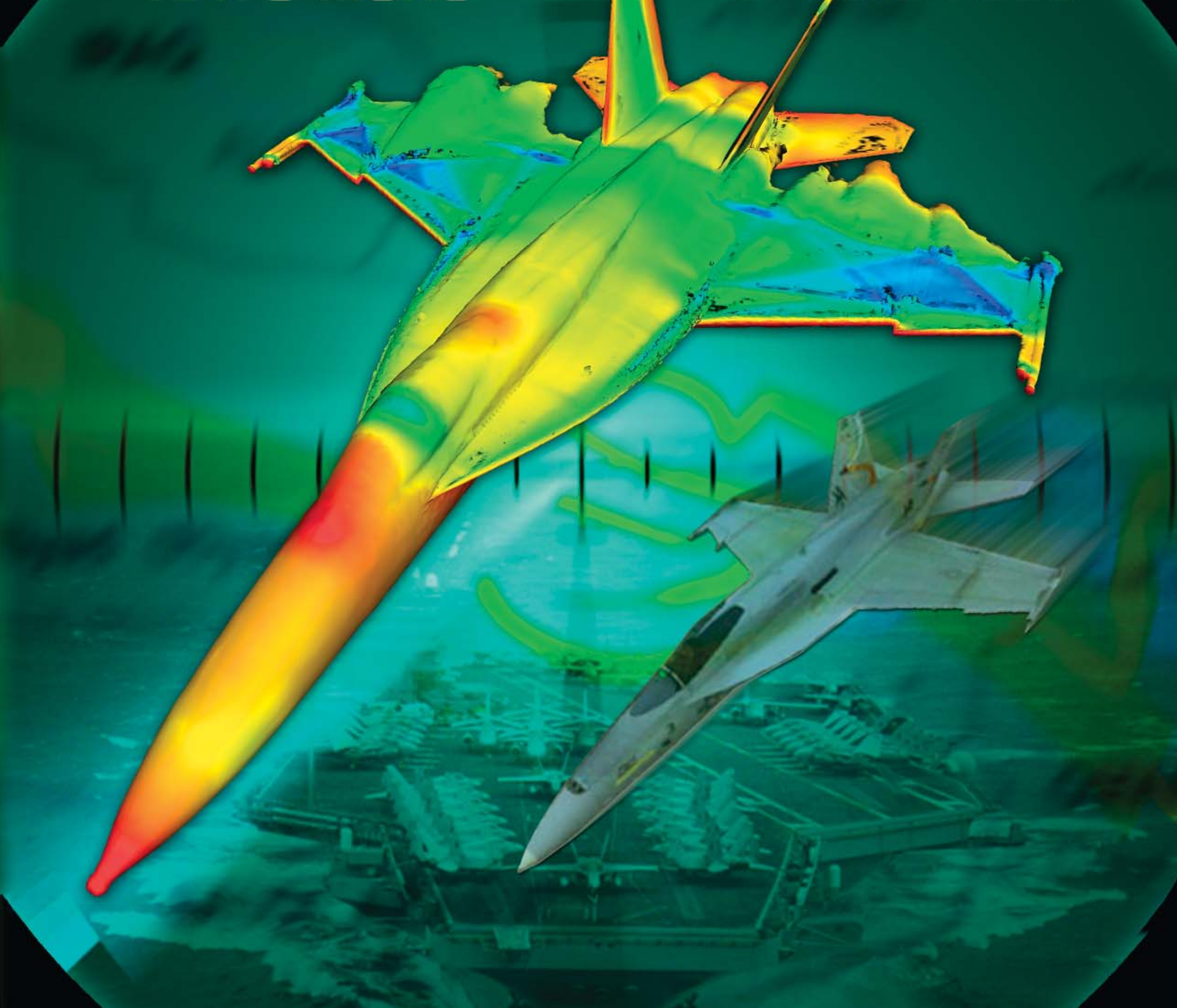
Navigator

NAVO MSRC SPRING 2005



News and information from...

The Naval Oceanographic Office Major Shared Resource Center

Since the last issue of The Navigator, there has been substantial improvement to the computational capability at the NAVO MSRC. As the result of the High Performance Computing (HPC) Modernization Program (HPCMP) hardware technology insertion process (TI-04), the MSRC hosts two new IBM POWER4+ systems (dubbed KRAKEN and ROMULUS) with a total of approximately 3,500 processors.

These new systems consist of 8-processor compute nodes, each with 16 Gigabytes (GB) of memory, all interconnected with the IBM Federation switch. With this additional computing capacity, the MSRC supports a primary DoD HPCMP goal of fielding the largest and most capable HPC environments in the world, serving more than 4,000 users across the DoD services and agencies.

More exciting is the service that the larger of the two news systems, KRAKEN, allows us to provide to our users. With KRAKEN's processing power, we hosted the first-ever HPCMP Capability Applications Project (CAP) program, during which selected computational projects tested their application codes on a substantial portion of that system to solve large, meaningful problems in a relatively short time.

We were very pleased that a majority of the Phase I CAP projects were selected to run on KRAKEN and that three of those projects were chosen to advance to the CAP Phase II. Two of this issue's articles focus on NAVO MSRC Phase II CAP projects—Early Atmospheric Turbulence Simulation Experiences in the HPCMO Capability Applications Project (CAP) (page 9)

NAVO MSRC and CAP–Saving Time, Delivering Results

and Free-to-Roll F/A-18E Capability Applications Project (page 14).

Please take a moment to see what the enhanced NAVO MSRC capabilities can help you, the user, achieve. And in the end, that's our purpose—to serve you, the user, and ensure that you have the tools and facilities needed to accomplish your mission. As always, we invite you to contact us and let us know how we can better serve you.

About the Cover:

Pictured are Navy Joint Strike Fighter (JSF) F/A18E Jets, future generations of which will benefit from the research described in Free-to-Roll F/A-18E Capability Applications Project (Page 14), as well as other projects underway at the NAVO MSRC.

The Naval Oceanographic Office (NAVO) Major Shared Resource Center (MSRC): Delivering Science to the Warfighter

The NAVO MSRC provides Department of Defense (DoD) scientists and engineers with high performance computing (HPC) resources, including leading edge computational systems, large-scale data storage and archiving, scientific visualization resources and training, and expertise in specific computational technology areas (CTAs). These CTAs include Computational Fluid Dynamics (CFD), Climate/Weather/Ocean Modeling and Simulation (CWO), Environmental Quality Modeling and Simulation (EQM), Computational Electromagnetics and Acoustics (CEA), and Signal/Image Processing (SIP).

NAVO MSRC
Code N7
1002 Balch Boulevard
Stennis Space Center, MS 39522
1-800-993-7677 or
msrchelp@navo.hpc.mil

NAVO MSRC Navigator
www.navo.hpc.mil/Navigator

NAVO MSRC Navigator is a biannual technical publication designed to inform users of the news, events, people, accomplishments, and activities of the Center. For a free subscription or to make address changes, contact NAVO MSRC at the above address.

EDITOR:
Gioia Furness Petro, petrogio@navo.hpc.mil

DESIGNERS:
Cynthia Millaudon, cynmill@navo.hpc.mil
Kerry Townson, ktownson@navo.hpc.mil
Lynn Yott, lynn@navo.hpc.mil

Contents


KRAKEN
IBM
A "high-end scalable parallel" system using IBM processors and a dedicated high-speed network.
The current implementation: 2944 processors 5.8 TB memory
High Performance Computing Major Shared Resource Center
NAVO MSRC
Top 10 world ranking among supercomputing centers.
Current computing capability of 30+ trillion operations per second.

DoD HPCMP's Impact on the Development of the Nation's Next Generation Mesoscale Numerical Weather Prediction Model, the Weather Research and Forecasting (WRF) System

Jerry W. Wegiel, Air Force Weather Agency

The Air Force Weather Agency (AFWA), one of Air Force Weather's strategic centers, delivers the highest quality tailored information, products, and services to the nation's combat forces. A key component of this support is the Air Force Weather Weapon System's (AFWWS) mesoscale numerical weather prediction (NWP) model.

Since all aspects of military operations are affected to some degree by the weather, it is essential the NWP be state-of-the-science and optimized for maximum accuracy. While the current AFWWS mesoscale NWP model (the Pennsylvania State University (PSU)/National Center for Atmospheric Research (NCAR) Mesoscale Model 5 (MM5)) has performed admirably in its role as the AFWWS's mesoscale numerical model since 1997, it no longer meets many warfighter requirements.

As a result, the AFWA has partnered with the numerical weather prediction community to build the nation's next generation mesoscale NWP model, the Weather Research and Forecasting (WRF) system. Addressing these deficiencies will allow AFWA to better anticipate and exploit the weather for battle anytime, anywhere—from the mud to the sun.

The U.S. inter-organizational modeling initiative, or WRF model, has a three-pronged objective, which is to develop (a) the next generation mesoscale NWP modeling system for research and operations; (b) a common modeling infrastructure that facilitates operational NWP collaboration and scientific interoperability, and that accelerates the transfer of new science from research into operations; and (c) a repeatable process that continuously infuses innovations and capabilities into the community mesoscale NWP modeling system.

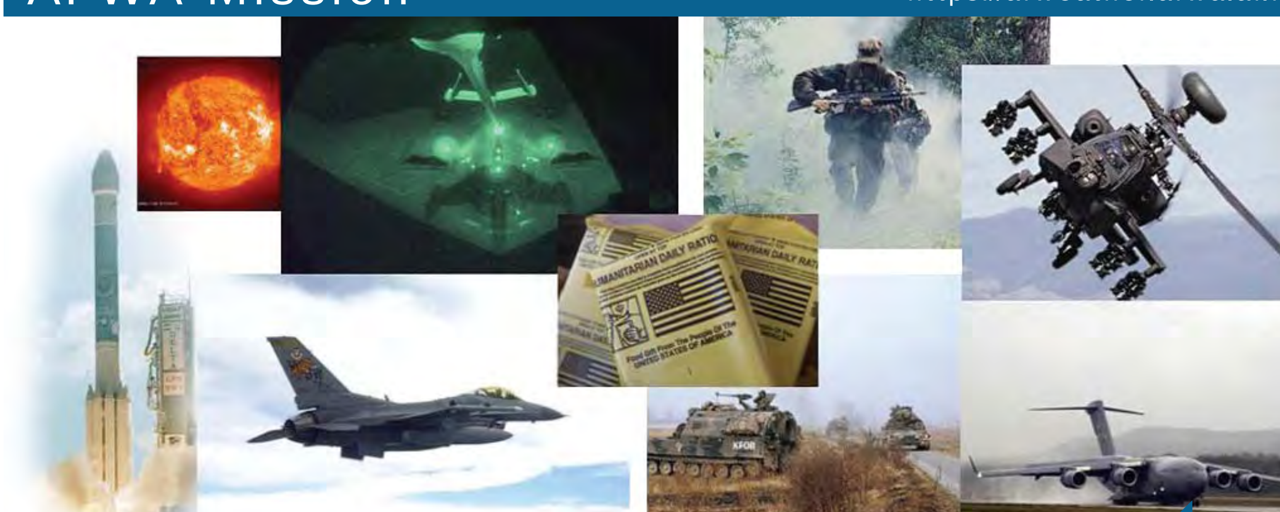
AFWA, a principal partner of this national effort, has been able to leverage the vast array of resources available only to Department of Defense (DoD) entities—the resources available through the DoD's High Performance Computing Modernization Program (HPCMP).

In the past four years, AFWA has leveraged every component of the HPCMP in the development of the

Continued Next Page...

AFWA Mission

https://afweather.afwa.af.mil/



Maximize our nation's aerospace and ground combat effectiveness by providing accurate, relevant and timely air and space weather information to Department of Defense, coalition, and national users, and by providing standardized training and equipment to Air Force Weather.

WRF modeling system. The AFWA link to the first HPCMP component, the Defense Research and Engineering Network (DREN), allowed researchers to use the immense computational and storage resources of the second component of the HPCMP, the Major Shared Resource Centers (MSRCs).

One such facility is the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC). Of particular value to this effort, the NAVO MSRC manages one of the largest IBM POWER4 platforms (the latest and most advanced processor developed for supercomputing by IBM) in the world and the architecture on which the AFWA common modeling infrastructure is based.

This platform served as the WRF community's proxy WRF Development Test Bed Center (DTC) in FY03-04, which allowed for the execution and evaluation of deterministic and ensemble forecast systems. The results of this effort enabled the National Centers for Environmental Prediction (NCEP) (at 1200 Coordinated Universal Time (UTC) on 21 September 2004) to transition WRF into operations.

The new NCEP WRF-based modeling system represents the first U.S. operational implementation of WRF and is the first step on the way to an implementation of a WRF-based NCEP High Resolution Window mesoscale ensemble scheduled for implementation in spring 2006.

The High Performance Computing Modernization Program Office (HPCMPO) granted a FY04 Distributed Center (DC) award to AFWA and the Fleet Numerical Meteorology and Oceanography Center (FLENUMMETOCCEN) in November 2003. The objective of the dedicated DC project, the Joint Operational Test Bed for the WRF Modeling Framework, is to field a platform to conduct operational tests of WRF.

The AFWA will become the first operational center in the world to implement the full end-to-end WRF system in 2005, while FLENUMMETOCCEN is slated to implement WRF toward the latter part of the decade.

These operational tests, in order to arrive at WRF configurations that best meet unique Navy and Air Force mesoscale NWP requirements, will include the multiple configurations of the model made possible by its interchangeable dynamic cores and physics packages.

In addition, operationally capable mesoscale ensemble runs will be tested using varying WRF configurations, perturbed initial conditions, and differing lateral boundary conditions.

Finally, the grid computing concepts and tools applied to the stringent and unique requirements of NWP, with the WRF Joint Operational Test Bed system, will be prototyped and tested.

This WRF system is physically split between the FLENUMMETOCCEN and AFWA sites, yet linked to form a distributed "weather grid" computing platform. It is hoped the WRF Joint Operational Test Bed, and its grid-computing capability, will be used to enhance collaboration between Navy and Air Force weather research and development and operations activities.

The overarching goal of these efforts is to transition the science and technology resulting from work performed on the WRF Joint Operational Test Bed rapidly into improved high-resolution operational weather prediction capabilities at both AFWA and FLENUMMETOCCEN.

