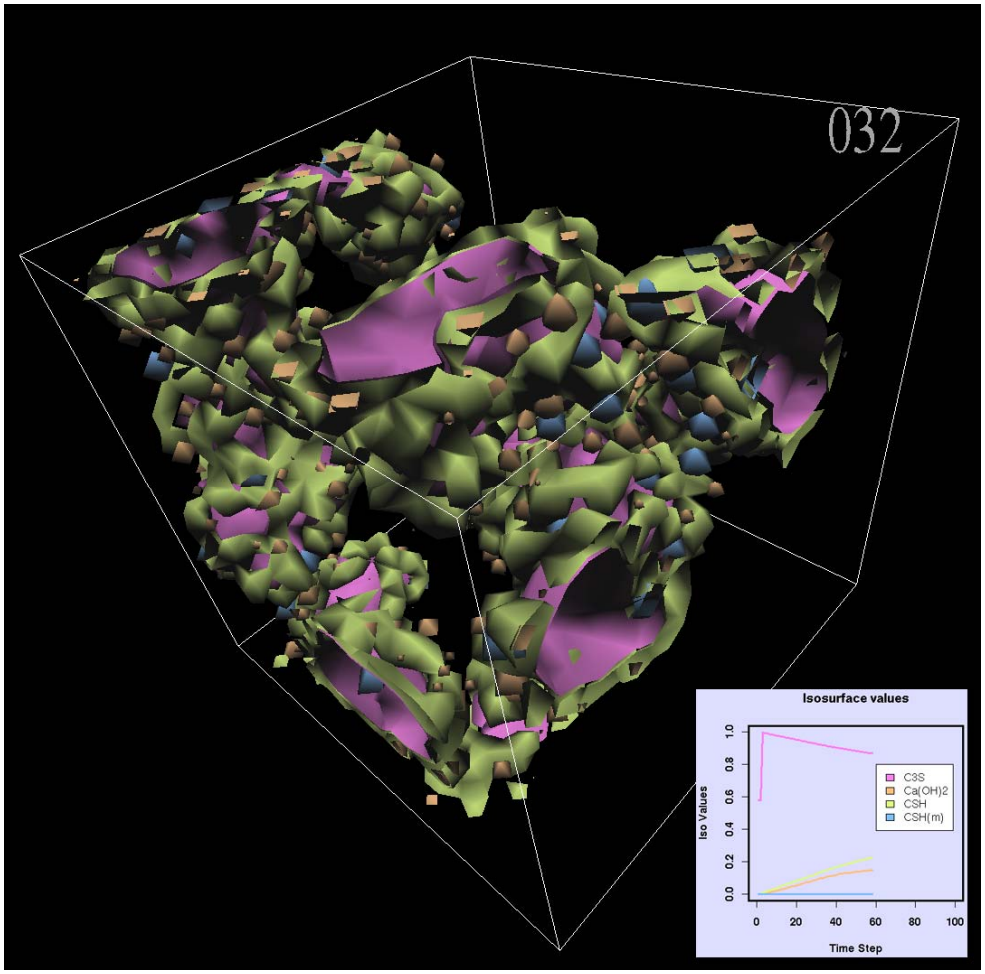
Mathematical and Computational Sciences Division

Summary of Activities for Fiscal Year 2007



Information Technology Laboratory
National Institute of Standards and Technology
U.S. Department of Commerce

January 2008

INFORMATION TECHNOLOGY LABORATORY

NIST National Institute of Standards and Technology U.S. Department of Commerce

Abstract

This report summarizes the technical work of the Mathematical and Computational Sciences Division (MCSD) of NIST's Information Technology Laboratory. Part I (Overview) provides a high-level overview of the Division's activities, including highlights of technical accomplishments during the previous year. Part II (Features) provides further details on eight particular projects of particular note this year. This is followed in Part III (Project Summaries) by brief summaries of all technical projects active during the past year. Part IV (Activity Data) provides listings of publications, technical talks, and other professional activities in which Division staff members have participated. The reporting period covered by this document is October 2006 through December 2007.

For further information, contact Ronald F. Boisvert, Mail Stop 8910, NIST, Gaithersburg, MD 20899-8910, phone 301-975-3812, email boisvert@nist.gov, or see the Division's web site at http://math.nist.gov/mcsd/.

Cover photo. Visualization and analysis of the microstructure a computational model of cement hydration showing four distinct phases. This is the result of research performed by William George, Steve Satterfield, and Edith Enjolras of MCSD in collaboration with Jeffrey Bullard of the NIST Building and Fire Research Laboratory.

Acknowledgement. We are grateful to Robin Bickel for collecting the information and organizing the first draft of this report.

Disclaimer. All references to commercial products in this document are provided only to document how results have been obtained. Their identification does not imply recommendation or endorsement by NIST.

Table of Contents

Part II

Features

Parallel Adaptive Multilevel Finite Elements

The numerical solution of partial differential equations is the most compute-intensive part of a wide range of scientific and engineering applications. As a result, the development of faster and more accurate methods for solving partial differential equations has received much attention in the past fifty years. Nevertheless, many applications at the cutting edge of research remain extraordinarily challenging. These problems necessitate the use of the most advanced numerical techniques and tools such as self-adaptive methods and parallel computers. This year we released the software package PHAML, which serves as a research platform to explore, improve, and apply advanced computational techniques of this type.

William F. Mitchell

The NIST-developed software package PHAML (Parallel Hierarchical Adaptive MultiLevel) can be used to solve a broad class of two-dimensional self-adjoint elliptic partial differential equations (PDEs), including systems of equations and eigenvalue problems, with a variety of boundary conditions. The spatial domain may be any connected region in the plane, including ones with curved boundaries and holes. Problems of this type are found in an extremely wide range of applications, from the study of heat diffusion to the modeling of fundamental properties of atoms.

To enable computer solution, the PDE is discretized by the finite element method. We first partition the domain into a set of triangles (the grid or mesh). We then approximate the solution u by a function

$$\tilde{u} = \sum_i \alpha_i \varphi_i$$

where the φ_i form a set of N piecewise polynomial basis functions defined on the mesh. Inserting this function into a variational formulation of the PDE leads to a system of N linear equations (a matrix) that determines the N coefficients α_i. Considerable skill is required in the selection of the mesh, the basis functions, and the linear solver to obtain a practical solution procedure.

Adaptive refinement is a critical extension of the finite element method when applied to complex simulations where most of the interesting activity is confined to a small part of the domain. For such problems a fine mesh is necessary to resolve rapidly varying portions of the solution. If such a mesh is applied uniformly over the domain the size of the linear system makes the problem intractable. An optimal grid is one in which small triangles are used where the solution varies rapidly and larger ones where it is relatively smooth. With adaptive refinement, such an optimal grid is determined automatically through an iterative process in which, given an initial grid and approximate solution on the grid, those triangles with the largest estimated error are refined to create a new grid yielding a more accurate approximate solution. Fig. 1 shows an example of an adaptively generated grid.

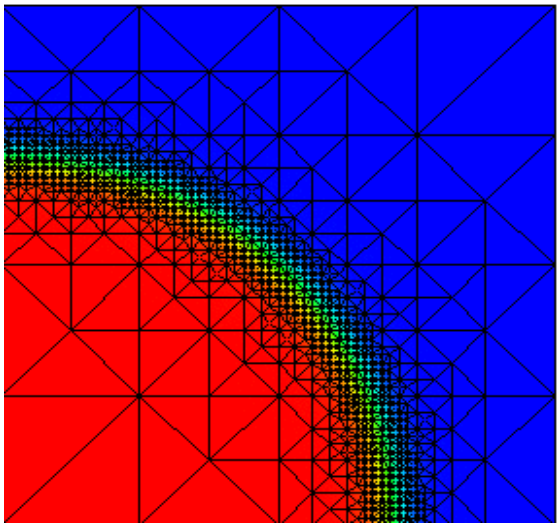


Figure 1. *An adaptively refined grid for a solution that contains a sharp circular wave front.*

The solution can be improved by not only reducing the size of the triangles but also by increasing the degree of the piecewise polynomials. PHAML allows the use of high order polynomials as well as the more common piecewise linear basis functions. Increasing the degree of the piecewise polynomial basis functions can be done locally, resulting in an approximate solution that is not uniform in polynomial degree over the grid.

When both the size of the triangles, h, and the polynomial degree, p, are determined adaptively, it is called *hp*-adaptive refinement. *hp*-adaptive methods are at the cutting edge of finite element research. The reduction of the error as a function of N can be much faster for an *hp*-adaptive method than for just h refinement, and even better than using a fixed high degree, as shown in Fig. 2. PHAML is currently being used to investigate various approaches to *hp*-adaptive refinement.

Even with using adaptive refinement to reduce N, it can still be very large for complex problems, perhaps in the millions. PHAML uses a multigrid solution technique for the linear system. Multigrid can solve the system in a number of operations proportional to N, which is optimal. PHAML is based on hierarchical bases, both in h and p, to give the different multigrid scales which

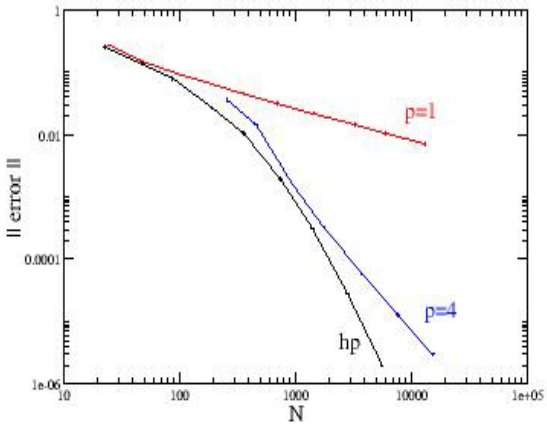


Figure 2. *Behavior of the error as a function of N for adaptive refinement with linear elements, 4th degree elements, and hp-adaptive refinement.*

results in rapid convergence. The use of multiple levels in p is new and is still a topic of active research.

An additional tool for solving complex problems with large N is the use of parallel computers. PHAML was one of the first systems to use both adaptive refinement and multigrid on parallel computers. The research associated with PHAML brought about a new paradigm for parallel implementations called the full domain partition. New algorithms and implementations of multigrid and adaptive refinement, along with the necessary load balancing required to make it successful have been developed under this paradigm [1, 2]. PHAML uses message passing parallelism implemented through the Message Passing Interface (MPI). PHAML is written in Fortran 90 and designed as a collection of modules. Each module deals with one aspect of the program (grid refinement, error estimation, multigrid, etc.) and contains data types and associated operations. The details, including parallelism, are hidden from the user, who simply calls subroutines for operations such as solving the PDE or evaluating the computed solution. The interface to these subroutines contains a very large number of arguments which gives the user a great deal of control of the algorithms. Most arguments are optional, with reasonable default values, so the interface remains simple and clean for routine use. The specification of the problem to be solved (coefficients of the PDE, boundary conditions, domain) are defined by a few subroutines that the user writes. The PHAML User's Guide [3] provides all the information a user of PHAML needs.

Interactive graphics provide another means of obtaining the results of PHAML's computations. The graphics engine is a separate parallel process which receives data from the computational processes by message passing. The user has interactive control of the graphics as PHAML is running. Control includes rotate, zoom and pan operations, and selection of what is to be displayed (solution, error, grid, parallel partition, etc.). Figs. 1 and 3 provide two examples of the graphics that are available. PHAML makes use of the OpenGL graphics library, which is freely available on nearly every computer platform, supporting the goal of providing freely available and highly portable tools.

Version 1 of PHAML was released in May 2007; see http://math.nist.gov/phaml. It has been used by scientists for solving application problems [4, 5], researchers in scientific computing for comparing different methods, and instructors of classes on numerical solution of PDEs and parallel computing.

