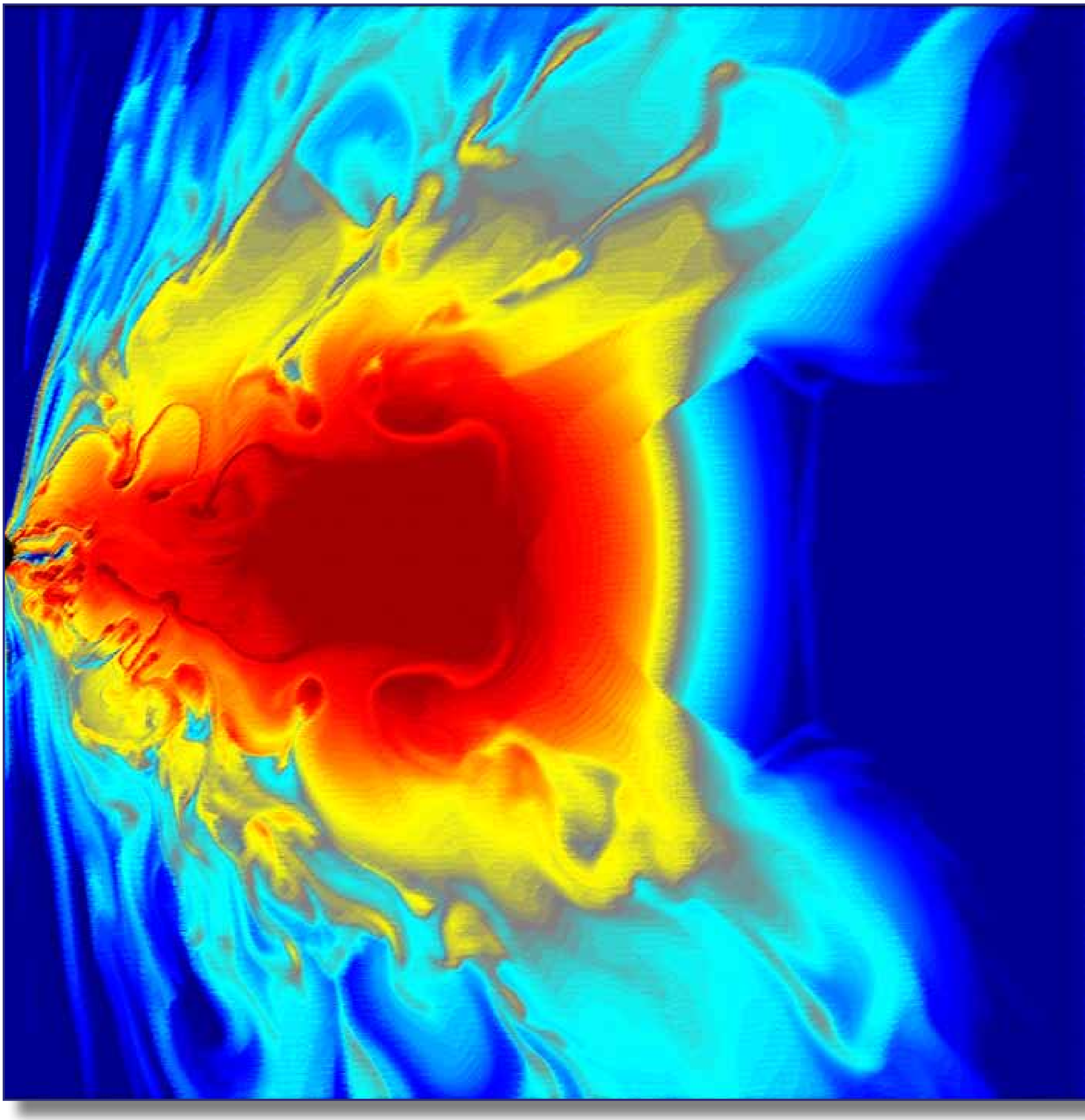
A VISION AND STRATEGY FOR SOFTWARE FOR SCIENCE, ENGINEERING, AND EDUCATION

CYBERINFRASTRUCTURE FRAMEWORK FOR THE 21ST CENTURY

NSF



On the cover:

This image shows the distribution of gas density (blue for low density to red for high density) in a black hole accretion disk simulation. The work was performed by John Hawley and Kris Beckwith of the University of Virginia. These models were designed to test the influence of different magnetic field configurations on disk and jet formation around a spinning black hole.

