



EARDC
MSARC
Major Shared Resource Center
RESOURCE

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE 2007		2. REPORT TYPE		3. DATES COVERED 00-00-2007 to 00-00-2007	
4. TITLE AND SUBTITLE ERDC MSRC Resource. Spring 2007				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center,ATTN: ERDC MSRC HPC Resource Center,3909 Halls Ferry Road,Vicksburg,MS,39180-6199				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			



John West

Director
Scientific Computing Research Center
ERDC Information Technology Laboratory
and former Director, ERDC MSRC

from the director . . .

New leadership at the ERDC MSRC

Five years ago, I wrote my first piece for the front of this newsletter. During that time, much has changed in industry, in the Modernization Program, and in the U.S. Army Engineer Research and Development Center Major Shared Resource Center (ERDC MSRC) itself.



David Stinson

Acting Director
ERDC MSRC

The MSRC is always a moving target, but I'm proud of the fact that today it hosts one of the largest production systems in the Department of Defense (DoD). With our expertise and resources, we're providing a tremendous competitive advantage to the DoD RDT&E communities and to our soldiers in harm's way.

It has been a humbling experience to have worked alongside the incredibly talented team at the ERDC MSRC. I've been part of this program since the very beginning and grown up professionally with the people in this center. We've built a world-leading high performance computing (HPC) infrastructure, staffed with the best expertise anywhere. And now, with efforts like the User Interface Toolkit and the Data Analysis and Assessment Center, we are continuing to build the future of supercomputing by leading programwide efforts to make HPC more accessible to an even broader community of scientists and engineers.

This is an exciting time in HPC, and I'm very pleased to hand the Director's office over to David Stinson, a long-time HPC and Modernization Program veteran.

Dave's history with the Program goes all the way back to the time we served together on the source selection team for the original MSRC center contracts in the mid-90s. He has had a variety of leadership roles in this center that intersect all facets of the user experience—from leading the Customer Assistance Center to serving most recently as the MSRC's Assistant Director for Operations and Administration.

Dave is well qualified to lead the MSRC through this transition and continue ERDC's leadership role in the Program. I wish him and the ERDC MSRC the best of luck.

John West

Director
Scientific Computing Research Center

About the Cover: The cover shows a numerical simulation of a free surface flow around a naval ship, including plunging and spilling breaking waves, formation of spray and entrainment of air. Insets show the isosurface colored by velocity (see article, page 2). The cover design is by the Unclassified Data Analysis and Assessment Center.

Contents

from the director . . .

A Numerical Wave Tank

By Dr. Douglas G. Dommermuth and Thomas T. O’Shea, Science Applications International Corporation (SAIC); Dr. Kelli Hendrikson, Massachusetts Institute of Technology (MIT); Donald C. Wyatt, SAIC; Dr. Mark Sussman, Florida State University; Gabriel D. Weymouth and Dr. Dick K. P. Yue, MIT; and Paul Adams and Miguel Valenciano, ERDC MSRC..... 2

ERDC MSRC Cray XT3 System Greatly Increases Army Explosive—Concrete Modeling Capabilities

By Dr. Kent T. Danielson, Army High Performance Computing Research Center and ERDC Geotechnical and Structures Laboratory (GSL); and Dr. James L. O’Daniel, Dr. Mark D. Adley, Dr. Stephen A. Akers, and Sharon B. Garner, ERDC GSL..... 6

ezVIZ Batch Reaches v1.3

By Randall E. Hand 11

Simulation of Engine Ground Vortex Control

By Drs. Arvin Shmilovich and Yoram Yadlin, The Boeing Company, Huntington Beach, CA 14

DoD Visualization Has a New Place on the Internet

By Paul Adams 17

ERDC MSRC Transitions New Visualization Hardware to Data Center

By Paul Adams 18

ERDC Infrastructure Takes a New Spin

By Greg Rottman and Paula Lindsey 19

ERDC Adds over 12,000 Processors to Offering

By Jay Cliburn 20

Reconfigurable Computers

By Dr. Gerald R. Morris 22

visitors 28

acronyms 31

training schedule 32

A Numerical Wave Tank

By Dr. Douglas G. Dommermuth and Thomas T. O'Shea, Science Applications International Corporation (SAIC); Dr. Kelli Hendrikson, Massachusetts Institute of Technology (MIT); Donald C. Wyatt, SAIC; Dr. Mark Sussman, Florida State University; Gabriel D. Weymouth and Dr. Dick K. P. Yue, MIT; and Paul Adams and Miguel Valenciano, ERDC MSRC

The traditional approach for predicting the performance of ships is to perform laboratory experiments in a wave tank. As a model of a ship is towed down a tank, various quantities such as forces and free-surface disturbances are measured. The laboratory experiments are difficult to perform, labor intensive, and expensive. Computational fluid dynamics (CFD) has recently become a useful alternative for simulating the flow around naval combatants. The ultimate goal in the field of numerical free-surface hydrodynamics is to develop a numerical wave tank. A step toward realizing this goal is embodied in the NFA (Numerical Flow Analysis) computer code, which is a CFD capability for simulating the free-surface flow around ships.

The primary objective of NFA is to provide a turnkey capability to the Navy for simulating free-surface flows. The turnkey aspects of NFA are ease of input, ease of use, and numerical robustness in combination with the ability to simulate a complex range of physical phenomena, including the breaking of waves, the entrainment of air, and the formation of spray. These qualities of NFA are made possible by using a Cartesian-grid formulation to impose boundary conditions on the ship hull and a volume-of-fluid (VOF) method to track the free-surface evolution.

A Cartesian-grid formulation permits an efficient method for representing and modeling a ship's geometry. A two-dimensional CAD representation of the ship hull is used to specify the geometry. Once the CAD geometry is imported into NFA, the ship hull is represented internally within NFA as a signed-distance function. The distance from a grid point to the ship hull is positive outside the ship hull and negative within the ship hull. A distance equal to zero constitutes the ship hull. This signed-distance function representation is used to calculate how the ship hull cuts the Cartesian grid. Once the intersection of the ship hull with the Cartesian grid is known, the governing equations are discretized using a finite-volume method. Since CAD representations of a ship's hull are necessary throughout the design process for basic ship-design calculations, NFA does not require any additional input over what is already available. Conventional body-fitted grids require much more time and expertise to create than NFA's geometry input. In addition, NFA's Cartesian-grid formulation provides better numerical conditioning than a body-fitted formulation. As a result, NFA

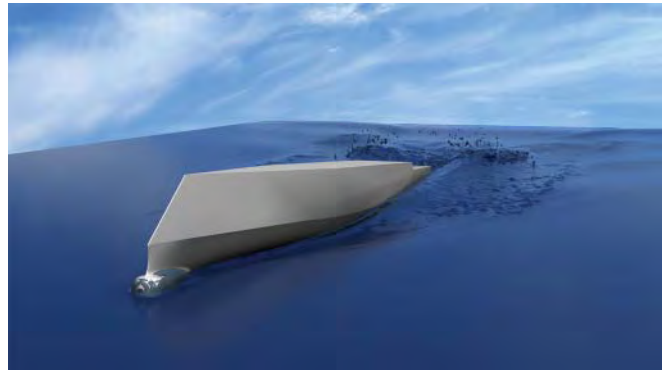


Figure 1. 5613 ship configuration at full pitch up

simulations are less prone to numerically break down than techniques that use body-fitted grids.

Signed-distance functions are capable of representing ship hulls that are arbitrarily complex as easily as ship hulls with simple geometries. An equally powerful technique is required to represent the free surface, which is the boundary that separates water from air. Some complex features of the free-surface boundary include the surface of breaking waves, the pockets of air that are entrained by wave breaking, and the sheets of spray that form near the bow and stern of the ship. A separate issue from the topology of the free surface is the evolution of the free surface. The primary concerns for free-surface evolution are accuracy and mass conservation. The VOF formulation that is used in NFA is capable of evolving free surfaces with high accuracy and good mass conservation. The VOF technique uses volume fractions to represent the portion of a grid cell that is filled with water. A volume fraction that is equal to one means that the cell is totally filled with water, and a volume fraction that is zero means that the cell is filled with air. Intermediate values of the volume fraction mean that the cell is partially filled with water. Rider et al. (1995) provide additional details of the VOF formulation. The blend of Cartesian-grid and VOF methods provides a powerful formulation for simulating ship flows.

The governing equations that are solved in NFA are for an incompressible fluid. Following Puckett et al. (1997) and Dommermuth et al. (2006), a cut-cell method is used to enforce free-slip boundary conditions on the hull. A second-order, variable-coefficient Poisson equation is used to project the velocity onto a solenoidal field, thereby ensuring mass conservation. A preconditioned conjugate-gradient method is used to solve the Poisson equation. Details of a similar projection

operator are provided in Puckett et al. (1997). The solution to the Poisson equation is the most computationally intensive portion of NFA's formulation. The convective terms in the momentum equations are accounted for using a slope-limited, third-order QUICK scheme as discussed in Leonard (1997). The governing equations are solved using a domain decomposition method. Communication between processors on the Cray XT3 is performed using Cray's shared memory access library. The central processing unit (CPU) requirements are linearly proportional to the number of grid points.

NFA has benefited from Challenge Project status for 10 years. The support of the Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP) has greatly accelerated the development of NFA such that the goal of developing a numerical wave tank is being realized. Recent applications of NFA include the following numerical studies: (1) two littoral combat ship (LCS) hulls moving with constant forward speed; (2) forced motion studies of two naval combatants (models 5514 and 5613—see Figure 1) with and without forward speed and ambient waves; (3) a validation study of a patrol gunboat (model 5365) moving at two speeds in calm water; (4) bow- and stern-wave studies of a DDG (model 5415); (5) parametric studies of transom-stern vessels; and (6) validation studies of a sphere impacting a free surface. All of these numerical simulations have been performed on the Cray XT3 at the U.S. Army Engineer Research and Development Center (ERDC) Major Shared Resource Center (MSRC). Selected animations of numerical simulations are available at www.saic.com/nfa. Most of the animations at this site have been prepared by the Scientific Visualization Center at ERDC.

Wilson et al. (2006) compare the results of numerical simulations using seven computer codes with experimental measurements of a model ship towed at constant forward speed in a wave tank. The results for a low and high speed are reported. The numerical simulations had been performed blind to ensure a fair assessment of current capabilities to predict flow around naval vessels. Wilson et al. (2006) conclude that NFA's numerical predictions are good in the bow region, good to excellent in the stern region, and very good along wave cuts off the body. Two issues not addressed in Wilson et al. (2006) are ease of usage and numerical stability. These are two of NFA's strengths because of its Cartesian-grid formulation.

The NFA simulations use $680 \times 192 \times 128 = 16,711,680$ grid points, $4 \times 8 \times 4 = 128$ subdomains, and 128 nodes

on a Cray XT3. The length, width, depth, and height of the computational domain are respectively 3.0, 1.0, 1.0, 0.5 ship lengths (L). Grid stretching is employed in all directions. The smallest grid spacing is $0.002L$ near the ship and mean waterline, and the largest grid spacing is $0.02L$ in the far field. The numerical simulations run 12,001 time-steps corresponding to six ship lengths. They each require 50 hours of wall-clock time.

Figure 2 shows wave cuts for the 10.5 knot case. The correlation coefficients between experimental measurements and numerical predictions for parts *a* through *d* of Figure 2 are 0.89, 0.91, 0.85, and 0.86, respectively. The solid and dashed lines respectively denote the experimental measurements and the numerical predictions. The correlation gets poorer in the region where the grid spacing along the *y*-axis gets poorer. The numerical simulations do not resolve the shortest waves, and more grid resolution is required. Convergence studies are in progress.

Figure 3 shows wave cuts for the 18 knot case. The correlation coefficients between experimental measurements and numerical predictions for parts *a* through *d* of Figure 3 are 0.89, 0.92, 0.88, and 0.91, respectively. In general, the high-speed simulation is in slightly better agreement with the experimental measurements than the low-speed simulation, probably because the

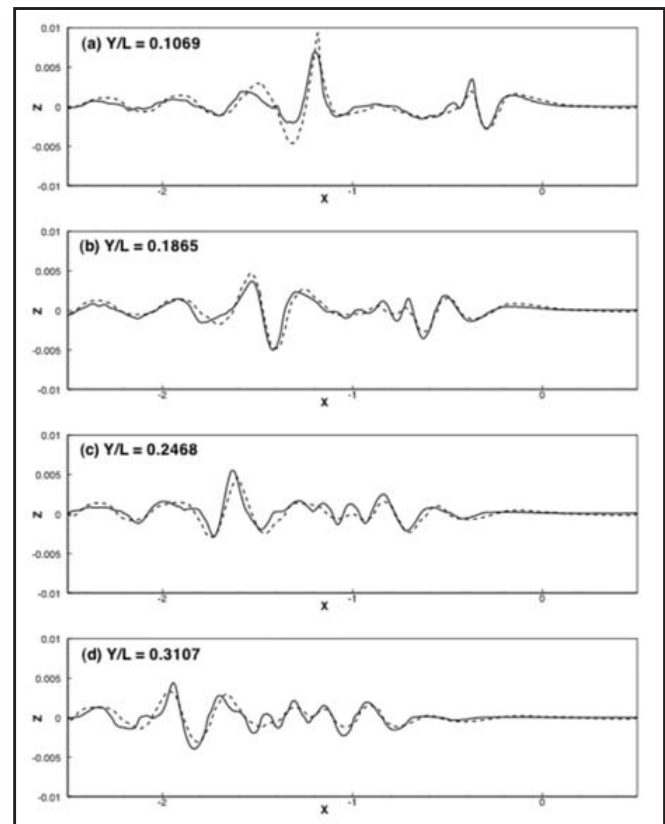


Figure 2. Wave cuts, Athena hull form, 10.5 knots

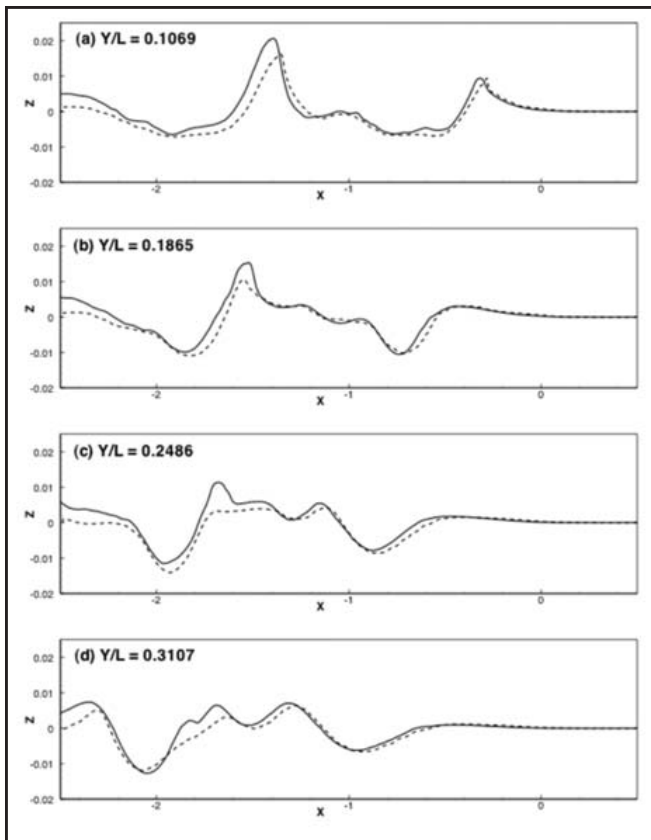


Figure 3. Wave cuts, Athena hull form, 18 knots

waves are longer. However, both simulations would benefit from using higher resolution, especially near the bow and transom where there is wave breaking and flow separation.

Experimental measurements of a DDG Model 5415 have been performed at an equivalent full-scale speed of 30 knots (see <http://www.dt.navy.mil/hyd/sur-shi-mod/>). The NFA simulation uses $800 \times 192 \times 192 = 9,491,200$ grid points, $4 \times 8 \times 8 = 256$ subdomains, and 256 nodes on a Cray XT3. The length, width, depth, and height of the computational domain are respectively 3.0, 1.0, 1.0, 0.5 ship lengths (L). Grid stretching is employed in all directions. The numerical simulation uses 28,000 time-steps to run the equivalent of 5.6 ship lengths. It requires 125 hours of wall-clock time.

