A Science-Based Case for Large-Scale Simulation

Volume 1

Office of Science
U.S. Department of Energy

July 30, 2003

A SCIENCE-BASED CASE FOR LARGE-SCALE SIMULATION

Office of Science
U.S. Department of Energy

July 30, 2003

“There will be opened a gateway and a road to a large and excellent science, into which minds more piercing than mine shall penetrate to recesses still deeper.” Galileo (1564–1642)

[*on the “experimental mathematical analysis of nature,” appropriated here for “computational simulation”*]

TRANSMITTAL

July 30, 2003

Dr. Raymond L. Orbach, Director
Office of Science
U.S. Department of Energy
Washington, DC

Dear Dr. Orbach,

On behalf of more than 300 contributing computational scientists from dozens of leading universities, all of the multiprogrammatic laboratories of the Department of Energy, other federal agencies, and industry, I am pleased to deliver Volume 1 of a two-volume report that builds a Science-Based Case for Large-Scale Simulation.

We find herein (and detail further in the companion volume) that favorable trends in computational power and networking, scientific software engineering and infrastructure, and modeling and algorithms are allowing several applications to approach thresholds beyond which lies new knowledge of both fundamental and practical kinds. A major increase in investment in computational modeling and simulation is appropriate at this time, so that our citizens are the first to benefit from particular new fruits of *scientific simulation*, and indeed, from an evolving culture of *simulation science.*

Based on the encouraging first two years of the Scientific Discovery through Advanced Computing initiative, we believe that balanced investment in scientific applications, applied mathematics, and computer science, with cross-accountabilities between these constituents, is a program structure worthy of extension. Through it, techniques from applied mathematics and computer science are systematically being migrated into scientific computer codes. As this happens, new challenges in the use of these techniques flow back to the mathematics and computer science communities, rejuvenating their own basic research. Now is an especially opportune time to increase the number of groups working in this multidisciplinary way, and to provide the computational platforms and environments that will enable these teams to push their simulations to the next insight-yielding levels of high resolution and large ensembles.

The Department of Energy can draw satisfaction today from its history of pathfinding in each of the areas that are brought together in large-scale simulation. However, from the vantage point of tomorrow, the Department’s hand in their systematic fusion will be regarded as even more profound.

Best regards,

David E. Keyes
Fu Foundation Professor of Applied Mathematics
Columbia University
New York, NY

EXECUTIVE SUMMARY

Important advances in basic science crucial to the national well-being have been brought near by a “perfect fusion” of sustained advances in scientific models, mathematical algorithms, computer architecture, and scientific software engineering. Computational simulation—a means of scientific discovery that employs a computer system to simulate a physical system according to laws derived from theory and experiment—has attained peer status with theory and experiment in many areas of science. The United States is currently a world leader in computational simulation, a position that confers both an opportunity and a responsibility to mount a vigorous campaign of research that brings the advancing power of simulation to many scientific frontiers.

Computational simulation offers to enhance, as well as leapfrog, theoretical and experimental progress in many areas of science critical to the scientific mission of the U.S. Department of Energy (DOE). Successes have been documented in such areas as advanced energy systems (e.g., fuel cells, fusion), biotechnology (e.g., genomics, cellular dynamics), nanotechnology (e.g., sensors, storage devices), and environmental modeling (e.g., climate prediction, pollution remediation). Computational simulation also offers the best near-term hope for progress in answering a number of scientific questions in such areas as the fundamental structure of matter, the production of heavy elements in supernovae, and the functions of enzymes.

The ingredients required for success in advancing scientific discovery are insights, models, and applications from scientists; theory, methods, and algorithms from mathematicians; and software and hardware infrastructure from computer scientists. Only major new investment in these activities across the board, in the program areas of DOE’s Office of Science and other agencies, will enable the United States to be the first to realize the promise of the scientific advances to be wrought by computational simulation.

In this two-volume report, prepared with direct input from more than 300 of the nation’s leading computational scientists, a science-based case is presented for major, new, carefully balanced investments in

- scientific applications
- algorithm research and development
- computing system software infrastructure
- network infrastructure for access and resource sharing
- computational facilities
- innovative computer architecture research, for the facilities of the future
- proactive recruitment and training of a new generation of multidisciplinary computational scientists

The two-year-old Scientific Discovery through Advanced Computing (SciDAC) initiative in the Office of Science provides a template for such science-directed, multidisciplinary research campaigns. SciDAC’s successes in the first four of these seven thrusts have illustrated the advances possible with coordinated investments. It is now time to take full advantage of the revolution in computational science with new investments that address the most challenging scientific problems faced by DOE.

A SCIENCE-BASED CASE FOR LARGE-SCALE SIMULATION

VOLUME 1

ATTRIBUTIONS

Report Editors

Phillip Colella, Lawrence Berkeley National Laboratory
Thom H. Dunning, Jr., University of Tennessee and Oak Ridge National Laboratory
William D. Gropp, Argonne National Laboratory
David E. Keyes, Columbia University, *Editor-in-Chief*

Workshop Organizers

David L. Brown, Lawrence Livermore National Laboratory
Phillip Colella, Lawrence Berkeley National Laboratory
Lori Freitag Diachin, Sandia National Laboratories
James Glimm, State University of New York–Stony Brook and
Brookhaven National Laboratory
William D. Gropp, Argonne National Laboratory
Steven W. Hammond, National Renewable Energy Laboratory
Robert Harrison, Oak Ridge National Laboratory
Stephen C. Jardin, Princeton Plasma Physics Laboratory
David E. Keyes, Columbia University, *Chair*
Paul C. Messina, Argonne National Laboratory
Juan C. Meza, Lawrence Berkeley National Laboratory
Anthony Mezzacappa, Oak Ridge National Laboratory
Robert Rosner, University of Chicago and Argonne National Laboratory
R. Roy Whitney, Thomas Jefferson National Accelerator Facility
Theresa L. Windus, Pacific Northwest National Laboratory

Topical Group Leaders

Science Applications

Accelerators
Kwok Ko, Stanford Linear Accelerator Center
Robert D. Ryne, Lawrence Berkeley National Laboratory

Astrophysics
Anthony Mezzacappa, Oak Ridge National Laboratory
Robert Rosner, University of Chicago and Argonne National Laboratory

Biology
Michael Colvin, Lawrence Livermore National Laboratory
George Michaels, Pacific Northwest National Laboratory

Chemistry
Robert Harrison, Oak Ridge National Laboratory
Theresa L. Windus, Pacific Northwest National Laboratory

Climate and Earth Science
John B. Drake, Oak Ridge National Laboratory
Phillip W. Jones, Los Alamos National Laboratory
Robert C. Malone, Los Alamos National Laboratory

Combustion
John B. Bell, Lawrence Berkeley National Laboratory
Larry A. Rahn, Sandia National Laboratories

Environmental Remediation and Processes
Mary F. Wheeler, University of Texas–Austin
Steve Yabusaki, Pacific Northwest National Laboratory

Materials Science
Francois Gygi, Lawrence Livermore National Laboratory
G. Malcolm Stocks, Oak Ridge National Laboratory

Nanoscience
Peter T. Cummings, Vanderbilt University and Oak Ridge National Laboratory
Lin-wang Wang, Lawrence Berkeley National Laboratory

Plasma Science
Stephen C. Jardin, Princeton Plasma Physics Laboratory
William M. Nevins, Lawrence Livermore National Laboratory

Quantum Chromodynamics
Robert Sugar, University of California–Santa Barbara

Mathematical Methods and Algorithms

Computational Fluid Dynamics
Philip Colella, Lawrence Berkeley National Laboratory
Paul F. Fischer, Argonne National Laboratory

Discrete Mathematics and Algorithms
Bruce A. Hendrickson, Sandia National Laboratories
Alex Pothen, Old Dominion University

Meshing Methods
Lori Freitag Diachin, Sandia National Laboratories
David B. Serafini, Lawrence Berkeley National Laboratory
David L. Brown, Lawrence Livermore National Laboratory

Multi-physics Solution Techniques
Dana A. Knoll, Los Alamos National Laboratory
John N. Shadid, Sandia National Laboratories

Multiscale Techniques
Thomas J. R. Hughes, University of Texas–Austin
Mark S. Shephard, Rensselaer Polytechnic Institute

Solvers and “Fast” Algorithms
Van E. Henson, Lawrence Livermore National Laboratory
Juan C. Meza, Lawrence Berkeley National Laboratory

Transport Methods
Frank R. Graziani, Lawrence Livermore National Laboratory
Gordon L. Olson, Los Alamos National Laboratory

Uncertainty Quantification
James Glimm, State University of New York–Stony Brook and
Brookhaven National Laboratory
Sallie Keller-McNulty, Los Alamos National Laboratory

Computer Science and Infrastructure

Access and Resource Sharing
Ian Foster, Argonne National Laboratory
William E. Johnston, Lawrence Berkeley National Laboratory

Architecture
William D. Gropp, Argonne National Laboratory

Data Management and Analysis
Arie Shoshani, Lawrence Berkeley National Laboratory
Doron Rotem, Lawrence Berkeley National Laboratory

Frameworks and Environments
Robert Armstrong, Sandia National Laboratories
Kathy Yelick, University of California–Berkeley

Performance Tools and Evaluation
David H. Bailey, Lawrence Berkeley National Laboratory

Software Engineering and Management
Steven W. Hammond, National Renewal Energy Laboratory
Ewing Lusk, Argonne National Laboratory

System Software
Al Geist, Oak Ridge National Laboratory

Visualization
E. Wes Bethel, Lawrence Berkeley National Laboratory
Charles Hanson, University of Utah

WORKSHOP PARTICIPANTS

Tom Abel
Pennsylvania State University

Marvin Lee Adams
Texas A&M University

Srinivas Aluru
Iowa State University

James F. Amundson
Fermi National Accelerator Laboratory

Carl W. Anderson
Brookhaven National Laboratory

Kurt Scott Anderson
Rensselaer Polytechnic Institute

Adam P. Arkin
University of California–Berkeley / Lawrence Berkeley National Laboratory

Rob Armstrong
Sandia National Laboratories

Miguel Arqaez
University of Texas–El Paso

Steven F. Ashby
Lawrence Livermore National Laboratory

Paul R. Avery
University of Florida

Dave Bader
DOE Office of Science

David H. Bailey
Lawrence Berkeley National Laboratory

Ray Bair
Pacific Northwest National Laboratory

Kim K. Baldridge
San Diego Supercomputer Center

Samuel Joseph Barish
DOE Office of Science

Paul Bayer
DOE Office of Science

Grant Bazan
Lawrence Livermore National Laboratory

Brian Behlendorf
CollabNet

John B. Bell
Lawrence Berkeley National Laboratory

Michael A. Bender
State University of New York–Stony Brook

Jerry Bernholc
North Carolina State University

David E. Bernholdt
Oak Ridge National Laboratory

Wes Bethel
Lawrence Berkeley National Laboratory

Tom Bettge
National Center for Atmospheric Research

Amitava Bhattacharjee
University of Iowa

Eugene W. Bierly
American Geophysical Union

John Blondin
North Carolina State University

Randall Bramley
Indiana University

Marcia Branstetter
Oak Ridge National Laboratory

Ron Brightwell
Sandia National Laboratories

Mark Boslough
Sandia National Laboratories

Richard Comfort Brower
Boston University

David L. Brown
Lawrence Livermore National Laboratory

Steven Bryant
University of Texas–Austin

Darius Buntinas
Ohio State University

Randall D. Burris
Oak Ridge National Laboratory

Lawrence Buja
National Center for Atmospheric Research

Phillip Cameron-Smith
Lawrence Livermore National Laboratory

Andrew Canning
Lawrence Berkeley National Laboratory

Christian Y. Cardall
Oak Ridge National Laboratory

Mark Carpenter
NASA Langley Research Center

John R. Cary
University of Colorado–Boulder

Fausto Cattaneo
University of Chicago

Joan Centrella
NASA Goddard Space Flight Center

Kyle Khem Chand
Lawrence Livermore National Laboratory

Jacqueline H. Chen
Sandia National Laboratories

Shiyi Chen
Johns Hopkins University

George Liang-Tai Chiu
IBM T. J. Watson Research Center

K.-J. Cho
Stanford University

Alok Choudhary
Northwestern University

Norman H. Christ
Columbia University

Todd S. Coffey
Sandia National Laboratories

Phillip Colella
Lawrence Berkeley National Laboratory

Michael Colvin
Lawrence Livermore National Laboratory

Sidney A. Coon
DOE Office of Science

Tony Craig
National Center for Atmospheric Research

Michael Creutz
Brookhaven National Laboratory

Robert Keith Crockett
University of California–Berkeley

Peter Thomas Cummings
Vanderbilt University

Dibyendu K. Datta
Rensselaer Polytechnic University

James W. Davenport
Brookhaven National Laboratory

Eduardo F. D'Azevedo
Oak Ridge National Laboratory

Cecelia Deluca
National Center for Atmospheric Research

Joel Eugene Dendy
Los Alamos National Laboratory

Narayan Desai
Argonne National Laboratory

Carleton DeTar
University of Utah

Chris Ding
Lawrence Berkeley National Laboratory

David Adams Dixon
Pacific Northwest National Laboratory

John Drake
Oak Ridge National Laboratory

Phil Dude
Oak Ridge National Laboratory

Jason Duell
Lawrence Berkeley National Laboratory

Phil Duffy
Lawrence Livermore National Laboratory

Thom H. Dunning, Jr.
University of Tennessee / Oak Ridge National Laboratory

Stephen Alan Eckstrand
DOE Office of Science

Robert Glenn Edwards
Thomas Jefferson National Accelerator Facility

Howard C. Elman
University of Maryland

David Erickson
Oak Ridge National Laboratory

George Fann
Oak Ridge National Laboratory

Andrew R. Felmy
Pacific Northwest National Laboratory

Thomas A. Ferryman
Pacific Northwest National Laboratory

Lee S. Finn
Pennsylvania State University

Paul F. Fischer
Argonne National Laboratory

Jacob Fish
Rensselaer Polytechnic University

Ian Foster
Argonne National Laboratory

Leopoldo P. Franca
University of Colorado–Denver

Randall Frank
Lawrence Livermore National Laboratory

Lori A. Freitag Diachin
Sandia National Laboratories

Alex Friedman
Lawrence Livermore and Lawrence Berkeley National Laboratories

Teresa Fryberger
DOE Office of Science

Chris Fryer
Los Alamos National Laboratory

Sam Fulcomer
Brown University

Giulia Galli
Lawrence Livermore National Laboratory

Dennis Gannon
Indiana University

Angel E. Garcia
Los Alamos National Laboratory

Al Geist
Oak Ridge National Laboratory

Robert Steven Germain
IBM T. J. Watson Research Center

Steve Ghan
Pacific Northwest National Laboratory

Roger G. Ghanem
Johns Hopkins University

Omar Ghattas
Carnegie Mellon University

Ahmed F. Ghoniem
Massachusetts Institute of Technology

John R. Gilbert
University of California–Santa Barbara

Roscoe C. Giles
Boston University

James Glimm
Brookhaven National Laboratory

Sharon C. Glotzer
University of Michigan

Tom Goodale
Max Planck Institut für Gravitationsphysik

Brent Gorda
Lawrence Berkeley National Laboratory

Mark S. Gordon
Iowa State University

Steven Arthur Gottlieb
Indiana University

Debbie K. Gracio
Pacific Northwest National Laboratory

Frank Reno Graziani
Lawrence Livermore National Laboratory

Joseph F. Grcar
Lawrence Berkeley National Laboratory

William D. Gropp
Argonne National Laboratory

Francois Gygi
Lawrence Livermore National Laboratory

James Hack
National Center for Atmospheric Research

Steve Hammond
National Renewable Energy Laboratory

Chuck Hansen
University of Utah

Bruce Harmon
Ames Laboratory

Robert J. Harrison
Oak Ridge National Laboratory

Teresa Head-Gordon
University of California–Berkeley / Lawrence Berkeley National Laboratory

Martin Head-Gordon
Lawrence Berkeley National Laboratory

Helen He
Lawrence Berkeley National Laboratory

Tom Henderson
National Center for Atmospheric Research

Van Emden Henson
Lawrence Livermore National Laboratory

Richard L. Hilderbrandt
National Science Foundation

Rich Hirsh
National Science Foundation

Daniel Hitchcock
DOE Office of Science

Forrest Hoffman
Oak Ridge National Laboratory

Adolfy Hoisie
Los Alamos National Laboratory

Jeff Hollingsworth
University of Maryland

Patricia D. Hough
Sandia National Laboratories

John C. Houghton
DOE Office of Science

Paul D. Hovland
Argonne National Laboratory

Ivan Hubeny
NASA / Goddard Space Flight Center

Thomas J. R. Hughes
University of Texas–Austin

Rob Jacob
Argonne National Laboratory

Curtis L. Janssen
Sandia National Laboratories

Stephen C. Jardin
Princeton Plasma Physics Laboratory

Philip Michael Jardine
Oak Ridge National Laboratory

Ken Jarman
Pacific Northwest National Laboratory

Fred Johnson
DOE Office of Science

William E. Johnston
Lawrence Berkeley National Laboratory

Philip Jones
Los Alamos National Laboratory

Kenneth I. Joy
University of California–Davis

Andreas C. Kabel
Stanford Linear Accelerator Center

Laxmikant V. Kale
University of Illinois at Urbana-Champaign

Kab Seok Kang
Old Dominion University

Alan F. Karr
National Institute of Statistical Services

Tom Katsouleas
University of Southern California

Brian Kauffman
National Center for Atmospheric Research

Sallie Keller-McNulty
Los Alamos National Laboratory

Carl Timothy Kelley
North Carolina State University

Stephen Kent
Fermi National Accelerator Laboratory

Darren James Kerbyson
Los Alamos National Laboratory

Jon Kettenring
Telcordia Technologies

David E. Keyes
Columbia University

Alexei M. Khokhlov
Naval Research Laboratory

Jeff Kiehl
National Center for Atmospheric Research

William H. Kirchhoff
DOE Office of Science

Stephen J. Klippenstein
Sandia National Laboratories

Dana Knoll
Los Alamos National Laboratory

Michael L. Knotek
DOE Office of Science

Kwok Ko
Stanford Linear Accelerator Center

Karl Koehler
National Science Foundation

Dale D. Koelling
DOE Office of Science

Peter Michael Kogge
University of Notre Dame

James Arthur Kohl
Oak Ridge National Laboratory

Tammy Kolda
Sandia National Laboratories

Marc Edward Kowalski
Stanford Linear Accelerator Center

Jean-Francois Lamarque
National Center for Atmospheric Research

David P. Landau
University of Georgia

Jay Larson
Argonne National Laboratory

Peter D. Lax
New York University

Lie-Quan Lee
Stanford Linear Accelerator Center

W. W. Lee
Princeton Plasma Physics Laboratory

John Wesley Lewellen
Argonne National Laboratory

Sven Leyffer
Argonne National Laboratory

Timothy Carl Lillestolen
University of Tennessee

Zhihong Lin
University of California–Irvine

Timur Linde
University of Chicago

Philip LoCascio
Oak Ridge National Laboratory

Steven G. Louie
University of California–Berkeley / Lawrence Berkeley National Laboratory