Another HPCMP resource used by AFWA for WRF model development has been the HPCMP Software Applications Support programs, more specifically, the Programming Environment and Training (PET) and the Common High-performance

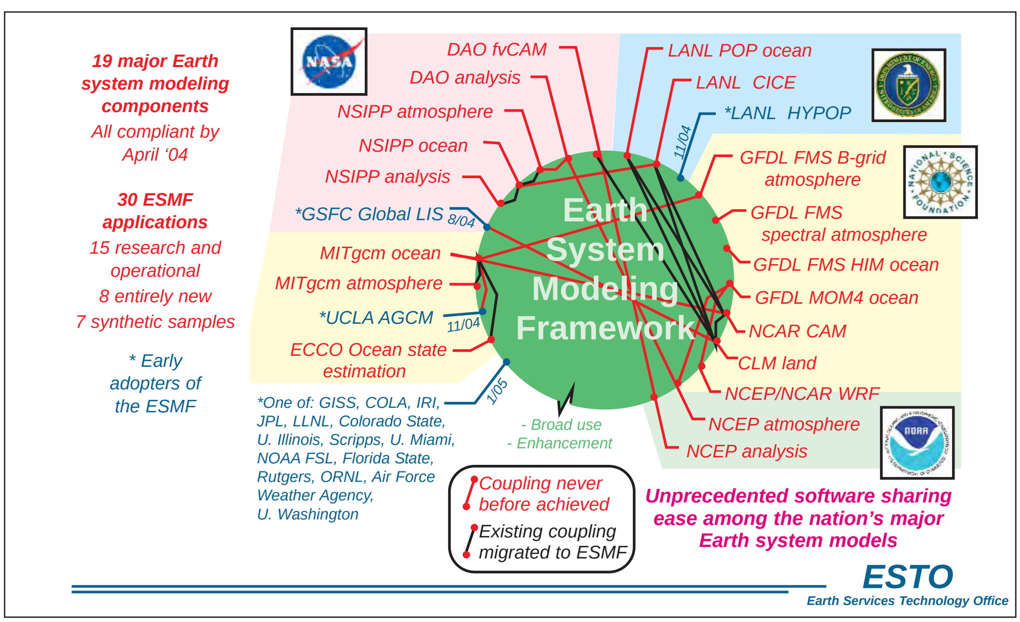


Figure 1. The BEI will become the primary means to couple earth system components within DoD. Stakeholders include the U.S. Navy, U.S. Air Force, U.S. Army, National Aeronautics and Space Administration, Department of Energy, Department of Commerce, and the National Science Foundation.

Software Support Initiative (CHSSI). CHSSI (via a 3-year, $1.5M project: "Weather Research and Forecast Model Development") was instrumental in the acceleration and enhancement of the development of the WRF model.

Thanks to CHSSI, WRF developers were able to deliver a robust, highly scalable, portable, and modular mesoscale NWP model suitable for both research and operations to the WRF community. Furthermore, these resources also allowed for the development of an advanced 3-Dimensional VARiational data assimilation system (3DVAR). Data assimilation experts at NCAR were so impressed with this system that they adapted it for use with their MM5 mesoscale modeling system.

As a result, AFWA was the first DoD NWP modeling center to implement a 3DVAR data assimilation system into operations on 26 September 2002.

The goal of developing this advanced data assimilation system was twofold: first, allow for full-spectrum utilization of this nation's multi-billion dollar remote sensing investment assets and, secondly, prepare AFWA for the National Polar-orbiting Operational Environmental Satellite System (NPOESS) era.

In 2004, the approach and scope of the Software Applications Support component of the HPCMP was modified to establish institutes to forge "...a critical mass of experts keenly focused on using computational science and high performance computing to accelerate solving the Department's highest priority challenges. With cross- Service and Agency teaming and multi-disciplinary approaches, the institutes have a strong potential to transform the DoD's science and technology and test and evaluation communities and to make the important advances in research, development, testing, and evaluation."[1]

In FY05, AFWA and its partners (the Naval Research Laboratory (NRL) Stennis Space Center (SSC), University Corporation for Atmospheric Research, and the U.S. Army Engineer Research and Development Center) were granted a six-year, $11.5 million award to establish a Battlespace Environments Institute (BEI). (See Figure 1.)

The BEI will migrate existing DoD Climate-Weather-Ocean Modeling and Simulation (CWO), Environmental Quality Modeling and Simulation, and space weather applications (including WRF) to the Earth System Modeling Framework (ESMF), plus assist in transitioning non-DoD ESMF applications to the DoD. The BEI will also augment ESMF with capabilities needed for the DoD battlespace environment.

Like CHSSI, PET is an HPCMP software applications support program and the final component of the HPCMP used by AFWA. In addition to awarding an on-site CWO Applications Specialist to AFWA in FY05 to assist in the enhancement of CWO applications on High Performance Computing (HPC) systems such as WRF, PET also contributed greatly to the development of the WRF system itself.

In FY03, PET funded the Infrastructure Development for Regional Coupled Modeling Environments project. This two-year project is significant because it allowed AFWA, NCAR, and the

Continued Next Page...

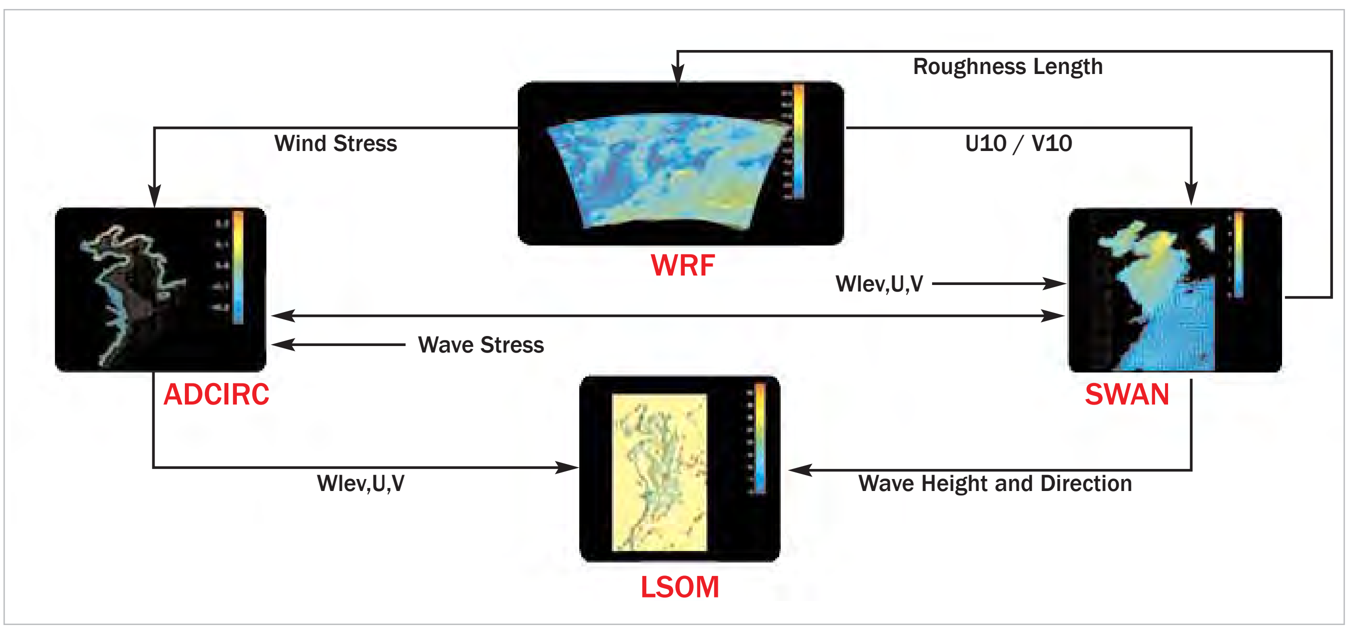


Figure 2. The project between AFWA, NCAR, Argonne National Laboratory, and University of Southern Mississippi delivered a grid-enabled, concurrent, multi-executable, parallel, coupling capability through a common, model independent interface to NRL Stennis for use in Research and/or Operations.

other principal partners in the WRF development effort to break down some of the technical and requirements-based barriers impeding other centers from joining the national development effort.

In its first year, this project developed and demonstrated a flexible, reusable software infrastructure for high-resolution regional coupled modeling systems that abstracts the details and mechanics of inter-model coupling behind an Application Program Interface (API) that also serves as the API to Input/Output (I/O) and data format functionality.

The focus in FY04 (2nd year) was to demonstrate the capability on a real-world problem of interest to a DoD operational forecast center: a severe weather event and ferry boat accident that took place on 25 November 1999 in the Yellow Sea.

This demonstration involved coupling the WRF atmospheric model with the ADvanced CIRCulation (ADCIRC) ocean model, the Simulating WAves Nearshore (SWAN) wave model, and the Littoral Sediment Optical Model (LSOM) in the configuration illustrated in Figure 2.

In a nutshell, WRF researchers delivered a grid-enabled, concurrent, multi-executable, parallel, coupling capability through a common, model independent interface to NRL SSC for use in research and/or operations in two years for $400,000—a phenomenal return on investment.

In summary, one cannot overstate the impact the DoD High Performance Computing Modernization Program has had in the development of the WRF modeling system.

As far as the community of WRF developers is concerned, the HPCMP fulfilled its mission and goals by significantly reducing research, development, testing, and evaluation costs and promoting an environment conducive for inter-agency/service strategic partnering.

The HPCMP has been the single most important contributor to the national WRF effort. The incredible level of HPCMP support toward WRF single-handedly enabled the community of developers to succeed in their quest to deliver to the nation the next generation mesoscale NWP modeling system.

HPCMP WRF Contributions

Contribution	Impact	Result
CHSSI CWO-06 (FY00-03)	$1.5M	Beta WRF and 3DVAR
WRF DTC (FY03-04)	400,000 high priority hours	RDT&E of WRF2.0
PET-CWO (FY03-04)	$400,000	A grid-enabled, concurrent, multi-executable, parallel, coupling capability through a common, model independent interface to NRL Stennis for use in research and/or operations.
PET-CWO on-site support (FY04>)	2 FTE or ~ $400,000 per year	Various DoD HPC capabilities delivered.
Dedicated Distributed Center (FY04-06)	$4.2M	DoD Operational Testbed Center
Battlespace Environments Institute (FY05-10)	11.5M	Negates existing DoD, CWO, EQM, and space weather applications to the ESMF and assists in transitioning non-DOD ESMF applications to DoD.

Acknowledgements

The Air Force Weather Agency would like to thank the High Performance Computing Modernization Program (HPCMP), National Center for Atmospheric Research (NCAR), National Oceanic and Atmospheric Administration (NOAA)/Forecast Systems Laboratory (FSL) and NOAA/National Centers for Environment Prediction (NCEP), Fleet Numerical Meteorology and Oceanography Center (FLENUMMETOCCEN), Naval Research Laboratory-Stennis, and the University Corporation for Atmospheric Research for their contributions.

References

1. Mark, A., http://www.hpcmo.hpc.mil, 2004.

Early Atmospheric Turbulence Simulation Experiences in the HPCMO Capability Applications Project (CAP)

Joseph Werne, Colorado Research Associates Division, NorthWest Research Associates, Inc., Air Force Research Laboratory

The High Performance Computing Modernization Office (HPCMO), in response to increasing demands from users for more computing resources and the recent integration of High Performance Computing (HPC) platforms, recently instituted the Capability Applications Project (CAP). CAP is designed to quantify the degree to which important application codes scale to thousands of processors and to enable new science and technology by applying these codes in dedicated, high-end capability environments.

Under CAP, the author and associates were able to use the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) IBM Power4+ (KRAKEN), a 2944-processor P655 IBM Power4+ comprised of 368 eight-processor nodes and nearly six Terabytes (TB) of Random Access Memory (RAM), to run a series of simulations, primarily for the U.S. Air Force.

METHODOLOGY

In generating these simulations, a Three Dimensional (3D) pseudo-spectral solver (which simulates the incompressible Navier-Stokes equations in the Boussinesq approximation) was used.[1]

Time integration is via a third-order Runge-Kutta algorithm with storage requirements typical of most second-order schemes. Spatial discretization is achieved through Fourier expansions of the field variables; hence, the spatial resolution is formally *N*th order, where *N* is the number of Fourier modes used in a given spatial direction.

More than 80 percent of the code's operation count is consumed by 3D Fast Fourier Transforms (FFTs), which are used to move between physical and spectral space. Comparatively less time (about five percent) is spent performing the communication-based transpose needed for each 3D FFT. Efficient one-sided communication is accomplished via David Klepacki's Shared Memory (SHMEM) library.*

CAP PARALLEL PERFORMANCE TESTS

Figure 1 shows results from parallel performance tests conducted on KRAKEN. The ratio *C* of the Central Processing Unit (CPU) hours to the total number of operations is plotted versus the number of processors (NCPU). *C* is given by *C=(walltime) NCPU/(NtNg logNg)*, where Nt is the number of time steps, and *Ng = NxNyNz* is the total grid size. The *logNg* factor appears because FFTs dominate the operation count, and the cost of a 3D FFT is proportional to *Ng logNg*.

Continued Next Page...

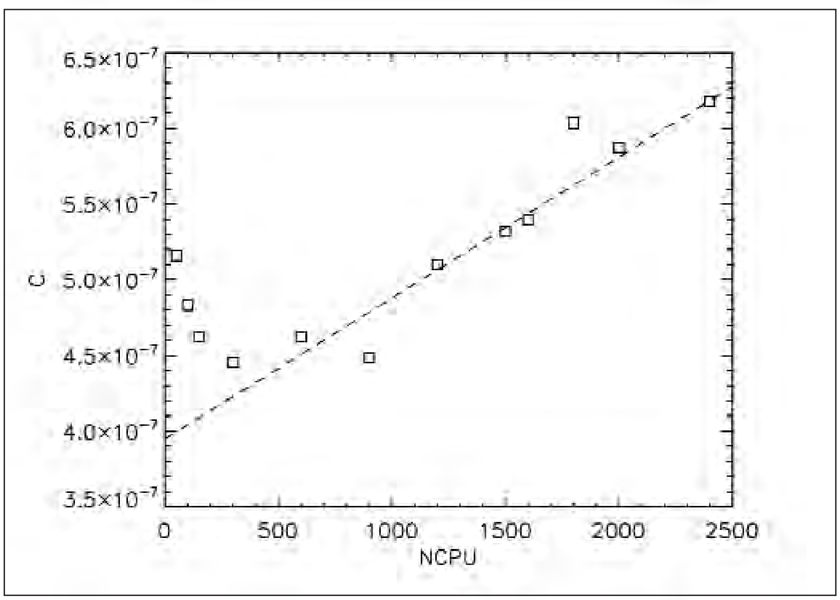


Figure 1. Code parallelization efficiency on KRAKEN. *C* is proportional to the total CPU time divided by the operation count. Tests are conducted with a fixed problem size per processor of 280 MB. The fit (dashed line) indicates $C=3.9\ 10^{-7}+9.31\ 0^{-11}NCPU$, which indicates a parallelization efficiency of 0.99976 according to Amdahl's Law.

