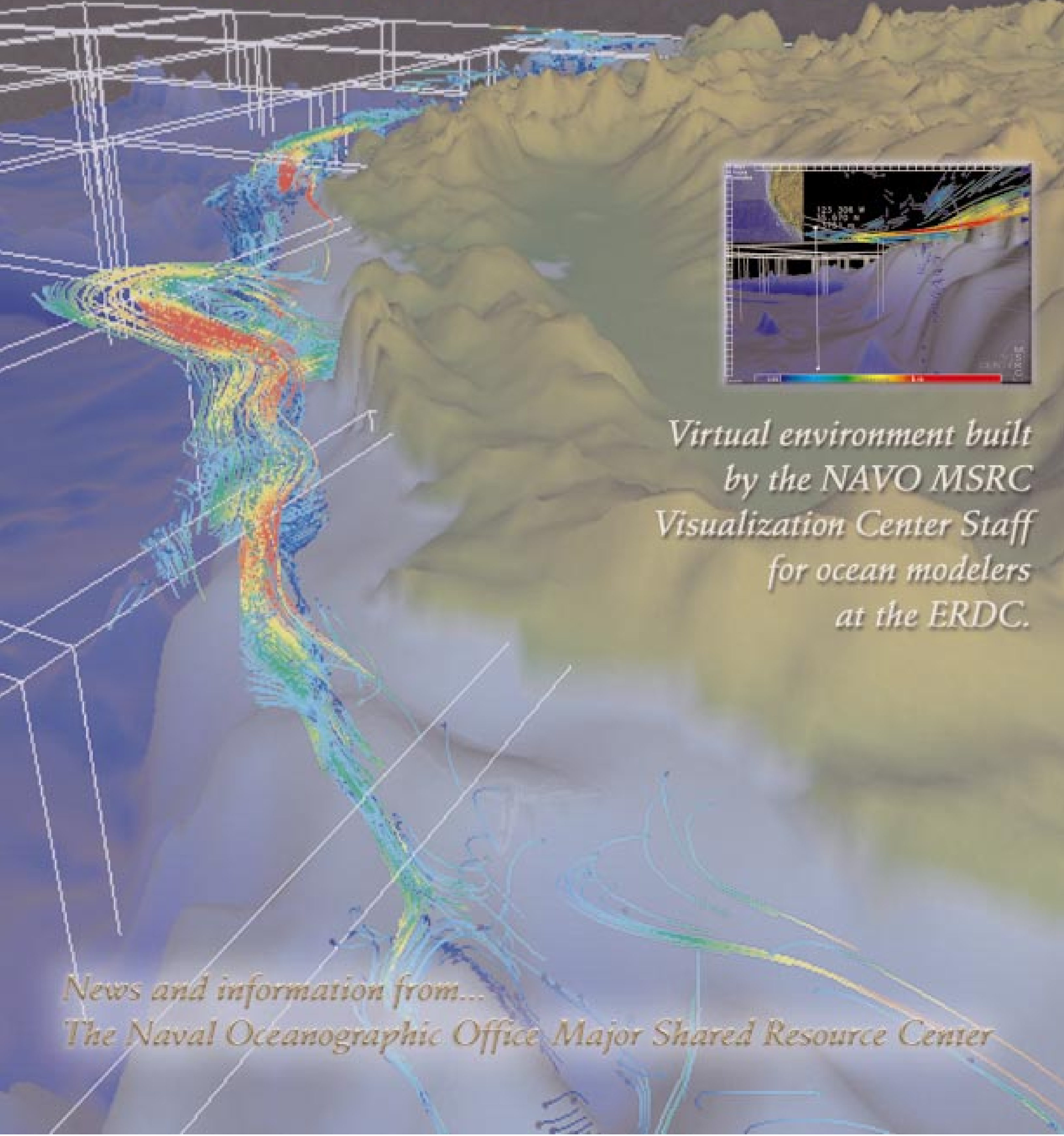
Navigator

NAVO MSRC Spring 2001



Virtual environment built by the NAVO MSRC Visualization Center Staff for ocean modelers at the ERDC.