Figure 4 compares NFA predictions with experimental measurements for the flow near the bow. The free-surface measurement along the ship hull is denoted by spherical symbols along the ship hull. The upper bounds of the free-surface measurements provided by whisker probes are indicated by the small spherical symbols transverse to the ship. NFA prediction of the surface is the color contour. The whisker-probe measurements agree well with the upper bound of the NFA free-surface predictions. NFA correctly predicts the overturning of the bow wave and the resulting splash

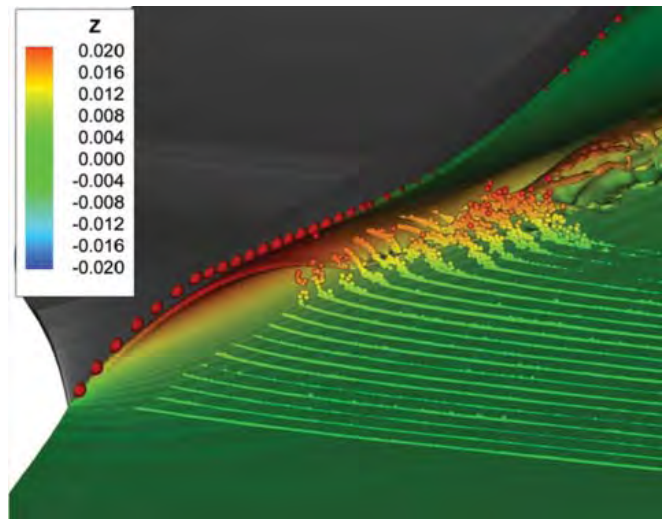


Figure 4. Wave cuts near bow for model 5415

up slightly aft of the stem. Where the very thin sheets that characterize the bow stem run-up are not resolved, the predictions are slightly lower than the measured profile.

Figure 5 compares NFA predictions to whisker-probe measurements for the flow near the stern. The portion above the centerline of the ship represents NFA results, while the portion below is based on experiments. Black lines mark the edges where spilling occurs. NFA accurately captures the flow separation from the transom stern, and agreement between predictions and measurements is good overall. However, at this resolution some spilling along the edges of the rooster tail is not captured. As a result, the predicted rooster-tail amplitude directly astern of the transom is higher than measurements.

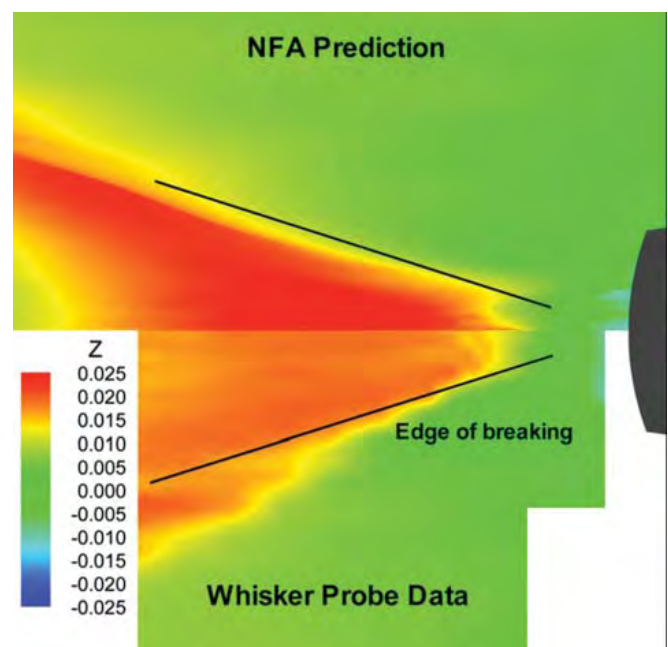


Figure 5. Wave elevations near stern for model 5415

Simulations that resolve the breaking in this region may provide the dissipation of energy that is necessary to reduce the wave amplitude in the rooster-tail region.

Figure 6 shows transverse cuts of the free-surface elevation near the bow. The hull cross section is colored gray. Circular symbols denote the profile measurements along the side of the hull. Solid lines denote whisker-probe measurements. Dashed lines denote NFA predictions. Results are shown for various stations aft of the bow from (a) $x = 0.107L$ to (d) $x = 0.133L$.

The figures show the overturning of the bow wave. The initial onset of air entrainment is evident in the NFA predictions. Even further aft, the plunging portion of the wave impacts the free surface and splashes upward. As expected, the whisker-probe measurements provide an upper envelope to the numerical predictions.

As improvements have been made in NFA's formulation over the years, computer hardware has increased significantly in power. In particular, the Cray XT4 will enable us to model the flow around a ship down to 5-10 cm. This parameter regime is important because it is the upper range of turbulent breakup of the free surface. As a result, the numerical simulations will resolve the onset of spray formation in the transom and bow regions and onset of spilling along the cusps of steep waves. Model-scale experiments in a wave tank are difficult to perform in this parameter regime because scaling issues strongly affect the free-surface portion of the flow. Together, these advances in software and hardware now provide a unique position to contribute to the design capabilities of the Navy.

Acknowledgments

The Office of Naval Research supports this research under contract number N00014-07-C-0184. Dr. Patrick Purtell is the program manager. The David Taylor Model Basin under the guidance of Dr. Arthur Reed also supports this research. This work is supported in part by a

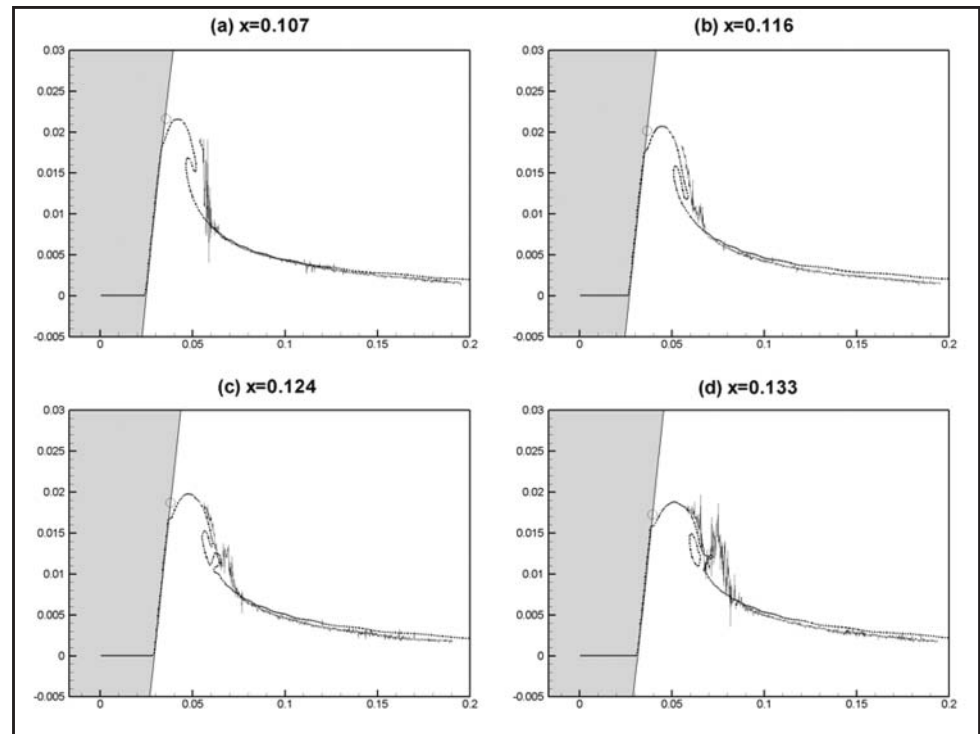


Figure 6. Transverse wave cuts near bow for model 5415

grant of computer time from the DoD HPCMP (<http://www.hpcmo.hpc.mil/>). The numerical simulations have been performed on the Cray XT3 at ERDC. The support of the Scientific Visualization Center at ERDC under the guidance of Paul Adams is gratefully acknowledged.

References

- Dommermuth, D.G., O'Shea, T.T., Wyatt, D.C., Sussman, M., Weymouth, G.D., and Yue, D.K.P. (2006) Numerical simulation of ship waves using Cartesian-grid and volume-of-fluid methods. In *Proc. 26th Symp. On Naval Hydro.*, Rome, Italy, To appear.
- Leonard, B.P. (1997) Bounded higher-order upwind multidimensional finite-volume convection-diffusion algorithms, *Advances in Numerical Heat Transfer*, edited by W.J. Minkowycz, W.J., and E.M. Sparrow, Taylor & Francis, Washington, D.C., 1-57.
- Puckett, E.G., Almgren, A.S., Bell, J.B., Marcus, D.L., Rider, W.J. (1997) A second-order projection method for tracking fluid interfaces in variable density incompressible flows. *J. Comp. Physics*, **130**, 269-282.
- Rider, W.J., Kothe, D.B., Mosso, S.J., Cerutti, J.H., and Hochstein, J.I. (1994) Accurate solution algorithms for incompressible multiphase flows. *AIAA paper 95-0699*.
- Wilson, W., Fu, T.C., Fullerton, A., and Gorski, J. (2006) The measured and predicted wave field of model 5365: An evaluation of current CFD capability. In *Proc. 26th Symp. On Naval Hydro.*, Rome, Italy, To appear.

ERDC MSRC Cray XT3 System Greatly Increases Army Explosive—Concrete Modeling Capabilities

By Dr. Kent T. Danielson, Army High Performance Computing Research Center and ERDC Geotechnical and Structures Laboratory (GSL); and Dr. James L. O'Daniel, Dr. Mark D. Adley, Dr. Stephen A. Akers, and Sharon B. Garner, ERDC GSL

Introduction

The large number of fast processors on the new Cray XT3 at the ERDC MSRC has tremendously improved the ability of researchers in the ERDC Geotechnical and Structures Laboratory (GSL) to model complex weapon interactions with reinforced concrete structures. Analyses that took many CPU days on the previous Compaq and Origin systems can now be performed in about half an hour. These analyses comprise the largest yet practical utilization of Cray XT3 resources at the ERDC MSRC (up to 4,096 processors).

A heightened threat to civil infrastructure, Government facilities, and military installations has led to an increased use of numerical simulations to evaluate their vulnerabilities to explosive detonations. Computer modeling is especially attractive for such applications, since full-scale destructive testing on large structures is infeasible. The authors have recently performed numerous simulations of this type for various DoD and other U.S. Government agencies that have utilized improved predictive capabilities of the microplane material model (Bažant et al. 1996, 2000) compared with other constitutive models. The microplane concrete model, developed jointly at Northwestern University and GSL, is a precursor for multiscale models, as it projects macroscale strains onto microplanes to use simpler more fundamental relations and then brings these results back to the macroscale with a thermodynamically consistent homogenization procedure. This “semi-multiscale” method has been proven to be an accurate, reliable, and robust constitutive relation for concrete subjected to blast loadings, but it can be nearly an order of magnitude more computationally intensive than other inelastic models. By making the analysis times more reasonable, the large Cray XT3 thus greatly allows analysts to more fully exploit this technology for larger and more complete structural applications than ever before.

The analyses of blast loading events on concrete structures were performed with the parallel explicit dynamic finite element code, *ParaAble*, developed by the authors (Danielson et al. 2000; Danielson and Namburu 1998). *ParaAble* is a transient Lagrangian solid dynamics program for three-dimensional large

strain/deformation problems with nonlinear boundary conditions. Finite elements are used for spatial modeling, and time is integrated by an explicit central difference scheme. The code was designed to perform large-scale analyses and to execute all of its capabilities on parallel computing platforms. The parallel development of the code has a similar structure to other parallel explicit dynamic codes, e.g., Hoover et al. 1995; Plimpton et al. 1996. A Single Program Multiple Data (SPMD) paradigm is used with the code written in FORTRAN 95, and all interprocessor communication can be made with explicit Message Passing Interface (MPI) calls or with a hybrid MPI/SHMEM option. *ParaAble*'s capabilities fall within those of other typical explicit dynamic/hydrocodes for complex solid mechanics applications (e.g., EPIC, DYNA3D, ABAQUS/Explicit, and PRONTO 3D) and contain many different options commonly available in other popular finite element codes (e.g., keyword command syntax, various material models and loading types, multipoint constraints, failure/erosion, and restart capabilities). By its modular nature, *ParaAble* has the hooks to easily implement new capabilities. It also contains a simple interface to rapidly implement constitutive models with a wide variety of popular strain and strain rate formulations.

The parallel procedure primarily consists of a mesh partitioning preanalysis phase, a parallel analysis phase that includes explicit message passing among each partition on separate processors, and a postanalysis phase to gather separate parallel output files into a single coherent database. Each partition can also be optionally placed on individual cores of multicore processors to further exploit this new parallelism of modern chip architectures. A material-weighting scheme using METIS (Karypis and Kumar 1995a, 1995b) was developed for parallel analysis involving multiple material types. Since the microplane model is typically much more computationally intensive than other inelastic models, drastic computational load imbalances among processors may occur when used with other materials. The weighted partitioning scheme alleviates this problem, and the use of large-scale parallel computing demonstrates the ability to reasonably perform such analyses for production purposes.

Interprocessor communication can be made entirely with MPI calls, or optionally for platforms with SHMEM, a hybrid is available that uses MPI for the minor problem setup/cleanup calls and SHMEM being used for the primary communications during the time integration loop. Experience has shown that this hybrid is essentially about as effective as MPI alone, but SHMEM has sometimes been more stable and efficient in early platform releases. Using the Catamount Virtual Node (CVN) capability on the Cray XT3 platform, each partition can also be optionally placed on individual cores of their dual-core processors for further parallelism. Although each partition/core will have less than half the memory and cache of the full processor, this approach has the potential to reduce CPU times by half. Scalable I/O is performed by using separate files (input, output, restart, etc.) for each partition. In addition to the preanalysis mesh partitioning tools, accompanying software was also developed for postanalysis assemblage of separate partition output files when necessary. Although *ParaAble* itself had no limitations for the large numbers of processors on the XT3, extension from hundreds to now thousands of processors did pose new difficulties for the system. Several minor rewrites were necessary, as certain collectives and “ALL” procedures had to be changed for the large numbers of processors. Whereas the problems may appear to be a result of nonconformance with MPI/SHMEM standards, it can be equally argued that the standards did not adequately anticipate these large processor difficulties when they were written (as the *ParaAble* developers did not).

Numerical Applications

Eight-noded hexahedral elements (Flanagan and Belytschko 1981) are exclusively used for all geometric modeling. Steel reinforcement is modeled with multiple elements through the thickness of each rebar using the Johnson-Cook viscoplastic model (Johnson and Cook 1983). Breakage of rebar elements is performed by erosion of elements to ensure that the restraint of the rebar on the concrete can be dissipated. High-explosive materials are modeled with a JWL equation-of-state, and ignition of the explosive is treated by a programmed-burn algorithm (e.g., Taylor and Flanagan 1989).

