NERSC “Visualization Greenbook”

Future Visualization Needs of the DOE Computational Science Community Hosted at NERSC

Report Prepared by:

Bernd Hamann, UC Davis
E. Wes Bethel, LBNL/NERSC
Horst Simon, LBNL/NERSC
Juan Meza, LBNL/NERSC

October 2002

Lawrence Berkeley National Laboratory

1 Executive Summary

This report presents the findings and recommendations that emerged from a one-day workshop held at Lawrence Berkeley National Laboratory (LBNL) on June 5, 2002, in conjunction with the NERSC User Group (NUG) Meeting. The motivation for this workshop was to solicit direct input from the application science community on the subject of visualization. The workshop speakers and participants included computational scientists from a cross-section of disciplines that use the NERSC facility, as well as visualization researchers from across the country. We asked the workshop contributors how they currently visualize their results, and how they would like to do visualization in the future. We were especially interested in each individual's view of how visualization tools and services could be improved in order to better meet the needs of future computational science projects. The outcome of this workshop is a set of findings and recommendations that are presented in more detail later in this report, and briefly summarized here.

Scientific visualization is a crucial technological capability that plays an important role in understanding data created by computational science projects as well as experiments. In order to be effective, visualization technology should be easy to use for a non-expert. The term "easy to use" encompasses a number of different categories, including a short learning curve, tight integration with computational frameworks, availability on the desktop as well as the fixed visualization facility, tools that are tailored for each specific application domain, and low cost.

Current visualization tools fall short in several key areas of capability. Few visualization tools are capable of processing large datasets, such as those commonly generated at NERSC. Better support for parallel visualization tools may prove useful in leveraging large parallel machines as visualization resources. Multivariate visualization — multiple grids, many species, and many dimensions — is needed in order to quickly gain insight into large datasets. Related "drill-down" capabilities, such as the ability to quickly move from macro to micro views (used in "data mining"), would be extremely helpful in understanding data but are missing from most visualization tools.

Many application scientists perceive a conundrum when it comes to visualization support. Support for visualization within each individual program level is often inadequate or nonexistent due to funding constraints, yet support for visualization at the institutional level is also often inadequate or nonexistent. Better solutions are needed for remote visualization. Current approaches are further constrained by network bandwidth and access to resources.

The proliferation of visualization tools and data formats poses challenges. Researchers must often master many different tools in order to achieve the desired results. Data format conversion is often required when moving between tools. Common data formats and frameworks for visualization tools are needed to reduce duplication of effort and better promote sharing of resources and results.

Better communication is needed between the visualization and computational science communities. The computational scientists are often unaware of current trends and practices in the visualization community. By being more aware of the needs of the computational science community, the visualization research programs can be crafted so as to be more responsive to their needs.

As a result of the workshop, we have developed a set of recommendations that can be summarized as follows:

?? *Establish a coherent program that focuses on remote visualization. A remote visualization program should provide tools and infrastructure that can be used by multiple "virtual teams."*

?? *Establish mechanisms whereby generally applicable visualization technology is developed and deployed in a centralized fashion.*

?? *Develop a research program in interactive visualization with running codes that stresses the integrated design and development of coupled simulation-visualization methods.*

?? *Establish a research program in the areas of multi-field visualization and multi-dimensional data visualization.*

?? *Establish a research program in the area of automated data exploration for next-generation petascale datasets.*

?? *Significantly enhance life science data visualization efforts, with particular emphasis upon the relationship with scientific data management.*

?? *Develop new programs that link visualization with data management and provide support for multiresolution representations of large datasets, support for simultaneous display of data from disparate sources, support for the ability to generate and display derived values, and the ability to pose queries and display results.*

2 Introduction

This report is the result of a one-day workshop that was held at Lawrence Berkeley National Laboratory (LBNL) on June 5, 2002, in conjunction with the NERSC User Group (NUG) Meeting. The co-organizers of this workshop were Bernd Hamann (UC Davis/LBNL), Wes Bethel (LBNL/NERSC), and Horst D. Simon (LBNL/NERSC). The workshop brought together scientists representing a large cross section of computational science applications and visualization experts, including participants from national laboratories and universities. As the objective of this workshop was to hear from application scientists about their needs concerning data visualization, no talks were included in the program that dealt with "visualization solutions." Instead, all speakers were encouraged to address visualization challenges posed by their specific applications. (See http://vis.lbl.gov/Events/VisGreenbookWorkshop-June02/ for more information about the workshop.)

In 2001, the Department of Energy (DOE) Office of Science created an ambitious program called Scientific Discovery through Advanced Computing (SciDAC). One of the goals of SciDAC is to develop and deploy the computer science tools that will enable computational scientists with DOE mission-relevant applications to take full advantage of terascale computing platforms. While a large number of computational science-centered projects are funded in virtual organizations called Integrated Software Infrastructure Centers (ISICs), it became apparent that visualization had not been integrated as a crucial component of the SciDAC program. In fact, only a small number of computational SciDAC projects include a specific visualization component. Yet it is clear that many DOE-supported computational scientists who use NERSC or other Office of Advanced Scientific Computing Research (OASCR) computational resources have visualization needs that are currently not met either at the programmatic or at the institutional level.

The one-day visualization workshop held at LBNL was dedicated to the identification of crucial data visualization needs. Computational scientists and engineers presented their research efforts, with a special emphasis on their visualization requirements for current and next-generation projects. Speakers at the workshop included leading computational scientists and engineers from across the DOE scientific computing programs that use the NERSC facility. All presenters were asked to provide summary statements about their research efforts with particular emphasis on their individual visualization needs. These statements form the basis of this report. We believe that this report will be valuable for further planning and prioritization of future visualization programs within DOE.

3 The Impact of Scientific Visualization on Science

The following sections present a number of different computational science projects hosted at NERSC. Each individual contributor has provided a brief description of their computational science project, along with some statements that capture their anticipated visualization needs now and in the future.

3.1 Macroscopic Magnetohydrodynamics

Linda Sugiyama, Massachusetts Institute of Technology

The overall goal is the development of extended macroscopic magnetohydrodynamics (MHD) models for the simulation of magnetically confined plasmas, starting from a fluid-like approach based on moments of the particle kinetic equations. MHD simulation has existed for a long time and is considered well understood, although important aspects are not (e.g., the effects of compressibility and density evolution, parallel thermal conduction). Extended MHD introduces more physical effects by allowing electrons and ions to move separately, while keeping many of the computational advantages of a fluid approach. The goal is to approach full particle simulation from a different direction, with different assumptions. A two-fluid model (electrons and ions) for general toroidal configurations has been developed and is being extended by adding a gyrokinetic particle ion population. Extended MHD and two-fluid models are not physically or mathematically complete like MHD and are more difficult numerically. Two-fluid instabilities have intrinsic frequencies of rotation. More and faster waves and smaller length scales exist. Other plasma effects beyond the strictly fluid appear in the closure terms that truncate the set of moment equations. Motions along the magnetic field become important, such as the "neoclassical" viscous force and the fast parallel streaming of electrons that equilibrates the electron temperature along the field lines

Visualization consists of a set of methods used to present mathematical results to people in a form that is easy to understand and analyze. Its primary purpose is to aid in understanding and insight. Good visualization tools should also minimize the "wear and tear" on the user, as well as the personal time required. User needs change with time, so fast, simple, and flexible solutions remain important. Full visualization solutions must have the support of major computing center policies to be effective.

Data size. There is general agreement that datasets will become larger and the handling of these datasets will be a major limitation on numerical simulations. There are two magnitudes of data size, before and after visualization. Efficient compression and expansion of visualized data is also needed. Many proposed technical solutions for large datasets are good and they will be needed. Nevertheless, there are other limits on data size than absolute ones. A primary limit is the size of the "basic job unit." Batch scheduling algorithms and single processor capability at the large-scale computing facility determine the maximum job size that can regularly have turnaround time of a day or less. (On the IBM SP at NERSC, this is presently a few-node job with an eight-hour time limit — say, two nodes and 32 processors, corresponding to a low- to moderate-resolution case for a few hundred time steps.)

Code and physics model development: The requirements are different from production or application runs. Production runs require comparatively little user time or brainpower, since the basic result is known beforehand. Long compute times mean that visualization can be done at the user's leisure, even if datasets are large. Absolute size limits may be important, since thorough analysis is desired, but time limits are less important if data can be transferred or stored during runtime. Development runs require more efficient and

flexible visualization, since they constitute the bulk of the code developer's time and effort. Extended MHD plasma models are being developed in part numerically and have little useful theoretical or numerical background to draw on. An unusual feature appearing in the simulation may be real or numerical instability. Concentrated layers and local wiggles must be resolved and viewed (post-run zooming). Time evolution of individual features is important. For debugging, MPP quantities should be viewable across processors.

The ideal job control scheme for code development and physics exploration is interactive and allows the regular viewing of important quantities, followed by the adjustment of the job parameters during the run. Viewing a running job requires user control of the job timing, which is unlikely on a shared MPP computer. The next best solution is for the MPP code to generate a set of simple diagnostics/plots, perhaps pre-transferred to the user's local viewing platform, with zero hands-on user time. Fast, simple graphics need to be integrated with fancier pictures and 3D. Contour and profile plots have great precision and analytical power. Pictures alone without quantitative measures are only marginally useful. It is useful to be able to run a smaller version of the MPP code to extract and calculate the desired quantities, perhaps on a local computer.

Sharing visualization results. Most large codes are multi-institutional projects, whose goal is distribution and use of the code by a larger community. For macro-MHD, outside application and use will rapidly become important over the next few years. Sharing of visualization results is needed, including shared storage or publishing location for developers and users. Even still, moving pieces of the same data many places will be necessary, so data compression and expansion techniques for visualization data will also be important. The ability to hold common discussions with minimal equipment, e.g., conference call with telephone and computer screen, should also be considered part of the visualization toolset, ideally including the capability to write mathematical equations and to draw on or off the visualization pictures. Thought should also be given to simple integration of visualization output in formal presentations and publications.

Visualization techniques need development at the theoretical and practical levels for use in major codes. Some important types of analysis can only be done by the human eye, such as extracting representative quantities out of discretized or noisy data. One example is determining the width of a magnetic island as a function of time. The computed magnetic field lines are necessarily discretized by a field-following algorithm, and there may be embedded fine structures or chaotic regions. The island itself has a 3D structure, so the maximum width must be determined over the whole structure.

In summary, visualization support for major codes should take into account practical as well as absolute limits. Techniques for handling large datasets are increasingly important, but should not be the sole priority. Reducing user time and effort should be the primary goal. Large computer centers should make a commitment to support visualization, including fast, efficient plotting packages and quick viewing of results, large-scale data transfer, and storage. Job scheduling algorithms should ideally be adjusted to support visualization needs. Multi-institutional and multi-user projects with users outside the core

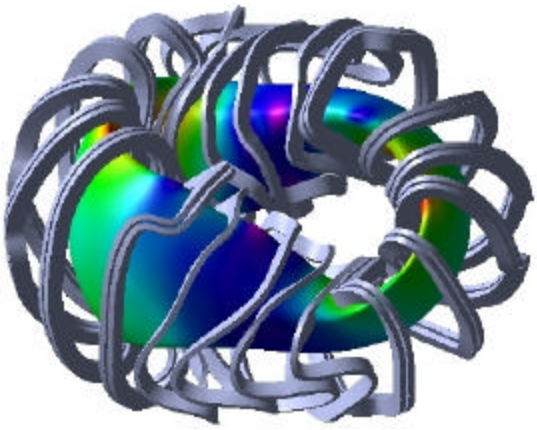
development group are rapidly becoming important. Major codes should choose and develop distributable, affordable visualization systems. Remote visualization information exchange and interaction should be developed. New developments in visualization, such as extracting information from discrete or incomplete data, should be supported.

3.2 Plasma Confinement, Stability, Heating and Optimization in Stellarators and Tokamaks

Don Spong, Oak Ridge National Laboratory

Our group works on the optimization and physics analysis of stellarators. These are non-axisymmmetric toroidal fusion devices that utilize optimal 3D shaping to achieve desirable properties such as good plasma confinement, stability, and self-consistent steady state equilibria. This work involves optimization of the plasma shape using either 30–40 variables that characterize the outer magnetic flux surface or several hundred variables that characterize the magnetic coil geometry (Fig. 1).

The optimization algorithms we use include Levenberg-Marquardt, differential evolution, and genetic algorithms. The optimization targets are comprised of various measures of plasma confinement and stability; these are obtained both from codes that we have developed as well as codes obtained from groups throughout the world fusion program. In order to connect all of these disparate codes together, our optimization effort has successfully addressed large-scale code integration issues.



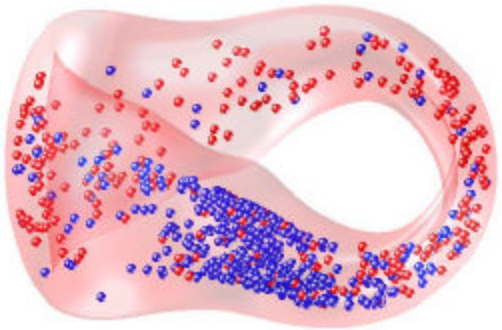


Figure 1. (Left) Magnetic flux surface and modular coils for a compact quasi-poloidal stellarator (QPS). Color coding on the flux surface indicates the magnetic field strength (blue is low field, red is high field).

Figure 2. (Right) One frame from a stellarator particle orbit simulation in a compact stellarator (red spheres are passing particles, blue spheres are locally trapped particles*).*

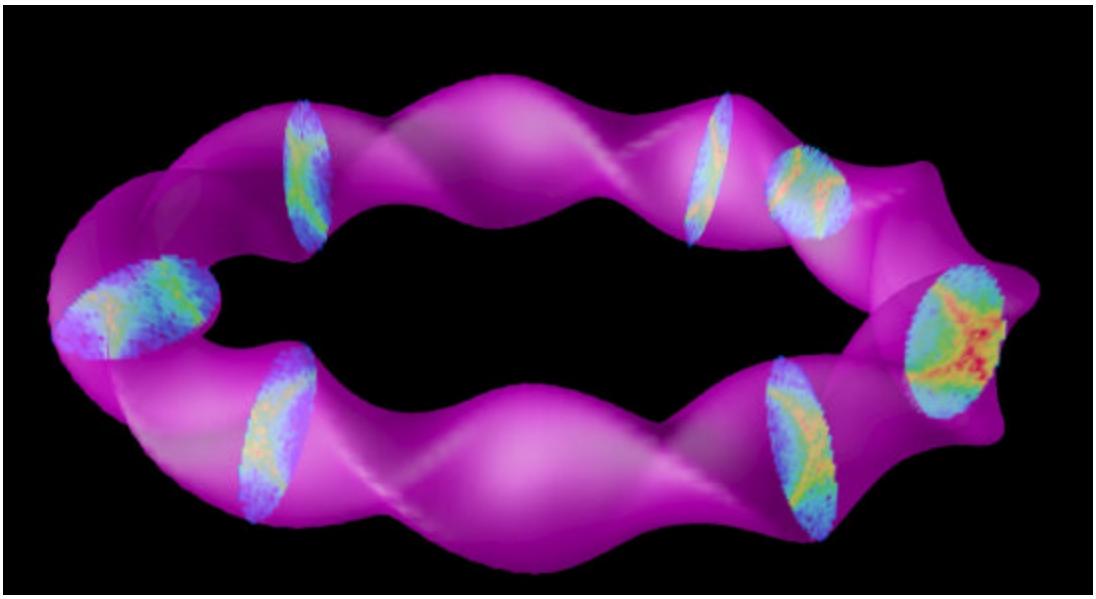


Figure 3. RF wave field visualization in the LHD (large helical device) stellarator. The purple surface is the outermost magnetic flux surface. Color-coded RF wave field strengths are shown on seven cut planes at fixed toroidal angles.

The physics analysis of stellarators includes such issues as transport and confinement, stability, turbulence, and RF heating. Confinement is being addressed both through Monte Carlo particle simulations (Fig. 2) and through direct solutions of the Boltzmann equation. Stability and turbulence are addressed through MHD codes that solve the coupled fluid/Maxwell equations in time. RF heating (Fig. 3) involves solving field equations in 3D that have both local and long-range couplings; these lead to large, dense matrix problems.

