AHPCRC
Army High Performance Computing Research Center

AHPCRC Bulletin

Volume 2 Issue 2

Distribution Statement A: Approved for public release; distribution is unlimited.

Education and Outreach

INSIDE THIS ISSUE

The Army High Performance Computing Research Center, a collaboration between the U.S. Army and a consortium of university and industry partners, develops and applies high performance computing capabilities to address the Army's most difficult scientific and engineering challenges.

AHPCRC also fosters the education of the next generation of scientists and engineers—including those from racially and economically disadvantaged backgrounds—in the fundamental theories and best practices of simulation-based engineering sciences and high performance computing.

AHPCRC consortium members are: Stanford University, High Performance Technologies Inc., Morgan State University, New Mexico State University at Las Cruces, the University of Texas at El Paso, and the NASA Ames Research Center.

http://www.ahpcrc.org

Ongoing efforts to develop and expand the U.S. Army's scientific and technological resources are key to the effectiveness of the Army's response to continually changing adversaries and environments. AHPCRC contributes to these efforts through its education and outreach program.

Top: AHPCRC Summer Institute. Bottom: PREP.

In addition to supporting faculty, graduate students, and postdoctoral fellows at the partner universities, AHPCRC sponsors a summer institute for undergraduate students, held at Stanford University since 2009. This institute is open to all eligible undergraduates, with special emphasis on recruiting women and under-represented minority groups. The student research presentations from the 2009 summer institute appear in this edition of the *AHPCRC Bulletin*, and more information appears in Vol. 1, No. 4 *(Publications link at www.ahpcrc.org)*.

AHPCRC supports the Pre-freshman Engineering Program (PREP) at New Mexico State University, a hands-on introduction to engineering and computing concepts for regional students, many of whom have had limited exposure to science and engineering courses *(Bulletin Vol. 2, No. 1)*. In this way, AHPCRC works to broaden the pool of young students who are considering science and engineering as career choices.

Three of the AHPCRC partner universities—Morgan State University, New Mexico State University, and the University of Texas at El Paso—are minority-serving institutions, and AHPCRC-funded faculty members participate in their host universities' outreach programs for pre-college students *(Bulletin Vol. 1, Nos. 2 and 3)*.

In 2010, two new outreach efforts were launched, the ARL Summer Intern program and the Hispanic Research and Infrastructure Development Program. Turn the page to read about these programs. ✯

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	2. REPORT TYPE	3. DATES COVERED
2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
AHPCRC (Army High Performance Computing Research Center) Bulletin. Volume 2, Issue 2, 2011	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) **Army High Performance Computing Research Center,c/o High Performance Technologies, Inc,11955 Freedom Drive, Suite 1100,Reston,VA,20190-5673**	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**	**Same as Report (SAR)**	**32**	

AHPCRC Launches Intern Program

Six students have become the first cohort of AHPCRC summer interns at ARL's Adelphi and Aberdeen campuses in Maryland. Three of these students participated in the 2009 inaugural season of the AHPCRC Summer Institute: Alex Sabbatini and Michael Hammersley from Stanford University, and Caraline Murphy from New Mexico State University. Their research reports appear on pages 6, 13, and 10, respectively.

Hammersley is working at Stanford on a collaboration with ARL Aberdeen's Weapons Materials Research Directorate. Sabbatini will work with ARL Adelphi's Micro-Autonomous Science and Technology (MAST) project, and Murphy is working in the Computer Information Systems Directorate with Raju Namburu (ARL Adelphi) and Mark Potts (HPTi).

Also at Adelphi will be Will Brown, a student at Missouri University of Science and Technology who is working with robotics. Justin LaPre and Mark Anderson, both students at Rensselaer Polytechnic Institute, will work with Dale Shires in the Computer Information Systems Directorate in Aberdeen.

Earlier this year, Hammersley and Sabbatini presented their 2009 research results to Mr. John Miller, Director of the U.S. Army Research Laboratory, and Dr. Cary Chabalowski, Acting Director for Research and Laboratory Management, Office of the Deputy Assistant Secretary of the Army for Research & Technology/Chief Scientist, during their respective visits to Stanford University to review the AHPCRC program. *(See related article, next page).* ✯



Left to right: Murphy, Hammersley, Sabbatini.

HRID Program Begins

On May 4, 2010, AHPCRC awarded approximately $1 million in computational science research and infrastructure support contracts to nine Hispanic Serving Institutions (HSI) under a Cooperative Agreement with the U.S. Army Research Laboratory (ARL). The Hispanic Research and Infrastructure Development Program (HRID) supports the education and outreach mandate of AHPCRC.

The colleges and universities funded under this initial award are: Adams State College (Alamosa, CO), three campuses of California State University (Long Beach, Northridge, and Stanislaus), Donnelly College (Kansas City, KS), Heritage University (Toppenish, WA), New Mexico State University (Las Cruces), Northern New Mexico College (Española), and St. Mary's University (San Antonio, TX).

HRID supports the institutional and infrastructure necessary to foster the next generation of Hispanic computational scientists and engineers. The program funds computer equipment and student research projects at the participating universities. The overarching goals of these programs are to:

- Improve educational opportunities for Hispanic students,
- Increase participation in computational science and computer science programs and related curricula, and
- Increase the resource pool of well-trained scientists and engineers who use computer-based modeling and simulation techniques.

The HRID Program is administered by High Performance Technologies, Inc., the Reston, VA-based technology services company that manages the infrastructure and administration of the AHPCRC consortium. A second set of awards under this program is anticipated for late 2010. ✯

Chabalowski, Miller Review AHPCRC Program

AHPCRC managers and researchers presented highlights of the program to leaders from the Army's research laboratories during two review sessions held at Stanford University. In January, Mr. John Miller, Army Research Laboratory Director was accompanied by several representatives of the ARL's Sensors and Electron Devices Directorate (SEDD). They were welcomed by Dr. Jim Plummer (Dean, Stanford School of Engineering), Dr. Charbel Farhat (AHPCRC Center Director), and Dr. Raju Namburu (AHPCRC Cooperative Agreement Manager). Miller summarized the research mission and goals of ARL. Primary investigators and student researchers from each of the four technical areas, representing Stanford, New Mexico State University, and the University of Texas at El Paso, gave presentations on their projects, and students in TA3 gave a short demonstration during the working lunch. Barbara Bryan, AHPCRC Research and Outreach Manager, summarized the AHPCRC Outreach program, and two Summer Institute Students presented their research. Computer scientist Will Law summarized features of the AHPCRC computer systems infrastructure. The meeting ended with a tour of the Stanford facilities.

Top: Dr. Cary Chabalowski, Bottom: Mr. John Miller

In March, Ms. Jill Smith, director of ARL's Weapons Materials Research Directorate, visited Stanford for a similar briefing, accompanied by Dr. Cary Chabalowski, Acting Director of the Research and Laboratory Management Office, Deputy Assistant Secretary of the Army for Research and Transition. Also attending were researchers from some of the Army's research laboratories, several of whom collaborate with researchers from the AHPCRC consortium. ✯

Farhat Receives AIAA Award

AHPCRC Center Director Charbel Farhat was honored by the American Institute of Aeronautics and Astronautics (AIAA) at the April 2010 AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference in Orlando, FL. Farhat received the AIAA Structures, Structural Dynamics, and Materials Award for pioneering research in fluid-structure interaction and its application to critical aeroelastic and engineering problems. The AIAA Structures, Structural Dynamics, and Materials Award is presented to an individual who has been responsible for an outstanding sustained technical or scientific contribution to the field of aerospace structures, structural dynamics, or materials. ✯

Attendees at the January 2010 Review
Mr. John Miller, Director, ARL
Dr. John Pellegrino, Director, SEDD/ARL
Ms. Patricia Fox, Contracting Officer
Dr. Romeo Del Rosario, Chief, Electronics Technology Branch, SEDD/ARL

Attendees at the March 2010 review
Dr. Cary Chabalowski, Deputy Assistant Secretary of the Army
Ms Jill Smith, Director, WMRD/ARL
Mr. Charles Nietubicz, Chief, Advanced Computing & Computational Sciences Division, ARL
Dr. Paul Amritharaj, Chief, RF & Electronics Division, ARL
Dr. Jeffrey Singleton, ASA ALT
Dr. Volker Weiss, ARL/ODIR
Dr. Mostafiz Chowdhury, ARL/ODIR
Dr. Mark VanLandigham, WMRD
Dr. Mick Maher, WMRD
Dr. Steven Bilyk, WMRD Dr. Jubaraj Sahu, ARL
Dr. Madan Dubey, ARL
Mr. Todd Turner
Dr. Richard Jow, Electro-Chemistry Branch, SEDD/ARL

Distribution Statement A: Approved for public release; distribution is unlimited.

Micro Air Vehicle Modeling

Ricardo Medina (New Mexico State University)
Mentors: Charbel Bou-Mosleh, Charbel Farhat

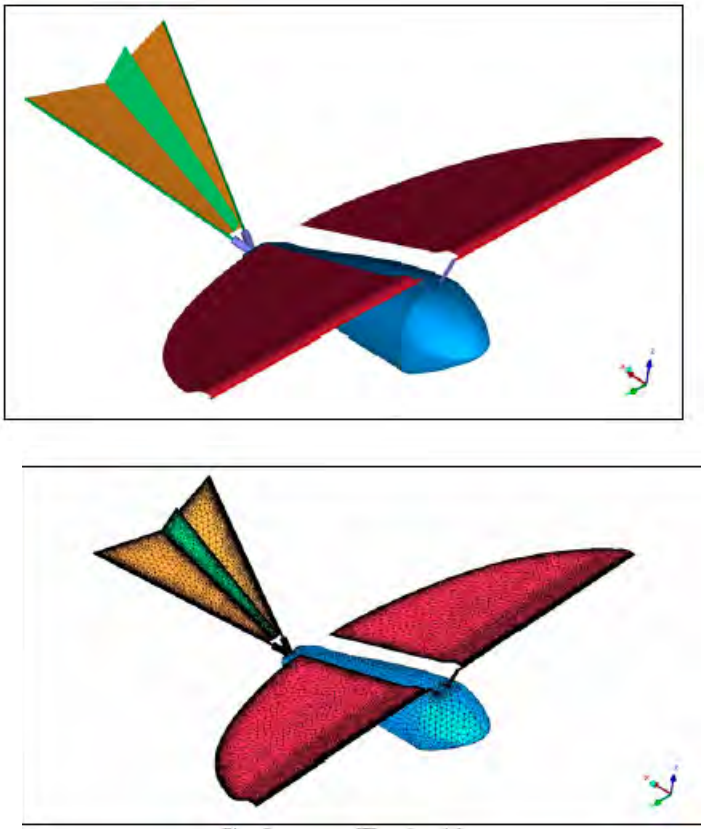
An Unmanned Air Vehicle (UAV) is an aircraft that can be controlled from a remote location or fly autonomously based on preprogrammed flight plans. A Micro Air Vehicle (MAV) is a type of UAV that is much smaller in size, just a few inches (approximately 6"). The micro air vehicles can perform a wide variety of functions. Some of these functions are remote sensing; this is central to the reconnaissance role most UAVs fulfill and that are too dangerous for a manned aircraft to perform. The MAV models usually have fixed wings, flapping wings or helicopter-like wings.

The objective of the project is to build a Finite Element Model (FEM) and a Computational Fluid Dynamic (CFD) model. Using ICEM–CFD, a CAD model for the FEM and for CFD were built. The mesh generator of the software was used to create the required FEM and CFD meshes for the flapping-wing model. The following step was to carry out simulations using the AERO code [developed by Charbel Farhat's group at Stanford]. The lift and drag will be predicted and the surface pressure will be computed and visualized using TOP-DOMDEC visualization tool.

The geometry of the CFD model was created to closely represent the actual flapping-wing toy. The toy's measurements were provided and used to build its CAD model, which took several days due to the complexity of the geometry *(top, figure at right)*.

The CFD mesh was then built using the mesh generator from the ICEM-CFD software. The size of the elements for the surface mesh had to be specified for Fig. 1 CAD model of the CFD each part. For example, very small sizes were specified on the leading edge of the wing to accurately represent its curvature. Larger sizes were used on the remaining wing surfaces due to the flatness of the faces *(bottom, figure at right)*.

After finishing the CFD model, building the FEM required a different geometry. The wings, the horizontal tail and the vertical tail are represented by only a single surface located midway to the top and the bottom surfaces of the CFD model. The small cylinders of the leading edge of the wings and the tail, the fuselage and the connectors of the wings to the fuselage and the tail to the fuselage are represented by lines connected one after the other located in the center of each part. When the mesh for the finite element model was created, the surfaces were represented by shell elements. The lines representing the cylinders, fuselage and connectors were specified as beam elements *(top figure, page 5)*.