Steve Louis
Lawrence Livermore National Laboratory

Ewing L. Lusk
Argonne National laboratory

Mordecai-Mark Mac Low
American Museum of Natural History

Arthur B. Maccabe
University of New Mexico

David Malon
Argonne National Laboratory

Bob Malone
Los Alamos National Laboratory

Madhav V. Marathe
Los Alamos National Laboratory

William R. Martin
University of Michigan

Richard Matzner
University of Texas–Austin

Andrew McIlroy
Sandia National Laboratories

Lois Curfman McInnes
Argonne National Laboratory

Sally A. McKee
Cornell University

Gregory J. McRae
Massachusetts Institute of Technology

Lia Merminga
Thomas Jefferson National Accelerator Facility

Bronson Messer
University of Tennessee / Oak Ridge National Laboratory

Juan M. Meza
Lawrence Berkeley National Laboratory

Anthony Mezzacappa
Oak Ridge National Laboratory

George Michaels
Pacific Northwest National Laboratory

John Michalakes
National Center for Atmospheric Research

Don Middleton
National Center for Atmospheric Research

William H. Miner, Jr.
DOE Office of Science

Art Mirin
Lawrence Livermore National Laboratory

Lubos Mitas
North Carolina State University

Shirley V. Moore
University of Tennessee

Jorge J. Moré
Argonne National Laboratory

Warren B. Mori
University of California–Los Angeles

Jose L. Munoz
National Nuclear Security Administration

Jim Myers
Pacific Northwest National Laboratory

Pavan K. Naicker
Vanderbilt University

Habib Nasri Najm
Sandia National Laboratories

John W. Negele
Massachusetts Institute of Technology

Bill Nevins
Lawrence Livermore National Laboratory

Esmond G. Ng
Lawrence Berkeley National Laboratory

Michael L. Norman
University of California–San Diego

Peter Nugent
Lawrence Berkeley National Laboratory

Joseph C. Oefelein
Sandia National Laboratories

Rod Oldehoeft
Los Alamos National Laboratory

Gordon L. Olson
Los Alamos National Laboratory

George Ostrouchov
Oak Ridge National Laboratory

Jack Parker
Oak Ridge National Laboratory

Scott Edward Parker
University of Colorado

Bahram Parvin
Lawrence Berkeley National Laboratory

Valerio Pascucci
Lawrence Livermore National Laboratory

Peter Paul
Brookhaven National Laboratory

William Pennell
Pacific Northwest National Laboratory

Arnie Peskin
Brookhaven National Laboratory

Anders Petersson
Lawrence Livermore National Laboratory

Ali Pinar
Lawrence Berkeley National Laboratory

Tomasz Plewa
University of Chicago

Douglas E. Post
Los Alamos National Laboratory

Alex Pothen
Old Dominion University

Karsten Pruess
Lawrence Berkeley National Laboratory

Hong Qin
Princeton University

Larry A. Rahn
Sandia National Laboratories

Craig E. Rasmussen
Los Alamos National Laboratory

Katherine M. Riley
University of Chicago

Martei Ripeanu
University of Chicago

Charles H. Romine
DOE Office of Science

John Van Rosendale
DOE Office of Science

Robert Rosner
University of Chicago

Doron Rotem
Lawrence Berkeley National Laboratory

Doug Rotman
Lawrence Livermore National Laboratory

Robert Douglas Ryne
Lawrence Berkeley National Laboratory

P. Sadayappan
Ohio State University

Roman Samulyak
Brookhaven National Laboratory

Todd Satogata
Brookhaven National Laboratory

Dolores A. Shaffer
Science and Technology Associates, Inc.

George C. Schatz
Northwestern University

David Paul Schissel
General Atomics

Thomas Christoph Schulthess
Oak Ridge National Laboratory

Mary Anne Scott
DOE Office of Science

Stephen L. Scott
Oak Ridge National Laboratory

David B. Serafini
Lawrence Berkeley National Laboratory

James Sethian
University of California–Berkeley

John Nicholas Shadid
Sandia National Laboratories

Mikhail Shashkov
Los Alamos National Laboratory

John Shalf
Lawrence Berkeley National Laboratory

Henry Shaw
DOE Office of Science

Mark S. Shephard
Rensselaer Polytechnic Institute

Jason F. Shepherd
Sandia National Laboratories

Timothy Shippert
Pacific Northwest National Laboratory

Andrei P. Shishlo
Oak Ridge National Laboratory

Arie Shoshani
Lawrence Berkeley National Laboratory

Andrew R. Siegel
Argonne National Laboratory / University of Chicago

Sonya Teresa Smith
Howard University

Mitchell D. Smooke
Yale University

Allan Edward Snavely
San Diego Supercomputer Center

Matthew Sottile
Los Alamos National Laboratory

Carl R. Sovinec
University of Wisconsin

Panagiotis Spentzouris
Fermi National Accelerator Facility

Daniel Shields Spicer
NASA / Goddard Space Flight Center

Bill Spotz
National Center for Atmospheric Research

David J. Srolovitz
Princeton University

T. P. Straatsma
Pacific Northwest National Laboratory

Michael Robert Strayer
Oak Ridge National Laboratory

Scott Studham
Pacific Northwest National Laboratory

Robert Sugar
University of California–Santa Barbara

Shuyu Sun
University of Texas–Austin

F. Douglas Swesty
State University of New York–Stony Brook

John Taylor
Argonne National Laboratory

Ani R. Thakar
Johns Hopkins University

Andrew F. B. Tompson
Lawrence Livermore National Laboratory

Charles H. Tong
Lawrence Livermore National Laboratory

Josep Torrellas
University of Illinois

Harold Eugene Trease
Pacific Northwest National Laboratory

Albert J. Valocchi
University of Illinois

Debra van Opstal
Council on Competitiveness

Leticia Velazquez
University of Texas–El Paso

Roelof Jan Versteeg
Idaho National Energy and Engineering Laboratory

Jeffrey Scott Vetter
Lawrence Livermore National Laboratory

Angela Violi
University of Utah

Robert Voigt
College of William and Mary

Gregory A. Voth
University of Utah

Albert F. Wagner
Argonne National Laboratory

Homer F. Walker
Worcester Polytechnic Institute

Lin-Wang Wang
Lawrence Berkeley National Laboratory

Warren Washington
National Center for Atmospheric Research

Chip Watson
Thomas Jefferson National Accelerator Facility

Michael Wehner
Lawrence Livermore National Laboratory

Mary F. Wheeler
University of Texas–Austin

Roy Whitney
Thomas Jefferson National Accelerator Facility

Theresa L. Windus
Pacific Northwest National Laboratory

Steve Wojtkiewicz
Sandia National Laboratories

Pat Worley
Oak Ridge National Laboratory

Margaret H. Wright
New York University

John Wu
Lawrence Berkeley National Laboratory

Ying Xu
Oak Ridge National Laboratory

Zhiliang Xu
State University of New York–Stony Brook

Steve Yabusaki
Pacific Northwest National Laboratory

Woo-Sun Yang
Lawrence Berkeley National Laboratory

Katherine Yelick
University of California–Berkeley

Thomas Zacharia
Oak Ridge National Laboratory

Bernard Zak
Sandia National Laboratories

Shujia Zhou Zhou
Northrop Grumman / TASC

John P. Ziebarth
Los Alamos National Laboratory

1. INTRODUCTION

National investment in large-scale computing can be justified on numerous grounds. These include national security; economic competitiveness; technological leadership; formulation of strategic policy in defense, energy, environmental, health care, and transportation systems; impact on education and workforce development; impact on culture in its increasingly digital forms; and, not least, supercomputing's ability to capture the imagination of the nation's citizens. The public expects all to see the fruits of these historically established benefits of large-scale computing, and it expects that the United States will have the earliest access to future, as yet unimagined benefits.

This report touches upon many of these motivations for investment in large-scale computing. However, it is directly concerned with a deeper one, which in many ways controls the rest. The question this report seeks to answer is: *What case can be made for a national investment in large-scale computational modeling and simulation from the perspective of basic science?* This can be broken down into several smaller questions: *What scientific results are likely, given a significantly more powerful simulation capability, say, a hundred to a thousand times present capability? What principal hurdles can be identified along the path to realizing these new results? How can these hurdles be surmounted?*

A WORKSHOP TO FIND FRESH ANSWERS

These questions and others were posed to a large gathering of the nation's leading computational scientists at a workshop convened on June 24–25, 2003, in the shadow of the Pentagon. They were answered in 27 topical breakout groups, whose responses constitute the bulk of the second volume of this two-volume report.

The computational scientists participating in the workshop included those who regard themselves primarily as natural scientists—e.g., physicists, chemists, biologists—but also many others. By design, about one-third of the participants were computational mathematicians. Another third were computer scientists who are oriented toward research on the infrastructure on which computational modeling and simulation depends.

A CASCADIC STRUCTURE

Computer scientists and mathematicians are scientists whose research in the area of large-scale computing has direction and value of its own. However, the structure of the workshop, and of this report, emphasizes a cascadic flow from natural scientists to mathematicians and from both to computer scientists (Figure 1). We are primarily concerned herein with the opportunities for large-scale simulation to achieve new understanding in the natural sciences. In this context, the role of the mathematical sciences is to provide a means of passing from a physical model to a discrete computational representation of that model and of efficiently manipulating that representation to obtain results whose validity is well understood. In turn, the computer sciences allow the algorithms that perform these manipulations to be executed efficiently on leading-edge computer systems. They also provide ways for the often-monumental human efforts required in this process to be effectively abstracted, leveraged across other applications, and propagated over a complex and ever-evolving set of hardware platforms.

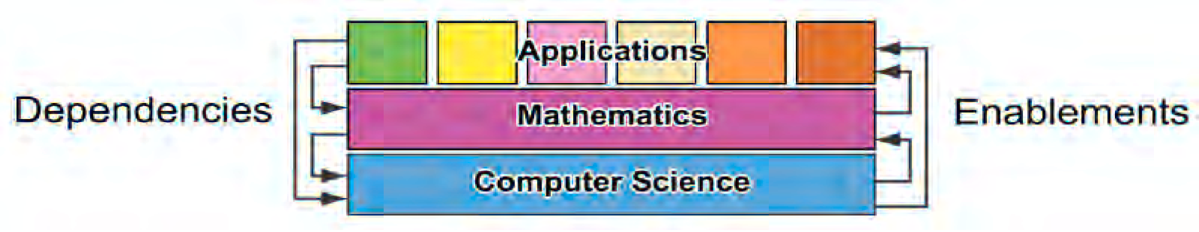


Figure 1. The layered, cascadic structure of computational science.

Most computational scientists understand that this layered set of dependencies—of natural science on mathematics and of both on computer science—is an oversimplification of the full picture of information flows that drive computational science. Indeed, one of the most exciting of many rapidly evolving trends in computational science is a "phase transition" (a concept to which we return in Chapter 2) from a field of solo scientific investigators, who fulfill their needs by finding computer software "thrown over the wall" by computational tool builders, to large, coordinated groups of natural scientists, mathematicians, and computer scientists who carry on a constant conversation about what can be done now, what is possible in the future, and how to achieve these goals most effectively.

Almost always, there are numerous ways to translate a physical model into mathematical algorithms and to implement a computational program on a given computer. Uninformed decisions made early in the process can cut off highly productive options that may arise later on. Only a bi-directional dialog, up and down the cascade, can systematically ensure that the best resources and technologies are employed to solve the most challenging scientific problems.

Its acknowledged limitations aside, the cascadic model of contemporary computational science is useful for understanding the importance of several practices in the contemporary computational research community. These include striving towards the abstraction of technologies and towards identification of universal interfaces between well-defined functional layers (enabling reuse across many applications). It also motivates sustained investment in the cross-cutting technologies of computational mathematics and computer science, so that increases in power and capability may be continually delivered to a broad portfolio of scientific applications on the other side of a standardized software interface.

SIMULATION AS A PEER METHODOLOGY TO EXPERIMENT AND THEORY

Historians of science may pick different dates for the development of the twin pillars of theory and experiment in the "scientific method," but both are ancient, and their interrelationship is well established. Each takes a turn leading the other as they mutually refine our understanding of the natural world—experiment confronting theory with new puzzles to explain and theory showing experiment where to look next to test its explanations. Unfortunately, theory and experiment both possess recognized limitations at the frontier of contemporary science.

The strains on theory were apparent to John Von Neumann (1903–1957) and drove him to develop the computational sciences of fluid dynamics, radiation transport, weather prediction, and other fields. Models of these phenomena, when expressed as mathematical equations, are inevitably large-scale and nonlinear, while the bulk of the edifice of mathematical theory (for algebraic, differential, and integral equations) is linear. Computation was to Von Neumann, and remains today, the only truly systematic means of making progress in these and many other scientific arenas. Breakthroughs in the theory of nonlinear systems come occasionally, but computational gains come steadily with increased computational power and resolution.

The strains on experimentation, the gold standard of scientific truth, have grown along with expectations for it (Figure 2). Unfortunately, many systems and many questions are nearly inaccessible to experiment. (We subsume under "experiment" both experiments designed and conducted by scientists, and observations of natural phenomena out of the direct control of scientists, such as celestial events.) The experiments that scientists need to perform to answer the most pressing questions are sometimes deemed unethical (e.g., because of their impact on living beings), hazardous (e.g., because of their impact on the life-sustaining environment we have on earth), politically untenable (e.g., prohibited by treaties to which the United States is a party), difficult (e.g., requiring measurements that are too rapid or numerous to be instrumentable or time periods too long to complete), or simply expensive.

The cost of experimentation is particularly important to consider when considering simulation as an alternative, for it cannot be denied that simulation can also be expensive. Cost has not categorically denied experimentalists their favored tools, such as accelerators, orbital telescopes, high-power lasers, and the like, at prices sometimes in the billions of dollars per facility, although such facilities are carefully vetted and planned. Cost must also not categorically deny computational scientists their foremost scientific instrument—the

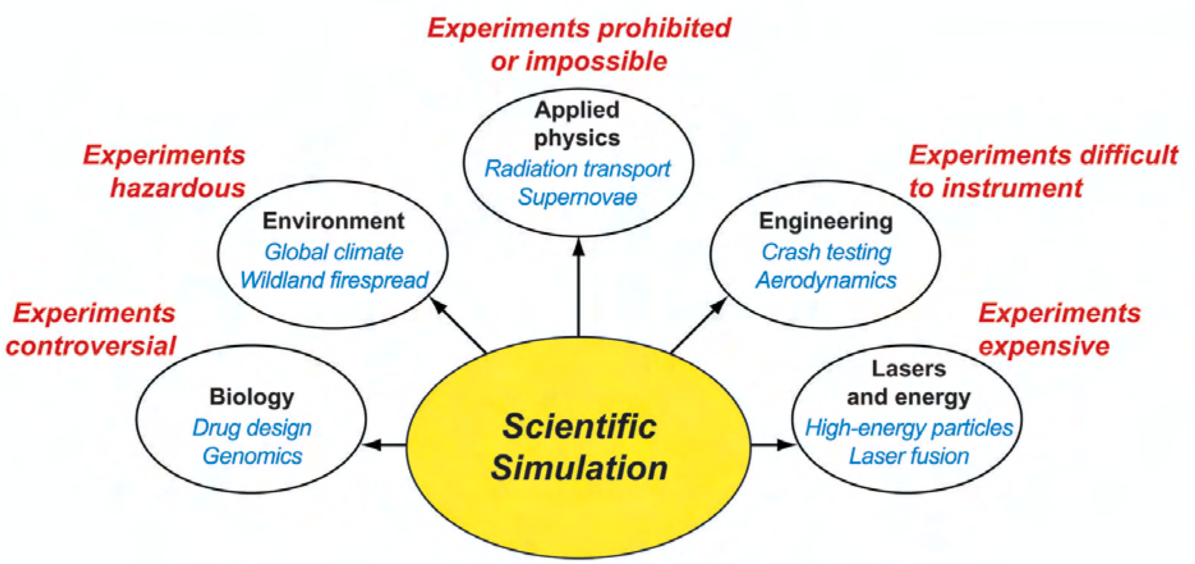


Figure 2. Practical strains on experimentation that invite simulation as a peer modality of scientific investigation.

supercomputer and its full complement of software, networking, and peripheral devices. It will be argued in the body of this report that simulation is, in fact, a highly cost-effective lever for the experimental process, allowing the same, or better science to be accomplished with fewer, better-conceived experiments.

Simulation has aspects in common with both theory and experiment. It is fundamentally theoretical, in that it starts with a theoretical model—typically a set of mathematical equations. A powerful simulation capability breathes new life into theory by creating a demand for improvements in mathematical models. Simulation is also fundamentally experimental, in that upon constructing and implementing a model, one observes the transformation of inputs (or controls) to outputs (or observables).

Simulation effectively *bridges* theory and experiment by allowing the execution of "theoretical experiments" on systems, including those that could never exist in the physical world, such as a fluid without viscosity. Computation also bridges theory and experiment by virtue of the computer's serving as a universal and versatile *data host*. Once experimental data have been digitized, they can be compared side-by-side with simulated results in visualization systems built for the latter, reliably transmitted, and retrievably archived. Moreover, simulation and experiment can *complement* each other by allowing a complete picture of a system that neither can provide as well alone. Some data may not be measurable with the best experimental techniques available, and some mathematical models may be too sensitive to unknown parameters to invoke with confidence. Simulation can be used to "fill in" the missing experimental fields, using experimentally measured fields as input data. Data assimilation can also be systematically employed throughout a simulation to keep it from "drifting" from measurements, thus overcoming the effect of modeling uncertainty.