Most of the data we currently visualize involves scalar fields localized to a single 3D surface. The typical size of this dataset is 10 MB. We also visualize collections of particles in 3D and make animations of their motion. A typical data size for the time series data that we currently work with is 50 MB; however, this is for a greatly reduced number of particles (10^3) and for a short time interval (8×10^{-5} seconds).

In both of these areas it would be very helpful to have access to hardware that would allow visualization of much larger datasets. The limits are largely set by what performs well on our existing desktop PC and Linux systems. As we move away from stellarator plasma surface optimizations to physics simulations (confinement, stability, turbulence), we would like to visualize volumes rather than surfaces. For marginally adequate resolution, this will require data on at least ~100 surfaces within the plasma, leading to 1 GB data files. As there are typically phenomena present with large separations of scale, we may need significantly more resolution, leading to 10–100 GB datasets. We would also like the capability to visualize vector fields within these volumes, leading to at least 3 GB datasets. Although such datasets can be attempted with our existing hardware, we

typically find that the responsiveness becomes sufficiently poor that it is not useful. For example, we already find that in visualizing magnetic surfaces and detailed coil models (e.g., such as shown in Fig. 1), the geometry data size reaches ~140 MB. This exceeds the memory available on our graphics card (128 MB) and so objects can no longer stay in cache. At this point, performance becomes very poor and discourages visualization of this many objects at once. Also, in the area of particle animations, we would like to visualize particle dynamics on collisional timescales (~10^{-1} seconds) and with the number of particles that we use in our Monte Carlo runs (~10^5–10^6). However, this would lead to an enormous dataset (~70 TB) that is well beyond the capacity of any resource we expect to be able to access at this time.

We do visualizations both on UNIX/Linux workstations and on Windows and Mac PCs. We use a variety of software, including AVS5 AVS/Express, OpenDX, Mathematica, IDL, as well as some direct OpenGL programming. All these solutions require that we constrain the size of the dataset to not exceed the capacities of the local workstation.

3.3 Numerical Tokamak Turbulence

Bruce Cohen and Bill Nevins, Lawrence Livermore National Laboratory

The performance of magnetic fusion confinement experiments is significantly degraded by fine-scale turbulence (electromagnetic fields interacting with self-consistent perturbations of the plasma density and flows). The understanding and control of this turbulence is an important component of the U.S. fusion research program. The Plasma Microturbulence Project undertakes 3D simulations of core turbulence in tokamak experiments. These simulations are proving to be important tools in analyzing current experiments, in extending theory, and in predicting future experiments. This research activity is a SciDAC multi-institutional project funded by the Office of Fusion Energy Sciences in the Department of Energy.

Computer simulations are useful proxies for experiments. They are generally easier to build and easier to run. They are also easier to diagnose, for they provide more scope for parameter variations. The Plasma Microturbulence Project has already developed high-fidelity numerical models. There are more improvements to come, but these simulation codes are already useful today. This project has developed sophisticated diagnostics tools specific to this domain, yet this group is constantly seeking ways to improve their data analysis tools. Sophisticated, interactive data analysis of simulation data is essential in understanding plasma turbulence. Visualization follows analysis and aids intuition and understanding (Figs. 4–6).

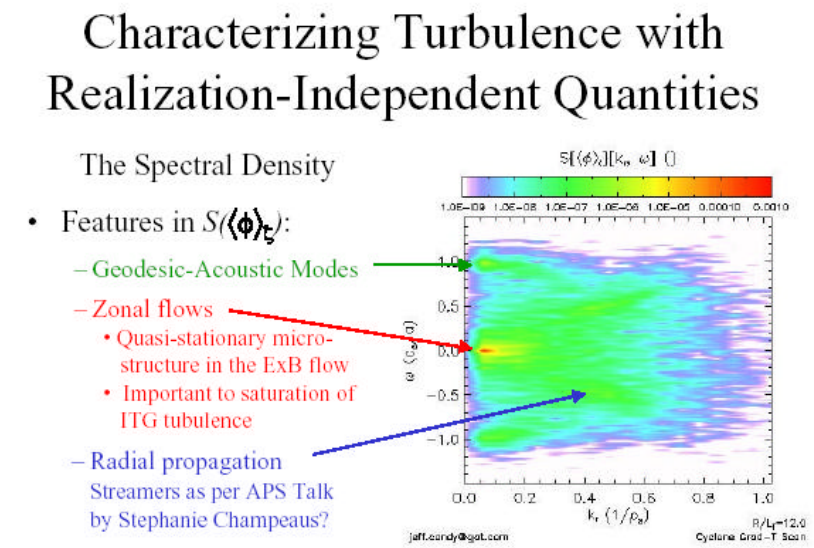


Figure 4. Spectral density of the electric potential in tokamak microturbulence, illustrating the use of visualization to assist physics analysis.

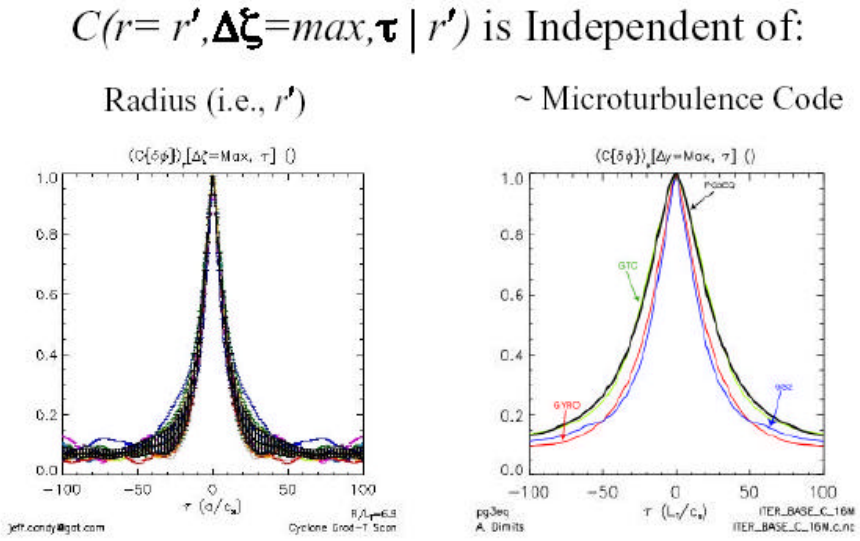


Figure 5. Numerical tokamak correlation functions for the electric potential showing generic behavior that is independent of both physical radius in the simulation domain and simulation code.

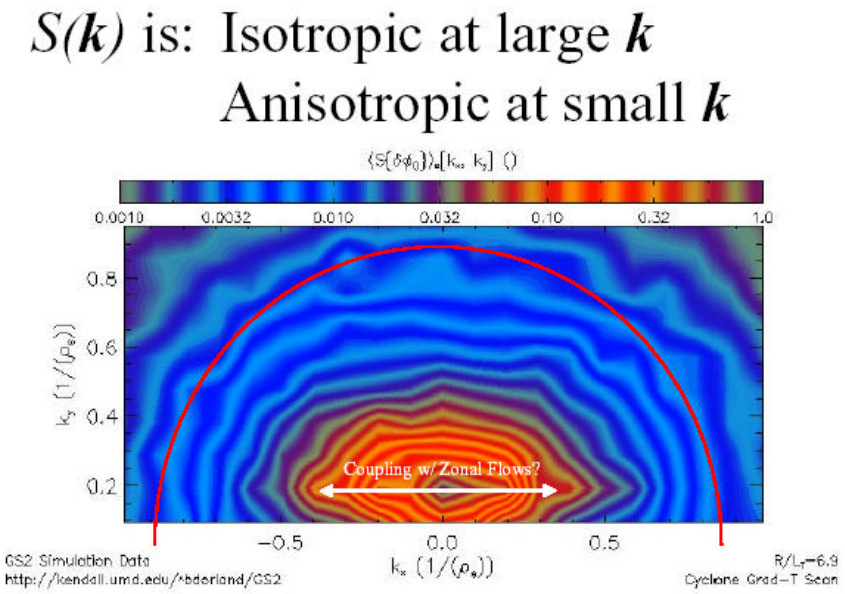


Figure 6. Contour plot of spectral density of the electric potential showing that the tokamak microturbulence is isotropic at short wavelengths and anisotropic at long wavelengths.

In order to achieve objectives of data understanding, a number of visualization requirements can be summarized as follows.

1. Basics: 2D vs. time: 1 scalar ?????????????????= 50 Mwords. These 200 MB datasets are typically downloaded to the workstation, where they are visualized locally.
2. Desirable: 3D vs. time: 1 scalar ??????????????????????= 5 ????? words. Big datasets challenge resources to move, manipulate, visualize, and store.
3. Kinetics introduce two more phase-space dimensions. (5D gyrokinetics: ?, v||, ?, ?, r, and time). More combinations of projected phase space 5D ? ?4D, 3D, 2D 5D + time datasets ~ ??? words per scalar — a very big dataset. Also, how do you visualize 4D or 5D data?
4. Need analysis/visualization tools that are interactive and cheap to the user.
5. Analysis and visualization tools are important. Stunning visualizations have political value.

3.4 Accelerator Design

Rob Ryne, Lawrence Berkeley National Laboratory, and Kwok Ko, Stanford Linear Accelerator Center

Particle accelerators are among the most important and most complex scientific instruments in the world. The nation's accelerators — including its high-energy colliders, synchrotron light sources, and spallation neutron sources — are critical to research in fields such as high energy physics, nuclear physics, materials science, chemistry, and the biosciences. Beyond applications to basic and applied science, accelerators have also

been proposed to solve problems of national importance. Examples include using accelerators to transmute and destroy nuclear waste and using accelerators to produce energy through fission or fusion. Beyond these large-scale applications, particle accelerators and the technology associated with them have many uses that are highly beneficial to society and to U.S. industry such as accelerators for medical applications (e.g., isotope production, tumor irradiation) and accelerators for beam lithography. All told, accelerators have had and will continue to have a profound impact on U.S. leadership in science and technology, and on improving the quality of people's lives.

The development of particle accelerators involves investments in all three paradigms of scientific research: theory, experiment, and simulation. In regard to the last of these, most activities in accelerator simulation fall naturally into three broad categories: simulation of beam dynamics, simulation of electromagnetic phenomena, and simulation of the self-consistent interaction of particles and electromagnetic fields. The importance of accelerators, and the challenges posed by the next generation of accelerators, have resulted in a significant national effort to develop a new generation of accelerator design tools targeted to high-performance computing platforms. The size and complexity of the data produced by these computer models has also led to an increased awareness of the need for new visualization tools and techniques to analyze the data produced in these simulations.

Visualizing beam halos provides an example of particle visualization needs. Fig. 7 shows two simulation results: the figure on the left is the transverse beam profile of a beam that has been properly injected into a beam transport system; the beam on the right is "mismatched," which has resulted in a large-amplitude, low-density region of charge (the halo) far from the beam core. Understanding and predicting halos is a key issue for present and future high-intensity accelerators. It also presents a challenge in visualization, in part due to the large dynamic range of the beam density function. Often beam physicists are interested in halo fractions of as little as parts per million.

As another example, Fig. 8 shows the computational mesh associated with a multi-cell accelerating structure. Producing these grids, which involve complex 3D geometries, and visualizing them, is important in order to assure the quality of the mesh and the resulting computational results.

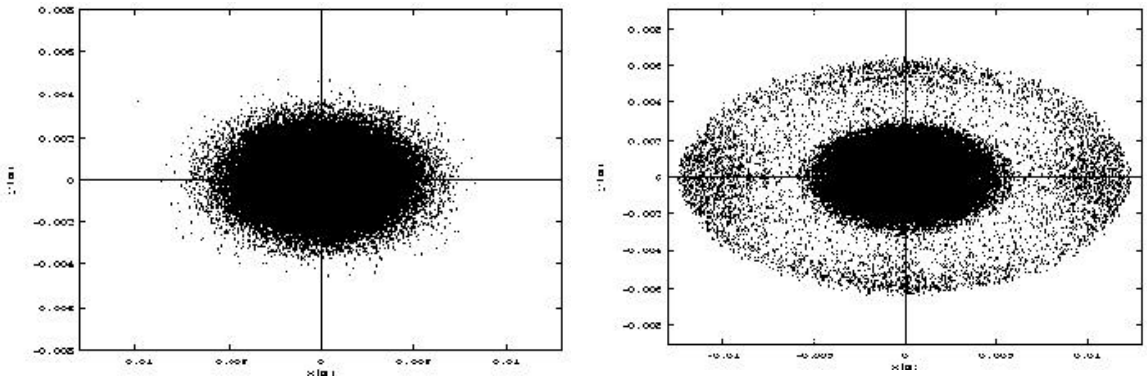


Figure 7. Plots of beam density in a transverse slice.

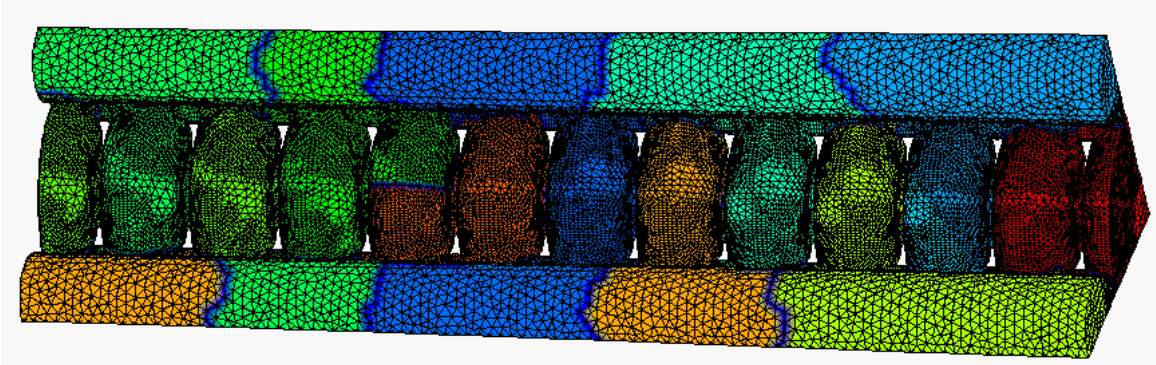


Figure 8. Computational grid corresponding to a multi-cell accelerating structure. The different colors show the domain decomposition.

In general, visualizations associated with accelerator modeling involve particles and/or fields, scalar and/or vector data, and data on regular and irregular grids. In some cases, the underlying data involves high dimensionality (e.g., the six-dimensional phase space in which particles exist); in these cases, visualization techniques are needed to view and analyze the high-dimensional data.

3.5 TeraScale Supernova Initiative

Doug Swesty, State University of New York, Stony Brook, and Tony Mezzacappa, Oak Ridge National Laboratory

The TeraScale Supernova Initiative (TSI) is one of the SciDAC projects funded under the DOE Office of High Energy and Nuclear Physics (HENP). Our mission is to develop next-generation terascale models of core collapse supernovae. The modeling efforts are carried out by means of parallel radiation-hydrodynamic simulations of the collapse of a stellar core and the subsequent explosion. The simulations involve radiating, chemically reactive flows that have a complex flow structure. The problem also has a long timescale, which means that the output datasets could potentially have thousands of time slices. In order to produce more scientific yield from our simulations, we require advancement in several key areas of scientific visualization.

Data from our simulations is produced in the form of files using the HDF5 format developed by NCSA. We carry out parallel I/O from our simulations in using the parallel HDF5 interface built on top of MPI I/O. The use of the HDF5 format ensures portability between architectures. It has also become a widely supported portable data standard. Our hope is that visualization tools developed under SciDAC will have the ability to directly read from HDF5 files. By supporting this rapidly growing standard, SciDAC visualization tools could eliminate the need for time-consuming conversions of output data into a form that is readable only by a specific tool.

Our simulations involve either two or three spatial dimensions and one or more energy and/or momentum dimensions that are needed to describe the radiation fields. We also

have six different types of radiation corresponding to six neutrino species. One of the key aspects of our efforts to uncover the supernova explosion mechanism is to understand how the radiation flows interact with the matter. Thus we need the ability to find ways to visually represent the interaction of the high- (4–7) dimensional radiation field with the chemically complex fluid-flow of matter. Most visualization tools are geared toward visualizing either scalar or vector data through the use of a combination of isosurfacing and slicing interfaces. Such techniques are well suited for hydrodynamic simulations. However, for visualization of radiation-hydrodynamic simulations, it would be highly desirable to find some means to represent the radiation flow, which is simultaneously a function of spatial position, direction, and radiation wavelength or energy.