Charge Detonation in a Reinforced Concrete Wall

This first example provides a benchmark to gauge the confidence in the modeling accuracy for these types of applications as well as the reasonableness of performing the computations. The simulated cylindrical C-4 charge detonation in a reinforced concrete wall is depicted by the deformed finite element model in Figure 1a, which consists of 995,192 hexahedral elements and 1,030,890 nodes. The event was experimentally staged at ERDC. Quarter symmetry was assumed for the simulation, and the transient analysis was performed to 1 millisecond. The damaged portion of the wall was predicted with the pressure dependent-effective inelastic strain damage model implemented in conjunction with the microplane model. The damaged portion, which was determined from solely postprocessing the damage evolution data, is depicted in Figure 1 and compares

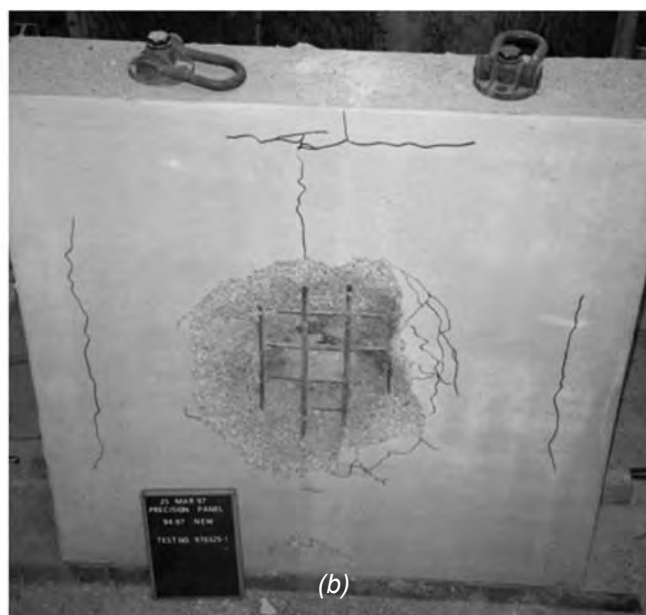
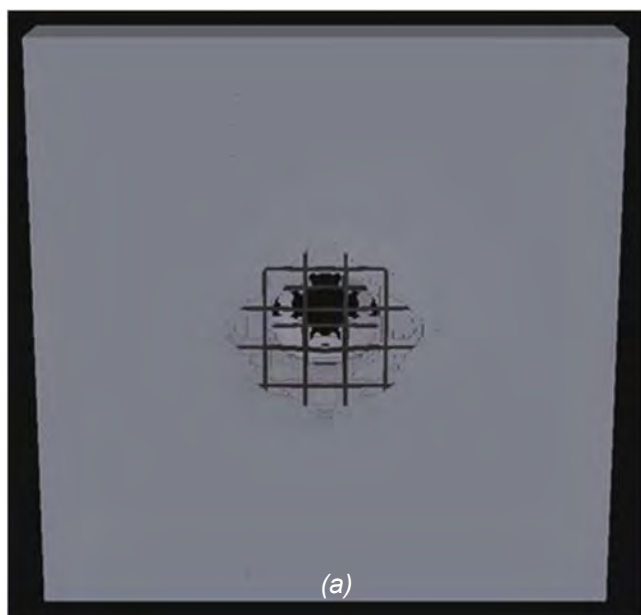


Figure 1. Embedded C-4 explosive detonation in a reinforced concrete wall. (a) *ParaAble* finite element predictions; (b) test performed at ERDC

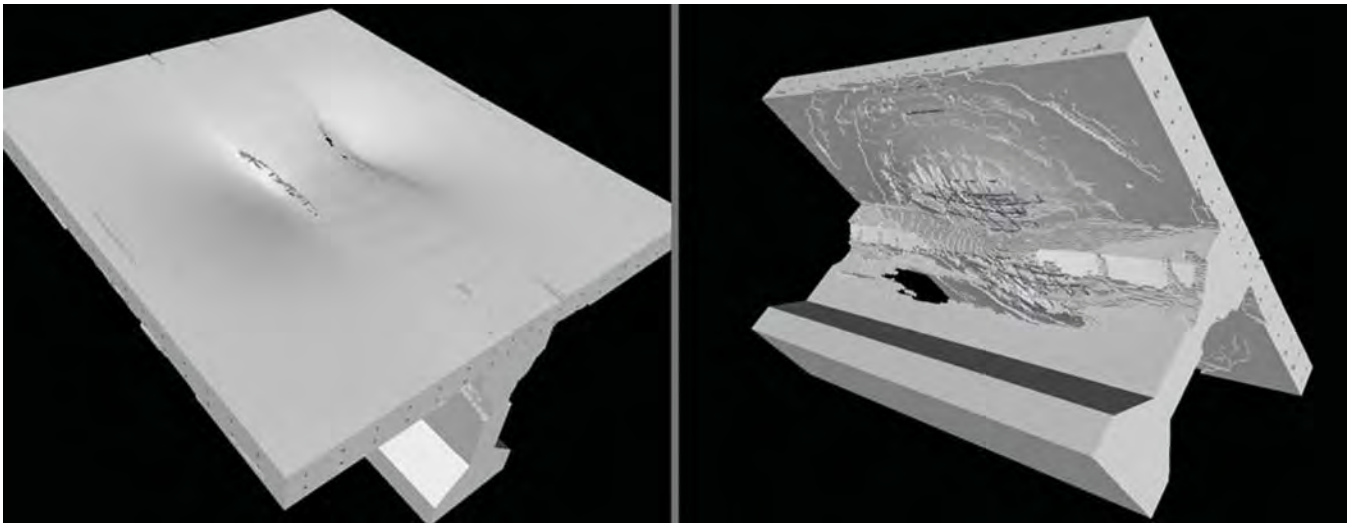


Figure 2. Finite element simulation of a bridge deck and girder assembly subjected to a truck bomb detonation

favorably with the experimental observations in Figure 1b—including the level of damage attained by the reinforcement. The scalability was excellent, as the analysis required approximately 8, 2, and 1 CPU hours on 8, 32, and 64 processing elements (PEs), respectively. Despite consisting of a fairly large number of elements, particularly with the microplane model, the simulation can be performed with a reasonable turn-around time on a small to moderate number of processors.

Truck Bomb Detonation on a Bridge Deck and Girder Assembly

This example is the simulation of a reinforced concrete bridge deck atop a prestressed concrete girder assembly subjected to a truck bomb detonation. These types of simulations were used for vulnerability assessments and improved design and retrofit concepts. The deformed finite element model, shown in Figure 2, consists of approximately three million elements and nodes. For a spectrum of bomb sizes, the BLASTX code (Britt et al. 2001) was used to predict pressure histories that were applied to the deck as radial varying concentric surface tractions. To represent the restraint of the rest of the bridge deck, the top boundary nodes were fixed in the lateral directions with the appropriate restrained boundary conditions. Whereas this problem is larger than the previous example, a three millisecond simulation used only about 1.5 CPU hours on 256 processors of the Cray, with the excellent scalability shown in Figure 3. The fully damaged concrete elements, which again are determined from solely post-processing the damage evolution data, are removed from the views depicted in Figure 2.

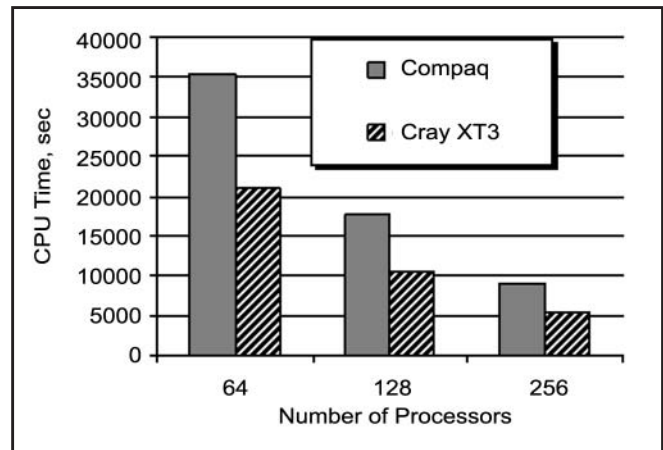


Figure 3. Parallel performances of bridge deck blast analysis on Compaq AlphaServer SC45 and Cray XT3 platforms

Charge Detonation in a Reinforced Concrete Bridge Pier

This final example is a much larger simulation than the previous ones, and it compares the MPI and hybrid MPI/SHMEM implementations. It models the detonation of a cylindrical explosive charge embedded in a reinforced concrete bridge pier. Since symmetry is never guaranteed in nonlinear analyses, the deformed finite element model shown in Figure 4 did not contain any symmetry assumptions; the model consists of nearly 15 million elements and nodes. The bottom boundary nodes at the pier foundation were fixed. To represent the restraint of the bridge deck, the top boundary nodes were fixed in the lateral directions, and the dead weight of the bridge span was applied in a normal manner to the top surface of the pier. The

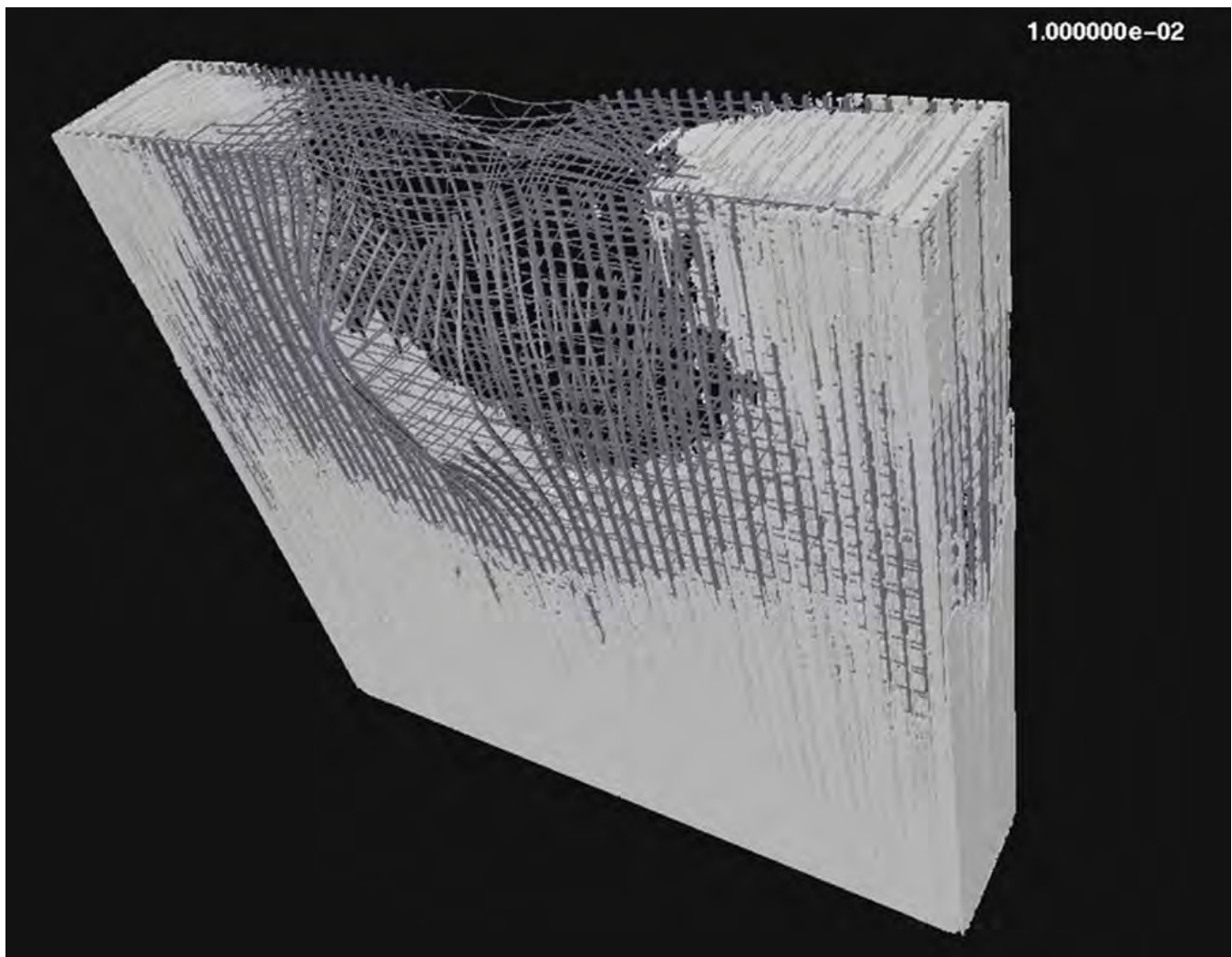


Figure 4. Finite element simulation of pier damage resulting from internal explosive detonation at 10 milliseconds

predicted level of damage sustained by the bridge pier is depicted in Figure 4, which again is determined from solely postprocessing the damage evolution data for the concrete, the erosion failure threshold of the steel, and by eliminating the explosive elements with nearly zero pressure (all of them). The transient analysis was performed to 10 milliseconds, and once again, a very large microplane model analysis was reasonably performed with the utilization of parallel computing. Results of the performance on the XT3 system are provided in Figure 5, whereby the scalability is excellent. Figure 5 indicates that both the MPI and hybrid MPI/SHMEM implementations perform almost equally, with MPI being only slightly better.

Concluding Remarks

The microplane constitutive model, a semi-multiscale model that is effective for predicting complex nonlinear behavior of high-explosive-reinforced concrete interaction, is implemented into an MPI- and MPI/

SHMEM-based massively parallel finite element code. The examples demonstrated that the code was portable from commodity Linux clusters to the XT3, but several minor rewrites were necessary to apply it to thousands of processors. The ERDC MSRC Cray XT3 was shown to be invaluable for very large-scale applications of this type, as analyses that would require hundreds to thousands of serial computing hours were performed in a few hours or minutes. With the aid of high performance computing, the viability of these types of analyses can thus be greatly extended, particularly for large-scale analysis of this type.

Acknowledgments

The research reported in this article was performed in connection with contract/instrument DAAD19-03-D-0001 with the U.S. Army Research Laboratory. The views and conclusions contained in the article are those of the authors and should not be interpreted as presenting the official policies or positions, either

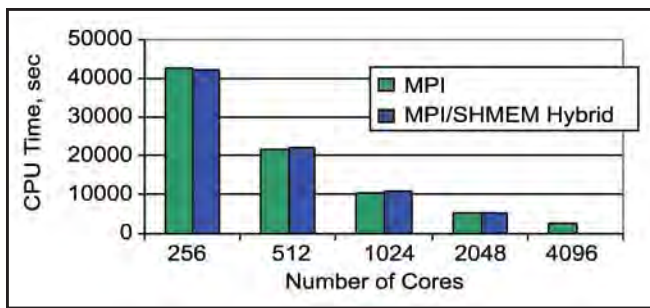


Figure 5. ParaAble scalability on the Cray XT3

expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer names or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. Permission to publish was granted by the Director of the GSL.

References

- Ba•ant, Z.P., Caner, F.C., Carol, I., Adley, M.D., and Akers, S.A. (2000). "Microplane Model M4 for Concrete. I: Formulation with Work-Conjugate Stress." *J. Engrg. Mech.*, 126(9) 944-953.
- Ba•ant, Z.P., Xiang, Y., Adley, M.D., Prat, P.C., and Akers, S.A. (1996). "Microplane Model for Concrete. I: Stress-Strain Boundaries and Finite Strain; II: Data Delocalization and Verification." *J. Engrg. Mech.*, 122(3) 245-262.
- Britt, J.R., Ranta, D.E., and Joachim, C.E. (2001) "BLASTX Code, Version 4.2, User's Manual," ERDC/GSL TR-01-2, U.S. Army Engineer Research and Development Center, Vicksburg, MS.
- Danielson, K.T., Uras, R.A., Adley, M.D., and Li, S. (2000). "Large Scale Application of Some Modern CSM Methodologies by Parallel Computation." *Advances in Engineering Software*, 31(8-9) 501-509.
- Danielson, K.T. and Namburu, R.R. (1998). "Nonlinear Dynamic Finite Element Analysis on Parallel Computers Using FORTRAN 90 and MPI." *Adv. Engrg. Software*, 29(3-6) 179-186.
- Flanagan, D.P. and Belytschko, T. (1981). "A uniform strain hexahedron and quadrilateral with orthogonal hourglass control." *Int. J. Numer. Meth. Engrg.*, 17, 679-706.
- Johnson, G.R. and Cook, W.H. (1983). "A constitutive model and data for metals subjected to large strains, high strain rates, and high temperatures." *Proceedings of seventh international Symposium on Ballistics*. The Hague, Netherlands.
- Hoover, C.G., DeGroot, A.J., Maltby, J.D., and Procassini, R.J. (1995). "ParaDyn - DYNA3D for massively parallel computers." *Engineering, Research, Development and Technology FY94, Lawrence Livermore National Laboratory, UCRL 53868-94*. Livermore, CA.
- Karypis, G. and Kumar, V. (1995a). "A fast and high quality multilevel scheme for partitioning irregular graphs." Technical Report TR 95-035, Department of Computer Science, University of Minnesota. Minneapolis, Minnesota.
- Karypis, G. and Kumar, V. (1995b). "A Parallel Algorithm for Multilevel Graph Partitioning and Sparse Matrix Ordering." TR 95-036, Department of Computer Science, University of Minnesota. Minneapolis, Minnesota
- Plimpton, S., Attaway, S., Hendrickson, B., Swegle, J., Vaughan, C., and Gardner, D. (1996). "Transient dynamics simulations: parallel algorithms for contact detection and smoothed particle hydrodynamics." *Proceedings of SuperComputing 96*. Pittsburgh, PA.
- Taylor, L.M. and Flanagan, D.P. (1989). "PRONTO 3D: A three-dimensional transient solid dynamics program." Report SAND 87-1912, Sandia National Laboratories, Albuquerque, NM.

ezVIZ Batch Reaches v1.3

By Randall E. Hand

Statistics as of March 29, 2007

Number of Runs: 229,547

CPU Hours: 10,082

56.05 Tera-points processed

55.83 Tera-cells processed

What is ezVIZ?