The dashed line in the plot is a linear fit to the data for large NCPU: $C_{fit} \approx 3.9\mathrm{x}10^{-7}+9.3\mathrm{x}10^{-11}NCPU$. The variation in the fit is because of the different radix FFTs used to maintain a fixed problem size per processor during testing. The FFT algorithm employed permits efficient FFTs of size $2^n3^m4^p5^q$, and the performance of the different radix routines varies for different memory configurations and problem shapes.

Using the fit and Amdahl's Law, [2-4] a parallelization efficiency of 0.99976 is calculated; i.e., the code is 99.976 percent parallel and only 0.024 percent serial. This parallelization efficiency is so high that before conducting testing on KRAKEN the parallel performance limits could not be easily evaluated; i.e., the asymptotic scalability is not revealed until $NCPU>1000$. (See Figure 1.)

During testing, 280 Megabytes (MB) per processor, or about 30 percent of each processor's then 1 Gigabyte (GB) Random Access Memory (RAM) capacity, was used.**

This size is sufficiently larger than the KRAKEN 0.7 MB L2 cache and each processor's portion of the shared 128 MB L3 cache, so anomalous speedup is avoided. At the same time, it is sufficiently smaller than the per-processor RAM limit, so tests can be carried out three times faster than if the KRAKEN memory had been filled.

In addition, the effect of running with a larger per-processor problem size by doubling and tripling the grid size for a case with $NCPU = 600$ was also tested. C differed from the data in Figure 1 by only 0.62 percent (3.7 percent) when the grid was doubled (tripled), indicating that the choice of 280 MB per processor is appropriate for testing.

The heterogeneous architecture of the IBM SP platforms introduces challenges for code optimization. Through experimentation with different combinations of code mappings onto the KRAKEN eight-way-node structure, it was learned that the housekeeping operations carried out by the IBM on each node consume resources and hamper optimal performance.

Because of this, the code runs best when only seven processors per node are employed. The IBM compiler uses the idle processor to conduct node-overhead operations, but this has the unpleasant side effect of leaving 12.5 percent (i.e., one out of every eight) of the processors idle. To minimize the impact of this effect, it is anticipated that a design incorporating more processors per node would be better.

Comparing Figure 1 with similar data collected on the U.S. Army Engineer Research and Development Center (ERDC) now-defunct Cray T3E[5] indicates that the simulation code runs 5.2 times faster on KRAKEN than on the 600 Megahertz (MHz) T3E (which was the processor available when Cray timing tests were conducted in 2000). The Cray DEC alpha chip performed two floating-point operations (one multiply and one add) simultaneously, giving it a theoretical peak speed of 1.2 Giga Floating Point Operations Per Second (GFLOPs)/CPU.

KRAKEN processors, in contrast, run at a clock speed of 1.7 Gigahertz (GHz), but each has two floating-point units capable of delivering a floating-point multiply-add in the same clock cycle. Hence the theoretical peak speed of the KRAKEN system is 6.8 GFLOPs, and the largest potential speedup from the T3E is 5.7. More detailed profiling on both systems is

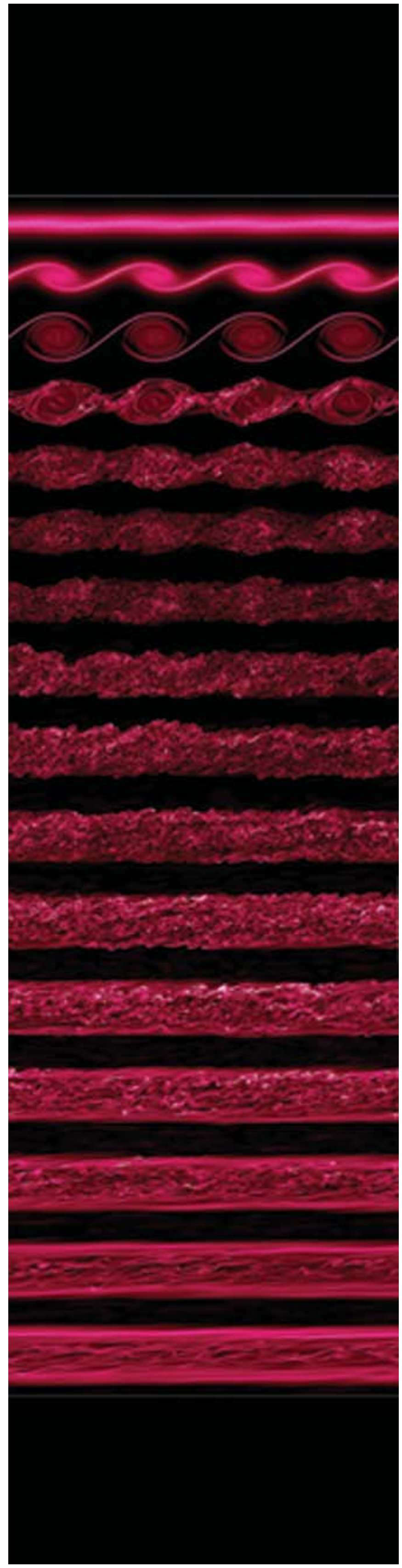


Figure 2. Sequence of images showing vorticity viewed from the side from a wind-shear event encompassing four Kelvin-Helmholtz (KH) billows. Each image is separated by 20 advection time units. The entire event spanned roughly 300 advection units.

required to understand the 10 percent reduction of the 5.2 speedup experienced compared to the theoretical speedup limit of 5.7.

CAP SIMULATIONS OF ATMOSPHERIC WIND SHEAR

With the performance of the code demonstrated, the next step was to perform production runs to simulate atmospheric turbulence for the Air Force Research Laboratory (AFRL).

These simulations are important because they overcome the finite-domain restrictions that hampered Airborne Laser (ABL) challenge simulations analyses attempted on the 600 MHz T3E machines.

These new CAP turbulence simulations are 24 times larger than those computed previously and provide much better statistics for assessing the impact of atmospheric turbulence on laser propagation. They also allow for better evaluation of SubGrid-Scale (SGS) approaches to stratified turbulence models and for development of a Bayesian Hierarchical Model for use as an SGS approach to real-time turbulence forecasting.

Figure 2 shows results from the CAP simulations of atmospheric wind shear. The simulations describe the evolution of four Kelvin-Helmholtz (KH) billows and employ a domain that is 4x2x2 in units of the most unstable KH wavelength. The flow was initiated with an unstable velocity shear in a linearly varying background density profile.

The initial vortex sheet rolls up into four well-defined, large-scale vortices. Many smaller-diameter vortices grow

Continued Next Page...

Figure 3. View from above of the mid-plane vorticity for our CAP wind-shear simulations. The domain is 24 times larger than achieved previously. The domain used for the simulation before the CAP program is shown as shaded in the lower left corner.

from and wrap around the edges of the billows in the form of a secondary instability. The small vortex tubes interact with one another and trigger the cascade to small-scale turbulent motion. At the highest resolution, the solution required 3000x1500x1500 spectral modes and was run on 1500 processors. Each 24-hour run consumed more than 40,000 CPU hours and generated more than four TB of data. When finished, the solution shown in Figure 2 required over 650,000 CPU hours to complete and generated over 80 TB of data for analysis.

Figure 3 shows a view of the mid-plane in the simulation viewed from above. The much larger domain possible via CAP is made apparent by the shaded lower left corner, which signifies the smaller domain used previously. Even though this solution has only just been completed, and therefore there has been little time to analyze the results, significant differences from previous solutions are apparent.

First, the larger domain affords more degrees of freedom and opportunities for larger-scale organization of the flow. At intermediate and late times this is manifest in the form of high- and low-speed streaks, which can now be quantified. This was not possible before because streak widths at late times are comparable to, or larger than, the smaller domains used previously.

Second, when the stratification is increased, significantly different dynamics than have been reported in the literature (e.g., billows which collapse immediately after forming and enhanced lateral spreading due to vortex pairing) are noted. From the CAP solutions it is apparent that the likelihood of vortex-pairing events is sufficiently small and that the original domain was unlikely to realize even one such event.

Finally, the spectrum of radiated gravity waves from the turbulent shear layer is much fuller than that possible in the smaller domain. But these immediate observations only scratch the surface of these rich stratified turbulence solutions. In the coming months we look forward to completing further analyses, and we are eager to use them to help develop real-time optical-turbulence-forecast models for the Air Force.

CONCLUSION

In conclusion, the CAP program has been enormously beneficial to our research program. It has allowed us to obtain solutions that are practically impossible through normal MSRC operations because it provided access to much larger numbers of processors than are permitted with standard queuing policies. We hope the CAP program becomes a permanent addition to the HPCMP, and we will certainly continue to participate if it does.

Acknowledgements

This work benefited significantly from help received from others, including: David Klepacki, who provided his IBM P4+ SHMEM library; John Cazes, who helped discover a workaround in a bug in the shmem64.c library routines and who educated us on KRAKEN hardware; John Skinner, who provided documentation for the KRAKEN software and who saw to it that my 2400-processor job made it through the block queue; and Hank Kuehn, who adjusted KRAKEN system parameters to increase the command-line memory so that data from these large jobs could be easily managed by our scripts. It also helped that the Uniform Command Line Interface (UCLI) was ported to KRAKEN and waiting for us when I logged on. I'd also like to acknowledge help from Alan Wallcraft, who communicated the status and history of the KRAKEN SHMEM library. Support for this work was provided by AFRL contract F19628-02-C-0037.

Footnotes

* Because of a bug in the default version of the library, we were forced to use the Trace version of the library. When not instrumented by specific profiling calls, the Trace and default versions of the SHMEM library ran at the same speed for large numbers of processors.

** Currently KRAKEN has a maximum of 14.062 GB of user-accessible memory per node (or roughly 1.75 GB per CPU), but during the CAP program, the machine was configured with only 1 GB per CPU.

References

1. Werne, J., and D.C. Fritts, "Stratified Shear Turbulence: Evolution and Statistics," Geophysical Research Letters, 26, 439, 1999.
2. Amdahl, G.M., "Validity of Single-Processor Approach to Achieving Large-Scale Computing Capability," Proceedings of American Federation of Information Processing Societies (AFIPS) Conference, Reston, VA, 483-485, 1967.
3. Gustafson, J.L., "Reevaluating Amdahl's Law," Communications of the Association for Computing Machinery (CACM), 31(5), 532-533, 1998.
4. Lewis, T.G. and El-Rewini, H., "Introduction to Parallel Computing," Prentice Hall, 32-33, 1992.
5. Werne J., P. Adams, and D. Sanders, "Hierarchical Data Structure and Massively Parallel I/O," 2000, www.cora.nwra.com/~werne/Public/Papers/pc.ps.gz.

Providing Ocean Model Information to Assist with the Rescue Efforts After the Indonesian Tsunami

Dr. Frank Bub, Modeling Division Director, Naval Oceanographic Office

On Sunday, 26 December 2004, an earthquake on the western side of Sumatra, Indonesia, initiated a tsunami that spread across the Bay of Bengal and the Indian Ocean, killing at least 250,000. The U.S. Navy was immediately called in to provide relief and help find survivors around northern Sumatra, Sri Lanka, and the Maldive Islands.

The oceanographers of the Naval Oceanographic Office (NAVOCEANO), in turn, were asked to forecast ocean conditions that might hamper the rescue and recovery operations.

Rescue officials were particularly interested in ocean currents in order to determine where survivors and flotsam might drift, water temperatures that might affect survival, and waves and surf that might impede rescue operations.

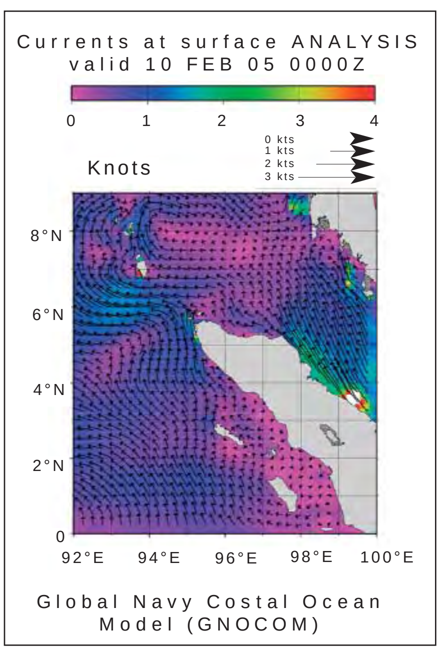
As oceanographic experts, NAVOCEANO personnel responded with information on tsunamis, waves and currents, ocean front locations, and projected drift paths in areas of Sumatra and the Andaman Islands, the Bay of Bengal, eastern Indian Ocean, Sri Lanka, and the Maldive Islands.

Ocean models at NAVOCEANO that forecast this information include the Global Navy Coastal Ocean Model (G-NCOM), the Modular Ocean Data Assimilation System (MODAS), the WAve Model (WAM), and a new near-shore wave model, Simulating WAves Nearshore (SWAN). These models all rely on the computational power available from the NAVO MSRC.

Immediately, JPEG graphics of currents and wave forecasts for these areas were placed on the NAVOCEANO Web site to aid in rescue efforts. (See Figures 1 and 2.)

Realizing that high-resolution wave and surf information was needed for coastal rescue operations, NAVOCEANO worked with Fleet Numerical Meteorology and Oceanography Center (FLENUMMETOCCEN) to bring up 9 kilometer (5 nautical miles) resolution wind forecasts from the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) to force SWAN wave forecasts. Figure 3 shows a SWAN forecast for the northern Sumatra coastline.

NAVOCEANO and the NAVO MSRC are pleased that their technical expertise and computing ability were able to assist the U.S. Navy and other relief and rescue providers in their efforts to render aid to the stricken people of Sumatra, Sri Lanka, and the Maldive Islands.