3.6 Terascale Numerical Relativity Using Cactus

John Shalf, Lawrence Berkeley National Laboratory/NERSC, for Ed Seidel, Albert Einstein Institute

The 2002 Big Splash Black Hole Merger Simulation at NERSC has been granted an allocation of 700,000 CPU hours on the NERSC IBM SP machine to compute the largest and most accurate simulations of black hole mergers ever performed. By running simulations on 2–4 TB configurations of the machine, we are able to perform 3D simulations on grids of roughly 1024 × 1024 × 512 or more, far larger than anything done previously. Such a resolution will enable us to follow the entire evolution through orbital decay and merger, and to compute detailed gravitational wave output emitted, starting from the innermost stable circular orbit (Fig. 9).

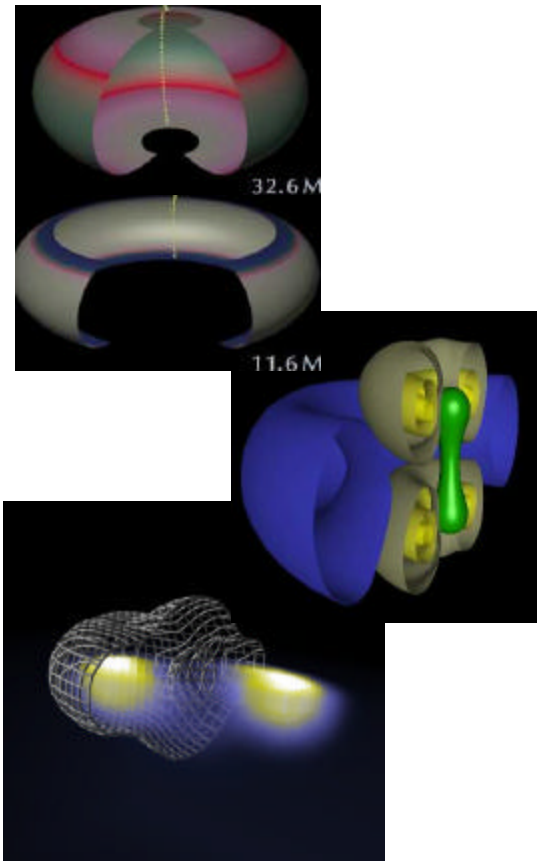


Figure 9. Visualization from black hole merger simulations.

As gravitational wave detectors, such as LIGO and GEO, will be taking data this year for the first time, and black holes are considered to be the most promising candidates to produce signals that are detectable for this generation of detectors, these calculations could be most timely and important.

Our simulation is performed using the Cactus framework, which is already being used to perform smaller versions of such calculations at other centers. As our calculations are memory bound, and as the NERSC

machine has a much larger capacity than any other accessible to us, a new class of simulations of a breakthrough nature may be possible there. Preliminary benchmarks indicate for a fiducial simulation of Einstein's equations, used to benchmark various machines, we can achieve over 300 Mflop/s per processor, and in scaling tests across 256 processors (the largest tested so far) we achieve 85% scaling. Based on tests on many platforms, we are confident that further tests will show similar scaling across the entire SP machine. On the entire machine, this would lead to a performance figure of over 3 Tflop/s sustained performance on our benchmark code.

Cactus is a freely available, modular, portable and manageable environment for collaborative development of parallel, high-performance, multi-dimensional simulations. Cactus originated in 1997 as a code for numerical relativity, following a long line of codes developed in Ed Seidel's research groups at the National Center for Supercomputing Applications and recently the Albert Einstein Institute. The modular component architecture supports multi-language development in C, C++, and F90/F77. Cactus is trivially grid-enabled, which means that it supports use of several low-level Grid services in order to achieve the ability to run distributed, parallel simulations codes. Cactus is an open-source framework that is distributed under the terms of the GNU GPL license. It is in active development and supports a number of contemporary HPC architectures, including IBM SP2, Cray T3E, Hitachi SR8000-F, NEC SX-5, Intel Linux IA32/IA64, Windows NT, Mac OSX, HP Exemplar, Sun Solaris, SGI Origin (n32/64), and Dec Alpha.

3.7 Direct Numerical Simulation of Turbulent Combustion with Detailed Chemistry

Jacqueline Chen, Sandia National Laboratories CA, Combustion Research Facility

Our group performs high-fidelity computer-based observations of the microphysics of turbulent combustion. Our code is a F90 MPP 3D DNS code (S3D) that scales to thousands of processors. It has been ported to most current MPP architectures, including the IBM SP2, Compaq SC, SGI Origin, and Cray T3E. The code performs computations on uniform grids as well as on adaptive meshes. Grid sizes are typically 1.8 M grid cells in 2D (1344 × 1344), and 64 M grid cells in 3D (400 × 400 × 400). Each run typically computes values for 10–20 species. Data files generated by the simulation are on the order of 200 MB for 2D runs, and 7 GB for 3D runs, per time step. There are on the order of 100–200 restart files that consume anywhere from 1–5 TB of storage.

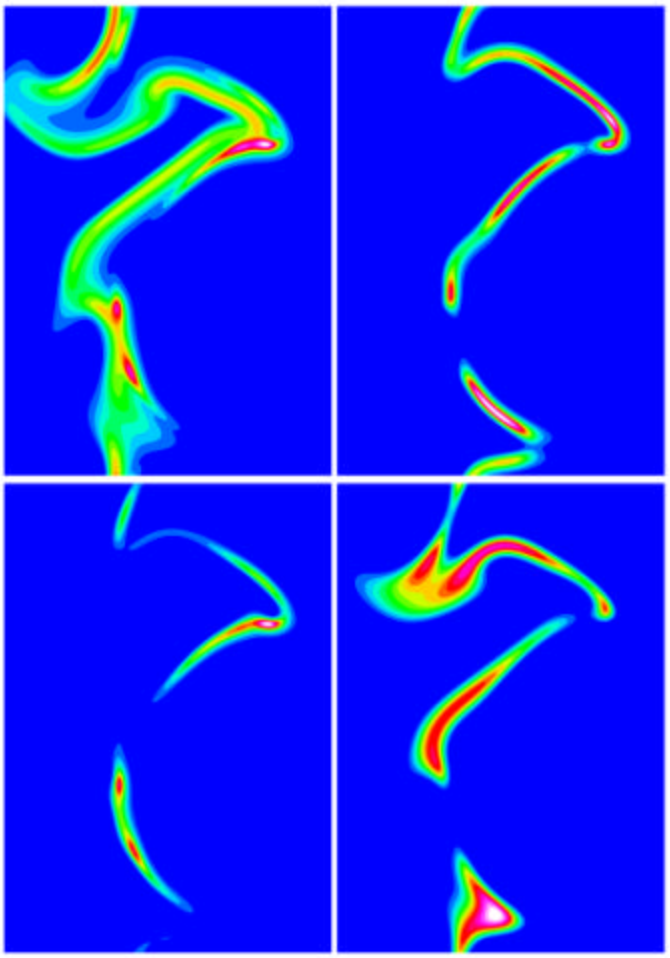


Figure 10. Turbulent methane-air diffusion flame (J. Chen, SNL-CA).

Our project can benefit from interactive visualization, analysis and control capabilities. Interactive data mining is useful to drill into data, effectively navigating through multiple resolutions, down to the finest resolution available. Simulation steering would be useful to interactively explore the effect of parameter changes upon simulation evolution. Interactive and multiresolution feature detection and tracking capabilities would be useful, as well. In order to be compatible with our codes, any such capabilities must be compatible with C or F90, preferably callable as subroutines directly from our codes.

As data sets grow in size, having the ability to perform efficient, out-of-core post-processing will become increasingly important. These non-interactive techniques will perform analysis and visualization of full and multi-resolution representations of data, and should be automated as much as possible. Having these tools operate directly on data that resides on secondary and tertiary storage would be useful to minimize the amount of manual setup required in order to use these techniques. Remote visualization — streaming graphics or visualization over the network — grows increasingly important as data sizes grow.

Although the graphics and processing capabilities of desktop machines continue to increase, what is missing from the desktop are high capacity I/O systems, such as RAID arrays and high performance networking. Having high performance graphics and visualization systems equipped with direct access to high performance storage and networking systems should be a priority for computing centers in order to provide the best possible remote visualization service to its customers.
We have the need to perform visualization of many different types of data. We need to visualize both scalar and vector/tensor data, as well as the visualization of many variables per grid cell (10–100 species per cell). We often have the need to display derived quantities, such as measures of error, concurrent with other types of visualization. We need to be able to simultaneously display computed (simulated) results concurrent with experimental data.

3.8 Visualization of a Global 0.1º POP Ocean Circulation Simulation

Mathew Maltrud, Los Alamos National Laboratory

The Parallel Ocean Program (POP) was developed at LANL by R. Smith, J. Dukowicz, R. Malone, M. Maltrud, and P. Jones. It is a descendant of the Bryan-Cox style z-level model which uses a Eulerian, finite difference method with a structured grid and no nesting. The code is written in Fortran90, uses a 2D domain decomposition, and is designed for parallel computers using both MPI and OpenMP parallel environments. In general, the primary objective of the POP model is to compute the time evolution of the 3D distribution of currents, temperature, and salinity (and other species such as CO_2 or phytoplankton, if desired). Simulations can be run for thousands of model years at fairly low resolution (e.g., for climate research), or at very high resolution for a few decades (e.g., for understanding the interaction of the large-scale circulation with turbulent eddies). At NERSC, Dr. Julie McClean of the Naval Postgraduate School, Monterey, and I are performing a simulation that fits into the latter category.

For this very high-resolution simulation, the grid is fully global (including the Arctic), consisting of 3600 × 2400 × 40 grid points. The horizontal resolution varies from about 11 km near the equator to about 3 km in the high latitudes, so we can resolve narrow currents and eddies, and thus we hope to obtain the best possible representation of the ocean circulation. However, it requires the most powerful supercomputers in order to achieve this targeted resolution. Simulation runs consume significant amounts of processing power: 500 processors at the Naval Research Laboratory Stennis Space Center SP, 448 processors at the NERSC SP. These runs take 8.5 days per model year, and currently generate ~700 GB per model year of output. In the near future, we expect to add more output variables producing close to 1 TB per model year on runs that span 20 to 30 years.

Our current strategy for meeting visualization needs favors in-house tools, and those available free of charge. Commercial products are typically too costly, do not provide all of the desired features, and often have trouble with large data sizes. Commercial products

such as IDL, Ensight, and AVS are also, in a sense, “too versatile,” and require a significant amount of effort to learn and to tailor to our needs.

One of the freeware tools we use extensively is called Ferret (http://ferret.wrc.noaa.gov/Ferret/). Ferret is free, easy to use, designed for oceanographic and atmospheric applications, and is used to generate false-colored 2D slices and 1D plots (Fig. 11). Ferret is not designed for large datasets, so we are required to perform data subsampling in order to make use of its capabilities.

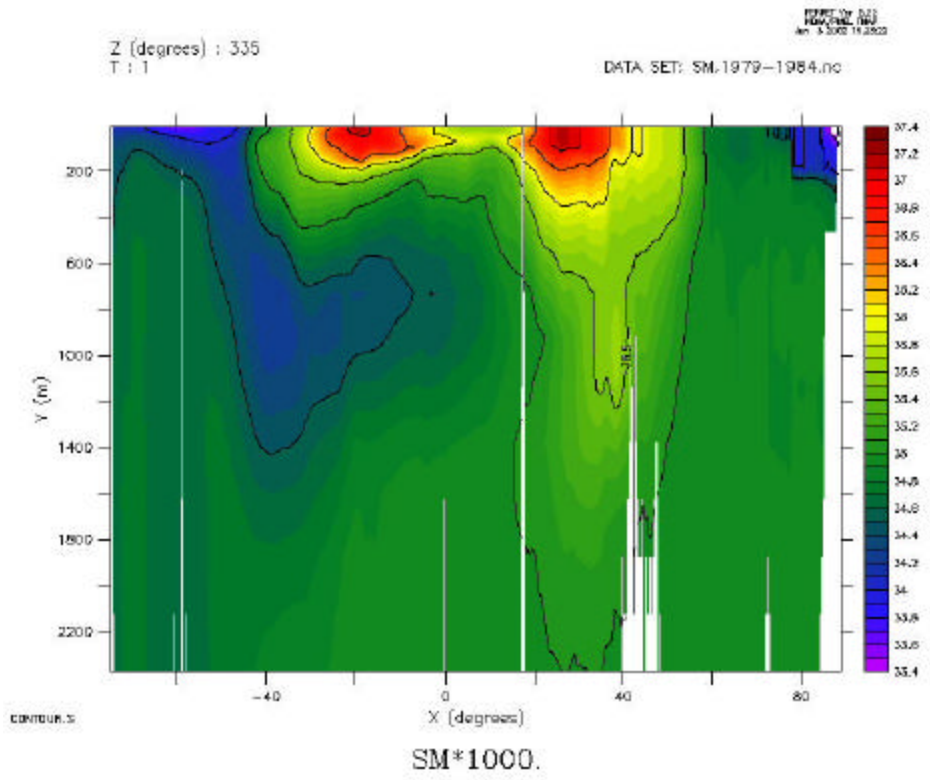


Figure 11. 2D false-colored, contour slice from FERRET (M. Maltrud, LANL).

We also make extensive use of a visualization tool developed at LANL called Poptex. Poptex makes extensive use of features available only on high-end SGI hardware (e.g., Infinite Reality pipes), and as such is not portable. Poptex is tailored for interactive animation of 2D slices, which has proven to be extremely valuable (Figs. 12–13).

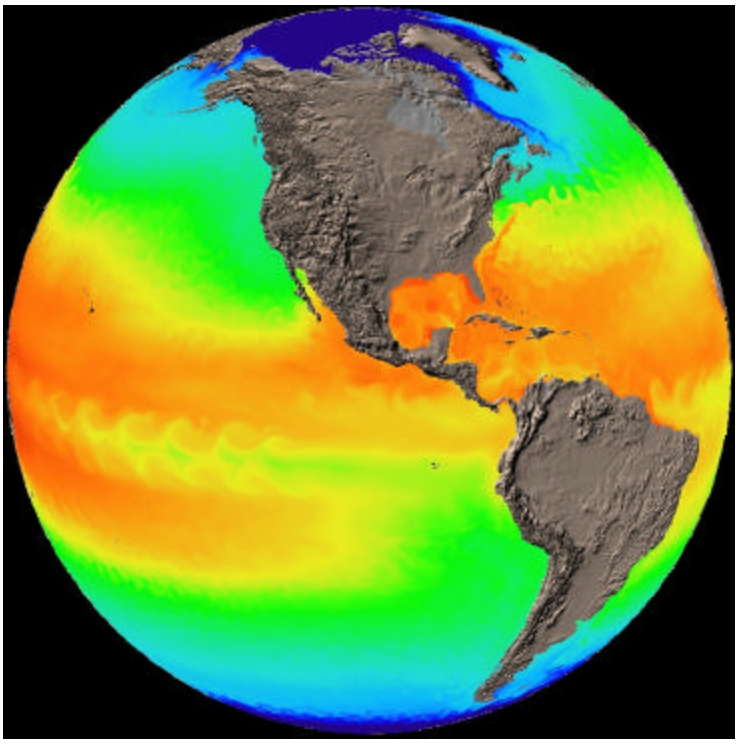


Figure 12. The LANL Poptex tool uses hardware acceleration to provide interactive animation of 2D slices (M. Maltrud, LANL).