Gaining useful information from data sets is a challenging task. This task often falls to the researcher or a visualization scientist. Extracting insight from a multiterabyte data set presents the researcher with several problems. These problems include transfer and storage of the data, graphics hardware to visualize it, as well as having visualization software capable of handling the data.

ezVIZ tackles the visualization problems of the researcher by providing one of two mechanisms. The first, which is currently available to users, is a batch visualization capability. This batch capability allows the users to create images from their data while it still resides on the supercomputer. These images, which are less than a few megabytes in size, can then be moved with ease to the researcher's workstation. Storage and network bandwidth are no longer a concern when visualizing the data. The second mechanism is to provide a Web interface to visualizing the data. This mechanism is currently under development.

ezVIZ Today

ezVIZ Version 1.3 has just been released with several bugfixes and a few new features not available in previous versions.

- New VTK support with better memory management and improved stability
- Newer version of Mesa improves rendering times
- Newer version of NetCDF improves Quoddy format support
- Improved documentation
- Improved makefiles for easier installation
- Much more "under the hood"

ezVIZ Deployments

ezVIZ is currently available on the following systems:

- (ERDC) Sapphire - Available on all login nodes
- (ERDC) Ruby - Available on all nodes
- (ERDC) Amethyst - Available on all nodes
- (ERDC) Plasma - Available on all nodes
- (NAVO) Babbage - Available on all nodes

ezVIZ is also available on selected workstations dedicated to visualization functions. Also, as an open-source project, the source code is available to all who wish to install it locally themselves.

ezVIZ Documentation

The documentation has also greatly improved since v1.2. There are now many options available to users for support and information, thanks to the new Unclassified Visualization Web site (refer to the article in this issue of the *Resource* for more information). The following is available to all users:

- Web Forum - Post questions to MSRC staff and other users to get answers to your questions
- Wiki - Review the online ezVIZ Batch page for tutorials and help, and even contribute your own for other users
- Script Generator - a Web-based tool for generating ezVizGeneric Scene Scripts
- News - Stay up to date on new features in ezVIZ and new versions

ezVIZ Case Studies

To wrap up this article, I've asked a few of ezVIZ's most prolific users to share their experiences for inclusion in this article. Read ahead for their impressions of ezVIZ and how it has helped them see their data in ways they have never been able to before.

Volume Rendering of Gravity Waves

By Ling Wang et al., Colorado Research Associates Division of NorthWest Research Associates

In this project, we perform direct numerical simulations (DNS) of atmospheric gravity wave breaking and turbulence generation and dissipation using an incompressible spectral code optimized for performance on various supercomputer architectures. The simulations were performed with resources provided as part of a DoD HPCMP “Challenge” resource allocation. A typical simulation creates ~ 10 to 20 TB of raw data, so volume rendering is one of the indispensable techniques to visualize the results.

As an example, Figure 1 shows volume rendering images of the λ_2 parameter at selected model times for a simulation with Reynolds number Re_0 of 10,000 and the initial nondimensionalized gravity wave amplitude of 1.1. A maximum resolution of $(N_x, N_y, N_z) = (2400, 1600, 800)$ was required to resolve the viscous scale for this simulation. λ_2 is the second eigenvalue of the symmetric tensor of the velocity gradient and is a very good parameter to identify vortex structures in turbulent flows. The left and right panels of the figure show the rendering images viewed from the side (or streamwise-vertical) and top (or streamwise-spanwise), respectively, of the computational domain. The boxes in the left panels are tilted by $\sim 18^\circ$, as the computational domain is aligned along the phase of the gravity wave that is not horizontal. The λ_2 rendering images clearly show the process of gravity wave breaking, the resulting turbulence, the dissipation of turbulence, and the excitation of secondary waves. Indeed, movies of λ_2 have been made from λ_2 rendering image at each model time (not shown), and they prove to be highly valuable and indispensable in unveiling the characteristics of the evolution of turbulent flows of different scales and the interactions among them.

In this project, the volume rendering images were all created using ezVIZ, which is versatile and very convenient to use. For each three-dimensional (3-D) data set, we first created an ezVizGeneric “scene” file, which was adapted from samples provided by Randall Hand and Paul Adams from the ERDC MSRC. The scene file includes the basic information on the 3-D data to be rendered, the output, and the colormap and opacity map. We then used the relevant ezVizGeneric commands to create the rendering images. For example, we used

```
ezVizGeneric SceneFile -yaw -270 -output ImageFile  
and
```

```
ezVizGeneric SceneFile -roll -18.44 -output ImageFile
```

to create the images in the left and right panels of Figure 1, respectively. In practice, we wrote a script to automate the above procedure to create scene files specific to a

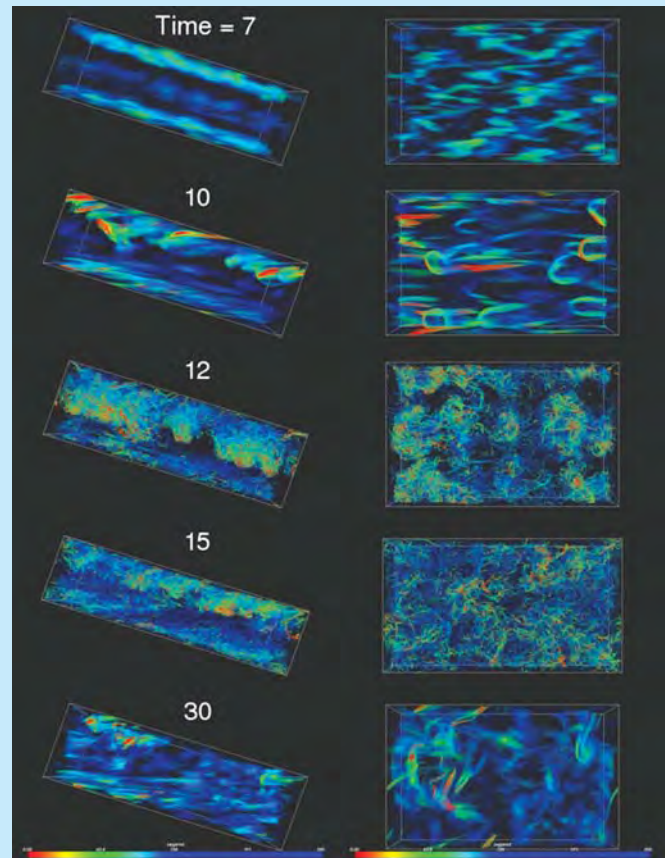


Figure 1. Volume rendering images of the λ_2 parameter at times chosen to represent different stages of gravity wave breaking and turbulence evolution. Left panels show the side of the computation domain; right panels show the top

particular data set and to submit the ezVizGeneric jobs to the queue system on the Cray XT3 (Sapphire) and SGI Origin (Ruby) machines, both located at the ERDC MSRC. Tens of thousands of such volume rendering images were created, and the images were then combined to create movies. From our experience, ezVIZ is rather efficient in dealing with data of very large size, and the memory requirement is generally low. A 3-D byte-scale data set of size $(2400, 1600, 800)$, for example, took a few minutes to render using ezVIZ on Sapphire. ezVIZ is highly flexible when dealing with different types of input data, positioning the camera (e.g., making “flyarounds”), and tuning the colormap and opacity map to produce high-quality images. Finally, the ezVIZ scripts and commands can be easily added to existing postprocessing scripts to automate the data visualization task.

Visualizing the Earth's Bow Shock

By Sam Cable, Computational Science and Engineering Group, ERDC MSRC

I am running magnetohydrodynamic (MHD) simulations of the interaction between the solar wind and the Earth's magnetic field. The Earth's magnetic field presents an obstacle to the solar wind, and like a supersonic aircraft flying through the air, it creates a shock wave in the solar wind called the "terrestrial bow shock." MHD fluid flow is more complicated than atmospheric fluid flow in that MHD supports three distinct waves, as opposed to air, which supports only one sound wave. I am studying a particular situation where the terrestrial bow shock is made up of regions dominated by two different sorts of shock waves. ezVIZ was instrumental in helping me find economical ways to display the relative importance of the different shock waves at different locations on the bow shock. In Figure 1, I used ezVIZ to plot the surface of the bow shock and then "painted" the surface with colors representing the difference in flow speed and a particular type of wave speed. Where the surface shows up blue, the bow shock is dominated by typical MHD "fast" shock waves; where it shows up red, it is dominated by unique MHD "intermediate" shock waves. The same data are presented differently in Figure 2. Instead of constructing a surface, I used ezVIZ to take slices through my data, displaying the same difference in speeds as in Figure 1. In the green areas, the "fast" mode dominates, while the "intermediate" mode dominates in the red areas. Figure 2a shows a data slice taken in the plane of the polar axis and the line connecting the Sun to the Earth (Sun not shown). Figure 2b shows a slice from a parallel plane taken 5 Earth radii west of Figure 2a.

Constructing isosurfaces, as in Figure 1, and taking data slices through arbitrary planes, as in Figure 2, are two important visualization tasks that are usually nontrivial. ezVIZ made them relatively easy.

Figure 2. Contours of difference between MHD flow speed and a particular wave speed. In green and blue regions, typical MHD fast waves dominate; in red areas, MHD intermediate mode dominates. (a) in plane of North-South pole and line connecting Sun and Earth. (b) in plane parallel to a, displaced laterally by 5 Earth radii to the west

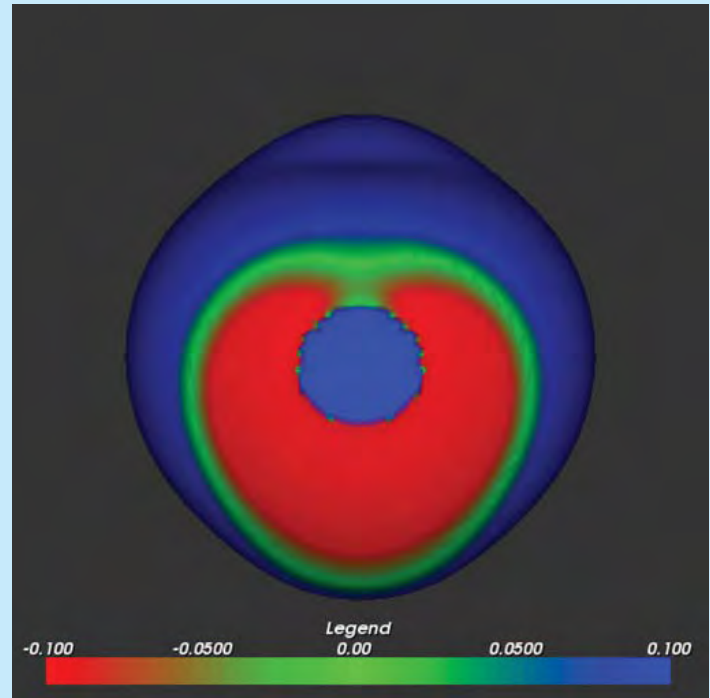
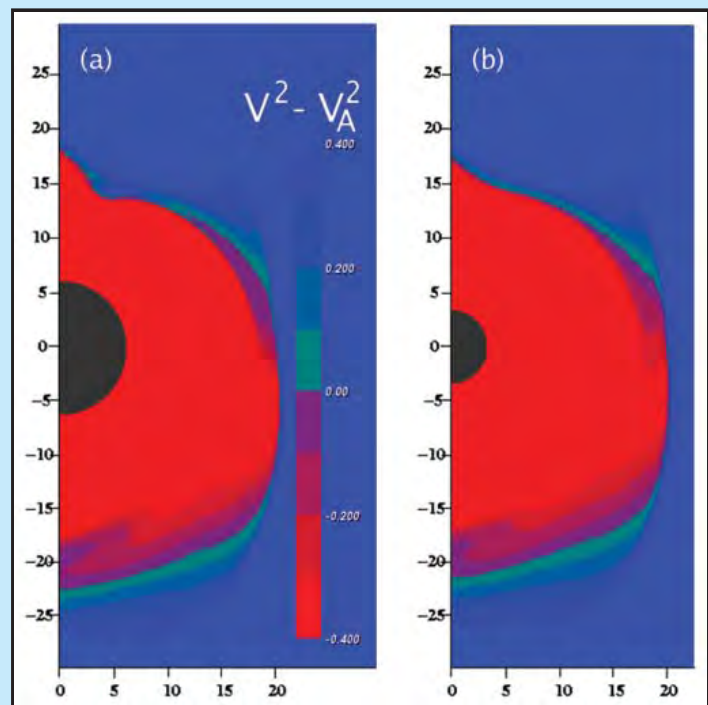


Figure 1. Terrestrial bow shock seen from the direction of the Sun, "painted" with the difference between MHD fluid flow speed and a particular wave speed. Blue areas show dominance of a typical MHD "fast" shock; red areas show dominance of a unique MHD "intermediate" shock



Simulation of Engine Ground Vortex Control

By Drs. Arvin Shmilovich and Yoram Yadlin, The Boeing Company, Huntington Beach, CA

Introduction

Aircraft with turboprop or turbojet engines mounted relatively close to the ground develop vortex activity during high-power, low-speed, and static-ground operation (Figure 1). The suction generated by the engine results in the formation of a stagnation point on the ground. Usually, the ambient flow contains significant amounts of vorticity (turbulence) because of gusts, ground turbulence, wake flow of neighboring aircraft components (i.e., wing, fuselage), and mixing of engine reverser plumes when thrust-reversers are deployed. The mechanism of ground vortex formation is the amplification of the seed vorticity in the ambient flow because of the contracting streamlines approaching the inlet. This interaction results in a concentrated vortex originating at the ground plane and terminating inside the engine. The rotational flow field induced by the ground vortex is the cause for kicked up dust and dirt, which can become entrained in the airflow drawn into the engine inlet. The tornado-like flow is capable of dislodging sizable foreign objects off the ground (for example, rocks, chunks of ice, or asphalt), causing foreign object damage (FOD) that may lead to engine failure. The vexing problem of ground vortex ingestion hinders the ability to land in austere fields and to perform essential ground maneuvers on unimproved terrain. Furthermore, the engine ingestion problem is exacerbated by the advent of larger and more powerful high bypass turbojet engines.

Ground vortex disruption methods that address the unsteady characteristics of realistic inlet vortex flows have been recently developed by The Boeing Company. The pulse jets device developed by Smith and Dorris² uses high-pressure air to alternatively eject fluid from two nozzles mounted underneath the engine nacelle close to the nacelle lips. The intermittent high-frequency ejection provides turbulent mixing to prevent the formation of a coherent vortex. The sprinkler jet actuator proposed by Shmilovich et al.³ uses continuous ejection through a moving nozzle mounted on the nacelle lip in order to provide wide area coverage, thereby reducing the risk of vortex ingestion even when the vortex moves rapidly. The effectiveness of these inlet vortex alleviation methods has been demonstrated for engines in proximity to the ground plane⁴. The current computational fluid dynamics (CFD) simulations focus on the evaluation of the sprinkler system for full airplane configurations. The control system has been incorporated into a model that includes all relevant aircraft components for adequate



Figure 1. Photograph of engine vortex ingestion

representation of the flow during airplane ground operations⁵. The vortex control technique will be briefly reviewed in this article, followed by flow diagnostics for establishing the effectiveness of the sprinkler jet system. Further details are described in Shmilovich and Yadlin⁶.

Sprinkler Jet System for Vortex Alleviation

The flow control technique utilizes fluidic injection in critical regions close to the engine inlet³. A schematic layout of the sprinkler jet system is depicted in Figure 2 for a typical wing engine installation. The flow actuation is accomplished by high-pressure bleed air from the compressor, which is piped to a valve located inside the engine cowl and close to the nacelle lip. The valve is connected to a nozzle located close to the nacelle lip. The nozzle is deployed during low aircraft speed and high engine power setting. During actuation the nozzle swivels according to a prescribed motion in order to inject flow into a large domain in front of the engine inlet in the general upstream direction. The slow motion of the ejected fluid disrupts the global flow field in front of the engine and prevents the formation of vortices. Since realistic full-scale engine vortex ingestion is a highly unsteady phenomenon, this method is intended to break up the nonstationary ground vortex. The amount of air required to affect the inlet vortex is less than one percent of total inlet flow, well within the bleed limit of a typical engine.

Numerical Procedure

The computational method is based on a modified version of the unsteady Reynolds Averaged Navier-Stokes OVERFLOW code originally developed by the National Aeronautics and Space Administration⁷. A special module has been developed by Boeing⁴ for modeling of flow excitation because of control devices.