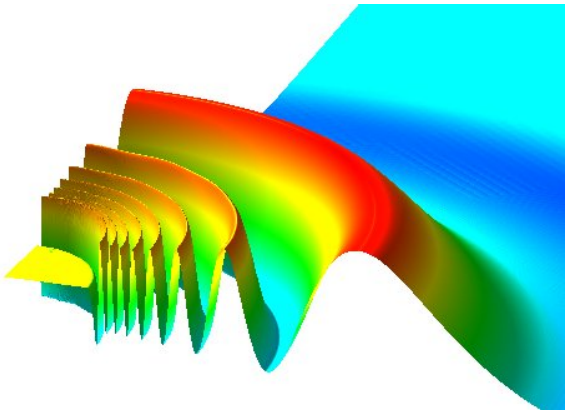


Figure 3. *The computed solution of a model of two interacting atoms, as displayed by the PHAML graphics engine.*

References

[1] W. F. Mitchell, Parallel Adaptive Multilevel Methods with Full Domain Partitions, *Applied Numerical Analysis and Computational Mathematics* **1** (2004), pp. 36-48.

[2] W. F. Mitchell, A Refinement-tree Based Partitioning Method for Dynamic Load Balancing with Adaptively Refined Grids, *Journal of Parallel and Distributed Computing* **67** (2007), pp. 417-429.

[3] W. F. Mitchell, PHAML User's Guide, NISTIR 7374, 2006.

[4] W. F. Mitchell and E. Tiesinga, Adaptive Grid Refinement For a Model of Two Confined and Interacting Atoms, *Applied Numerical Mathematics* **52** (2005), pp. 235-250.

[5] P. Naidon, E. Tiesinga, W. F. Mitchell, and P. Julienne Effective-range Description of a Bose Gas Under Strong One- or Two-dimensional Confinement**,** *New Journal of Physics* **9** (2007) 19.

Participants

W. F. Mitchell, M. McClain (MCSD); E. Tiesinga (PL)

Computable Error Bounds for Delay Differential Equations

A problem of increasing importance in many applications is how to assess the quality of computed results from models involving differential equations. Quite often the best error estimates provide only an order of magnitude assessment rather than rigorous numerical bounds. In the early 1970's A. P. Stokes developed a method of proving the existence of periodic solutions for nonlinear differential and functional differential equations in the neighborhood of approximate solutions which yields precise error bounds. He demonstrated several examples for ordinary differential equations, but gave no computed error bound in the case of functional differential equations. Recently D. E. Gilsinn [1] developed the numerical methods needed to prove the existence of periodic solutions for delay differential equations in the neighborhood of approximate periodic solutions and calculating exact error bounds. Here we provide some background on delay differential equations, outline the argument that leads to exact error bounds, and give an example of the application of the method to a classic Van der Pol equation with delay.

David E. Gilsinn

Systematic work with mathematical models in medicine and biology involving differential equations with delay terms began in the early 1900's with the studies of R. Ross [5] on problems in malaria epidemiology. This early work was extended during the period 1920-1940 for applications in areas such population ecology, business cycles, and control systems [3, 4, 7]. All of these studies pointed out the need to consider the "transmission time" between when an input is introduced into a system and when a response is expected, that is, the *delay* inherent in the system.

Such problems have also arisen in manufacturing. Machine tool chatter is a self-excited oscillation of a cutting tool against the surface being machined. It typically is heard as a high frequency noise in a machine shop. In the manufacturing literature it is referred to as regenerative chatter. In the 1950's Tobias and Fishwick [8] developed models to explain this self-excited oscillation using delay terms. In the process of a turning operation with a lathe, the surface being cut at a certain time is affected by the results from the cutting tool during the cut at the previous revolution on the same surface. Any slight imperfection in this previous cut affects the current cut and can start an oscillation that could generate chatter.

Early work in the studies of machine tool chatter led to linear delay differential equations of the form

$$m\ddot{x}(t)+\frac{h}{\omega}\dot{x}(t)+\lambda x(t)=-k\big(x(t)-x(t-T)\big), \quad (1)$$

where the coefficients are determined from experiments. A linear theory would predict, however, that the amplitude of vibration would increase indefinitely once the depth of cut exceeded some critical value. In the early 1970's Hanna and Tobias [2] introduced cubic nonlinearities in the model and modified the equation to the form

$$\begin{aligned} &m\ddot{x}(t)+\frac{h}{\omega}\dot{x}(t)+\lambda(x(t)+b_1x(t)^2+b_2x(t)^3) \\ &=-k[(x(t)-x(t-T)) \\ &\quad +C_1(x(t)-x(t-T))^2 \\ &\quad +C_2(x(t)-x(t-T))^3]. \end{aligned} \quad (2)$$

With this modification some good qualitative correspondence between predictions and theoretical studies was obtained.

Many such problems in biology, population dynamics, and machining can be written in the vector form

$$\omega\,\dot{x}(t)=X(x(t),x(t-\omega)), \quad (3)$$

where we look for periodic solutions with a frequency of ω on the interval $[0,2\pi]$. Approximate periodic solution can be found by optimization methods using finite trigonometric series of the form

$$\hat{x}_m=a_2\cos(t)+\sum_{n=2}^{m}\left[a_{2n}\cos(nt)+a_{2n-1}\sin(nt)\right] \quad (4)$$

The sin(*nt*) term is often dropped so that one can estimate the frequency $a_1=\overline{\omega}$. To shorten notation one usually uses $x_h = x(t\text{-}h)$. The problem then is to determine conditions for which equation (3) has an exact periodic solution with frequency ω and determine numerically computable error bounds for the approximate periodic solution and frequency. There exist programs that allow you to compute numerical solutions to (3); however they usually can only give approximate estimates of error bounds based on grid size.

The fundamental argument that establishes the existence and bounds for the approximate periodic solution and frequency rests on generalizations of two results, one from linear algebra, and one from elementary optimization theory. The linear algebra result can be stated as follows. The algebraic system $Ax=b$ has a solution if and only if $b^T\overline{u}=0$ for all solutions of $A^*u=0$, where $A^*=\overline{A}^T$ is the adjoint, and the overbar represents complex conjugates. This result states that the system $Ax=b$ has a solution if and only if the right hand side, b, is orthogonal to all solutions of the adjoint system. The result is sometimes referred to as the solvability condition. Its extension into the space of

functions and operators is often referred to as the Fredholm alternative. The other result comes from optimization theory and is called the fixed point theorem. In effect, we are given a function, *S(x),* defined on some space, and we look for a solution to the problem *x = S(x).*

Assume that one has constructed an approximate 2π-periodic solution as in (4) and an approximate frequency $\hat{\omega}$. If you substitute these into (3) you will get a system similar to (3) but with an error term

$$\hat{\omega}\dot{\hat{x}} = X(\hat{x}, \hat{x}_{\hat{\omega}}) + k, \tag{5}$$

where *k(t)* is 2π-periodic and bounded by some constant *r* so that $| k(t) | \leq r$. In order to study any solutions to (3) in the neighborhood of $(\hat{\omega}, \hat{x})$ one introduces the variational equation about $(\hat{\omega}, \hat{x})$

$$\hat{\omega}\dot{z} = dX(\hat{x}, \hat{x}_{\hat{\omega}}; z, z_{\hat{\omega}}) = \hat{A}z + \hat{B}_{\hat{\omega}}, \tag{6}$$

where $\hat{A} = X_1(\hat{x}, \hat{x}_{\hat{\omega}})$, $\hat{B} = X_2(\hat{x}, \hat{x}_{\hat{\omega}})$. The adjoint equation relative to (6) is given by

$$\hat{\omega}\dot{\nu} = -\nu\hat{A} - \nu_{-\hat{\omega}}\hat{B}. \tag{7}$$

An important relationship between the 2π-periodic solutions of (6) and (7) is that they both have the same finite number of 2π-periodic solutions.

In order to state a property that $(\hat{\omega}, \hat{x})$ must satisfy we define a characteristic multiplier. ρ is a characteristic multiplier of the linear delay equation

$$y'(t) = A(t)\, y(t) + B(t)\, y(t-\omega) \tag{8}$$

where *A(t), B(t)* are 2π-periodic, if there is non-trivial solution *y(t)* of (8) such that

$$y(t + 2\pi) = \rho\, y(t) \tag{9}$$

Note: *y(t)* would be a 2π-periodic solution of (8) if ρ=1.

There is a property that $(\hat{\omega}, \hat{x})$ must satisfy that gives a condition that essentially guarantees that $(\hat{\omega}, \hat{x})$ is in some sense isolated. In particular, $(\hat{\omega}, \hat{x})$ is said to be non-critical if the variational equation (6) with respect to $(\hat{\omega}, \hat{x})$ has a simple characteristic multiplier, ρ_0, that need not be unity, and all other characteristic multipliers are not unity. Next, if ν_0 is the single solution of the adjoint equation (7) associated with ρ_0, then

$$\int_0^{2\pi} \nu_0^T J\left(\dot{\hat{x}}, \hat{\omega}\right) dt \neq 0, \tag{10}$$

where

$$J\left(\dot{\hat{x}}, \hat{\omega}\right) = \dot{\hat{x}} + \hat{B}\dot{\hat{x}}_{\hat{\omega}}$$

The form of the Fredholm alternative that is relevant here is as follows. The nonhomogeneous system

$$\hat{\omega}\dot{x} = \hat{A}x + \hat{B}x_{\omega} + f, \tag{11}$$

has a unique 2π-periodic solution if and only if

$$\int_0^{2\pi} \nu_0^T f\, dt = 0 \tag{12}$$

for all independent solutions ν_0 of (7). Furthermore there exists a constant $M>0$, independent of *f*, such that

$$| x | \leq M | f |. \tag{13}$$

This is just an operator version of the linear algebra solvability condition.

One looks for an exact solution and frequency for (3) by perturbing the approximate solution and frequency. To do this one can introduce the perturbations

$$\omega = \hat{\omega} + \beta, \qquad x = \hat{x} + \frac{\hat{\omega}}{\omega} z. \tag{14}$$

If these are substituted into (3), then a perturbed variational equation about the approximate solution can be written in the form

$$\hat{\omega}\dot{z} = dX(\hat{x}, \hat{x}_{\hat{\omega}}; z, z_{\hat{\omega}}) + R(z, \beta) - \beta J(\dot{\hat{x}}, \hat{\omega}) - k, \tag{15}$$

where $R(z,\beta)$ is a rather long error term, but is only a function of z and β defined in (14). Now consider an associated equation

$$\hat{\omega}\dot{z} = dX(\hat{x}, \hat{x}_{\hat{\omega}}; z, z_{\hat{\omega}}) + g - \beta J(\dot{\hat{x}}, \hat{\omega}) - k. \tag{16}$$

and apply the Fredholm solvability result to (16). In particular, if $(\hat{\omega}, \hat{x})$ is noncritical with respect to (3), then there exists a unique β, designated $\beta(g)$, such that

$$g - \beta(g)\, J(\dot{\hat{x}}, \hat{\omega}) - k \perp \nu_0, \tag{17}$$

where ν_0 is the solution of the adjoint equation (7) corresponding to the characteristic multiplier ρ_0 of equation (6). Furthermore there exists a 2π-periodic solution of (16), designated by $z(g)$, that satisfies

$$\left|z(g)\right| \leq M\left| g - \beta(g) J(\dot{\hat{x}}, \hat{\omega}) - k \right|, \tag{18}$$

for some $M>0$. Finally there exist computable constants, λ_0, λ_1, such that

$$\begin{aligned} | \beta(g) | &\leq \lambda_0 (|g| + r) \\ | z(g) | &\leq \lambda_1 (|g| + r) \end{aligned} \tag{19}$$