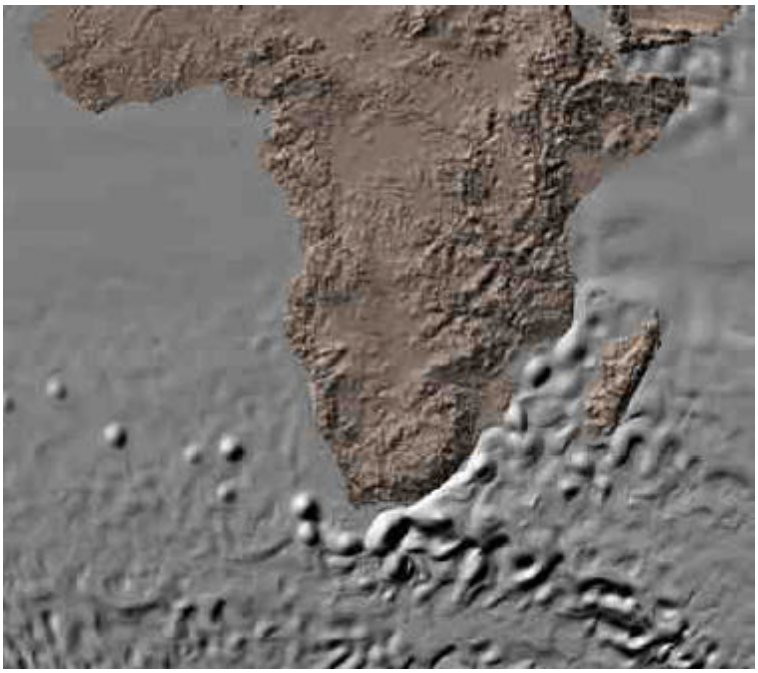


Figure 13. Bump mapping of sea-surface elevation with Poptex (M. Maltrud, LANL).

For 3D visualization needs, we have tested the possibility of using VTK for isosurface generation and viewing, and the in-house tool TRex for volume rendering, though not much progress has been made in this area. Both tools are used on parallel computers, but TRex uses multiple SGI Infinite Reality pipes to hardware-accelerate the volume rendering process, while VTK uses software only.

So, we do have tools that work well for some aspects of our analysis, but we still have problems with large datasets, uncertainties in the future of computer platforms (in particular, the non-portability of Poptex), and the availability of tools and expertise in starting to make advances in 3D visualization.

3.9 Materials Sciences

Lin-Wang Wang and Andrew Canning, Lawrence Berkeley National Laboratory/NERSC

Our basic needs are to plot isosurfaces of charge densities on top of ball-and-stick models of atoms and bonds (Fig. 14). A typical charge density would be on a 100 ? 100 ? 100 real space grid, although these grids can be larger. We would like to be able to rotate, slice, and navigate through our data in interactive fashion. As we move our codes to larger computers, these images become larger. Current dataset sizes are structured grids on the order of 200 ? 200 ? 200 with multiple variables.

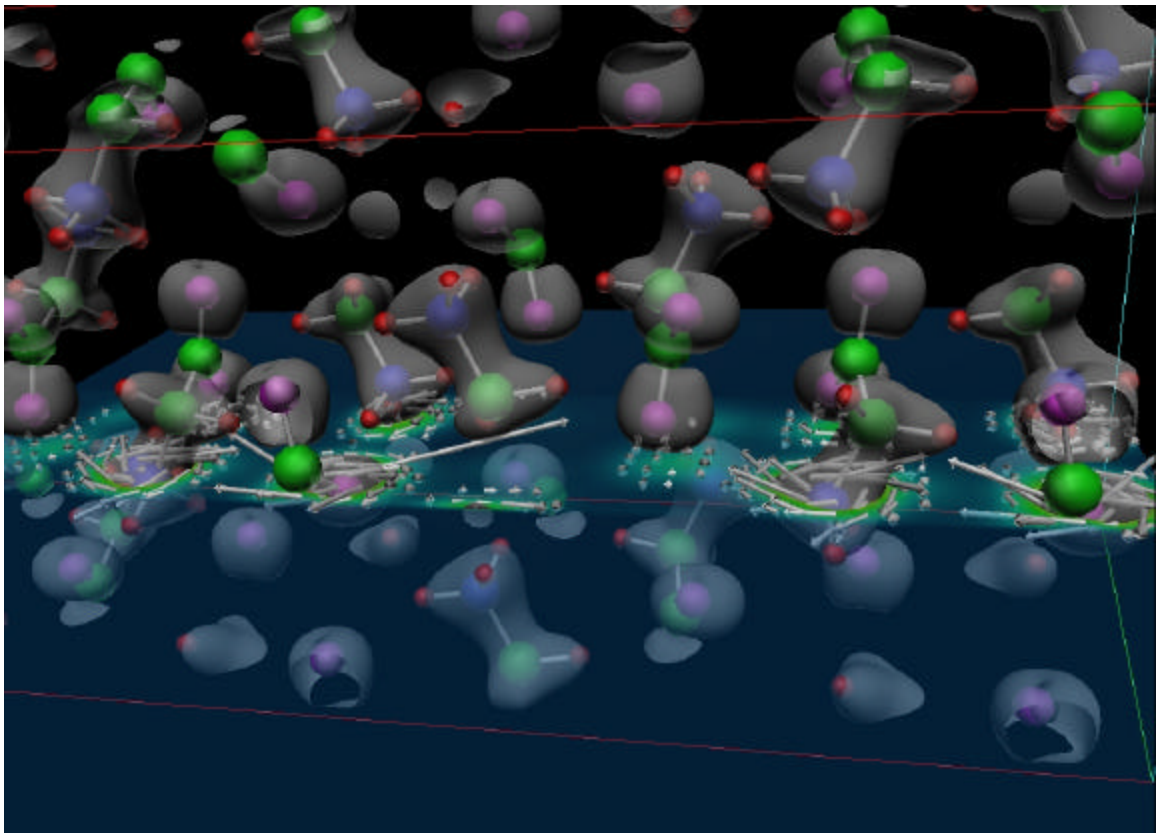


Figure 14. Charge density visualization of glycine molecule.

Our needs are similar to those of others who perform visualization of molecules: display of ball-and-stick molecular models. Our needs are slightly different in that we require display of additional variables, such as electron density, along with the structural representation of molecules. In addition, our molecules are from periodic systems (crystals), and the visualization tool needs to support replication of structural units in a physically meaningful way that is easy to use.

Existing graphics tools such as IBM's Data Explorer, gopenmol, and AVS fulfill these needs to a greater or lesser extent, and require different levels of input from the user to obtain the required visualizations. Some of these tools are limited to ball-and-stick visualizations, and do not support the display of additional fields.

For some simulations it would also be useful to make videos of the evolution of the charge density and atomic positions, and even have real-time output from a code running on computational platforms. We did this at SC97 in San Jose when there was a very fast data link, but normally this is not possible for remote users.

3.10 Clay Mineral Surface Geochemistry

Gary Sposito, Lawrence Berkeley National Laboratory/University of California, Berkeley

A central problem in subsurface hydrology is the detailed mathematical description of water flow through a heterogeneous porous formation with highly irregular layering or lensing of earth materials. This variability is reflected by a pronounced, explicit dependence of the hydraulic conductivity on spatial position at an appropriate local continuum scale. However, few models are available that are not conditioned on specific statistical assumptions about spatial variability. One useful approach to obtaining a more general model is based on the theory of dynamical systems. A dynamical system can be defined as the time-transformation of a set of points in a domain on which a vector field (the groundwater velocity) exists. Thus, a dynamical system describes how moving spatial points flow under the influence of a velocity field, usually by means of an ordinary differential equation featuring time as the independent variable (Lagrangian approach).

Emphasis is placed on the qualitative properties of the flow. The questions posed and settled, therefore, are global in character, applicable to a broad class of groundwater flows. No particular model of the spatial variability of the hydraulic conductivity is considered, and no hypotheses of a stochastic nature are invoked concerning spatial variability. Current topics under investigation include chaotic advection of solutes in groundwater and wave propagation through unsaturated soils.

In both Monte Carlo and molecular dynamics, we often encounter the situation where we are interested in the water network structure (especially first hydration shell) around a cation, anion, or any hydrophobic molecule. In this case, we would like to get the geometry of these water molecules around the particle (Figs. 15–17). So far, we have been using certain equilibrium snapshots where we selectively obtained this structural information by removing all other unnecessary structures (water molecules and clay

structure in this case). This network structure can be both water molecules (freely moving) and clay structure (restricted moving or immobile). Certain types of rectangles, pentagons, and hexagons can be used, or a database of polyhedra (ideal polygons, platonic solids) can be used to describe the clathrate/hydrate structure around the central molecule (cation, anion, and hydrophobic molecules in this case).

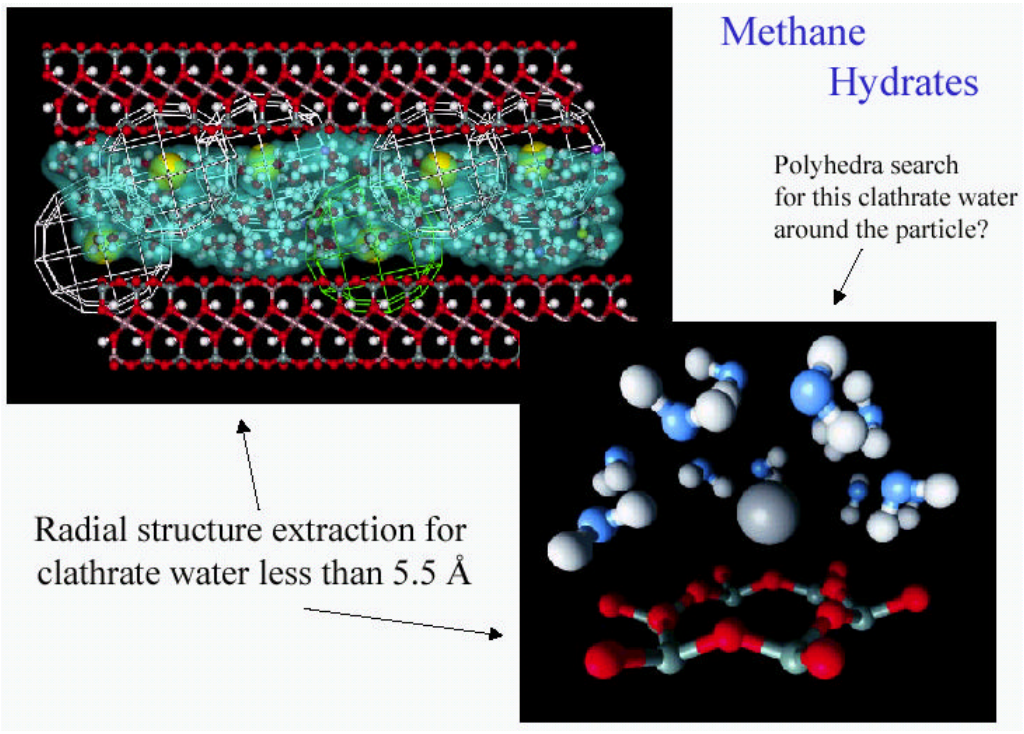


Figure 15. Methane hydrates.

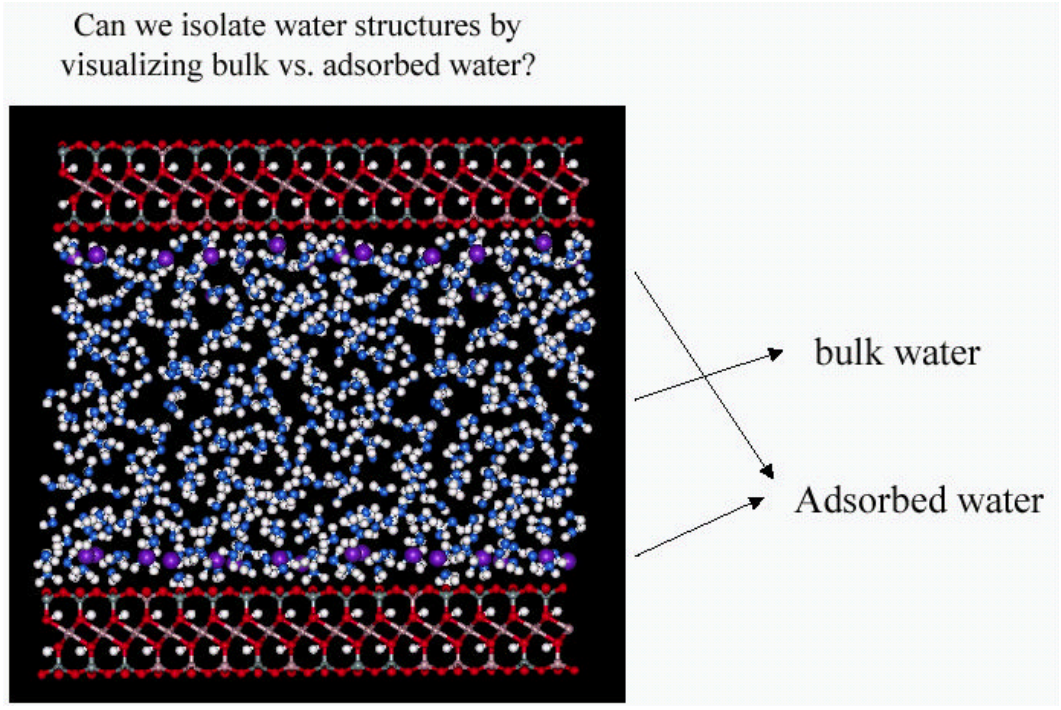


Figure 16. Structure of water adsorbed on a mica surface.

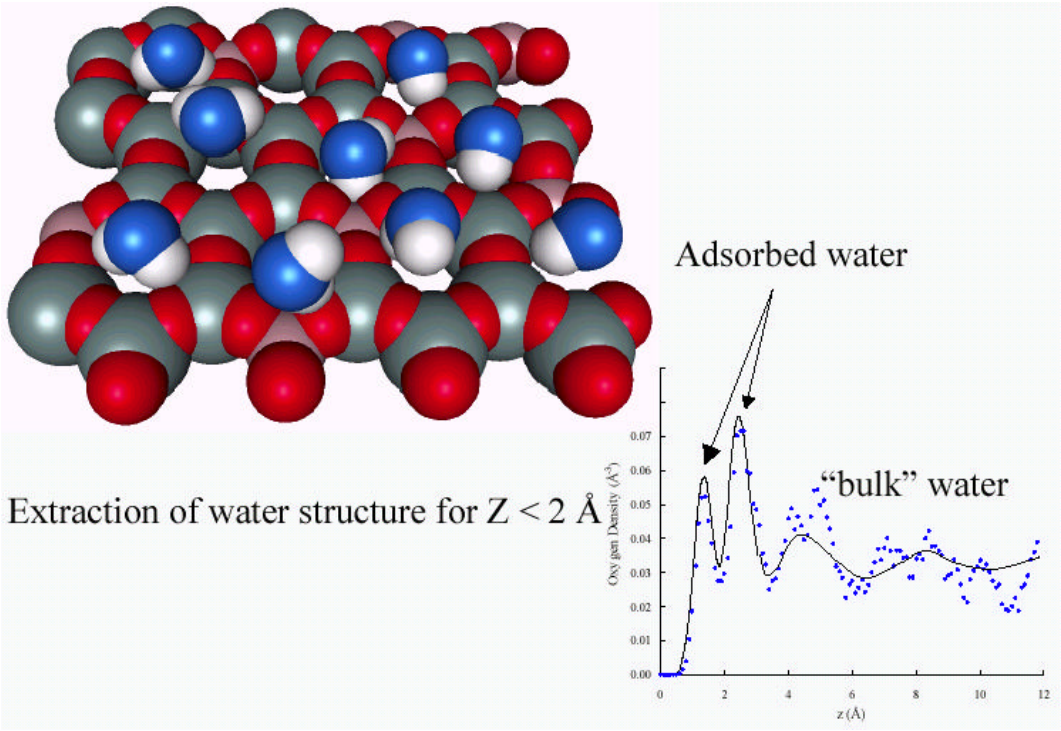


Figure 17. Adsorbed vs. bulk water.