To learn more about this research, see the article “Assurance of things not seen,” published in NCSA’s Access magazine (summer 2008). [Research supported by NSF’s Information Technology Research for National Priorities grant PHY 02-05155.] (Date of Image: April 2008)

Credit: John F. Hawley, Kris Beckwith, University of Virginia

A VISION AND STRATEGY FOR SOFTWARE FOR SCIENCE, ENGINEERING, AND EDUCATION

CYBERINFRASTRUCTURE FRAMEWORK FOR THE 21ST CENTURY

NSF will take a leadership role in providing software as enabling infrastructure for science and engineering research and education, and in promoting software as a principal component of its comprehensive CIF21 vision. This includes ensuring comprehensive, usable, and secure software and services to further new scientific discovery and innovative education approaches by its researchers working in a globally connected and data-enabled world; fostering sustainable communities of software users, researchers, developers, industrial scientists and engineers, educators, and students that span disciplines, professions, and regions/countries; and promoting new approaches to learning and workforce development in software, and supporting investigations in the use of software for novel learning mechanisms. Reducing the complexity of software will be a unifying theme across the CIF21 vision, advancing both the use and development of new software and promoting the ubiquitous integration of scientific software across all disciplines, in education, and in industry.

INTRODUCTION AND BACKGROUND

Software is a critical and pervasive component of cyberinfrastructure for science, engineering, and education. Software is essential at all levels, providing the low-level drivers and transport protocols for networks; operating and file systems for scientific computing; runtime or development environments for high performance and/or distributed computing; new collaborative environments for virtual organizations; improved and coupled modeling, simulation, design, and visualization capabilities; as well as expressing the complex algorithms used to model and then analyze and understand data and processes in science and engineering.

Software is fundamentally computer code. It can be delivered to end users in multiple formats, ranging from an archive that a user downloads and builds to an executable or a service running on a remote system to which a user connects. Especially at large scale, software is generally difficult to design, implement and then maintain, and the software needed by the science, engineering, and education communities is particularly complex. Software must be reliable, robust, and secure; able to produce trustable and reproducible scientific results; yet its architecture must be flexible enough to easily incorporate new scientific algorithms, new capabilities, and new opportunities provided by emerging technologies. Software also must be supported, maintained, developed and eventually replaced in part or in entirety, over its lifecycle.

As computation and data analysis become increasingly important in scientific research, education, and industrial innovation, "enabling software and systems are needed to create an environment in which the barrier to access is low for innovation and new discovery," as stated in the PCAST report ***Designing a Digital Future***[1]. Software needs to transition from a set of individual research projects to a production infrastructure.

1 PCAST, "Designing a Digital Future: Federally Funded Research and Development in Networking and Information Technology," December 2010, ***http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-nitrd-report-2010.pdf***

The recent task force report of the Advisory Committee on Cyberinfrastructure[2] identifies several key challenges in software covering compute-intensive science, data, software evolution, and institutional barriers to software. Key recommendations from the task force include providing long term, comprehensive support for software at different levels; addressing verification, validation, uncertainty quantification, sustainability and reproducibility; providing policies for open source software; coordinating software activities across NSF, with other federal agencies, and with industry; and mechanisms for community input on software priorities. Improving education at all levels and training across disciplines in software use and development is also a recognized challenge.

This Strategic Vision identifies priority areas for NSF investment that will facilitate important and tangible progress in moving 21st Century science, engineering, and education toward more effective use of software and services, treating software as a principal component of cyberinfrastructure, and recognizing that the complexity of software has historically been underappreciated. The NSF vision is to facilitate software infrastructure that works easily at scale, encourages reuse, and efficiently promotes innovation while retaining reliability. This vision promotes greater balance in priorities, coordination, and leveraging, and encourages new strategies for fulfilling the maximal potential of prior cyberinfrastructure investments and new NSF investments.