The fixed point property can now be used to obtain the final result. Define a map

$$S(g) = R(z(g), \beta(g)). \tag{20}$$

This map satisfies the following properties that provide sufficient conditions for a fixed point. There exist two functions $E_1(\delta)$, $E_2(\delta)$ and two positive constants F_1, F_2 so that

$$\begin{aligned} &\left|S(g)\right| \leq E_1(g), \\ &\left|S(g) - S(\tilde{g})\right| \leq E_2(g)\left|g - \tilde{g}\right|, \end{aligned} \tag{21}$$

where

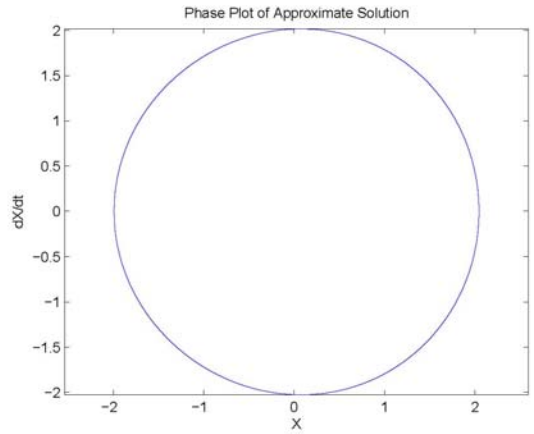


Figure 2. Phase *plot of approximate solution for the Van der Pol equation*

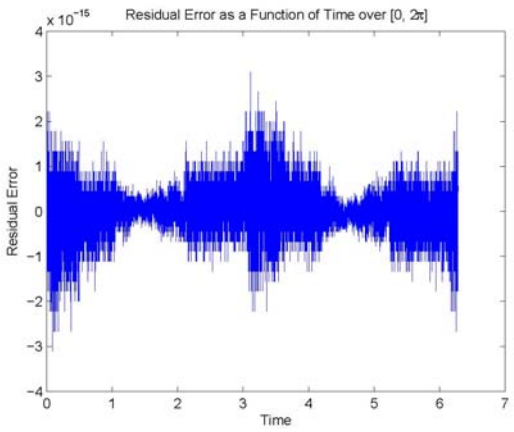


Figure1. *Residual error of approximate solution for the Van der Pol equation*

$$E_1(\delta) \le F_1 \delta^2, \qquad (22)$$
$$E_2(\delta) \le F_2 \delta.$$

The final result can then be stated as follows. If $(\hat{\omega}, \hat{x})$ is noncritical with respect to (3) and δ is selected so that

$$\delta \le \min\{1/F_1, 1/2F_2, \hat{\omega}/4\lambda_0\} \qquad (23)$$

With $r \lesssim \delta$, then there exists an exact frequency ω, and solution of (3) such that

$$|x - \hat{x}| \le 4\lambda_1 \delta, \qquad (24)$$
$$|\omega - \hat{\omega}| \le 2\lambda_0 \delta.$$

Although there seems to be a large number of parameters to compute, they can all be estimated. Of all the parameters the one that tends to be the most critical is M in (18).

This result was applied to the classic Van der Pol equation with delay in the form

$$\omega^2 \ddot{x} + \omega\lambda\left(x(t-\omega)^2 - 1\right)\dot{x}(t-\omega) + x = 0 \qquad (25)$$

For $t \in [0, 2\pi]$. An approximate solution and frequency were estimated by a Galerkin projection method as

$$\begin{aligned} \hat{x}(t) &= 2.0185\cos(t) \\ &+ 2.5771\times10^{-3}\sin(2t) + 2.5655\times10^{-2}\cos(2t) \\ &+ 1.0667\times10^{-4}\sin(3t) - 5.2531\times10^{-4}\cos(3t) \\ &- 7.1780\times10^{-6}\sin(4t) - 2.2043\times10^{-6}\cos(4t), \\ \hat{\omega} &= 1.0012, \end{aligned} \qquad (26)$$

where only the first few harmonics have been displayed. The residual was estimated by substituting (25) into (24) and computing the maximum value over the interval [0, 2π]. The result was $r = 3.1086\times10^{-15}$. The distribution of errors is shown in Fig. 1 and the approximate solution if shown in Fig. 2. Some lengthy estimates yield the following parameter values $M = 2.7618\times10^2$, $\lambda_0 = 8.4091$, $\lambda_1 = 6.6736\times10^3$. One can then compute $F_1 = 2.5941\times10^9$, $F_2 = 1.0798\times10^{10}$, $\delta = 4.6305\times10^{-11}$. With these values (21) and (22) will produce a convergent fixed point iteration. One can then finally estimate the error between the approximate frequency and solution and the exact as

$$|x - \hat{x}| \le 1.2361\times10^{-6}, \qquad (27)$$
$$|\omega - \hat{\omega}| \le 7.7877\times10^{-10}.$$

References

[1] D. E. Gilsinn, Computable error bounds for approximate periodic solutions of autonomous delay differential equations, *Nonlinear Dynamics* **50** (2007) pp. 73-92.

[2] N. H. Hanna and S. A. Tobias, A theory of nonlinear regenerative chatter, *ASME Journal of Engineering for Industry* **96** (1974) pp. 247-255.

[3] J. Lotka, Contributions to the analysis of malaria epidemiology: I General part, *Supplement to American Journal of Hygiene* **3** (1923) pp. 1-37.

[4] N. Minorski, Self excited oscillations in dynamical systems possessing retarded action, *Journal of Applied Mechanics* **9** (1942) pp. 65-71.

[5] R. Ross, *The Prevention of Malaria*, 2nd edition., London, John Murray, 1911.

[6] P. Stokes, On the existence of periodic solutions of functional differential equations, *Journal of Mathematical Analysis and Applications* **54** (1976), pp. 634-652.

[7] J. Tinbergen, Annual survey: suggestions on quantitative business cycle theory, Econometrica, 3 (1935) 241-308.

[8] S. A. Tobias and W. Fishwick, The chatter of lathe tools under orthogonal cutting conditions, *Transactions of the ASME* (1958) pp. 1079-1088.

Participants

David E. Gilsinn, Sita Ramamurti

Making Optical "Schrödinger Cat" States

We are developing technologies that exploit exotic quantum effects in optical systems for practical use. For example, photons may be useful for encoding information as quantum bits (qubits) for quantum computation and communication. Quantum entanglement between photons can also be used to enhance the sensitivity of interferometers. The ability to prepare and measure a particular class of photon states called "Schrödinger cat" states is a prerequisite to developing applications of this type. We have completed an analysis of many cat creation schemes, and are working with the NIST Electronic and Electrical Engineering Lab (EEEL) to build an experiment that will produce them.

Scott Glancy

Quantum optics is a field rich with potential for developing new technologies and fundamental physics experiments. There is a strong history of optical experiments testing and verifying some of the foundations of quantum theory such as superpositions of states and entanglement. Researchers are now developing useful technologies that exploit these quantum phenomena. Some examples include quantum computation, communication, and cryptography, interferometry, and lithography. Because of the extreme fragility of the quantum systems, creating and maintaining superpositions of states of many photons and entangling them is still very challenging. We are developing strategies for optical state preparation, manipulation, and measurement, and we are building experiments to implement these strategies.

We have completed a thorough analysis of many schemes for making a class of optical states, which we call "Schrödinger cat" states [1]. These states contain an equal superposition of two coherent states of opposite phase. We have made significant progress toward making cat states in the laboratory.

A classical beam of light, produced for example by a laser, is composed of oscillating electric and magnetic fields. As a classical light wave travels through space its electric field vector oscillates up and down between two extreme positive and negative values. However, for light beams in "Schrödinger cat" states the electric field vector exists in a coherent superposition of pointing both up and down. At each of the wave's antinodes one may measure a large positive or negative electric field, each with probability 1/2, as shown in Fig. 1. At the node, one expects to find zero electric field, but instead we see a probability distribution centered at zero with some interesting structure. This is caused by quantum interference between the electric-field-up and the electric-field-down parts of the cat state. In Schrödinger's thought experiment a diabolical device produces a cat which is in superposition of dead and live. Our cat states are analogous in that they exist in a superposition of two distinct electric field configurations. Of course they are quite different from Schrödinger's original cat because they contain only a few photons and do not purr.

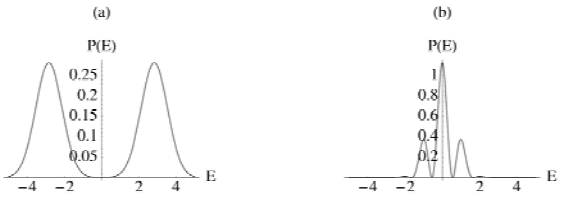


Figure 1. *Part (a) shows the probability distribution for measuring electric field* E *at an anti-node of a light beam in a cat state. (b) shows the probability distribution at a node. The electric field units are arbitrary.*

Cat states are extremely fragile, because the absorption of a single photon destroys the superposition, washing out the interference seen in Fig. 1(b). Cat states containing many photons are very sensitive to absorption, because the probability that zero photons are lost decreases exponentially with the number of photons.

Cat states are necessary for several future technologies. One proposal for an optical quantum computer represents its quantum bits with beams of light whose electric fields are pointing up (logical 1) or down (logical 0). The cat state is an equal superposition of these states. This scheme also requires cat states as a resource for logic operations and teleportation of qubits.

Cat states may also be useful for precision measurements of small distances. When interferometry is performed with classical light, a laser beam is split in two; the two parts are then recombined and allowed to interfere with one another, creating a pattern of light and dark fringes spaced every $\lambda/2$, where λ is the light's wavelength. If the difference in the path lengths traveled by the two beams changes, then the pattern of fringes will shift. The uncertainty in the measured distance is proportional to $\lambda/\sqrt{N}$, where N is the mean number of photons in the beam. If instead cat states are used in the interferometer, the distance between fringes (similar to that in Fig. 1(b)) is proportional to $\lambda/\sqrt{N}$, and the uncertainty of the distance measurement is λ/N. However, to take advantage of this enhanced interference in the cat state, one must ensure that none of the cat's photons are absorbed. Furthermore, current experiments struggle to create cat states with $N>1.5$, while conventional lasers can easily produce classical

beams with $N>10^{10}$. The enhanced measurement precision is likely to be useful only in environments with low photon loss and limited laser power.

The creating of cat states remains an experimental challenge. The original proposal for transforming classical laser light into a cat requires a "Kerr" material that has both a strong nonlinear interaction and low photon absorption. Unfortunately our calculations show that the best currently available material (fused silica fibers) has a ratio of nonlinear strength to photon absorption that is nearly 300 times too small. Progress is being made in engineering Kerr materials, for example using photonic crystals and electromagnetically induced transparency.

In the meantime, we require a more clever method to make cats. Several methods exist which exploit a general strategy of using an "easier" nonlinear interaction, splitting off part of the light, and making a measurement. Depending on the measurement result one may infer that a cat has been created in the remaining light beam. Each scheme exploits some trade-offs in the experimental requirements such as the type of nonlinearity used and the efficiency and noise level of photon detectors. These choices impact the degree to which the created state approximates an ideal cat, the number of photons the cat contains, the probability with which a cat is produced in each attempt, etc.

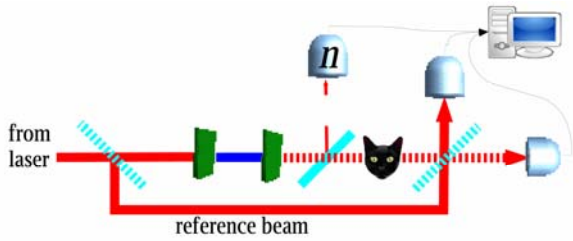


Figure 2. *Diagram of photon subtraction scheme to make cat states. Red laser light enters from the left. The red light is converted to blue through frequency doubling in the green $KNb0_3$ crystal. The blue light is then converted back to red during downconversion, creating the squeezed light. When the photon counter registers n photons, we know a cat has been created, which we verify in a final measurement.*

The simplest of these schemes, "photon subtraction", is shown in Fig 2. We first create a squeezed state of light through the lower order nonlinear process of frequency downconversion. During downconversion photons of frequency ω are split into two photons with frequency ω/2. We then use a mostly silvered mirror (with transmissivity 99%) to split off a small fraction of the squeezed beam, which is then sent to a photon detector. When the photon detector clicks, we know that a photon has been subtracted from the squeezed beam, which has now been transformed into a state that closely approximates a cat state.