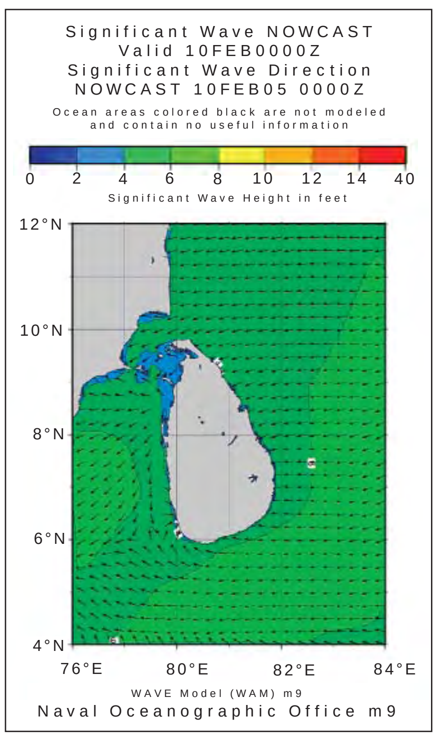
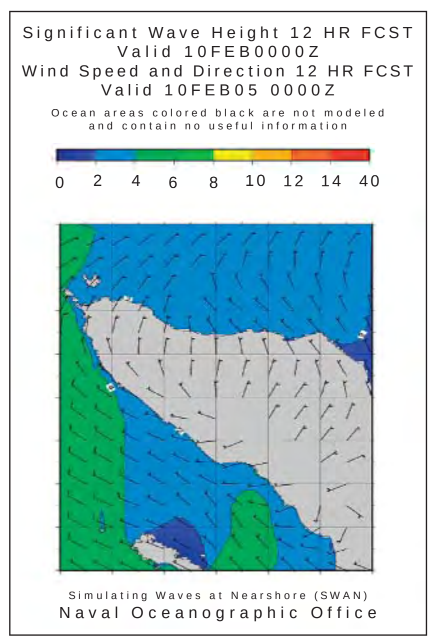


Figure 1 (left), 2 (above). Current and wave forcasts for the Tsunami area. Figure 3 (right). SWAN forecast for the northern Sumatra coastline.



free-to-roll F/A-18E

"Wing drop" is an abrupt un-commanded lateral motion in an aircraft that is caused by an Abrupt Wing Stall (AWS) of one of the wings. This phenomenon is present in many aircraft (and may be present in oncoming fighter aircraft) and adversely affects performance and safety. Development of a computational tool to predict this phenomenon will have a large impact on the design of future aircraft. The preproduction F/A-18E was selected for study due to its susceptibility to "wing drop" in the transonic range.

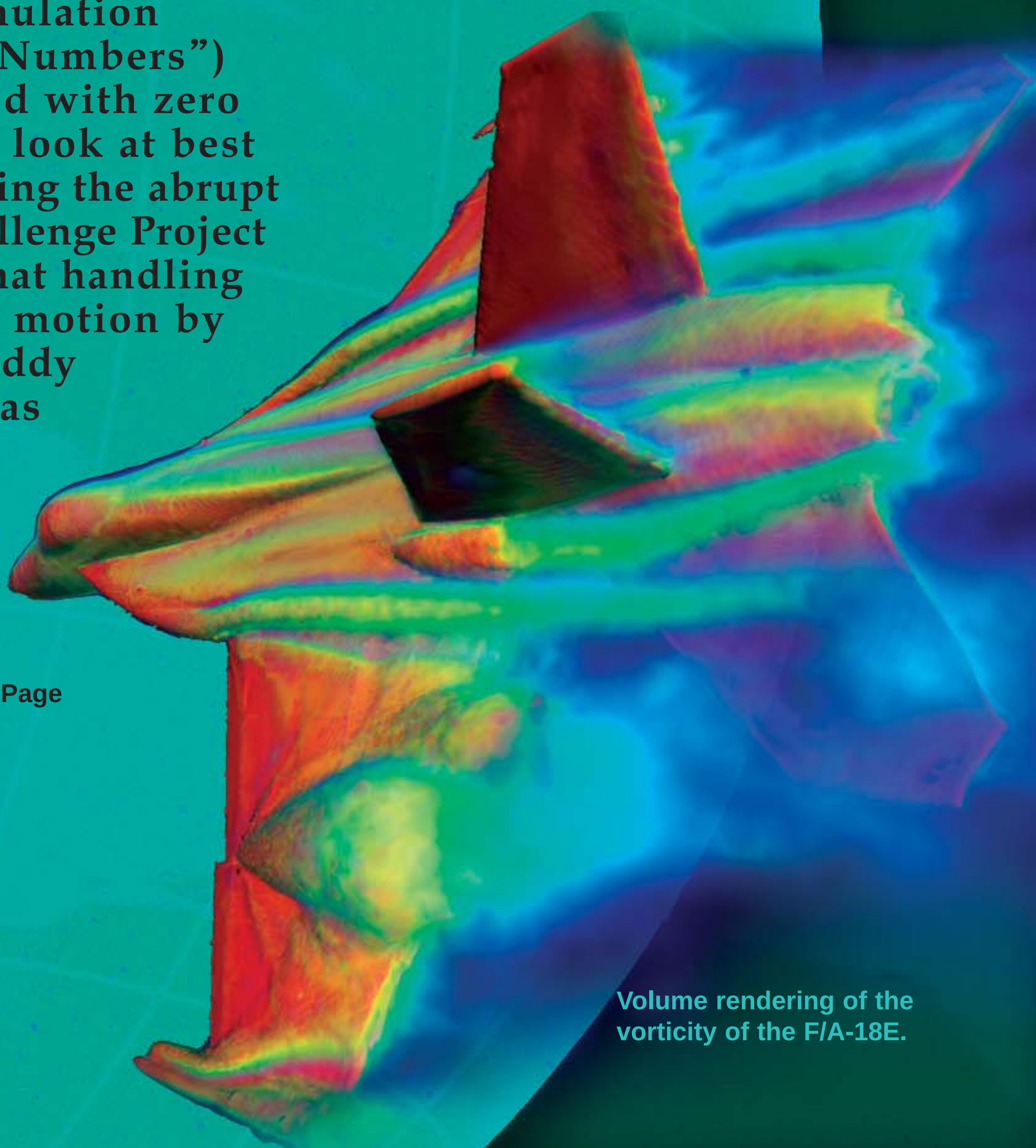
The study took a very careful build-up approach, culminating in a single degree of freedom (or free-to-roll) calculation under a Capability Applications Project (CAP). Calculations (under the Challenge Project, "Multidisciplinary

Capability Applications Project

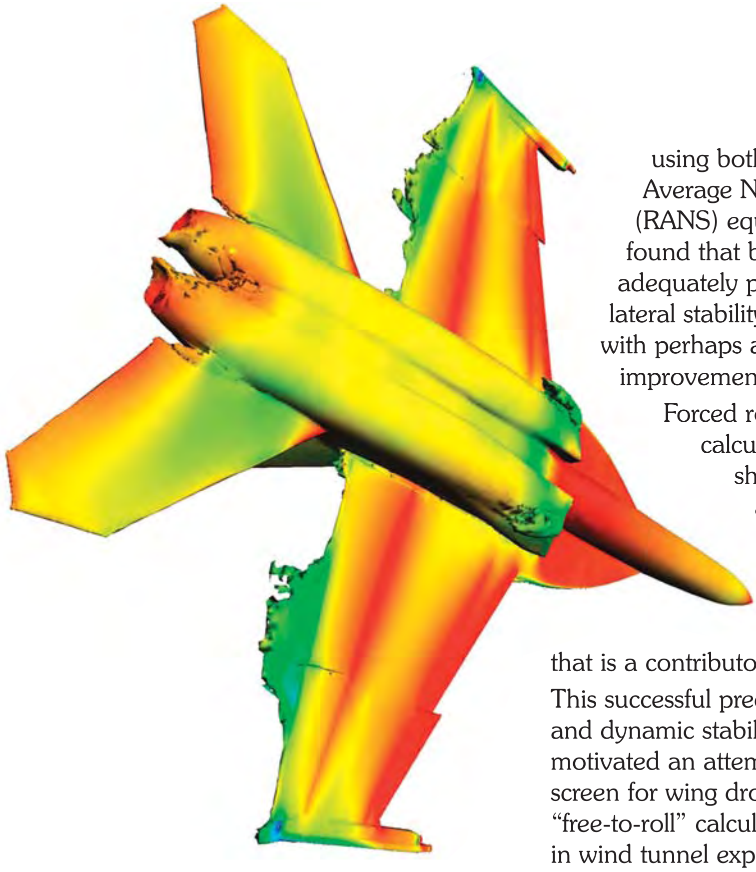
James R. Forsythe, USAF Reserves
Air Force Office of Scientific Research
Civilian: Cobalt Solutions, LLC
John van der Zwaag
VADIC, NAVO MSRC
Scott A. Morton
USAF Academy, Dept. of Aeronautics

Applications of Detached-Eddy Simulation at Flight Reynolds Numbers”) were first performed with zero sideslip in order to look at best practices for predicting the abrupt wing stall. This Challenge Project study determined that handling the unsteady shock motion by using a Detached-Eddy Simulation (DES) was crucial in obtaining adequate predictions.[1] Follow-on work calculated cases in bank/sideslip

Article Continues Next Page



Volume rendering of the vorticity of the F/A-18E.



ABOVE. Bottom view of F/A-18E with pressure color mapped onto an isosurface of vorticity.

using both Reynolds Average Navier-Stokes (RANS) equations and DES found that both methods adequately predicted the static lateral stability characteristics, with perhaps a slight improvement with DES.[2]

Forced roll oscillation calculations, however, showed a large advantage in DES since it successfully predicted a loss/reduction of roll damping that is a contributor to wing drop.[2]

This successful prediction of the static and dynamic stability characteristics motivated an attempt to directly screen for wing drop by performing a "free-to-roll" calculation as was done in wind tunnel experiments using a free-to-roll rig. For both the calculations and experiments, the model was free-to-roll around its longitudinal axis (i.e., it had a single Degree of Freedom (1-DOF)).

The free-to-roll calculation at wind tunnel conditions, however, posed a very serious computational problem. Due to relatively large model inertias (as compared to flight), the wind tunnel model oscillated at a very large period compared to the characteristic timescales of the fluids. In other words, to resolve the fluid dynamics accurately (i.e., the unsteady separation) a small time step was required compared to the model oscillation period.

The range of timescales from smallest to largest was several orders of magnitude. This meant that to capture only two oscillations of the model, 100,000 iterations would be required. At the typical 64 to 128 processors available per job, this would take approximately two to four months of run time.

With the time waiting in the queues added in, the run time could be as

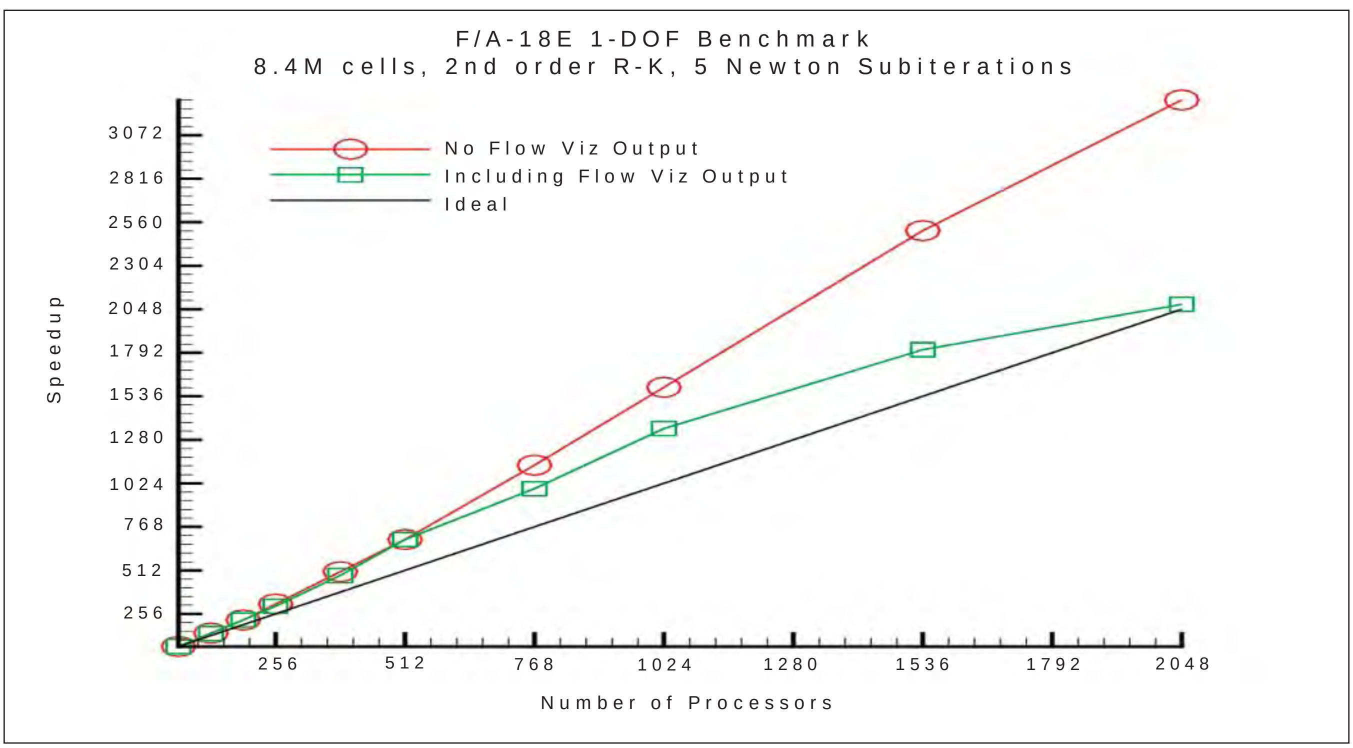


Figure 1. Benchmarks on KRAKEN. Speedup from the 64 processor run versus the number of processors.

long as four to eight months. Thus CAP was identified as the best way to complete these studies in a reasonable length of time. With the 1400 processors available under a CAP, each F/A-18E 1-DOF case was run in five days.

For phase I of the CAP, benchmarks were run on the F/A-18E 1-DOF, with the results shown in Figure 1.

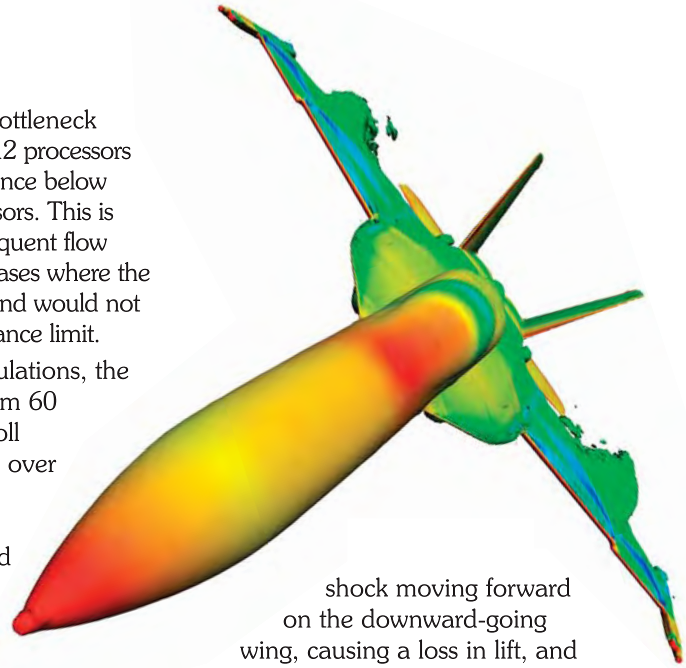
Speedup (in terms of time per iteration) was compared to the 64-processor run. The super-linear speedup and better than ideal efficiency were presumed (as seen in prior simulations) to be caused by an increase in cache efficiency as the problem size per processor is reduced since this is a fixed problem size. Full flow solution files were output every 20 iterations and were included in the timing for the second line plotted.