Further comparing simulation to experiment, one observes a sociological or political advantage to increased investment in the tools of simulation: they are widely usable across the entire scientific community. A micro-array analyzer is of limited use to a plasma physicist and a tokamak of limited use to a biologist. However, a large computer or an optimization algorithm in the form of portable software may be of use to both.

HURDLES TO SIMULATION

While the promise of simulation is profound, so are its limitations. Those limitations often come down to a question of resolution. Though of vast size, computers are triply finite: they represent individual quantities only to a finite precision, they keep track of only a finite number of such quantities, and they operate at a finite rate.

Although all matter is, in fact, composed of a finite number of particles (atoms), the number of particles in a macroscopic sample of matter, on the order of a trillion particles, places simulations at macroscopic scales from "first principles" (i.e., from the quantum theory of electronic structure) well beyond any conceivable computational capability. Similar problems arise when time scales are considered. For example, the range of time scales in protein folding is 12 orders of magnitude, since a process that takes milliseconds occurs in molecular dance steps that last femtoseconds (a trillion times shorter)—again, far too wide a range to routinely simulate using first principles.

Even simulation of systems adequately described by equations of the macroscopic continuum—fluids such as air or water—can be daunting for the most powerful computers available today. Such computers are capable of execution rates in the tens of teraflop / s (one teraflop / s is one trillion arithmetic operations per second) and can cost tens to hundreds of millions of dollars to purchase and millions of dollars per year to operate. However, to simulate fluid-mechanical turbulence in the boundary and wake regions of a typical vehicle using "first principles" of continuum modeling would tie up such a computer for months, which makes this level of simulation too expensive for routine use.

One way to describe matter at the macroscopic scale is to divide space up into small cubes and to ascribe values for density, momentum, temperature, and so forth, to each cube. As the cubes become smaller and smaller, a more and more accurate description is obtained, at a cost of increasing memory to store the values and time to visit and update each value in accordance with natural laws, such as the conservation of energy applied to the cube, given fluxes in and out of neighboring cubes. If the number of cubes along each side is doubled, the cost of the simulation, which is proportional to the total number of cubes, increases by a factor of 2^3 or 8. This is the "curse of dimensionality": increasing the resolution of a simulation quickly eats up any increases in processor power resulting from the well-known Moore's Law (a doubling of computer speed every 18–24 months). Furthermore, for time-dependent problems in three dimensions, the cost of a simulation grows like the fourth power of the resolution. Therefore, an increase in computer performance by a factor of 100 provides an increase in resolution in each spatial and temporal dimension by a factor of only about 3.

One way to bring more computing power to bear on a problem is to divide the work to be done over many processors. Parallelism in computer architecture—the concurrent use of many processors—can provide additional factors of hundreds or more, beyond Moore's Law alone, to the aggregate performance available for the solution of a given problem. However, the desire for increased resolution and, hence, accuracy is seemingly insatiable. This simply expressed and simply understood "curse of dimensionality" means that "business as usual" in scaling up today's simulations by riding the computer performance curve will never be cost-effective—and is too slow!

The "curse of knowledge explosion"—namely, that no one computational scientist can hope to track advances in all of the facets of the mathematical theories, computational models and algorithms, applications and computing systems software, and computing hardware (computers, data stores, and networks) that may be needed in a successful simulation effort—is another substantial hurdle to the progress of simulation. Both the curse of dimensionality and the curse of knowledge explosion can be addressed. In fact, they have begun to be addressed in a remarkably successful initiative called Scientific Discovery through Advanced Computing (SciDAC), which was launched by the U.S. Department of Energy's (DOE's) Office of Science in 2001. SciDAC was just a start; many difficult challenges remain. Nonetheless, SciDAC (discussed in more detail in Chapter 2) established a model for tackling these "curses" that promises to bring the power of the most advanced computing technologies to bear on the most challenging scientific and engineering problems facing the nation.

In addition to the technical challenges in scientific computing, too few computational cycles are currently available to fully capitalize upon the progress of the SciDAC initiative. Moreover, the computational science talent emerging from the national educational pipeline is just a trickle in comparison to what is needed to replicate the successes of SciDAC throughout the many applications of science and engineering that lag behind. Cycle-starvation and human resource–starvation are serious problems that are felt across all of the computational science programs surveyed in creating this report. However, they are easier hurdles to overcome than the above-mentioned “curses.”

SUMMARY OF RECOMMENDATIONS

We believe that the most effective program to deliver new science by means of large-scale simulation will be one built over a broad base of disciplines in the natural sciences and the associated enabling technologies, since there are numerous synergisms between them, which are readily apparent in the companion volume.

We also believe that a program to realize the promises outlined in the companion volume requires a balance of investments in hardware, software, and human resources. While quantifying and extrapolating the thirst for computing cycles and network bandwidth is relatively straightforward, the scheduling of breakthroughs in mathematical theories and computational model and algorithms has always been difficult. Breakthroughs are more likely, however, where there is critical mass of all types of computational scientists and good communication between them.

The following recommendations are elaborated upon in Chapter 6 of this volume:

- Major new investments in computational science are needed in all of the mission areas of the Office of Science in DOE, as well as those of many other agencies, so that the United States may be the first, or among the first, to capture the new opportunities presented by the continuing advances in computing power. Such investments will extend the important scientific opportunities that have been attained by a fusion of sustained advances in scientific models, mathematical algorithms, computer architecture, and scientific software engineering.

- Multidisciplinary teams, with carefully selected leadership, should be assembled to provide the broad range of expertise needed to address the intellectual challenges associated with translating advances in science, mathematics, and computer science into simulations that can take full advantage of advanced computers.

- Extensive investment in new computational facilities is strongly recommended, since simulation now cost-effectively complements experimentation in the pursuit of the answers to numerous scientific questions. New facilities should strike a balance between capability computing for those “heroic simulations” that cannot be performed any other way and capacity computing for “production” simulations that contribute to the steady stream of progress.

- Investment in hardware facilities should be accompanied by sustained collateral investment in software infrastructure for them. The efficient use of expensive computational facilities and the data they produce depends directly upon multiple layers of system software and scientific software, which, together with the hardware, are the “engines of scientific discovery” across a broad portfolio of scientific applications.

- Additional investments in hardware facilities and software infrastructure should be accompanied by sustained collateral investments in algorithm research and theoretical development. Improvements in basic theory and algorithms have contributed as much to increases in computational simulation capability as improvements in hardware and software over the first six decades of scientific computing.

- Computational scientists of all types should be proactively recruited with improved reward structures and opportunities as early as possible in the educational process so that the number of trained computational science professionals is sufficient to meet present and future demands.

- Sustained investments must be made in network infrastructure for access and resource sharing as well as in the software needed to support collaborations among distributed teams of scientists, in recognition of the fact that the best possible computational science teams will be widely separated geographically and that researchers will generally not be co-located with facilities and data.

- Federal investment in innovative, high-risk computer architectures that are well suited to scientific and engineering simulations is both appropriate and needed to complement commercial research and development. The commercial computing marketplace is no longer effectively driven by the needs of computational science.

SCOPE

The purpose of this report is to capture the projections of the nation’s leading computational scientists and providers of enabling technologies concerning what new science lies around the corner with the next increase in delivered computational power by a factor of 100 to 1000. It also identifies the issues that must be addressed if these increases in computing power are to lead to remarkable new scientific discoveries.

It is beyond the charter and scope of this report to form policy recommendations on how this increased capability and capacity should be procured, or how to prioritize among numerous apparently equally ripe scientific opportunities.

2. SCIENTIFIC DISCOVERY THROUGH ADVANCED COMPUTING: A SUCCESSFUL PILOT PROGRAM

THE BACKGROUND OF SciDAC

For more than half a century, visionaries anticipated the emergence of computational modeling and simulation as a peer to theory and experiment in pushing forward the frontiers of science, and DOE and its antecedent agencies over this period have invested accordingly. Computational modeling and simulation made great strides between the late 1970s and the early 1990s as Seymour Cray and the company that he founded produced ever-faster versions of Cray supercomputers. Between 1976, when the Cray 1 was delivered to DOE’s Los Alamos National Laboratory, and 1994, when the last traditional Cray supercomputer, the Cray T90, was introduced, the speed of these computers increased from 160 megaflop / s to 32 gigaflop / s, a factor of 200! However, by the early 1990s it became clear that building ever-faster versions of Cray supercomputers using traditional electronics technologies was not sustainable.

In the 1980s DOE and other federal agencies began investing in alternative computer architectures for scientific computing featuring multiple processors connected to multiple memory units through a wide variety of mechanisms. A wild decade ensued during which many different physical designs were explored, as well as many programming models for controlling the flow of data in them, from memory to processor and from one memory unit to another. The mid-1990s saw a substantial convergence, which found commercial microprocessors, each with direct local access to only a relatively small fraction of the total memory, connected in clusters numbering in the thousands through dedicated networks or fast switches. The speeds of the commercial microprocessors in these designs increased according to Moore’s Law, producing distributed-memory parallel computers that rivaled the power of traditional Cray supercomputers. These new machines required a major recasting of the algorithms that scientists had developed and polished for more than two decades on Cray supercomputers, but those who invested the effort found that the new massively parallel computers provided impressive levels of computing capability.

By the late 1990s it was evident to many that the continuing, dramatic advances in computing technologies had the potential to revolutionize scientific computing. DOE’s National Nuclear Security Agency (DOE-NNSA) launched the Accelerated Strategic Computing Initiative (ASCI) to help ensure the safety and reliability of the nation’s nuclear stockpile in the absence of testing. In 1998, DOE’s Office of Science (DOE-SC) and the National Science Foundation (NSF) sponsored a workshop at the National Academy of Sciences to identify the opportunities and challenges of realizing major advances in computational modeling and simulation. The report from this workshop identified scientific advances across a broad range of science, ranging in scale from cosmology through nanoscience, which would be made possible by advances in computational modeling and simulation. However, the report also noted challenges to be met if these advances were to be realized.

By the time the workshop report was published, the Office of Science was well along the path to developing an Office-wide program designed to address the challenges identified in the DOE-SC / NSF workshop. The response to the National Academy report, as well as to the 1999 report from the President's Information Technology Advisory Committee—the Scientific Discovery through Advanced Computing (SciDAC) program—was described in a 22-page March 2000 report by the Office of Science to the House and Senate Appropriations Committees. Congress funded the program in the following fiscal year, and by the end of FY 2001, DOE-SC had established 51 scientific projects, following a national peer-reviewed competition.

SciDAC explicitly recognizes that the advances in computing technologies that provide the foundation for the scientific opportunities identified in the DOE-SC / NSF report (as well as in Volume 2 of the current report) are not, in themselves, sufficient for the "coming of age" of computational simulation. Close integration of three disciplines—science, computer science, and applied mathematics—and a new discipline at their intersection called computational science are needed to realize the full benefits of the fast-moving advances in computing technologies. A change in the sociology of this area of science was necessary for the fusion of these streams of disciplinary developments. While not uniquely American in its origins, the "new kind of computational science" promoted in the SciDAC report found fertile soil in institutional arrangements for the conduct of scientific research that are highly developed in the United States, and seemingly much less so elsewhere: multiprogram, multidisciplinary national laboratories, and strong laboratory-university partnerships in research and graduate education. SciDAC is built on these institutional foundations as well as on the integration of computational modeling, computer algorithms, and computing systems software.

In this chapter, we briefly examine the short but substantial legacy of SciDAC. Our purpose is to understand the implications of its successes, which will help us identify the logical next steps in advancing computational modeling and simulation.

SciDAC'S INGREDIENTS FOR SUCCESS

Aimed at Discovery

SciDAC boldly affirms the importance of computational simulation for new scientific discovery, not just for "rationalizing" the results of experiments. The latter role for simulation is a time-honored one from the days when it was largely subordinate to experimentation, rather than peer to it. That relationship has been steadily evolving. In a frontispiece quotation from J. S. Langer, the chair of the July 1998 multiagency National Workshop on Advanced Scientific Computing, the SciDAC report declared: "The computer literally is providing a new window through which we can observe the natural world in exquisite detail."

Through advances in computational modeling and simulation, it will be possible to extend dramatically the exploration of the fundamental processes of nature—e.g., the interactions between elementary particles that give rise to the matter around us, the interactions between proteins and other molecules that sustain life—as well as advance our ability to predict the behavior of a broad range of complex natural and engineered systems—e.g., nanoscale devices, microbial cells, fusion energy reactors, and the earth's climate.

Multidisciplinary

The structure of SciDAC boldly reflects the fact that the development and application of leading-edge simulation capabilities has become a multidisciplinary activity. This implies, for instance, that physicists are able to focus on the physics and not on developing parallel libraries for core mathematical operations. It also implies that mathematicians and computer scientists have direct access to substantive problems of high impact, and not only models loosely related to application-specific difficulties. The deliverables and accountabilities of the collaborating parties are cross-linked. The message of SciDAC is a "declaration of interdependence."

Specifically, the SciDAC report called for

- creation of a new generation of scientific simulation codes that take full advantage of the extraordinary computing capabilities of terascale computers;
- creation of the mathematical and systems software to enable the scientific simulation codes to effectively and efficiently use terascale computers; and
- creation of collaboratory software infrastructure to enable geographically separated scientists to work together effectively as a team and to facilitate remote access to both facilities and data.

The $57 million per year made available under SciDAC is divided roughly evenly into these three categories. SciDAC also called for upgrades to the scientific computing hardware infrastructure of the Office of Science and outlined a prescription for building a robust, agile, and cost-effective computing infrastructure. These recommendations have thus far been modestly supported outside of the SciDAC budget.

Figure 3, based on a figure from the original SciDAC report, shows the interrelationships between SciDAC program elements as

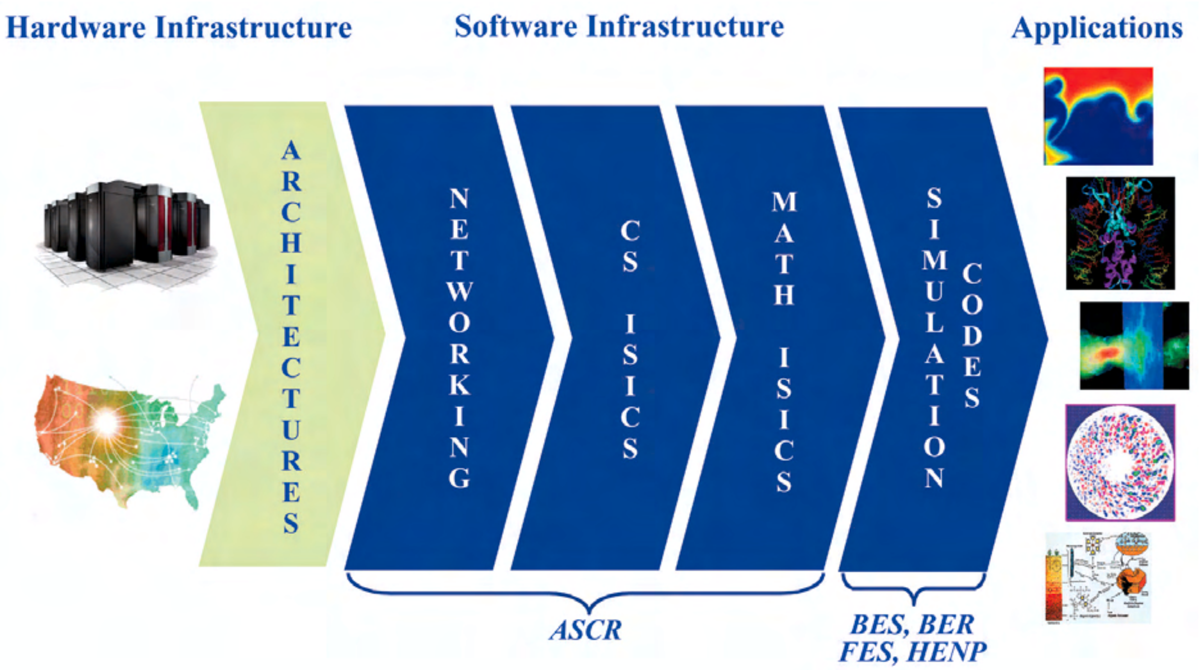


Figure 3. Relationship between the science application teams, the Integrated Software Infrastructure Centers (ISICs), and networking and computing facilities in the SciDAC program.

implemented, particularizing Figure 1 of this report to the SciDAC program. All five program offices in the Office of Science are participating in SciDAC. The disciplinary computational science groups are being funded by the Offices of Basic Energy Sciences, Biology and Environmental Research, Fusion Energy Sciences, and High Energy and Nuclear Physics, and the integrated software infrastructure centers and networking infrastructure centers are being funded by the Office of Advanced Scientific Computing Research.

Multi-Institutional

Recognizing that the expertise needed to make major advances in computational modeling and algorithms and computing systems software was distributed among various institutions across the United States, SciDAC established scientifically natural collaborations without regard for geographical or institutional boundaries—laboratory-university collaborations and laboratory-laboratory collaborations. Laboratories and universities have complementary strengths in research. Laboratories are agile in reorganizing their research assets to respond to missions and new opportunities. Universities respond on a much slower time scale to structural change, but they are hotbeds of innovation where graduate students explore novel ideas as an integral part of their education. In the area of simulation, universities are often sources of new models, algorithms, and software and hardware concepts that can be evaluated in a dissertation-scale effort. However, such developments may not survive the graduation of the doctoral candidate and are rarely rendered into reusable, portable, extensible infrastructure in the confines of the academic environment, where the reward structure emphasizes the original discovery and proof of concept. Laboratory teams excel in creating, refining, "hardening," and deploying computational scientific infrastructure. Thirteen laboratories and 50 universities were funded in the initial round of SciDAC projects.