Top: **Geometry of the CAD model used for the CFD study, designed to simulate the flapping wing toy used in this study.**
Bottom: **Mesh model for the CFD study.**

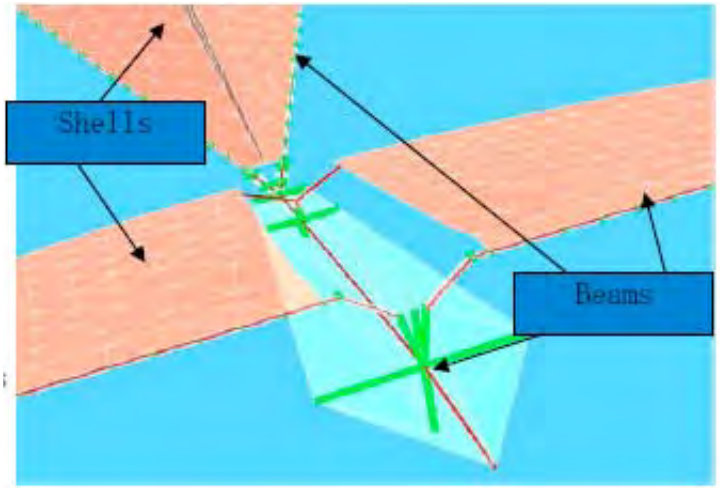
After building the FEM mesh, an "input" file was created using ICEM-CFD software. This input file contains the mesh information such as nodal coordinates and elements topology. It also contains the elements' material properties and the information required to model the hinges. Extensive research of different kinds of plastics and their common usage determined the decision of choosing nylon 66 for the beam elements

and polyethylene terephthalate (PET) for the shell elements. The material properties are listed below:

	Nylon 66	PET
Young's Modulus (GPa)	5.55	4.05
Poisson Ratio	0.43	0.43
Density (g/cc)	1.465	1.49

An eigen-analysis of the FEM model was performed to compute its natural mode shapes and frequencies. The analysis was done using the AERO-S code developed at Stanford University. This analysis was used to verify that the different parts of the model are properly attached to each other and that no extra nodes exist. The different mode shapes are visualized using the TOP/DOMDEC visualization tool developed at Stanford University as well. Using a scientific strobe light, the first dominant vibrating mode of the actual toy is visualized and compared to the computed one. As can be seen in the photograph below, the experiment and the computed result show a good agreement.

The CFD model was used to perform a steady state simulation using the AERO-F code developed at Stanford University. In this simulation, the viscous effects were ignored and the wings were left fixed (i.e., no flapping). The simulation was performed at sea level conditions with a pressure of 101 kPa and a density of 1.23 kg/m^3. The air speed was set to 1m/s which corresponds to a Mach number of $3e^{-03}$. The angle of attack



The finite element model is a simplified representation, using shells and beams to represent structural elements.

was set to zero because the wings of the flapping wing toy are already on an angle of attack close to 17°. Some of the results obtained from the steady state simulation were lift, drag, pressure and velocity. The lift obtained was small, and this is due to the fact that wings were not flapping.

Simple understanding of Computational Fluid Dynamics were needed in building the CFD model of the MAV. For example, in order to obtain accurate results in a simulation of lift, drag, pressure and velocity of an aircraft by the interaction of the air, a proper CFD model has to be created. Some knowledge of finite elements was required on building the FEM model and for studying the structural integrity of the model. ✯

First dominant vibration mode for the MAV toy and the computational model.

Distribution Statement A: Approved for public release; distribution is unlimited.

Flutter Prediction for the F-16 Block 40

Alex Sabbatini (Stanford University)
Mentor: David Amsallem, Charbel Farhat

Flutter is a phenomenon of vibrations in the structure of an aircraft that risk amplification and threaten the aircraft with failure. Computational methods that accurately predict this unsteady phenomenon provide a useful alternative to empirical testing via scaled wind tunnel models. However, the conventional methods involved in determining these solutions are computationally expensive (on the order of hours). Reduced order methods based on snapshots (solutions) of the flow can be used to reduce the computational cost. A reduced basis is generated from these snapshots and the reduced-order solution is computed as a linear combination of these basis vectors. Computing an accurate basis is critical to retrieving an accurate solution and has to be done for each flight condition (Mach number, angle of attack). The reduced-order basis (ROB) is denoted $\Phi(M, \alpha)$ in this report.

Interpolation between a small set of pre-computed local ROBs (usually 4 or 5) can be used to construct a new fluid ROB with the desired (M, α) parameters. The algorithm used in interpolating a new fluid ROB is more favorable to building a $\Phi(M, \alpha)$ from scratch because it is much faster and delivers excellent to acceptable results. The two goals of this study are to
1) Provide an insight on the variation of the flow in the parameter space of flight conditions
2) Design a GUI that facilitates the selection of points intended for aeroelastic analysis.

Visualizing Distances Between ROBs

The distance between the subspaces described by the ROBs provides insight on the variation of the flow and as such can be used to detect the nonlinearities in the flow across a large spectrum of flight conditions.

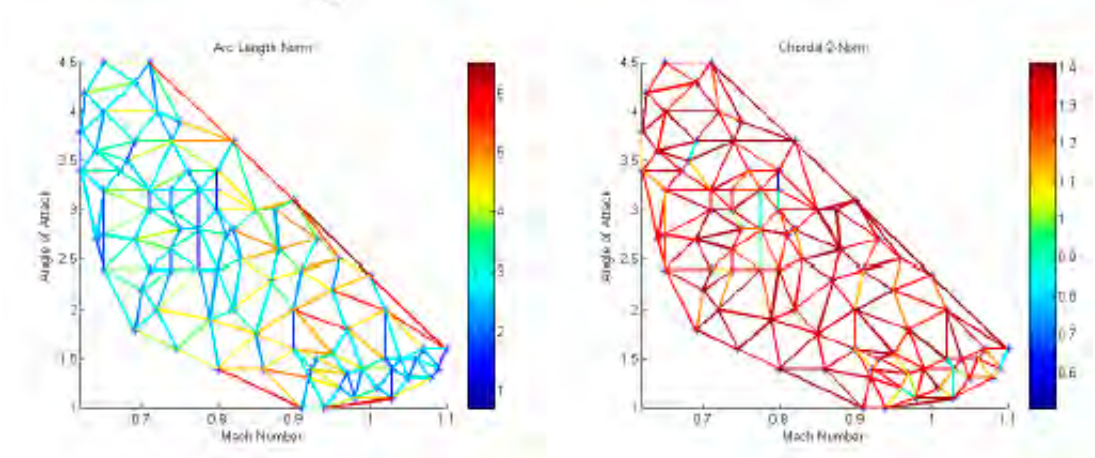
Distance between subspaces. Denote the ROBs of the two subspaces that one wants to determine the distance between as $\Phi_1(M_1, \alpha_1)$ and $\Phi_2(M_2, \alpha_2)$. To find a distance between these two subspaces, one can implement any predefined vector norm using $[\theta_1 \ldots \theta_n]$ as the vector's components. These angles are computed using Algorithm 1:

Algorithm 1: Determine Theta Values for Norm

$$R = \Phi_1(M_1, \alpha_1)^T \Phi_2(M_2, \alpha_2)$$
$$[U, \Sigma, V] = \text{SVD}(R)$$
$$\Theta = \arccos(\Sigma)$$

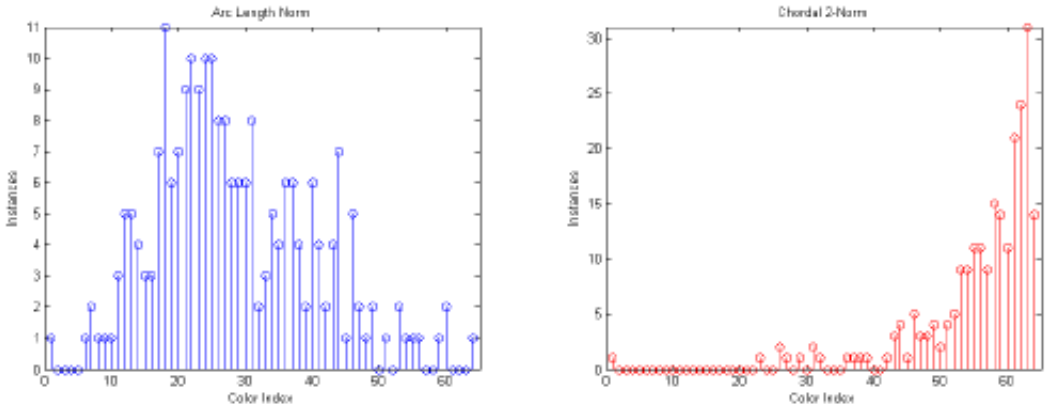
Delaunay triangulation. A Delaunay triangulation can provide an automated method to selecting pairs of nearby vertices. For each pair a distance is computed according to the method outlined above. This distance is reflected in the color of the line connecting these two points. MATLAB is used to take the Delaunay triangulation of the normalized data. If there are 83 vertices, and the Delaunay triangulation creates 145 triangles from those vertices, then Euler's formula can be used to determine the number of edges involved:

$$v - e + f = 2$$



Distance between sub-spaces, plotted as angle of attack vs. Mach number. *Left:* arc-length norm (Euclidian-based). *Right:* chordal 2-norm (infinity-based).

Histograms of frequencies of occurrence for the colors in the figures on previous page.



This returns 227 edges (because the faces include the number of triangles plus the additional infinitely large outer region). Thus 227 computations need to be made if the Delaunay triangulation will be followed as a guide to connecting neighboring points.

Decomposing the Delaunay triangulation. A Delaunay triangulation may be decomposed into groups of an arbitrary number of adjacent vertices.

A problem in conducting these 227 edge computations rests in the limitation on the number of matrices that can be loaded onto the RAM at a given time. Each matrix is quite large and requires 1.6GB of memory to store. For the current configuration, it was empirically shown that the maximum number of matrices that can be loaded at a given time is six. This limitation calls for an algorithm which can efficiently assign groups of six vertices such that a minimal amount of matrices (ROBs) must be loaded and freed from memory but computations for every edge can be carried out. A new algorithm is created to meet these conditions. It outputs a list of the order in which these matrices should be loaded from memory, computed, and then removed from memory.

The algorithm assumes that there are no duplicate points and that the data set is regular in the sense that a Delaunay triangulation can be found. The algorithm begins by locating a unique point on the convex hull that corresponds to the maximum valued data point in the "Angle of Attack"-dimension. Disputes that arise from points that are both at the maximum value are resolved by choosing the point with the lowest "Mach number." Triangles that include this point as one of their vertices are registered into a group. Because of the limitation of six vertices, the maximum number of adjacent triangles that can be chosen from this group is four. This is not a problem if the group contains four or fewer triangles.

The vertices of the chosen triangles are first written to an instructions file under "Loads" if their corresponding matrices haven't already been loaded from memory. After the loads, the "Computes" are written (which are simply the unique edge pairs). Following the computations, the vertices that are no longer required for future computations are written to the file under "Delete," as the memory needs to be freed for other matrices. Once this is accomplished, the algorithm removes the initial point that was found on the convex hull from consideration as well as the triangles that were chosen.

The algorithm then repeats itself by finding another point on the convex hull. This is repeated until the Delaunay triangulation is exhausted, at which point the instructions file is complete and ready for use. These instructions are read in by another program that outputs the $[\theta_1 \ldots \theta_n]$ for any two vertices connected by the Delaunay triangulation.

Results

Several predefined norms were selected to calculate the distance between two given subspaces but as it hap-

continued on page 8

Flutter Prediction

continued from page 7

pens, only two variations of the results were created. The Euclidean-based norms make up one category of similar results and the infinity-based norms make up the other. The former category produces more meaningful results as the number of instances each color is used follows a nice distribution that takes advantage of the full color map. Plots of the arc length norm (Euclidean-based) and the chordal 2-norm (infinity-based) along with their color distributions are found in the figures on pages 6 and 7.

Interpolation GUI

A GUI was created to make it easier for the user to select points at which aeroelastic analysis will be carried out. For that purpose a corresponding ROB is computed using a computationally inexpensive interpolation procedure. The main function of the GUI is to record the user's selection of parameters and write five new files according to five separate templates. These new files reflect the user's choice of Mach number, Angle of Attack, and Number of CPUs and are also stored in a directory unique to these choices. Configurations can be inputted either manually or via mouse selection. The GUI makes a few assurances about the configurations selected for interpolation (let P denote the set of predefined flight conditions): the parameters lie within one of the cells of the Voronoi diagram of P, the parameters are not duplicates of P or any other parameters already chosen in the current session, and the user chose values for all three fields. The GUI's extra functionalities include a component to load six separate visualizations of P and a history table that includes past configuration entries. ✯

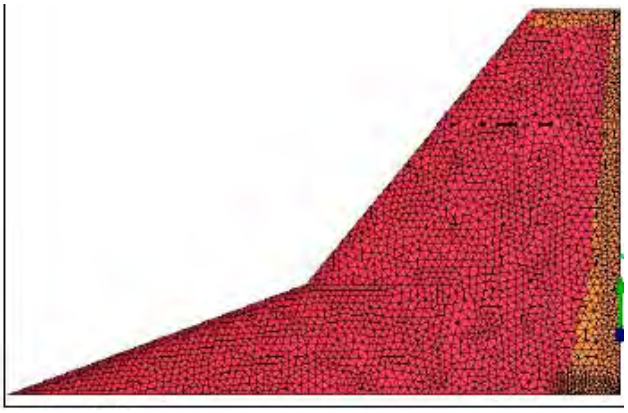
The Aerodynamic Analysis of a Damaged Wing

Samir Patel (Harvard University)
Mentors: Charbel Bou-Mosleh, Charbel Farhat

Combat damage is an ever-present threat to military operations. This project focused specifically on damage sustained in an airborne setting and the effects of that damage on the survivability of an aircraft wing. Advances in computational fluid dynamics have allowed the effects of airborne damage to be studied through computer simulations. This progress has rendered the aerodynamic testing much more cost-effective in that computer simulations eliminate the necessity of wind tunnels and other expensive inputs As an added benefit, the simulations can be manipulated, allowing for greater maneuverability.

Procedure

The project was composed of several stages. The first involved the planning and design of the damaged wings. The geometries for the different damage scenarios were created and meshed using the commercial mesh-generation software Ansys ICEMCFD. The purpose of the meshes was to decompose the wing into small regions, triangles for surfaces and tetrahedrons for volumes, for the prediction of fluid flow around the wing. The meshes were then split into subdomains using the AERO suite of codes developed at Stanford University. The simulations were run using the code AERO-F, which is massively parallel. TOP/DOMDEC, a visualization tool developed at Stanford, Microsoft Excel, and MATLAB were used to post-process the results.



Final mesh of High-Speed Civil Transport Wing.

The wing used for the simulations is classified as High Speed Civil Transport, a type employed by the Concorde jet. The chord lengths at the root and the tip were 155.96 in. and 22.44 in. respectively, while the

length of the wingspan was 96.17 in. The mesh sizes of the wings varied between 70,000 and 184,000 nodes; the number of nodes increased from 70,000 nodes as the complexity and the amount of damage grew.