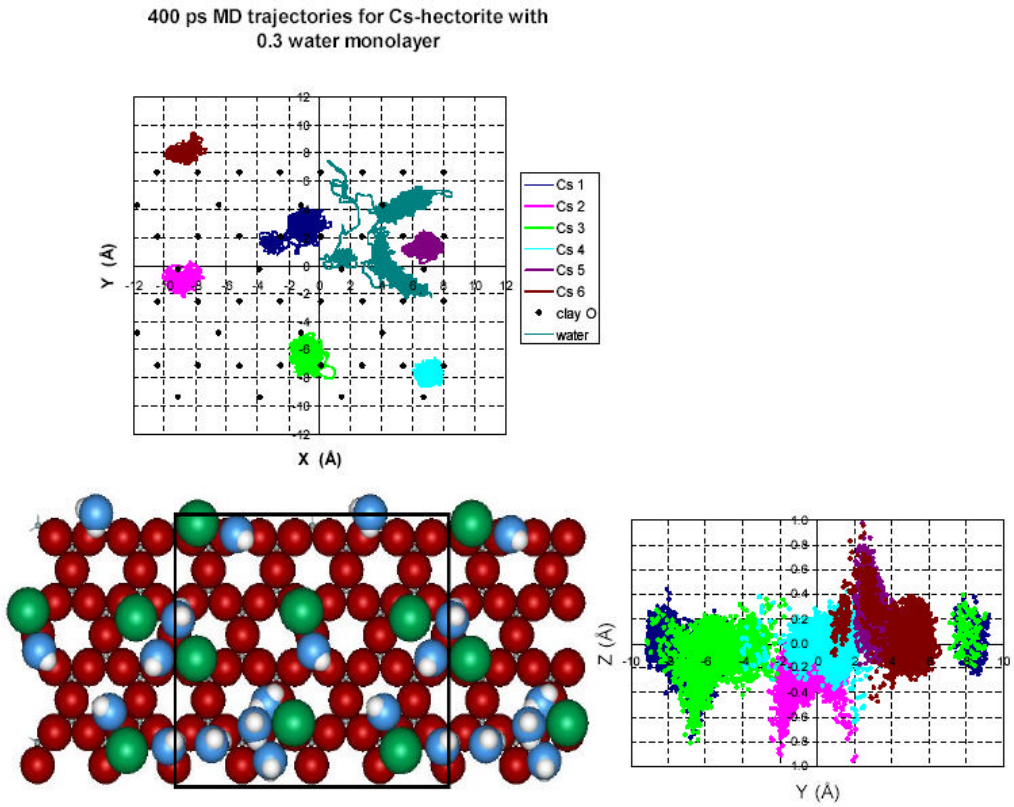


Figure 18. MD trajectories.

Since we are often interested in the surface bound structure of ions and water or at least close proximity of surface (clay, mineral/metal oxides), we would like to isolate the structure of these at certain distance from either surface or ditrigonal cavities in which we have most of our clay mineral simulations. In molecular dynamics, color, sound, and size have an effect on the particle's state of immobilization. Depending on the residency time (time step where the particle does not move for the next trajectory), it would be great to color-code or use a different size description or sound effect (in the case of a movie). Another way of presenting the ion's moving trajectory projected to x/y plane is a 2D contour map for our MD trajectory results (Fig. 18).

3.11 Simulation of Turbulent Flows

John Bell, Lawrence Berkeley National Laboratory/NERSC

Our group performs computational simulations of turbulent flows using a software framework that supports parallelism and a block-structured refinement method known as adaptive mesh refinement (AMR). AMR provides a unique capability of performing simulations at an extremely high spatial resolution, but only in regions of the domain where interesting events are occurring, and without requiring such high spatial resolution across the entire computational domain. AMR grids consist of patches (2D) or blocks (3D) that are locally refined in space and in time (Fig. 19). Each patch is a structured and logically rectangular grid. Grids are dynamically created and destroyed as needed.

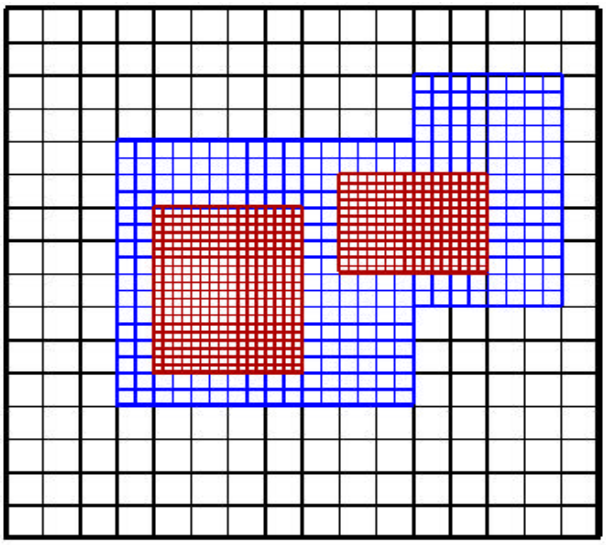


Figure 19. Adaptive mesh refinement grid.

A typical turbulence-flame interaction simulation models premixed CH_4-air flame interaction over time, producing 19 species and 84 chemical reactions. When run at a 256 × 256 × 512 effective resolution over 400 time steps, the simulation produces approximately 2 TB of data.

Visualization is part of the analysis process. Analysis requires computation of additional variables, known as *derived quantities.* These additional variables, which are not produced directly by the simulation, but which are computed through linear combinations of simulation variables saved to disk, may result in an order of magnitude more data than was originally written to disk by the simulation. The derived quantities include reaction rates, diffusion coefficients, thermodynamic properties, optical properties, and so forth.

We require that our analysis framework directly support the AMR grids. We do not want to first flatten the grids to their finest resolution for a number of reasons. These include the fact that flattening the grids produces an unnecessary and artificial inflation in data size. Another is that flattening grids may cause loss of crucial information at the coarse-fine grid boundaries. Analysis and visualization tools are used to debug codes, so loss of information can have a substantial negative impact on the quality of the analysis.

We also require that our analysis framework directly support large datasets. We typically generate large datasets on remote machines, then need to perform the analysis remotely because the data is too large to move across the network.

Our current analysis framework uses a distributed, client-server architecture built from Python and C++ modules (Fig. 20). It incorporates thermochemistry knowledge, such as the computation of relevant derived quantities. It also directly supports hierarchical data, and supports parallel operation. The system is extensible, allowing us to add new modules for simulation or analysis.

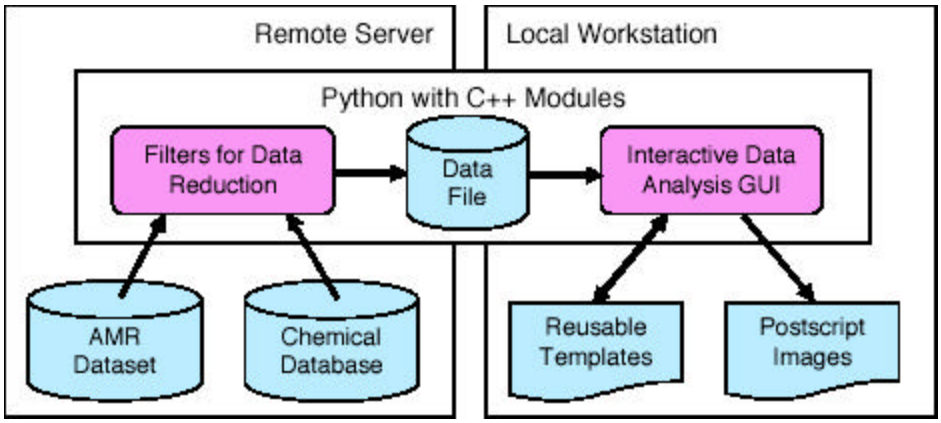


Figure 20. Distributed simulation and analysis framework.

Our current visualization needs include the ability to perform simple x/y plots, false-coloring of slices of AMR data, as well as the ability to track species through complex chemical interactions (Figs. 21–22).

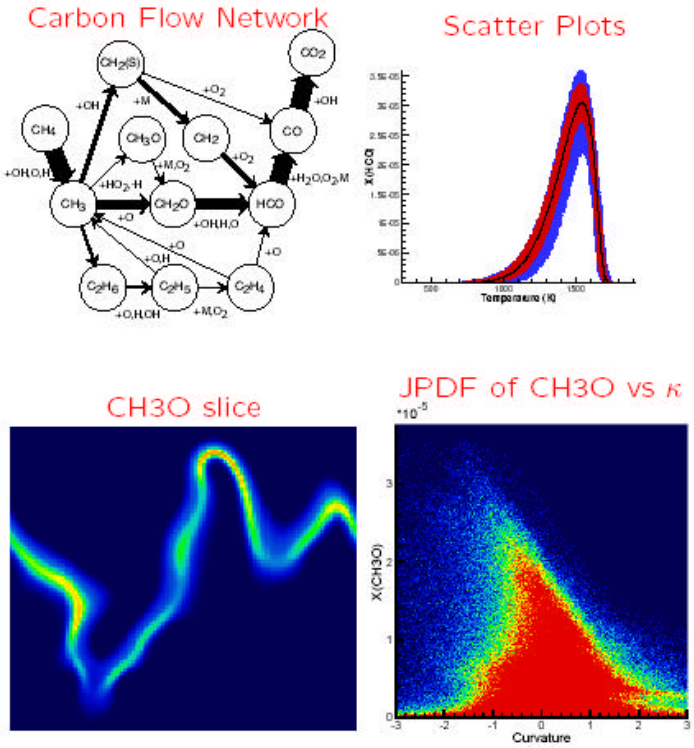


Figure 21. Visualizations in flame simulation.

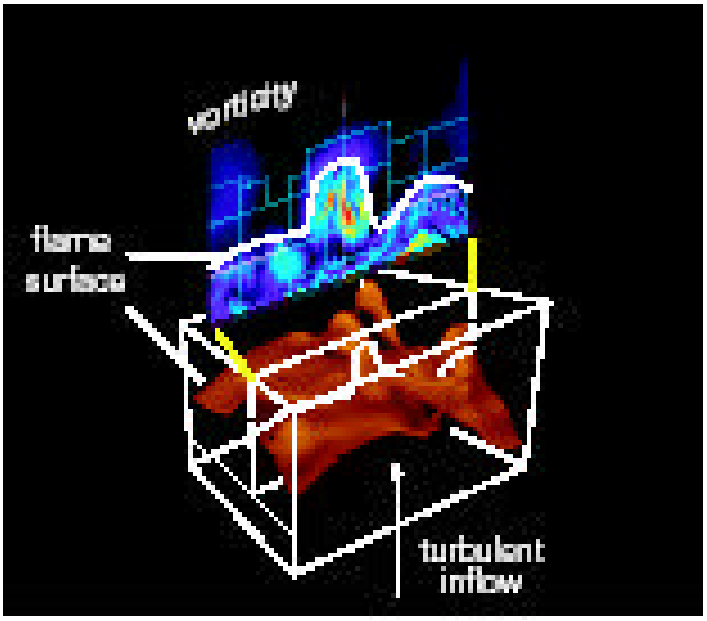


Figure 22. Combination of features.

3.12 Computational Genomics

Inna Dubchak and Dan Rokhsar, Lawrence Berkeley National Laboratory

The difficulty in visualizing and browsing genomic databases has become critical since biologists struggle to interact with current datasets. One such example, Genbank, currently listing 16 billion base pairs, is doubling every six to ten months.

Traditional genomic data visualization and browsing tools often fail because:

- They are based on the WIMP (Windows icons, menus, and pointers) paradigm.
- They force methodology on the viewer rather than offering substance.
- They do not offer a global view of the data, but rather concentrate on the small details.
- Data are represented in a single method without offering multiple perspectives or levels of detail.

Genomes are linear sequences of DNA (a single DNA molecule per chromosome) ranging in length from megabases for microbes to hundreds of megabases and even gigabases for animals and plants. There are several major projects in our group where sophisticated visualization tools are urgently needed. Among them:

- **Tracking and presenting an assembly process** by displaying a raw genomic sequence in a 600-base pair fragment and tracking these small fragments as they are assembled into a complete genome.
- **Annotations of various features of a genome** (genes, parts of genes, other signals controlling turning genes on and off, similarities with other sequences). These features are jumping-off points for additional information/views pertinent to that feature.

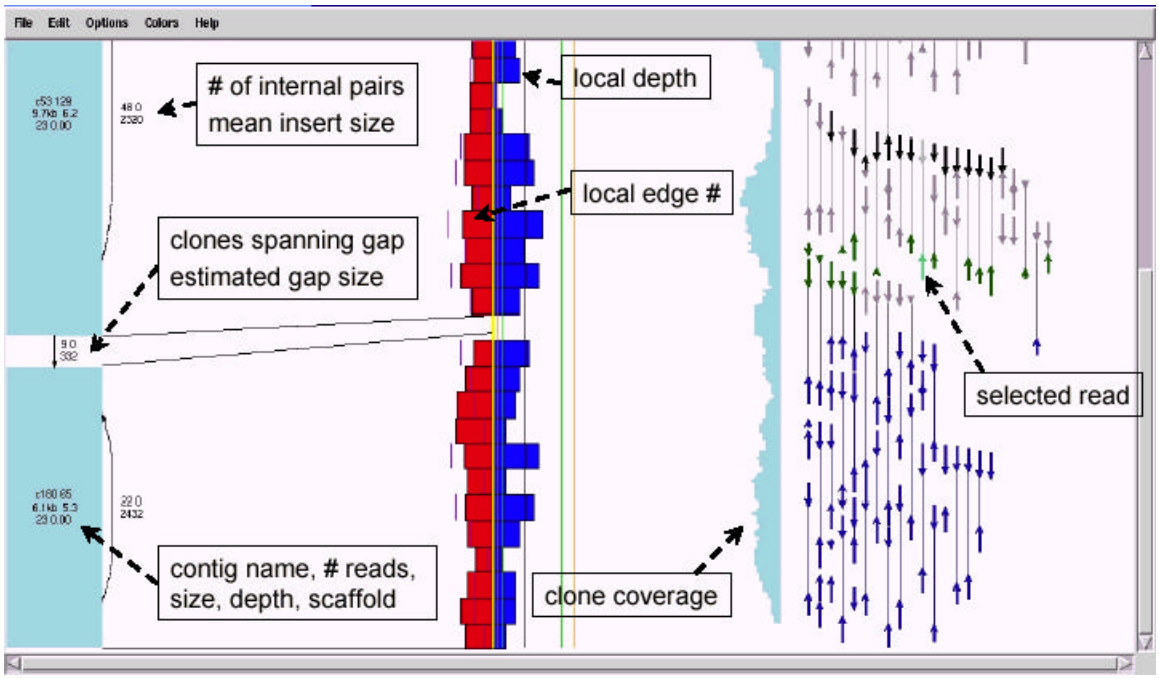


Figure 23. JAZZ view of assembly.

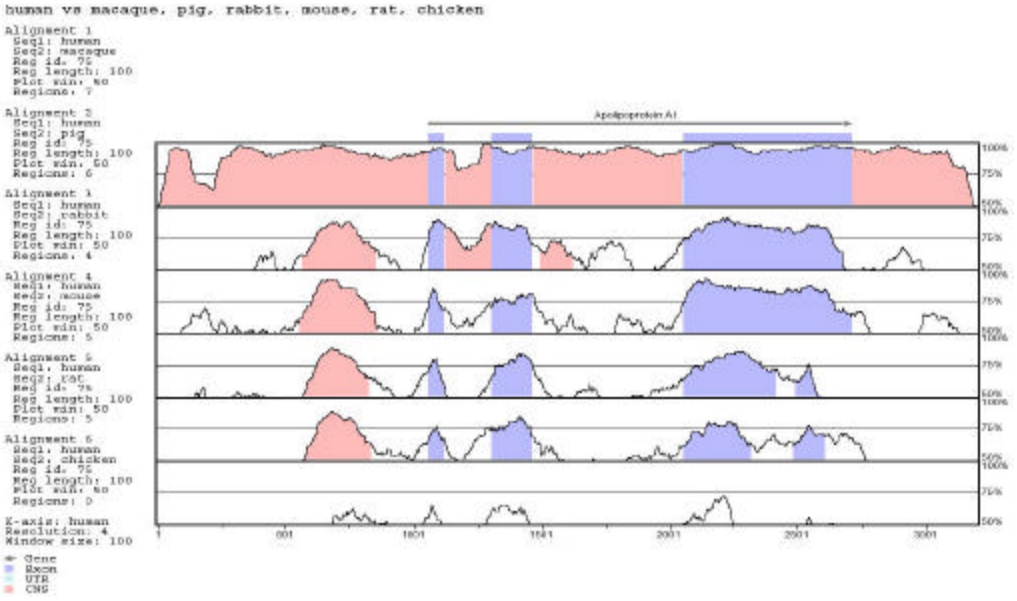


Figure 24. MultiVista Plot. Simultaneous views of multiple feature types at a genomic locus are especially informative for the purpose of biological analysis.