Setting New Standards for Software

For many centuries, publication of original results has been the gold standard of accomplishment for scientific researchers. During the first 50 years of scientific computing, most research codes targeted a few applications on a small range of hardware, and were designed to be used by experts only (often just the author). Code development projects emphasized finding the shortest path to the next publishable result. Packaging code for reuse was not a job for serious scientists. As the role of computational modeling and simulations in advancing scientific discovery grows, this culture is changing. Scientists now realize the value of high-quality, reusable, extensible, portable software, and the producers of widely used packages are highly respected.

Unfortunately, funding structures in the federal agencies have evolved more slowly and it has not always been easy for scientists to find support for the development of software for the research community. By dedicating more than half its resources to the development of high-value scientific applications software and the establishment of integrated software infrastructure centers, SciDAC has supported efforts to make advanced scientific packages easy to use, port them to new hardware, maintain them, and make them capable of interoperating with other related packages that achieve critical mass of use and offer complementary features.

SciDAC recognizes that large scientific simulation codes have many needs in common: modules that construct and adapt the computational mesh on which their models are based, modules that produce the discrete equations from the mathematical model, modules that solve the resulting equations, and modules that visualize the results and manage, mine, and analyze the large data sets. By constructing these modules as interoperable components and supporting the resulting component libraries so that they stay up to date with the latest algorithmic advances and work on the latest hardware, SciDAC amortizes costs and creates specialized points of contact for a portfolio of applications.

The integrated software infrastructure centers that develop these libraries are motivated to ensure that their users have available the full range of algorithmic options under a common interface, along with a complete description of their resource tradeoffs (e.g., memory versus time). Their users can try all reasonable options easily and adapt to different computational platforms without recoding the applications. Furthermore, when an application group intelligently drives software infrastructure research in a new and useful direction, the result is soon available to all users.

Large Scale

Because computational science is driven toward the large scale by requirements of model fidelity and resolution, SciDAC emphasizes the development of software for the most capable computers available. In the United States, these are now primarily hierarchical distributed-memory computers. Enough experience was acquired on such machines in heroic mission-driven simulation programs in the latter half of the 1990s (in such programs as DOE’s ASCI) to justify them as cost-effective simulation hardware, provided that certain generic difficulties in programming are amortized over many user groups and many generations of hardware. Still, many challenges remain to making efficient and effective use of these computers for the broad range of scientific applications important to the mission of DOE’s Office of Science.

Supporting Community Codes

As examples of simulations from Office of Science programs, the SciDAC report highlighted problems of ascertaining chemical structure and the massively parallel simulation code NWCHEM, winner of a 1999 R&D 100 Award and a Federal Laboratory Consortium Excellence in Technology Transfer Award in 2000.

NWCHEM, currently installed at nearly 900 research institutions worldwide, was created by an international team of scientists from twelve institutions (two U.S. national laboratories, two European national laboratories, five U.S. universities, and three European universities) over the course of five years. The core team consisted of seven full-time theoretical and computational chemists, computer scientists, and applied mathematicians, ably assisted by ten postdoctoral fellows on temporary assignment. Overall, NWCHEM represents an investment in excess of 100 person-years. Yet, the layered structure of NWCHEM permits it to easily absorb new physics or algorithm modules, to run on environments from laptops to the fastest parallel architectures, and to be ported to a new supercomputing environment with isolated minor modifications, typically in less than a week. The scientific legacy embodied in NWCHEM goes back three decades to the Gaussian code, for which John Pople won the 1998 Nobel Prize in chemistry.

Community codes such as Gaussian and NWCHEM can consume hundreds of person-years of development (Gaussian probably has 500–1000 person-years invested in it), run at hundreds of installations, are given large fractions of community computing resources for decades, and acquire an authority that can enable or limit what is done and accepted as science in their respective communities. Yet, historically, not all community codes have been crafted with the care of NWCHEM to achieve performance, portability, and extensibility. Except at the beginning, it is difficult to promote major algorithmic changes in such codes, since change is expensive and pulls resources from core progress. SciDAC application groups have been chartered to build new and improved community codes in areas such as accelerator physics, materials science, ocean circulation, and plasma physics. The integrated software infrastructure centers of SciDAC have a chance, due to the interdependence built into the SciDAC program structure, to simultaneously influence all of these new community codes by delivering software incorporating optimal algorithms that may be reused across many applications. Improvements driven by one application will be available to all. While SciDAC application groups are developing codes for *communities of users*, SciDAC enabling technology groups are building codes for *communities of developers*.

Enabling Success

The first two years of SciDAC provide ample evidence that constructing multidisciplinary teams around challenge problems and providing centralized points of contact for mathematical and computational technologies is a sound investment strategy. For instance, Dalton Schnack of General Atomics, whose group operates and maintains the simulation code NIMROD to support experimental design of magnetically confined fusion energy devices, states that software recently developed by Xiaoye (Sherry) Li of Lawrence Berkeley National Laboratory has enhanced his group's ability to perform fusion simulations to a degree equivalent to three to five years of riding the wave of hardware improvements alone. Ironically, the uniprocessor version of Li's solver, originally developed while she was a graduate student at the University of California–Berkeley, predated the SciDAC initiative; but it was not known to the NIMROD team until it was recommended during discussions between SciDAC fusion and solver software teams. In the meantime, SciDAC-funded research improved the parallel performance of Li's solver in time for use in NIMROD's current context.

M3D, a fusion simulation code developed at the Princeton Plasma Physics Laboratory, is complementary to NIMROD; it uses a different mathematical formulation. Like NIMROD and many simulation codes, M3D spends a large fraction of its time solving sparse linear algebraic systems. In a collaboration that predated SciDAC, the M3D code group was using PETSc, a portable toolkit of sparse solvers developed at Argonne National Laboratory and in wide use around the world. Under SciDAC, the Terascale Optimal PDE Solvers integrated software center provided another world-renowned class of solvers, Hypre, from Lawrence Livermore National Laboratory "underneath" the same code interface that M3D was already using to call PETSc's solvers. The combined PETSc-

Hypre solver library allows M3D to solve its current linear systems two to three times faster than previously, and this SciDAC collaboration has identified several avenues for improving the PETSc-Hypre code support for M3D and other users.

While these are stories of performance improvement, allowing faster turnaround and more production runs, there are also striking stories of altogether new physics enabled by SciDAC collaborations. The Princeton fusion team has combined forces with the Applied Partial Differential Equations Center, an integrated software infrastructure center, to produce a simulation code capable of resolving the fine structure of the dynamic interface between two fluids over which a shock wave passes—the so-called Richtmyer-Meshkov instability—in the presence of a magnetic field. This team has found that a strong magnetic field stabilizes the interface, a result not merely of scientific interest but also of potential importance in many devices of practical interest to DOE.

SciDAC's Terascale Supernova Initiative (TSI), led by Tony Mezzacappa of Oak Ridge National Laboratory, has achieved the first stellar core collapse simulations for supernovae with state-of-the-art neutrino transport. The team has also produced state-of-the-art models of nuclei in a stellar core. These simulations have brought together two fundamental fields at their respective frontiers, demonstrated the importance of accurate nuclear physics in supernova models, and motivated planned experiments to measure the nuclear processes occurring in stars. This group has also performed simulations of a rapid neutron capture process believed to occur in supernovae and to be responsible for the creation of half the heavy elements in the universe. This has led to a surprising and important result: this process can occur in environments very different than heretofore thought, perhaps providing a way around some of the fundamental difficulties encountered in past supernova models in explaining element synthesis. To extend their model to higher dimensions, the TSI team has engaged mathematicians and computer scientists to help them overcome the "curse of dimensionality" and exploit massively parallel machines. Mezzacappa has declared that he would never go back to doing computational physics outside of a multidisciplinary team.

Setting a New Organizational Paradigm

By acknowledging sociological barriers to multidisciplinary research up front and building accountabilities into the project awards to force the mixing of "scientific cultures" (e.g., physics with computer science), SciDAC took a risk that has paid off handsomely in technical results. As a result, new DOE-SC initiatives are taking SciDAC-like forms—for instance, the FY 2003 initiative Nanoscience Theory, Modeling and Simulation. The roadmap for DOE's Fusion Simulation Program likewise envisions multidisciplinary teams constructing complex simulations linking together what are today various freestanding simulation codes for different portions of a fusion reactor. Such complex simulations are deemed essential to the on-time construction of a working demonstration International Thermonuclear Experimental Reactor.

In many critical application domains, such teams are starting to self-assemble as a natural response to the demands of the next level of scientific opportunity. This self-assembly represents a phase transition in the computational science universe, from autonomous individual investigators to coordinated multidisciplinary teams. When experimental physicists transitioned from solo investigators to members of large teams centered at major facilities, they learned to include the cost of other professionals (statisticians, engineers, computer scientists, technicians, administrators, and others) in their projects. Computational physicists are now learning to build their projects in large teams built around major computational facilities and to count the cost of other professionals. All such transitions are sociologically difficult. In the case of computational simulation, there is sufficient infrastructure in common between different scientific endeavors that DOE and other science agencies can amortize its expense and provide organizational paradigms for it. SciDAC is such a pilot paradigm. It is time to extend it.

3. ANATOMY OF A LARGE-SCALE SIMULATION

Because the methodologies and processes of computational simulation are still not familiar to many scientists and laypersons, we devote this brief chapter to their illustration. Our goal is to acquaint the reader with a number of issues that arise in computational simulation and to highlight the benefits of collaboration across the traditional boundaries of science. The structure of any program aimed at advancing computational science must be based on this understanding.

Scientific simulation is a vertically integrated process, as indicated in Figures 1 and 3. It is also an iterative process, in which computational scientists traverse several times a hierarchy of issues in modeling, discretization, solution, code implementation, hardware execution, visualization and interpretation of results, and comparison to expectations (theories, experiments, or other simulations). They do this to develop a reliable "scientific instrument" for simulation, namely a marriage of a code with a computing and user environment. This iteration is indicated in the two loops (light arrows) of Figure 4 (reproduced from the SciDAC report).

One loop over this computational process is related to *validation* of the model and *verification* that the process is correctly executing the model. With slight violence to grammar and with some oversimplification, validation poses the question "Are we solving the *right equations*?" and verification, the question "Are we solving the *equations right*?"

The other loop over the process of simulation is related to *algorithm and performance tuning*. A simulation that yields high-fidelity results is of little use if it is too expensive to run or if it cannot be scaled up to the resolutions, simulation times, or ensemble sizes required to describe the real-world phenomena of interest. Again with some oversimplification, algorithm tuning poses the question, "Are we getting enough *science per operation*?" and

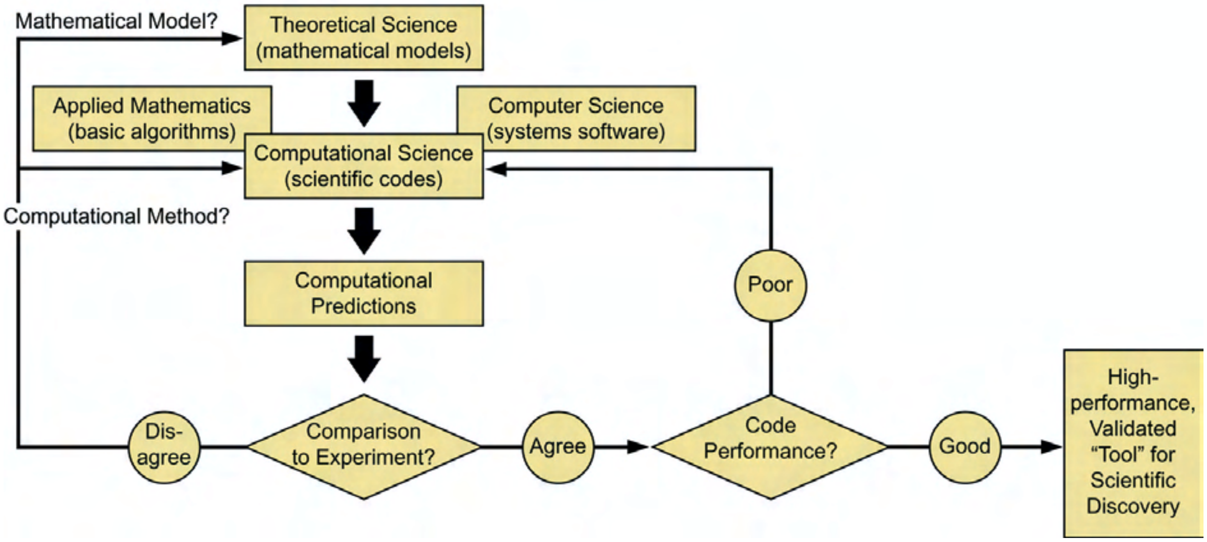

Mathematical Model?
Theoretical Science (mathematical models)
Applied Mathematics (basic algorithms)
Computer Science (systems software)
Computational Science (scientific codes)
Computational Method?
Computational Predictions
Poor
Dis-agree
Comparison to Experiment?
Agree
Code Performance?
Good
High-performance, Validated "Tool" for Scientific Discovery

Figure 4. The process of scientific simulation, showing the validation and verification loop (*left*) and algorithm and performance tuning loop (*right*).

performance tuning, “Are we getting enough *operations per cycle*?”

It is clear from Figure 4 that a scientific simulation depends upon numerous components, all of which have to work properly. For example, the theoretical model must not overlook or misrepresent any physically important effect. The model must not be parameterized with data of uncertain quality or provenance. The mathematical translation of the theoretical model (which may be posed as a differential equation, for instance) to a discrete representation that a computer can manipulate (a large system of algebraic relationships based on the differential equation, for instance) inevitably involves some approximation. This approximation must be controlled by *a priori* analysis or *a posteriori* checks. The algorithm that solves the discrete representation of the system being modeled must reliably converge. The compiler, operating system, and message-passing software that translate the arithmetic manipulations of the solution algorithm into loads and stores from memory to registers, or into messages that traverse communication links in a parallel computer, must be reliable and fault-tolerant. They must be employed correctly by programmers and simulation scientists so that the results are well defined and repeatable. In a simulation that relies on multiple interacting models (such as fluids and structures in adjacent domains, or radiation and combustion in the same domain), information fluxes at the interface of the respective models, which represent physical fluxes in the simulated phenomenon, must be consistent. In simulations that manipulate or produce enormous quantities of data on multiple disks, files must not be lost or mistakenly overwritten, nor misfiled results extracted for visualization. Each potential pitfall may require a different expert to detect it, control it at an acceptable level, or completely eliminate it as a possibility in future designs. These are some of the activities that exercise the left-hand loop in Figure 4.

During or after the validation of a simulation code, attention must be paid to the performance of the code in each hardware / system software environment available to the research team. Efforts to control discretization and solution error in a simulation may grossly increase the number of operations required to execute a simulation, far out of proportion to their benefit. Similarly, “conservative” approaches to moving data within a processor or between processors, to reduce the possibility of errors or miscommunication, may require extensive synchronization or minimal data replication and may be inefficient on a modern supercomputer. More elegant, often adaptive, strategies must be adopted for the simulation to be execution-worthy on an expensive massively parallel computer, even though these strategies may require highly complex implementation. These are some of the activities that exercise the right-hand loop in Figure 4.

ILLUSTRATION: SIMULATION OF A TURBULENT REACTING FLOW

The dynamics of turbulent reacting flows have long constituted an important topic in combustion research. Apart from the scientific richness of the combination of fluid turbulence and chemical reaction, turbulent flames are at the heart of the design of equipment such as reciprocating engines, turbines, furnaces, and incinerators, for efficiency and minimal environmental impact. Effective properties of turbulent flame dynamics are also required as model inputs in a broad range of larger-scale simulation challenges, including fire spread in buildings or wildfires, stellar dynamics, and chemical processing.

Figure 5 shows a photograph of a laboratory-scale premixed methane flame, stabilized by a metal rod across the outlet of a cylindrical burner. Although experimental and theoretical investigations have been able to provide substantial insights into the dynamics of these types of flames, there are still a number of outstanding questions about their behavior. Recently, a team of computational scientists supported by SciDAC, in collaboration with experimentalists at Lawrence Berkeley National Laboratory, performed the first simulations of these types of flames using detailed chemistry and transport. Figure 6 shows an instantaneous image of the flame surface from the simulation. (Many such instantaneous configurations are averaged in the photographic image of Figure 5.)

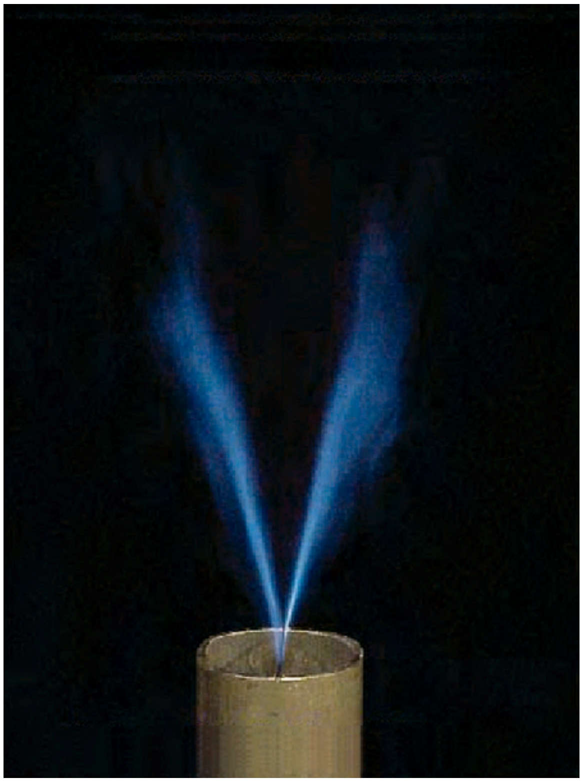


Figure 5. Photograph of an experimental turbulent premixed rod-stabilized “V” methane flame.

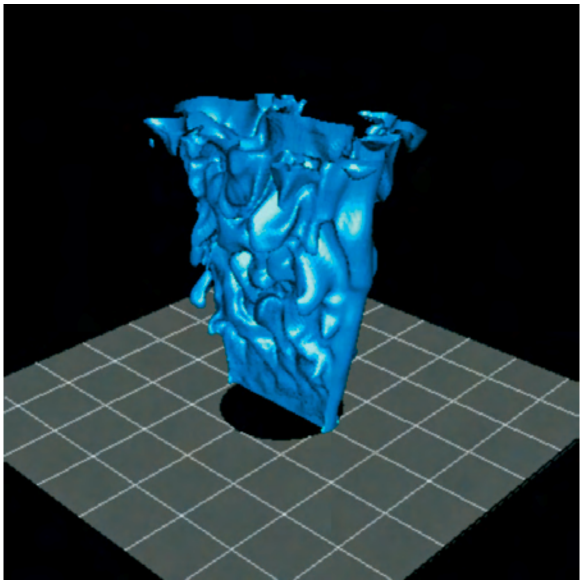


Figure 6. Instantaneous surface of a turbulent premixed rod-stabilized V-flame, from simulation.