The wing was subjected to damage that varied in positioning, quantity, geometrical shape, and angle. In Group 1, holes with a 5-in. radius were placed at strategic midpoints on the leading edge, the root, the trailing edge, and the tip. In Group 2, the number of holes on the wing ranged from one to four with each at a different position. In Group 3, the hole at the leading edge was varied in shape from a circle to a triangle, 5-point star, a 9-point star, and a circular hole with 11 petals (of thickness 0.04 in.) on its upper and lower rims, for the examination of fragmentation effects *(see www1.nasa.gov/centers/dryden/about/Organizations/Technology/Facts/TF-2004-12-DFRC.html)*. In Group 4, the angle of the hole at the leading edge was varied from –30° to 90° by using the intersections of the wing and a 5-in. radius cylinder incident at the angle.

The simulations were run under similar conditions. The environmental characteristics were set at an altitude of 10,000 ft, an air pressure of 10.083 lb/in^2, and a density of 8.46 $\times$ 10^{-8} slugs/in^3. The airflow in these simulations had two major characteristics: it was kept steady and its viscous effects were ignored. In addition, the airflow was given a mach number of 0.6 and the wing was positioned at an angle of attack of 1 degree.

The code used to run the simulations, AERO-F, was developed at Stanford University. To obtain the results, the Euler equations were solved. However these results may not be entirely accurate due to certain parameters of the project; for example the meshes of the wings were very coarse. Greater accuracy would have demanded much finer meshes.

Results

The effects of the damage were quantified using the undamaged wing as a baseline for comparison. The values achieved through testing of the undamaged wing were 2805.75 lbs and 72.56 lbs of lift and drag,

Simulated lift for a wing with a hole in the leading edge. With every iterative segment, the solution became more accurate, reaching a steady state oscillation around the correct value.

respectively. The various lift and drag values for each damage condition were then compared using ratio of the respective lift and drag values to the baseline. The results were as follows:

Group 1: The results varied by position. The holes on the leading edge and root reflected lost less than 0.5% of lift but increased drag by nearly 40%. The hole on the trailing edge lost almost 7% of lift and increased drag by 3.7%, while the hole on the wing tip lost close to 2% of lift but increased drag by about 18%.

Group 2: The results were consistent with the logic of losing lift with the loss of surface area. With 2 holes the wing lost less than 1% of lift, but with the additions of the third and fourth holes the wing lost almost 7% additional lift. With 2 holes the drag force on the wing had increased by 85%, but with the additions of the third and fourth holes the drag increased by about 29%.

Group 3: The lift loss appeared to relate to the shape of the hole The triangle and the five-point star lost close to 6% of lift, but the 9-point star lost more than 10% of lift and the hole with the petals lost more than 7% of lift. By comparison the circular hole at the leading edge lost less than 0.5% of lift. The drag did not seem to vary as much, however. The triangle and the stars all increased drag between 140% and 150%, showing signs of consistent increase in drag regardless of the number of edges. However, the hole with petals

continued on page 10

Damaged Wing
continued from page 9

increased drag by more than 250%. In contrast, the circular hole increased drag by about 40%.

Group 4: The angles did not reflect a consistent trend. The hole at –30° lost just over 4% of lift and the hole at 90° lost less than 0.5% of lift. However, the holes at 30°, 45°, and 60° all lost between 8.2% and 9.2% of lift. An opposite trend was somewhat reflected in the quantification of drag. The –30° and 30° holes increased drag by just over 140%. The 45° and 60° holes increased drag between 160% and 170%, serving as the upper threshold of the angle results. The 90° hole only increased drag by about 40%.

Conclusion

Hole positioning seemed to have a significant impact on wing performance; trailing edge damage appeared to affect lift more than drag, where as the root and leading edge seemed to impact drag more than lift. The different impacts may be regulated by wing thickness, since the root and leading edge regions are thicker than the trailing edge region. Straight-edged shapes seem to have a greater impact on lift and drag than circular holes; the emergence of such figures above surfaces as petals can dramatically increase drag. Angular holes also seemed to have a significant impact on lift and drag but their effects, in comparison to perpendicular circular holes, appeared to vary with the angle value. The shape and angle of a hole will make a difference in the outcome of the results, as opposed to a standard cylindrical hole.

Acknowledgment

Special thanks to Stanford University and the United States Army for their support in making this research project possible! ✯

Modeling Differing Structural Fabrics in Ballistic Shields

Brandon Moultrie (Morgan State University) and Caraline Murphy (New Mexico State University)
Mentors: David Powell and Charbel Farhat

Ballistic fabrics possess a complex micro-scale structure that involves thousands of fibrils bonded together to form yarns, which are then tightly woven to create fabric sheets. Because experimental ballistic testing is both expensive and can be dangerous in field, computational simulation methods are a much simpler and cost-effective approach as opposed to real-world testing. The purpose of this research path is to develop and test different weaving patterns and their responses to dynamic bullet penetration in order to find superior, cheaper and lighter-weight multi-scale fabric designs.

For this project, three weaving patterns were analyzed. One pattern focused on is the standard grid pattern, which is the weave used in most ballistic fabrics. The other two weave designs analyzed are the diagonal grid pattern, which is the standard grid rotated 45 degrees, and the spider-web pattern weave. Each fabric pattern has advantages and disadvantages, but the determined advantages found show that two designs, the rotated grid and spider-web, have an extra edge towards increasing energy absorption of an impact in comparison to the standard grid design.

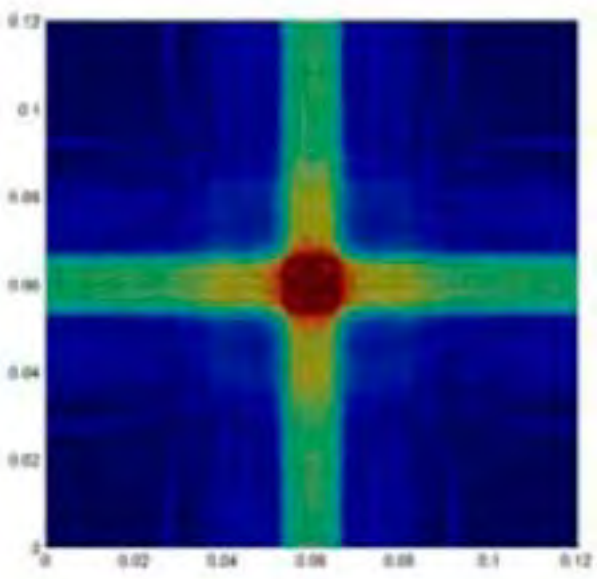
Laboratory fabric mount setup for ballistic fabric testing.

One advantage of the standard grid pattern is that it enables the fabric's strength to be uniformly distributed throughout the entire material. Whether it experi-

ences impact in the middle or towards the edges of the material, the protective strength is consistent. However, a disadvantage of this design is that the network of fibril weaves does not allow many fibrils outside of the impact area to support the fibrils within the area of impact. Previous testing has shown that the majority of the impact stress is distributed horizontally and vertically, because those are the two directions in which the fibrils are aligned *(figure at right)*. This alignment causes the force of impact to be distributed only to the fibrils directly above, below, and to both sides of the impact zone. Therefore, this design does not utilize a significant portion of fibrils on the grid to absorb energy from the impact.

In attempt to compensate for the grid pattern's disadvantage, the idea was proposed to rotate the mesh forty-five degrees so that the fibrils are aligned diagonally. This would distribute the force of impact in all four forty-five degree directions, thus utilizing the entire fibril network, not just the fibrils solely in the impact area. This would then contribute to the intended increase in strength of the mesh while simultaneously preserving its nearly homogeneous strength across the entire mesh. One disadvantage, however, is that the diagonal weave is still unable to utilize many of the fibrils outside of the impact area.

The spider-web mesh design, although not the most practical of design schemes, has one significant advantage over both of the grid patterns. This design enables the force of impact to be evenly distributed over almost all of the fibrils throughout the mesh and in all directions, as the stress moves out in a radial pattern. This design also has one significant disadvantage. Its major disadvantage is that its strength is not uniform throughout the entire mesh. A spider-web mesh is designed to provide maximum impact protection at the center of the weave. Impact protection toward the outside of the weave is very poor, because fibril length increases linearly with each added circle. This design is ideal if one wants extra protection to be provided over a certain area, for example in areas that will cover vital organs in a bulletproof vest or surrounding vital machine and mechanical parts in an aircraft. Because the spider-web weave does not currently exist for ballistic fabric, only computational simulations have been used to analyze the response of this weave design to dynamic projectile impact.



Stress along a standard grid pattern is distributed along the directions of the fibrils.

Experimental tests are extremely important and can factor in real-life variables that cannot be simulated in computational tests. Fortunately, significant trends indicate that the computational tests of the given fabric weaves react virtually identically to computational simulations.

In simulation, the grid pattern reacted to impact by stressing along the ninety degree axes and experiencing breakage along the middle edges of the standard grid fabric, which is visible in experimental tests. Interestingly, the calculated velocity data from the simulations showed less of a velocity decrease than in actual tests. In the deformation of the spider web with use of velocities of 100, 150, 200 and 250, across the board readings showed a decreased velocity range between 10 m/s and 12 m/s. Similar result ranges were found for the other fabric weaves with the same given velocities, however, the spider web showed the largest decreased velocity range in comparison to the other two weave patterns. Using the velocity data calculated in the simulations, data points were found for a comparison

continued on page 12

Ballistic Shields

continued from page 11

of the initial velocities versus the residual velocities. To get the most accurate residual velocity, the last ten velocity readings in the simulation were averaged together for both the velocities of a node on the front end of the projectile and the back end of the projectile. In most cases, the velocity readings were almost identical for the front node and the back node of the projectile. This was done to give the most accurate end velocity reading of the entire projectile in attempts to prevent inaccurate data caused from oscillations caused by contact with the fabric.

The velocity data was then used to compare the initial energy and the overall energy change in the projectile by calculating the initial energy using the kinetic energy equation and then calculating the residual velocity of the projectile to calculate the energy of the projectile after fabric penetration. In the simulation, the energy changes comparable to the standard grid pattern were far smaller than the percentages found in the experimental results which will be discussed later in this paper.

In order to validate computational and numerical results for fabric reactions, projectile velocity decrease and energy loss, and absorption using accurate simulations, actual tests were performed at the ballistics lab at UC Berkeley. Small sheets of Zylon were cut from a large roll and held tight in a bolted square metal frame *(figure, page 10)*. The metal frame is attached in a vertical position to a triangular support mount positioned in such a way that impact will occur on a predetermined location on the fabric target. The determined test impact location was established to be the center of the fabric visible within the frame. Using compressed air, a designated level of pressure was applied to the custom-built powder gun with a barrel length of 1.6 mm and inside diameter of 12.9 mm, to shoot the steel cylindrical projectile. The initial velocity of the projectile was determined by the time it takes the projectile to cross two parallel laser beams 1.5 m in front of the fabric target. The final velocity was determined by the use of a high-speed digital video camera, which captures 16,000 frames per second, to record the projectile being fired at the mounted fabric.

Several tests were performed for a single Zylon sheet, multiple Zylon sheets, a single diagonally-rotated Zylon sheet and multiple diagonally-rotated Zylon sheets. Data recovered found that the rotated Zylon sheets absorbed more energy. These results were largely anticipated because of the stress diagrams of the grid weaving, which showed increased stress along the 90° axes *(figure, page 11)*, assuming the center position to be designated as point (0,0,0). When rotated, the axes rotate 45°, and thus the distance to the outside of the fabric corner is increased, changing the overall stress patterns. Because the displacement moved in a straight path along the vertical sheet, the path was the shortest distance, unlike the corners, where a step displacement occurs requiring more time to reach the edge.

The average change in experimental velocity for a single vertical fabric sheet was 37.6 m/s; for two vertical fabric sheets it was 46.6 m/s; for a single rotated fabric sheet it was 47.4 m/s; and for two rotated fabric sheets it was 84.4 m/s. In analysis of the results, the double of the velocity value for two rotated sheets in comparison to one is logical. It can be assumed that if one sheet decreases the velocity by a certain percentage, then two sheets will decrease it by twice that percentage. Interestingly, the two vertical sheet test only decreased the velocity by an additional 10 m/s, and only 30% of the single sheet velocity. This is most likely the result of human error. One can best assume that inconsistent attachment of Zylon to the metal frame and incorrect data calculations from camera readings most likely provided these errors.

Overall, the diagonal grid pattern energy absorption was about 46% larger than the standard grid absorption and the spider web grid absorption was about 101% better than the standard grid absorption. These values were determined by calculating the initial energy and using the residual velocity of the projectile to calculate the energy of the projectile after fabric penetration. One can then calculate the difference between the two values as the energy absorbed by the fabric and thus, the energy change of the projectile.