The first line plots the Central Processing Unit (CPU) time for only the flow solution itself and excludes file Input/Output (I/O). This shows that file I/O begins to bottleneck the scalability beyond 512 processors and brings the performance below ideal, near 2048 processors. This is solely because of the frequent flow solution output-steady cases where the flow is the only output and would not suffer from this performance limit.

For the free-to-roll calculations, the model was released from 60 degrees of bank, and roll response was observed over at least two cycles. Increasing amplitude meant that the case had negative (unstable) roll damping, while decreasing amplitude indicated positive (stable) roll damping. Of four cases examined, one case exhibited unstable roll damping, with the bank angle getting amplified beyond 90 percent. (See Figure 2.)

The spike in rolling moment in Figure 2 prior to *t=0.5* seconds is due to the shock moving forward on the downward-going wing, causing a loss in lift, and accelerating the roll.

Continued Next Page...



ABOVE. Front view of F/A-18E with pressure color mapped onto an isosurface of vorticity.

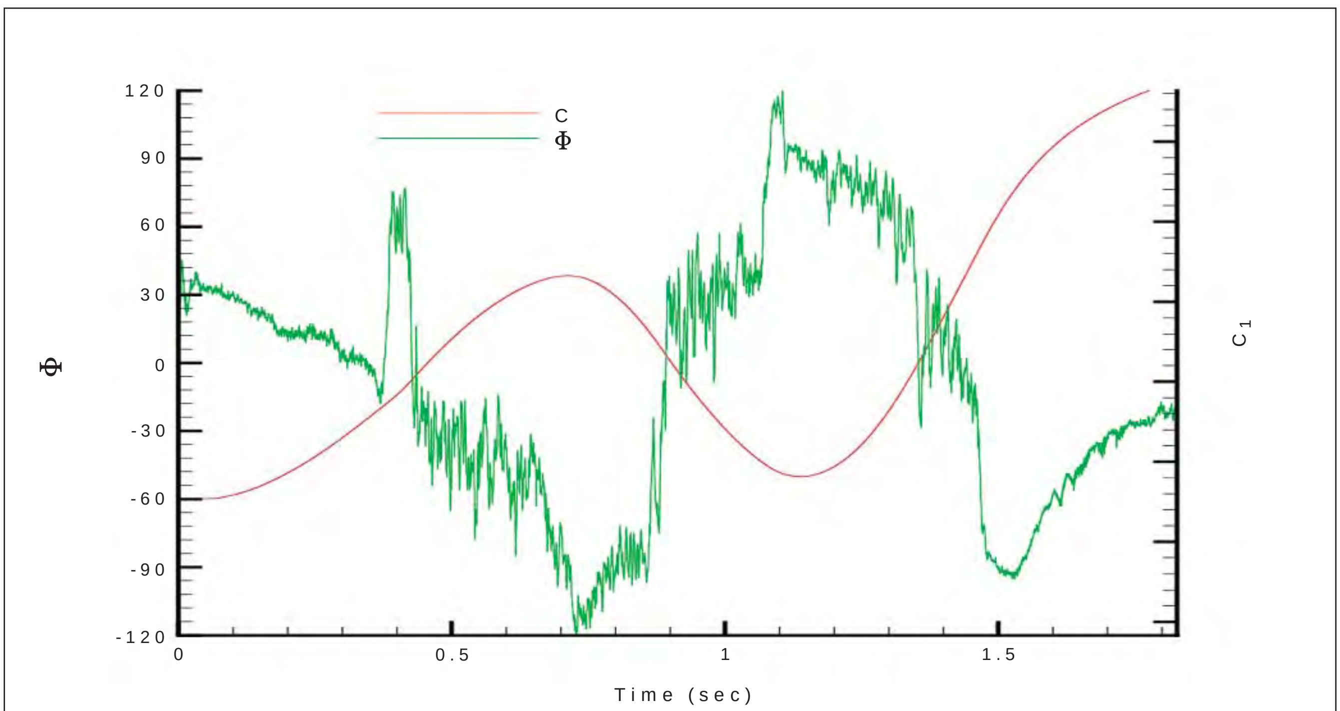
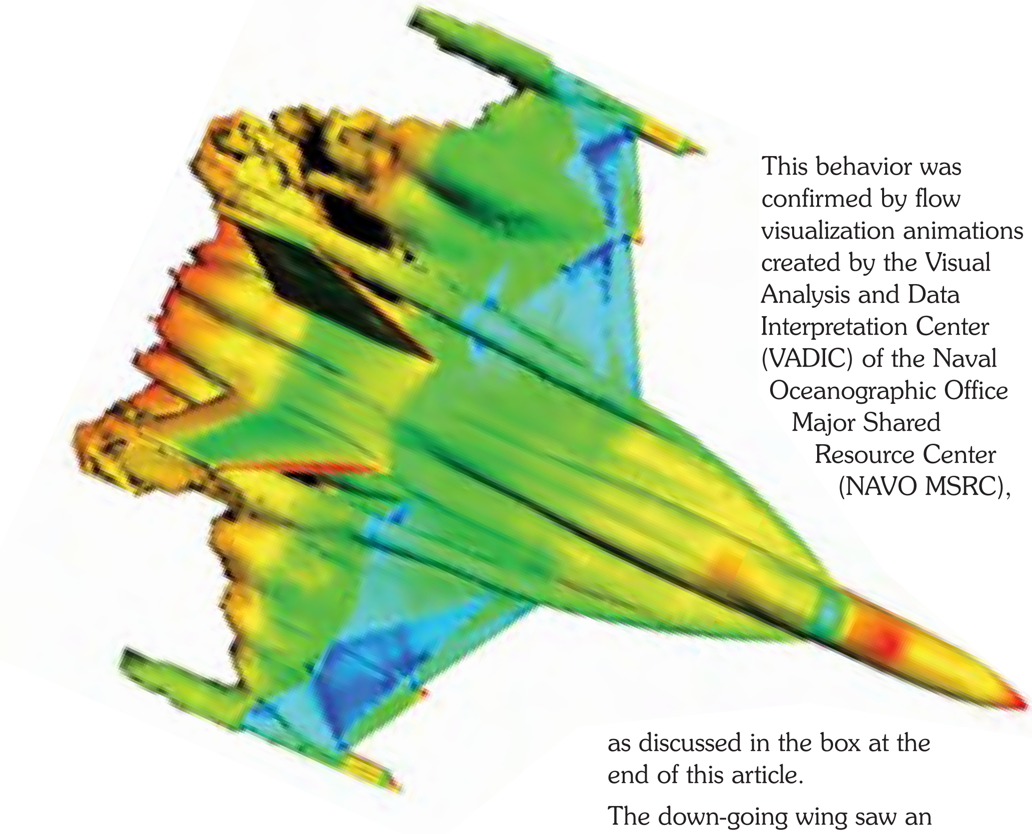


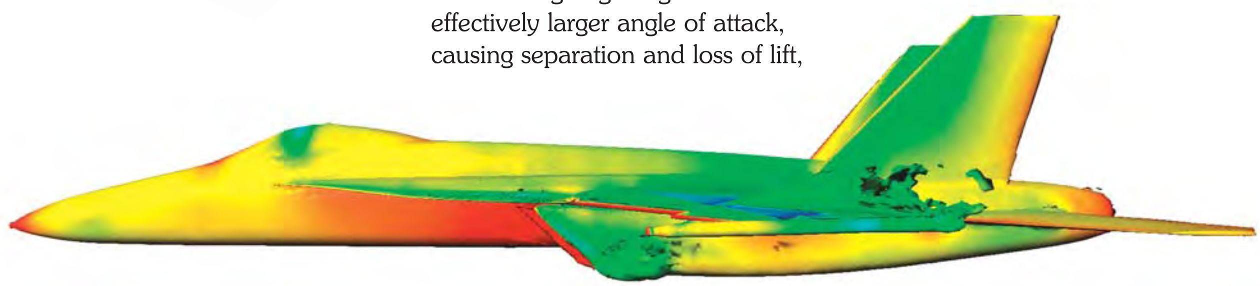
Figure 2. Rolling moment coefficient and bank angle versus time for pitch angle = 6°.



This behavior was confirmed by flow visualization animations created by the Visual Analysis and Data Interpretation Center (VADIC) of the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC), as discussed in the box at the end of this article.

The down-going wing saw an effectively larger angle of attack, causing separation and loss of lift, accelerating the roll. The upward-going wing saw an effectively lower angle of attack, causing the shock separated flow to retreat to the trailing edge, enhancing lift, and increasing the roll rate.

These calculations have helped to provide better physical understanding of the nature of wing drop thanks to the detailed flow visualizations created. The calculations also provide a "proof-of-concept" of a method that industry could use to screen future configurations for wing drop tendencies. Finally, by using large amounts of processors with good scalability, the CAP has provided an example of how difficult problems can be rapidly solved using large parallel computers.



Two views of the DES calculation of the F/A-18E with 1-DOF (free-to-roll). Isosurface of vorticity colored by pressure.

Visualization Creation Methodology

The output data for this project was visualized using an IBM cluster at the NAVO MSRC. Vorticity was computed from the flow field and was then visualized using isosurfaces colored by pressure or directly with volume rendering. By using the open source library, Mesa, animations could be divided across the separate nodes of the IBM cluster and rendered off screen without the need for graphics hardware. Mesa also allowed higher resolution images to be generated and was necessary for generating the volume renderings due to the need for 96-bit color.

Acknowledgements

The authors would like to thank Shawn Woodson of the Naval Air Systems Command (NAVAIR) and Bob Hall and Mike Fremaux of NASA Langley for supporting this work. Also the support of John Skinner and the personnel at NAVO MSRC was greatly appreciated during the CAP.

References

1. Forsythe, J.R., S.H. Woodson, "Unsteady CFD Calculations of Abrupt Wing Stall Using Detached-Eddy Simulation," AIAA Paper 2003-0594, January 2003. Accepted to AIAA Journal of Aircraft.
2. Forsythe J.R, C.M. Fremaux, R.M. Hall, "Calculation of Static and Dynamic Stability Derivatives of the F/A-18E in Abrupt Wing Stall Using RANS and DES," International Conference for CFD, Toronto, Canada, 2004.

Scientific Visualization as Part of the Computational Model Development Process

Sean Ziegeler, Visualization Software Engineer, NAVO MSRC VADIC; George Vahala, The College of William and Mary, Physics Department; Linda Vahala, Old Dominion University, Department of Electrical and Computer Engineering; and Jeffrey Yepez, Air Force Research Laboratory, Hanscom Field

Challenges in Computational Modeling

Computational models are computer programs designed to simulate the behavior of complex processes that are difficult or impossible to measure in the real world. Computational models are applied in nearly every scientific field for many purposes, including weather prediction, oceanography, fluid dynamics, molecular behavior, quantum mechanics, magnetics, biological processes, and genomics—just to name a few.

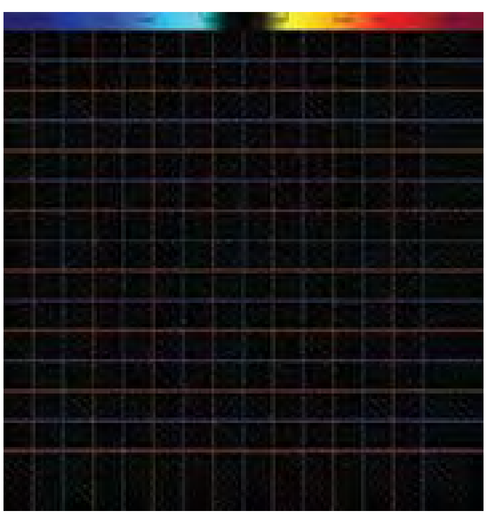
The typical computational model is a sophisticated collection of mathematical expressions and numerical calculations. This intricacy can be a significant source of errors. Moreover, it is often desirable for models to execute as quickly as possible, compelling the model developer to parallelize the computations across many processors or computing systems. This exacerbates the problem by introducing yet another level of complexity.

The purpose of this complex and sophisticated mathematical algorithm is to elucidate some phenomena and sometimes help solve a specific problem. Thus, the developer of the model must overcome three serious challenges: (1) develop an elegant and functional mathematical solution; (2) develop a solution that can work in parallel over multiple processors; and (3) develop a solution that accurately reproduces or represents the phenomena under study.

To mitigate these challenges in the very early stages of model development, it is usually sufficient to examine a sampling of the model output. This requires the developer to look at a few key numbers, or scan through several pages of numbers.

Unfortunately, typical model output ranges from thousands to trillions of numbers, making it difficult or impossible to locate every potential

Continued Next Page...



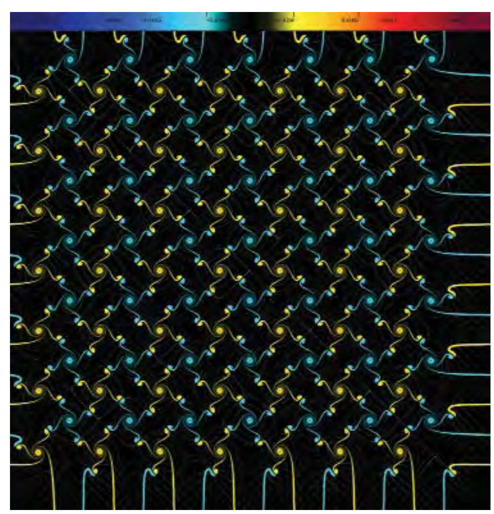


Figure 1. (TOP) An example of a color map for model output values.

Figure 2. Model output at initial state.

Figure 3. Model output at time step 12.

problem. Another approach is to apply traditional program analysis techniques, e.g., stepping through segments of code to ensure proper operation. However, as the model becomes larger and more sophisticated, these techniques become less feasible. The basic shortcoming of these techniques is that only random samples of output are verified.

A more comprehensive solution is to look at the model output all at once. In this case, scientific visualization provides the means to aggregate large amounts of data into a single view. The objective is to render the data into images so the model developer can see all aspects of the data at once to determine if it is behaving correctly.

