# Mathematical Software

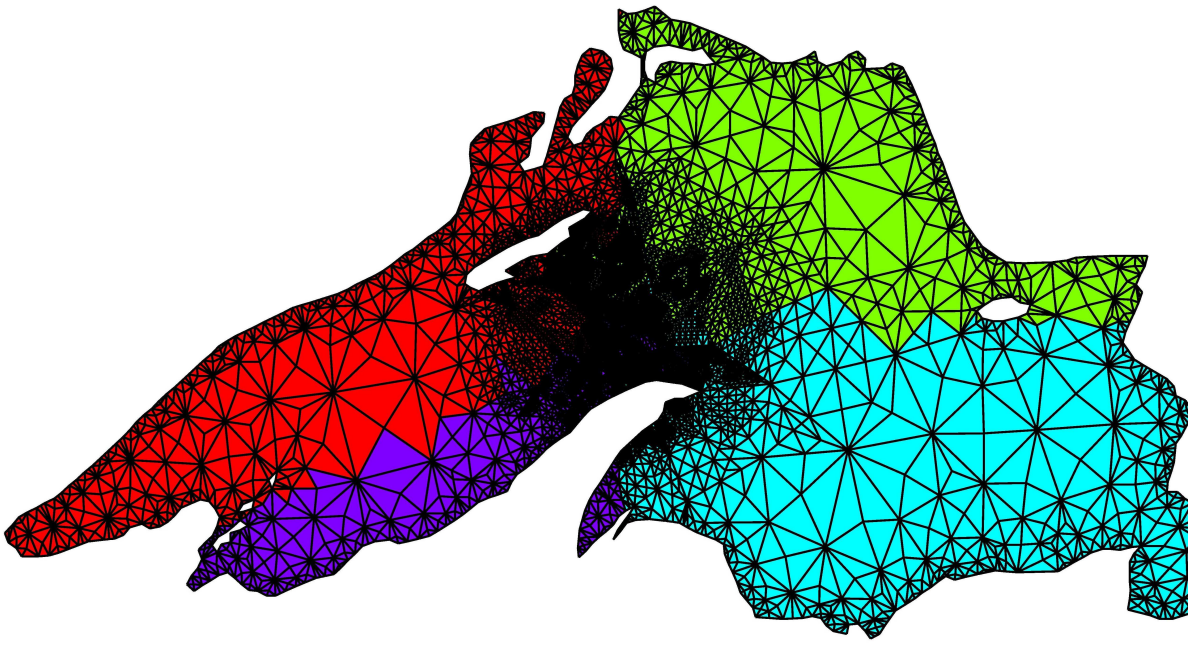
## Parallel Adaptive Refinement and Multigrid Finite Element Methods

*William F. Mitchell*
*Eite Tiesinga (NIST PL)*

http://math.nist.gov/phaml

Finite element methods using adaptive refinement and multigrid techniques have been shown to be very efficient for solving partial differential equations on sequential computers. Adaptive refinement reduces the number of grid points by concentrating the grid in the areas where the action is, and multigrid methods solve the resulting linear systems in an optimal number of operations. W. Mitchell has been developing a code, PHAML, to apply these methods on parallel computers. The expertise and software developed in this project are useful for many NIST laboratory programs, including material design, semiconductor device simulation, and the quantum physics of matter.

This year the effort was focused on two areas: improvement and extension of the PHAML software, and application of PHAML to solve Schrödinger's Equation for eigenvalues and eigenstates relevant to optical traps for neutral atoms, in collaboration with E. Tiesinga of the Quantum Processes group of NIST's Atomic Physics division. Understanding the interactions of adjacent atoms corresponding to qubits of a quantum gate involves computing multiple eigenvalues in the middle of the spectrum, with eigenstates that have sharp gradients, which is a very challenging computation. The major accomplishments for FY 2004 are the following:



An adaptive grid generated by PHAML on 4 processors for a solution with a sharp peak over a Lake Superior shaped domain. Color indicates the assignment to processors.

- Performed an experiment to study the performance of preconditioners applied to GMRES for the solution of the linear systems that arise when using ARPACK to solve the discretized Schroedinger equation. Results of this experiment were presented at Preconditioning '03.
- Determined that the most trivial trajectory through the space of pseudopotential functions is likely the most effective path to use to initialize the solution of the relevant Schrödinger equation.
- Made improvements to the PHAML code that sped up the solution of Schrödinger's equation by a factor of 50, allowing the solution of a more realistic model of Cesium atoms.
- Extended PHAML to solve coupled systems of equations (i.e. multcomponent solutions), and applied the new code to solve 2-channel Schrödinger equations.
- Extended PHAML to solve partial differential equations on arbitrary polygonal domains, with the help of the freely available grid generation package Triangle. This involved developing a new method to create an initial grid suitable for PHAML from an arbitrary triangulation.
- Extended PHAML to use high order finite elements to obtain more accurate solutions with fewer grid points.
- Extended PHAML to handle periodic boundary conditions.
- Several other minor improvements to PHAML were made, and eight minor releases of the code occured as the code evolved.