The first step in simulations such as these is to specify the computational models that will be used to describe the reacting flows. The essential feature of reacting flows is the set of chemical reactions taking place in the fluid. Besides chemical products, these reactions produce both temperature and pressure changes, which couple to the dynamics of the flow. Thus, an accurate description of the reactions is critical to predicting the properties of the flame, including turbulence. Conversely, it is the fluid flow that transports the reacting chemical species into the reaction zone and transports the products of the reaction and the released energy away from the reaction zone. The location and shape of the reaction zone are determined by a delicate balance of species, energy, and momentum fluxes and are highly sensitive to how these fluxes are specified at the boundaries of the computational domain. Turbulence can wrinkle the reaction zone, giving it much more area than it would have in its laminar state, without turbulence. Hence, incorrect prediction of turbulence intensity may under- or over-represent the extent of reaction.

From first principles, the reactions of molecules are described by the Schrödinger equation, and fluid flow by the Navier-Stokes equations. However, each of these equation sets is too difficult to solve directly for the turbulent premixed rod-stabilized "V" methane flame illustrated in Figure 5. So we must rely on approximations to these equations. These approximations define the computational models used to describe the methane flame. For the current simulations, the set of 84 reactions describing the methane flame involves 21 chemical species (methane, oxygen, water, carbon dioxide, and many trace products and reaction intermediates). The particular version of the compressible Navier-Stokes model employed here allows detailed consideration of convection, diffusion, and expansion effects that shape the reaction; but it is "filtered" to remove sound waves, which pose complications that do not measurably affect combustion in this regime. The selection of models such as these requires context-specific expert judgment. For instance, in a jet turbine simulation, sound waves might represent a non-negligible portion of the energy budget of the problem and would need to be modeled; but it might be valid (and very cost-effective) to use a less intricate chemical reaction mechanism to answer the scientific question of interest.

Development of the chemistry model and validation of its predictions, which was funded in part by DOE's Chemical Sciences Program, is an interesting scientific story that is too long to be told here. Suffice it to say that this process took many years. In some scientific fields, high-fidelity models of many important processes involved in the simulations are not available; and additional scientific research will be needed to achieve a comparable level of confidence.

These computations pictured here were performed on the IBM SP, named "Seaborg," at the National Energy Research Scientific Computing Center, using up to 2048 processors. They are among the most demanding combustion simulations ever performed. However, the ability to do them is not merely a result of improvements in computer speed. Improvements in algorithm technology funded by the DOE Applied Mathematical Sciences Program over the past decade have been instrumental in making these computations feasible. Mathematical analysis is used to reformulate the equations describing the fluid flow so that high-speed acoustic transients are removed analytically, while compressibility effects due to chemical reactions are retained. The mathematical model of the fluid flow is discretized using high-resolution finite difference methods, combined with local adaptive mesh refinement by which regions of the finite difference grid are automatically refined or coarsened to maximize overall computational efficiency. The implementation of this methodology uses an object-oriented, message-passing software framework that handles the complex data distribution and dynamic load balancing needed to effectively exploit modern parallel supercomputers. The data analysis framework used to explore the results of the simulation and create visual images that lead to understanding is based on recent developments in "scripting languages" from computer science. The combination of these algorithmic innovations reduces the computational cost by a factor of 10,000 compared with a standard uniform-grid compressible-flow approach.

Researchers have just begun to explore the data generated in these simulations. Figure 7 presents a comparison between a laboratory measurement and a simulation. The early stages of the analysis of this data indicate that excellent agreement is obtained with observable quantities. Further analyses promise to shed new light on the morphology and structure of these types of flames.

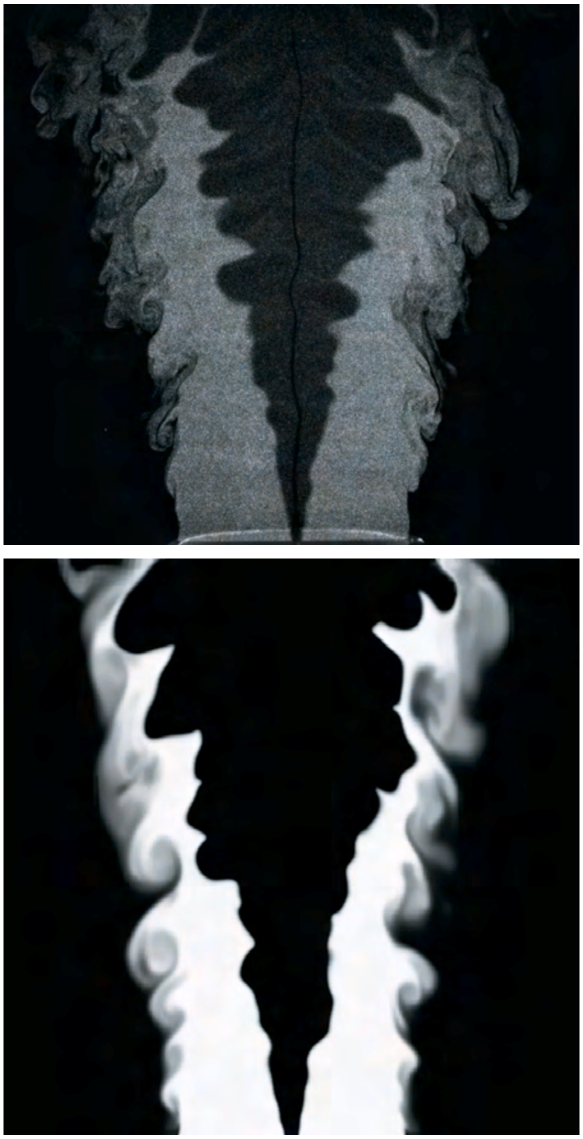


Figure 7. Comparison of experimental particle image velocimetry crossview of the turbulent “V” flame (top) and a simulation (bottom).

This simulation achieves its remarkable resolution of the thin flame edge by means of *adaptive refinement*, which places fine resolution around those features that need it and uses coarse resolution away from such features, saving orders of magnitude in computer memory and processing time. As the reaction zone shifts, the refinement automatically tracks it, adding and removing resolution dynamically. As described earlier, the simulation also employs a mathematical transformation that enables it to skip over tiny time steps that would otherwise be required to resolve sound waves in the flame domain. The four orders of magnitude saved by these algorithmic improvements, compared with previous practices, is a larger factor than the performance gain from parallelization on a computer system that costs tens of millions of dollars! It will be many years before such a high-fidelity simulation can be cost-effectively run using previous practices on *any* computer.

Unfortunately, the adaptivity and mathematical filtering employed to save memory and operations complicate the software and throw the execution of the code into a regime of computation that processes fewer operations per second and uses the thousands of processors of the Seaborg system less uniformly than a “business as usual” algorithm would. As a result, the simulation runs at a small percentage of the theoretical peak rate of the Seaborg machine. However, since it models the phenomenon of interest in much less execution time and delivers it many years earlier than would otherwise be possible, it is difficult to argue against its “scientific efficiency.” In this case, a simplistic efficiency metric like “percentage of theoretical peak” is misleading.

The simulation highlighted earlier is remarkable for its physical fidelity in a complex physics multiscale problem. Experimentalists are easily attracted to fruitful collaborations when simulations reach this threshold of fidelity. Resulting comparisons help refine both experiment and simulation in successive leaps. However, the simulation is equally noteworthy for the combination of the sophisticated mathematical techniques it employs and its parallel implementation on thousands of processors. Breakthroughs such as this have caused scientists to refer to leading-edge large-scale simulation environments as “time machines.” They enable us to pull into the present scientific understanding for which we would otherwise wait years if we were dependent on experiment or on commodity computing capability alone.

The work that led to this simulation recently won recognition for one of its creators, John Bell. He and his colleague Phil Colella, both of Lawrence Berkeley National Laboratory, were awarded the first Prize in Computational Science and Engineering awarded by the Society for Industrial and Applied Mathematics and the Association for Computing Machinery in June 2003. The prize is awarded “in the area of computational science in recognition of outstanding contributions to the development and use of mathematical and computational tools and methods for the solution of science and engineering problems.” Bell and Colella have applied the automated adaptive meshing methodologies they developed in such diverse areas as shock physics, astrophysics, and flow in porous media, as well as turbulent combustion.

The software used in this combustion simulation is currently being extended in many directions under SciDAC. The geometries that it can accurately accommodate are being generalized for accelerator applications. For magnetically confined fusion applications, a particle dynamics module is being added.

Software of this complexity and versatility could never have been assembled in the traditional mode of development, whereby individual researchers with separate concerns asynchronously toss packages written to arbitrary specifications “over the transom.” Behind any simulation as complex as the one discussed in this chapter stands a tightly coordinated, vertically integrated team. This process illustrates the computational science “phase transition” that has found proof of concept under the SciDAC program.

4. OPPORTUNITIES AT THE SCIENTIFIC HORIZON

The availability of computers 100 to 1000 times more powerful than those currently available will have a profound impact on computational scientists' ability to simulate the fundamental physical, chemical, and biological processes that underlie the behavior of natural and engineered systems. Chemists will be able to model the diverse set of molecular processes involved in the combustion of hydrocarbon fuels and the catalytic production of chemicals. Materials scientists will be able to predict the properties of materials from knowledge of their structure, and then, inverting the process, design materials with a targeted set of properties. Physicists will be able to model a broad range of complex phenomena—from the fundamental interactions between elementary particles, to high-energy nuclear processes and particle-electromagnetic field interactions, to the interactions that govern the behavior of fusion plasmas. Biologists will be able to predict the structure and, eventually, the function of the 30–40,000 proteins coded in the human DNA. They will also mine the vast reservoirs of quantitative data being accumulated using high-throughput experimental techniques to obtain new insights into the processes of life.

In this section, we highlight some of the most exciting opportunities for scientific discovery that will be made possible by an integrated set of investments, following the SciDAC paradigm described in Chapter 2, in the programmatic offices in the Office of Science in DOE—Advanced Scientific Computing Research, Basic Energy Sciences, Biological and Environmental Research, Fusion Energy Sciences, High Energy Physics, and Nuclear Physics. In Table 1 we provide a list of the major scientific accomplishments expected in each scientific area from the investments described in this report. The table is followed by an expanded description of one of these accomplishments in each category. The progress described in these paragraphs illustrates the progress that will be made toward solving many other scientific problems and in many other scientific disciplines as the methodologies of computational science are refined through application to these primary Office of Science missions. Volume 2 of this report provides additional information on the many advances that, scientists predict, these investments will make possible.

PREDICTING FUTURE CLIMATES

While existing general circulation models (GCMs) used to simulate climate can provide good estimates of continental- to global-scale climatic variability and change, they are less able to produce estimates of regional changes that are essential for assessing environmental and economic consequences of climatic variability and change. For example, predictions of global average atmospheric temperature changes are of limited utility unless the effects of these changes on regional space-heating needs, rainfall, and growing seasons can be ascertained. As a result of the investments described here, it will be possible to markedly improve both regional predictions of long-term climatic variability and change, and assessments of the potential consequences of such variability and change on natural and managed ecosystems and resources of value to humans (see Figure 8).

Accurate prediction of regional climatic variability and change would have

Table 1. If additional funding becomes available for Office of Science scientific, computer science, and mathematics research programs, as well as for the needed additional computing resources, scientists predict the following major advances in the research programs supported by the Office of Science within the next 5–10 years

Research Programs	Major Scientific Advances
Biological and Environmental Sciences	• Provide global forecasts of Earth's future climate at regional scales using high-resolution, fully coupled and interactive climate, chemistry, land cover, and carbon cycle models. • Develop predictive understanding of subsurface contaminant behavior that provides the basis for scientifically sound and defensible cleanup decisions and remediation strategies. • Establish a mathematical and computational foundation for the study of cellular and molecular systems and use computational modeling and simulation to predict and simulate the behavior of complex microbial systems for use in mission application areas.
Chemical and Materials Sciences	• Provide direct 3-dimensional simulations of a turbulent methane-air jet flame with detailed chemistry and direct 3-dimensional simulations of autoignition of *n*-heptane at high pressure, leading to more-efficient, lower-emission combustion devices. • Develop an understanding of the growth and structural, mechanical, and electronic properties of carbon nanotubes, for such applications as strengtheners of polymers and alloys, emitters for display screens, and conductors and nonlinear circuit elements for nanoelectronics. • Provide simulations of the magnetic properties of nanoparticles and arrays of nanoparticles for use in the design of ultra-high-density magnetic information storage. (Nanomagnetism is the simplest of many important advances in nanoscience that will impact future electronics, tailored magnetic response devices, and even catalysis.) • Resolve the current disparity between theory and experiment for conduction through molecules with attached electronic leads. (This is the very basis of molecular electronics and may well point to opportunities for chemical sensor technology.)
Fusion Energy Sciences	• Improve understanding of fundamental physical phenomena in high-temperature plasmas, including transport of energy and particles, turbulence, global equilibrium and stability, magnetic reconnection, electromagnetic wave/particle interactions, boundary layer effects in plasmas, and plasma/material interactions. • Simulate individual aspects of plasma behavior, such as energy and particle confinement times, high-pressure stability limits in magnetically confined plasmas, efficiency of electromagnetic wave heating and current drive, and heat and particle transport in the edge region of a plasma, for parameters relevant to magnetically-confined fusion plasmas. • Develop a fully integrated capability for predicting the performance of magnetically confined fusion plasmas with high physics fidelity, initially for tokamak configurations and ultimately for a broad range of practical energy-producing magnetic confinement configurations. • Advance the fundamental understanding and predictability of high-energy density plasmas for inertial fusion energy. (Inertial fusion and magnetically confined fusion are complementary technological approaches to unlocking the power of the atomic nucleus.)

Table 1 (continued)

Research Programs	Major Scientific Advances
High Energy Physics	• Establish the limits of the Standard Model of Elementary Particle Physics by achieving a detailed understanding of the effects of strong nuclear interactions in many different processes, so that the equality of Standard Model parameters measured in different experiments can be verified. (Or, if verification fails, signal the discovery of new physical phenomena at extreme sub-nuclear distances.) • Develop realistic simulations of the performance of particle accelerators, the large and complex core scientific instruments of high-energy physics research, both to optimize the design, technology and cost of future accelerators and to use existing accelerators more effectively and efficiently.
Nuclear Physics	• Understand the characteristics of the quark-gluon plasma, especially in the temperature-density region of the phase transition expected from quantum chromodynamics (QCD) and currently sought experimentally. • Obtain a quantitative, predictive understanding of the quark-gluon structure of the nucleon and of interactions of nucleons. • Understand the mechanism of core collapse supernovae and the nature of the nucleosynthesis in these spectacular stellar explosions.

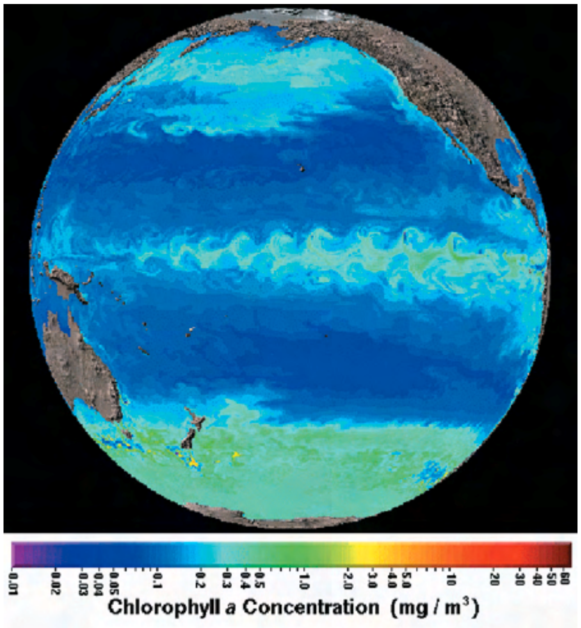


Figure 8. Surface chlorophyll distributions simulated by the Parallel Ocean Program for conditions in late 1996 (a La Niña year) showing intense biological activity across the equatorial Pacific due to upwelling and transport of nutrient-rich cold water. Comparisons with satellite ocean color measurements validate model accuracy and inform our understanding of the regional environmental response to global climate change. [S. Chu, S. Elliott, and M. Maltrud]

tremendous value to society, and that value could be realized if the most significant limitations in the application of GCMs were overcome. The proposed investments would allow climate scientists to

- refine the horizontal resolution of the atmospheric component from 160 to 40 km and the resolution of the ocean component from 105 to 15 km (both improvements are feasible now, but there is insufficient computing power to routinely execute GCMs at these resolutions);
- add interactive atmospheric chemistry, carbon cycle, and dynamic terrestrial vegetation components to simulate global and regional carbon, nitrogen, and sulfur cycles and simulate the effects of land-cover and land-use changes on climate (these cycles and effects are known to be important to climate, but their implementation in GCMs is rudimentary because of computational limitations); and
- improve the representation of important subgrid processes, especially clouds and atmospheric radiative transfer.

Even at 40-km resolution, cloud dynamics are incompletely resolved. Recent advances in the development of cloud sub-models, i.e., two-dimensional and quasi three-dimensional cloud-resolving models, can overcome much of this limitation; but their implementation in global GCMs will require substantially more computing power.

SCIENCE FROM THE NANOSCALE UP

Our ability to characterize small molecules or perfect solids as collections of atoms is very advanced. So is our ability to characterize materials of engineering dimensions. However, our abilities fall far short of what is necessary because many of the properties and processes necessary to develop new materials and optimize existing ones occur with characteristic sizes (or scales) that lie between the atomistic world (less than 1 nanometer) and the world perceived by humans (meters or more), a spatial range of at least a billion. Bridging these different scales, while keeping the essential quantum description at the smallest scale, will require unprecedented theoretical innovation and computational effort, but it is an essential step toward a complete fundamental theory of matter.