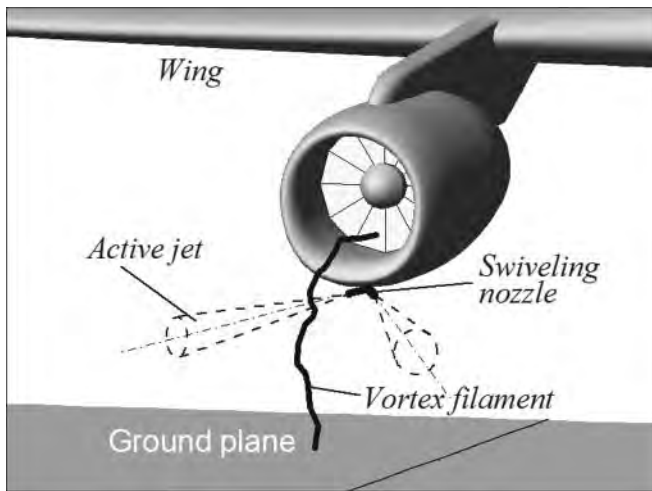


Figure 2. Sprinkler system for reduced vortex ingestion

For the sprinkler jet, this approach avoids the need for moving grid systems and greatly simplifies the analysis. The nozzle is assumed to be stationary, but the jet flux vector at the exit plane is prescribed as a time-varying boundary condition to mimic the swiveling motion of the nozzle.

The engine power setting is defined via the mass flux ratio by specifying fully developed flow at the inlet. The calculations were obtained using a second-order upwind differencing scheme and the shear stress transport (SST) turbulence model. The flow control computations use a second-order, time-accurate scheme with 800 time-steps per actuation cycle. The calculation starts with a steady-state solution obtained for the flow in the absence of any actuation. Limit cycle convergence is usually achieved after approximately 120 actuation cycles.

The control system is applied at the lower part of each engine, close to the inlet lips. Figure 3 shows the grid topology at the engine highlight where the actuator is mounted on the cowl. Coarse grid point distributions are used for the sake of clarity. The overset system consists of 9.4 million points. A set of embedded fine grids are used adjacent to the nozzle exit sections and toward the ground plane in order to accurately capture the jet interaction with the surrounding flow.

Simulation of Ground Vortex Control

Mild tail wind of $M_\infty=0.007$ is considered in this case. High engine power setting is used at the engines to simulate realistic operational conditions. At these conditions the outboard engine is largely exposed to the oncoming tail wind and therefore does not develop a vortex off the ground plane. In contrast, the inboard engine experiences flow blockage because of the fuselage and the outboard engine, resulting in high

suction power in order to satisfy the inlet airflow requirement. The suction results in the formation of the ground vortex leading to inboard engine ingestion. Moreover, because of the proximity of the inboard engine to the fuselage, an additional vortex element is formed off the fuselage surface. The fuselage vortex is also ingested by the inboard engine, but it does not pose risk of FOD. A more thorough description of the vortical structure around airplanes in ground operations may be found in Yadlin and Shmilovich⁵.

The sprinkler jet actuation is applied at both engines. The side-to-side nozzle motions are confined to $\pm 30^\circ$, and the actuation frequency is 140 Hz. The short time-scale flow development in the vicinity of the engine highlight regions is presented in Figure 4, where particles are released from the nozzles. The particles are colored by the local Mach number, where red represents high velocity.

The long time-scale flow development is examined in Figure 5. The vortex structure is described by continuous release of particles off the ground plane underneath both engines and off the fuselage opposite of the inboard engine. The black lines are markers on the ground plane denoting engine axis projections and engine highlight stations. The baseline flow ($t = 0$) illustrates inboard engine ingestion of both ground and fuselage vortex elements. No ground vortex ingestion occurs because of the outboard engine. The induced suction field because of the powered engines is also described by the instantaneous pressure on the ground plane and the fuselage.

The intermittent motion provided by the periodic excitation at the bottom side of each of the engine cowls perturbs the flow in front of the engines. After

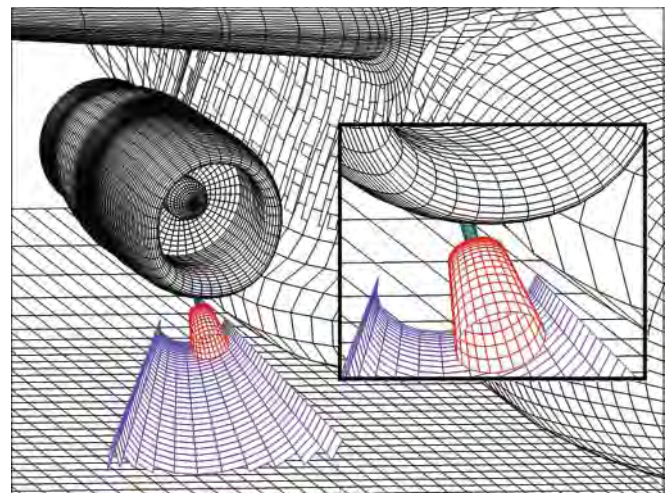


Figure 3. Grid system for modeling of the flow control mechanism

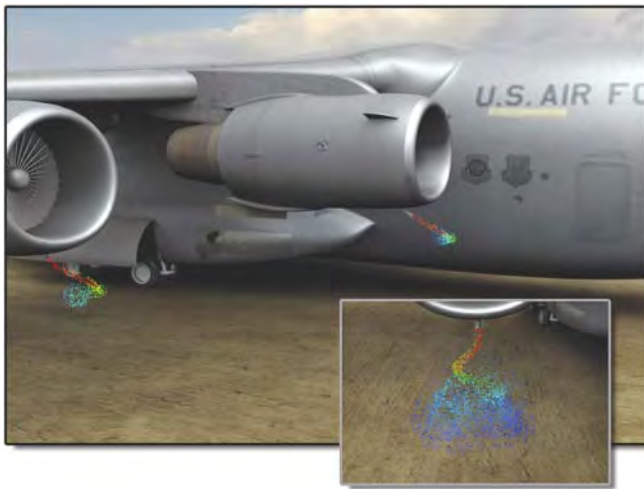


Figure 4. Jet ejection described by particles released from actuators (0.036 seconds from start of actuation)

0.14 seconds the perturbations reach the ground surface, and pressure waves are generated in the near field underneath the respective engines. The ripple effects propagate radially with decreasing intensity in pressure fluctuations until the flow reaches a limit-cycle behavior at $t > 1.00$ seconds. The ground vortex is disrupted close to the inboard engine inlet at $t = 0.14$ seconds. The subsequent time frames show that the vortex filament is altered by the ejecting flow and is expelled away from the engine. Engine vortex ingestion from the bottom side has been curbed by the sprinkler actuation, while the vortex off the fuselage is only slightly affected.

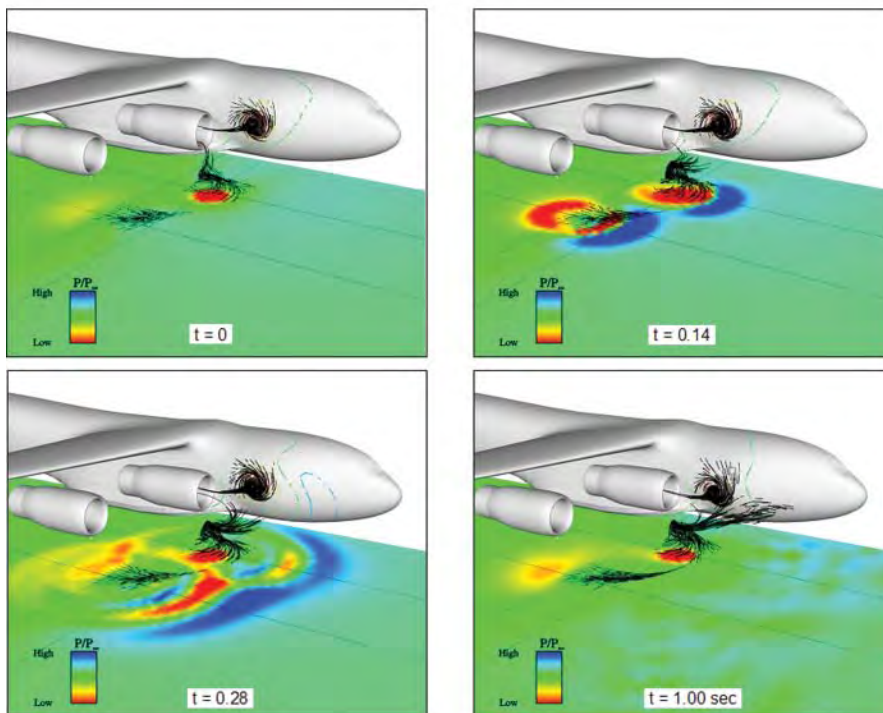


Figure 5. Global flow development

Conclusion

Fluid dynamics simulations have been performed on the ERDC Cray XT3 system for analysis and evaluation of an active flow control concept for vortex alleviation or outright vortex removal. The efficacy of the sprinkler vortex inhibitor has been confirmed by the examination of the vortex structure, particle traces, and engine capture streamtube. The actuation technique is based on flow ejection out of a swiveling nozzle, resulting in reduced suction power at the ground plane, thereby suppressing ground vortex ingestion and its concomitants, the risk of FOD, and engine surge. The sprinkler jet method is particularly attractive since it does not assume that the ground vortex is stationary and does not rely on prior knowledge of vortex locus. Its wide area coverage is a key attribute, which makes it especially suitable for unsteady engine vortex control.

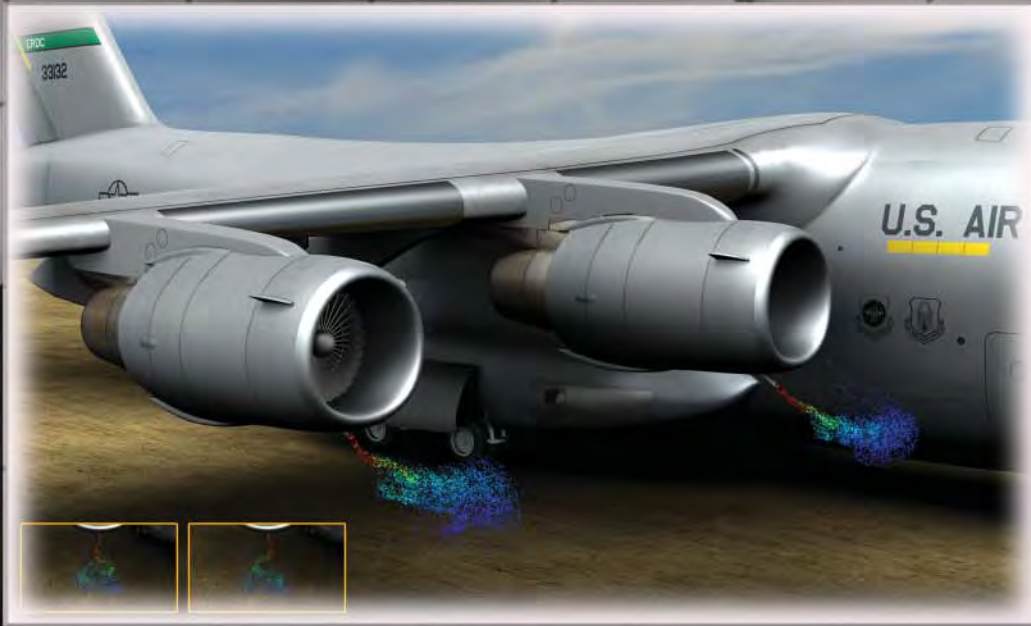
Acknowledgments

This work was sponsored by the Air Force Research Laboratory and supported in part by an allocation of computer time from the DoD HPCMP at ERDC.

References

- ¹Blincow, K., www.airliners.net, photo ID 326528, 2000.
- ²Smith, D.M. and Dorris, J., McDonnell Douglas Corporation, St. Louis, MO, "Aircraft Engine Apparatus with Reduced Inlet Vortex," U.S. Patent 6,129,309, 10 Oct. 2000.
- ³Shmilovich, A., Yadlin, Y., Smith, D.M. and Clark, R.W., The Boeing Company, Chicago, IL, "Active System for Wide Area Suppression of Engine Vortex," U.S. Patent 6,763,651, 20 July 2004.
- ⁴Shmilovich, A. and Yadlin, Y., "Engine Vortex Flows and Methods of Ground Vortex Alleviation," *Proceedings of the 3rd International Conference on Vortex Flows and Vortex Models*, Yokohama, Japan, 2005.
- ⁵Yadlin, Y. and Shmilovich, A., "Simulation of Vortex Flows for Airplanes in Ground Operations," AIAA Paper 2006-0056.
- ⁶Shmilovich, A. and Yadlin, Y., "Engine Ground Vortex Control," AIAA Paper 2006-3006.
- ⁷Buning, P.G., Chiu, I.T., Obayash, S., Rizk, Y.M and Steger, J.L., "Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent," AIAA Paper 1988-4359.

Pullout Poster



See what's in your data!

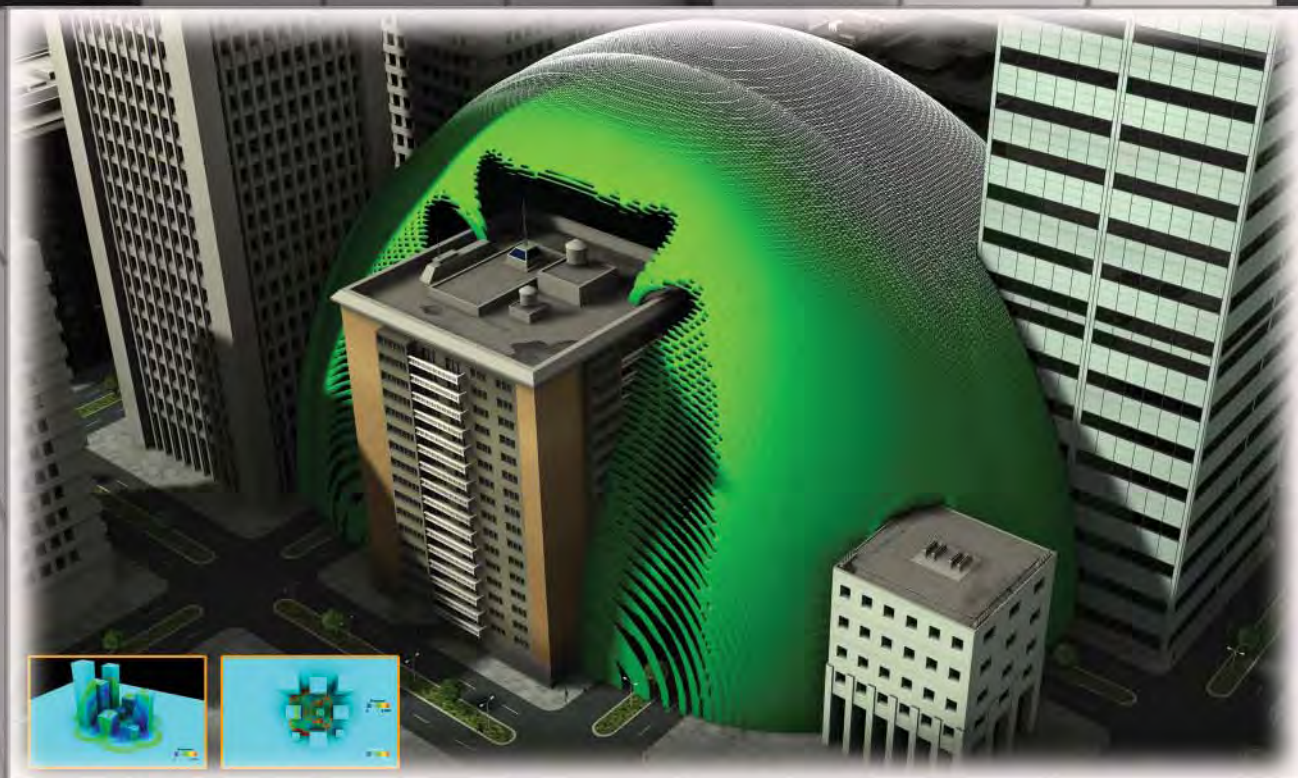


A New Name. A New Mission.

The ERDC MSRC Scientific Visualization Center is the new HPCMP's Unclassified Data Analysis and Assessment Center (DAAC).