News and information from...
The Naval Oceanographic Office Major Shared Resource Center

OpenMP Parallelization of a 3-D Finite Element Circulation Model

Dr. Timothy J. Campbell, NAVO MSRC Programming Environment & Training
Dr. Cheryl Ann Blain, Oceanography Division, Naval Research Laboratory

The Naval Oceanographic Office (NAVO) Major Shared Resource Center (MSRC) Programming Environment and Training (PET) program offers Department of Defense (DoD) researchers and engineers the opportunity to work in close collaboration with PET analysts to bring about serial and parallel optimizations to their applications.

One such collaboration occurred in a project that successfully ported an advanced three-dimensional (3-D) finite element (FE) circulation model to shared memory parallel machines.

The goal of this project was to produce, inasmuch as possible, a scalable code that required no change to the user interface and configuration files, while at the same time, to educate the researchers in parallel programming techniques.

OpenMP multithreading directives were chosen to port the model as they can provide a minimally intrusive and incremental method for producing a parallel code.

> "The goal of this project was to produce, inasmuch as possible, a scalable code that required no change to the user interface and configuration files, while at the same time, to educate the researchers in parallel programming techniques."

Physical & Mathematical Model

Parallelization efforts were focused on the Dartmouth College circulation model, QUODDY, which represents the most physically advanced finite element model to date.

This model is a time-marching simulator based on the 3-D hydrodynamic equations subject to the conventional Boussinesq and hydrostatic assumptions. A wave-continuity form of the mass conservation equation, designed to eliminate numerical noise at or below two times the grid spacing, is solved in conjunction with momentum conservation and transport equations for temperature and salinity.

Vertical mixing is represented with a level 2.5 turbulence closure. This turbulence closure scheme accounts for processes occurring over the vertical extent of the water column, such as diffusion, shear production, buoyancy, production, and dissipation. Variable horizontal resolution is provided on unstructured triangular meshes. A general terrain-following vertical coordinate allows smooth resolution of surface and bottom boundary layers.

The QUODDY model is dynamically equivalent to the often used Princeton Ocean Model. The advantage of the current model lies in its finite element formulation that allows for greater flexibility in representing geometric complexity and strong horizontal gradients in either bathymetry and/or velocity.

Parallel Implementation

OpenMP is a parallel programming model for shared memory and distributed shared memory multiprocessors that works with either standard Fortran or C/C++.

OpenMP consists of compiler directives, which take the form of source code comments, that describe the parallelism in the source code. A supporting library of subroutines is also available to applications. The OpenMP specification and related material can be found at the OpenMP web site: http://www.openmp.org.

Online training in OpenMP is part of the NAVO MSRC PET distance learning (http://www.navo.hpc.mil/pet/Video) and links to other online training material can be found at the NAVO PET Parallel Computing Portal (http://www.navo.hpc.mil/Tools/pcomp.html).

In Fortran, OpenMP compiler directives are structured as comments, written as C$OMP or !$OMP. An OpenMP program begins as a single process, called the master thread. When a parallel region, which is preceded by either a parallel or parallel-do construct, is encountered, threads are forked to execute the statements enclosed within the parallel construct.

At the end of the parallel region, the threads synchronize, and only the master thread remains to continue execution of the program. The parallel-do construct is commonly discussed and provides a convenient and incremental way to parallelize computationally intensive loops within a program.

The downside to this approach is that the creation of threads at the beginning and their subsequent destruction at the end of the parallel-do construct can require a large number of cycles. The developer must be sure that the loop being parallelized has enough computational work to make the overhead, due to the OpenMP constructs, worthwhile.

The approach used in this project is in the spirit of the Single Program Multiple Data (SPMD) model which is common in Message Passing Interface (MPI) programming. The parallel/end parallel directives were used to enclose the entire time-stepping portion of the code, including subprogram calls within the parallel execution region. Work decomposition within the parallel region is based on the horizontal mesh.