In collaboration with members of the EEEL, we are implementing a photon subtraction experiment in the laboratory. The current optics arrangement appears in Fig. 3. Two crucial elements for the experiment's success are the purity of the squeezed state created during downconversion and the efficiency of the photon detector. Our collaborators in EEEL have developed a high-efficiency, low noise photon detector which is capable of counting the number of photons. Our simulations show that subtracting multiple photons will significantly enhance the fidelity and size of the cat. We are also investigating innovative methods for high purity squeezing and algorithms for inferring what state our experiment has created from measured data.

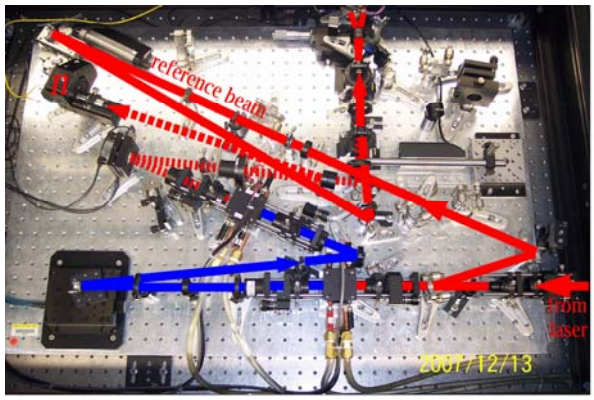


Figure 3. *Photograph of cat making laboratory experiment. Light enters at the lower right, and the final cat verification measurement occurs at the top, where the cat and reference beams are combined and then split.*

This work is a component of the Quantum Optical Metrology Innovations in Measurement Science project, the goal of which is to create a "test-bed" of quantum optical state preparation (including cat state and others), manipulation, and measurement. Such a test-bed and the competence we have gained building it will allow us to explore and evaluate new technologies that exploit the quantum nature of light, including quantum information and measurement applications.

References

[1] S. Glancy and H. M. Vasconcelos, Methods for Producing Optical Coherent State Superpositions, http://arxiv.org/abs/0705.2045, *Journal of the Optical Society of America B*, to appear.

Participants

S. Glancy, E. Knill (MCSD); T. Gerrits, T. Clemment, S. W. Nam, R. Mirin (EEEL); H. M. Vasaconcelos (Universidade Federal do Ceará, Brazil)

Modeling the Rheological Properties of Suspensions

Understanding the mechanisms of dispersion or agglomeration of particulate matter in complex fluids, such as suspensions, is of technological importance in many industries such as pharmaceuticals, coatings, and concrete. These fluids are disordered systems consisting of a variety of components with disparate properties that can interact in many different ways. Modeling and predicting the flow of such systems represents a great scientific and computational challenge requiring large-scale simulations.

Our goal in this project is to advance our understanding of the flow properties (rheology) of a specific suspension, fresh concrete, which is composed of cement, water, sand, and stones. Concrete rheology has a tremendous impact on the construction of concrete structures, an industry with a $110B per year impact on the US economy. In collaboration with scientists in the NIST Building and Fire Research Laboratory (BFRL), we are developing a dissipative particle dynamics code, called QDPD, which is capable of performing large scale simulations of suspensions. QDPD is highly parallel and has been shown to efficiently scale up to at least 1,000 processors when running on the NASA supercomputer Columbia. .

William George

Background. Fluids can be characterized by properties known as yield stress and viscosity, each of which is a function of shear rate. Yield stress is the force applied per unit area to initiate the flow. Viscosity is the applied force per unit area needed to maintain a shear rate. Shear rate is the velocity gradient perpendicular to the flow direction. Many factors control viscosity and yield stress. In building materials like concrete (a combination of water, cement, sand, and stones) viscosity and yield stress depend on the ratio of water to cement, the volume of sand or stones used, as well as their shape and size distribution. In addition, there can be great variability in each of the component materials themselves. As a "concrete" example, rocks mined in quarries are usually angular because they are crushed when processed, while rocks taken from river beds are typically rounded due to erosion. It turns out that the more uniform the size of the rocks the harder it is to get a concrete suspension composed of those rocks to flow. In this case, the concrete fluid may actually jam when poured through a narrow opening thus causing delays in construction. Clearly, to optimize the flow properties of complex suspensions, one needs to understand the relationship between flow and properties of the fluid's constituents.

QDPD. Modeling and predicting the flow of complex disordered systems of this type represents a great scientific challenge. Simply accounting for the size and shape variation of the solid components of the suspension (cement particles, sand or rocks) presents difficult computational problems. Modeling a representative system may entail keeping track of up to 100,000 solids of varying shape and size. Further, many of the forces between rocks (aggregate) depend on the local surface curvature of the aggregate at points close to neighboring aggregates. This requires keeping track of the location, orientation and shortest distance between neighboring solids. Clearly, predicting the properties of such systems based upon modeling of such fundamental interactions necessitates large-scale simulations.

We have adopted and developed some novel modeling approaches originally based on cellular automata which can successfully take into account many of the complex features of a suspension. QDPD [1], which stands for Quaternion-based Dissipative Particle Dynamics (DPD), uses the recently developed DPD technique [2] to simulate the fluid and its interaction with the inclusions of the suspension. We have added other forces are to this system to better account for the interaction between the large particles in the suspensions, the inclusions, e.g., lubrication forces that help keep the inclusions separated and van der Waals forces that introduce an attractive inter-particle force. Brownian forces are also added to maintain system temperature. Our original (serial) code was validated by both theory and experiments on idealized systems and has been extended to account for random-shaped objects with different inter-particle interactions.

We have recently enhanced QDPD to utilize the power of large parallel machines, regularly using 500+ processors, and sometimes using 1,000 or more processors, to simulate systems composed of over 32,000 inclusions and one million fluid particles. QDPD remains under constant development to improve its capabilities, such as ability to simulate non-Newtonian fluids, as well as to improve its parallel performance.

NASA's Columbia supercomputer. During the past year we were awarded 1 million CPU-hours of compute time on the NASA supercomputer Columbia to perform simulations with QDPD. Located at NASA'a Ames Research Center in California, Columbia is a 10,240-CPU system based on SGI's NUMAflex architecture. Not only has this access enabled us to perform very large simulations, it has allowed us to analyze the effect of varying system parameters such as the density of inclusions in the suspensions, the applied shearing speed, and the distribution of inclusion sizes. With the added compute power we were also able to move from

spherical inclusions to realistically shaped inclusions for a more accurate simulation.

Results. We have developed a new scalable parallel algorithm that has enabled us to make advances in understanding the influence of finite size effects, stress transmission, time scales and system equilibration. Through numerical modeling and visualization, this research has provided greater insights into the physical mechanisms associated with the onset of flow. Aspects of yield stress and viscosity can be linked to spatio-geometric properties of suspensions including the number of neighboring rocks and their relative orientation. Further, by examining the very long time scale behavior of the rocks, we can link their motion to visco-elastic properties of the suspension.

Hybrid Realistic / Non-Photorealistic Visualization. Visualization has proven to be critical to understanding the nature of the results generated from large-scale simulations of this type. We visualize the rocks directly, but augment the visualization with embedded data to provide additional information on interactions between rocks and the per-rock stress (see Fig. 1). The addition of these values allows quantitative observations and measurement, as opposed to the purely qualitative insight provided by most visualizations. In addition, blue dots show the points of interaction between rocks in the simulation. As the suspension becomes denser, it becomes increasingly difficult to find the high-stress particles in a visualization, much less chains of such particles. While a smaller system may be simple enough to understand without further visualization tools, the larger system in Fig. 1 and our biggest system to date, 2025 rocks, require extra tools, such as culling the less interesting rocks, in order to allow the scientist to find potential jamming.

In some cases the user would like to focus on the highest stress rocks, but still have some indication of the surrounding rocks. To achieve this, we turn to non-photorealistic rendering techniques to show these rocks in a sketchy style. The result, shown in Fig. 2, is quite effective, especially when seen in motion. The coherent motion of each low-stress rock's silhouette enhances the ability to recognize and distinguish individual rocks. Here we can clearly see a diagonal chain of rocks from the upper right, wrapping briefly to the opposite side (the simulation wraps toroidally).

In addition, it is useful to track the high-stress rocks from a single frame of through the entire simulation. That allows the scientist to see how those specific rocks move into gridlock, and how they escape, and continue on afterwards. For example, in some simulation runs, a small rock can be seen to move and twist to a new location while larger rocks remain locked in place. Then the larger rocks release, the stress goes down, and the aggregate motion continues.

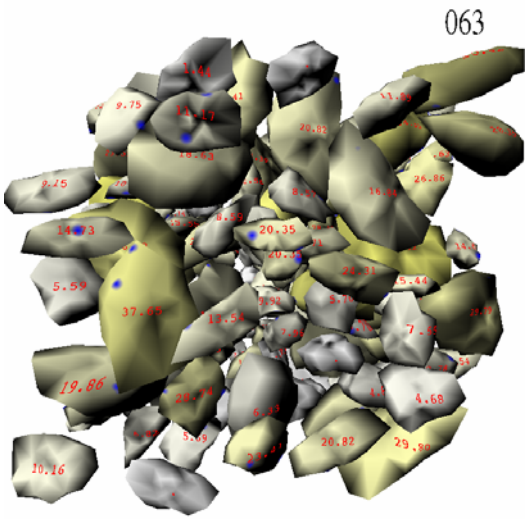


Figure 1. *Frame 63 in a simulation of the flow of rocks in a suspension. The per-rock stress is shown with a gray-yellow color scale, and also with a numeric value printed on the face for each rock.*

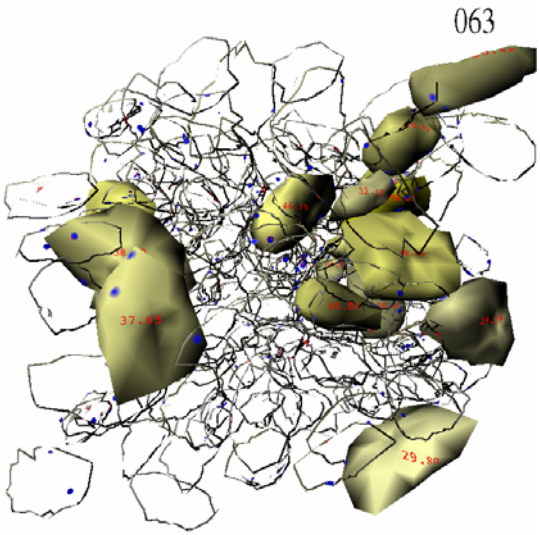


Figure 2. *Same frame 63 in a simulation of the flow of rocks in a suspension. Lower stress rocks are shown in sketch mode.*

In addition, it is useful to track the high-stress rocks from a single frame of through the entire simulation. That allows the scientist to see how those specific rocks move into gridlock, and how they escape, and continue on afterwards. For example, in some simulation runs, a small rock can be seen to move and twist to a new location while larger rocks remain locked in place. Then the larger rocks release, the stress goes down, and the aggregate motion continues.

The sparse visualization of Fig. 2 makes it much easier to see and understand the per rock digital stress measurements. The stress value is centered on the face of the rock closest to the viewer, so as the system is rotated and manipulated, whether on the desktop or in the immersive visualization environment, the text always

remains at the correct orientation. Some interactive rotation is generally necessary to see the values on hidden rocks, but with many fewer rocks displayed as solid photo-realistic entities, it is straightforward to find a suitable view to read the data from each rock.

To assist the scientist in choosing a frame to display, we created an input analysis tool providing a graph of the total system stress per frame, with a slider to indicate a frame of interest (Fig. 3). For example, frames just before a steep drop-off in stress are of particular interest since they may indicate a time when the rocks in the aggregate were binding, but then moved suddenly to release the pressure. Other interactions with the GUI enable controlling other visual features.