The nanoscale, 10–100 nanometers or so—the first step in the hierarchy of scales leading from the atomic scale to macroscopic scales—is also important in its own right because, at the nanoscale, new properties emerge that have no parallel at either larger or smaller scales. Thus, the new world of nanoscience offers tremendous opportunities for developing innovative technological solutions to a broad range of real-world problems using scientists' increasing abilities to manipulate matter at this scale. Theory, modeling, and simulation are essential to developing an understanding of matter at the nanoscale, as well as to developing the new nanotechnologies (see Figure 9). At the nanoscale, experiment often cannot stand alone, requiring sophisticated modeling and simulation to deconvolute experimental results and yield scientific insight. To simulate nanoscale entities, scientists will have to perform *ab initio* atomistic calculations for more atoms than ever before, directly treat the correlated behavior of the electrons associated with those atoms, and model nanoscale systems as open systems rather than closed (isolated) ones. The computational effort required to simulate nanoscale systems far exceeds any computational efforts in materials and molecular science to date by a factor of at least 100 to 1000. The

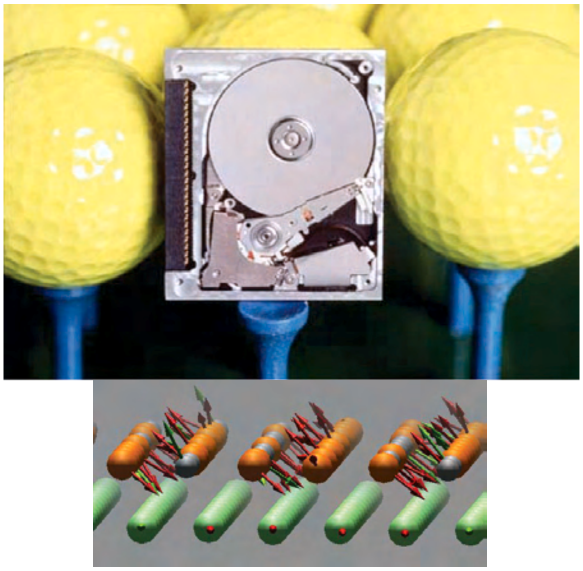


Figure 9. To continue annual doubling of hard drive storage density (top: IBM micro-drive), improved understanding of existing materials and development of new ones are needed. First principles spin dynamics and the use of massively parallel computing resources are enabling researchers at Oak Ridge National Laboratory to understand magnetism at ferromagnetic-antiferromagnetic interfaces (bottom: magnetic moments at a Co/FeMn interface) with the goal of learning how to store binary data in ever smaller domains, resulting in further miniaturization breakthroughs. [M. Stocks]

investments envisioned in this area will make nanoscale simulations possible by advancing the theoretical understanding of nanoscale systems and supporting the development of computational models and software for simulating nanoscale systems and processes, as well as allowing detailed simulations of nanoscale phenomena and their interaction with the macroscopic world.

DESIGNING AND OPTIMIZING FUSION REACTORS

Fusion, the energy source that fuels the sun and stars, is a potentially inexhaustible and environmentally safe source of energy. However, to create the conditions under which hydrogen isotopes undergo nuclear fusion and release energy, high-temperature (100 million degrees centigrade) plasmas—a fluid of ions and electrons—must be produced, sustained, and confined. Plasma confinement is recognized internationally as a grand scientific challenge and a formidable technical task. With the research program and computing resources outlined in this report, it will be possible to develop a fully integrated simulation capability for predicting the performance of magnetic confinement systems, such as tokamaks.

In the past, specialized simulations were used to explore selected key aspects of the physics of magnetically confined plasmas, e.g., turbulence or micro-scale instabilities. Only an integrated model can bring all of the strongly interacting physical processes together to make comprehensive, reliable predictions about the real-world behavior of fusion plasmas. The simulation of an entire magnetic fusion confinement system involves simultaneous modeling of the core and edge plasmas, as well as the plasma-wall inter-actions. In each region of the plasma, turbulence can cause anomalies in the transport of the ionic species in the plasma; there can also be abrupt changes in the form of the plasma caused by large-scale instabilities (see Figure 10). Computational modeling of these key processes requires large-scale simulations covering a broad range of space and time scales. An integrated simulation capability will dramatically enhance the efficient utilization of a burning fusion device, in particular, and the optimization of fusion energy development in general; and it would serve as an intellectual integrator of physics knowledge ranging from advanced tokamaks to innovative confinement concepts.

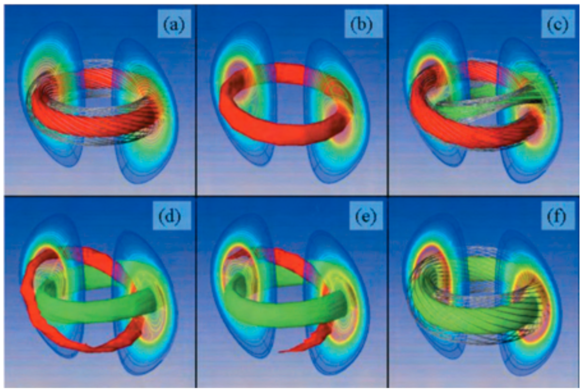


Figure 10. Simulation of the interchange of plasma between the hot (red) and cold (green) regions of a tokamak via magnetic reconnection. [W. Park]

PREDICTING THE PROPERTIES OF MATTER

Fundamental questions in high-energy physics and nuclear physics pivot on our ability to simulate so-called strong interactions on a computational lattice via the theory of quantum chromodynamics (QCD).

The ongoing goal of research in high-energy physics is to find the elementary constituents of matter and to determine how they interact. The current state of our knowledge is summarized by the Standard

Model of Elementary Particle Physics. A central focus of experiments at U.S. high-energy physics facilities is precision testing of the Standard Model. The ultimate aim of this work is to find deviations from this model—a discovery that would require the introduction of new forms of matter or new physical principles to describe matter at the shortest distances. Determining whether or not each new measurement is in accord with the Standard Model requires an accurate evaluation of the effects of the strong interactions on the process under study. These computations are a crucial companion to any experimental program in high-energy physics. QCD is the only known method of systematically reducing all sources of theoretical error.

Lattice calculations lag well behind experiment at present, and the precision of experiments under way or being planned will make the imbalance even worse. The impact of the magnitude of the lattice errors is shown in Figure 11. For the Standard Model to be correct, the parameters ρ and η must lie in the region of overlap of all the colored bands. Experiments measure various combinations of ρ and η, as shown by the colored bands. The reduction of theory errors shown will require tens of teraflop-years of simulations, which will be available through the construction of new types of computers optimized for the problems at hand. A further reduction, when current experiments are complete, will need another order of magnitude of computational power.

A primary goal of the nuclear physics program at the Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Laboratory is the discovery and characterization of the quark–gluon plasma—a state of matter predicted by QCD to exist at high densities and temperatures. To confirm an observation of a new state of matter, it is important to determine the nature of the phase transition between ordinary matter and a quark–gluon plasma, the properties of the plasma, and the equation of state. Numerical simulations of QCD on a lattice have proved to be the only source of *a priori* predictions about this form of matter in the vicinity of the phase transition.

Recent work suggests that a major increase in computing power would result in definitive predictions of the properties of the quark–gluon plasma, providing timely and invaluable input into the RHIC heavy ion program.

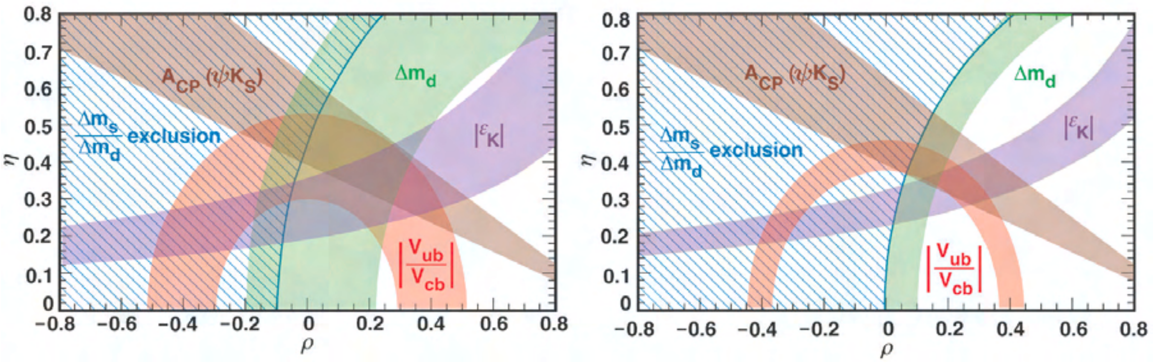


Figure 11. Constraints on the Standard Model parameters *r* and *h*. For the Standard Model to be correct, they must be restricted to the region of overlap of the solidly colored bands. The figure on the left shows the constraints as they exist today. The figure on the right shows the constraints as they would exist with no improvement in the experimental errors, but with lattice gauge theory uncertainties reduced to 3%. [R. Patterson]

Similar lattice calculations are capable of predicting and providing an understanding of the inner structure of the nucleon where the quarks and gluons are confined, another major nuclear physics goal. Precise lattice calculations of nucleon observables can play a pivotal role in helping to guide experiments. This methodology is well understood, but a "typical" calculation of a single observable requires over a year with all the computers presently available to theorists. A thousandfold increase in computing power would make the same result available in about 10 hours. Making the turn-around time for simulations commensurate with that of experiments would lead to major progress in the quantitative, predictive understanding of the structure of the nucleon and of the interactions between nucleons.

BRINGING A STAR TO EARTH

Nuclear astrophysics is a major research area within nuclear physics—in particular, the study of stellar explosions and their element production. Explosions of massive stars, known as core collapse supernovae, are the dominant source of elements in the Periodic Table between oxygen and iron; and they are believed to be responsible for the production of half the heavy elements (heavier than iron). Life in the universe would not exist without these catastrophic stellar deaths.

To understand these explosive events, their byproducts, and the phenomena associated with them will require large-scale, multidimensional, multiphysics simulations (see Figure 12). Current supercomputers are enabling the first detailed two-dimensional simulations. Future petascale computers would provide, for the first time, the opportunity to simulate supernovae realistically in three dimensions.

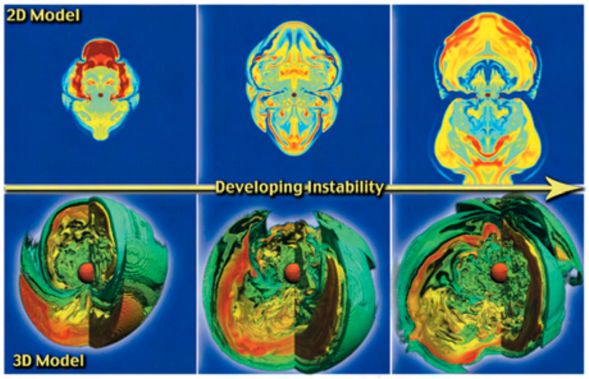


Figure 12. Development of a newly discovered shock wave instability and resulting stellar explosions (supernovae), in two- and three-dimensional simulations. [J. Blondin, A. Mezzacappa, and C. DeMarino; visualization by R. Toedte]

While observations from space are critical, the development of theoretical supernova models will also be grounded in terrestrial experiment. Supernova science has played a significant role in motivating the construction of the Rare Isotope Accelerator (RIA) and the National Underground Laboratory for Science and Engineering (NUSEL). As planned, RIA will perform measurements that will shed light on models for heavy element production in supernovae; and NUSEL will house numerous neutrino experiments, including a half-megaton neutrino detector capable of detecting extra-galactic supernova neutrinos out to Andromeda.

Supernovae have importance beyond their place in the cosmic hierarchy: the extremes of density, temperature, and composition encountered in supernovae provide an opportunity to explore fundamental nuclear and particle physics that would otherwise be inaccessible in terrestrial experiments. Supernovae therefore serve as cosmic laboratories, and computational models of supernovae constitute the bridge between the observations, which bring us information about these explosions, and the fundamental physics we seek.

5. ENABLING MATHEMATICS AND COMPUTER SCIENCE TOOLS

MATHEMATICAL TOOLS FOR LARGE-SCALE SIMULATION

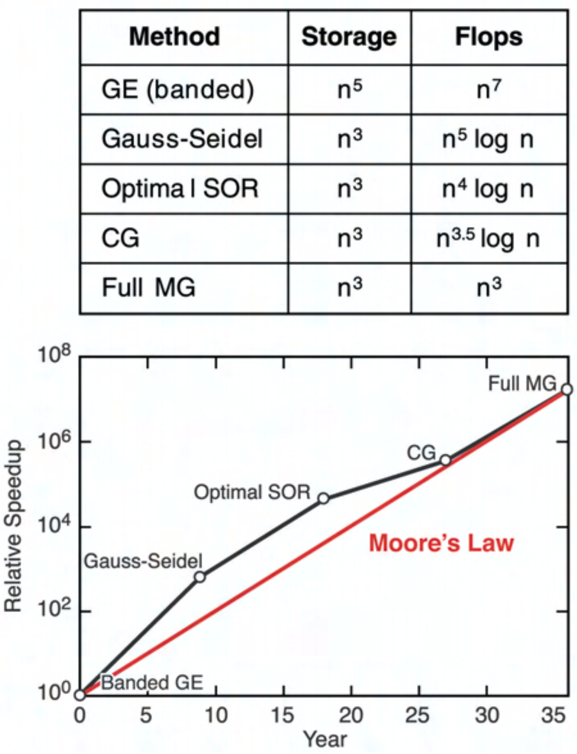
Mathematics is the bridge between physical reality and computer simulations of physical reality. The starting point for a computer simulation is a mathematical model, expressed in terms of a system of equations and based on scientific understanding of the problem being investigated, such as the knowledge of forces acting on a particle or a parcel of fluid. For such a model, there are three types of mathematical tools that may be brought to bear in order to produce and use computer simulations.

- **Model Analysis**. Although the mathematical model is motivated by science, it must stand on its own as an internally consistent mathematical object for its computer representation to be well defined. For that reason, it is necessary to resolve a number of issues regarding the mathematical structure of the model. Do the equations have a unique solution for all physically meaningful inputs? Do small changes in the inputs lead only to small changes in the solution, or, as an indication of potential trouble, could a small uncertainty be magnified by the model into large variability in the outcome?

- **Approximation and Discretization**. For many systems, the number of unknowns in the model is infinite, as when the solution is a function of continuum variables such as space and time. In such cases, it is necessary to approximate the infinite number of unknowns with a large but finite number of unknowns in order to represent it on a computer. The mathematical issues for this process, called “discretization,” include (1) the extent to which the finite approximation better agrees with the solution to the original equations as the number of computational unknowns increases and (2) the relationship between the choice of approximation and qualitative mathematical properties of the solution, such as singularities.

- **Solvers and Software**. Once one has represented the physical problem as the solution to a finite number of equations for a finite number of unknowns, how does one make best use of the computational resources to calculate the solution to those equations? Issues at this stage include the development of optimally efficient algorithms, and the mapping of computations onto a complex hierarchy of processors and memory systems.

Although these three facets represent distinct mathematical disciplines, these disciplines are typically employed in concert to build complete simulations. The choice of appropriate mathematical tools can make or break a simulation code. For example, over a four-decade period of our brief simulation era, algorithms alone have brought a speed increase of a factor of more than a million to computing the electrostatic potential induced by a charge distribution, typical of a computational kernel found in a wide variety of problems in the sciences. The improvement resulting from this algorithmic speedup is comparable to that resulting from the hardware speedup due to Moore’s Law over the same length of time (see Figure 13). The series of algorithmic improvements producing this factor are all based on a fundamental mathematical property of the underlying model—namely, that the function describing electrostatic coupling between disjoint regions in space is very smooth. Expressed in the right way, this coupling can therefore be resolved accurately with little computational effort. The various improvements in algorithms for solving this problem came about by successive redesigns of the discretization methods and solver algorithms to better exploit this mathematical property of the model.



Method	Storage	Flops
GE (banded)	n^5	n^7
Gauss-Seidel	n^3	$n^5 \log n$
Optimal SOR	n^3	$n^4 \log n$
CG	n^3	$n^{3.5} \log n$
Full MG	n^3	n^3

Figure 13. Top: A table of the scaling of memory and processing requirements for the solution of the electrostatic potential equation on a uniform cubic grid of n × n × n cells. Bottom: The relative gains of some solution algorithms for this problem and Moore's Law for improvement of processing rates over the same period (illustrated for the case where n = 64). Algorithms yield a factor comparable to that of the hardware, and the gains typically can be combined (that is, multiplied together). The algorithmic gains become more important than the hardware gains for larger problems. If adaptivity is exploited in the discretization, algorithms may do better still, though combining all of the gains becomes more subtle in this case.

Another trend in computational science is the steady increase in the intellectual complexity of virtually all aspects of the modeling process. The scientists contributing to this report identified and developed (in Volume 2) eight cross-cutting areas of applied mathematics for which research and development is needed to accommodate the rapidly increasing complexity of state-of-the-art computational science. They fall into the following three categories.

- **Managing Model Complexity**. Scientists want to use increasing computing capability to improve the fidelity of their models. For many problems, this means introducing models with more physical effects, more equations, and more unknowns. In ***Multiphysics Modeling***, the goal is to develop a combination of analytical and numerical techniques to better represent problems with multiple physical processes. These techniques may range from analytical methods to determine how to break a problem up into weakly interacting components, to new numerical methods for exploiting such a decomposition of the problem to obtain efficient and accurate discretizations in time. A similar set of issues arises from the fact that many systems of interest have processes that operate on length and time scales that vary over many orders of magnitude. ***Multiscale Modeling*** addresses the representation and interaction of behaviors on multiple scales so that results of interest are recovered without the (unaffordable) expense of representing all behaviors at uniformly fine scales. Approaches include the development of *adaptive methods*, i.e., discretization methods that can represent directly many orders of magnitude in length scales that might appear in a single mathematical model, and *hybrid methods* for coupling radically different models (continuum vs. discrete, or stochastic vs. deterministic), each of which represents the behavior on a different scale. ***Uncertainty Quantification*** addresses issues connected with mathematical models that involve fits to experimental data, or that are derived from heuristics that may not be directly connected to physical principles. Uncertainty quantification uses techniques from fields such as statistics and optimization to determine the sensitivity of models to inputs with errors and to design models to minimize the effect of those errors.