During execution in the parallel region, the threads remain in existence, and proper data flow is ensured through minimal use of the barrier synchronization construct. Also, code that must be executed in serial is handled by the master thread.

Since the barrier construct can be 30 to 50 percent less expensive than a parallel do, this approach significantly reduces the amount of overhead associated with OpenMP.

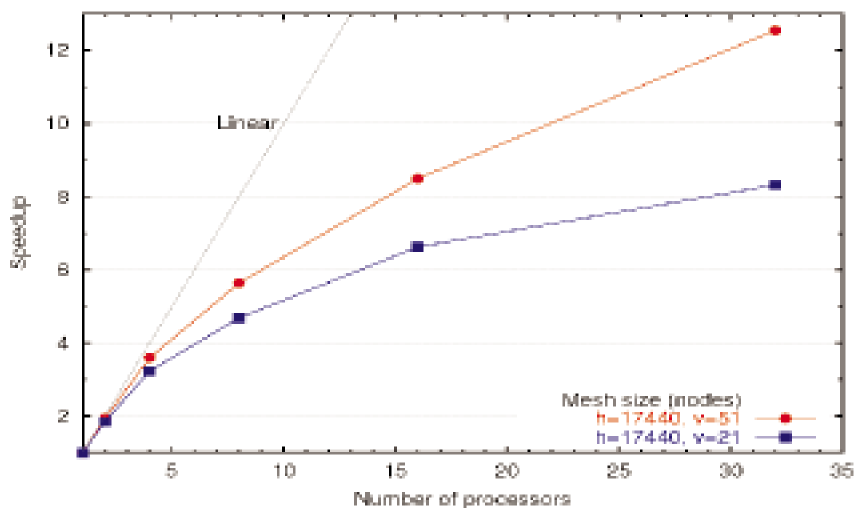


Figure 1. **Speed-up of OpenMP QUODDY4 for the Arabian Gulf (17440 horizontal nodes) on the NAVO Sun E10000. Results for two vertical mesh resolutions are shown: 21 vertical nodes (blue-filled squares) and 51 vertical nodes (red-filled circles).**

The QUODDY software model consists of four sets of programs and includes files for the dimensioning of variables. Parallelization work focuses on three of the program sets that consist of main, core, and fixed routines. When a user applies the QUODDY application to a particular regional model, these three sets of programs remain unmodified. The fourth program set consists of user-built subroutines that are built with a standardized interface.

These routines are used to specify things such as physical forcing, vertical meshing, boundary conditions, and the manner in which results are to be analyzed and written.

By restricting the OpenMP code changes to the main, core, and fixed routines, the user is able to seamlessly apply the parallel QUODDY to different regional models. The user need only compile with the subroutines defined for the regional model of choice.

Verification & Proper Performance

Correctness of the parallel code execution has been verified through direct comparison with the original serial code execution for the Yellow Sea Regional Model (6847 horizontal & 21 vertical nodes).[1] This verification was done using the full "seasonal" mode in which wind is applied and temperature and salinity are transported prognostically.

Since the user-defined output data was of limited precision, verification was done by directly comparing (at full precision) all time-integrated variables. Possible race conditions were "fleshed out" by running with the number of threads greater than the number of processors. An exact match between the serial and parallel execution has been achieved.

Performance measurements were done using the Arabian Gulf regional model (17440 horizontal nodes and either 21 or 51 vertical nodes).[2] The speed-up on p processors is defined as the single processor execution time divided by the time for execution on p processors. Figure 1 shows the speed-up achieved for the OpenMP version of QUODDY on the NAVO MSRC Sun E10000 (64 processors with 64 Gigabyte (GB) shared memory).

Two vertical grid resolutions (21 and 51) were measured. The increase in vertical grid resolution provides more work per horizontal node, thus increasing the scalability of the code. The overall scalability of the OpenMP QUODDY is limited by the remaining serial portions of work (about 5 percent, handled by the master thread) and the synchronization overhead.