GOALS FOR SOFTWARE FOR SCIENCE, ENGINEERING, AND EDUCATION

To meet the challenges before it, NSF will adopt, as part of its larger CIF21 mission and program, five interconnected strategic goals for delivering and sustaining software to advance science and engineering research and education:

- **Capabilities**: Support the creation and maintenance of an innovative, integrated, reliable, sustainable and accessible software ecosystem providing new capabilities that advance and accelerate scientific inquiry and application at unprecedented complexity and scale.
- **Research**: Support the foundational research necessary to continue to efficiently advance scientific software, responding to new technological, algorithmic, and scientific advances.
- **Science**: Enable transformative, interdisciplinary, collaborative, science and engineering research and education through the use of advanced software and services.
- **Education**: Empower the current and future diverse workforce of scientists and engineers equipped with essential skills to use and develop software. Further, ensure that the software and services are effectively used in both the research and education process realizing new opportunities for teaching and outreach.
- **Policy**: Transform practice through new policies for software addressing challenges of academic culture, open dissemination and use, reproducibility and trust of data/models/ simulation, curation and sustainability, and that address issues of governance, citation, stewardship, and attribution of software authorship.

(Left) Entropy distribution in developing Mach 1 turbulence is visualized with a National Science Foundation (NSF)-sponsored system for interactive analysis and visual exploration of multi-terabyte data sets. The white regions in the image show where the earlier passage of strong shock fronts has heated the gas in this turbulent flow. This visualization system and simulation data was used to develop, test and validate theoretical models of turbulence for use in simulations in astrophysics, as well as other areas of science and engineering.

Credit: Paul Woodward, Laboratory for Computational Science and Engineering, University of Minnesota

2 National Science Foundation Advisory Committee for CyberInfrastructure Task Force on Software for Science and Engineering, Final Report, March 2011, ***http://www.nsf.gov/od/oci/taskforces/TaskForceReport_Software.pdf***

STRATEGIES FOR SOFTWARE FOR SCIENCE, ENGINEERING, AND EDUCATION

The NSF Software for Science, Engineering, and Education strategy plan is part of the overall, coordinated CIF21 framework (see Figure 1) in which software fulfills the potential of new advanced computing infrastructure; provides new data infrastructure; bridges education and cyber-infrastructure between campuses, cities, towns, and countries; advances and expresses computational and data-enabled science and engineering; and enables a new generation of scientific and engineering communities.

To meet each of the software goals, NSF has developed a set of strategies.

1. Support the creation and maintenance of an innovative, integrated, reliable, sustainable and accessible ecosystem of software and services that advances scientific inquiry and application at unprecedented complexity and scale.

The software ecosystem must catalyze and support emerging new thinking, paradigms, and practices for science that are fundamentally inter-disciplinary and data-driven, and that integrate as an enabling layer with all other CIF21 activities. The software cyberinfrastructure must help researchers and application scientists and engineers address problems of unprecedented complexity, scale, resolution, and accuracy by integrating computation, data, networking, observations, experiments, and disciplines in novel ways. Previous and ongoing investments in software across the Foundation need to be leveraged and connected to a well-coordinated software plan that addresses sustainability. Evaluation of software efforts will be an important activity, tracking and monitoring both the quality of software and the scientific impact.