Two more common applications of scientific visualization are for discovery of novel scientific information and for presentation of information to others; a third use is for verification purposes. The following is a case study of the application of scientific visualization as a verification tool to a Magnetohydrodynamics (MHD) model that is currently under development with the assistance of the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC).

THE MODEL

MHD is the study of the dynamics and flow behavior of electrically conductive fluids such as liquefied metals and plasmas, which compose over 99 percent of the universe. MHD applications include plasma confinement in fusion reactors, liquid-metal cooling for nuclear reactors, fusion reactions within stars, MHD jet thrusters, and the flow of ferro-magnetic material within the Earth (the earth dynamo problem).

Traditional MHD models are based on a combination of Maxwell's equations of electromagnetism and the Navier-Stokes equations. Navier-Stokes equations are a set of nonlinear, convective, partial derivative equations, and solving them computationally is a difficult task.

The use of Lattice Boltzmann (LB) methods resolves this difficulty by embedding the problem into a higher dimensional phase space. This avoids the nonlinear convective derivatives altogether. When using this method, however, developers must choose an appropriate discrete lattice grid in this higher dimensional phase space.

At this point, there is only one nonlinear, algebraic element to the equations, and the original MHD equations can be recovered asymptotically. The resulting Three Dimensional (3D) algorithm, referred to as the Entropic Lattice Boltzmann Navier-Stokes (ELBNS) algorithm, is ideally suited for multi-processor supercomputers. Previous versions of the ELBNS model have registered 22 TeraFlops on the Earth Simulator.

The "Achilles' Heel," however, of simple LB algorithms is nonlinear numerical instabilities, particularly as one turns down the viscosity, thereby increasing the Reynolds number, which is a measure of the turbulence level of the flow. Also, since LB uses discrete lattice symmetry, while the original Navier-Stokes has continuous symmetry, developers must make sure that discrete kinetic effects do not infiltrate the final solution.

This model explores entropic LB schemes that require the enforcement of a discrete H-theorem that governs the increase in entropy of a fluid. This will ensure that the result is an unconditionally stable, explicit numerical scheme.

It should be noted that as the magnetic field is allowed to be zero, the MHD equations reduce to the Navier-Stokes equations, and the preliminary

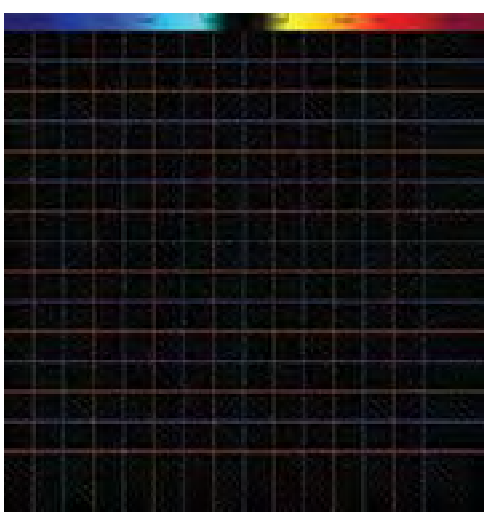


Figure 4. Model output at time step 62.

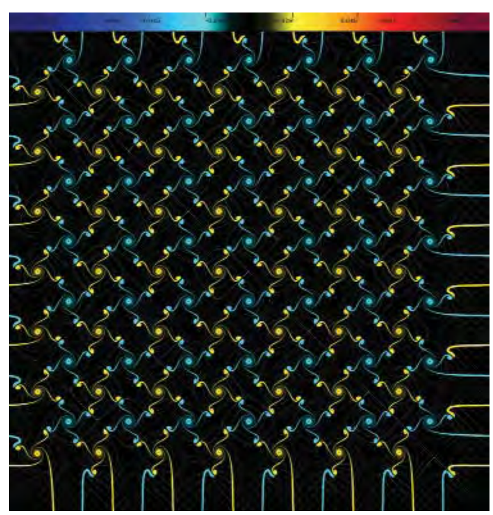


Figure 5. Model output at time step 260.

modeling here is on the Navier-Stokes turbulence. Also, for simplicity, the model was originally designed for Two-Dimensional (2D) Navier-Stokes turbulence, and much of the initial development remained 2D. However, the LB code is readily generalized to 3D flows. Both problem types are presented below along with a discussion of how visualization was able to assist in the development of this model.

TWO-DIMENSIONAL VISUALIZATION

The output of the model in 2D is simply a set of three values: two for vector velocity and one scalar for vorticity (which is a measure of the circularity of flow). The values are arranged on a regular, rectangular grid. This means that the values are evenly-spaced and form the shape of a square. In the case of vorticity, this allows for a straightforward visualization technique that maps the values directly to an image so that each pixel of the image corresponds to one vorticity value.

The value of vorticity for a given pixel is represented by a color, and the color is chosen based on a “color map.” Figure 1 illustrates a color map where -0.3 is deep blue, 0.0 is green, and 0.3 is deep red, and those colors blend for values in between. See Figures 2 through 5 for examples of color-mapped images.

For additional information about the flow, the vector velocity is represented by a set of evenly spaced stream lines. A stream line shows the direction of flow by tracing itself through the vector field until it stops or until it reaches the edge of the data set.

Since the stream lines will be blended into the color-mapped image, it is necessary to use a neutral color such as black or white to avoid confusion with the other colors. The best examples of stream lines from this model are shown in Figures 4 and 5.

The model starts with an initial condition that consists of many interlaced Kelvin-Helmholtz sheets—the lines that criss-cross and form “boxes” in Figure 2. There is global diagonal symmetry (down the diagonal from top-left to bottom-right) as well as the smaller scale symmetry of the many boxes.

By time step 12 (Figure 3), the sheets have collapsed, and one finds beautiful small-scale symmetries as well as the global reflection symmetry about the diagonal. By time step 62 (Figure 4), the small-scale symmetry is starting to break down because of the turbulence, but the global diagonal reflection symmetry persists.

Finally, at time step 260 (Figure 5), the global diagonal is now broken, and the vortices no longer remain localized. This is a situation that the model designers would like to avoid, and it would not have been discovered without this type of visualization.

Because of the high Reynolds number for this flow (on the order of 100,000), the vortices retain their identities until like-rotating vortices come close together and merge. This is an expected behavior that was verified by the visualization. Figure 6 shows four time steps of two vortices rotating in the same direction that merge together.

The visualization also provided some unexpected information on the interaction between unlike-rotating vortices. While these types of vortices do not merge, they can create local flows that can tear a vortex apart.

Figure 7 illustrates two vortices rotating in the opposite direction that, at first, compress together, and then begin to tear each other apart. In the end, the vortex with positive vorticity (orange) was disrupted the most.

Continued Next Page...

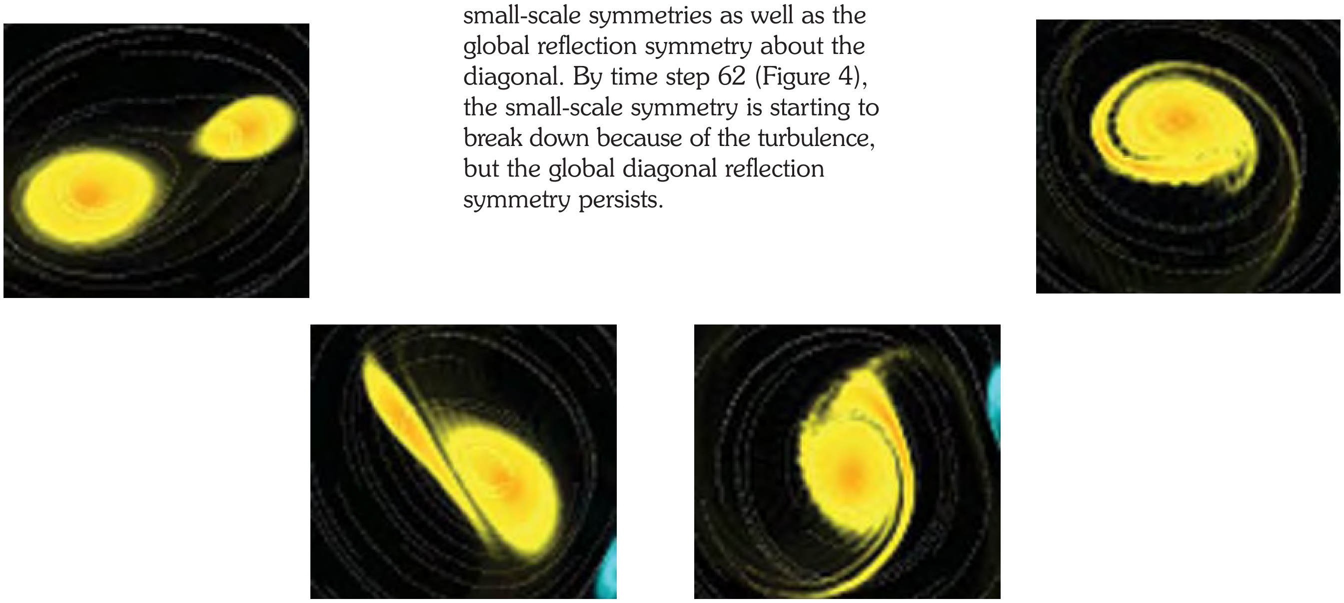


Figure 6. Four time steps, zoomed in to two vortices merging together.

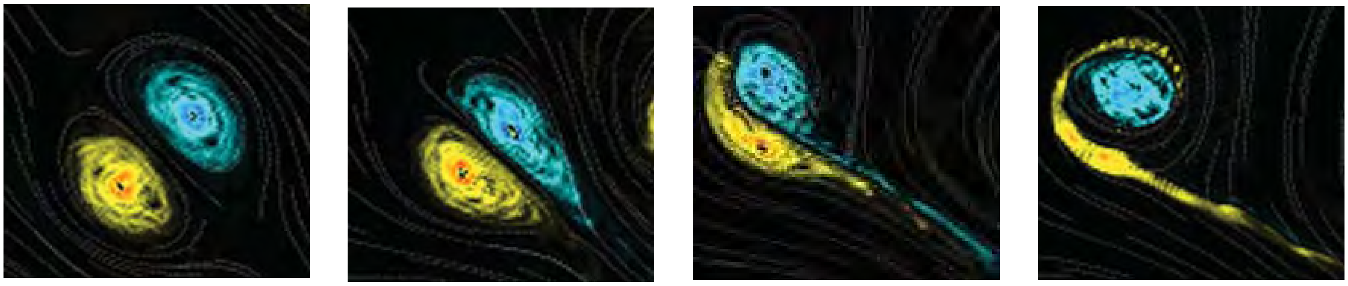


Figure 7. Four time steps, zoomed in to two vortices tearing each other apart.

Three-Dimensional Visualization

The output of the model in 3D is very similar to that of 2D. In this case, however, only the scalar values are present, and the values are arranged onto evenly spaced points within a cube.

In addition, rather than investigating vorticity, which is now a 3D vector, there are two important available scalars: enstrophy (a measure of the magnitude of the vorticity squared) and kinetic energy.

The most straightforward method of visualizing data in this form is Direct Volume Rendering (DVR) and is similar to the 2D method. The similarity is that the data are plotted directly in 3D using a color map to determine colors for specific values. However, rendering the data in this fashion forms a cubic volume in which some points are obscured. The points in front will occlude the points in the back.

The solution is to make the data points somewhat transparent, or less opaque. It is usually best if the opacity is determined by the data value, similar to the way color is determined. A common approach, and the one used for this research, is to have low data values map to low opacity and high data values to high opacity. Figures 8 and 9 show DVR images for enstrophy and kinetic energy densities.

While DVR presents an excellent overview of the data set as a whole, the model designers are also interested in specific key regions of the data set. To address this concern, cutting planes are used and can be placed at any arbitrary location within the data set.

A cutting plane is a plane that slices through the inside of the data set, color mapping the values from the data set that lies on the plane.

The cutting planes can be integrated into the visualization as a whole so that the overall view from the DVR is still present in addition to the specific view of the cutting planes. Figures 10 and 11 illustrate cutting planes integrated with the DVR for enstrophy and kinetic energy. There are two cutting planes placed in the middle of the data set facing away from the view point.

Even with the opacity, it is still difficult to see data near the back end of the data set. Also, seeing certain aspects of the 3D structure of the data is only possible by looking at the data from view points other than the front and sometimes even multiple view points. This is an additional complication that is not an issue in 2D.

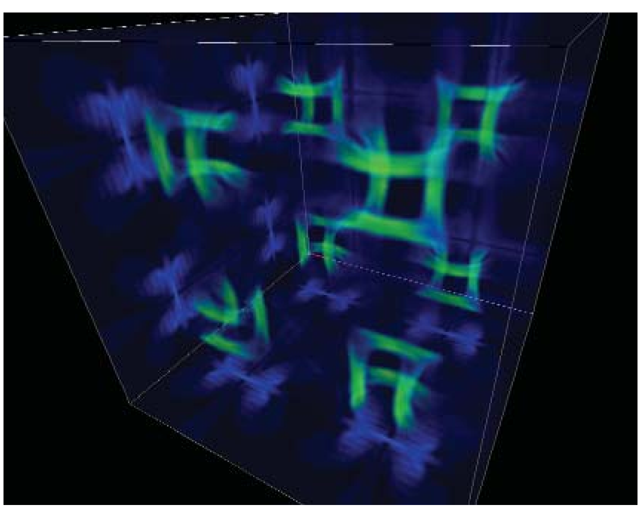


Figure 8. DVR of enstrophy.

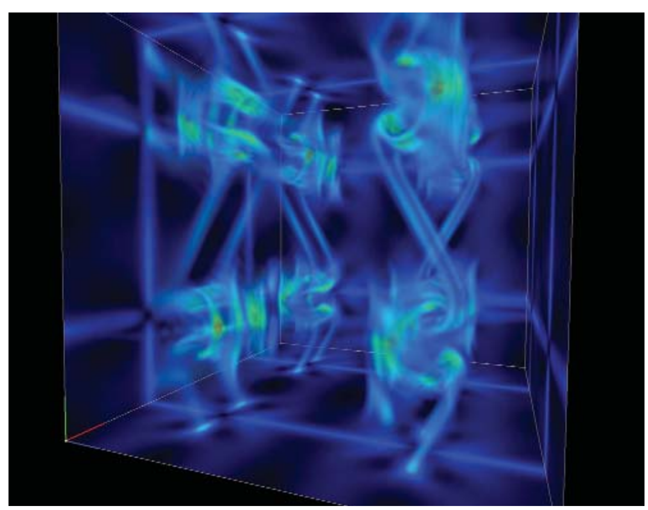


Figure 9. DVR of kinetic energy.