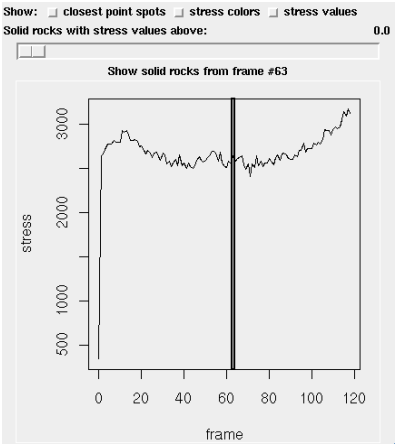


Figure 3. *GUI for interactive exploration of the visualization.*

Shading-based visualization environment. This set of visualizations was created with a procedural-shading-based add-on to our visualization environment. A shader is a procedure which determines the color (i.e. shading) and opacity of each point on a surface. Our shaders are written in the OpenGL Shading Language and run in the graphics hardware at each pixel on each rock. To determine the color and opacity of each pixel, the shader refers to the current rock and frame number, position of the pixel on the rock, and data packed into floating point texture variables. This procedure is responsible for creating the sketchy appearance for less important rocks, the stress color scale for solid rocks, placing the spots on the rock surface, and writing the numeric values on each rock.

This shading framework provides a very flexible mechanism for experimenting with new visualization ideas. The shading procedure can be modified and reloaded while the application is running. It can use an arbitrary number of slider-controllable parameters for interactive manipulation of the shader's behavior. For example, we can interactively change the frame used to decide which rocks are solid, and the stress level at which we show a blue closest point and the level to show a solid rock rather than an outline.

Graphics hardware has a number of limitations that impact shaders. Some, like the maximum available texture memory, are hard limits (512 MB on our system), while others, like the number of texture accesses made by the shader, are soft limits affecting the frame rate and interactivity of the resulting visualization. For both reasons, we must process our sometimes multi-gigabyte raw data into a compact form to be stored in the limited texture memory for use by the shader. We keep the total stress for each rock on each frame, as well as the closest point location and stress for the highest-stress closest points for each rock on each frame. In each case, we need to turn a 3x3 stress tensor into a single scalar stress value for use and comparison, and combine these to produce a single stress value per rock. There are two principal choices, referred to as shear stress and normal stress, although in some cases we have performed visualization using distance or log stress. Rather than make a single choice, our software takes an arbitrary expression to guide texture creation.

These visualization methods provide insight by combining several quantitative methods to reveal spatial-temporal relationships. The user can find regions of interest rocks by visualizing lines of stress and embedded numerical values. They can graphically ask interactive questions and easily reduce their search for areas of interest. Our scientific collaborators report that the visualizations enable them to validate the physical correctness of the simulation, to detect problems, and to tune parameters of the model.

References

[1] N. Martys, Study of a Dissipative Particle Dynamics Based Approach for Modeling Suspensions, *Journal of Rheology* **49** (March/April 2005), pp. 401-424.

[2] P. J. Hoogerbrugge and J. M. V. A. Koelman, Simulating Microscopic Hydrodynamic Phenomena with Dissipative Particle Dynamics, *Europhysics Letters* **19** (1992), pp. 155-160.

[3] J. Terrill, W. George, T. Griffin, J. Hagedorn, J. Kelso, M. Olano, A. Peskin, S. Satterfield, J. Sims, J. Bullard, J. Dunkers, N. Martys, A. O'Gallagher, and G. Haemer, Extending Measurement Science to Interactive Visualization Environments, in *Trends in Interactive Visualization: a State-of-the-Art Survey* (E. Zudilova-Seinstra, T. Adriaansen and R. van Liere, eds.), Springer, to appear.

Participants

William L. George, Julien Lancien, Steve Satterfield, Marc Olano, Judith Terrill (MCSD); Nicos Martys, Edward Garboczi (NIST BFRL)

http://math.nist.gov/mcsd/savg/vis/fluid/

Computation, Visualization of Nano-structures and Nano-optics

Research and development of nanotechnology, with applications ranging from smart materials to quantum computation to biolabs on a chip, has the highest national priority. Semiconductor nanoparticles, also known as nanocrystals and quantum dots, are one of the most intensely studied nanotechnology paradigms. Nanoparticles are typically 1 nm to 10 nm in size with a thousand to a million atoms. Precise control of particle size, shape and composition allows one to tailor charge distributions and control quantum effects to tailor properties completely different from the bulk and from small clusters. As a result of enhanced quantum confinement effects, nanoparticles act as artificial, man-made atoms with discrete electronic spectra that can be exploited as light sources for novel enhanced lasers, discrete components in nanoelectronics, qubits for quantum information processing, and enhanced ultrastable fluorescent labels for biosensors to detect, for example, cancers, malaria or other pathogens, and to do cell biology. We are working with the NIST Physics Laboratory to develop computationally efficient large scale simulations of such nanostructures. We are also working to develop immersive visualization techniques and tools to enable analysis of highly complex computational results of this type.

James Sims

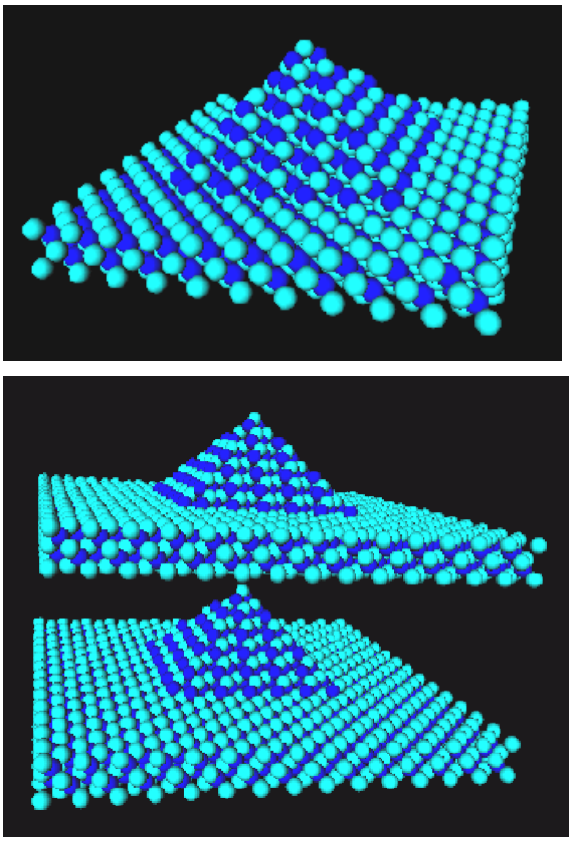


Figure 1. *A pyramid structure and a double pyramid structure of InAs quantum dots embedded in GaAs. The surrounding matrix of GaAs is not shown, but would be included in calculations. Coupling between InAs dots is done through the intervening GaAs matrix.*

We study the electrical and optical properties of semiconductor nanocrystals and quantum dots such as the pyramidal dot shown in Fig. 1. We also study more complex nanocrystal structures, with the nanocrystal coordinated with capping molecules and functionalized with linker molecules, and nanodevice architectures formed by linking together complex dot structures, also shown in Fig. 1. In the most complex structures this entails modeling on the order of a million atoms. Highly parallel computational and visualization platforms are critical for obtaining the computational speeds necessary for systematic, comprehensive study of such structures.

Parallelization. Often it is easy to define the simple nanosubsystems that make up a complex, heterogeneous nanosystem. However, it may be difficult to explicitly define the entire structure. A novel feature of our code is the ability to link together heterogeneous nanostructures (also referred to here as nanosubsystems). For example, when a nanoparticle nanostructure includes conjugating and linker molecules, these molecules can be assigned separately to different computational nodes to take advantage of the parallelization. If the nanosystem includes multiple smaller nanosubsystems linked together, then each smaller nanosubsystem can be parallelized on a different set of nodes with only minimal communication required between different nodes. Since we can do multiprocessor runs routinely, we have the basic building blocks for making larger structures by "stitching" together disparate subsystems into composite structures, each separate subsystem to be stitched together being a smaller multiprocessor run. The basic idea is to consider each smaller nanosubsystem as its own cluster, using the same input data as before, but at each iteration in the computation, information about atoms in the cluster that are intercluster neighbors (see Fig. 2) has to be distributed to the appropriate processors for the neighboring clusters, thereby "stitching" the calculations on the nanosubsystems in the heterogeneous structure together.

Now that we can routinely model nanosystems, our focus has shifted to modulating and controlling the optical properties of these self-assembled quantum dots using external strain which is modeled using an em-

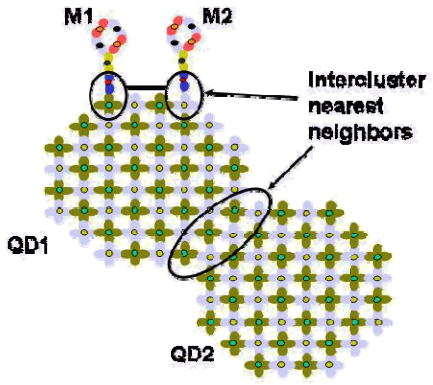


Figure 2. *Illustration of intercluster nearest neighbors in a nanosystem with four subsystems: two quantum dots (QD1 and QD2) and two conjugating molecules (M1 and M2).*

pirical valence force field. External strain leads to an internal lattice relaxation which manifests itself in a modification of the electron and hole states which are computed for the quantum dots. In this way the optical properties of the quantum dots are modified. A paper on this new development has been recently delivered at the Material Research Society meeting in Boston [3].

Visualization. Visual models of laboratory experiments and computational simulations to explore the nanoworld can be critical to comprehension. However, increasing amounts of data are being generated. For example, in one of our calculations, the region considered has nearly 700,000 atoms. Since each atom has 5 orbitals, there are 3.5 million pieces of data to describe one state. Both high performance computing and experiment must be augmented by high performance visualization. At NIST our visual analysis capabilities include both coarse grain capabilities and finer grain capabilities (which are more demanding of CPU and visualization resources), as well as static graphical representations and fully 3D immersive capabilities.

In our quantum dot simulations we visualize the atomic scale structure of the lattice and the charge density of the electrons and holes at both the fine grain and coarser grain levels. At the finer grained level, we visualize the charge density. For example, by displaying the contribution of s and p orbitals to the charge distribution of an eigenstate of a triple quantum dot structure, we are able to visualize when and how tunneling occurs between the structures and what orbitals are involved.

Finer detail can be represented in our visualizations. Fig. 3 shows the charge density of the lowest hole state in a CdS nanocrystal. In this case, much greater detail is apparent. The contributions from p_z orbitals (green) and p_x orbitals (blue) are shown. The contributions of p_y and s orbitals are not visible in this example. The orbitals are centered on the corresponding atom. The shape, size, and color represent the orbital type and the magnitude of its contribution. The different colors of the orbital lobes (for example, lighter and darker blue for p_x) indicate the phase of the orbital. In this way, complete information about electron and hole states can be obtained. For example, state symmetries can be discerned immediately from these visualizations. Such symmetries are more difficult to discern otherwise.

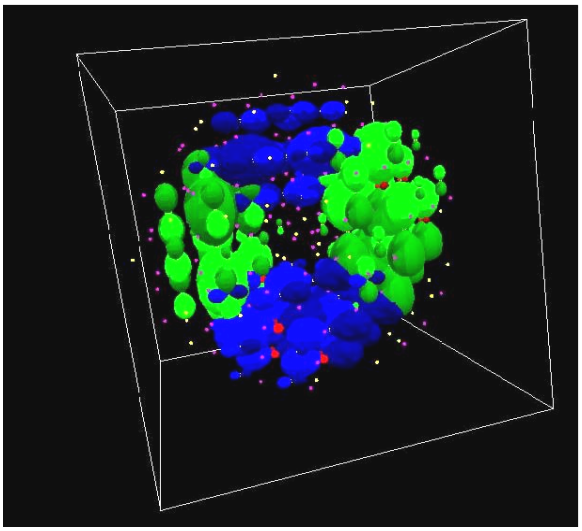


Figure 3. *Charge density of the lowest hole state in a CdS nanocrystal.*

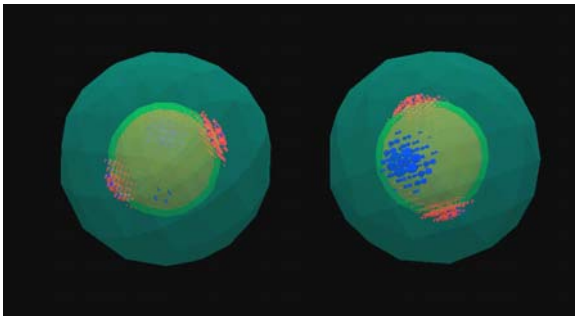


Figure 4. *Two different views of atomic state density of an electronic state trapped in the well region of a nanoheterostructured nanocrystal.*

Even for these examples, the amount of data to be visualized can be prohibitive. Coarser grained visualizations can avoid that problem. Fig. 4 shows contours and transparent surfaces which illustrate charge densities in a coarser grained way. The figure shows the atomic scale charge density of an electronic state trapped in the well region of a CdS/HgS/CdS core/well/clad nanoheterostructured nanocrystal.