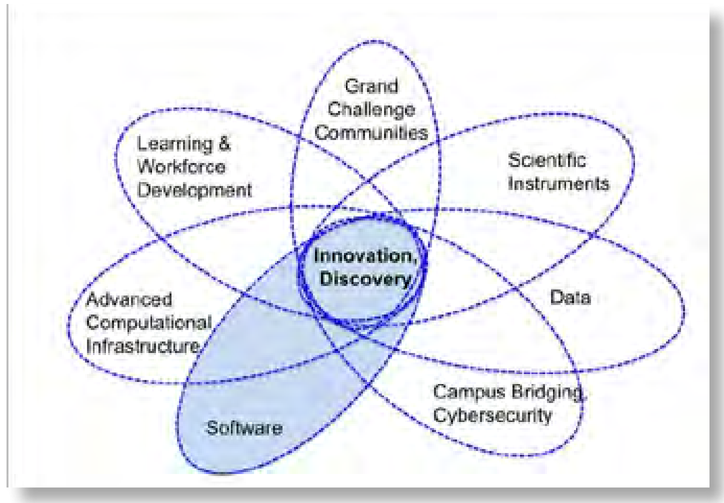
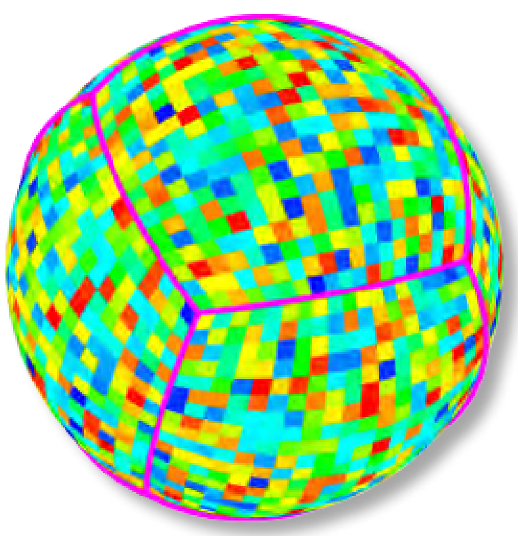


Figure 1 (Above): Software is an integrated component of the overall coordinated CIF21 framework. This figure illustrates the connections between the components of CIF21, including Software. All the components of CIF21 are generally needed to create the conditions for innovation and discovery, as shown in Figure 1.

NSF will address software though a tiered approach that is embedded throughout the Foundation's research programs, building capabilities through different activities for:

- Software Elements: targeting small groups that will create and deploy robust software elements for which there is a demonstrated need that will advance one or more significant areas of science and engineering.
- Software Frameworks: targeting larger, interdisciplinary teams organized around the development and application of common software infrastructure aimed at solving common research



(Left) This graphic was taken from a spectral-element application called SPECFEM3D GLOBE that uses a fine mesh of hexahedral finite elements, pictured here, and high-performance computers to help seismologists learn more about the inner sanctum of the Earth's core. Somewhat like the way a CAT scan images the brain, seismologists use the application to track seismic wave patterns from earthquakes to model the structure of the earth s core. One of the great challenges is to capture the propagation of high-frequency waves, with periods of 1 to 2 seconds, as they travel across the globe.

Credit: D. Komatitsch, Université de Pau; L. Carrington, San Diego Supercomputer Center at UC San Diego