The solution is to make the visualization program interactive. For example, FlowFusion, the visualization program VADIC personnel created for the ELBNS model, allows dragging the mouse to turn the data set so that it can be seen from any angle. Other similar features include moving the data set, zooming in or out, advancing through multiple time steps, and taking snapshots of the current view.

In the 2D case, only initial tweaking of the color map was necessary. However, DVR in 3D often requires changing the color and opacity maps to fit every new data set. The solution is an interactive color and opacity map editor, allowing changes to the maps on the fly.

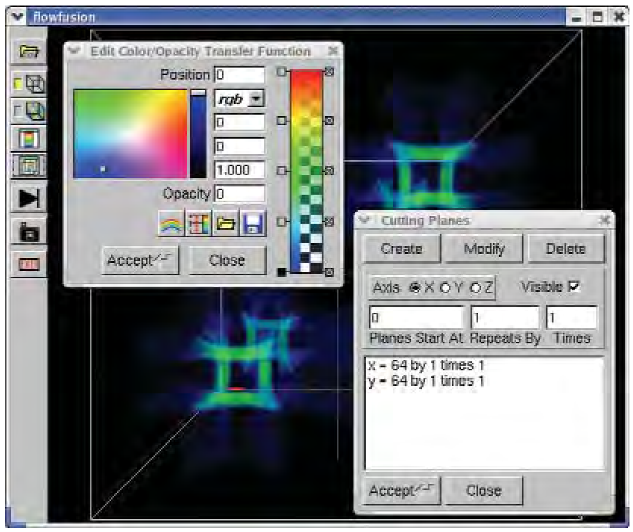
Since the color and opacity maps are both related to the data-values, they can be rolled together, simplifying the issue somewhat. FlowFusion has a combined color/opacity map editor, which is shown in the left window of Figure 12.

The final feature of FlowFusion is an interactive cutting plane editor. This is required to allow creating, deleting, and moving cutting planes through the data. The cutting plane editor for this program also allows repetition of cutting planes: for example, five cutting planes space evenly from one side to the other. The right window in Figure 12 shows the cutting plane editor.

Conclusion

Computational models can be very complex, making it cumbersome and sometimes impossible to develop without the assistance of scientific visualization. In many cases, the model developers were able to quickly verify that certain common problems were not present. When such problems were found, the model was rectified, and the visualization process was repeated to verify that the problem was solved.

In other cases, the model designers discovered unexpected results that the traditional model verification methods probably would never have found. This provided the designers with a highly effective quality control mechanism. Without the visualization techniques above, the development of this model would have likely taken much longer and potentially been less accurate.



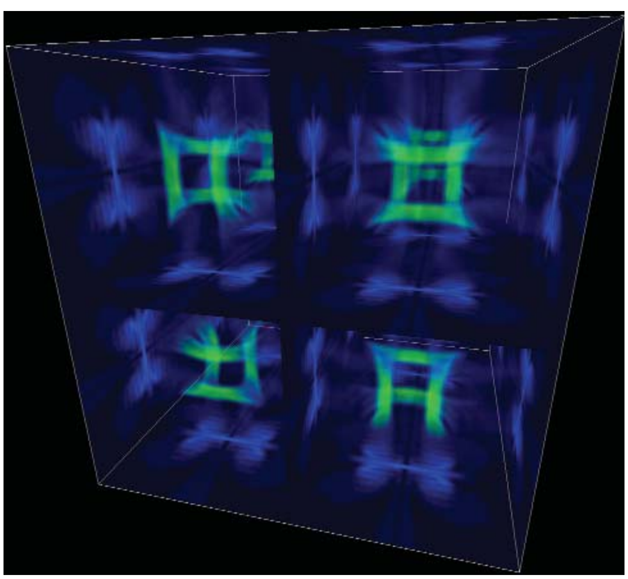


Figure 10. (LEFT) DVR and two cutting planes.

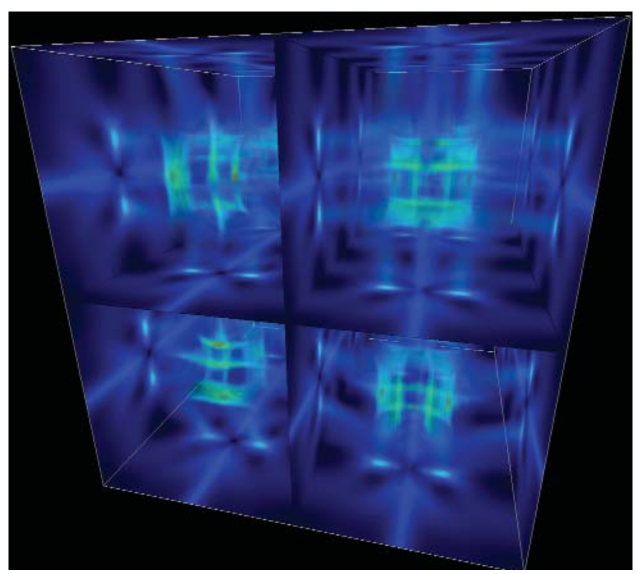


Figure 11. (RIGHT) DVR and two cutting planes of enstrophy planes of kinetic energy.

Figure 12. (TOP) A snapshot of the visualization program with color/opacity map and cutting plane editors.

The Porthole

Joint Air Force-Boeing group visits NAVO MSRC Operations

MSRC Director Steve Adamec leads Mr. Stephen Perry, Administrator of GSA, and party on a tour of Operations

Personnel from the Republic of Korea Navy visit the Visual Analysis and Data Interpretation Center

Joint Air Force-Boeing group tours the Visual Analysis and Data Interpretation Center

Dr. Parney Albright, Assistant Secretary for the Department of Homeland Security, visits NAVO MSRC Operations with his party

N21 New Hire Orientation

Foreign Officers participating in the NAVOCEANO International Hydrographic Management and Engineering Program visit theVisual Analysis and Data Interpretation Center

NAVOCEANO's Paul Stephens and Pete Gruzinskas, with visitors from the Oceanographer of the Navy's staff during a tour of the Visual Analysis and Data Interpretation Center

LCDR Chris Sterbis, LCDR Oscar Monterossa, Christine Cuicchi, and Dave Cole in NAVO MSRC Operations

Pete Gruzinskas, Phil Webster (NASA Goddard), and Steve Adamec

Dave Cole leads a tour of Coahoma Community College Students through NAVO MSRC Operations

Programming Environment and Training (PET) Collaborative and Distance Learning Technologies (CDLT) personnel meet in the Visual Analysis and Data Interpretation Center

Coahoma Community College Students visit NAVO MSRC Operations

Naval Meteorology and Oceanography (METOC) Students visit the Visual Analysis and Data Interpretation Center

NAVO MSRC PET Update

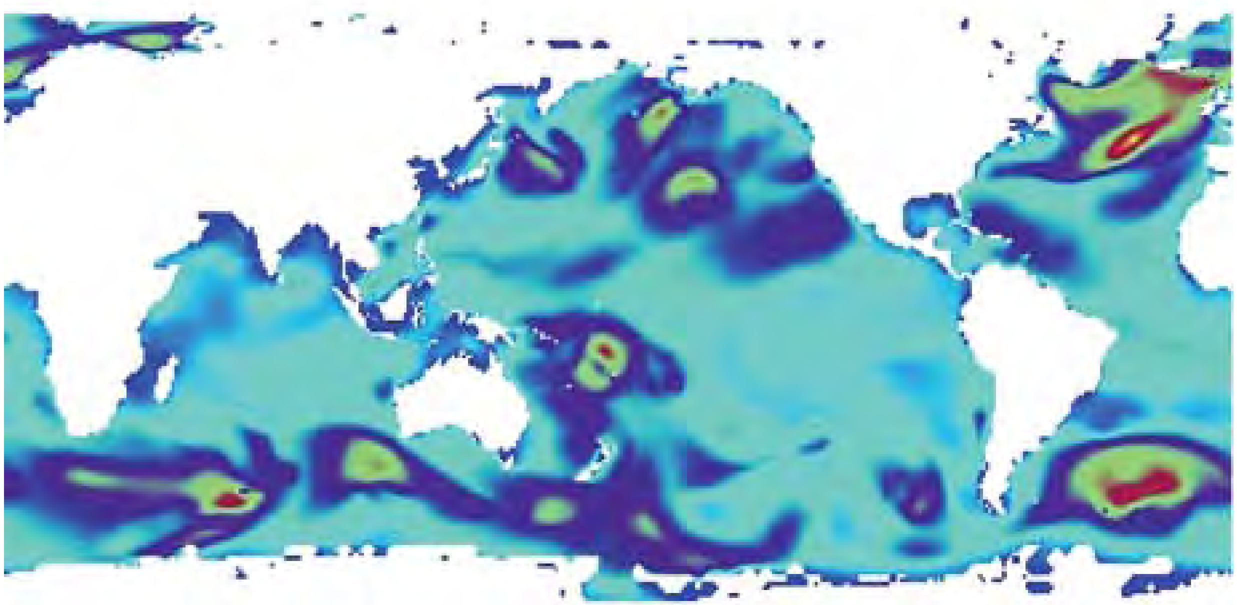
Eleanor Schroeder, NAVO MSRC Programming Environment and Training Program (PET) Government Lead, and Tom Cortese, ICL/UTK PET Computer Environment On-Site

During the summer of 2004 Programming Environment and Training (PET) Component One had the privilege of hosting five undergraduate students through the PET Summer Intern Program. Three of those students worked on projects within the Computational Environments (CE) Functional Area. A summary of their experiences and accomplishments written by Tom Cortese (PET CE On-Site), is in the Fall 2004 Navigator. The other two students, Allison Scogin and Benjamin Payment, both from Mississippi State University, worked on projects within the Climate, Weather, and Ocean Modeling (CWO) Functional Area. This article provides a summary of Allison's and Benjamin's experiences and accomplishments.

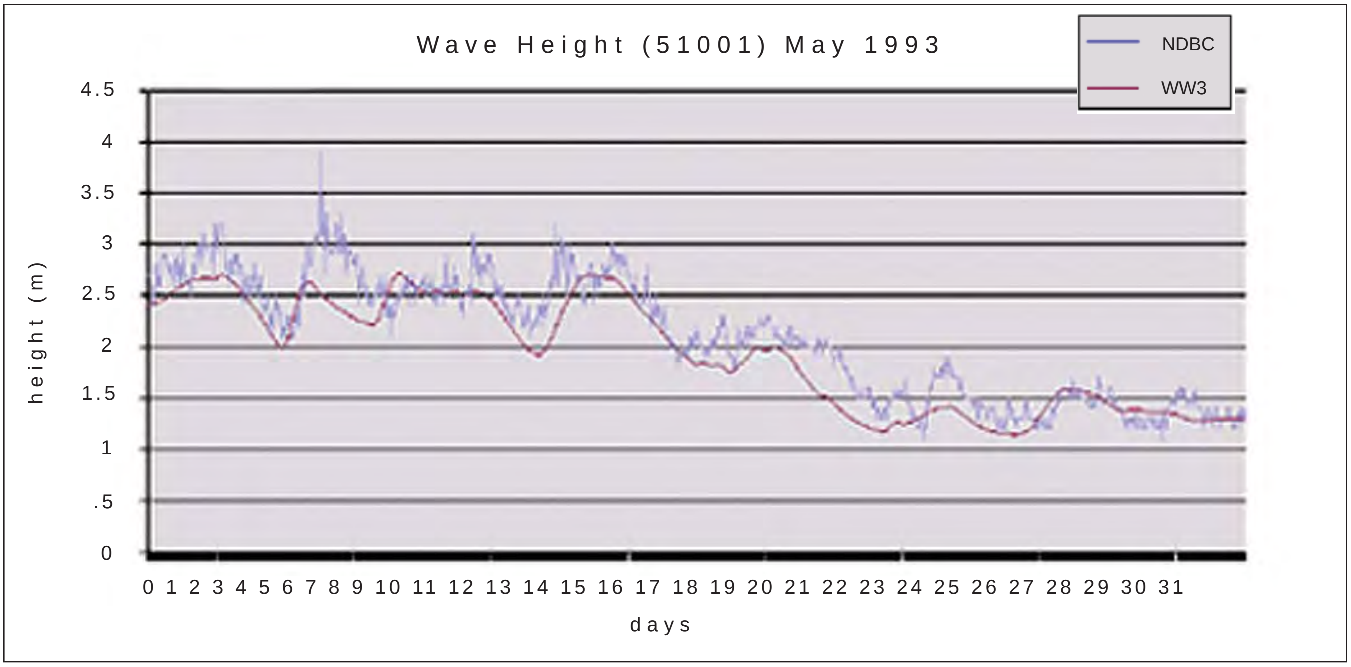
Allison Scogin

Allison Scogin worked in the Ocean Dynamics and Prediction Branch at the Naval Research Laboratory (NRL), Stennis. Under the mentorship of Dr. James Dykes, Allison undertook a 10-year simulation of the worldwide wave conditions. The simulations used WAVEWATCH III, a third generation Message Passing Interface (MPI) parallel wave model developed at National Oceanic and Atmospheric Administration/National Centers for Environmental Prediction (NOAA/NCEP), that solves the spectral action density balance equation for wave number and direction spectra. High-Performance Computing (HPC) made it possible to compute wave conditions over the entire Earth for 1 month in less than 2 hours wall-clock time. Completing the 10-year simulations required Allison to learn to set up the model input data, run the model, and process and evaluate the output. These tasks in turn required learning in the areas of shell scripting and batch schedulers.

Several data-checking measures were used to assure the accuracy of model results. Comparisons of wave heights from WAVEWATCH III were made with corresponding National Data Buoy Center (NDBC) data. After it was discovered that the model wave heights were not as accurate as expected, comparisons were made for the



Snapshot of WAVEWATCH III significant wave heights during 1993.



Comparison of WAVEWATCH III significant wave heights and NOAA NDBC data during the month of May 1993.

input wind speeds and directions. A significant amount of time was spent troubleshooting for the cause of these problems. Eventually it was discovered that some of the wind input files were missing either their *u*-component or their *v*-component. While the incomplete input files were accepted by the model, the missing data were not properly accounted for. The wind input files were reprocessed with interpolation for the missing data, and the 10-year simulation repeated. Improved results were observed.

In describing her summer intern experience Allison states: "The PET Summer Internship Program has been both challenging and rewarding. I do believe that this experience will help me in my future endeavors. The information that I have been exposed to through PET has enhanced my knowledge and skills with both problem solving and computer systems. I would recommend the PET internship to any student interested in learning both about High Performance Computing and how [it] can be used to enhance scientific research."