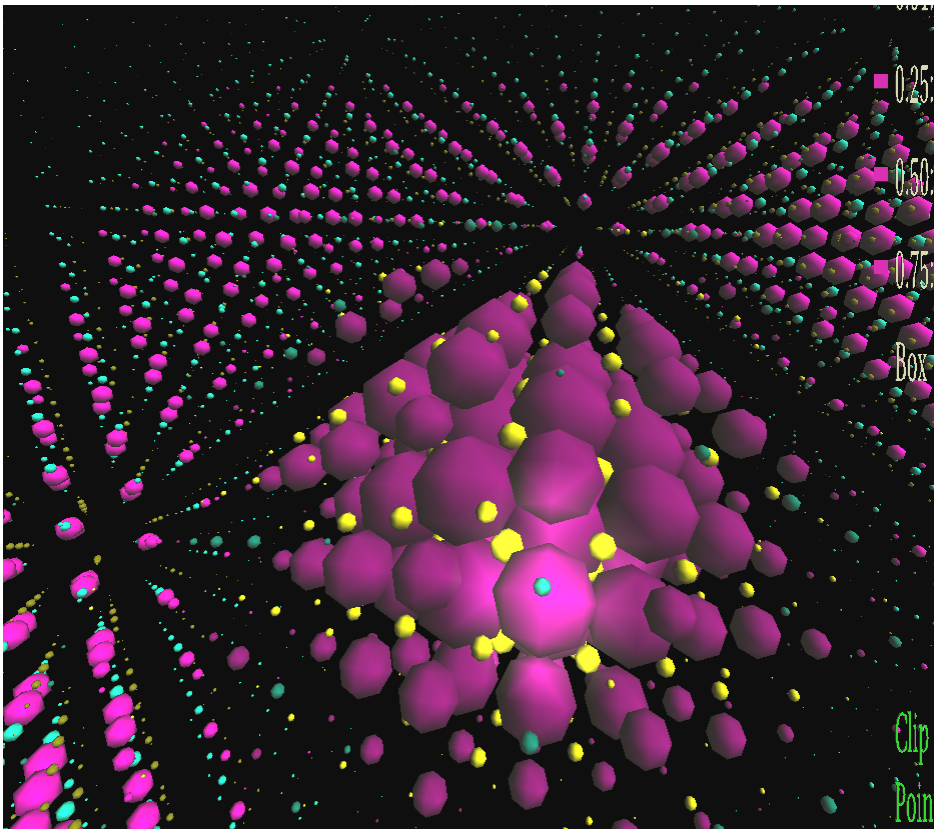


Figure 5. *Snapshot from an immersive visualization of a quantum dot. The spheres represent s orbitals, which also are representative of the atoms in the structure.*

We can do much more with the output of our nanostructure calculations. Our visual analysis capabilities include an immersive environment that allows scientists to interact with data by navigating through a three-dimensional virtual landscape of the data rather than by simply viewing pictures of the data. Our nanostructure calculations output detailed charge distributions which are transferred to the NIST immersive environment where they can be studied interactively. One can move through space, going inside the structure and moving around inside the structure. In this way one can visualize the structure looking in from the outside, or looking out from the inside. One can visualize both the nanostructure (see, for example, Fig. 5), and the atomic scale variation of calculated nanostructure properties from any orientation and position in space. This is not possible with any static graphical representation.

Such representations are tremendously helpful. They encapsulate the physics and allow one to easily see features that might be missed by just perusing the voluminous output from a supercomputer size calculation. Such insights are very helpful and greatly speedup the extraction of useful understanding and insights as we explore the properties of new and unfamiliar systems.

References

[1] G. W. Bryant and W. Jaskolski, Electronic Properties of Quantum-dot Molecules, *Physica E* **17** (2003), pp. 40-41.

[2] J. S. Sims, W. L. George, T. J. Griffin, J. G. Hagedorn, H. K. Hung, J. T. Kelso, M. Olano, A. P. Peskin, S. G. Satterfield, J. D. Terrill, G. W. Bryant, and J. G. Diaz, Accelerating Scientific Discovery through Computation and Visualization III. Tight-Binding Wave Functions for Quantum Dots, 2007, submitted.

[3] G. W. Bryant, W. Zielinski, W. Jaskolski, and J. Aizpurua, Controlling the Optical Properties of Self-assembled Quantum Dots using External Strain, *Proceedings of the Materials Research Society* **1055E**, 2007.

Participants

James S. Sims, William L. George, Terrence J. Griffin, John G. Hagedorn, John T. Kelso, Marc Olano, Adele P. Peskin, Steven G. Satterfield, Judith Devaney Terrill (MCSD); Garnett Bryant (NIST PL)

http://math.nist.gov/mcsd/savg/vis/nano/

Error Correction for Electromagnetic Motion Tracking Devices

MCSD researchers have developed algorithms for correcting both location and orientation errors in motion tracking devices. Such devices are used in a variety of fields such as immersive visualization, military, and medical applications. Positional errors in the data produced by these tracking devices compromise their utility. We present techniques for measuring and correcting these errors. In particular we use a method for interpolating orientation corrections that has not previously been used in this context. This method, unlike previous ones, is rooted in the geometry of the space of rotations. It is used in conjunction with a Delaunay tetrahedralization to enable both location and rotation correction based on scattered data samples. Our results show large improvements in both types of errors while imposing minimal computational burden.

John Hagedorn

Motion tracking devices are used in a wide variety of applications. At NIST we use an electromagnetic motion tracking system as a critical part of our immersive visualization environment (Fig. 1). Motion tracking is essential because the location and orientation of the user's eyes must be tracked continuously so that the system can render images of a spatially stable virtual world, in stereo, and in real-time. In addition, motion tracking is also used on hand-held devices that operate as tools in the virtual world.

Fig. 1 illustrates several important components of the environment: three screens that provide the visual display, an electromagnetic motion tracker transmitter, and sensors. The screens are used to display a single three-dimensional scene. Motion trackers are very commonly used in such systems. Unfortunately, errors in motion tracking have substantial deleterious effects, such as: virtual objects move inappropriately as the user moves; straight lines appear bent when they cross screen boundaries; and virtual objects tied to the tracked hand-held device appear incorrectly positioned.

Fig. 2 illustrates one of these effects. Here, the grid lines should all be straight. To the user in the immersive environment, the lines appear bent at the points where they cross the boundaries between screens because the images are being drawn based on an incorrect tracked location for the eyes of the observer. This figure is based on actual errors observed in the motion tracking system at NIST. It is by no means the worst case that could have been provided. In informal observations made before initiating this project, we observed location errors in excess of 50 cm and orientation errors that appeared to be more than 15 degrees.

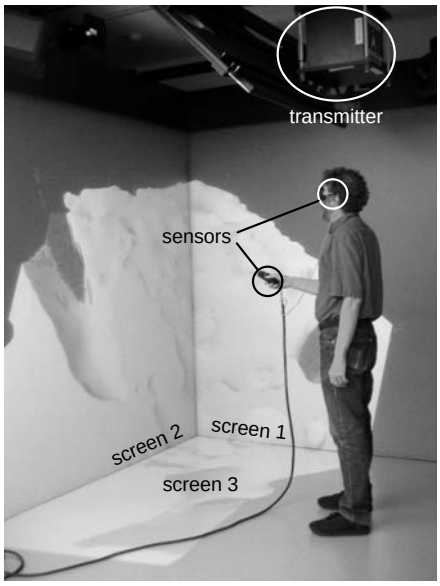


Figure 1. *The NIST immersive visualization system. A single 3D scene is being displayed across the three screens. Note the tracker transmitter, a tracker sensor on the user's stereo glasses, and another sensor on a hand-held pointing device.*

Our approach to correcting the tracker was to collect data at known locations and orientations and then to use these data to construct functions that represent the error [1]. This approach was simplified by verifying that errors were dependent on location, but not on orientation. This enabled us to construct correction functions based on measurements at a large number of locations using a fixed orientation at each location.

Our correction method is distinguished from prior work partly in our handling of the interpolation of orientation corrections. Our orientation interpolation method works entirely within the space of rotations and does not assume that the space of rotations is Euclidean. We use a quaternion representation of rotations, and we interpolate orientation corrections directly from measured orientation errors.

To correct the raw tracker data, we first record the tracker's measurements at a large set of known locations (and a fixed orientation) that encompass the volume that we need to accurately track. This enables us to calculate corrections at each of these points. We then perform a Delaunay tetrahedralization of the points based on the measured locations. Then, in real time, as the tracker reports each location and orientation, we find the tetrahedron that contains the measured location and generate barycentric coordinates for this location relative to the containing tetrahedron. The barycentric coordinates (which sum to 1 by construction) are used as weights for averaging the corrections at the vertices of the containing tetrahedron.

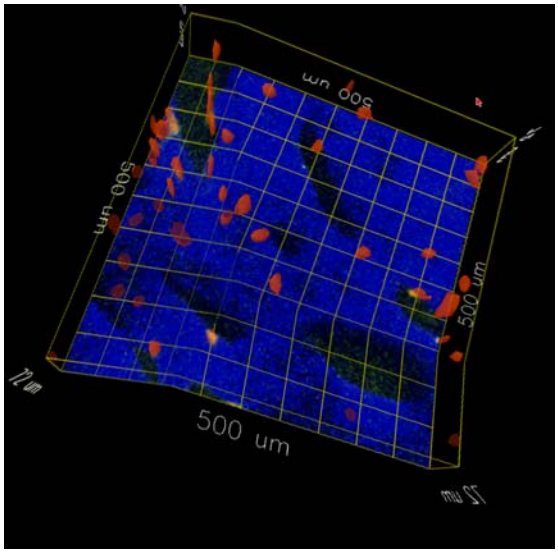


Figure 2. *Distortions due to tracker miscalibration. The grid lines should be straight; they bend at the points where they cross the boundaries between screens in the immersive environment.*

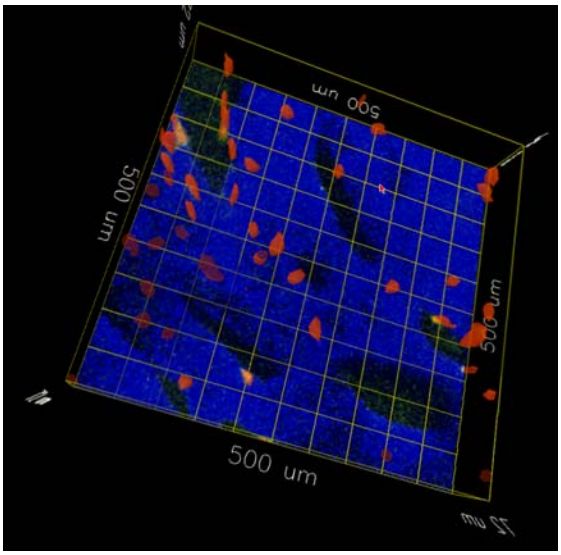


Figure 3. *The scene shown in Fig. 2 with tracker data corrected by the methods described here.*

For orientation averaging, we use these weights with a spherical weighted averaging technique [2] to average the correction rotations at each of the four vertices of the tetrahedron. This use of barycentric coordinates with spherical weighted averaging has a much clearer geometric rationale than previous methods.