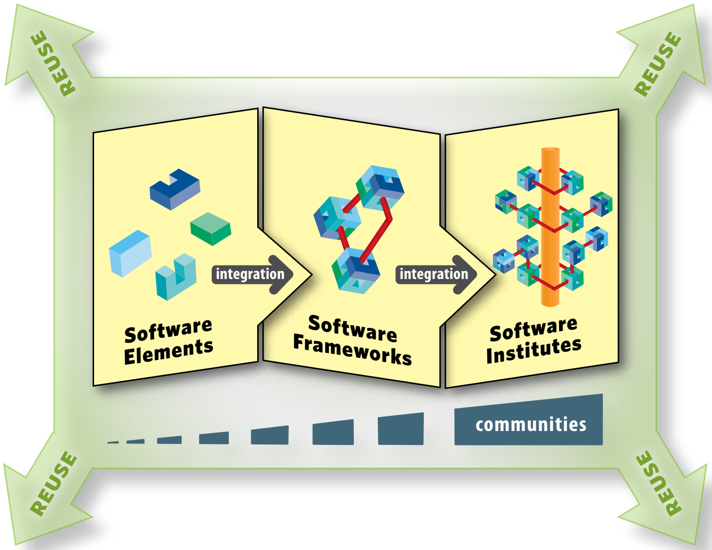


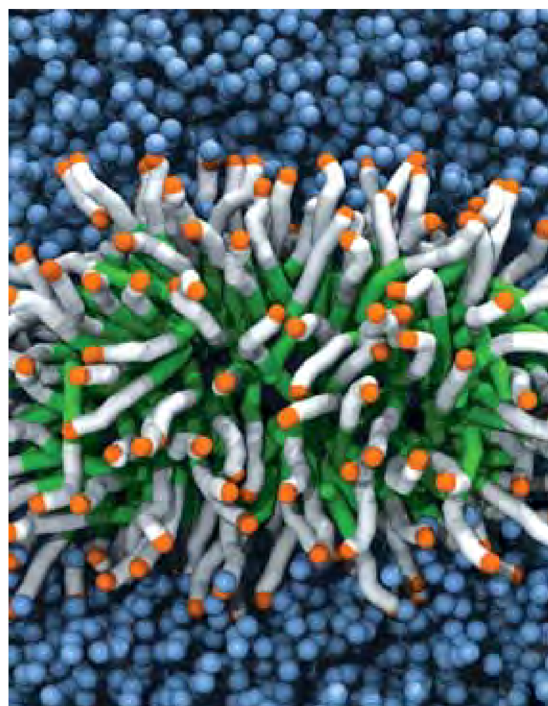
Figure 2 (Above): A tiered approach to software. NSF's software vision includes various types of projects, as illustrated in Figure 2, from small software elements to larger software frameworks, which may make use of software elements, to software institutes, which may work with multiple software frameworks and elements. Additionally, the user communities increase along this axis. A final element of the vision is reuse of software developed at all levels.

and industrial problems, resulting in sustainable community software frameworks serving diverse communities.

- Software Institutes: establishing long-term hubs of excellence in software infrastructure and technologies, research and application communities of substantial size and disciplinary breadth.
- Reuse mechanisms: Incentivizing individuals and communities to use and build on existing infrastructure to advance science and engineering.

NSF will provide long-term investment in activities that:

- Sustain and advance software infrastructure and software development at different tiers, from individual disciplinary research groups innovating new software elements to community-scale tools and frameworks to large-scale institutes coordinating broad activities across multiple disciplinary areas.
- Address all parts of the software lifecycle, from transitioning new research into practice, to operating and supporting well-used software, to ceasing support for superseded or lesser-used software.
- Recognize that software strategies must include the secure and reliable deployment and operation of services, for example by campuses or national facilities or industry, where identity, authentication, authorization and assurance are crucial operational capabilities.
- Provide mechanisms to incorporate innovations in software, and for software to rapidly evolve to leverage advances in technology and new functionality required by scientific disciplines.
- Result in high-quality, usable, secure, vulnerability-free, sustainable, robust, well-tested, and maintainable/evolvable software; and which promotes the sustainability of solid and useful on-going investments.
- Ensure that software is well documented, disseminated and discoverable, with accessible resources for training and user ratings/reviews.
- Promote the use of elements of the software infrastructure in learning and workforce development activities.
- Create generic models of both the cost of making software reusable and the benefit of software reuse, and use specific instances of these models for software decisions.



(Above): Snapshot of a self-assembled elongated micelle of non-ionic surfactant molecules (penta- (ethyleneglycol)--dodecylether,C12E5) in water from a coarse grain molecular dynamics simulation. These sorts of molecules are used in everything from detergents and shampoo to drug-delivery systems. Structures like micelles and vesicles form and can trap or protect other materials. This image was created using visual molecular dynamics from the University of Illinois' Theoretical and Computational Biophysics Group, and it was rendered with the embedded Tachyon renderer.

Credit: Axel Kohlmeyer, Institute for Computational Molecular Science, Temple University

2. Support the foundational research necessary to continue to efficiently advance scientific software, responding to new technological, algorithmic and scientific advances.