Impact & Application

The state-of-the-art QUODDY 3D FE model is a principal tool in the NRL Arabian Gulf project, of which, the primary objective is development of a circulation model for the Arabian Gulf and connecting waters that realistically predicts the complex 3-D circulation and mixing patterns in the region over seasonal, tidal, sub-tidal, and storm event time scales. Mesh resolution is variable, approximately 3 kilometer (km) for depths less then 40 meters (m) and 6 km elsewhere out to 200-m depth in the Gulf of Oman.

Article Continues Page 14...

U.S. West Coast

Researchers at the Naval Postgraduate School, Monterey, California, in conjunction with the NAVO MSRC Visualization Center staff, have undertaken a long-term project to improve the performance of the Multiblock Grid Princeton Ocean Model (MGPOM). The use of multiblock grids in the development of ocean models facilitates domain composition and varying grid resolutions to provide the ability to concentrate grid resolution in the dynamic near-shore regions and save resolution in the less dynamic deep-ocean areas.

Traditional one-block rectangular grids (286 x 286, 4 arc minute resolution), while invaluable, consume large quantities of wall time, slowing research and raising costs. For example, a traditional single block, serial (vector) code takes approximately 1,116 minutes (18.6 hours) of wall time to complete a 10-day simulation. In comparison, a 29-block grid with the same resolution, using MPI-Pthreads MGPOM code, takes only 27 minutes of wall time.

The model produced with this new and improved code provides three-dimensional (3-D) temperature, salinity, and circulation (currents) data as shown in Figures 1 through 5. These images represent screen captures of an analysis environment built for these researchers by the NAVO MSRC Visualization Center staff. This application, and others developed by the Visualization Center staff, provides researchers with a portable analysis environment for ocean model output that supports a variety of functions for both the military and civilian communities.