?? **Development of modular, open-source toolkits for common computational and visualization tasks.** Such interoperable tools currently under development in different groups (UCSC, CSH Laboratory, LBNL) are intimately related to the standardization of biological data formats and data structure.

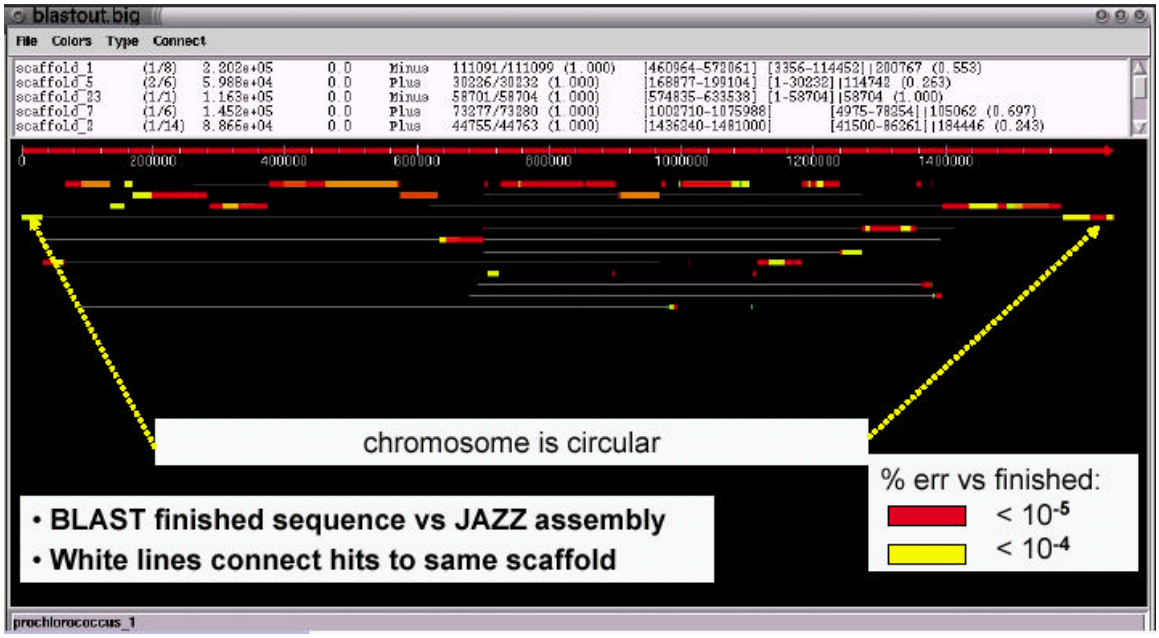


Figure 25. Megabase scale views.

?? **Comparative analysis of genomic sequences**. Biological sequence similarity analysis presents visualization challenges, primarily because of the massive amounts of discrete, multi-dimensional data. Genomic data generated by molecular biologists is first analyzed by algorithms that search for similarities to known sequences in large genomic databases. The output from these algorithms can be several thousand pages of text, which makes it difficult to analyze because of its length and complexity.

We have developed the software tool VISTA for aligning long DNA sequences of two or more species. The VISTA tool can visualize the alignment, sequence annotation, and a number of different sequence features using a concise, user-friendly format . VISTA also includes an alignment program, an efficient visualization algorithm, and a number of analytical modules that allow for the examination of the level of conservation in different regions of genomes. The VISTA family of tools includes several modifications allowing one to analyze a wide range of comparative biological data. VISTA is implemented as a publicly available Web server (http://www-gsd.lbl.gov/vista) which receives ~ 1500 queries per month, and also as a standalone package, with more than 700 copies distributed in academic institutions.

?? **Functional genomics data types** such as 2D or 3D image data, for example from serial thin sections, tomography, and confocal microscopy, are more variable. Color, staining, and tagging of anatomical and cellular features relate specific genes to pathways. Gene expression data can show which genes are turned on or off, when, where and by how much. Navigating, searching, and viewing large datasets are necessary components of any successful analysis. Integration of diverse data types into a small number of access points (for example, accessing genome, gene

expression, etc. from a 3D image of organism) presents a non-trivial computational problem.

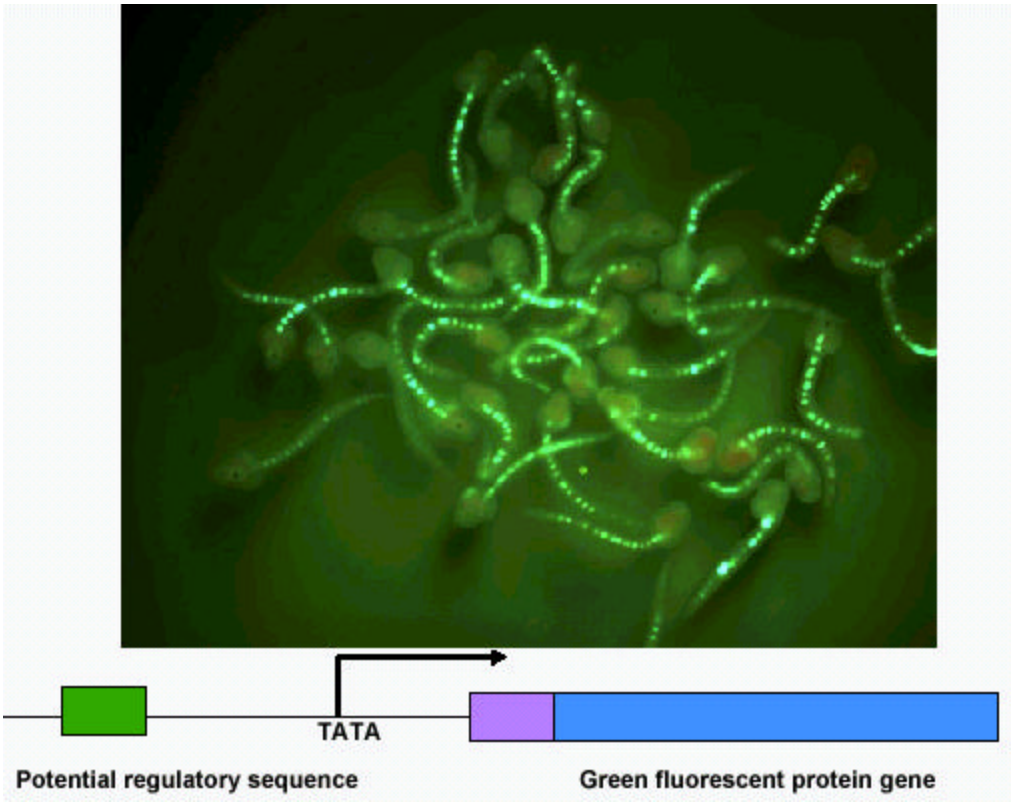


Figure 26. Functional genomics: How and when are genes turned on or off?

4 Future Requirements and Challenges

The previous section presents current requirements from a number of computational science projects hosted at NERSC. In this section, we present the anticipated and projected future visualization requirements for many of these computational science projects.

4.1 Plasma Confinement, Stability, Heating and Optimization in Stellarators and Tokamaks

The main problems are that as we try to move to higher-resolution models, performance becomes poor (especially as the data size exceeds the available on-board cache on the graphics card) and thus discourages further development in this direction. We would also like to add further 3D visualization capabilities (such as stereographic viewing, spaceball control, etc.) to our workstations.

Although we have been able to achieve stereo for static images using inexpensive commodity-based graphics cards, we would like to have full quad-buffered stereo views for interactive work. This is impeded by several factors. First, the hardware is rapidly

evolving and it is difficult for the casual user to find out which stereo-capable graphics cards are really compatible with the needs of the software one is using and whether the appropriate stereo drivers are provided. Second, the higher-end graphics cards that are known to provide adequate stereo capabilities and larger on-board cache sizes become significantly more expensive than the more standard cards (although they are less costly than they have been in the past). Since our program budgets are already stretched thin paying for people, overhead costs, travel, etc., it would be quite helpful if an additional source of funds were available to us for upgrading our desktop visualization workstations. Besides our desktop systems, it would be helpful if we could obtain systems for stereo viewing of projected 3D images by small groups (5–10 people). Also, portable laptop based systems for 3D viewing that could be easily taken to meetings would be desirable.

In general, in both the 3D hardware and software areas, it would be helpful if DOE scientific users could be organized into a more coherent market force (e.g., possibly through more coordinated bulk purchases of hardware and software). For example, at computer graphics conferences, one finds strong user groups from the oil and gas industry, medical imaging, games developers, etc. These groups have been able to influence software and hardware developers into supplying their needs, often at much more favorable cost/performance ratios than they could have obtained as individual users. Although we get some "trickle-down" benefit from the demands of these groups, our needs differ enough from theirs that their solutions are not always very well optimized to our needs.

In the area of software, the main problems relate to the steep learning curves involved in most of the packages mentioned above. We typically learn enough to do several basic types of visualization that we are interested in, but then may find it difficult to move beyond this. Although most software vendors offer courses, these tend to be fairly expensive and often target skill levels that are not optimal for us (either too basic or too advanced). What is often needed are answers to fairly specific questions relating to capabilities we would like to add to our visualizations. It would also be helpful if NERSC would incorporate tutorials for the various scientific visualization packages into their remote training offerings. Some specific capabilities that are desired in visualization software are indicated below:

?? Ray tracers/volume renderers that allow the user to ray trace within oddly shaped geometric regions (e.g., toroidal shaped devices) and not just over a box-shaped Cartesian grid, which will often include a lot of empty space in it if one is only interested in what happens inside the toroidal region.
?? Time series animators that allow one to program in simultaneous volume rotations and translations while the time series animation of some field within the volume is going on.
?? Methods for automatically parsing large, high-resolution datasets onto multi-resolution grids so that the user can both view the big picture as well as be able to drill down into small regions of interest. The software should also provide some form

of automatic pattern resolution/data mining to help the user recognize where such regions of interest are.

?? Intelligent isosurfacing algorithms that only form isosurface geometries down to some prescribed resolution level and then display the remaining, more detailed structures using texture maps so that interactive graphics performance does not suffer.

?? Access to storage (~10 to 100 TB) for doing long time series animations coupled with automatic pattern recognition/data mining software for finding interesting phenomena in these animations.

?? Develop computational steering capabilities. Currently this is not useful since one has little control over when a job will run. Some level of dedicated, massively parallel computing should be provided where users can schedule their runs at times convenient to them for online interactive computational steering. Access to higher-speed network connections during these times may also be necessary.

?? Our stellarator optimization project involves higher-dimensional spaces (30–40 independent variables at a minimum). Although one can always take 2D or 3D slices of this type of data, better methods for understanding such high-dimensional spaces are needed. Also, plasma kinetic calculations typically involve 5D spaces.

In addition to the capabilities of visualization software, learning how to use the tools is also an issue. Although training material is available on the Web for most existing scientific visualization software packages, there is so much of it that it is often difficult for new users to know where to start and difficult for experienced users to find answers to specific questions. Providing teleconferenced lectures for new users as well as online consulting is of help. Also, my experience has been that two- to three-day hands-on courses are very helpful. NERSC should try to incorporate this into their regular training courses.

As indicated in the previous sections, we need to develop the capability to visualize larger datasets (in the range from 1 GB to 1 TB). We also need to develop a more diverse set of capabilities in our use of software; this will involve better user education and access to expert users. We need better desktop hardware to work with; for example, graphics cards with more on-board memory cache (>128 MB), support for stereographic viewing, etc. Some of our graphics needs might also best be solved by having occasional access to immersive (e.g., CAVE) systems. We will need to understand what types of software and data file formats need to be used to port our visualizations over to these systems.

4.2 Accelerator Design

With the advent of multi-teraflop/s computers and "ultra-computing" on systems that achieve 50 or more teraflop/s, there will be a crucial need for new visualization capabilities to handle the associated large and complicated datasets. Simulations involving billions of particles or mesh points will involve 10–100 GB of data per frame, and multi-time-step datasets will reach 10–100 TB or more. For the accelerator modeler, remote visualization will be the key to viewing and analyzing the results without having to move massive amounts of data from a remote site to a local site. In addition, a further

key requirement will be the need to visualize multiple data fields simultaneously. For example, Fig. 27 shows particle data superimposed on a tetrahedral mesh.

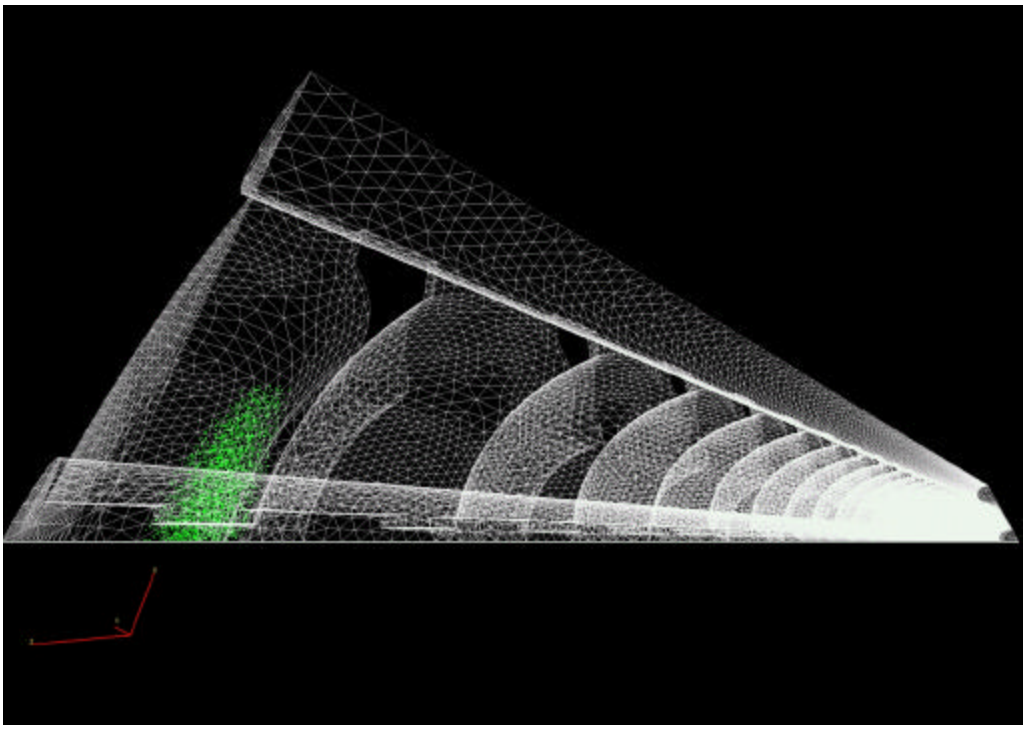


Figure 27. Particle tracking in a tetrahedral mesh.

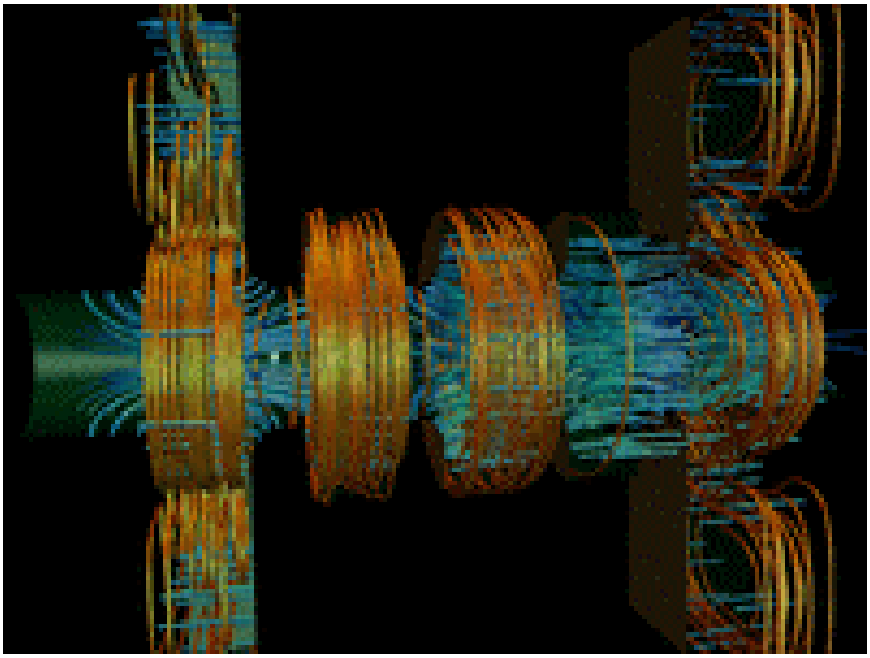


Figure 28. Electric field lines (blue) and magnetic field lines (gold) in an accelerator structure.