These results show that both non-standard grid fabric designs are better at absorbing the energy of the projectile and thus decreasing force felt by the object utilizing the protective fabric more quickly than the standard grid. Therefore, one will hope that fewer sheets will be required for body and shield armor in order to decrease overall mass and create cheaper armor while still achieving similar if not better energy absorption than current standard ballistic shields. ✯

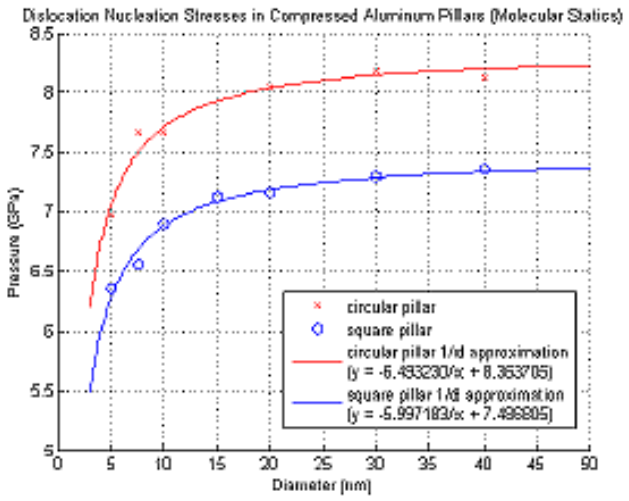
Characterizing High-Strength Nanoscale Gold and Aluminum

Michael Hammersley (Stanford University)
Mentors: Christopher Weinberger, Sylvie Aubry, Wei Cai

For much of early architecture, builders could increase the maximum height of a building largely through innovations in design. Creating the vast skyscrapers of today, however, required a revolution in materials—but for the introduction of steel as the primary structural material, buildings like the TransAmerica pyramid would be virtually impossible. To achieve further jumps in strength, materials must be re-invented once again, and the next logical step is engineering at the nanoscale. One recently discovered nano-material is nano-twinned copper, which is both very strong and highly ductile *(1)*. An immediate consequence resulting from this combination is an extremely high modulus of toughness (i.e., ability to resist fracturing), which makes it a strong candidate for armor, among other uses. Exploration into how surfaces, grain boundaries, and other interfaces behave at these small scales is bound to lead to continuing discoveries of novel and useful properties *(2)*.

Background

Small-scale pillars have been a subject of interest ever since it was discovered experimentally that, at relatively large scales (measured in μm), smaller pillars required higher stresses to compress them to failure *(3)*. Frequently, these higher stresses were orders of magnitude above the maximum stress of the bulk material. Assessing the strength of even smaller pillars, however, is beyond current empirical capabilities, and so research has looked to computational modeling to test pillars whose diameters can be measured in nanometers (nanopillars). Preliminary research on gold nanopillars postulated that, below a critical diameter, nanopillars grow weaker as they grow smaller *(4)*. However, they did not take into account gold's phase transitions, and as such, this phenomenon requires further study. Furthermore, despite modeling pillars both as cylinders and as square-prisms, there has been surprisingly little investigation into the effects of the pillar's shape on its behavior. We had two major questions, then: firstly, whether smaller nanopillars are, in fact, weaker; and, secondly, how the shape of nanopillars affects their behavior.



Stress-strain curves for square-prism and cylindrical pillars. At the nanoscale, smaller pillars reach failure at lower stresses.

Method

To achieve these ends, we simulated displacement-controlled compression of gold and aluminum nanopillars with a 2:1 aspect ratio and ranging in size from 5 to 40 nm (as measured by diameter for circular pillars and by edge-length for square pillars) using the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) software *(5)*. Pillars had either square or circular cross-sections, and both molecular statics (0K) and molecular dynamics (300K) simulations were run. Periodic boundary conditions in all dimensions were

continued on page 14

Nanoscale Gold and Aluminum
continued from page 13

enforced, though enough space was left around each pillar to avoid self-interference. Strain rates for the molecular dynamics simulations were 10^9 s^{-1} up to strains of approximately 15%.

Modeling an aluminum 40nm square pillar in a molecular dynamics test involved running 192 processors for 48 hours to simulate the motion of 7,762,392 atoms for 150,000 timesteps.

Results and Analysis

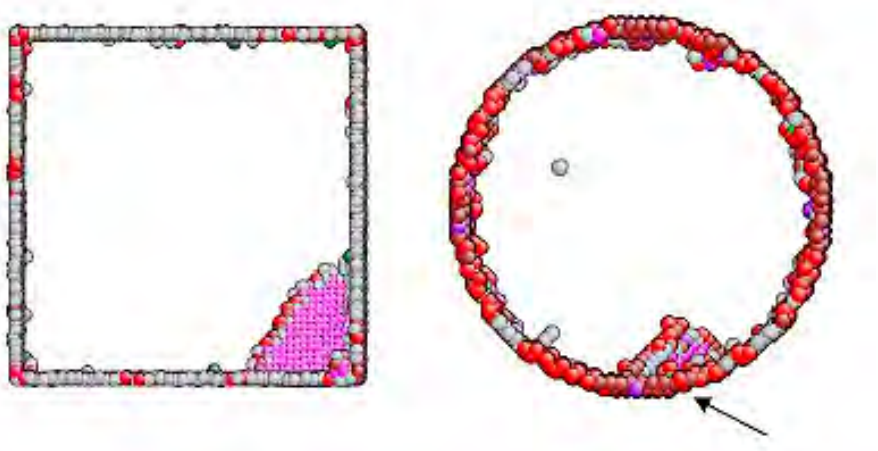
In each run, we measured the stress and longitudinal strain and then analyzed the data to determine at what stress and strain the nanopillar failed. For the gold simulations, we could also determine the stress and strain at which each nanopillar would undergo a phase transformation. After each run, we could create figures and videos to see how the pillars failed, which gave us insight into how the dislocations nucleated and propagated.

Stress-Strain Curves. The statics gold runs were inconclusive, since the differing phase transformation stresses can somewhat mask the true dislocation nucleation behavior. In the gold dynamics simulations, however, the phase-transition stresses and the dislocation nucleation stresses were nearly identical, and the circular pillars were approximately 15% stronger than the square pillars. These results were confirmed by the aluminum statics and dynamics runs, in which square pillars under compression consistently failed at lower stresses than did their circular counterparts *(figure, p. 13)*. In the 0K simulations, aluminum circles were approximately 10% stronger than their similarly-sized square counterparts, and they were approximately 20% stronger in the 300K simulations.

Dislocation Dynamics. To see how dislocations form and propagate inside a solid pillar, we want to visualize only those atoms involved with the dislocation. This visualization is achieved through the use of a centrosymmetry parameter, a measure of how close an atom is to locally being part of a perfect crystal *(6)*. Those that are too close are made invisible, which leaves the surface atoms and dislocations readily visible.

In both the square and circular cases, dislocations clearly nucleate at surface facets—in square pillars, at the corners, and in circular pillars, at one of the many surface facets *(black arrow, figure below)*. After nucleating at the surface, dislocations propagate throughout the pillar. This holds for both circular and square pillars and confirms what is in the literature *(7, 8)*.

We suspect that this phenomenon explains, in part, why square pillars are weaker. It requires less energy to nucleate a dislocation at a right-angle, as in a square, than it does at an angle approaching 180°, as in a circle. We are in agreement with Marian et al. *(4)*, then, in that, below a critical diameter, higher surface stresses start dominating over volumetric effects so that this energy is reached more easily in the smaller nanopillars.



Top views of a 10nm square and a 10nm circular nanopillar at the beginning stages of failure. Coloring is by number of nearest neighbors. Only those atoms which break local fcc symmetry are shown.

Conclusion

We ran displacement-controlled compression simulations on gold and aluminum to determine whether strength continues to increase with decreasing size at the nanoscale. We found that, below a certain diameter, pillars actually grow weaker as they grow smaller. We confirmed that dislocations nucleate at a surface facet and propagate from there throughout the pillar, and we also found that, at temperature, square pillars fail at lower strains and stresses than do circular pillars. Further research can include studying the effects of shape on pillars in tension, and finding the critical diameter below which surface effects begin to dominate.

We gratefully acknowledge the support of Barbara Bryan, HPTi, and the AHPCRC in facilitating this research. ✯

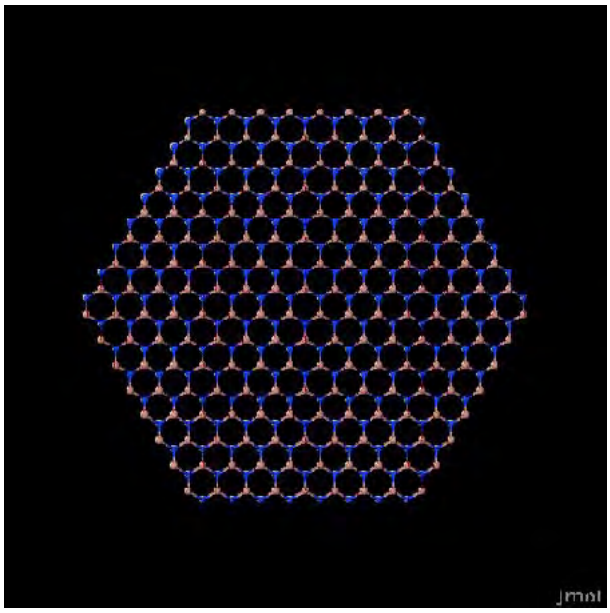
References

1. Lu et al. Revealing the Maximum Strength in Nanotwinned Copper. *Science 323* (5914), 607. (2009)
2. Wei et al. Nanoengineering Opens a New Era for Tungsten as Well. *JOM* 40–44 (September 2006)
3. Uchic et al. Sample Dimensions Influence Strength and Crystal Plasticity. *Science 305* (5686), 986. (2004)
4. Marian et al. Breakdown of Self-Similar Hardening Behavior in Au Nanopillar Microplasticity. *IJMCE 5* (3&4) 287–294. (2007)
5. S. J. Plimpton. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comp. Phys. 117,* 1–19 (1995) (http://lammps.sandia.gov)
6. Kelchner et al. Dislocation nucleation and defect structure during surface indentation. *Physical Review B, 58* (17), 11 085–088. (1998)
7. Zepeda-Ruiz et al. Mechanical response of freestanding Au nanopillars under compression. *App. Phys. Lett. 91,* 101907 (2007)
8 Rabkin et al. Onset of Plasticity in Gold Nanopillar Compression. *Nano Letters 7* (1) 100–107. (2007)

Thermal Conductivity in Hexagonal GaN Nanowires

Abraham Chukwuka (Morgan State University)
Mentor: Sylvie Aubry, Wei Cai

Wireless systems are widely used in Army applications like high power wireless. To build better MEMS/NEMS [micro- or nano-electromechanical systems] sensors and high election technology, there is a need to find a material with good and cheap optoelectronic properties. Researchers have recently focused their attention on the semiconductor materials used in power transistors by searching for a high performance building block that combines lower costs with improved performance. Of the contenders, gallium nitride (GaN) is emerging as the front runner. Gallium nitride laser diodes and HEMTs [high electron mobility transistors] operate at high temperature and power density leading to excessive heat generation and lower thermal conductivity. Low thermal conductivity causes nanowire failure. Also it was shown experimentally that as the nanowire diameter decreases, the thermal conductivity is reduced due to phonon scattering effects. It is therefore important to measure and compute accurately the thermal conductivity at the nanoscale.



Structural schematic cross-section of gallium nitride nanowire.

continued on page 16

Distribution Statement A: Approved for public release; distribution is unlimited.

GaN Nanowires

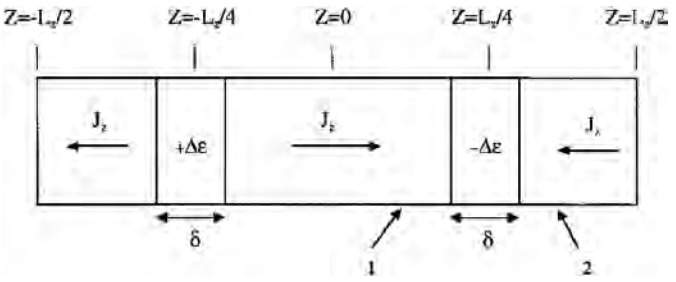
continued from page 15

Method to compute thermal conductivity

The thermal conductivity κ is given by Fourier's law:

$$J = -\kappa \frac{\partial T}{\partial x}$$

This law states that the heat current J and the gradient of temperature have a linear dependence *(illustrated below)*. The direct heat flux method for computing the thermal conductivity is analogous to experimental measurements. It was successfully applied to compute the thermal conductivity of bulk materials (Zhou, Aubry, Jones, Greenstein and Schelling, *Phys. Rev. B 79*, 115201, 2009).



Briefly, a hot and cold region are created in the simulation cell block by adding a small amount of kinetic energy Δε in the hot region and removing it in the cold one while preserving linear momentum at each MD time step using velocity rescaling. Periodic boundary conditions are maintained in the system and the Stillinger–Weber potential is used to model the GaN atoms. A heat flux J is then generated between the hot and cold regions. This heat flux is given by

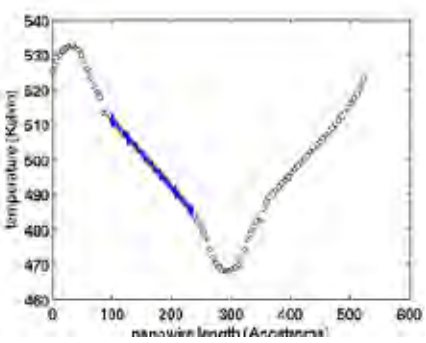
$$J = \frac{\Delta\epsilon}{2A\Delta t}$$

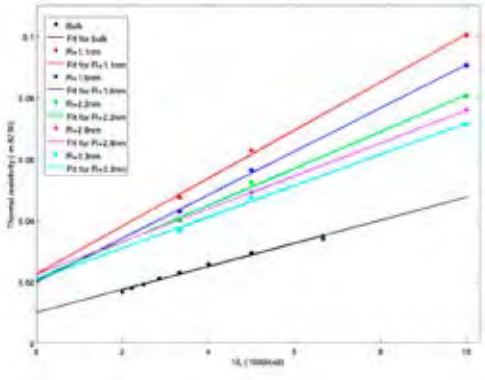
The system is equilibrated running MD for 3 ns using LAMMPS until a steady state current flow is achieved and the heat flux law above is valid. At steady state, a stationary temperature profile as a function of time is collected *(figure, top right)*. Large scale simulations are conducted to obtain a quasi-linear profile. Several million time steps are required to obtain the temperature profile shown. From this temperature profile, the temperature gradient is computed. It is the slope of the temperature plot versus length away from the hot and cold regions. These regions are visible in the figure.