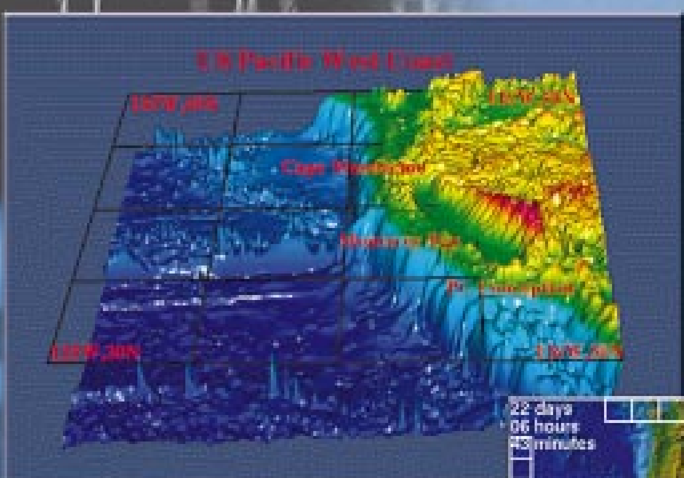


Figure 1. **Plain view of U.S. West Coast with multiblock grid.**

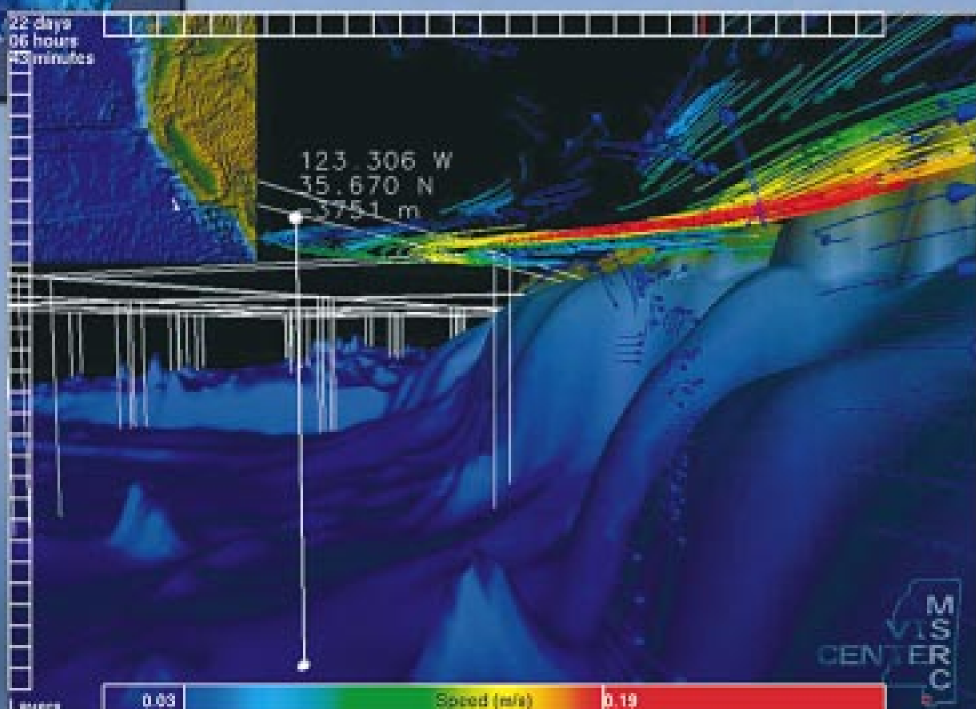


Figure 2. **Perspective view showing the streamlines of surface currents off the U.S. West Coast.**

Ocean Currents

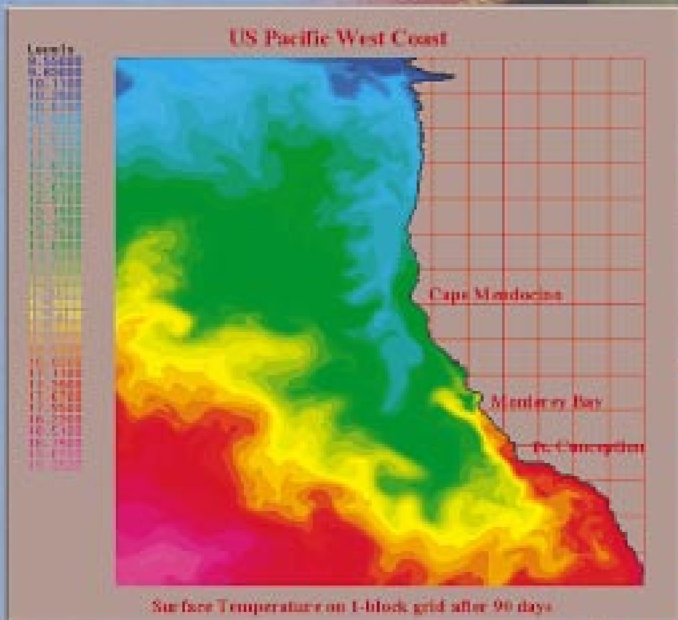


Figure 5. **Surface temperatures after a 90-day simulation within a one-block grid.**

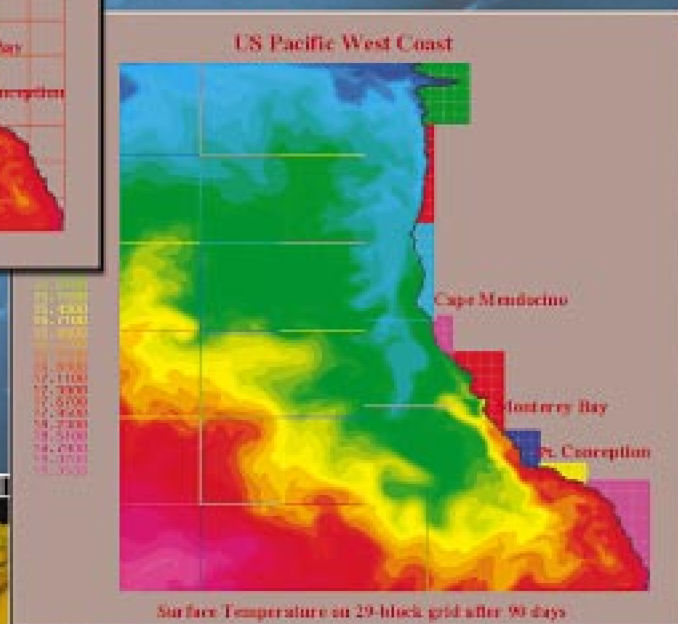


Figure 4. **Surface temperatures after a 90-day simulation within a 29-block grid.**