Other examples of multi-data techniques include the combination of particle data and volume rendering, and the combination of both electric field data and magnetic field data in a single graphic (Fig. 28). Eventually it will be necessary to include particle information along with the field information in a single image. Still other examples include the combination of vector data and scalar data (e.g., electric field vectors plotted along with temperature rise data), or particle data in combination with volume rendering.

Besides the visualization technologies themselves, accelerator designers also need collaboration tools and technologies to enable geographically distributed team members to work together to design and optimize accelerator components and accelerator systems.

4.3 Supernova Initiative

Collaborative Interaction and Analysis Tools. The TSI has a geographically distributed set of researchers, and it would be valuable to have the capability for these researchers to engage in interactive data analysis sessions. For this reason we are strongly interested in collaborative technologies that would allow at least one participant in a session to steer a visualization engine while other participants remotely observe this same session from their own workstations. It would also be highly desirable to have such a tool possess an electronic whiteboard capability so that users could be able to collaboratively engage in discussions about what they are visualizing. Our wish-list for such a collaborative tool would include the ability to step through a sequence of time-varying datasets in order to view the evolution of a simulation.

Distributed-Memory Parallel Visualization Tools. Most of the computation under the TSI initiative involves long-timescale integrations that are done in batch mode on parallel architectures. Such simulations produce large (>100 GB) datasets that can be difficult to visualize using single-processor visualization engines. Furthermore, a single time slice in such a dataset could be quite large, thus making it an excellent candidate for parallel rendering. It would be desirable to have the capability to carry out parallel visualization on local, smaller-scale parallel computing resources such as a Beowulf Linux cluster. We strongly encourage the development of parallel visualization tools that rely on portable parallel visualization software such as parallel VTK. Any software developed for this purpose would be most useful if it supported an interface to a GUI-capable scripting language, such as Python, which could enable the development of custom GUIs that are application specific.

4.4 Terascale Numerical Relativity

In the future, the Cactus Numerical Relativity Community needs to perform simulations of higher resolution, needs to execute more simulation runs with faster throughput, and wants to examine results on the local desktop without having to master new systems or techniques. This community has access to high-end resources in over ten centers located in the USA and in Europe (Fig. 29).

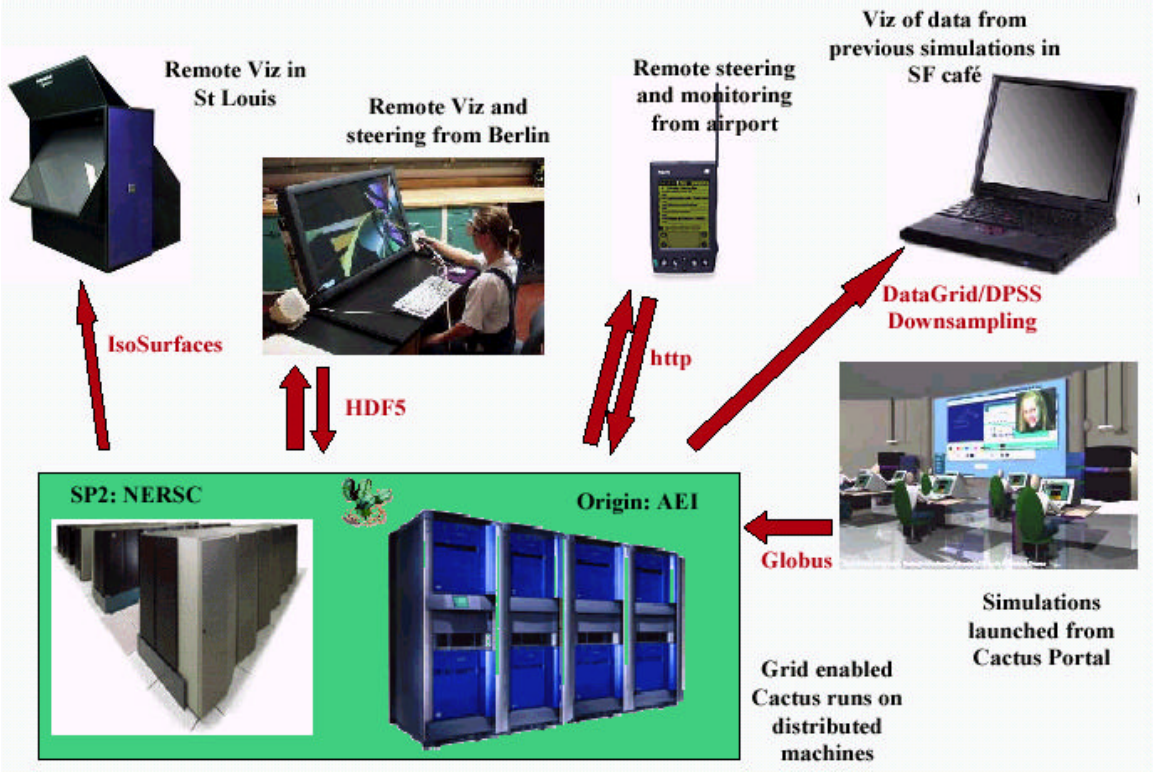


Figure 29. Grid-enabled Cactus.

In order to make these remote and distributed resources available to researchers, a significant investment in infrastructure is required in order to provide easier access to the resources. In the present state, users who wish to employ diverse resources are required to remember login information for each platform, understand idiosyncrasies of each different batch system and file system, understand details of network topology and performance, and so forth. In the future, users will want to combine resources for larger production runs in order to achieve higher resolution, which is an acute requirement for better computational science. The environment will evolve to include dynamic scenarios in which the application will automatically use whatever resources are available. As Grid-enabled applications evolve, the need for pervasive remote and distributed visualization tools will become more acute.

A *portal* is a Web-based interface to a collection of resources. A portal should hide and simplify use of the Grid for users. The portal provides a single point of access, locates resources, builds/finds executables, provides central management of parameter files/job output, submit jobs to local batch queues, tracks active jobs, and provides submission and management of distributed runs.

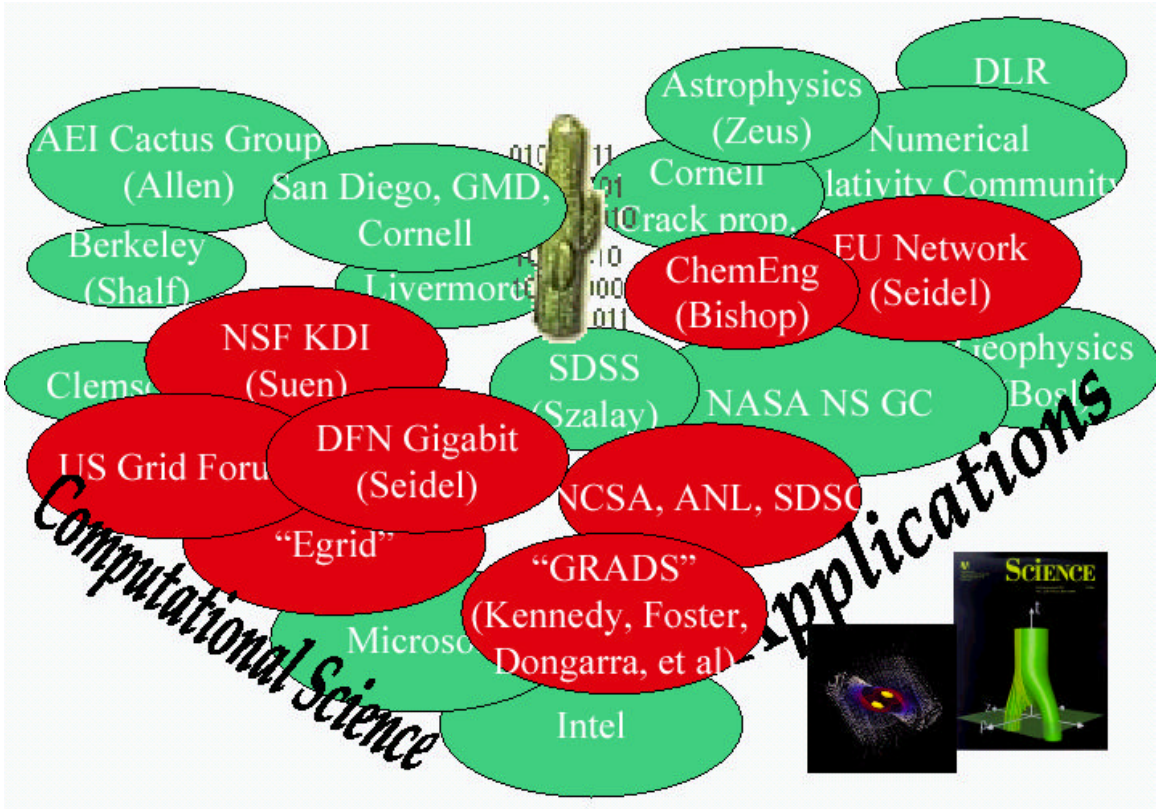


Figure 30. The Cactus community.

There are three broad architectures used to deliver services to a user through a portal interface. A *thin client* provides dynamic HTML to a Web browser and will achieve the highest level of portability, as only a Web browser is required on the user's machine. No user time is required to install or configure software. A *slender client* is implemented using Java applets or signed Java applications. A significant gain of slender clients over thin clients is the ability to leverage some of the processing capabilities on the user desktop, as the slender client consists of executable code. The slender client model may require download of the executable on each invocation, but allows for central management (on the server side) of software distributions. Users need not be concerned with making sure the slender client code is kept up to date, but may be required to ensure their Java Virtual Machine is kept up to date. The third category of portal applications is the *fat client*, which consists of a native executable that is invoked using the MIME-types association within the browser. In this case, the portal is merely a data broker between distributed resources and the helper application. Fat clients provide the greatest degree of performance, but also require the greatest amount of user time to set up and install the helper application.

As simulations increase in size, the need for tools to help understand data becomes more acute. Larger simulation runs mean larger spatial dynamic range. Understanding the connection between large-scale and small-scale features is critical. Larger simulations take advantage of new types of grids and data structures in order to achieve higher resolution. Such data structures include adaptive mesh refinement hierarchies.

Computational monitoring is important, especially for large and long-running simulations. Rapid visual inspection for quick turnaround during development is needed to maximize productive use of resources. The best way to deal with big data is to move it as little as possible, which implies an increasingly acute need for remote and distributed visualization tools that will perform their tasks close to the data and deliver images to the remote user. Offline analysis is also important, but may involve a completely different set of tools and methods. Physicists have little tolerance for complex or difficult-to-install software. Such a disposition motivates the need for visualization portals and thin-client interfaces to visualization tools.

Higher data dimensionality implies the need for more qualitative tools: 1D visualization is still a critical element in a visualization portfolio. The most effective visualization tools are those that are customized for the domain. General-purpose tools often have too many options, making them confusing and unwieldy.

4.5 Combustion with Detailed Chemistry

In the future, our visualization requirements are similar to today's, but with an increase in capabilities. We will need to perform analysis and visualization of ever-larger datasets on large, parallel architectures. We would like to see visualization more closely integrated into our analysis framework, rather than as an add-on module. We will still require simple x/y plots, but also want the ability to perform volume rendering and generate isosurfaces on large, parallel machines. Even today's most powerful workstations do not have adequate processing capacity for us to perform interactive visualization of our large datasets. Large parallel machines at supercomputing centers are most often operated only in batch mode in order to maximize throughput for the entire user community. They are not operated in a manner that would support large, interactive jobs such as a large, parallel analysis or visualization session. Lack of large, interactive parallel resources is a hindrance to our work.

Another problem we face is the sheer volume of data generated by our simulations. Such large data is difficult to store, difficult to move, and difficult to operate on. Often we are not sure exactly what we are looking for in the data. Serendipitous discovery of interesting features is increasingly impractical as datasets grow in size. Our fantasy is to have the ability to launch intelligent agents that can locate interesting things in the data. The trouble is that it is difficult to define "interesting." We would like to have help from the analysis and visualization community to provide new tools that are capable of finding and reporting interesting features in our large datasets.

4.6 Global Climate Modeling

In the future, we are most interested in two things: 3D visualization, and techniques for visualization of flow fields. For 3D, we plan to explore VTK and TRex more fully, as well as immersive environments, but this will likely move slowly unless funding sources are found. For flow fields, we need both direct and indirect visualization techniques. Direct techniques are those that show data directly, such as a direct representation of each and every velocity vector. Indirect techniques are those the show a property computed from the data, such as streamlines and particle advection. Streamlines and fronts are

expensive to compute and can be difficult to understand, but are useful. We currently use a particle viewer that uses precomputed trajectories for advection, but it is limited. We are also exploring the use of color, intensity, and opacity to show information about a vector field (Fig. 31, courtesy of Naval Research Lab, Stennis). We are also interested in exploring simultaneous visualization and display of data from separate codes as well as from multiple simulation runs. Although we do get some background help from the visualization personnel at LANL, they are not currently funded to help us with these issues.

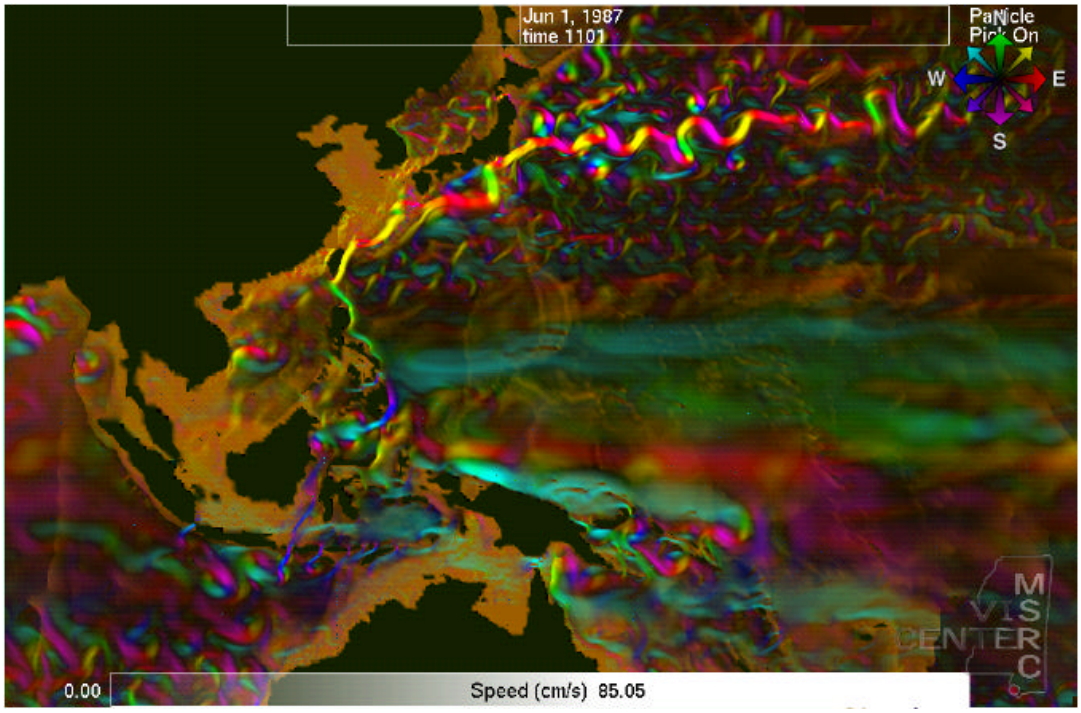


Figure 31. Vector field visualization (M. Maltrud, LANL).