In order to validate our methods, we collected raw tracker data at a second set of known locations and orientations; we refer to these as the *validation points*. We apply the correction algorithm to the validation points and compare the corrected data to the true locations and orientations. Table 1 reports the results. We see a 95.5% reduction in location errors and an 87.0% reduction in orientation errors. These improvements, as best we can determine, equal or substantially surpass previous motion tracker correction methods.

	Location Errors		Orientation Errors	
	Ave	**Std Dev**	**Ave**	**Std Dev**
Uncorrected	37.51	19.30	19.57	8.96
Corrected	1.69	1.30	2.55	0.63

Table 1. *Location errors (cm) and orientation errors (in degrees) for validation points before and after applying corrections.*

When we apply this correction to the tracker data in our immersive visualization environment, the difference is clear to the user. Objects that should be stable appear to be stationary as the user moves through the virtual scene, the displayed pointer tracks the tracked hand-held device accurately, and there are no visual artifacts when objects cross the divisions between screens. The real-time correction causes no significant performance degradation of the interactive system. Fig. 3 shows the scene depicted in Fig. 2 after correction of the viewer's tracked position with our method.

Bringing together the use of spherical weighted averages with Delaunay tetrahedralization and barycentric coordinates provides a valuable approach to correcting tracker data based on samples at scattered points. The correction methods described here are in daily use in our immersive visualization environment and have yielded substantial improvement to the immersive experience. In addition to the qualitative benefits of this correction, this effort is an important step toward the assessment of uncertainty of virtual measurements of models and experimental data that are made in the interactive immersive visualization environment [3].

References

[1] J. G. Hagedorn, S. G. Satterfield, J. T. Kelso, W. Austin, J. E. Terrill, and A. P. Peskin, Correction of location and orientation errors in electromagnetic motion tracking. *Presence: Teleoperators and Virtual Environments* **16** (2007), pp. 352-366.

[2] S. R. Buss and J. P. Fillmore, Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics* **20** (2001), pp. 95-126.

[3] J. G. Hagedorn, J. P. Dunkers, S. G. Satterfield, A. P. Peskin, J. T. Kelso, and J. E. Terrill, Measurement tools for the immersive visualization environment. *Journal of Research of NIST* **112** (5) (2007), pp. 257-270.

Participants

J. Hagedorn, S. Satterfield, J. Kelso, W. Austin[5], J. Terrill, A. Peskin (MCSD)

http://math.nist.gov/mcsd/savg/vis/tracker/

[5] Student, Jackson State University

Automated Combinatorial Testing for Software Systems

Most often, outages in modern business, industrial, medical, scientific, and transport systems are caused by unexpected faults in the underlying software, which is becoming increasingly complex. In complex software, failures are often the result of interaction bugs. An interaction bug is a fault that causes failure only when particular values of certain input parameters are specified in combination with particular values of other parameters. Interaction bugs may remain dormant until the unfortunate combination of parameter values is encountered in practice. Combinatorial testing is an approach to detect interaction bugs. Generally, combinatorial testing is used in functional or black-box testing where specifications of the functionality of the software system are given. It may also be useful in structural or white-box testing (i.e., when information about the software architecture and the internal structure is available). Much of the recent practice of combinatorial testing is focused on detecting interaction bugs involving two parameters. Many interaction bugs do involve two parameters only; however, a NIST investigation of actual faults suggests that up to six parameters do occur in practice. This NIST/ITL interdivisional project has produced several algorithms and a combinatorial testing tool that advance the technology of combinatorial testing for complex software systems from detecting interaction bugs involving two parameters to six or more parameters.

Raghu Kacker

Combinatorial testing is a relatively new approach, becoming popular beginning in the 1990s, for testing software systems. The initial focus was on pair-wise (2-way) testing in which all interaction bugs involving two parameters are sought. A NIST study [1] of actual faults in medical devices, browsers, servers, and a NASA application indicated that about 95 % of interaction bugs involved four or fewer parameters and no faults involved more than six parameters (Table 1). Thus, while interaction bugs typically involve a relatively small number of parameters, 2-way testing may not be adequate. On the other hand, a t-way testing strategy, for t from 2 to 6, may be adequate for many practical applications. The goal of this project is to advance the technology from pair-wise to multi-way testing and to demonstrate successful application to one or more applications.

The first step in combinatorial testing is to map the problem of detecting interaction bugs to a combinatorial framework. The parameters to be tested are defined. A finite subset of the (possibly infinite number) of input values is selected as test values. The number of test values for a parameter is its domain size. A test run can then be represented as a row consisting of one test value for each parameter. If there are k parameters with domain sizes v_1, v_2, …, v_k, respectively, then the test space consists of $v_1 \times v_2 \times \ldots \times v_k$ test runs (rows). Some knowledge of the functionality or of the structure of the software system is needed to specify the parameters and their test values such that the test space includes potential interaction bugs. Even for modest software the number $v_1 \times v_2 \times \ldots \times v_k$ (size of test space) could be very large to exercise all test runs in the test space. A combinatorial test suite is a subset of test runs selected from the test space such that all t-way combinations between the k parameters are exercised, for a chosen value of t.

	Cumulative percent of actual faults			
Parameters	M	B	S	N
1	66	29	42	68
1 - 2	97	76	70	93
1 - 3	99	95	89	98
1 - 4	100	97	96	100
1 - 5	100	99	96	100
1 - 6	100	100	100	100

Table 1. *Cumulative distribution (expressed as percent) of the number of parameters involved in actual faults which occurred in software in medical devices (M), browsers (B), servers (S), and a NASA application (N)*

Covering arrays. A covering array of strength t is a subset of N rows out of $v_1 \times v_2 \times \ldots \times v_k$ rows (test runs) that includes, at least once, each possible t-way combination of the test values of every possible subset of t parameters out of the k parameters. If t is less than k then the covering array size N is a fraction of the size of the test space. Table 2 shows a covering array of strength $t = 3$ for $k = 10$ where each parameter has $v = 2$ test values (indicated by 0 and 1). The size of the test space is 2^{10}, i.e., 1024 test runs would be necessary to exhaustively run all combinations of input parameters. Yet $N = 13$ test runs include all 3-way combinations. (Note that there are 960 3-way combinations for ten 2-valued parameters). Thus, a test suite of 13 test runs based on this covering array will exercise all 3-way combinations among the values of $k = 10$ parameter each having $v = 2$ test values.

Construction of covering arrays of strength t (for $t = 2$, … , 6 or more) with N as small as possible for practical number of parameters k and domain sizes v_1, v_2, …, v_k is a challenging problem. The state-of-the art in combinatorial mathematics deals with the special case in which each parameter takes on the same number of

values, i.e., $v_1 = v_2 = \ldots, = v_k = v$, and even then yields covering arrays of minimum possible size N only for very few special values of k, v, and t. In devising computational methods one must determine the entry in each cell such that the combinatorial property of the covering array is satisfied, but doing so with the smallest possible value of N is a difficult combinatorial optimization problem.

Collaboration between NIST/ITL and the University of Texas at Arlington has led to a tool, called FireEye, to generate covering arrays. FireEye [2] includes several algorithms. The basic algorithm, called IPOG, is described in [3]. The principle performance criteria for constructions of covering arrays are that the covering array size N should be as small as possible and the time taken for construction should also be as small as possible. On both criteria, FireEye using IPOG performs better than other publicly available tools. We are continually improving the algorithms. A faster version of IPOG, called IPOG-D, is described in reference [4]. Another algorithm, called Paintball, which can use a cluster of processors, is under development. A refinement of IPOG, called IPOG-F, is also under development. Using IPOG-F, a catalog of covering arrays of smaller covering array size N than indicated in the literature is being developed [5].

Combinatorial testing based on covering arrays generated using IPOG has also been shown to be useful for reachability testing in concurrent (parallel processing) programs [6]. We are investigating approaches to extend known covering arrays is various directions such as changes in domain sizes, increase in number of parameters, and increase in strength.

	Parameters									
Rows	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	0	1	0	0	0	0	1
4	1	0	1	1	0	1	0	1	0	0
5	1	0	0	0	1	1	1	0	0	0
6	0	1	1	0	0	1	0	0	1	0
7	0	0	1	0	1	0	1	1	1	0
8	1	1	0	1	0	0	1	0	1	0
9	0	0	0	1	1	1	0	0	1	1
10	0	0	1	1	0	0	1	0	0	1
11	0	1	0	1	1	0	0	1	0	0
12	1	0	0	0	0	0	0	1	1	1
13	0	1	0	0	0	1	1	1	0	1

Table 2. *A covering array of 13 rows includes all eight triplets (000, 001, 010, 011, 100, 101, 110, and 111) between two possible values (0 and 1) for every three of the 10 parameters represented by columns (for example, see colored entries)*

Test oracles. A covering array of strength t yields a combinatorial test suite for t-way testing. The next task is to determine the expected correct output of each test run against which the actual output may be compared. A test suite along with the correct output for each test run is the test oracle. We are investigating use of the model-checking tools, such as NuSMV (initially developed by Carnegie Mellon University and then improved by others), to create a test oracle. Determination of the expected correct outputs for test runs requires either complete specification of the functionality, a reference implementation of the software, or some model for the software.

We are investigating application of combinatorial testing for assurance of access control systems and for firewall policy testing. In these applications, functional specifications are available to construct test oracles. Other potential applications include communication protocol and process control systems. In these, specifications using finite state automata are common and could be used in constructing test oracles. We are also investigating the integration of our combinatorial test methods with modeling tools such as SIMULINK.

References

[1] D. R. Kuhn, D. R. Wallace and A. Gallo, Software Fault Interactions and Implications for Software Testing, *IEEE Transactions on Software Engineering* **30** (2004) pp. 418-421.

[2] Y. Lei et al., User Manual for FireEye, in process.

[3] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence, IPOG: A General Strategy for *t*-way Software Testing, in *Proceedings of the IEEE International Conference on Engineering of Computer-Based Systems*, Tucson AZ, USA, 2007, pp. 549-556.

[4] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence, IPOG/IPOG-D: Efficient Test Generation for Multi-way Combinatorial Testing, *Software Testing, Verification and Reliability*, to appear.

[5] M. Forbes, J. Lawrence, R. Kacker, Y. Lei, and D. R. Kuhn, A catalog of covering arrays, in process.

[6] Y. Lei, R. H. Carver, R. Kacker, and D. Kung, A Combinatorial Testing Strategy for Concurrent Programs, *Software Testing, Verification and Reliability* **17** (2007), pp. 207-225.

Participants

Raghu N. Kacker, Jeff Yu Lei, James F. Lawrence, Michael Forbes (MCSD); D. Richard Kuhn, Vincent Hu, Richard M. Rivello (NIST ITL); Tao Xie (NCSU); Renee Bryce (UNLV); Sreedevi Sampath (UMBC), Sagar Chaki, Arie Gurfinkel (CMU)

http://csrc.nist.gov/acts/

Math Search

As digital libraries of mathematical and scientific contents become widely available, it is essential to have search systems that empower users to search not only for text but also for equations and other mathematical constructs. To do this, the system must be math-aware, that is, it must recognize math symbols and structures in queries and contents, and it must present search results in a way that facilitates finding the desired information rapidly. We have developed a math-aware search system for the NIST Digital Library of Mathematical Functions (DLMF). This search system is general enough to be easily adapted to other math digital libraries.

Abdou Youssef

When it is released in 2008, the NIST Digital Library of Mathematical Functions (DLMF) will be a freely available, interactive, online resource providing reference information on the special functions of applied mathematics. As such, it is designed to be a modern successor to the *Handbook of Mathematical Functions* published by NBS in 1964 [1]. This classic handbook has become an indispensable resource for those doing research in the physical sciences and engineering. With nearly a million copies in print, the *Handbook* has become the most cited work in the mathematical literature. The DLMF is expected to become the new authoritative reference work on this subject. Updated to reflect developments in applied mathematics and computer science in the ensuing 40+ years, the DLMF will contain more than twice as much the technical material as the original *Handbook*.