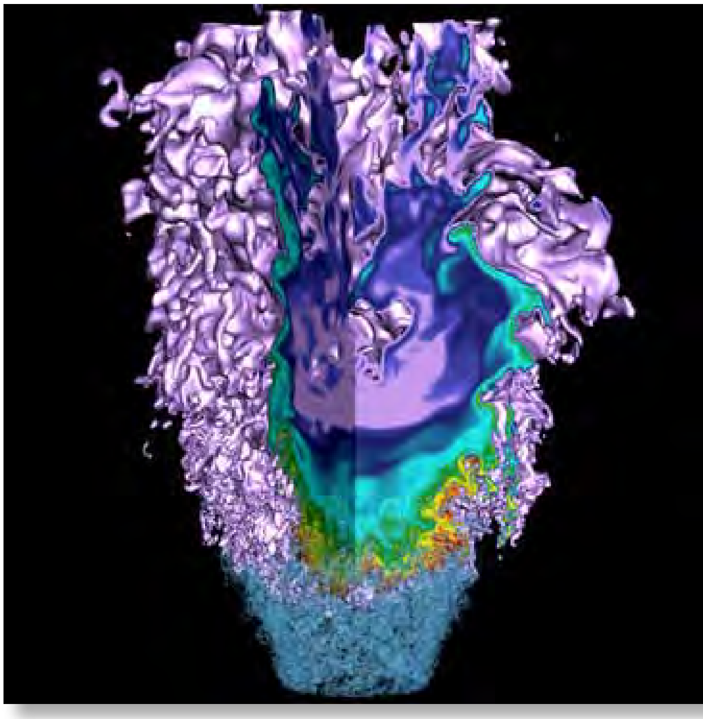
Core programs at NSF already support areas of foundational research to advance software development. Clear pathways must be in place for such foundational research to impact operational software and services, so key gaps in research can be identified and resolved.

NSF will invest in:

- Activities that identify gaps between available software and science, engineering, and education needs in the near and long term.
- Research activities that advance software development, use, and accessibility, including:
  - Programming paradigms that address the use of massively parallel computers, highly distributed computers systems (including private, public, and hybrid clouds), complex file systems (both parallel and distributed), new accelerator architectures and the potentially hybrid systems that will be built from them;
  - High-level abstractions and frameworks that promote code reuse and sharing, model extensibility and interoperability, and simplify domain specific programming while achieving high performance;
  - Middleware for dynamic data-driven workflows;
  - Software for collaborative science and engineering (technologies for teams, data, and computing);
  - Paradigms for verification, validation, uncertainty quantification, assurance, and provenance that will ensure trustworthy and reproducible scientific and engineering findings;
  - Resilience/fault-tolerance: algorithms and techniques for avoiding, discovering, and recovering from anomalous conditions;
  - Tools and services for gateways/portals/hubs;
  - Associated software productivity tools, addressing for example testing, debugging, profiling and visualization;
  - Interfaces and integration between heterogeneous entities: imaging, devices and systems, software, visualization and data mining, and control and decision making;
  - Goal-based (autonomous and adaptive) software and services.
- Understanding issues that prevent reuse; and incentivizing reuse in the development process.
- Developing pathways to practice that move the outputs of research activities into the infrastructure, advertise them, document them, and train users in academia and industry.
- Encourage the transition of mature software into industrial and medical practice to lead innovation and product development.
- Developing principles and practices for sunsetting existing software elements, and possibly replacing them with new ones, with the participation and awareness of the user and developer communities.

(Left) The Very Large Array (VLA) radio telescope at sunrise. Astronomers recently used the National Science Foundation's Very Large Array (VLA) radio telescope (above) to help find the most distant water yet seen in the universe, in a galaxy more than 11 billion light-years from Earth. Previously, the most distant water had been seen in a galaxy less than 7 billion light-years from Earth. The soggy galaxy is dubbed MG J0414+0534. In a region near its core, water molecules are acting as masers, the radio equivalent of lasers, to amplify radio waves at a specific frequency.