Results

The direct heat flux method is applied to compute the thermal conductivity of hexagonal nanowires *(figure, bottom right)* of different lengths and radii. The data obtained is post-processed to obtain the average gradient of temperature and its standard deviation, also shown by a blue line in the top right figure. Using the temperature profiles and Fourier's law, the thermal conductivity for each given length and radius is obtained. The bottom right figure shows a plot of the thermal resistivity versus the inverse of length for each radius. A linear approximation is used to extrapolate the data and obtain the thermal conductivity of different nanowires. Values for thermal conductivity are shown below for different nanowire radii:

Top: Temperature profile obtained using molecular dyanamics
Bottom: Thermal resistivity vs. inverse length for wires of varying radii.





Radius (Å)	Bulk	11	16	22
Thermal cond. (w /m K)	102	45	50	48

Conclusion

Molecular dynamics simulations using large parallel computer simulations have been carried out to accurately compute the thermal conductivity of GaN nanowires. Good temperature profiles should be computed to get accurate values for extrapolating thermal conductivity. Scaling laws are being devised to deduce the thermal conductivity for larger nanowires using Molecular Dynamic data. ✯

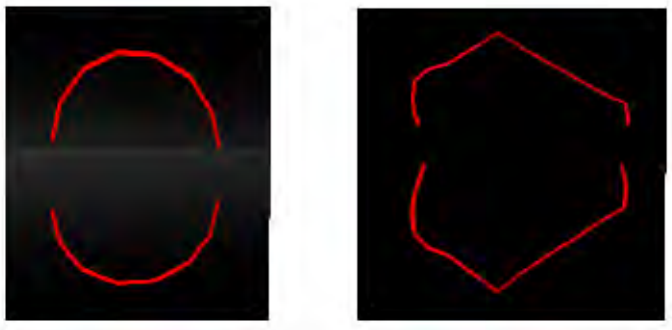
Highly Anisotropic iron in Nuclear Fusion Power Plants

Stacey Oriaifo (Morgan State University)
Mentors: Sylvie Aubry and Wei Cai

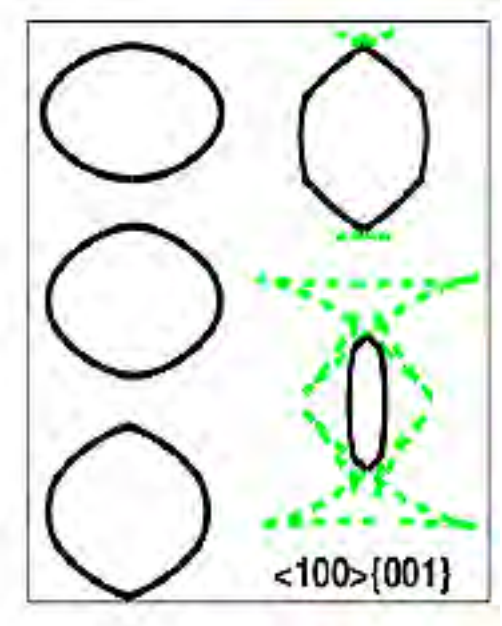
In nuclear fusion power plants, the first wall blanket structure is critical for safety and environmental considerations. The performance of the first wall blanket is dependent on structural material properties. Current designs of these power plants involve steels as the first wall structural materials and assume operating temperatures below 550°C. However, a considerable increase in efficiency can be gained by raising this temperature using ferritic–martensitic materials. A good candidate is iron (Fe) that can operate at 950°C. This brought about the need to study iron at elevated temperatures and model the dislocation dynamics in the metal in both isotropic and anisotropic elastic media so its mechanical properties can be fully understood.

Theory

Elastic deformation occurs when a body is deformed in response to a stress, but returns to its original shape when stress is removed. Usually, isotropic elasticity is used to model the system response because it is much simpler than anisotropic elasticity. According to research, iron at elevated temperatures is known to be one of the most anisotropic elastic materials. A measure of anisotropy is the H factor: $H = 2C_{44} + C_{12} - C_{11}$, which was calculated to be 1.714×10^{11} Pa in iron at high temperature. To model correctly iron, anisotropic effects cannot be overlooked.



Simulations of Frank–Read source for Case 1. Isotropic (left) and anisotropic (right) elasticity. Results show similarity of the dislocation shapes at critical stress to the line tension model (LT) model.



Theoretical calculation using line tension model for different initial segment sizes. (Graphic: Fitzgerald et al., GG6.3, MRS Fall Meeting, 2009)

All real crystals contain imperfections which may be point, line, surface, or volume defects. Dislocations are line defects within a crystal structure caused by atomic planes sliding over each other. A dislocation is entirely defined by the Burgers vector *b*, the glide plane *N*, and the line direction ξ. Dislocations could be edge, screw, or mixed.

Simulations

To model the dislocation interactions in iron, ParaDiS was used. ParaDiS is a massively parallel high performance computing (HPC) simulation code developed by Lawrence Livermore National Laboratory to model the dynamics of dislocations in FCC and BCC metals. Its scalability on 132,000 BG/L processors has been demonstrated. It is used to predict the macroscopic strength of a material from the collective behavior of its dislocations. Recently, anisotropic elasticity was added to ParaDiS. The goal of this work is to test this new capability on Frank–Read sources. A Frank–Read (FR) source is responsible for dislocation multiplication in metals under shear stress.

An FR source starts out as a segment of a dislocation line pinned between two points. The segment will

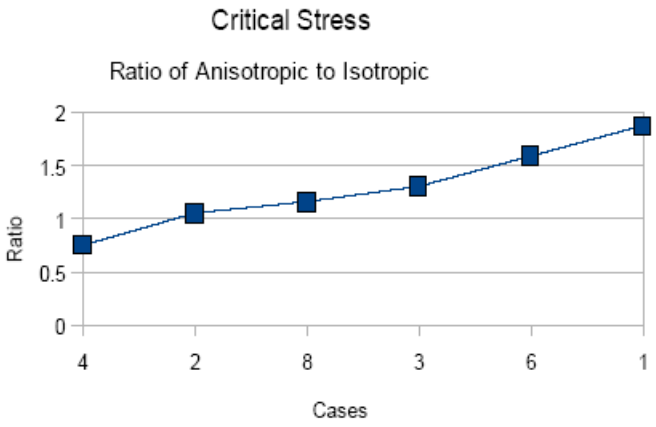
continued on page 18

Anisotropic Iron

continued from page 17

Case Number	b	N	Initial t (ξ)	Character
1	(1,0,0)	(0,1,1)	(0,1,–1)/√2	Edge
2	(1,0,0)	(0,1,1)	(1,0,0)	Screw
3	(1,0,0)	(0,0,1)	(0,–1,0)	Edge
4	(1,0,0)	(0,0,1)	(1,0,0)	Screw
5	(0.5,0.5.,0.5)	(0,1,–1)	(–2,1,1)/√6	Edge
6	(0.5,0.5.,0.5)	(0,1,–1)	(1,1,1)/√3	Screw
7	(0.5,0.5.,0.5)	(1,1,–2)	(–1,1,0)/√2	Edge
8	(0.5,0.5.,0.5)	(1,1,–2)	(1,1,1)/√3	Screw

Frank–Read cases considered in this project.



Critical stress: ratio of anisotropic to isotropic elasticity for the cases given in the table at left.

bow out under an appropriate applied stress. Once the critical stress is reached, the bowed-out segment becomes unstable, and loops back on itself. At this point, sections then annihilate and pinch off, resulting in a dislocation loop.

To calculate the critical stress of the FR cases *(table above)*, two methods were implemented: namely, the fixed stress method and the fixed strain rate method. In the fixed stress method, simulations with fixed stress were run using ParaDiS and the critical stress for bowing out the segments were determined by adjusting the stress manually. In the fixed strain rate method, a strain rate was specified and simulations were also run using ParaDiS. As the time is increased, stress increases until the segment bows out and the critical stress is determined. It was found that both methods are equivalent; however, the second method is faster.

Results

Comparison between isotropic and anisotropic elasticity was made for the eight different Frank–Read source cases shown in the table. For each case, the critical stress was also computed *(graph, above)*. Also, comparison of the speed of the two different methods shows that isotropic elasticity is 2.6 times faster than anisotropic elasticity using tables, while it is 33.5 times faster without using tables. Therefore, anisotropic elasticity runs 12.9 times faster with tables. Tables were devised to make the code run faster in ParaDiS.

The figure at the bottom of the previous page shows results for Case 1, the final shape of the Frank–Read source for both isotropic and anisotropic elasticity as it bows out, as well as the shape obtained using a theoretical calculation using a simplified model: the line tension model.

Conclusion

There is a large discrepancy between the anisotropic and isotropic approximations. It is seen that the isotropic theory alone is not an adequate approximation for certain materials such as iron. This research brings us closer to understanding the mechanical properties of iron, through modeling the dislocation dynamics, for use in nuclear fusion power plants. ✯

Automatic Calibration of Camera-Enabled Wireless Sensor Networks

Daniel Shaffer (Stanford University)
Mentors: Branislav Kusy, Martin Wicke, Leonidas Guibas

Abstract

Wireless sensor networks present an exciting new area for research. One interesting development is the inclusion of cameras as part of such a network. This technology has known applications ranging from surveillance to localization, as well as many other unexplored fields. One key software challenge to making use of such networks is that the camera calibration parameters (consisting of three rotational and three translational degrees of freedom per camera) must be known in some global coordinate system. Currently these parameters must be obtained manually, which is a tedious and time-intensive process. We are developing a robust system for the automated calibration of camera-enabled wireless sensor networks. Automation is achieved through the use of a cooperative mobile robot.

Experimental set-up: four cameras in a well-lit conference room. The entire system is controlled by a laptop computer on the table in the background.

Introduction

Megapixel-resolution cameras are dropping in price to the point where they can be deployed almost at will. Camera networks are now constrained not by cost but by our ability to process the data. Increasingly, the algorithms used to process video streams leverage not only the raw values of pixels but also the locations and orientations of cameras in a global coordinate system. Calibrated camera networks can be used to track objects or people as they move through the monitored space. Ongoing AHPCRC research is looking for other ways to utilize data from such networks.

This iRobot Create moved the light bulb from site to site, using IR and a bump sensor to stay within the walls. The 'mast' modular component can be swapped out for masts of different heights to prevent a degenerate case of the calibration problem in which all of the data points are coplanar.

Existing solutions for calibrating camera networks are designed for a laboratory environment. Many require that the entire space be made dark so that a single point light source can be accurately located in each camera's field of view. This is impractical for real-world applications: secure or outdoor spaces cannot be made dark for long periods of time. These solutions also require that a human "fill the space" with the point light source, which is impractical for building- or city-scale networks. Finally, these systems are run as an offline utility, offering no guidance about whether the system has enough data to perform a successful calibration within the specified tolerances.

Network Hardware

Our wireless sensor network is based on the Crossbow Telos platform. These motes offer extremely low power consumption, integrated USB-to-Serial ports for

continued on page 20

Camera Calibration
continued from page 19

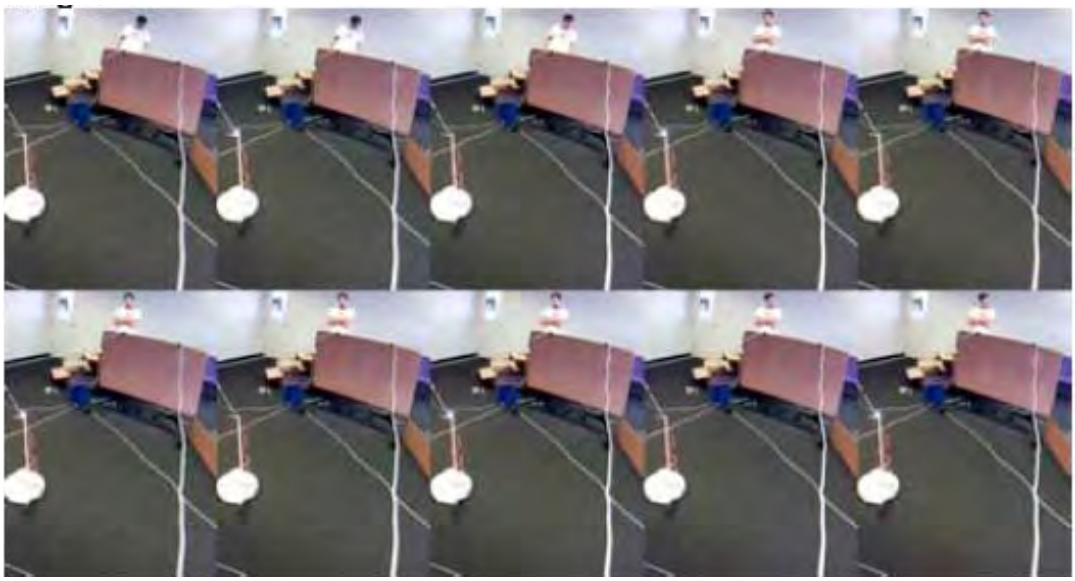
programming, small size for ubiquitous deployments and integrated 802.15.4 radios. These motes are designed to be used with a variety of sensors; our deployment uses Citric Camera motes soldered directly to the Telos motes. Each Citric mote includes a 600 MHz Intel processor running Linux and a 1.3 megapixel camera. The Citric motes rely on the Telos motes to handle radio communication.

Image Processing
One of the key difficulties in using existing systems is that they only work well in dark environments. We solved this problem by using a light bulb under the control of a Telos mote instead of a laser point light source. Instead of extracting the only bright spot from a darkened image (as in existing techniques), we combine a set of frames into a "site" in which the point light source is immobile. From a global control standpoint, the status of the light bulb at any given frame is the next value in a globally known or predefined series of bits. For our initial experiment a "010101..." bitstream was used; however, such a simple pattern may be susceptible to noise in the system (a random sequence might be more robust against noise or other disturbances). In theory, any bitstream of any length can be used. OpenCV (a computer vision library) is used to calculate the pixel which behaves most like the expected pattern for a given site and camera. This data is then passed on to the calibration algorithm.