Benjamin Payment

Benjamin Payment worked within NAVO MSRC PET under the mentorship of Dr. Tim Campbell (PET CWO On-Site). Benjamin's project focused on improving the linear solver in a Two Dimensional (2D) time-dependent fluid model used for internal solitary wave research at NRL Stennis Space Center (SSC). Internal waves (i.e., waves below the ocean surface) are typically generated by the interaction between tidal flow and bottom topography. A better understanding of the internal solitary waves can improve modeling of ocean acoustics which can, in turn, benefit a range of applications from ocean floor mapping to mine detection.

During each time step of the fluid model, a large set of linear equations, known as the projection matrix ($Ax=b$), is solved. Solving the projection matrix consumes a majority of the time and memory in the model. The projection matrix, which depends on the numerical grid, is symmetric, positive definite, and block tridiagonal with tridiagonal blocks. If the grid has dimensions M by N, then the matrix has dimensions $M*N$ by $M*N$. When the grid is fixed, the projection matrix does not change during the time stepping. However, plans are being made to change the model to an adaptive grid, thus causing the projection matrix to change with each time step.

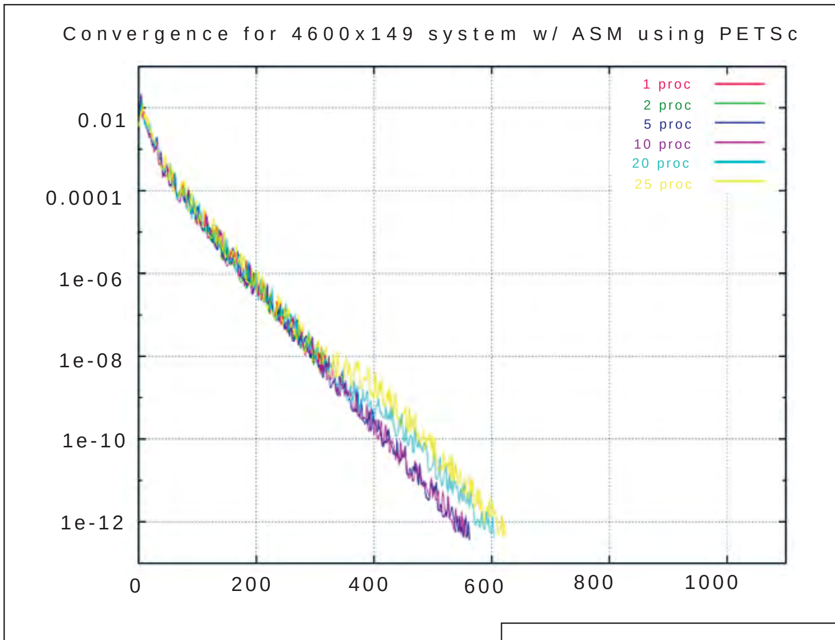
A sequential direct solver is currently used in the model to solve the linear system. One approach to improving the solver is to use a parallel direct solver library, like SuperLU. Benjamin performed some timings of the SuperLU solver that Dr. Campbell implemented prior to the summer. Timing results showed that the LU decomposition time decreased as the number of processors increased. However, the time for the sequential forward/backward substitution increased with the number of processors.

With regard to parallelism, a better approach is to use an iterative solver technique. Benjamin's primary task was to implement a projection matrix solver using the Portable Extensible Toolkit for Scientific Computation (PETSc).

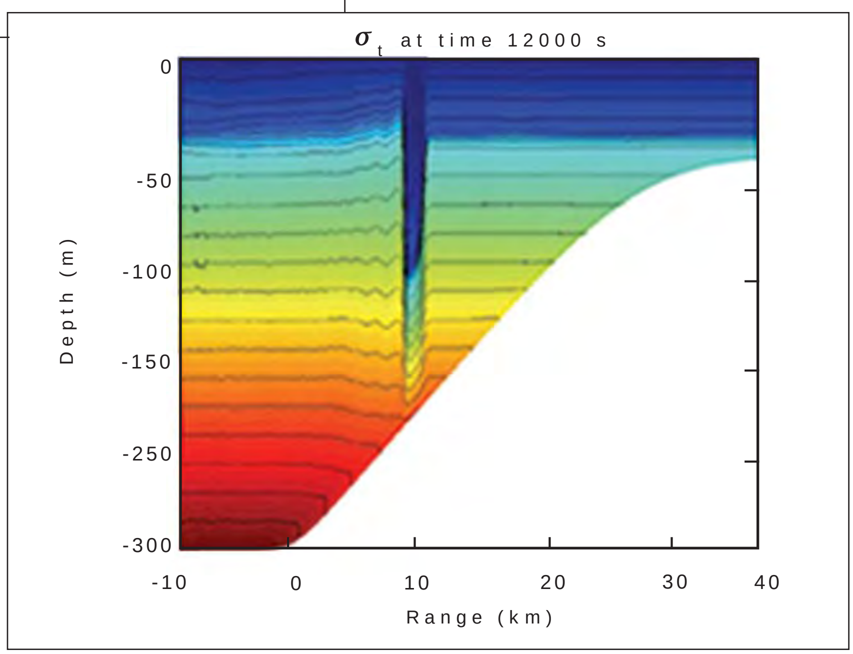
PETSc is a software toolkit that contains a powerful set of tools for the numerical solution of partial differential equations and related problems on high-performance computers (http://www-unix.mcs.anl.gov/petsc/petsc-2/). PETSc has many iterative solvers and preconditioners and supports many matrix formats. One very powerful tool that PETSc offers is the ability to change almost all aspects of a solver via command line arguments.

Benjamin successfully developed a projection matrix solver in the PETSc environment. Subsequently, Benjamin measured the performance of the solver on the NAVO MSRC IBM for various sparse matrix storage formats and types of preconditioners. The results of Benjamin's project have become important in making decisions about how to improve the fluid model.

In describing his summer internship experience Benjamin states: “My summer internship with PET has been challenging, educational, and rewarding, and I had some fun along the way. Initially, I was inundated with a completely new work environment and tools, but along the way I have acquired many new skills and fine-tuned others.”



Left: Plot of convergence of the PETSc Additive-Schwarz preconditioned Conjugate-Gradient for different numbers of processors.



Right: Schematic of a shoaling solitary wave (moving left to right); the color represents density (blue: lower, red: higher).

Navigator Tools and Tips

Using and Comparing Load Share Facility (LSF) and LoadLeveler

Sheila Carbonette, NAVO MSRC User Support

The Platform Computing Load Share Facility (LSF) queuing system will soon replace LoadLeveler as the batch queuing system on the unclassified IBMs at the NAVO MSRC. (It is already available on the classified IBM.) This article is intended to serve as a brief overview of LSF and offer a comparison of LSF and LoadLeveler queuing system commands, environment variables, and batch scripts.

In order to use LSF, users do not have to add anything to their own setup files. The needed environment variables have been added to the system default setup files.

LSF and LoadLeveler are alike in that they are both systems that schedule users' jobs. They both have commands that allow users to submit jobs, check the status of the job/queues, and hold/cancel the job. The main difference is the syntax of these commands. For example, users who submit parallel MPI jobs now have to specify a submit option "#BSUB -a poe" and then use "mpirun.lsf" to run the executable. Below are example LSF scripts to run serial, parallel (MPI), and parallel (OpenMP) jobs. Tables that list some of the more common queuing system commands, submit options, and environment variables can be found at the end of this article.

Sample LSF Script to Run a Serial Job

```
#!/bin/csh
#BSUB -J serialjob          # Name of the job.
#BSUB -o %J.out             # Appends  std
                            output to file
                            %J.out. (%J is the
                            Job ID)
#BSUB -e %J.err             # Appends std
                            error to file
                            %J.err.
#BSUB -P NAVOSLMA           # Project ID.
#BSUB -q batch              # queue
#BSUB -n 1                  # Number of CPUs
# Compile Fortran code
xlf90-o serial.exe serial.f
# Run the serial executable
./serial.exe
#End of Sample LSF Script
```

Sample LSF Script to Run a Parallel (MPI) Job

```
#!/bin/csh
#BSUB -J mpijob             # Name of the job.
#BSUB -o %J.out             # Appends std
                            output to file
                            %J.out. (%J is the
                            Job ID)
#BSUB -e %J.err             # Appends error to
                            file %J.err.
#BSUB -a poe                # Esub parameter.
#BSUB -P NAVOSLMA           # Project ID.
#BSUB -W 2:00               # Wall clock time
                            of 2 hours.
#BSUB -q batch              # Queue name.
#BSUB -n 32                 # Number of CPUs.
#BSUB -R "span[ptile=8]"    # Number of tasks
                            per node.
# Run the MPI job with "mpirun.lsf"
mpirun.lsf ./c_hello
#End of Sample LSF Script
```

Sample LSF Script to Run a Parallel (OpenMP) Job

```
#!/bin/csh
#BSUB -J ompjob             # Name of the job.
#BSUB -o %J.out             # Appends std
                            output to file
                            %J.out. (%J is the
                            Job ID)
#BSUB -e %J.err             # Appends std
                            error to file
                            %J.err.
#BSUB -a 'poe ompenmp'      # Esub parameter.
```

```
#BSUB -P NAVOSLMA          # Project ID.

#BSUB -W 4:00              # Wall clock time
                           of 4 hours.

#BSUB -q batch             # Queue name.

#BSUB -n 8                 # Number of CPUs.

#BSUB -R "span[ptile=8]"   # Number of tasks
                           per node.

# Run the OpenMP job with "mpirun.lsf"

mpirun.lsf ./ompj.exe

#End of Sample LSF Script
```

The tables below list some of the more common queuing system commands, submit options, and environment variables. More information can be found on the NAVO MSRC Web site: http://www.navo.hpc.mil.

Queuing System Command Comparison

LoadLeveler	LSF	Description
llsubmit script	bsub< script	Submit a job script for execution.
llq	bjobs bhist	Show Status of running and pending jobs. Displays historical information about your jobs.
llcancel	bkill	Kill a job.
llhold	bstop	Hold a job.
llclass showqlimits	bqueues	Show configuration of queues.
	busers	Displays information about users and groups.
	bpeek	Displays the **stderr** and **stdout** of an unfinished job.
	bacct	Displays accounting information for finished jobs.
llstatus	bhosts	Summarize load on each host.

Frequently Used Options in Job Scripts

LoadLeveler	LSF	Description
#@ job_name = jobname	#BSUB -J jobname	Assigns name to job.
#@ notify_user = login_name #@ notification = start	#BSUB -B	Sends email when job begins execution.
#@ notification = complete	#BSUB -N	Emails finished job report.
#@ error = errfile	#BSUB -e errfile	Redirects **stderr** to specified file.

LoadLeveler	LSF	Option
#@ output = out_file	#BSUB -o out_file	Redirects **stdout**.
	#BSUB -a application	esub parameter.
#@ account_no = project_name	#BSUB -P project_name	Assigns job to specified project.
#@ wall_clock_limit = runtime	#BSUB -W runtime	Sets the run limit of the job.
#@ class = queue_name	#BSUB -q queue_name	Submit the job to the specified queue.
#@ node = num_nodes	#BSUB -n num_procs	Specifies number of processors to use.
#@ tasks_per_node = num_procs	#BSUB -R "req_req"	Specifies resource requirements.

Environment Variable Comparison

LoadLeveler	LSF	Variable Description
LOADL_JOB_NAME	LSB_JOBID	Unique job number.
LOADL_STEP_ID	LSB_JOBINDEX	Job index for array jobs.
LOADL_STEP_COMMAND	LSB_JOBNAME	Name of the job.
LOADL_PID	LS_JOBPID	Process ID of the job.

Platform's LSF is a powerful batch scheduling system that runs on multiple operating systems with a common command set. Consequently, LSF is an integral part of establishing a common batch scheduling system across all Major Shared Resource Centers. In addition to the basic batch scheduling commands described in this article, LSF has additional features and capabilities to simplify job submission and data transfer activities. These additional features will be highlighted in future Navigator articles.

Coming Events

27 June - 1 July
Users Group Conference 2005
Nashville, TN
www.hpcmo.hpc.mil/Htdocs/UGC/UGC05/

12 - 15 July
SCC 2005: IEEE International Conference on Services Computing
Orlando, FL
http://conferences.computer.org/scc/2005

24 July
CLADE 2005: Workshop on Challenges of Large Applications in Distributed Environments
Research Triangle Park, NC
www.cs.umd.edu/CLADE2005/

24 - 27 July
HPDC 2005: 14th IEEE International Symposium on High-Performance Distributed Computing
Research Triangle Park, NC
www.hpdc.org

11 - 13 August
SERA 2005: 3rd ACIS International Conference on Software Engineering Research, Management & Applications
Mt. Pleasant, MI
http://acis.cps.cmich.edu:8080/SERA2005

15 - 17 August
SIP 2005: 7th IASTED International Conference on Signal & Image Processing
Honolulu, HI
www.iasted.com/conferences/2005/hawaii/c479.htm

16 - 18 August
ICSEng 2005, 18th International Conference on Systems Engineering
Las Vegas, NV
www.icseng.info/

30 - 31 August
MAS&S 2005: IEEE 2nd Symposium on Multi-Agent Security & Survivability
Philadelphia, PA
www.cs.drexel.edu/mass2005

27 - 29 September
ICVS 2005, 4th IEEE International Conference on Computer Vision Systems
New York, NY
www.cs.colostate.edu

27 - 29 September
CLUSTER 2005: IEEE International Conference on Cluster Computing
Boston, MA
www.cluster2005.org

2 - 5 October
ICCD 2005 - International Conference on Computer Design: VLSI in Computers & Processors
San Jose, CA
www.iccd-conference.org

7 - 8 October
GridNets 2005: 2nd International Workshop on Networks for Grid Applications (with BroadNets 2005)
Boston, MA
www.gridnets.org

19 - 22 October
FIE 2005: Frontiers in Education Conference
Indianapolis, IN
www.fie-conference.org

23 - 28 October
IEEE Visualization 2005
Minneapolis, MN
http://vis.computer.org/vis2005

26 - 28 October
ANCS 2005: Symposium on Architectures for Networking & Communication Systems
Princeton, NJ
www.ancsconf.org

8 - 11 November
ISSRE 2005: 16th IEEE International Symposium on Software Reliability Engineering
Chicago, IL
www.issre.org

12 - 18 November
SC|05: Supercomputing 2005
Seattle, WA
http://sc05.supercomputing.org/

26 - 30 November
ICDM 2005: 5th IEEE International Conference on Data Mining
New Orleans, LA
www.cacs.louisiana.edu/~icdm05

Naval Oceanographic Office * MAJOR SHARED RESOURCE CENTER

1002 Balch Boulevard . Stennis Space Center, Mississippi . 39522