Credit: Dave Finley, NRAO/AUI/NSF



(Above) A still from a simulation produced during a study on lean, hydrogen-air mixtures showing a cutaway profile of the concentration of hydroxyl molecules (that are produced and consumed at the flame). Red areas mark regions of intense combustion; fine gray-blue vortex structures at the base of the flame mark turbulence.

Credit: Lawrence Berkeley National Laboratory

3. Enable transformative, interdisciplinary, collaborative, science and engineering research and education through the use of advanced software and services.

Broad investments need to be made to support the use, and associated advancement, of production software for all scales of science problems where software integrates the use of new algorithms and scientific capabilities with cyberinfrastructure resources. This software must be driven by the requirements of scientists and engineers, considering both immediate needs and longer-term goals.

NSF will invest in activities that:

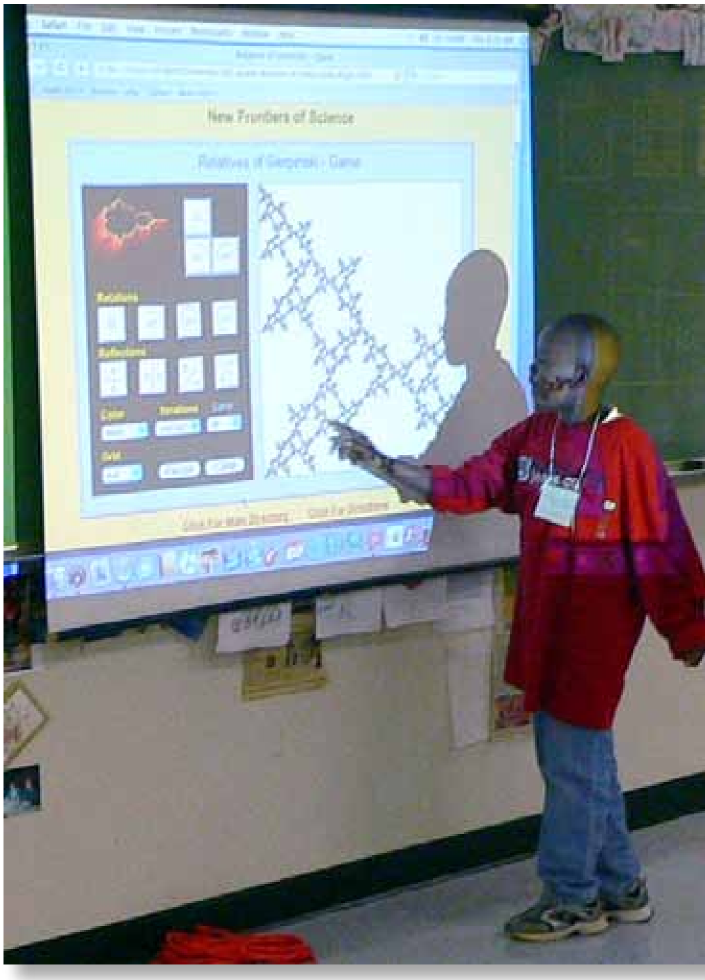
- Develop an NSF-wide program in computational and data-enabled science and engineering (CDS&E) to provide the research and innovation in new methods, algorithms, and approaches that will advance software for science and engineering.
- Address a range of research questions, from grand-challenges to long-tail research, by engaging a comprehensive and integrated approach to science and engineering, utilizing software, high-end computing, data, networking, facilities, and multidisciplinary expertise.
- Engage early-stage researchers in the use of complex and visionary end-to-end scientific use cases in different disciplines that drive innovation in cyberinfrastructure development and use.
- Support community-building activities to promote collaboration that crosses disciplinary, institutional, and geographic boundaries through shared software concepts and standards.

4. Empower the current and future diverse workforce of scientists and engineers equipped with essential skills to use and develop software. Further, ensure that the software and services are effectively used in both the research and education process realizing new opportunities for teaching and outreach.