As a mathematical reference work built on the same general model as the *Handbook*, the DLMF will contain some 36 chapters with mathematical definitions, identities, relations, information on computational methods and software, graphs, tables, and extensive references, but very little traditional text. To provide online support for users to locate the equations and expressions they need, the DLMF search system must be able to recognize mathematical symbols and structures, a requirement that is fundamentally different from Google-like text search. Field-based search systems have been built for math content providers, such as Zentralblatt [5] and MathSCiNet [6], but they are based on text search and are unable to recognize mathematical notation. What is needed in all these cases is a math-aware, fine-grained, general-purpose search system. We have pioneered such a system for the DLMF [2, 3], which stands as the most advanced math search system to date. This tool is intended for students, educators, researchers, and professionals in mathematics, physical sciences, and engineering. With minor adaptations, it is deployable by math/science publishers, professional societies, and other public and private math/science contents providers.

Features of Math Search. The following are essential components and features that we believe math search systems must have.

- *Math awareness*. This is the ability to recognize mathematical notation and structures, both in user queries and the digital library contents being searched. For example, the symbols "–" and "/" in "$x–y/z$" are recognized as the subtraction and division operators (text search ignores such symbols), and the "2" in "x^2" is recognized as the power of x. (See Fig. 1.)
- *Query language and processing.* Users should be able to express their queries easily and intuitively, and the system must recognize the user's intent, and process the query to satisfy that intent. In particular, the user's query can be math fragments, where much information is implicit or expressed in wildcards. DLMF search analyzes the query and transforms it to a more explicit form that can be searched for effectively.
- *Ranking of search results.* To locate the needed information among hits, hits must be rank-ordered by relevance not only to the query but also to the general needs of math users. Math-specific factors unbeknown to text search ranking such as definitions, theorems, proofs, and names of certain mathematical concepts, carry more weight than other types of contents. Ranking of hits by the DLMF search engine takes such factors into account. (See Fig. 2.)
- *Hit description.* Each hit must be accompanied with a descriptive, query-relevant summary of the contents. The summary can be a combination of equations, graphs, portions of tables, or text. The DLMF search provides this important feature.

DLMF search has other features. These include specifying types of hits and highlighting math keywords (see Fig. 3). Also, we have enriched the math contents with metadata to enhance the ability of search systems to find relevant matches.

In the future, math search will be used in other applications and extended in different directions [4]. For example, it can be used for finding similarity of patterns and thus for scientific data mining (knowledge discovery). It can be synergistically combined with

instructional systems, proof assistants, and computational systems. Federated search against multiple, geographically distributed math digital libraries is another important direction worth pursuing. We fully expect that as math search becomes more widely used, other unforeseen opportunities will emerge that warrant further research.

References

[1] M. Abramowitz and I. Stegun (eds), Handbook of Mathematical Functions, AMS 55, National Bureau of Standards, Washington, DC, 1964.

[2] B. Miller and A. Youssef, Technical Aspects of the Digital Library of Mathematical Functions, *Annals of Mathematics and Artificial Intelligence* **38** (2003), pp. 121-136.

[3] A. Youssef, Methods of Relevance Ranking and Hit-content Generation in Math Search, in *Proceedings of the 6th Mathematical Knowledge Management Conference*, Austria, June 27-30, 2007, pp. 393-406.

[4] A. Youssef, Roles of Math Search in Mathematics, in *Proceedings of the 5th International Conference on Mathematical Knowledge Management*, UK, August 11-12, 2006, pp. 2-16.

[5] Zentralblatt MATH, Fachinformationzentrum Karlsruhe, http://www.zentralblatt-math.org/zmath/en/search/.

[6] MathSciNet, American Mathematical Society. http://www.ams.org/mathscinet/.

Participants

Abdou Youssef, Bruce Miller, Daniel Lozier, Ronald Boisvert

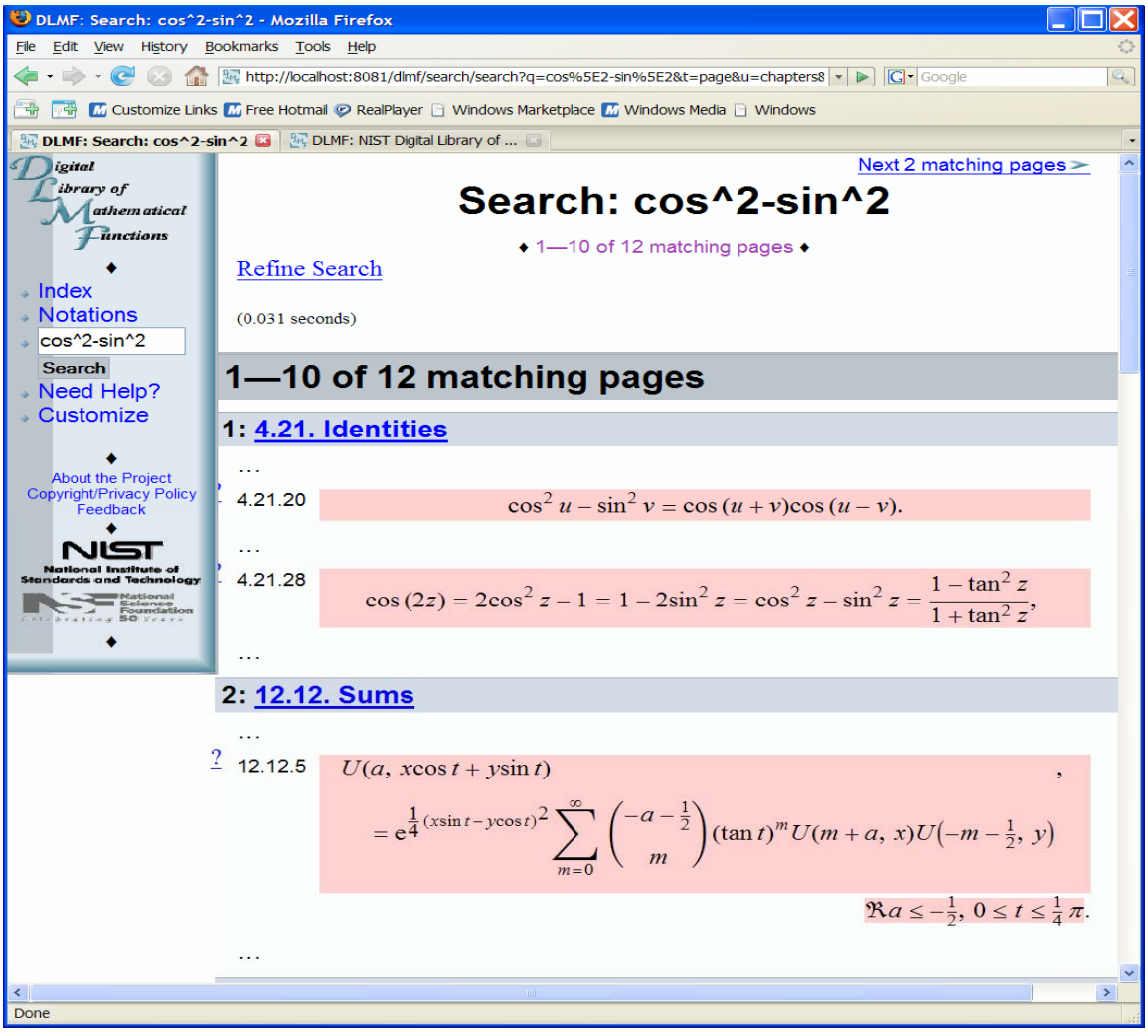


Figure 1. *Search results of query* ***cos^2-sin^2****, showing math-awareness & intent-understanding.*

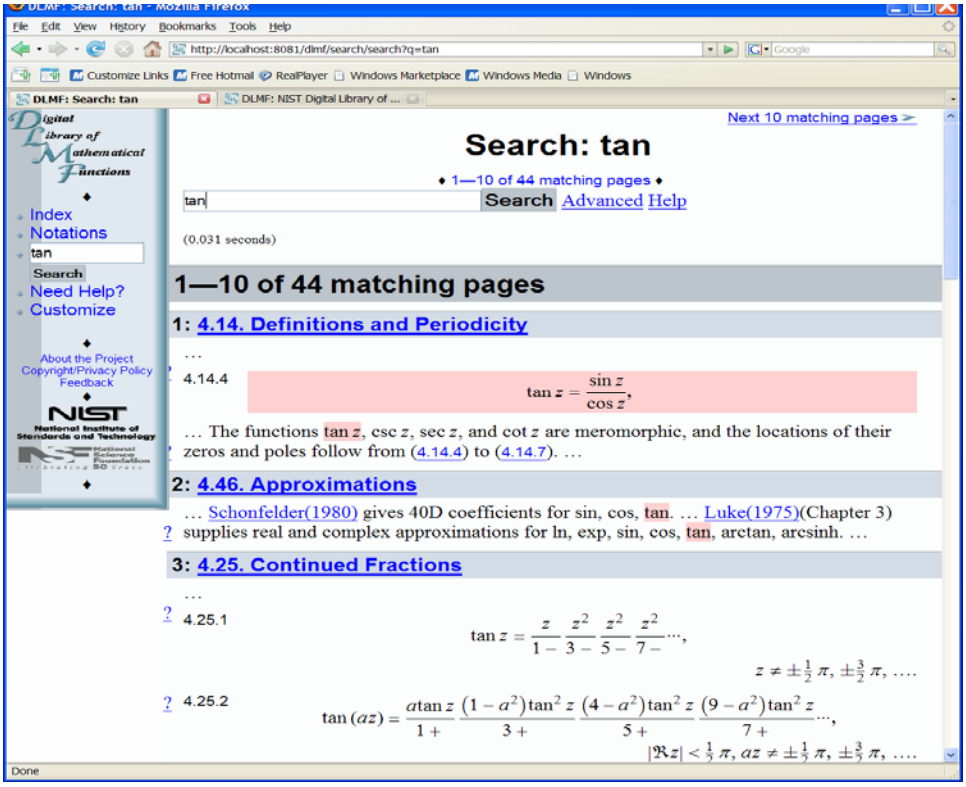


Figure 2. *Search results of query "tan" (for the tangent function), showing good ranking and informative hit-description.*

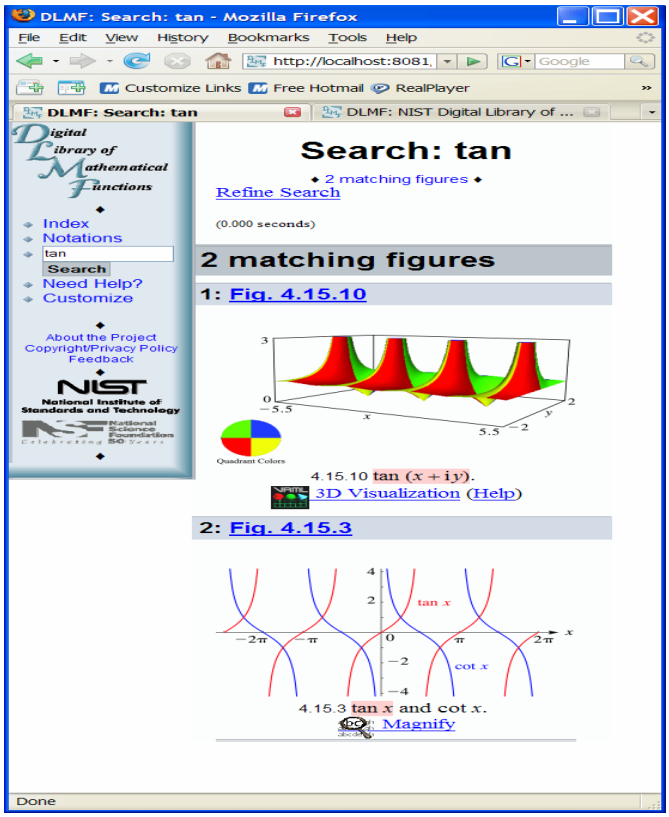


Figure 3. *Limiting the search to graphs. It also illustrates highlighting of the search keyword "tan".*