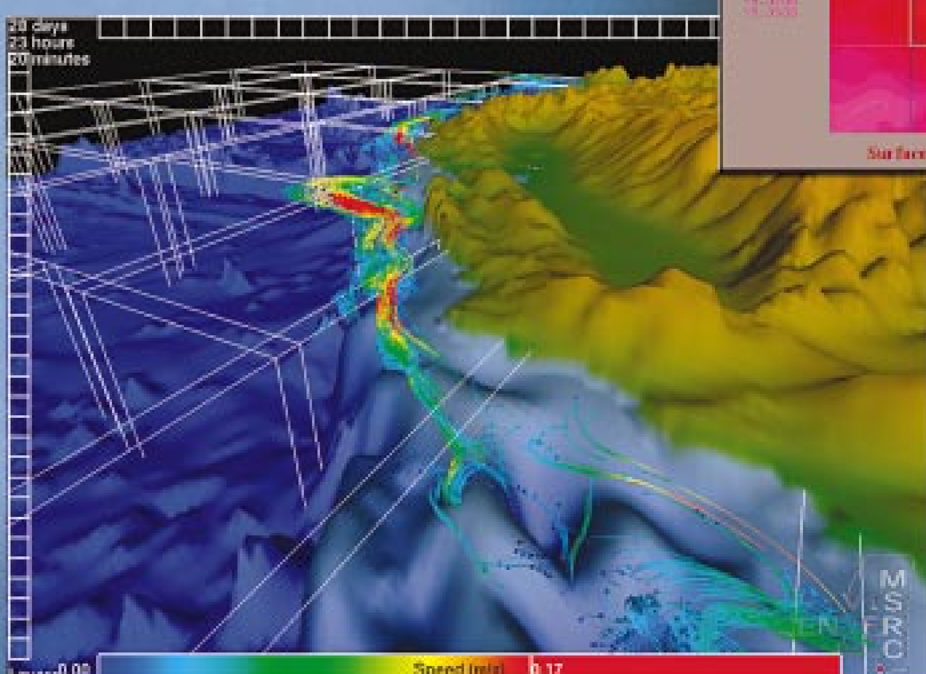


Figure 3. **Top-down view showing the streamlines of surface currents off the U.S. West Coast.**

Continued from Page 11...

The model is designed with modular dynamics in which certain mechanisms, such as heat flux, wind forcing, stratification, tides, or river inflow can be independently included or excluded from model equations. This modularity is used to examine the contributions of each component to the overall circulation dynamics.

Simulations are forced by seasonal hydrography, seasonal winds, and tides. For each month, the initial temperature and salinity fields prognostically evolve subject to tidal rectification and a constant wind stress.

The summer mean circulation is primarily driven by the baroclinic pressure gradient. Fresh water entering the Arabian Gulf from the Gulf of Oman at the surface, coupled with strong evaporation in the north, creates a cyclonic circulation gyre that runs the length of the basin. The northwesterly wind strengthens southward flow along the western edge of the gyre. A westward component of the wind in the southern Arabian Gulf pushes water across the very shallow shelf of the United Arab Emirates (UAE) coast and out through the Strait of Hormuz (Figure 2a).

During winter, the strong northwest winds (3 times the magnitudes in summer) set-up southeastward flowing coastal currents in the northern Gulf along each shoreline. The winds also impede penetration of the freshwater into the Gulf and greatly reduce the strength of the counter-clockwise (CCW) circulation along the axis of the basin (Figure 2b).