Mobile Robotics
A second issue with existing systems is that they require a human operator to traverse most of the space. When combined with our system to overcome the lighting problem, the human operator would have to then stand perfectly still for a number of seconds before moving on to a new site. Robots are well suited for this type of task requiring precision and patience.

We chose to use an iRobot Create (similar to a Roomba robotic vacuum cleaner) for this task. This choice was made primarily because the Create is designed to "fill the space" in the process of its vacuuming cycle. Its bump sensor and virtual walls make it a suitable tool for moving our light bulb between the various sites while staying within the constraints of the experiment and not getting stuck against obstacles. The Roomba is controlled over UART [universal asynchronous receiver/transmitter] by a Telos mote.

Results
We calibrated a network of four cameras in a well-lit conference room. The test consisted of 51 sites of 11 frames each, with a simple on-off bulb pattern at 2 Hz. The results were run through a post-processing point detection routine to extract the point which behaved most like the bulb. These 204 image sets were then

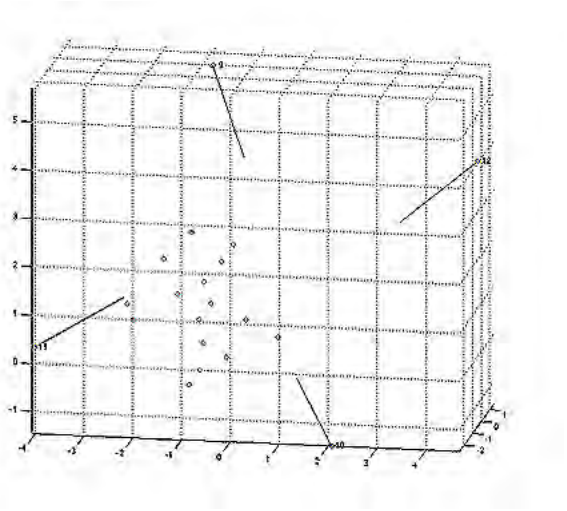


The data from one site from our test set. The Computer Vision portion of the project required that I isolate the pixel or pixels that were bright in odd frames and dark in even frames.

manually classified as to whether they contained a clear image of the bulb. A threshold was selected to eliminate false positives; points which did not behave sufficiently like the bulb were discarded.

These pixels were then fed into Svoboda et al.'s Matlab toolbox, along with intrinsics from a second toolbox (calculated from images of a checkerboard pattern).

Svoboda's Matlab toolbox returned the correct positions of all four cameras, with a relative error of approximately 10% *(figure at right)*. Most of this error comes from the low number of data points that were actually used by the toolbox (18 of the 51 possible). A higher level of accuracy can be achieved with more data, and thanks to the level of automation achieved in this system, collecting an order of magnitude more data is simply a matter of letting the calibration run during a lunch break or afternoon.



Output from Svoboda et al.'s Matlab toolbox using our data shows four cameras facing in toward a point cloud. From only 18 points, we were able to locate the cameras to within one ft. of our hand-measured ground truth. (Reconstructed points / camera setup only inliers are used)

Next Steps

Our immediate next step is to move from an offline to an online calibration package. Since we are not immediately aware of any, it is likely that we will be writing this from scratch, however the methods to do so are well documented in the literature.

Once we have a built an online calibration toolbox, we can use this to make the calibration process more robust by, for example, sending the robot into a region which lacked sufficient data. It can be used to speed up the process: we would have a live estimate of each position with confidence intervals, so we can stop once sufficient confidence was achieved. Finally, we can use the robot and cameras to build a map of the space and locate the robot on the map while calibrating the network. This would further speed up the calibration process while creating a useful by-product. ✯

Higher Order Acoustic Scattering of Submerged Objects

Kalesanmi Kalesanwo (Morgan State University)
Mentors: Paolo Massimi, Charbel Farhat
With help from Radek Tezaur and Charbel Bou-Mosleh

Higher order acoustic scattering involves the reflection, refraction and diffraction of high frequency incident acoustic waves in different directions from an object in a medium. Factors affecting Acoustic Scattering include the type of object as well as the medium in which the object is immersed.

Acoustic Scattering has a lot of applications both military and civilian. In terms of military usage, detection of underwater vessels such as submarines and missiles occurs as a result of acoustic scattering. In fact, all forms of sonar communications are a direct result of acoustic scattering. Also, acoustic scattering can be applied in the civilian world. For example, it can be used to detect mineral resources such as crude oil underwater. Scattering is also used for sea floor mapping. For example, communications companies laying underwater fiber optic data cables between continents need to know the sea terrain. Only through acoustic scattering can they come about this information.

continued on page 22

Acoustic Scattering
continued from page 21

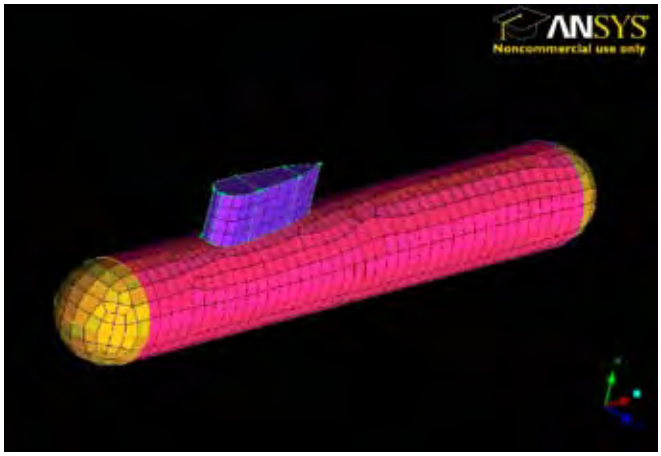
When acoustic waves are sent into a medium, the resulting echo provides us with information about any object that may be in the medium. To increase the amount of information gotten, the resolution must be increased. This effect can only be put in place by sending high frequency acoustic waves.

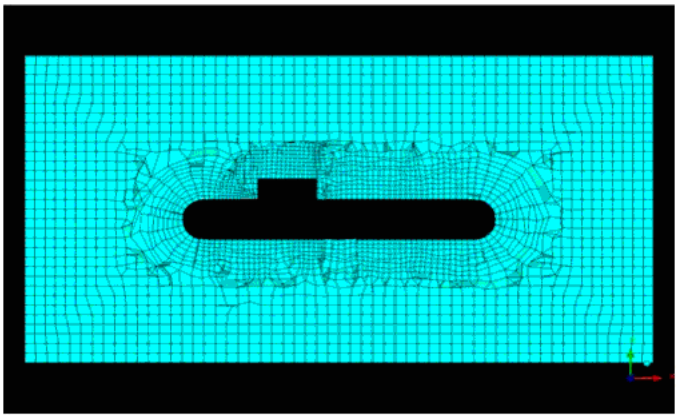
The evaluation of scattered waves from three-dimensional objects is a very important problem in computational science. This may be as a result of the large number of changing parameters that are particular to different instances of acoustic scattering. Such factors include the geometric and acoustic properties of the scatterer as well as the mathematical complexities of the oscillations of the incident waves.

For acoustic problems at mid and high frequency regimes, it is a well-known fact that the standard Finite Element method becomes unfeasible. Hence, there is a need for newer ways by which such problems can be solved with a relatively higher degree of accuracy.

The Discontinuous Element Method (DEM)
The discontinuous element method is a method proposed by Charbel Farhat, Isaac Harari, and Leopoldo Franca for solving the above problem. They describe the method best, with this abstract *(1)*:

"We propose a finite element based discretization method in which the standard polynomial field within each element by a non-conforming field that is added to it. The enrichment contains free-space solutions of the homogeneous differential equation that are not represented by the underlying polynomial field. Continuity of the enrichment across element interfaces is enforced weakly by Lagrange multipliers. In this manner we expect to attain high coarse-mesh accuracy without significant degradation of conditioning, assuring good performance of the computation at any mesh resolution..."

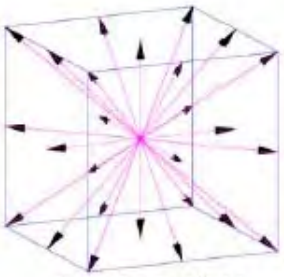


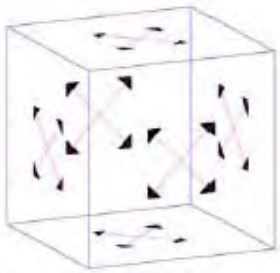


Top: **Submarine model with all-quad surface mesh.**
Bottom: **Cross section of volume mesh with hexa, prism, tetra, and pyramidal elements.**

With acoustic scattering, the enrichment solutions are represented with a summation of plane waves in the direction specified by theta. Also, the Lagrange multipliers are also represented with wave equations as can be seen on the screen *(illustrations, below left)*. It is also pertinent to note that for increased stability, the number of Lagrange multipliers must be far less than the degrees of freedom.

The DEM as been successfully implemented with a variety of phenomena such as the Acoustic/Helmholtz problem for 2D and 3D rigid obstacles, and the wave propagation problem for multi-fluid, multi-solid and fluid-solid media. It has also been proven to produce identical results to the standard Finite Element Method in a far shorter period of time thus reducing computational cost.


Enrichment field


Lagrange Multipliers

Scattering Problem and Mesh Generation

For the purpose of this project, the scatter problem is mathematically represented as: $-\Delta p - k^2 p = 0$

Boundary condition on the scatterer: $\frac{\partial p}{\partial n} = -\frac{\partial}{\partial n}\left(e^{ikdx}\right)$

Boundary condition on the exterior domain: $\frac{\partial p}{\partial n} - ikp = 0$

The first boundary determines the properties of the waves acting on the scatter surface while the second boundary determines the properties of a reflected traveling wave moving towards infinity.

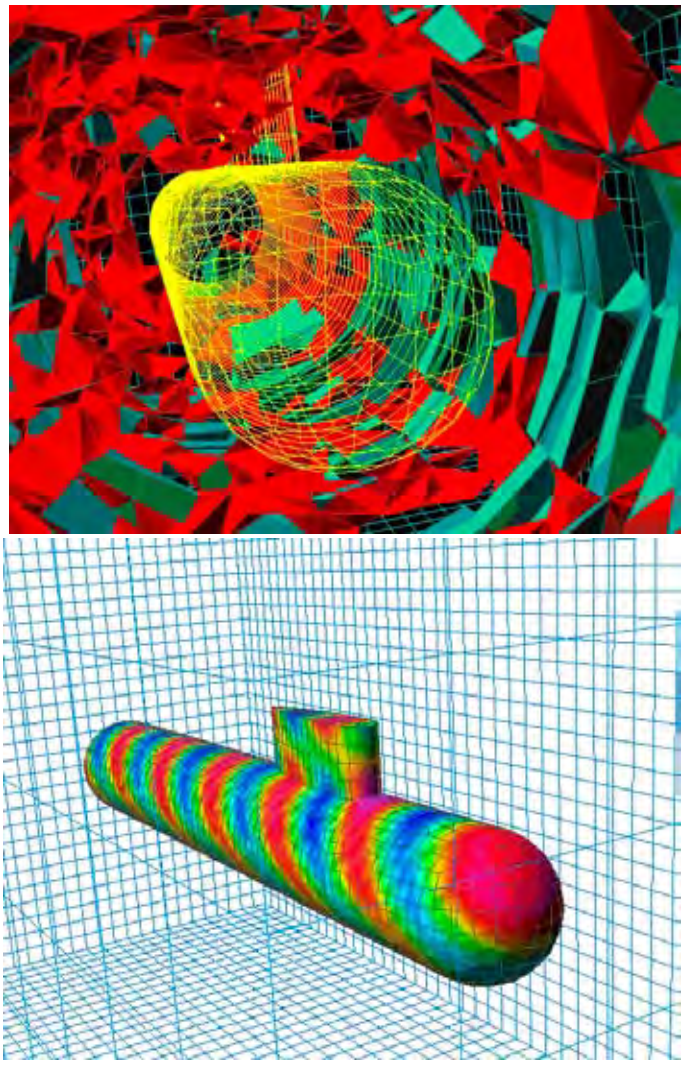
To implement the DEM technique over a realistic case, we needed a mesh. Normally all hexa meshes are created for the DEM because they are known to work well together. However, Radek Tezaur of the Farhat Group at Stanford University has come up with a new DEM implementation code that makes use of multi-polygonal meshes. As such I was able to run the code using a multi-polygonal mesh that I built around a model. The model was a submarine. I used ANSYS as the commercial mesh generator. The surface mesh I created on my model was an all-quad mesh because I wanted my mesh to be hexadominant. My volume mesh was made up of hexa, prism, tetra, and pyramidal elements *(figures, page 22)*.

We find that the multi-polygonal mesh is a better means for solving acoustic problems. This is because with multi-polygonal meshes, there is a reduction in the number of elements needed for our mesh, hence reducing analysis and post processing times especially when dealing with complex acoustic problems.

Results

We then ran simulations on the submarine and this picture gives a visualization of our simulation. The colors on the surface of the submarine *(bottom right figure)* show the acoustic pressure on the submarine from an acoustic wave hitting the submarine from the *x*-axis with an incident angle of 45° and a frequency of 1000 Hz.

We got about 2,009,228 enrichment degrees of freedom and only about 874,044 Lagrange multipliers. The number of elements we had was just over 77,278 for our simulations.