Our mission and our goal is to put visualization and analysis tools and services into the hands of every user.

Visit visualization.hpc.mil to learn how you can use our tools and services, or contact us at svchelp@visualization.hpc.mil



visualization.hpc.mil

Pullout Poster

DoD Visualization Has a New Place on the Internet

By Paul Adams

In January 2007, the Unclassified Data Analysis and Assessment Center launched a new visualization.hpc.mil Web site. The single purpose of this visualization Web site is to be helpful to users. With ERDC providing all the unclassified visualization for all HPCMP users, we expect to have many users who are not only new to visualization but also new to our center. This site is geared to help these users in several ways.

First, and foremost, the visualization.hpc.mil Web site has a News section that answers the most common questions:

- How do I get an account on VIZ systems?
- How do I use VIZ software?
- What systems are available for visualization?
- How do I get help with my visualization?

The News section also contains information on maintenance events on our systems and useful tips and tricks with visualization software. If a user wants to know how to create high-quality movies for presentations, we answer that question here. If users want information on performing remote visualization, here is where we discuss how it is done and what some of the likely problems are that they will encounter.

Secondly, the visualization.hpc.mil Web site has a Forum section. The Forum is where users can ask questions of the visualization experts and may receive an answer from other users. Do you have a particular problem with ParaView or EnSight? Ask us here. Do you need to know how to write out a streaming binary file (a.k.a. C binary file) in FORTRAN 2003? We have answered that question here.

Third, the visualization.hpc.mil Web site has a Wiki section for more in-depth training with visualization software. We have entire sections on ParaView,

Forum	Topic	Last Post
Amethyst	X Errors with Exported Displays	04/02/07 14:14 PM
Website	Wiki content types allowed	03/26/07 10:38 AM
Website	New Layout	03/26/07 16:42 PM
Amethyst	Writing C binary files from Fortran	03/08/07 11:16 AM
Website	Complex forum posting pages	03/07/07 13:38 PM

EnSight, and ezVIZ. Did you miss the information on performing remote visualization on the News page? Not to worry—it is also found under the Wiki. Do you want to download some podcasts (a.k.a. software training videos)? We have them here as well.

Under the Gallery section, you can see examples of our past work. In this section we have posted smaller versions of our posters that we create for our users. But this section is not just for posters. For some of the projects, we include smaller examples of the movies we created for the user. We also explain the process we used to create the visualization in the first place, along with the tools that we used and the visualization techniques.

Please feel free to drop by the new visualization.hpc.mil Web site.

ERDC MSRC Transitions New Visualization Hardware to Data Center

By Paul Adams

The HPCMP continues to increase its computing capability by installing machines that perform over 20 trillion operations per second. Along with this capability comes an increase in the complexity and size of research performed. With the recent upgrades to the Unclassified Data Analysis and Assessment Center (DAAC), located at the former ERDC MSRC Scientific Visualization Center, the users have a 25-fold increase in data analysis capabilities.

A new state-of-the-art graphics cluster and file server were recently installed at the DAAC. This visualization cluster is named Amethyst and is manufactured by Graphstream Incorporated. Amethyst contains six visualization nodes. Each visualization node contains eight dual-core CPUs (for a total of 16 cores) running at 2.4 GHz. Each node also has access to 128 GB of shared memory and an NVidia Quadro 5500 with 1 GB of graphics memory. Storage for Amethyst is provided by a file server that consists of 20 TB of shared disk space. Additional computational capability comes with a render farm that consists of 60 blades, each containing two dual-core 2.8 GHz Intel Xeon processors, 4 GB of memory, and 73 GB SCSI drive.

Amethyst allows the DAAC to reduce the time to discovery. The capability that Amethyst adds to the DAAC allows a researcher to interactively view, for example, data sets with tens of millions of polygons. The interactive capability to view large data sets gives scientists unprecedented opportunity to explore and discover phenomena within their data that they might not have otherwise seen.

In addition, Amethyst allows the DAAC to reduce the time to delivery. Combined with the capability of the render farm, this allows researchers to view their data



Paul Adams standing with Amethyst

within some contextual situation. A recent example had seven fluid-flow scenarios. Each scenario had hundreds of data sets, with each data set representing a different time-step. These data sets were processed in parallel on Amethyst in hours. They were then ray traced in parallel on the render farm with context added into the scene. The seven finished movies were delivered to the researcher in less than a month, whereas previously the process would have taken a couple of months.

With these advancements and expanded multimedia authoring capabilities, the DAAC continues to be a leader in delivering to users the capability to display their conceptual and scientific data in any forum.

ERDC Infrastructure Takes a New Spin

By Greg Rottman and Paula Lindsey

We have been busily renovating the ERDC MSRC facilities to support the Technology Insertion 2007 (TI-07) acquisition and upgrades to the mass storage archive system. In early 2006, we realized the future requirements for facilities upgrades to support more new systems. Immediately, we began to develop plans to increase the amount of available backup power and cooling for critical computing systems. To increase the flexibility of our current facility, we are modifying our computer room layout.

Electrical

The current Uninterruptible Power Facility was near its maximum capacity and could not be upgraded because of limitations in the existing switchgear. A plan was developed to offload the chiller pad and to install four Liebert 650 kVA UPS and twenty Liebert flywheels to provide an additional 1.8 MW of backup power to the MSRC's computers.

A new 2800 square foot equipment shelter (Figure 1) was built on the roof of the ERDC Information Technology Laboratory building to provide environmental protection for the UPS, flywheels, switchgear, transformers, and other equipment.

A new 2.25 MW Caterpillar generator was installed behind the existing powerhouse to provide backup power to the chiller pad. The new generator will ensure continued operation of the chillers during electrical outages.

Cooling

One new 500 ton chiller (Figure 2) and a 75 horsepower pump were installed to provide the additional cooling necessary to support the TI-07 system, resulting in a total of 1200 tons of critical cooling. In order to deliver this additional cooling capacity to building 8000, over 200 feet of new 10-inch pipe was installed. Additional piping was installed inside the main computer room to distribute the chilled water to the required areas.

Physical Space Changes

In order to maximize flexibility of the floor space in the main computer room and to allow



Figure 1. New UPS equipment shelter located on the roof of ERDC Information Technology Laboratory building



Figure 2. Delivery of 500 ton chiller

for the effective installation of air handlers, changes were implemented. Two walls, which presently divide the main computer room into several smaller rooms, were removed, providing an expanded contiguous raised floor area. Ceiling and floor alignment was corrected to present a seamless compute facility.

These modifications are part of the ever changing compute environment at ERDC. Efforts to provide an effective infrastructure in support of the ERDC MSRC will continue. Look forward to more changes as demand for high performance computing systems continues to grow.

ERDC Adds over 12,000 Processors to Offering

By Jay Cliburn

The ERDC MSRC is pleased to provide details of two major system changes in connection with the HPCMP Technology Insertion 2007 (TI-07) acquisition cycle.

Cray XT3 Upgrade

The Cray XT3, hostname *Sapphire*, was significantly upgraded in March 2007. The upgrade included replacing all 4,176 single-core 2.6 GHz AMD Opteron CPUs with dual-core Opterons of the same 2.6 GHz clock speed, coupled with increasing the compute node memory from 2 to 4 GB of RAM per node, thus preserving the 2 GB per processor memory-to-CPU ratio. A new disk subsystem was also added, providing 210 TB of additional Lustre workspace storage. This new disk subsystem is accessible as the `/work` directory; the pre-upgrade workspace directory remains available, but has been renamed to `/work2`.

In addition to Sapphire's new hardware, there are significant software changes, too. Most visibly, the batch queuing system changed from LSF to PBS. Not surprisingly, this requires modifications to user job submission scripts; however, the HPC Service Center stands ready to assist users in making the necessary changes. Other software changes include an upgrade of the system software to version 1.5, bringing in all the latest bugfixes and feature content from Cray, and the installation of the Qlogic PathScale Compiler Suite, installed alongside the existing Portland Group compiler suite. The PathScale compiler suite is optimized for AMD64 processors and provides support for C, C++, and FORTRAN 77/90/95.

Another major change relates to the configuration of Sapphire's login nodes. Before the upgrade, users

Before Upgrade		After Upgrade	
Node	Purpose	Node	Purpose
sapphire01	login, batch control	sapphire01	login
sapphire02	login, batch control	sapphire02	login
sapphire03	login, batch control	sapphire03	login
sapphire04	login, batch control	sapphire04	login
sapphire05	login, batch control	sapphire05	login
sapphire06	login, batch control	sapphire06	login, compiler license server
sapphire07	login, batch control	sapphire07	interactive, restricted access
sapphire08	login, batch control	sapphire08	interactive, restricted access
sapphire09	login, batch control	sapphire09	batch control
sapphire10	login, batch control, compiler license server	sapphire10	batch control
sapphire11	ERDC-NAVO file transfer	sapphire11	batch control
sapphire12	ERDC-NAVO file transfer	sapphire12	batch control
sapphire13	ERDC-NAVO file transfer	sapphire13	batch control
sapphire14	ERDC-NAVO file transfer	sapphire14	batch control
sapphire15	unused	sapphire15	batch control
sapphire16	unused	sapphire16	batch control
sapphire17	unused	sapphire17	batch control
sapphire18	unused	sapphire18	batch control
sapphire19	unused	sapphire19	batch control
sapphire20	unused	sapphire20	batch control

Cray XT3 Login Node Configuration

accessed Sapphire by logging directly in to sapphire01-sapphire10. Users may not have known it back then, but they shared their login node with job-related processes—processes used by the batch queuing system to control and monitor jobs and to perform pre- and postprocessing. This sharing sometimes taxed the CPU and memory resources of login nodes, resulting in sluggish response to user commands and an overall slowdown on the affected nodes. As part of the upgrade, Sapphire’s login nodes were reconfigured to separate user interactive processes from job management processes. The preceding table summarizes the before and after configuration of Sapphire’s login nodes.

Keeping user login processes separated from job control and execution processes should reduce contention for scarce resources on login nodes. If users have other suggestions for improvements on the XT3, we’d like to hear them. Please send any such suggestions to the HPC Service Center.

Cray XT4

The ERDC MSRC expects to take delivery of a Cray XT4 in the first or second quarter of fiscal year 2008.

This system’s hostname will be *Jade*. It will consist of 24 cabinets and will provide an estimated 80 teraflop/s of computational capacity. Each of its 538 compute blades will contain 4 quad-core 2.3 GHz Opteron, for a total of 8,608 compute cores. (The 2.3 GHz clock speed is an estimate and depends upon what’s available from AMD at the time of Jade’s delivery.) Each compute node will run Linux—unlike Sapphire, which runs Catamount on its compute nodes—and will be populated with 32 GB of memory. The system will contain over 370 TB of Lustre workspace disk storage. The XT4 also sports an improved internal node interconnect, the SeaStar2, which provides a sustained bandwidth of over 6 GB/sec. (By comparison, the older SeaStar on Sapphire provides 4 GB/sec of sustained bandwidth.)

The ERDC MSRC is excited about the computational capacity offered by these powerful systems and looks forward to bringing them into production service to meet the needs and goals of its users. Please don’t hesitate to contact the HPC Service Center at msrhelp@erdc.hpc.mil if you have questions or need additional information.



Upgrades are made to the Cray XT3

Reconfigurable Computers

By Dr. Gerald R. Morris

Introduction

The silicon transistor technology used in general-purpose processors (GPPs) is rapidly approaching a brick wall in terms of clock rate and density. GPP organizational improvements such as pipelining, branch prediction, and multiple instruction issue have exploited much of the available instruction-level parallelism. Furthermore, there is an upper limit on the extent to which clusters can improve performance. At processor counts above a thousand or so, the communication costs often exceed the computational costs. Finally, processor hours are a critical resource; large jobs may be queued up for several days or even weeks. These considerations as well as resource constraints such as power consumption and floor space mean that large computer centers must be on the lookout for new technologies to improve the performance of scientific codes. Reconfigurable computers (RCs), which are based on field programmable gate arrays (FPGAs), may be one of these technologies.

FPGA Primer

FPGAs are integrated circuit devices that can be configured by end users to implement customized digital logic circuits. FPGAs were invented in the mid-1980s by Xilinx cofounder Ross Freeman [26]. His idea was to store the truth table for each logic function in small one-bit-wide memories called look up tables (LUTs). This was a very radical idea at the time because silicon real estate was an expensive commodity. It did not seem reasonable to use precious memory to emulate the behavior of digital logic circuits when actual minimized logic implementations were an order of magnitude smaller and faster. History and Moore's Law have vindicated Freeman, and FPGAs are now one of the fastest growing semiconductor markets [17, 6]. FPGA-based logic is still slower than actual logic, but the lower nonrecurring engineering costs, faster development time, and ability to configure the FPGAs in the field make them a viable alternative in many applications. In theory, any digital logic circuit can be placed on an FPGA. In practice, the primary constraints are the programmable logic area, clock rate, and input/output (I/O). Principal FPGA vendors include Xilinx, Altera, and Actel [25, 3, 1]. Terminology varies among vendors, but the concepts are the same; this article uses Xilinx terminology.

FPGA Architecture

LUTs mimic digital logic behavior by storing the logic function truth table. The address bits correspond to the logic function inputs, and the bit stored at each address corresponds to the function value. Figure 1 illustrates the idea for a two-input exclusive-or (XOR) function; one will recall that XOR is true when exactly one input is true. As shown in Figure 2a, multiple LUTs, flip-flops, adder carry and other logic, multiplexers, and control logic are grouped together into what Xilinx calls a "slice." To allow for modest-sized subcircuits without the need for external routing resources, multiple slices and a fast internal interconnect (switch) are grouped together into configurable logic blocks (CLBs), as shown in Figure 2b. As suggested by Figure 3, contemporary FPGAs have tens of thousands of CLBs as well as fixed logic blocks such as random access memory (RAM), multipliers, clock managers, and even GPPs embedded in a programmable interconnection mesh surrounded by programmable I/O blocks. Antifuse-based FPGAs can only be configured one time; however, static RAM (SRAM)-based FPGAs can

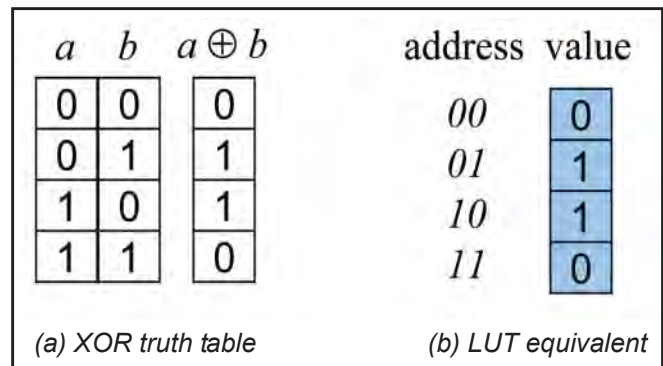


Figure 1. Truth tables are mapped onto LUTs

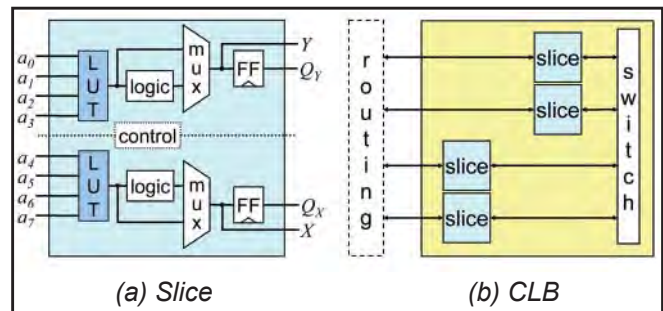


Figure 2. Lower level FPGA elements

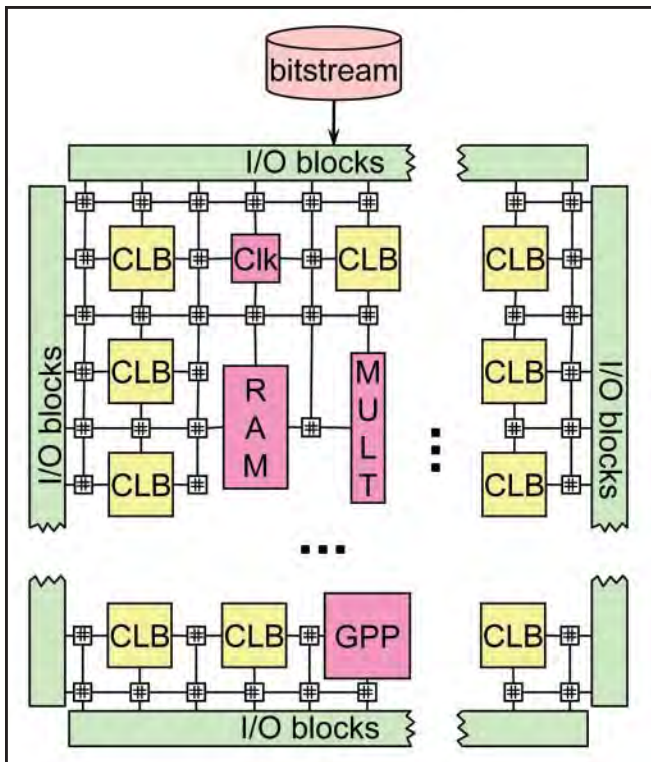


Figure 3. Idealized FPGA

be reconfigured an arbitrary number of times. For these devices, the circuit design information is contained within a configuration bitstream that is loaded onto the FPGA. One obtains a new logic device by simply loading a new bitstream.