In addition, three talks were given at conferences on this topic, two of which were invited lectures, and two papers were accepted for publication in refereed journals.

Future work will continue to enhance PHAML with additional capabilities and robustness, extend PHAML to an hp-adaptive method with adaptivity in the polynomial degree as well as spacial resolution, study multigrid methods for high order finite elements, study error estimators for eigenvalue problems and high order finite elements, improve the robustness of the Schrödinger application code, and extend the application to a multi-channel model with time-dependent systems of equations.

## SciMark, a Web-based Benchmark for Numerical Computing in Java

*Roldan Pozo*
*Bruce Miller*

http://math.nist.gov/scimark

The NIST SciMark benchmark continues to be one of the most widely used Java scientific benchmarks. SciMark includes computational kernels for FFTs, SOR, Monte Carlo integration, sparse matrix multiply, and dense LU factorization, comprising a representative set of computational styles commonly found in numerical applications. SciMark can be run interactively from Web browsers, or can be downloaded and compiled for stand-alone Java platforms. Full source code is provided, in Java and C programming languages for comparison under different compilers and execution environments. The SciMark result is recorded as megaflop rates for the numerical kernels, as well as an aggregate score for the complete benchmark. The current results database lists over 2,400 submissions form computational platforms ranging from hand-held devices to high-end servers and supercomputers. The reports contains data representing nearly every operating system and Java Virtual Machine (JVM) environment currently in use, including Solaris, FreeBSD, MacOS, Sun OS, IRIX, OSF1, Linux, OS/2, and Windows 95, 98, 2K, ME, NT, XP platforms.

SciMark and its kernel components have become a pseudo-standard in industry and academia. They were adopted by the Java Grande Benchmark Forum, and Sun Microsystems used

SciMark 2.0 to demonstrate the floating-point improvements to their JVM 1.4.2[2]. As of December 2004, the record for SciMark is over 570 Mflops, with some of its kernels, such as LU factorization, running at over 1.2 Gflops. One major area of improvement for JVM implementations was the Monte Carlo kernel, which makes extensive use of synchronized methods. This helped identify synchronization overheads in common JVMs, and as a result the latest implementations show an improvement from single-digit Mflops to over 100 Mflops.

## Sparse BLAS Standardization

*Roldan Pozo*
*Iain Duff (Rutherford Appleton Labs)*
*Michael Heroux (Sandia National Laboratory)*

http://math.nist.gov/spblas
http://www.netlib.org/blas/blast-forum

MCSD has played a leading role in the standardization effort for the Basic Linear Algebra Subprograms (BLAS) and continues to be a major contributor for the design and development of reference software and documentation. The BLAS are kernels for computational linear algebra subroutines comprising of fundamental matrix/vector operations common to most scientific computing applications. By standardizing such interfaces, computer manufacturers and software vendors can provide high-performance implementations especially suited to a specific hardware platform. By developing their applications in terms of BLAS, computational scientists can achieve high levels of performance and portability. The original BLAS, which were developed for dense vector and matrix operations from the late 1970s through the early 1990s, achieved this goal very well. To this, the BLAS Technical Forum (an international consortium of industry, academia, and government institutions, including Intel, IBM, Sun, HP/Compaq/Digital, SGI/Cray, Lucent, Visual Numerics, and NAG) has formed an updated BLAS standard which addresses several new extensions.


Among the most significant components of the updated BLAS standard is support for sparse matrix computations. R. Pozo of MCSD served as chair of the Sparse BLAS subcommittee during the standardization process, and NIST was first to develop and release a public domain reference implementation in ANSI C for early versions of the standard, which helped shape the final specification. After the standard was formally approved and accepted, the complete technical specification was published, and a special issue of the ACM Transactions of the Mathematical Software (TOMS) was devoted to the new BLAS standard, including a paper co-authored by R. Pozo and other subcommittee members providing an overview of the sparse matrix interface.

This year saw the public release of an ANSI C reference implementation of the Sparse BLAS (http://math.nist.gov/spblas/SparseBLAS.html) and the development of a new compact C++ implementation that adheres to the reference interface. This new version is quite small, and uses advanced features of ANSI/ISO C++ and key ingredients of object oriented design to provide similar functionality in a small fraction of the lines of code, and a framework that is much easier to extend with user-defined sparse matrix formats. This new design was motivated by integration of the Sparse BLAS with the NIST Template Numerical Toolkit (see below). The TNT interface provides an object oriented framework to support fundamental sparse matrix operations and memory management in application codes, while the reference implementation provides stand-alone code that serves as a testing and validation module for platform-dependent versions of the Sparse BLAS library.