- **Discretizations of Spatial Models**. Many of the applications described in this document have, as core components of their mathematical models, the equations of fluid dynamics or radiation transport, or both. ***Computational Fluid Dynamics*** and ***Transport and Kinetic Methods*** have as their goal the development of the next generation of spatial discretization methods for these problems. Issues include the development of discretization methods that are well-suited for use in multiphysics applications without loss of accuracy or robustness. ***Meshing Methods*** specifically address the process of discretizing the computational domain, itself, into a union of simple elements. Meshing is usually a prerequisite for discretizing the equations defined over the domain. This process includes the management of complex geometrical objects arising in technological devices, as well as in some areas of science, such as biology.

- **Managing Computational Complexity.** Once the mathematical model has been converted into a system of equations for a finite collection of unknowns, it is necessary to solve the equations. The goal of efforts in the area of ***Solvers and "Fast" Algorithms*** is to develop algorithms for solving these systems of equations that balance computational efficiency on hierarchical multiprocessor systems, scalability (the ability to use effectively additional computational resources to solve increasingly larger problems), and robustness (insensitivity of the computational cost to details of the inputs). An algorithm is said to be "fast" if its cost grows, roughly, only proportionally to the size of the problem. This is an ideal algorithmic property that is being obtained for more and more types of equations. ***Discrete Mathematics and Algorithms*** make up a complementary set of tools for managing the computational complexity of the interactions of discrete objects. Such issues arise, for example, in traversing data structures for calculations on unstructured grids, in optimizing resource allocation on multiprocessor architectures, or in scientific problems in areas such as bioinformatics that are posed directly as "combinatorial" problems.

COMPUTER SCIENCE TOOLS FOR LARGE-SCALE SIMULATION

The role of computer science in ultrascale simulation is to provide tools that address the issues of complexity, performance, and understanding. These issues cut across the computer science disciplines. An integrated approach is required to solve the problems faced by applications.

COMPLEXITY

One of the major challenges for applications is the complexity of turning a mathematical model into an effective simulation. There are many reasons for this, as indicated in Figure 4. Often, even after the principles of a model and simulation algorithm are well understood, too much effort still is required to turn this understanding into practice because of the complexity of code design. Current programming models and frameworks do not provide sufficient support to allow domain experts to be shielded from details of data structures and computer architecture. Even after an application code produces correct scientific results, too much effort still is required to obtain high performance. Code tuning efforts needed to match algorithms to current computer architectures require lengthy analysis and experimentation.

Once an application runs effectively, often the next hurdle is saving, accessing, and sharing data. Once the data are stored, since ultrascale simulations often produce ultrascale-sized datasets, it is still too difficult to process, investigate, and visualize the data in order to accomplish the purpose of the simulation—to advance science. These difficulties are compounded by the problems faced in sharing resources, both human and computer hardware.

Despite this grim picture, prospects for placing usable computing environments into the hands of scientific domain experts are improving. In the last few years, there has been a growing understanding of the problems of managing complexity in computer science, and therefore of their potential solutions. For example, there is a deeper understanding of how to make programming languages expressive and easy to use without sacrificing high performance on the sophisticated, adaptive algorithms.

Another example is the success of component-oriented software in some application domains; such “components” have allowed computational scientists to focus their own expertise on their science while exploiting the newest algorithmic developments. Many groups in high-performance computing have tackled these issues with significant leadership from DOE. Fully integrated efforts are required to produce a qualitative change in the way application groups cope with the complexity of designing, building, and using ultrascale simulation codes.

PERFORMANCE

One of the drivers of software complexity is the premium on performance. The most obvious aspect of the performance problem is the performance of the computer hardware. Although there have been astounding gains in arithmetic processing rates over the last five decades, users often receive only a small fraction of the theoretical peak of processing performance. There is a perception that this fraction is, in fact, declining. This perception is correct in some respects. For many applications, the reason for a declining percentage of peak performance is the relative imbalance in the performance of the subsystems of high-end computers. While the raw performance of commodity processors has followed Moore’s Law and doubled every 18 months, the performance of other critical parts of the system, such as memory and interconnect, have improved much less rapidly, leading to less-balanced overall systems. Solving this problem requires attention to system-scale architectural issues.

As with code complexity issues, there are multiple on-going efforts to address hardware architecture issues. Different architectural solutions may be required for different algorithms and applications. A single architectural convergence point, such as that occupied by current commodity-based terascale systems, may not be the most cost-effective solution for all users. A comprehensive simulation program requires that several candidate architectural approaches receive sustained support to explore their promise.

Performance is a cross-cutting issue, and computer science offers automated approaches to developing codes in ways that allow computational scientists to concentrate on their science. For example, techniques that allow a programmer to automatically generate code for an application that is tuned to a specific computer architecture address both the issues of managing the complexity of highly-tuned code and the problem of providing effective portability between high-performance computing platforms. Such techniques begin with separate analyses of the “signature” of the application (e.g., the patterns of local memory accesses and inter-processor communication) and the parameters of the hardware (e.g., cache sizes, latencies, bandwidths). There is usually plenty of algorithmic freedom in scheduling and ordering operations and exchanges while preserving correctness. This freedom should not be arbitrarily limited by particular expression in a low-level language, but rather chosen close to runtime to best match a given application–architecture pair. Similarly, the performance of I/O and dataset operations can be improved significantly through the use of well-designed and adaptive algorithms.

UNDERSTANDING

Computer science also addresses the issue of understanding the results of a computation. Ultrascale datasets are too large to be grasped directly. Applications that generate such sets already currently rely on a variety of tools to attempt to extract patterns and features from the data. Computer science offers techniques in data management and understanding that can be used to explore data sets, searching for particular patterns. Visualization techniques help scientists explore their data, taking advantage of the unique human visual cortex and visually-stimulated human insight. Current efforts in this area are often limited by the lack of resources, in terms of both staffing and hardware.

Understanding requires harnessing the skills of many scientists. Collaboration technologies help scientists at different institutions work together. Grid technologies that simplify data sharing and provide access to both experimental and ultrascale computing facilities allow computational scientists to work together to solve the most difficult problems facing the nation today. Although these technologies have been demonstrated, much work remains to make them a part of every scientist’s toolbox. Key challenges are in scheduling multiple resources and in data security.

In Volume 2 of this report, scientists from many disciplines discuss critical computer science technologies needed across applications. ***Visual Data Exploration and Analysis*** considers understanding the results of a simulation through visualization. ***Computer Architecture*** looks at the systems on which ultrascale simulations run and for which programming environments and algorithms (described in the section on computational mathematics) must provide software tools. ***Programming Models and Component Technology for High-End Computing*** discusses new techniques for turning algorithms into efficient, maintainable programs. ***Access and Resource Sharing*** looks at how to both make resources available to the entire research community and promote collaboration. ***Software Engineering and Management*** is about disciplines and tools—many adapted from the industrial software world—to manage the complexity of code development. ***Data Management and Analysis*** addresses the challenges of managing and understanding the increasingly staggering volumes of data produced by ultrascale simulations. ***Performance Engineering*** is the art of achieving high performance, which has the potential of becoming a science. ***System Software*** considers the basic tools that support all other software.

The universality of these issues calls for focused research campaigns. At the same time, the inter-relatedness of these issues to one another and to the application and algorithmic issues above them in the simulation hierarchy points to the potential synergy of integrated simulation efforts.

6. RECOMMENDATIONS AND DISCUSSION

The preceding chapters of this volume have presented a science-based case for large-scale simulation. Chapter 2 presents an on-going successful initiative, SciDAC—a collection of 51 projects operating at a level of approximately $57 million per year—as a template for a much-expanded investment in this area. To fully realize the promises of SciDAC, even at its present size, requires immediate new investment in computer facilities. Overall, the computational science research enterprise could make effective use of as much as an order of magnitude increase in annual investment at this time. Funding at that level would enable an immediate tenfold increase in computer cycles available and a doubling or tripling of the researchers in all contributing disciplines. A longer, 5-year view should allow for another tenfold increase in computing power (at less than a tenfold increase in expense). Meanwhile, the emerging workforce should be primed now for expansion by attracting and developing new talent.

Chapter 3 describes the anatomy of a state-of-the-art large-scale simulation. A simulation such as this one goes beyond established practices by exploiting advances from many areas of computational science to marry raw computational power to algorithmic intelligence. It conveys the multidisciplinary spirit intended in the expanded initiative. Chapter 4 lists some compelling scientific opportunities that are just over the horizon—new scientific advances that are predicted to be accessible with an increase in computational power of a hundred- to a thousand-fold. Chapter 5 lists some advances in the enabling technologies of computational mathematics and computer science that will be required or that could, if developed for widespread use, reduce the computational power required to achieve a given scientific simulation objective. The companion volume details these opportunities more extensively. This concluding section revisits each of the recommendations listed at the end of Chapter 1 and expands upon them with additional clarifying discussion.

1. Major new investments in computational science are needed in all of the mission areas of DOE's Office of Science, as well as those of many other agencies, so that the United States may be the first, or among the first, to capture the new opportunities presented by the continuing advances in computing power. Such investments will extend the important scientific opportunities that have been attained by a fusion of sustained advances in scientific models, mathematical algorithms, computer architecture, and scientific software engineering.

The United States is an acknowledged international leader in computational science, both in its national laboratories and in its research universities. However, the United States could lose its leadership position in a short time because the opportunities represented by computational science for economic, military, and other types of leadership are well recognized; the underlying technology and training are available without restrictions; and the technology continues to evolve rapidly. As Professor Jack Dongarra from the University of Tennessee recently noted:

> The rising tide of change [resulting from advances in information technology] shows no respect for the established order. Those who are unwilling or unable to adapt in response to this profound movement not only lose access to the opportunities that the information technology revolution is creating, they risk being rendered obsolete by smarter, more agile, or more daring competitors.

The four streams of development brought together in a “perfect fusion” to create the current opportunities are of different ages, but their confluence is recent. Modern scientific modeling still owes as much to Newton’s *Principia* (1686) as to anything since, although new models are continually proposed and tested. Modern simulation and attention to floating point arithmetic have been pursued with intensity since the pioneering work of Von Neumann at the end of World War II. Computational scientists have been riding the technological curve subsequently identified as “Moore’s Law” for nearly three decades now. However, scientific software engineering as a discipline focused on componentization, extensibility, reusability, and platform portability within the realm of high-performance computing is scarcely a decade old. Multidisciplinary teams that are savvy with respect to all four aspects of computational simulation, and that are actively updating their codes with optimal results from each aspect, have evolved only recently. Not every contemporary “community code” satisfies the standard envisioned for this new kind of computational science, but the United States is host to a number of the leading efforts and should capitalize upon them.

2. Multidisciplinary teams, with carefully selected leadership, should be assembled to provide the broad range of expertise needed to address the intellectual challenges associated with translating advances in science, mathematics, and computer science into simulations that can take full advantage of advanced computers.

Part of the genius of the SciDAC initiative is the cross-linking of accountabilities between applications groups and the groups providing enabling technologies. This accountability implies, for instance, that physicists are concentrating on physics, not on developing parallel libraries for core mathematical operations; and it implies that mathematicians are turning their attention to methods directly related to application-specific difficulties, moving beyond the model problems that motivated earlier generations. Another aspect of SciDAC is the cross-linking of laboratory and academic investigators. Professors and their graduate students can inexpensively explore many promising research directions in parallel; however, the codes they produce rarely have much shelf life. On the other hand, laboratory-based scientists can effectively interact with many academic partners, testing, absorbing, and seeding numerous research ideas, and effectively “hardening” the best of the ideas into long-lived projects and tested, maintained codes. This complementarity should be built into future computational science initiatives.

Future initiatives can go farther than SciDAC by adding self-contained, vertically integrated teams to the existing SciDAC structure of specialist groups. The SciDAC Integrated Software Infrastructure Centers (ISICs), for instance, should continue to be supported, and even enhanced, to develop infrastructure that supports diverse science portfolios. However, mathematicians and computer scientists capable of understanding and harnessing the output of these ISICs in specific scientific areas should also be bona fide members of the next generation of computational science teams recommended by this report, just as they are now part of a few of the larger SciDAC applications teams.

3. Extensive investment in new computational facilities is strongly recommended, since simulation now cost-effectively complements experimentation in the pursuit of the answers to numerous scientific questions. New facilities should strike a balance between capability computing for those "heroic simulations" that cannot be performed any other way, and capacity computing for "production" simulations that contribute to the steady stream of progress.

The topical breakout groups assessing opportunities in the sciences at the June 2003 SCaLeS Workshop have universally pleaded for more computational resources dedicated to their use. These pleas are quantified in Volume 2 and highlighted in Chapter 4 of this volume, together with (partial) lists of the scientific investigations that can be undertaken with current know-how and an immediate influx of computing resources.

It is fiscally impossible to meet all of these needs, nor does this report recommend that available resources be spent entirely on the computational technology existing in a given year. Rather, steady investments should be made over evolving generations of computer technology, since each succeeding generation packages more capability per dollar and evolves toward greater usability.

To the extent that the needs for computing resources go unmet, there will remain a driving incentive to get more science done with fewer cycles. This is a healthy incentive for multidisciplinary solutions that involve increasingly optimal algorithms and software solutions. However, this incentive exists independently of the desperate need for computing resources. U.S. investigators are shockingly short on cycles to capitalize on their current know-how. Some workshop participants expressed a need for very large facilities in which to perform "heroic" simulations to provide high-fidelity descriptions of the phenomena of interest, or to establish the validity of a computational model. With current facilities providing peak processing rates in the tens of teraflop/s, a hundred- to thousand-fold improvement in processing rate would imply a petaflop/s machine. Other participants noted the need to perform ensembles of simulations of more modest sizes, to develop quantitative statistics, to perform parameter identifications, or to optimize designs. A carefully balanced set of investments in capability and capacity computing will be required to meet these needs.

Since the most cost-effective cycles for the capacity requirements are not necessarily the same as the most cost-effective cycles for the capability requirements, a mix of procurements is natural and optimal. In the short term, we recommend pursuing, for use by unclassified scientific applications, at least one machine capable of sustaining one petaflop/s, plus the capacity for another sustained petaflop/s distributed over tens of installations nationwide.

In earlier, more innocent times, recommendations such as these could have been justified on the principle of "build it and they will come," with the hope of encouraging the development of a larger computational science activity carrying certain anticipated benefits. In the current atmosphere, the audience for this hardware already exists in large numbers; and the operative principle is, instead, "build it or they will go!" Two plenary speakers at the workshop noted that a major U.S.-based computational geophysicist has already taken his leading research program to Japan, where he has been made a guest of the Earth Simulator facility; this facility offers him a computational capability that he cannot find in the United States. Because of the commissioning of the Japanese Earth Simulator last spring, this problem is most acute in earth science. However, the problem will rapidly spread to other disciplines.

4. Investment in hardware facilities should be accompanied by sustained collateral investment in software infrastructure for them. The efficient use of expensive computational facilities and the data they produce depends directly upon multiple layers of system software and scientific software, which, together with the hardware, are the engines of scientific discovery across a broad portfolio of scientific applications.

The President's Information Technology Advisory Committee (PITAC) report of 1999 focused on the tendency to under-invest in software in many sectors, and this is unfortunately also true in computational science. Vendors have sometimes delivered leading-edge computing platforms delivered with immature or incomplete software bases; and computational science users, as well as third-party vendors, have had to rise to fill this gap. In addition, scientific applications groups have often found it difficult to make effective use of these computing platforms from day one, and revising their software for the new machine can take several months to a year or more. Many industry-standard libraries of systems software and scientific software originated and are maintained at national laboratories, a situation that is a natural adaptation to the paucity of scientific focus in the commercial marketplace.

The SciDAC initiative has identified four areas of scientific software technology for focused investment—scientific simulation codes, mathematical software, computing systems software, and collaboratory software. Support for development and maintenance of portable, extensible software is yet another aspect of the genius of the SciDAC initiative, as the reward structure for these vital tasks is otherwise absent in university-based research and difficult to defend over the long term in laboratory-based research. Although SciDAC is an excellent start, the current funding level cannot support many types of software that computational science users require.

It will be necessary and cost-effective, for the foreseeable future, for the agencies that sponsor computational science to make balanced investments in several of the layers of software that come between the scientific problems they face and the computer hardware they own.

5. Additional investments in hardware facilities and software infrastructure should be accompanied by sustained collateral investments in algorithm research and theoretical development. Improvements in basic theory and algorithms have contributed as much to increases in computational simulation capability as improvements in hardware and software over the first six decades of scientific computing.

A recurring theme among the reports from the topical breakout groups for scientific applications and mathematical methods is that increases in computational capability comparable to those coming directly from Moore's Law have come from improved models and algorithms. This point was noted over 10 years ago in the famous "Figure 4" of a booklet entitled *Grand Challenges: High-Performance Computing and Communications* published by FCCSET and distributed as a supplement of the President's FY 1992 budget. A contemporary reworking of this chart appears as Figure 13 of this report. It shows that the same factor of 16 million in performance improvement that would be accumulated by Moore's Law in a 36-year span is achieved by improvements in numerical analysis for solving the electrostatic potential equation on a cube over a similar period, stretching from the demonstration that Gaussian elimination was feasible in the limited arithmetic of digital computers to the publication of the so-called "full multigrid" algorithm.

There are numerous reasons to believe that advances in applied and computational mathematics will make it possible to model a range of spatial and temporal scales much wider than those reachable by established practice. This has already been demonstrated with automated adaptive meshing, although the latter still requires too much specialized expertise to have achieved universal use. Furthermore, multiscale theory can be used to communicate the net effect of processes occurring on unresolvable scales to models that are discretized on resolvable scales. Multiphysics techniques may permit codes to "step over" stability-limiting time scales that are dynamically irrelevant at larger scales.

Even for models using fully resolved scales, research in algorithms promises to transform methods whose costs increase rapidly with increasing grid resolution into methods that increase linearly or log-linearly with complexity, with reliable bounds on the approximations that are needed to achieve these simulation-liberating gains in performance. It is essential to keep computational science codes well stoked with the latest algorithmic technology available. In turn, computational science applications must actively drive research in algorithms.