FPGA Design Flows

Figure 4a depicts a hardware description language (HDL)-based FPGA design flow. Using this approach, the hardware engineer creates a description of the circuit using a hardware description language like VHDL [13]. The engineer verifies circuit operation at the HDL level via a simulation (SIM) environment such as Modelsim [14]. As noted in the diagram, the design can also be simulated, with increasing accuracy, at later stages in the design flow. After circuit operation has been verified, the HDL is sent through a front-end synthesis (SYNTH) tool such as Synplify Pro to produce netlist files [24]. The netlists, which are essentially text-based descriptions of the schematic, are processed by the vendor-specific back-end place and route (PAR) and bit generation (BITGEN) tools to produce a configuration bitstream.

Several commercial and open-source high-level language (HLL)-to-HDL development environments are now available. Examples include Mitrion-C and the Brigham Young University (BYU) JHDL initiative [16,

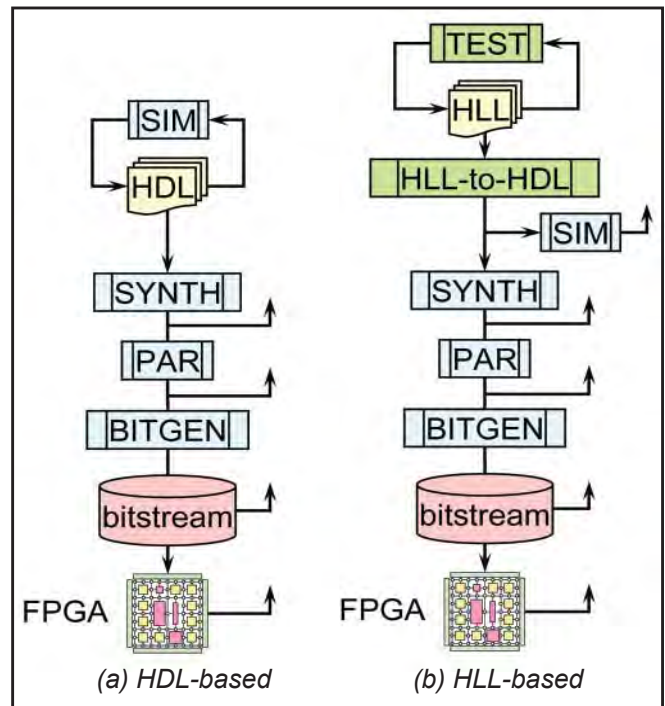


Figure 4. FPGA design flows

5]. These tool sets allow the development of FPGA configuration bitstreams using HLL-based programming rather than HDL-based hardware design. As shown in Figure 4b, HLL-to-HDL environments typically provide a functional testing (TEST) mechanism, which operates at the HLL level. After the design functionality has been verified, the HLL-to-HDL compiler inputs the HLL and emits HDL. The HDL is processed by the FPGA tool chain, as described previously, to produce a configuration bitstream. As with the HDL flow, the design can be simulated at later stages using a SIM tool.

Reconfigurable Computers

The reconfigurable computer, which was proposed by Gerald Estrin in 1960, is a “fixed plus variable structure” computer consisting of fixed digital logic modules and a variable structure that can be “temporarily distorted” into a problem-oriented special purpose computer [10]. Technological limitations and the advent of the GPP caused further research and development of the RC to wither. However, FPGAs, with their mega-gate capacity, high-speed I/O, and other features, have precipitated a resurgence. Modern RCs combine GPPs with SRAM-based FPGAs; the FPGAs are, in effect, reconfigurable application-specific processing elements (PEs). During one run, the FPGA might be a matrix-vector multiply PE; during another run, it might be a linear equation solver. Several companies now

offer FPGA-based RCs. The late Seymour Cray's startup company, SRC Computers, offers the MAP processor, which includes two user-programmable FPGAs in a number of different configurations ranging from single-MAP workstations to high performance multiple-MAP clusters [23]. SGI offers the dual-FPGA RC100 blade, which can be placed as a peer compute node in SGI's NUMALink switching fabric [21]. Mercury Computer Systems has the triple-FPGA Powerstream FCN Module, which can be placed as a peer compute node in Mercury's RapidIO switching fabric [15]. Cray announced they would be using the DRC direct-connect module to replace the capability of their recently discontinued XD1 line of FPGA-augmented RCs [12].

RC System Architecture

While different in detail, all of the RCs mentioned previously are similar in the sense that the FPGA-based portions can be viewed as the variable structure PE described by Estrin [10]. Figure 5 is an idealized diagram of an RC. The fixed PEs correspond to traditional GPPs and the associated memory hierarchy. The variable structure PEs typically have one or more FPGAs surrounded by multiple memory banks. The local memory banks, which are independent from the GPP memory, give the FPGAs the ability to store large amounts of data and to access multiple values in a single FPGA clock cycle. The fine-grained resolution of FPGAs allows the RC hardware to be reconfigured specifically for the problem to be solved. For applications that have some combination of large-strided or random data reuse, streaming, parallelism, and computationally intensive loops, RCs can achieve higher performance than GPPs.

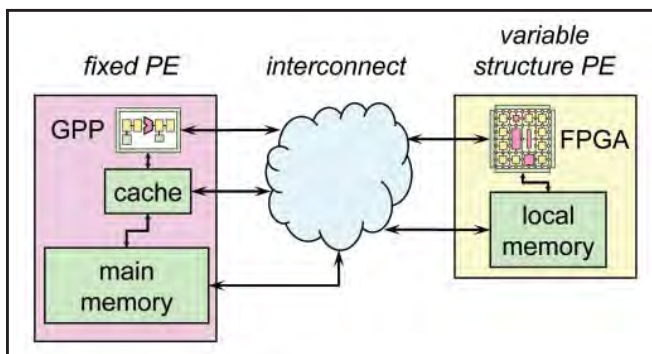


Figure 5. RC system architecture

RC Design and Performance Issues

To achieve performance that is competitive with GHz scale GPPs, the MHz scale FPGA-based logic designs have to be both deeply pipelined and highly parallelized. The author coined the phrase “the three

p’s” to encapsulate this important relationship among performance, pipelining, and parallelism [18]. For single-cycle integer and fixed-point designs, adhering to the three p’s is relatively easy. Alex et al. demonstrate a 30-fold speedup over software for an FPGA-based protein sequencer [2]. Cheung et al. describe an FPGA-based elliptic curve cryptosystem that achieves a 25-fold speedup over software [8]. Baker and Prasanna show a 24-fold speedup over software for the Apriori data mining algorithm [4].

However, scientific computing generally requires the greater precision and range afforded by floating-point arithmetic. Unfortunately, designs that employ multiple-cycle pipelined floating-point intellectual property (IP) cores do not always map well onto RCs. There is also the determination of what the author terms “the FPGA design boundary,” i.e., the portion of the application that is mapped onto the FPGA. Unresolved loop-carried dependences and similar issues that violate the three p’s can significantly affect the performance of RCs. It can take a significant amount of effort to efficiently map floating-point applications onto RCs. The author’s first attempt to map a simple floating-point sparse matrix vector multiply kernel onto an RC actually resulted in a tenfold slowdown. A full discussion is beyond the scope of this article. However, researchers have begun to address these issues and to realize actual runtime speedups over software for floating-point RC-based applications. Morris and Prasanna achieve more than a twofold speedup for two double-precision floating-point RC-based sparse matrix solvers [19]. Scrofano et al. obtain a twofold speedup over software for their single-precision floating-point RC-based molecular dynamics code [20]. Devlin et al. describe a single-precision floating-point RC-based one-dimensional convolution kernel that can achieve a tenfold speedup over software [9]. Gokhale et al. report a tenfold speedup for a single-precision floating-point RC-based heat transfer simulation [11].

HDL Design Flow

Figure 6 illustrates the HDL-based RC design flow. The design is partitioned into software modules, which are written in a traditional HLL and targeted for execution on the GPPs, and FPGA modules, which are written in an HDL and targeted for execution on the FPGAs. Software modules that call FPGA modules also include some vendor-specific application programmer interface (API) calls to control and use the FPGA. The software modules are compiled with the normal software compiler to produce object files. The

FPGA module HDL code is input to the synthesis tool. The netlists produced by synthesis are fed into PAR, which feeds BITGEN to produce the configuration bitstream. The linker inputs the object and library files and produces the executable. At runtime, the GPP and FPGA cooperatively execute the application.

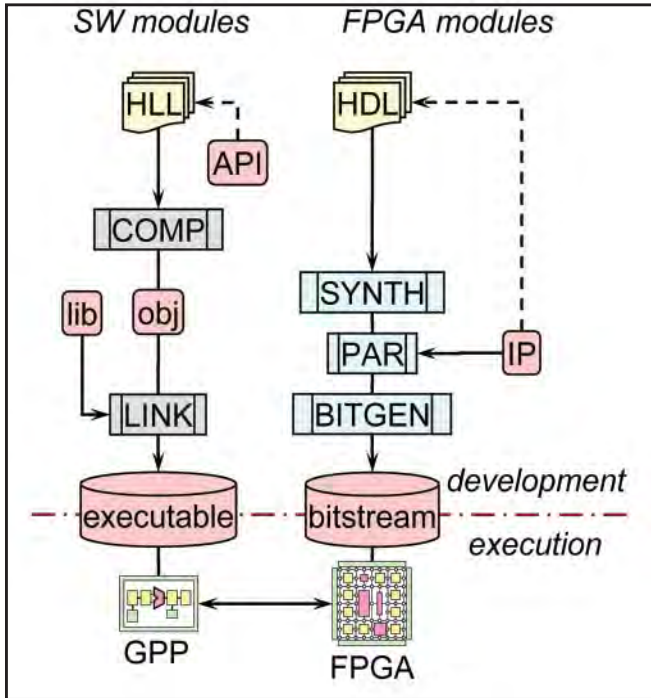


Figure 6. HDL-based RC design flow

The HDL-based RC design flow is relatively primitive. Despite the use of the nomenclature “module,” these designs are not really modular; there is a significant amount of coupling between the software module and FPGA module. The software must have an intimate knowledge of the hardware functionality and is usually responsible for data synchronization. Furthermore, the FPGA module designer must be an experienced hardware designer. The HDL-based RC design flow is not really practical for mainstream scientific computing on RCs.

HLL Design Flow

Estrin required general-purpose computers in the fixed structure to facilitate “higher level languages for man-machine communication.” An HLL-based RC development approach is important because it introduces a man-machine communication model that abstracts away many of the FPGA details such as the I/O interface and especially the clock. It also provides a truly modular programming model. In the HLL-based RC design flow, the FPGA module appears to be a standard call within the software module. One can pass parameters to the FPGA module without having to under-

stand how the FPGA module operates.

HLL-based compilers allow FPGA module development using HLL-based programming rather than HDL-based hardware design. However, there are differences between the “normal” HLL in the software modules and the “special” HLL in the FPGA modules. Traditional HLLs, which were developed for von Neumann uniprocessor architectures, do not have mechanisms for expressing parallelism. Therefore, HLL-to-HDL compiler developers have taken one of four approaches: (1) modify an existing HLL as with Celoxica’s Handel-C; (2) create a new HLL as with Mitrionics’ Mitrion-C; (3) add appropriate classes to an object-oriented language as with BYU’s JHDL; or (4) use a standard HLL but include pragmas to guide the compiler as with SRC’s Carte compiler [7, 16, 5, 22]. Independent of the mechanism, the goal is deeply pipelined, highly parallelized hardware. Therefore, in addition to parallel blocks, HLL-to-HDL compilers provide features such as pipelined loops, communication channels, synchronization primitives, and API calls to access specialized IP cores.

As Figure 7 illustrates, the design is still partitioned into software modules and FPGA modules. The software modules are written with normal HLL and compiled with the normal software compiler to produce object files. The FPGA modules are written using the special HLL provided by the target HLL-to-HDL compiler. The FPGA module HLL code is compiled

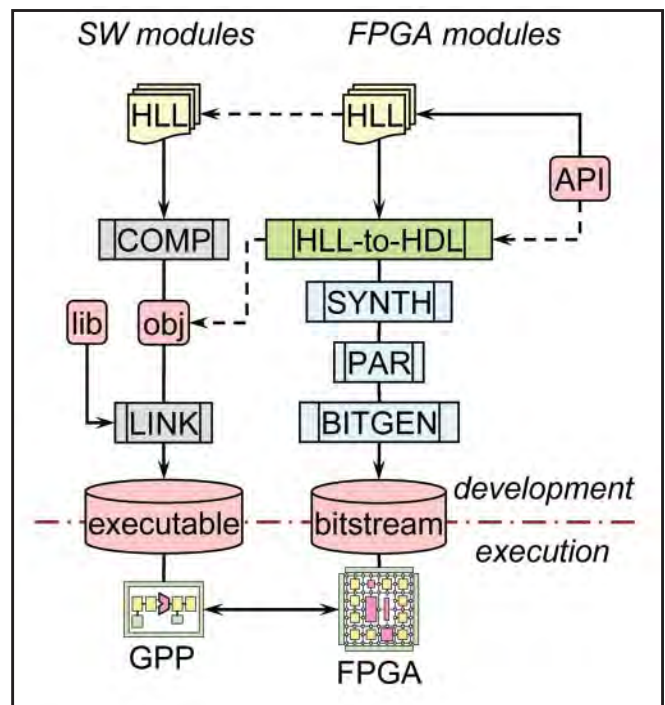


Figure 7. HLL-based RC design flow

with the HLL-to-HDL compiler to produce HDL input to the synthesis tool. Synthesis feeds PAR, which feeds BITGEN to produce the configuration bitstream. The linker inputs the object files, library files, the FPGA module call specification, and produces the executable.

Even though the FPGA module HLL code looks like software, it is still a hardware design. Certainly the modularity and more familiar programming languages make the HLL-based RC design flow an improvement over the HDL-based flow. But the parallel blocks, pipelined loops, and other features added to the HLLs; the requirement to adhere to the three p's; and several other issues force one to concede that the HLL-based approach is still not quite ready for mainstream super-computer users who require floating-point arithmetic.

Hybrid Design Flow

HLL-based RC development environments such as Celoxica's DK Design Suite and SRC's Carte support a hybrid development approach. This hybrid design flow allows the developer to use an HDL or other design approach to create customized IP cores and import them into the HLL environment. As a result, the developer can use all the vendor HLL features such as parallel code blocks, pipelined loops, and channels, yet still have HLL access to the customized IP cores.

Figure 8 illustrates the hybrid approach. The development tools provide some interface mechanism allowing the HLL-to-HDL compiler to obtain visibility into the user IP cores. This interface specifies the format of the

call statement used within the FPGA module's HLL code to access the IP core. The interface also provides information to the HLL-to-HDL compiler allowing it to integrate the custom IP core at the HDL level. The software modules are compiled with the software compiler to produce object files. The FPGA module HLL code is compiled with the HLL-to-HDL compiler. The HDL output and the user HDL code are input to the synthesis tool. Synthesis outputs are fed to PAR, which feeds BITGEN to produce the bitstream. The linker inputs the object files, library files, the FPGA module call specification, and produces the executable.

In the hybrid design flow, one can use all the HLL features yet still have access to the customized user-defined IP cores from within the HLL-based portion of the FPGA module design. Unfortunately, the FPGA module is still a hardware design. All of the considerations that kept the HLL-based approach from being ready for mainstream floating-point supercomputer usage apply to the hybrid approach.