[2] See http://java.sun.com/j2se/1.4.2/1.4.2_whitepaper.html.

## TNT: Object Oriented Numerical Programming

*Roldan Pozo*

http://math.nist.gov/tnt

MCSD maintains an active research program in the design of object oriented mathematical software libraries. This work has led to some of the most highly used object oriented linear algebra packages, including Lapack++, Iterative Methods Library (IML++), Sparse Matrix Library (SparseLib++), Matrix/Vector Library (MV++), and the Template Numerical Toolkit (TNT). Together, these packages receive more than 60,000 downloads a year from the NIST web site, not counting numerous mirror sites which also host these codes. This software is currently in use in a variety of industrial and commercial applications.


TNT incorporates many of the ideas explored by R. Pozo and colleagues with previous software designs and advanced features of the ANSI/ISO C++ programming language. The package includes support for both C and Fortran multidimensional array layouts, arrays sections, and application modules, such as linear algebra, which includes fundamental algorithms for matrix decompositions (LU, Cholesky, SVD, QR), eigenvalue problems, and sparse matrix support such as vector scatter/gather operations, and matrix multiplication.

This year, the TNT project saw several public software releases and critical design updates. The latest TNT implementation has a separate interface specification which allows library developers to create specialized modules that take advantage of particular hardware platforms, utilize vendor-specific libraries, or implement different C++ strategies. An important new extension is the support for iterative methods used in the solution of large scale linear systems, including commonly used nonstationary methods such as Conjugate Gradient (CG), Generalized Minimal Residual (GMRES), Quasi-Minimal Residual (QMR), and related variants. The TNT web site provides a basic implementation for testing and development, as well as links to other library packages that utilize the TNT interface.

## OOF: Finite Element Analysis of Material Microstructures

*Stephen Langer*
*Andrew Reid (Drexel University)*
*Seung-Ill Haan (UMBC)*
*Edwin Garcia (Penn State University)*
*Eric Ma (Montgomery Blair High School)*
*Edwin Fuller (NIST MSEL)*
*Craig Carter (MIT)*

http://www.ctcms.nist.gov/oof

The OOF Project, a collaboration between MCSD, MSEL's Ceramics Division and the Center for Theoretical and Computational Materials Science, and MIT, is developing software tools for analyzing real material microstructure. The microstructure of a material is the (usually) complex ensemble of polycrystalline grains, second phases, cracks, pores, and other features occurring on length scales large compared to atomic sizes. The goal of OOF is to use data from a micrograph of a real material to compute the macroscopic behavior of the material via finite element analysis. OOF is intended to be a general tool, applicable to a wide variety of microstructures in a wide variety of physical situations.

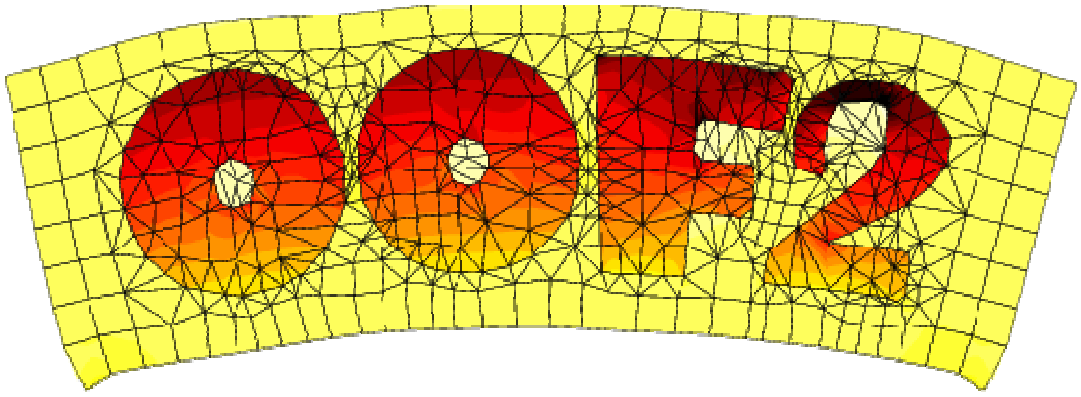
This year saw the first beta release of OOF2. OOF2 is a completely new version of the program, designed to be much more powerful and flexible than the original. The first release of OOF2 solved linear elasticity and thermal conductivity problems, like OOF1, but can easily be

extended to a wide variety of other physical problems, such as chemical diffusion and reactions, and piezoelectricity.