Top: **Post-processed mesh showing tetra and prism waves.**
Bottom: **Submarine with acoustic pressure from a 45° incident acoustic wave.**

Conclusion

We were able to validate the new code with different elements for the DEM using a realistic case study that is the submarine. At mid-frequency our results were feasible and there was no quadrature rule required for analytically computing the matrices. As such, our computations were very fast. Besides, this DEM code can be implemented with parallel computation. ✯

Reference

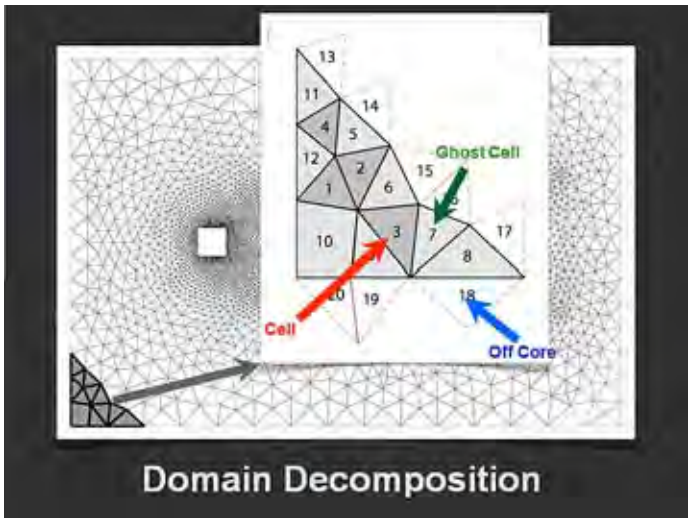
(1) C Farhat, I. Harari, L.P. Franca, The discontinuous enrichment method, *Comput. Methods Appl. Mech. Engr. 190,* 6455–6479 (2001).

Distribution Statement A: Approved for public release; distribution is unlimited.

Liszt Mesh Visualization Tool

Edgar Padilla, Essau Ramirez (University of Texas at El Paso); Richard Gutierrez (New Mexico State University)
Mentors: Zach Devito, Pat Hanrahan, Eric Darve

For the Army High Performance Computer Research Center (AHPCRC) Summer Institute, we created a mesh visualization tool. This tool will prove to be useful in visualizing meshes as well as debugging code used in the Liszt project at Stanford University. To understand the importance of a mesh visualization tool and the benefits that our tool provides, some background discussion is needed. This discussion centers on meshes and the tools currently available for using meshes in producing simulations of complex phenomena.



Domain decomposition for distributing computation on a mesh, using the Liszt language. (Figure courtesy of Z. Devito, Stanford University, Ref. 2)

Meshes are a collection of vertices, faces and edges that approximate a geometric domain. There are different representations of polygon meshes, which are used for different application and goals. Some of these applications include computer graphics, rigid-body dynamics and collision detection. Meshes prove to be extremely useful in scientific applications, as well. Meshes are used in numerical techniques to find approximate solutions to partial differential equations (PDEs). Numerical techniques such as the finite element method, finite difference method and finite volume method all employ the use of meshes. These methods are then used to produce accurate simulations of complex phenomena in various areas. These areas include Aerodynamics, Flow and Structure interactions or the drag created around a race car. It is in the tools for simulations such as these, that a mesh visualization tool becomes very useful.

At Stanford University, as part of the Predictive Science Academic Alliance Program (PSAAP), the main core of work is centered on the simulation of high speed flow around hypersonic vehicles and through its supersonic combustion propulsion system (scramjet). In addition to the simulation of high speed flow, the core of work also focuses on the prediction of unsteady thermal loads on the vehicle structure and fuel. One tool used in these simulations is the application "Joe." Joe is a state-of-the-art unstructured Reynolds-average Navier Stokes (RANS) solver. While implemented in C++, Joe has its heritage in Fortran. It is the main tool for system-level simulation and is highly optimized for a Message Passing Interface (MPI) cluster. A cluster is a group of linked computers that uses a specification for an Application Programming Interface (API) to communicate with one another. Although Joe is currently the main tool for system level simulation, it does have its limitations. Joe is highly optimized for running on a cluster that uses MPI, but cannot run on other architectures. In addition, minor changes in Joe's application level code can require major architectural changes in framework. Finally, there is a large learning curve associated with Joe for incoming students who are not "parallel-aware." The program Liszt, which is still under development at Stanford University, aims to eliminate these limitations.

Liszt is a domain specific language, a language with features that support a limited set of programs very well, for mesh-based PDEs. One goal of Liszt is to provide a language that allows one program to run on many types of machines such as a cluster, multi-core symmetric multiprocessor (SMP) or a multi-core graphics processing unit (GPU). In addition, Liszt aims to make the learning curve smaller, thus making the programming of PDEs easier. Liszt features a built-in mesh interface (vertices, edges, faces, cells), collections of mesh elements, mesh-based data storage (fields and

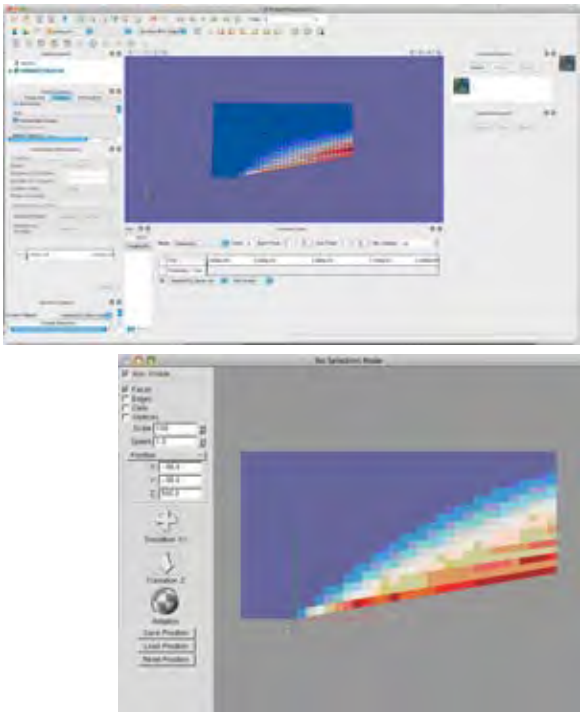
sparse matrices) and allows for parallelizable iteration. The computation in Liszt is centered on the mesh topology and mostly local. In addition, the computation is iterative and mostly regular. These features allow for domain specific optimization that cannot be done to the Joe code directly.

The runtime responsibilities of Liszt include decomposing the domain *(figure, page 24)* to distribute computation on the mesh (partitioning the mesh), establish communication between distributed memories to make the computation correct (such as in MPI), provide an implementation of the mesh interface, and layout and access the field/sparse matrix data.

The ghost cells in the figure correspond to cells that are being used by several partitions that share information with each other. It is within these tools (Joe and Liszt) that the importance of a mesh visualization tool and the benefits of our tool become apparent.

There are existing visualization programs such as Paraview that are open source and allow multi-platform data analysis and visualization *(figure, top right)*. In addition, Paraview provides the user with a desirable and highly customizable user interface. Also, Paraview is developed to analyze large data sets and provides a good final visualization. However, these programs are often difficult to use. For example, with Paraview the user must write separate code (which can often range in hundreds of lines of code) to output the faces of a mesh in a format that Paraview understands. Once this is done, Paraview works nicely, but only for the faces of the mesh. If a user also wanted to visualize the ghost cells, the partitions of a mesh, vertices and edges, he would have to write separate code in Paraview for each of these. This process becomes tedious, especially if the visualization was only needed for debugging purposes. With this in mind, we set out create a mesh visualization tool that could provide a good final visualization (presenting results), but also be great for debugging purposes.

To accomplish this, we chose to use OpenGL (Open Graphics Language), which is a cross-platform API, to create a user interface that provides a lot of options for visualizing a mesh. These options include abilities to:



Top: **Mesh visualized in Paraview.**
Bottom: **Mesh visualized by the Liszt Visualization Tool.**

- Display vertices, faces and edges individually or simultaneously
- Select cells, vertices, edges and faces
- Display values, positions, ID's, ghost cells and partitions of the mesh
- Provide a color map to help indicate the intensities (magnitude) of the values
- Rotate and translate the view around a mesh

The ID's will aid in debugging because if a mistake is seen in the visualization the user can go straight to the error in the Liszt code.

Liszt provides information such as ID, positions, values, ghost cells and partitions, while our tool visualizes the mesh and gives the user access to all of the above options in a very simple and easy to use interface *(figure, bottom right)*. There is no extra code needed, like Paraview, thus, making it extremely useful for debugging purposes in Liszt. The reason that our visualization tool does not require any extra code is because it is tightly coupled with Liszt. In addition, our mesh visualization tool also provides a good final visualiza-

continued on page 26

Mesh Visualization
continued from page 25

tion that can be used to present results.
In conclusion, although existing visualization programs exist, a simple and easy to use mesh visualization tool, like the one we created, is useful in many applications within Liszt. It is extremely useful for debugging purposes and also as a final visualization of a mesh. Also, there is no worry about how it will work since the visualization will be built to run with code already created in Liszt. ✯

References

1. Predictive Science Academic Alliance Program (PSAAP) at Stanford University. PSAAP/Stanford University, Aug. 11, 2009. http://psaap.stanford.edu/
2. Devito, Zach et al. "Liszt: A DSL for Mesh-Based PDEs." Stanford University

Sparse Matrix Solvers for Multi-Core and Parallel Platforms

Emilio López, Andrés Morales (Stanford University)
Mentors: Cris Cecka, Eric Darve

Real-world engineering problems attempt to formulate, model, and simulate the world around us. As such, they need to take several factors (independent variables) into consideration, things such as position, velocity, acceleration, temperature, atmospheric conditions, gravity, etc. Thus, these engineering problems lead to Partial Differential Equations (PDEs). Engineering problems of interest to the army (in areas such as Mechanical Engineering, Civil Engineering, Fluid Dynamics, Materials Modeling, Heat Distribution) often fall into this category.

$$
\begin{aligned}
\mathbf{q}_k &= A\mathbf{p}_k, \\
\alpha_k &= \frac{\rho_k}{\mathbf{p}_k \cdot \mathbf{q}_k}, \\
\mathbf{u}_{k+1} &= \mathbf{u}_k + \alpha_k \mathbf{p}_k, \\
\mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{q}_k, \\
\rho_{k+1} &= \mathbf{r}_{k+1} \cdot \mathbf{r}_{k+1}, \\
\beta_k &= \frac{\rho_{k+1}}{\rho_k}, \\
\mathbf{p}_{k+1} &= \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k.
\end{aligned}
$$

A single iteration of the conjugate gradient method.

Most PDEs can be discretized to yield sparse linear systems, or linear algebraic equations involving matrices and vectors that are mainly composed of zero entries. Despite this sparseness, though, several of these systems are so massive that they become impossible to solve on a single CPU. The vastness of the data involved requires amounts of memory so large that the CPU may not be able to allocate the data it needs to perform computation on. Additionally, memory overhead makes the computation very expensive. Moreover, the amount of computation necessary to solve such large systems is such that a CPU, approaching the problem with a serial methodology, cannot do so in a time-efficient manner, if at all. Thus, one could either use better algorithms, or faster hardware. A parallel computing approach meets both of these goals.

A heterogeneous computing system with multiple CPUs and GPUs.

In our research, we sought to solve large sparse linear systems using different implementations: (1) a serial approach on a single CPU, to compare the improvement of parallel implementations to; (2) a single CPU performing parallel computations on a GPU; and (3) a heterogeneous computing system (multiple CPUs, multiple GPUs) *(figure above)*.

The method we used to solve the aforementioned sparse linear systems is the Conjugate Gradient method, an iterative method for solving symmetric positive definite matrices. The method makes use of additional data structures and, starting from an initial u, approximates the solution to $Au = b$ by minimizing the residual at every iteration. A single iteration is demonstrated at left, and it involves matrix-vector multiplies, dot products, scalar multiplication, and vector addition.

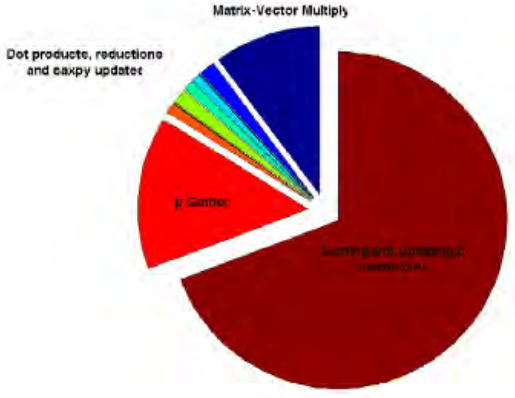
For the dot products and the saxpys ($u_k+1 = u_k + \alpha p_k$), we could easily use CBLAS, a linear algebra library

for use with C, and CUBLAS, the CUDA equivalent. However, for the matrix-vector multiply, a straightforward computation would have been inefficient, not taking advantage of the sparsity of the matrix. To make a more efficient one, we stored the matrix in Compressed Row Storage (CRS) format. Instead of a two-dimensional array representing the matrix, CRS stores it as three arrays, one with all the nonzero values, one with their corresponding column indices, and one with indices at which the values in a certain row start in the first two arrays (i.e., the third value in this last array is the index in the first and second arrays at which the first value from the third row is located).

For our parallel implementations of the Conjugate Gradient solver, we made a simple kernel, where every thread represented a row and computed a dot product of that row and the vector being multiplied. Additionally, due to CRS, this kernel only multiplied the non-zero entries of the matrix.

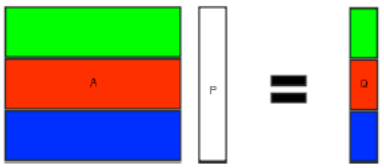
The heterogeneous computing implementation of our CG solver split up all the data, except for the *p* vector, and distributed among all the compute nodes. Each processor was given a set number of rows in the matrix and they only computed a partial matrix-vector product, as shown above. Similarly, each processor only had segments of the *p, q,* and *r* involved in the CG algorithm, so they only computed partial dot products and only updated parts of each of these vectors at each iteration.

It would logically seem that this would result in a faster computation of the solution, however, one has to take into account the time involved in the communication



Time analysis shows that the time needed to gather and update the *p* vector on the GPUs (brown section) limits the overall efficiency of the computation.