6. Computational scientists of all types should be proactively recruited with improved reward structures and opportunities as early as possible in the educational process so that the number of trained computational science professionals is sufficient to meet present and future demands.

The United States is exemplary in integrating its computational science graduate students into internships in the national laboratory system, and it has had the foresight to establish a variety of fellowship programs to make the relatively new career of "computational scientist" visible outside departmental boundaries at universities. These internships and fellowship programs are estimated to be under-funded, relative to the number of well-qualified students, by a factor of 3. Even if all presently qualified and interested U.S. citizens who are doctoral candidates could be supported—and even if this pool were doubled by aggressive recruitment among groups in the native-born population who are underrepresented in science and engineering, and / or the addition of foreign-born and domestically trained graduate students—the range of newly available scientific opportunities would only begin to be grazed in new computational science doctoral dissertations.

The need to dedicate attention to earlier stages of the academic pipeline is well documented elsewhere and is not the central subject of this focused report, although it is a related concern. It is time to thoroughly assess the role of computational simulation in the undergraduate curriculum. Today's students of science and engineering, whose careers will extend to the middle of the twenty-first century, must understand the foundations of computational simulation, must be aware of the advantages and limitations of this approach to scientific inquiry, and must know how to use computational techniques to help solve scientific and engineering problems. Few universities can make this claim of their current graduates. Academic programs in computational simulation can be attractors for undergraduates who are increasingly computer savvy but presently perceive more exciting career opportunities in non-scientific applications of information technology.

7. Sustained investments must be made in network infrastructure for access and resource sharing, as well as in the software needed to support collaboration among distributed teams of scientists, recognizing that the best possible computational science teams will be widely separated geographically and that researchers will generally not be co-located with facilities and data.

The topical breakout group on access and resource sharing addressed the question of what technology must be developed and / or deployed to make a petaflop / s-scale computer useful to users not located in the same building. Resource sharing constitutes a multilayered research endeavor that is presently accorded approximately one-third of the resources available to SciDAC, which are distributed over ten projects. Needs addressed include network infrastructure, grid / networking middleware, and application-specific remote user environments.

The combination of central facilities and a widely distributed scientific workforce requires, at a minimum, remote access to computation and data, as well as the integration of data distributed over several sites.

Network infrastructure research is also recommended for its promises of leading-edge capabilities that will revolutionize computational science, including environments for coupling simulation and experiment; environments for multidisciplinary simulation; orchestration of multidisciplinary, multiscale, end-to-end science process; and collaboration in support of all these activities.

8. Federal investment in innovative, high-risk computer architectures that are well suited to scientific and engineering simulations is both appropriate and needed to complement commercial research and development. The commercial computing marketplace is no longer effectively driven by the needs of computational science.

Scientific computing—which ushered in the information revolution over the past half-century by demanding digital products that eventually had successful commercial spin-offs—has now virtually been orphaned by the information technology industry. Scientific computing occupies only a few percentage points of the total marketplace revenue volume of information technology. This stark economic fact is a reminder of one of the socially beneficial by-products of computational science. It is also a wake-up call regarding the necessity that computational science attend to its own needs vis-à-vis future computer hardware requirements. The most noteworthy neglect of the scientific computing industry is, arguably, the deteriorating ratio of memory bandwidth between processors and main memory to the computational speed of the processors. It does not matter how fast a processor is if it cannot get the data that it needs from memory fast enough—a fact recognized very early by Seymour Cray, whom many regard as the father of the supercomputer. Most commercial applications do not demand high memory bandwidth; many scientific applications do.

There has been a reticence to accord computational science the same consideration given other types of experimental science when it comes to federal assistance in the development of facilities. High-end physics facilities—such as accelerators, colliders, lasers, and telescopes—are, understandably, designed, constructed, and installed entirely at taxpayer expense. However, the design and fabrication of high-end computational facilities are expected to occur in the course of profit-making enterprises; only the science-specific purchase or lease and installation are covered at taxpayer expense. To the extent that the marketplace automatically takes care of the needs of the computational science community, this represents a wonderful savings for the taxpayer. To the extent that it does not, it stunts the opportunity to accomplish important scientific objectives and to lead the U.S. computer industry to further innovation and greatness.

The need to dedicate attention to research in advanced computer architecture is documented elsewhere in many places, including the concurrent interagency High-End Computing Revitalization Task Force (HECRTF) study, and is not the central concern of this application-focused report. However, it is appropriate to reinforce this recommendation, which is as true of our era as it was true of the era of the “Lax Report” (see the discussion immediately following). The SciDAC report called for the establishment of experimental computing facilities whose mission was to “provide critical experimental equipment for computer science and applied mathematics researchers as well as computational science pioneers to assess the potential of new computer systems and architectures for scientific computing.” Coupling these facilities with university-based research projects could be a very effective means to fill the architecture research gap.

CLOSING REMARKS

The contributors to this report would like to note that they claim little original insight for the eight recommendations presented. In fact, our last six recommendations may be found, slightly regrouped, in the four prescient recommendations of the Lax Report (*Large-Scale Computing in Science and Engineering*, National Science Board, 1982), which was written over two decades ago, in very different times, before the introduction of pervasive personal computing and well before the emergence of the World Wide Web!

Our first two recommendations reflect the beginnings of a “phase transition” in the way certain aspects of computational science are performed, from solo investigators to multidisciplinary groups. Most members of the Lax panel—Nobelists and members of the National Academies included—wrote their own codes, perhaps in their entirety, and were capable of understanding nearly every step of their compilation, execution, and interpretation. This is no longer possible: the scientific applications, computer hardware, and computing systems software are now each so complex that combining all three into today's engines of scientific discovery requires the talents of many. The phase transition to multidisciplinary groups, though not first crystallized by the SciDAC report (*Scientific Discovery through Advanced Computing*, U.S. Department of Energy, March 2000), was recognized and eloquently and persuasively articulated there.

The original aspect of the current report is not, therefore, its recommendations. It is, rather, its elaboration in the companion volume of what the nation's leading computational scientists see just over the computational horizon, and the descriptions by cross-cutting enabling technologists from mathematics and computer science of how to reach these new scientific vistas.

It is encouraging that the general recommendations of generations of reports such as this one, which indicate a desired direction along a path of progress, are *relatively constant*. They certainly adapt to local changes in the scientific terrain, but they do not swing wildly in response to the siren calls of academic fashion, the international politics of information technology, or the economics of science funding. Meanwhile, the supporting arguments for the recommendations, which indicate the distance traveled along this path, *have changed dramatically*. As the companion volume illustrates, the two decades since the Lax Report have been extremely exciting and fruitful for computational science. The next 10 to 20 years will see computational science firmly embedded in the fabric of science—the most profound development in the scientific method in over three centuries!

The constancy of direction provides reassurance that the community is well guided, while the dramatic advances in computational science indicate that it is well paced and making solid progress. The Lax Report measured needs in megaflop/s, the current report in teraflop/s—a unit one million times larger. Neither report, however, neglects to emphasize that it is not the instruction cycles per second that measure the quality of the computational effort, but the product of this rating times the scientific results per instruction cycle. The latter requires better computational models, better simulation algorithms, and better software implementation.

POSTSCRIPT AND ACKNOWLEDGMENTS

This report is built on the efforts of a community of computational scientists, DOE program administrators, and technical and logistical professionals situated throughout the country. Bringing together so many talented individuals from so many diverse endeavors was stimulating. Doing so and reaching closure in less than four months, mainly summer months, was nothing short of exhilarating.

The participants were reminded often during the planning and execution of this report that the process of creating it was as important as the product itself. If resources for a multidisciplinary initiative in scientific simulation were to be presented to the scientists, mathematicians, and computer scientists required for its successful execution—as individuals distributed across universities and government and industrial laboratories, lacking familiarity with each other's work across disciplinary and sectorial boundaries—the return to the taxpayer certainly would be positive, yet short of its full potential. Investments in science are traditionally made this way, and in this way, sound scientific building blocks are accumulated and tested. But as a pile of blocks does not constitute a house, theory and models from science, methods and algorithms from mathematics, and software and hardware systems from computer science do not constitute a scientific simulation. The scientists, mathematicians, and computer scientists who met at their own initiative to build this report are now ready to build greater things. While advancing science through simulation, the greatest thing they will build is the next generation of computational scientists, whose training in multidisciplinary environments will enable them to see much further and live up to the expectations of Galileo, invoked in the quotation on the title page of this volume.

The three plenary speakers for the workshop during which the input for this report primarily was gathered—Raymond Orbach, Peter Lax, and John Grosh—perfectly set the perspective of legacy, opportunity, and practical challenges for large-scale simulation. In many ways, the challenges we face come from the same directions as those reported by the National Science Board panel chaired by Peter Lax in 1982. However, progress along the path in the past two decades is astounding—literally factors of millions in computer capabilities (storage capacity, processor speed) and further factors of millions from working smarter (better models, optimal adaptive algorithms). Without the measurable progress of which the speakers reminded us, the aggressive extrapolations of this report might be less credible. With each order-of-magnitude increase in capabilities comes increased ability to understand or control aspects of the universe and to mitigate or improve our lot in it.

The DOE staff members who facilitated this report often put in the same unusual hours as the scientific participants in breakfast meetings, late-evening discussions, and long teleconferences. They did not steer the outcome, but they shared their knowledge of how to get things accomplished in projects that involve hundreds of people; and they were incredibly effective. Many provided assistance along the way, but Dan Hitchcock and Chuck Romine, in particular, labored with the scientific core committee for four months.

Other specialists have left their imprints on this report. Devon Streit of DOE Headquarters is a discussion leader nonpareil. Joanne Stover and Mark Bayless of Pacific Northwest National Laboratory built and maintained web pages, often updated daily, that anchored workshop communication among hundreds of people. Vanessa Jones of Oak Ridge National Laboratory's Washington office provided cheerful hospitality to the core planners. Jody Shumpert of Oak Ridge Institute for Science and Education (ORISE) single-handedly did the work of a conference team in preparing the workshop and a pre-workshop meeting in Washington, D.C. The necessity to shoehorn late-registering participants into Arlington hotels during high convention season, when the response to the workshop outstripped planning expectations, only enhanced her display of unflappability. Rachel Smith and Bruce Warford, also of ORISE, kept participants well directed and well equipped in ten-way parallel breakouts. Listed last, but of first importance for readers of this report, members of the Creative Media group at Oak Ridge National Laboratory iterated tirelessly with the editors in meeting production deadlines and creating a presentation that outplays the content.

The editor-in-chief gratefully acknowledges departmental chairs and administrators at two different universities who freed his schedule of standard obligations and thus directly subsidized this report. This project consumed his last weeks at Old Dominion University and his first weeks at Columbia University and definitely took his mind off of the transition.

All who have labored to compress their (or their colleagues') scientific labors of love into a few paragraphs, or to express themselves with minimal use of equations and jargon, will find ten additional readers for each co-specialist colleague they thereby de-targeted. All involved in the preparation of this report understand that it is a snapshot of opportunities that will soon be outdated in their particulars, and that it may prove either optimistic or pessimistic in its extrapolations. None of these out-comes detracts from the confidence of its contributors that it captures the current state of scientific simulation at a historic inflection point and is worth every sacrifice and risk.

APPENDIX 1: A BRIEF CHRONOLOGY OF “A SCIENCE-BASED CASE FOR LARGE-SCALE SIMULATION”

Under the auspices of the Office of Science of the U.S. Department of Energy (DOE), a group of 12 computational scientists from across the DOE laboratory complex met on April 23, 2003, with 6 administrators from DOE’s Washington, D.C., offices and David Keyes, chairman, to work out the organization of a report to demonstrate the case for increased investment in computational simulation. They also planned a workshop to be held in June to gather input from the computational science community to be used to assemble the report. The group was responding to a charge given three weeks earlier by Dr. Walt Polansky, the Chief Information Officer of the Office of Science (see Appendix 2). The working acronym “SCaLeS” was chosen for this initiative.

From the April meeting, a slate of group leaders for 27 breakout sessions (11 in applications, 8 in mathematical methods, and 8 in computer science and infrastructure) was assembled and recruited. The short notice of the chosen dates in June meant that two important sub-communities, devoted to climate research and to Grid infrastructure, could not avoid conflicts with their own meetings. They agreed to conduct their breakout sessions off-site at the same time as the main workshop meeting. Thus 25 of the breakout groups would meet in greater Washington, D.C., and 2 would meet elsewhere on the same dates and participate synchronously in the workshop information flows.

Acting in consultation with their scientific peers, the group leaders drew up lists of suggested invitees to the workshop so as to guarantee a critical mass in each breakout group, in the context of a fully open workshop. Group leaders also typically recruited one junior investigator “scribe” for each breakout session. This arrangement was designed both to facilitate the assembly of quick-look plenary reports from breakouts at the meeting and to provide a unique educational experience to members of the newest generation of DOE-oriented computational scientists.

Group leaders, invitees, and interested members of the community were then invited to register at the SCaLeS workshop website, which was maintained as a service to the community by Pacific Northwest National Laboratory.

A core planning group of approximately a dozen people, including Daniel Hitchcock and Charles Romine of the DOE Office of Advanced Scientific Computing Research, participated in weekly teleconferences to attend to logistics and to organize the individual breakout sessions at the workshop in a way that guaranteed multidisciplinary viewpoints in each discussion.

More than 280 computer scientists, computational mathematicians, and computational scientists attended the June 24 and 25, 2003, workshop in Arlington, Virginia, while approximately 25 others met remotely. The workshop consisted of three plenary talks and three sessions of topical breakout groups, followed by plenary summary reports from each session, and discussion periods. Approximately 50% of the participant-contributors were from DOE laboratories, approximately 35% from universities, and the remainder from other agencies or from industry.

In the plenary talks, Raymond Orbach, director of the DOE Office of Science, charged the participants to document the scientific opportunities at the frontier of simulation. He also reviewed other Office of Science investments in simulation, including a new program at the National Energy Research Scientific Computing Center that allocates time on the 10 Tflop/s Seaborg machine for grand challenge runs. Peter D. Lax of the Courant Institute, editor of the 1982 report *Large-Scale Computing in Science and Engineering*, provided a personal historical perspective on the impact of that earlier report and advised workshop participants that they have a similar calling in a new computational era. To further set the stage for SCaLeS, John Grosh, co-chair of the High-End Computing Revitalization Task Force (HECRTF), addressed the participants on the context of this interagency task force, which had met the preceding week. (While the SCaLeS workshop emphasized applications, the HECRTF workshop emphasized computer hardware and software.) Finally, Devon Streit of DOE’s Washington staff advised the participants on how to preserve their focus in time-constrained multidisciplinary discussion groups so as to arrive at useful research roadmaps.

From the brief breakout group plenary reports at the SCaLeS workshop, a total of 53 co-authors drafted 27 chapters documenting scientific opportunities through simulation and how to surmount barriers to realizing them through collaboration among scientists, mathematicians, and computer scientists. Written under an extremely tight deadline, these chapters were handed off to an editorial team consisting of Thom Dunning, Phil Colella, William Gropp, and David Keyes. This team likewise worked under an extremely tight schedule, iterating with authors to encourage and enforce a measure of consistency while allowing for variations that reflect the intrinsically different natures of simulation throughout science.

Following a heroic job of technical production by Creative Media at Oak Ridge National Laboratory, *A Science-Based Case for Large-Scale Simulation* is now in the hands of the public that is expected to be the ultimate beneficiary not merely of an interesting report but of the scientific gains toward which it points.

APPENDIX 2: CHARGE FOR “A SCIENCE-BASED CASE FOR LARGE-SCALE SIMULATION”

Date: Wed, 2 Apr 2003 09:49:30 -0500
From: “Polansky, Walt” <Walt.Polansky@science.doe.gov>
Subject: Computational sciences—Exploration of New Directions

Distribution:

I am pleased to report that David Keyes (keyes@cs.odu.edu), of Old Dominion University, has agreed to lead a broad-based effort on behalf of the Office of Science to identify rich and fruitful directions for the computational sciences from the perspective of scientific and engineering applications. One early, expected outcome from this effort will be a strong science case for an ultra-scale simulation capability for the Office of Science.

David will be organizing a workshop (early summer 2003, Washington, D.C. metro area) to formally launch this effort. I expect this workshop will address the major opportunities and challenges facing computational sciences in areas of strategic importance to the Office of Science. I envision a report from this workshop by July 30, 2003. Furthermore, this workshop should foster additional workshops, meetings, and discussions on specific topics that can be identified and analyzed over the course of the next year.

Please mention this undertaking to your colleagues. I am looking forward to their support and their participation, along with researchers from academia and industry, in making this effort a success.

Thanks.

Walt P.

ACRONYMS AND ABBREVIATIONS

ASCI	Accelerated Strategic Computing Initiative
ASCR	Office of Advanced Scientific Computing Research (DOE)
BER	Office of Biology and Environmental Research (DOE)
BES	Office of Basic Energy Sciences (DOE)
DOE	U.S. Department of Energy
FES	Office of Fusion Energy Sciences (DOE)
FCCSET	Federal Coordinating Council for Science, Engineering, and Technology
GCM	general circulation model
HECRTF	High-End Computing Revitalization Task Force
HENP	Office of High Energy and Nuclear Physics (DOE)
Hypre	a solver library
ISIC	Integrated Software Infrastructure Center
ITER	International Thermonuclear Experimental Reactor
M3D	a fusion simulation code
NIMROD	a fusion simulation code
NNSA	National Nuclear Security Agency (DOE)
NSF	National Science Foundation
NUSEL	National Underground Laboratory for Science and Engineering
NWCHEM	a chemistry simulation code
ORISE	Oak Ridge Institute for Science and Education
QCD	quantum chromodynamics
PETSc	Portable, Extensible Toolkit for Scientific Computing, a solver library
PITAC	President’s Information Technology Advisory Committee
RHIC	Relativistic Heavy Ion Collider
RIA	Rare Isotope Accelerator
SC	Office of Science (DOE)
SciDAC	Scientific Discovery through Advanced Computing
TSI	Terascale Supernova Initiative

“We can only see a short distance ahead,
but we can see plenty there that needs to be done.”

Alan Turing (1912—1954)