In fact, the winds push the circulation gyre to the south and toward the center of the Gulf. Since there is no west-

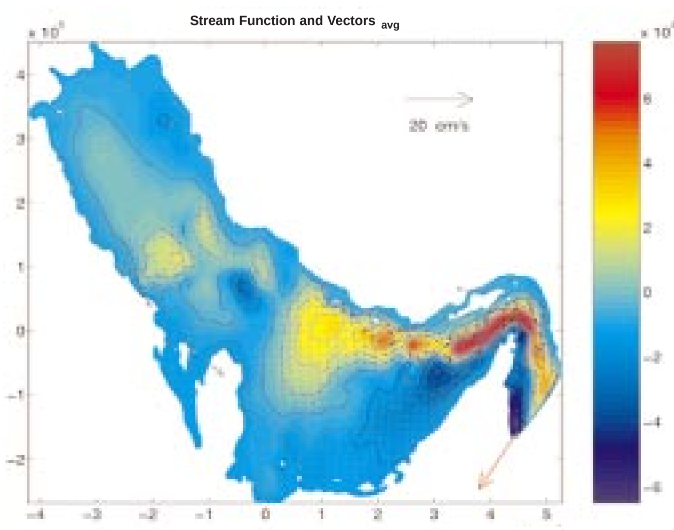


Figure 2a. **Simulated summer seasonal circulation in the Arabian Gulf. Stream function (color) and depth-averaged currents (vectors).**

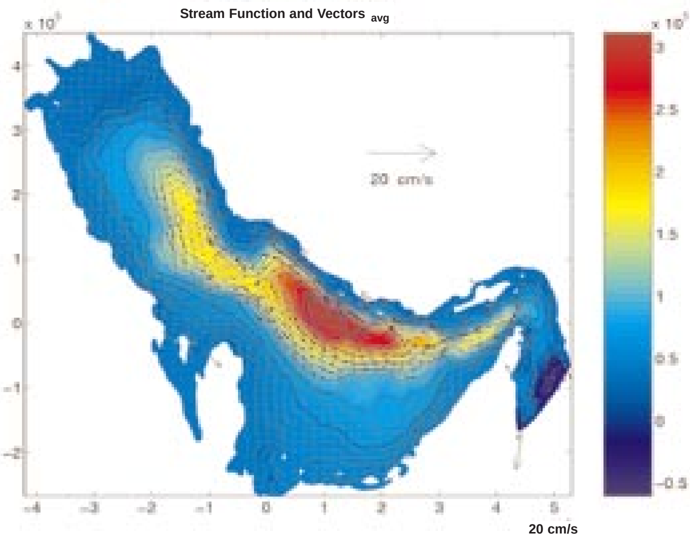


Figure 2b. **Simulated winter seasonal circulation in the Arabian Gulf. Stream function (color) and depth-averaged currents (vectors).**

erly component to the wind in winter, the circulation on the shallow southern shelf is quite complex and varied from that seen in summer.

The speed-up achieved by the OpenMP version of QUODDY is immediately useful to the Arabian Gulf and other planned modeling work. Prior to porting QUODDY model, it would not execute properly on the MSRC resources, thus restricting the researchers to perform simulations only on their workstations.

Performing 10-model-day seasonal simulation experiments required up to several days of execution with limited vertical grid resolution. Now, on 8 processors of the NAVO MSRC Sun E10000, researchers can perform the same 10-model-day seasonal simulation with increased vertical grid resolution in just over 6 hours (with 71 percent parallel efficiency).

The reduced turnaround time will greatly accelerate the model development process. Additionally, because this was a collaborative effort, the researchers are now familiar with the OpenMP code changes and are able to modify and improve the parallel code.

References

1. C. E. Naimie, C. A. Blain, and D. R. Lynch, 2000. Seasonal mean circulation in the Yellow Sea—A model-generated climatology, Continental Shelf Research, in press.

2. Blain, C. A., 2000. Modeling three-dimensional, thermohaline-driven circulation in the Arabian Gulf, in Estuarine and Coastal Modeling, Proceedings of the Sixth International Conference, M. L. Spaulding and H. L. Butler, eds., American Society of Civil Engineers, pp. 74-93.