Conclusion

For integer and fixed-point applications, RCs are ready to go; but for floating-point applications, RCs are not quite ready for prime time. Certainly the research to date shows a 2- to 10-fold speedup over software for some floating-point applications. However, these results required a significant design effort. The author's experience with the scientists and engineers who use supercomputers is that they want their codes to compile and run, with minimal modifications, on whatever new platform comes along. They are interested in their science, not in writing code. It is unlikely that they have the experience or can afford to spend the time needed to obtain speedups for their floating-point codes. Even though the research mentioned in this article demonstrates that RCs can be used to speed up floating-point applications, these speedups were obtained after a significant design effort. The challenge for computer engineering and compiler researchers is to find ways to make RCs more accessible to supercomputer users who have floating-point codes.

References

- [1] Actel, Inc. Programmable logic solutions. www.actel.com.
- [2] A. Alex, J. Rose, R. Isserlin-Weinberger, and C. Hogue. Hardware accelerated novel protein identification. In *Proceedings of the 14th International Conference on Field Programmable Logic and Application (FPL'04)*, pages 13–22, Antwerp, Belgium, September 2004.
- [3] Altera, Inc. FPGAs, CPLDs, and Structured ASICs. www.actel.com.

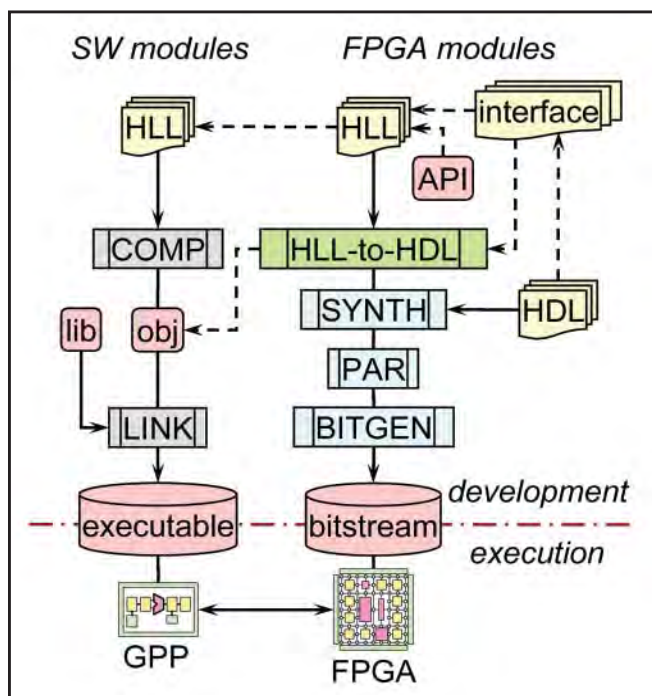


Figure 8. Hybrid RC design flow

- [4] Z. K. Baker and V. K. Prasanna. An architecture for efficient hardware data mining using reconfigurable computing systems. In *Proceedings of the 14th IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'06)*, pages 67–75, Napa, CA, USA, April 2006.
- [5] Brigham Young University. JHDL. www.jhdl.org.
- [6] C. Cadden and J. Worchel. FPGA market will reach \$2.75 billion by decade's end. In *In-Stat Market Research*. www.instat.com/press.asp?Sku=IN0603187SI&ID=1674, May 2006.
- [7] Celoxica, Ltd. Handel-C Language Reference Manual. celoxica.com/techlib/files/CEL-W0410251JJ4-60.pdf.
- [8] R. C. C. Cheung, W. Luk, and P. Y. K. Cheung. Reconfigurable elliptic curve cryptosystems on a chip. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE'05)*, pages 24–29, Washington, DC, USA, 2005.
- [9] M. Devlin, R. Bruce, and S. Marshall. Implementation of floating-point VSIPL functions on FPGA-based reconfigurable computers using high-level languages. In *Proceedings of the 8th MAPLD International Conference (MAPLD'05)*, page www.klabs.org/mapld05/papers/186devlinpaper.doc, Washington, DC, USA, September 2004.
- [10] G. Estrin. Organization of computer systems—the fixed plus variable structure computer. In *Proceedings of the Western Joint Computer Conference*, pages 33–40, San Francisco, CA, USA, May 1960.
- [11] M. Gokhale, J. Frigo, C. Ahrens, J. L. Tripp, and R. Minnich. Monte Carlo radiative heat transfer simulation on a reconfigurable computer. In *Proceedings of the 14th International Conference on Field Programmable Logic and Application (FPL'04)*, pages 95–104, Antwerp, Belgium, September 2004.
- [12] HPCwire. Cray selects DRC FPGA coprocessors for supercomputers. In *HPCwire*. www.hpcwire.com/hpc/644554.html, May 2006.
- [13] IEEE Std 1076-2002. *IEEE Standard VHDL Language Reference Manual*. Institute of Electrical and Electronics Engineers, New York, NY, USA, May 2002.
- [14] Mentor Graphics, Inc. ModelSim. www.model.com.
- [15] Mercury Computer Systems, Inc. Powerstream FCN module. www.mc.com/products/boards.cfm.
- [16] Mitronics, Inc. Mitrion-C. www.mitronics.com.
- [17] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38:114–117, April 1965.
- [18] G. R. Morris. *Mapping Sparse Matrix Scientific Applications onto FPGA-Augmented Reconfigurable Supercomputers*. Ph.D.E.E. dissertation, University of Southern California, Los Angeles, CA, USA, December 2006.
- [19] G. R. Morris and V. K. Prasanna. Sparse matrix computations on reconfigurable hardware. *Computer*, 40(3):58–64, March 2007.
- [20] R. Scrofano, M. Gokhale, F. Trouw, and V. K. Prasanna. A hardware/software approach to molecular dynamics on reconfigurable computers. In *Proceedings of the 14th IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'06)*, pages 23–32, Napa, CA, USA, April 2006.
- [21] Silicon Graphics, Inc. SGI RASC Technology. www.sgi.com/products/rasc.
- [22] SRC Computers, Inc. Carte programming environment. www.srccomp.com/SoftwareElements.htm.
- [23] SRC Computers, Inc. General purpose reconfigurable computing systems. www.srccomp.com.
- [24] Synplicity, Inc. FPGA solutions. www.synplicity.com.
- [25] Xilinx, Inc. The programmable logic company. www.xilinx.com.
- [26] Xilinx, Inc. New technology and a new way of working. In *History of Xilinx*. www.xilinx.com/company/xilinxstory/history.htm, 2006.

(Left to right) Ken Pathak, ERDC Information Technology Laboratory (ITL) computer scientist; Dr. Gerald R. Morris, ERDC MSRC computer scientist; Dr. Deborah Dent, ERDC ITL Acting Director; Dr. Mark G. Hardy, Interim Dean, School of Engineering, Jackson State University (JSU); and Sheldon Swanier, Director of Strategic Initiative, JSU, March 16



(Left to right) LTC Mike McGuire, Commander, 5th Engineer Battalion; COL James R. Rowan, ERDC Geotechnical and Structures Laboratory; LTC Bill Duddleston, U.S. Army Corps of Engineers; LTC Jeff Anderson, Office of the Chief of Engineers, the Pentagon; Paul Adams, DAAC Lead; and COL Mike Helmick, U.S. Army Training and Doctrine Command, February 28

(Left to right) John E. West, Scientific Computing Research Center Director, ITL; Dr. Deborah Dent, ERDC ITL Acting Director; James C. Dalton and Greg Baer, U.S. Army Engineer Division, South Atlantic; and Tom Richardson, ERDC Coastal and Hydraulics Laboratory Director, February 22





(Left to right) COL Richard B. Jenkins, ERDC Commander; John E. West; Paul Adams; The Honorable Jay M. Cohen, Under Secretary for Science and Technology, Department of Homeland Security, Washington, D.C.; and Dr. Jeffery P. Holland, ERDC Deputy Director, February 20

(Left to right) LTC Mike Wehr, U.S. Army War College and Incoming Deputy Commander, U.S. Army Engineer District, Vicksburg; and Dr. Robert S. Maier, ERDC MSRC Assistant Director, January 5



(Left to right) David Stinson, ERDC MSRC Acting Director; and COL Bill Haight, Director, Office of the Chief of Engineers, the Pentagon, November 16

(Left to right) Greg Rottman, ERDC MSRC Assistant Director; Dr. Johannes Westerink, Interagency Performance Evaluation Task Force (IPET) member and Professor at Notre Dame; and Notre Dame students, November 7



(Left to right) Greg Rottman; BG Bo Temple, Director, Military Programs Directorate, U.S. Army Corps of Engineers, Washington, D.C.; BG Robert Crear, Commanding General, Mississippi Valley Division/ President, Mississippi River Commission, U.S. Army Corps of Engineers; and Dr. James Houston, ERDC Director, October 26

Below is a list of acronyms commonly used among the DoD HPC community. These acronyms are used throughout the articles in this newsletter.

API	Application Programmer Interface	JSU	Jackson State University
CAD	Computer-Aided Design	LCS	Littoral Combat Ship
CFD	Computational Fluid Dynamics	LUT	Look Up Table
CLB	Configurable Logic Block	MHD	Magnetohydrodynamic
CPU	Central Processing Unit	MPI	Message Passing Interface
CVN	Catamount Virtual Node	MSRC	Major Shared Resource Center
DAAC	Data Analysis and Assessment Center	MW	Megawatt
DNS	Direct Numerical Simulations	NAVO	Naval Oceanographic Office
DoD	Department of Defense	NFA	Numerical Flow Analysis
ERDC	Engineer Research and Development Center	PE	Processing Element
FOD	Foreign Object Damage	RAM	Random Access Memory
FPGA	Field Programmable Gate Array	RC	Reconfigurable Computer
GPP	General-Purpose Processor	RDT&E	Research, Development, Test, and Evaluation
GSL	Geotechnical and Structures Laboratory	SHMEM	Shared MEMory
GB	Gigabyte	SPMD	Single Program Multiple Data
HDL	Hardware Description Language	SRAM	Static RAM
HLL	High-Level Language	SST	Shear Stress Transport
HPC	High Performance Computing	TB	Terabyte
HPCMP	High Performance Computing Modernization Program	TI-07	Technology Insertion 2007
I/O	Input/Output	UPS	Uninterruptible Power Supply
IP	Intellectual Property	VOF	Volume of Fluid
IPET	Interagency Performance Evaluation Task Force	VTK	Visualization Toolkit
ITL	Information Technology Laboratory	XOR	Exclusive-Or

training schedule

For the latest on training and online registration, one can go to the User Productivity Enhancement and Technology Transfer (PET) Online Knowledge Center Web site:

<https://okc.erd.hpc.mil>

Questions and comments may be directed to PET at (601) 634-3131, (601) 634-4024, or *PET-Training@erd.usace.army.mil*



ERDC MSRC *Resource* Editorial Staff

Chief Editor/Technology Transfer Specialist

Rose J. Dykes

Visual Information Specialist

Betty Watson

ERDC MSRC HPC Service Center

Web site: www.erdhpc.mil

E-mail: msrchelp@erdhpc.mil

Telephone: 1-800-500-4722

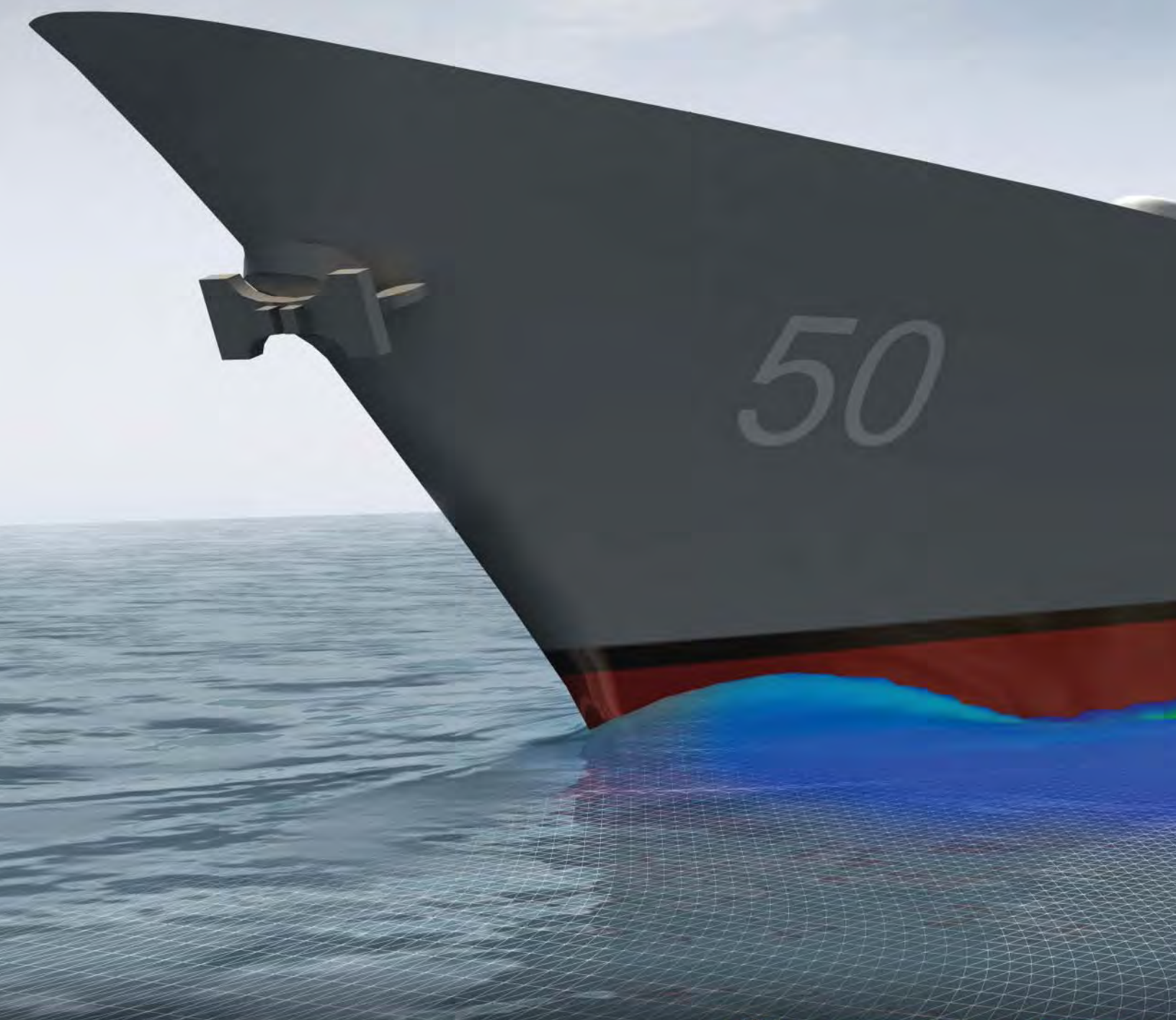
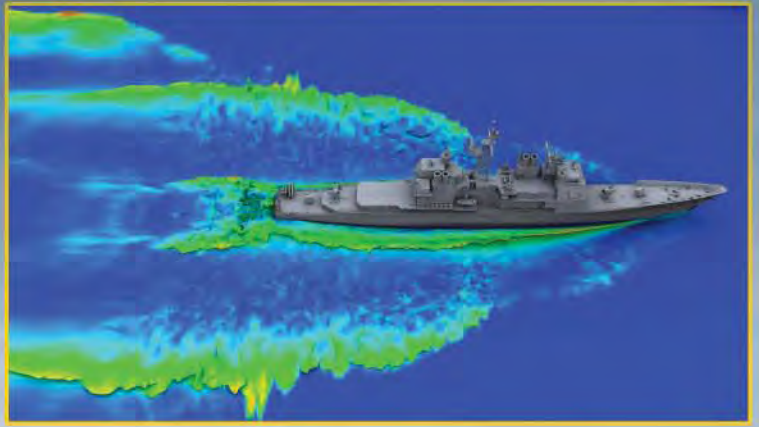
The ERDC MSRC welcomes comments and suggestions regarding the *Resource* and invites article submissions.
Please send submissions to the above e-mail address.

The contents of this publication are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the DoD.

Design and layout provided by the Visual Production Center, Information Technology Laboratory, U.S. Army Engineer Research and Development Center.

Approved for public release; distribution is unlimited.



U.S. ARMY ENGINEER RESEARCH
AND DEVELOPMENT CENTER
INFORMATION TECHNOLOGY LABORATORY