Most of the effort this fiscal year centered on the large number of features, bug fixes, and optimizations required to get OOF2 ready for release. These included improvements to the contour plotting code, better methods for defining graphical outputs, better tools for examining solutions, more compact data file formats. The program is now threaded, so it can perform multiple simultaneous tasks with the user interface remaining sensitive during CPU-intensive computations.

OOF2 can automatically create documentation of its command set. The OOF2 developers are working with the MatCASE (Materials Computation and Simulation Environment) project at Penn State University to create a suite of tools to simulate materials on all length scales, from microscopic quantum mechanical to macroscopic effective property calculations. OOF2 will perform the macroscopic calculations. Automatic generation of the API documentation facilitates the integration of OOF2 into the MatCASE environment. The documentation is written in XML so that it can also be converted into both on-line and printed manuals.

Eric Ma, a student in the Math and Science Magnet Program at Montgomery Blair High School, spent his second summer developing OOF code at NIST. He is working on techniques to detect grain boundaries in images of polycrystalline materials. These grain boundaries can be easy for humans to see but very difficult for a program to detect. If OOF2 can automatically detect grain boundaries, then it will be much easier for a user to assign material properties to individual grains. Eric has adapted and merged several edge detection algorithms from the image processing literature to form a promising new technique for finding boundaries in micrographs.



Results of a calculation performed with OOF2. The lower corners of the sample are fixed in place and a temperature gradient has been applied from bottom to top. Thermal expansion of the letters has led to elastic deformation.

## A Reference-Benchmark-Based Approach to Verification and Validation of Computer Models of High-Consequence Engineering Systems

*Jeffrey Fong*
*Barry Bernstein*
*Ronald Boisvert*
*Dan Lozier*
*Geoffrey McFadden.*
*James Filliben (ITL SED)*
*Hung-kung Liu (ITL SED)*
*Nell Sedransk.(ITL SED)*
*Terri McAllister (NIST BFRL)*
*Emil Simiu (NIST BFRL)*
*Howard Baum (NIST BFRL)*
*Kuldeep Prasad (NIST BFRL)*
*Ronald Rehm (NIST BFRL)*
*Roland deWit (NIST MSEL)*
*Richard Fields (NIST MSEL)*

Beginning in Feb. 2004, researchers from MCSD (Fong and McFadden), ITL Statistical Engineering Division (James Filliben), and BFRL (Howard Baum, Terri McAllister, Kuldeep Prasad, and Emil Simiu) have been working together on a five-year Competence Project entitled *Complex System Failure Analysis: A Computational Science Based Approach.* The objective of the project is to create the scientific basis for building failure investigation procedures that (1) will allow NIST to accomplish its mission under the National Construction Safety Team Act (P.L. 107-231, Oct. 1, 2002), and (2) is likely, over time, to be applicable for failure analysis and uncertainty determination of a broad range of complex physical, chemical, biological, and engineered systems of interest throughout NIST.

Since the physics of structural failure due to fire and other extreme loadings involves complex models and multi-physics simulations based on an incomplete knowledge of the governing equations and a strong variability in material properties, geometric dimensions, joint characteristics, and loading spectra, one of the first questions to address is to find a rational basis for the verification and validation of computational models of complex engineering systems. On Nov. 8-9, 2004, a workshop co-sponsored by NIST and DOD was held to address this important question. Two MCSD working documents containing preprints of 41 papers and 25 discussions were distributed to 50 attendees of the workshop. The attendees represented such organizations as Applied Research Associates (Littleton, CO), Auburn University, DOD Defense Modeling & Simulation Office, Lawrence Livermore National Laboratory, MIT, NSF, Oak Ridge National Laboratory, Sandia National Laboratory, Southwest Research Institute, SRI International Inc., Stanford University, Texas A&M University, University of Cincinnati, University of Houston, University of Iowa, and XYZ Scientific Applications (Livermore, CA). The workshop provided MCSD with valuable input regarding its potential role in the growing Verification and Validation community.

## Fortran 90 Version of ODRPACK Used in Package StatistiCAL

*Abbie O'Gallagher*
*D. Williams (NIST EEEL)*

Abbie O'Gallagher and D. Williams of EEEL have developed a special version of an unreleased, Fortran 90 version of ODRPACK, a package for orthogonal distance regression originally developed by MCSD staff in the 1990s. The work was in response to a problem that arose when Williams interfaced a DLL containing the Fortran 90 version of ODRPACK with StatistiCAL, a network analysis package that he has developed. A version suitable for a desired Visual Basic code is currently under development.