Each processor computes a partial matrix-vector product for a set number of rows.



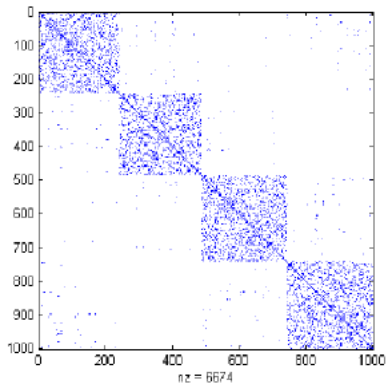
between different compute nodes through MPI. To obtain a dot product, for example, we need to retrieve all the partial dot products that each processor computed, add them together, and distribute them to every processor (done with an MPI_Allgather function).

The time that it takes to distribute the data to every node and put it up on the accelerator card unfortunately dwarfs the time that it takes to do the actual communication. This does not allow the total heterogeneous time to be lower than the single node GPU time. The pie graph time analysis *(below left)* shows that what is keeping the heterogeneous algorithm from being more efficient than the single node is the communication of the large *p* vector, which is needed by each process at every iteration.

In order to address this communication problem, our team proposed using a communication matrix method, in which each process only has to communicate a small part of their *p*. By partitioning the mesh before creating the matrix with METIS we are able to get blocks of density around the diagonal that correspond to each process's section of *p*. The communication that is then required is that of the indices in *p* corresponding to the sparse areas around the dense block *(below right)*. Ide-

continued on page 28

Modifying the communications method to localize sections of *p* greatly increases efficiency.



Sparse Matrix Solvers
continued from page 27

ally this will reduce the communication by 3 orders of magnitude. Moreover, due to the geometry of the mesh, the amount of boundary of a given partition will not increase dramatically with an increased problem size. The algorithm will thus be highly scalable and hopefully outperform CUDA at very large mesh sizes.

The communication matrix, in our first implementation, did not increase the efficiency of the algorithm. Time constraints did not allow us to come up with a general method of communication of the different parts of p such that all the nodes were communicating at all times. Thus, we hard coded it for 4. The fact that we were not able to vary the amount of processors, as well as the fact that we overlooked some of the memory allocation overhead, made the optimization initially slower. In future implementations, we will attempt to use shared memory space and allocate send vectors before the first iteration. Furthermore, we will implement a direct send buffer update in the dot product kernel that will decrease the memory accesses and allow for quicker device-host p transfers. We will also look in to general algorithms to get N processes sending and receiving from each other such that none are idle at any point in the communication algorithm.

In general, our project was geared towards developing computational software for the rapidly emerging paradigm of heterogeneous computing. Although at the moment we were not able to get the speed-ups we were looking to achieve, we are confident that, because the graphs show a decrease in compute time in the heterogeneous model, we will be able to get improvement over the single GPU for large matrices and thus have implemented a highly scalable program for solving sparse matrices. ✯

Full Cache Coherency on an FPGA-Based Accelerator

Kevin Thompson (New Mexico State University)
Mentors: Sungpack Hong and Kunle Olukotun

I worked with Kunle Olukotun's research group on the Flexible Architecture Research Machine, or FARM, project. FARM is an industrial strength cluster that uses state of the art CPUs, GPUs, memory, and input/output. Each node in the cluster currently has three quad-core AMD Opteron CPUs and one Altera Stratix II FPGA. This machine is scalable to 10s or 100s of nodes. The components of this machine are connected with Infiniband or PCIe interconnect using HyperTransport technology. As a research machine, the FARM cluster's main beauty is that its computing, memory, and input/output can all be personalized by programming the FPGA.

This research can be directly applied to the Army's interests. Using an FPGA as an accelerator has many benefits. First, it has lower power consumption than higher clocked GPUs. Second, it is programmable and flexible. This makes it customizable to accelerate a specific task by basically redesigning and optimizing the hardware itself. Finally, GPUs cannot take advantage of cache coherency since they have so many cores. Using FPGA acceleration, you can create a somewhat portable supercomputer that can compute complex scenarios and simulations in the field. Since it can be transported and powered in a vehicle, there is no need to send data back to headquarters and wait for a response. Immediate results lead to immediate action. Dr. [Raju] Namburu from the Army Research Laboratory shared information with us about using a GPU accelerated portable supercomputer to find IED's in the field. Before, the GPU acceleration data was collected from radar and sent back to headquarters to

be analyzed. This would take a few days. After GPU acceleration, data collected can be analyzed in real time so the vehicle with a supercomputer onboard can drive down a road and easily locate buried IED's!
My project goals this summer were to learn how to use and implement Verilog HDL and ModelSim simulation software, build hardware using a more efficient protocol than previously was implemented, and upgrade the current cache architecture on the FARM FPGA from a primitive state to a more advanced state. I spent the first half of the program learning Verilog and computer architecture and the second half of the program designing hardware using these new tools.

I will now address a little background about cache coherency in general. The problem here is when you have more than one CPU (or in our case, an FPGA) working together on the same computer. If both processors read some data, then they both think it is a certain value. Now if one processor writes a new value, the second processor still thinks it is still the original value. This would lead to terrible results and an inoperable machine. This is where cache coherency comes in. By developing these coherency protocols, scientists and engineers were able to allow the processors to "snoop," or spy, on each other's actions. So, when our first processor writes the new data, the second processor sees this and updates its copy. This keeps all copies up to date. There are five basic states to this protocol that I addressed in my research: modified, owned, exclusive, shared, and invalid, otherwise known as the MOESI cache coherence protocol. Every cache line in our cache is in one of these five states at any given time. The modified state holds valid information that has been changed since it was read from the main memory (DRAM); it also is not shared with any other processor. When a line is in the owned state, it is similar to the modified state, except that it may be shared with other processors. The shared state means that the data is valid, but has not been modified since it was read from DRAM. Exclusive is similar to this state, but it is not shared with any other processor. Finally, if a cache line is in the invalid state, it no longer holds a current value of the data and can safely be evicted and overwritten.

Prior to my work on this project, the coherency protocol only used modified and invalid states. This meant that as soon as a second processor wanted to read the data, it had to be marked as invalid on the first processor. If the first processor wanted to read it back, it would have to be copied and marked invalid on the second processor. By designing this hardware, I was able to implement a much more efficient and advanced protocol. This will help cut run time by up to 50%. In order to make this work, I needed to append three bits to each cache line: a valid, dirty, and shared bit. Depending on which bits are high, the cache line is marked as one of the five coherency states. For instance, 101 added to the cache line tells the controller that the cache line is valid, not dirty, and shared. This means it is presently in the shared state.

All in all, I created synthesizable hardware using Verilog HDL to write the cache and ModelSim to test it. The waveform produced by ModelSim confirmed that by the end of the summer, I had a functioning cache ready to be synthesized. I successfully designed hardware that is roughly 50% more efficient than the previous implementation. I upgraded the cache from a primitive state to an advanced protocol. This has been a great experience that will be invaluable to my future as a professional engineer. ✯

Publications and Presentations

Distribution Statement A: Approved for public release; distribution is unlimited.

AHPCRC Publications and Presentations March–May 2010

A complete list of publications and presentations is available at http://www.ahpcrc.org/publications.html

Project 1–1:Multifield Simulation of Accelerated Environmental Degradation of Fabric, Composite and Metallic Shields, and Structures

- High-speed impact with electromagnetically sensitive fabric and induced projectile spin. Zohdi, T. I. *Computational Mechanics,* published online 9 March 2010. doi: 10.1007/s00466-010-0481-5.

Project 1–2: Simulation of Ballistic Gel Penetration

- Some new algorithmic ideas for the simulation of ballistic gel penetration. M. Potts, A. Lew, R. Ryckman, R. Rangarajan. Presentation at First Annual ARL Ballistic Protection Technologies Workshop, Herndon, VA, May 2010.

Project 1–4: Flapping and Twisting Aeroelastic Wings for Propulsion

- Numerical Study of Flexible Flapping Wing Propulsion. T. Yang, M. Wei, H. Zhao. Paper 2010-0553, 48th AIAA Aerospace Sciences Meeting, Orlando, FL, January 2010.
- Computational Analysis of Hovering Hummingbird Flight. Z. Liang, H. Dong, M. Wei. Paper 2010-0555, 48th AIAA Aerospace Sciences Meeting, Orlando, FL, January 2010.
- Optimal Flight of Rufous Hummingbirds in Hover: An Experimental Investigation. H. B. Evans, J. J. Allen, B. J. Balakumar. Paper 2010-1028, 48th AIAA Aerospace Sciences Meeting, Orlando, FL, January 2010.

Project 1–7: Advanced Optimization Algorithms and Software

- MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems. S.-C. T. Choi, C. C. Paige, and M. A. Saunders. *SIAM J. Sci. Comp.* (submitted March 2010), 26 pp.
- Sparse least-squares problems. M. Saunders, D. Fong. Presentation at Eleventh Copper Mountain Conference on Iterative Methods, April 2010.

Project 2–2: Micro- and Nano-fluidic Devices for Sensing BWAs and Blood Additive Development

- The dynamics of vesicles in simple shear flow. H. Zhao, E.S.G. Shaqfeh. *J. Fluid Mechanics.* Submitted May 2010.

Project 2–4: Protein Structure Prediction for Virus Particles

- Constraint-based Protein Fragment Assembly. A. Dal Palu, A. Dovier, F. Fogolari, E. Pontelli. *Theory and Practice of Logic Programming.* To appear, 2010.
- Constraint Logic Programming in Determining and Composing Protein Fragments. A. Dal Palu, A. Dovier, F. Fogolari, E. Pontelli. National Conference on Computational Logic. Submitted, 2010.
- Answer Set Programming in 2010. E. Pontelli. *Symposium on Practical Aspects of Declarative Languages,* Springer Verlag, pp. 1–3, 2010.
- Applications of Constraint Programming Technology. E. Pontelli. Presentation at Symposium on Practical Aspects of Declarative Languages, Madrid, Spain, January 2010.
- Structure Prediction for the Helix Skeletons Detected from the Low Resolution Protein Density Map. K. Al Nasr, J. He. Submitted to Asia Pacific Bioinformatics Conference, India, January 2010. Also, *Proceedings of the 2009 BIBM Computational Structural Bioinformatics Workshop,* Washington DC, November 2009. *BMC Bioinformatics, 11,* Suppl. 1:s44, 2010.

Project 3–1: Information Dissemination and Aggregation Under Mobility

- Mobile Image Webs. Zixuan Wang. Presentation at POMI retreat, February 2010
- Fingerprinting Mobile User Positions in Sensor Networks. Mo Li, Xiaoye Jiang, Leonidas Guibas. 30th International Conference on Distributed Computer Systems (ICDSC) 2010. Accepted (to appear in June 2010).

Project 3–2: Scalable Design Methods for Topology Aware Networks

- Subgraph Sparsification and Nearly Optimal Ultrasparsifiers. A. Kolla, Y. Makarychev, A. Saberi, S. Teng. To appear in the *42nd ACM Symposium on Theory of Computing (STOC 2010).* Available at: http://www.stanford.edu/~saberi/sparsifier.pdf

Project 4–3: Specifying Computer Systems for Field-Deployable and On-Board Systems

- FAIRIO: An Algorithm for I/O Performance Differentiation. Sarala Arunagiri, Yipkei Kwok, Seetharami R. Seelam, Patricia J. Teller, and Ricardo Portillo. Submitted to *IEEE International Conference on Cluster Computing 2010,* Cluster 2010.

Project 4–6: Hybrid Schemes for Parameter Estimation Problems

- Miguel Hernandez IV, invited poster. DOD High-Performance Computing Modernization Program (HPCMP) User-Group Conference (UGC), Schaumburg, IL June 14–17, 2010.
- A comparison of wavelet-based schemes for parameter estimation. *IEEE-CS DOD HPCMP UGC Conference Proceedings.* To be published.
- Miguel Hernandez IV, Carlos Ramirez, and Reinaldo Sanchez, three posters. Sixth Annual Minority Serving Institution Research Partnership Consortium (MSIRPC) Conference, Baltimore, MD, April 15, 2010.
- Miguel Hernandez IV. Outstanding Scientific Presentation in Engineering & Physics. Awarded at the MSIRPC Conference, Baltimore, MD April 15, 2010.
- Ramirez, C., Sanchez, R. Two talks at the 6th Joint UTEP/NMSU Workshop on Mathematics, Computer Science and Computational Sciences, University of Texas at El Paso. El Paso, Texas, November 7th, 2009.
- A Path Following Method for Large-Scale l_1 underdetermined problems. Submitted revisions to the *IEEE Transactions on Signal Processing,* February 2010.

Project 4–7: Evaluating Heterogeneous High Performance Computing for Use in Field-Deployable Systems

- Processor Modeling Using Monte Carlo Techniques: An Alternative to Cycle-Accurate Simulation. Jeanine Cook. Talk given at Sandia National Laboratories, April 10, 2010.
- Modeling Superscalar, Out-of-Order Processors using Statistical Techniques: A Case Study of the Opteron Processor Model. Jeanine Cook. To be submitted to International Symposium on High-Performance Computer Architecture (HPCA), August 2010.

AHPCRC Consortium Members

Stanford University
High Performance Technologies, Inc.
Morgan State University
New Mexico State University at Las Cruces
University of Texas at El Paso
NASA Ames Research Center

Army High Performance Computing Research Center

c/o High Performance Technologies, Inc.
11955 Freedom Drive, Suite 1100
Reston, VA 20190-5673
ahpcrc@hpti.com
703-682-5368
http://www.ahpcrc.org

INSIDE THIS ISSUE

US ARMY RDECOM
TECHNOLOGY DRIVEN. WARFIGHTER FOCUSED.

This work was made possible through funding provided by the U.S. Army Research Laboratory (ARL) through High Performance Technologies, Inc. (HPTi) under contract No. W911NF-07-2-0027, and by computer and software support provided by the ARL DSRC.