NSF must integrate software use with educational activities at all levels to prepare students for 21st Century careers and to maintain competitiveness in the international marketplace and create and support career paths for software developers.

NSF will invest in activities that:

- Provide students with the expertise to use and extend the most up-to-date tools and techniques and contribute to the scientific software infrastructure,
- Provide new curricula and methods to teach students at all levels and across all disciplines best practices for software engineering,
- Provide general curricula that enable software appreciation and literacy among the public and policy making communities,
- Facilitate and encourage professional career tracks in computational science and software engineering.
- Encourage proactively broadening participation of women and underrepresented minorities.



(Above) A middle school student describes a mathematical image at the Summer Institute for Middle School Math Teachers in Broward County, Florida. The summer institute, which trains middle school math teachers, is sponsored by the Institute for Standards Mapped Graduate Education and Mentoring Math and Science Partnership (SMGEM) program.

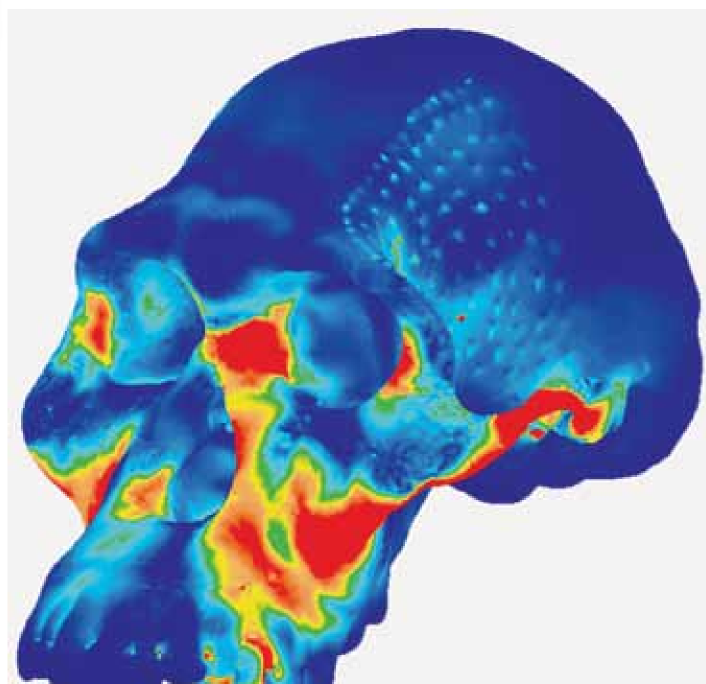
Credit: Richard F. Voss, Heinz-Otto Peitgen

5. Transform practice through new policies for software addressing challenges of academic culture, open dissemination and use, reproducibility and trust of data/models/simulation, curation and sustainability, and that address issues of governance, citation, stewardship, and attribution of software authorship.

Sustainable models for software infrastructure development will require the evolution of existing policies as well as social change in the research and academic communities.

NSF will lead initiatives targeted at evolution and change in the following areas:

- Widespread adoption of open source models for software development and dissemination that include high quality documentation and accepted engineering practices, leading to software that is accessible, understandable and reusable;
- Mechanisms for citation of software as distinct products of scholarship, promoting standards of academic credit and rigor for software;
- Mechanisms for community advertisement and effective software dissemination;
- Cost-effective strategies for broad, interdisciplinary collaboration in software development, including with international and industrial entities;
- Governance and sustainability models, including those currently in use by open source communities and by industry;
- Development and use of metrics that measure software usage and impact on science, engineering and education;
- Determining existing barriers to software reuse, from both the developer and user points-of-view, and promoting mechanisms to overcome them.



(Above) Computer-generated model showing compressive stress in the cranium of Australopthecus africanus, an extinct early human that lived around 2.5 million years ago, imposed by biting on premolar teeth. Bright colors correspond to high stresses and indicate that a bony pillar running alongside the opening of the nasal cavity acts as a strut that structurally reinforces the face against premolar loads.

Credit: Simulation courtesy of Arizona State University and the "Hominid Feeding Biomechanics" research team

NSF 12-113