4.7 Computational Genomics

Vertebrate genomes are typically composed of approximately three billion base pairs. They exhibit a remarkable degree of common structure across species. The areas containing gene-coding data are interspersed with large intergenic regions, which also exhibit some degree of similarity across species. Application of VISTA to a whole-genome scale project presents a nontrivial computational challenge complicated by two facts: (1) the sequencing projects are continuing at a fast pace, and thus there is an ever increasing amount of data to deal with; (2) even advanced alignment and comparison tools require significant computer time and memory to run. We have addressed these challenges by building a fully automated pipeline based on the following algorithmic components with associated software: a global alignment program optimized for fast and accurate alignment of large genomic regions in both finished and draft format; efficient homology mapping scheme; tools for discovery of conserved sequences; and a comprehensive suite of visualization options on a genomic scale.

In the future there are several directions to be pursued, including development of interactive tools to fully analyze long regions across more than two species to identify

and characterize similarities and dissimilarities. The next generation of the pair-wise VISTA Genome Browser currently implemented at LBNL (Fig. 32) will allow one to browse multiple species alignments with associated annotation features on the whole-genome scale. The overall goal is the development of visualization techniques that facilitate interactive sequence data exploration, analysis, and understanding. Instead of comparing just a pair of aligned genomic sequences, a user will be able to align several sequence datasets and define statistical measures for sequence similarity across multiple species. Again, a user would be able to visually explore the similarity function/plot using scalable resolution.

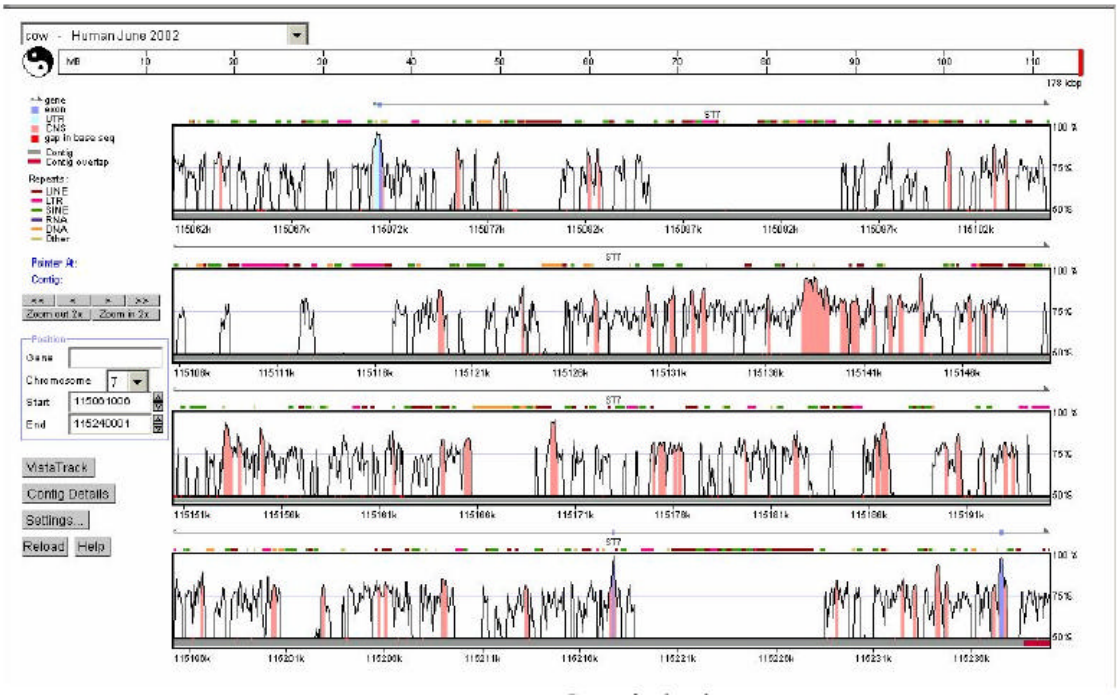


Figure 32. Screenshot of VISTA Genome Browser display of the ST7 gene region in the alignment of the whole human and mouse assemblies.

5 Findings and Recommendations

Finding: Remote visualization is becoming increasingly important. Science, and computational science in particular, is becoming more and more distributed. Massive datasets should not and cannot be duplicated in their entirety at all sites involved in large-scale scientific (computing) experiments. Furthermore, network restrictions do not support efficient-enough transfer of data. Remote and distributed visualization approaches must be developed that support easy and real-time data exploration over current networks, where powerful, centralized servers (supercomputers, clusters, clusters of clusters, etc.) perform the time-consuming steps needed for image generation. Portal-based technology promises to be crucial for highly distributed visualization applications, where large numbers of remote users, diverse computing and data servers, and multiple visualization sites must be considered.

Recommendation: ***Establish a coherent program that focuses on remote visualization. A remote visualization program should provide tools and infrastructure that can be used by multiple "virtual teams," such as the Integrated Software Infrastructure Centers (ISICs) within SciDAC.*** Remote visualization allowing distributed teams of scientists to explore their data over high-speed networks is becoming increasingly important. The technology to support this form of collaborative data exploration must be developed now. Substantial improvements are necessary concerning fundamental multi-resolution algorithms, portal environments, distributed and parallel processing applied to massive scientific data for visualization, and networking infrastructure. The required research and development to be done will lead to next-generation visualization environments supporting the computational science community at large.

Finding: Easy-to-install, easy-to-use, easy-to-maintain visualization software is desired by the user community. Lack of funds and lack of time on the users' side are two main reasons for this requirement. For visualization technology to be acceptable by most users, it must not require a large amount of the researcher's time to use and learn, including the amount of time that has to be spent to produce "good pictures." Visualization techniques, when used interactively, must support real-time rendering.

There is a need to communicate capabilities of current visualization technology to the users as well as to the potential user community. Scientists generally cannot be expected to be aware of the latest state of the art in visualization technology, and more technology transfer is needed. This transfer could be done, for example, via two- to three-day short courses at major computing centers, including hands-on tutorials.

Large computing centers should have the responsibility to develop, deploy, and support visualization technology for the community of users. Highly customized solutions as well as more generally applicable technologies should be the centers' responsibility.

While data visualization is a crucial component of computational science applications, computational scientists who are not experts in data visualization themselves can exploit the power of modern visualization techniques only if (1) it is simple to install and maintain the required hardware (including stereo and immersive visualization environments) and software; (2) it is easy to learn how to use visualization software; (3) it is possible to convert between scientific data formats; (4) costs for visualization technology/services are low; and (5) centers at national laboratories and elsewhere provide hands-on short courses concerned with the latest in production-level visualization systems. DOE should consider establishing mechanisms whereby generally applicable visualization technology is developed and deployed in a rather centralized fashion, with clear emphasis on the evolving requirements posed by scientific applications.

Recommendation: ***Establish mechanisms whereby generally applicable visualization technology is developed and deployed in a centralized fashion.*** Centralized coordination will simplify access by researchers, as well as distribution for developers. Related, more programmatic mechanisms are needed to fund product development and support of the most promising prototypes from the research programs.

The development, debugging, and maintenance of computer simulation codes is greatly enhanced by crude, mostly automatic and simple visualization methods that enable a computational scientist to more rapidly identify flaws in simulation codes. Such tools should support viewing variables of an ongoing simulation, interactive adjustment of parameters (geometry, initial/boundary conditions, etc.) controlling a simulation, and querying the current values of certain variables at arbitrary locations. It is also desirable to develop visualization technology that supports the monitoring of an ongoing simulation, coupled with primitive interaction capabilities. Such technology would greatly assist in the early development stages of new simulation software.

There is a need for direct visual monitoring and interactive steering of ongoing simulations. There are two reasons for this need: (1) during the early development phase of simulation software, it is important to have crude visual means to see how certain values are being computed at specific places; (2) certain applications can be improved and accelerated by allowing a user to interact with parameters that control an ongoing simulation. Currently, projects that stress the integrated design and development of coupled simulation-visualization methods are not funded at sufficiently high levels. It is recommended that funding be made available to support the necessary research in this context.

Recommendation: ***Develop a research program in interactive visualization with running codes.*** This program should stress the integrated design and development of coupled simulation-visualization methods.

Finding: Traditionally, "field" visualization methods were developed solely for the analysis of scalar fields, vector fields, or tensor fields. Complex computational experiments are more and more computing multiple fields simultaneously, and visualization methods are needed that support the visual superposition of multiple fields, or even the combined visualization of the same field as computed by two or more different simulation codes. Furthermore, we now have the ability to produce massive amounts of data both via simulation and via experiment/observation (MEMS technology, ultra-high-resolution digital imaging, etc.). This fact requires us to develop approximation, meshing, and visualization technology supporting the visual integration of both simulated and empirical data and/or field continua reconstructed from discrete data.

Certain simulations and experiments produce particle datasets in addition to discrete field variables. Often the time-dependent characteristics of the particles are of interest, especially as particles undergo the specific influences of their surrounding fields. Visualization techniques are needed that support a combined visual representation of particle datasets and continuous field data. In some applications it is also of interest to characterize the interplay and geometrical configuration arising in the context of a physical process, e.g., the transport of water through porous media. A visual depiction of evolving molecular configurations may substantially aid our understanding of a specific phenomenon.

There is an increasing need to visualize data that is inherently multi-dimensional. In this context, it is important to consider that both the number of independent variables (like space and time) and the number of dependent variables (like temperature, salinity, and velocity vector) can be arbitrary. To date, no truly powerful visualization techniques exist for such multi-variate and multi-valued data.

Mesh generation is an integral part of all mesh-based simulation techniques. Visualization of meshes and mesh quality is essential to ensure that a simulation performs well. Dedicated visualization methods are needed for the display of meshes.

Fundamental research efforts must be funded in the areas of multi-field visualization and multi-dimensional data visualization. It is becoming increasingly important to have a means to visually overlay/superimpose multiple scalar, vector, and possibly even tensor fields — to compare experimental and simulated data, to compare data produced for the same field variable by different simulations, or to be able to study the interplay between a scalar and a vector variable. Furthermore, there is a need to visually explore high-dimensional data — datasets where a multitude of dependent variables depend on a multitude of independent variables. The visualization research community has not yet been able to develop truly intuitive and easy-to-use methods for such purposes. DOE should fund more basic research in this regard.

Recommendation: ***Establish a research program in the areas of multi-field visualization and multi-dimensional data visualization.***

Finding: Visualization techniques must take into account the ever-growing size of datasets. Terabyte datasets exist today for which adequate visualization technology does not yet exist, and petabyte-size datasets will be commonly produced in a few years. Visualization solutions must be developed for highly distributed and remote environments, where powerful servers, supercomputers, and/or clusters perform the bulk of the computations needed to enable interactive data exploration. Current networking infrastructure is not adequate to support demanding visualization applications over networks, and substantial upgrades are required in this regard.

Today physical phenomena are being modeled for which the spatial and temporal resolutions can vary substantially. Resolutions may vary over tens of orders of magnitude. Various meshing and simulation techniques exist that are used routinely today to deal with physical phenomena exhibiting major variations in resolution. In order to exploit the resulting datasets in an interactive visualization context, new approaches are needed to assist a scientist in identifying the regions in space or time where an interesting feature may exist. Since such features may very well be several orders of magnitude smaller than what we can resolve as a pixel, we need to integrate “intelligent agents” into the visualization process to steer the exploration process. More automatic data mining technology must be developed and be integrated effectively into the overall data exploration process.

Identifying features or patterns in the data prior to the execution of an interactive visualization process promises to significantly reduce the time a scientist needs to spend identifying regions of potential interest.

Today's simulations and imaging devices are producing datasets whose sizes and scale variations push current visualization hardware and our human visual perception ability to their limits. In the near future, it will no longer be feasible to expect a human user to be able to fully explore massive datasets. Thus, there will be a need to perform the needed fundamental research for more automatic data exploration, where intelligent agents can assist a user in the identification of those regions in a dataset that are worthy of interactive visual inspection. DOE must fund fundamental research in the area of coupled automatic-visual data exploration to ensure that next-generation petascale datasets can be exploited scientifically.

Recommendation: ***Establish a research program in the area of automated data exploration for next-generation petascale datasets.*** Datasets are being produced today (by simulations and imaging devices) whose sizes and scale variations push current visualization hardware and our human visual perception ability to their limits. Automated methods aid by finding interesting features in large datasets.

Finding: Life science applications are producing massive datasets via simulation (e.g., protein folding) and modern imaging technology (e.g., DNA-/gene-chip data and microscopy). These datasets can be geometrical in nature and highly abstract. Substantial advances are required concerning the visualization of data generated in contemporary computational biology applications to ensure that the data being produced can be analyzed to the greatest extent possible. Biology has traditionally not been a computational discipline, and the rapid advances in the computational aspects of biology make necessary a much more concerted effort in massive life science data visualization. Furthermore, biological systems are extremely complex. They vary across many levels of scale (from molecule to cell), and at the functional level they typically are characterized at a more abstract, not spatial/geometric, level. Thus, visualization methods to be developed must pay attention to multi-resolution needs, annotation requirements, and focus-and-context paradigms.

Life science applications pose particularly challenging problems, as they are beginning to integrate data resulting from physics, chemistry, and biology simulations, often spanning several orders of magnitude in physical scales but also various levels of abstraction. Data visualization technology for life science applications is in its infancy, and the gap is rapidly growing between our ability to generate more and more life science data and to visually explore them. Much more powerful coupled scientific database management-visualization systems will be needed in the very near future.

Recommendation: ***Significantly enhance life science data visualization efforts, with particular emphasis upon the relationship with scientific data management.***

Finding: Visualization methods can produce additional, derived data (e.g., isosurfaces extracted from a 3D scalar field) that one may want to store, compress, annotate, etc. Users require technology that considers the needs of database management technologies supporting the storage, annotation, and later efficient retrieval of massive datasets extracted from original scientific data.

File formats and data models are inconsistent, and severe data visualization problems arise due to this fact. There is a need to define and deploy a standard data model or a very limited number of standard models and storage formats that take into account the types of data most commonly being used by scientists. In addition, mechanisms should be established to convert data to such a standard data model and to develop visualization techniques for it.

Scientific data management — including data formats, data storage, data conversion, database-like query technology, etc. — is becoming increasingly important as formerly disparate subfields in one scientific domain start to integrate, and diverse datasets must be combined for visual exploration and annotation.

Recommendation: ***Develop new programs that link visualization with data management and provide support for multiresolution representations of large datasets, support for simultaneous display of data from disparate sources, support for the ability to generate and display derived values, and the ability to pose queries and display results.***

6 Workshop Contributors and Speakers

John Bell, Lawrence Berkeley National Laboratory (jbbell@lbl.gov).
Andrew Canning, Lawrence Berkeley National Laboratory (acanning@lbl.gov).
Jacqueline Chen, Sandia National Laboratory – California (jhchen@sandia.gov).
Bruce Cohen, Lawrence Livermore National Laboratory (bcohen@llnl.gov).
Inna Dubchak, Lawrence Berkeley National Laboratory, Genome Sciences Dept. (ildubchak@lbl.gov).
Kwok Ko, Stanford Linear Accelerator Center (kwok@slac.stanford.edu).
Mathew Maltrud, Los Alamos National Laboratory (maltrud@lanl.gov).
Anthony Mezzacappa, Oak Ridge National Laboratory (mezzacappaa@ornl.gov).
Bill Nevins, Lawrence Livermore National Laboratory (nevins@llnl.gov).
Daniel Rokhsar, Lawrence Berkeley National Laboratory/Joint Genome Institute (dsrokhsar@lbl.gov).
Rob Ryne, Lawrence Berkeley National Laboratory (rdryne@lbl.gov).
Ed Seidel, Max Planck Institute (eseidel@aei-potsdam.mpg.de).
John Shalf, Lawrence Berkeley National Laboratory (jshalf@lbl.gov).
Don Spong, Oak Ridge National Laboratory (spongda@ornl.gov).
Garrison Sposito, University of California at Berkeley and Lawrence Berkeley National Laboratory (gsposito@lbl.gov).
Doug Swesty, State University of New York, Stony Brook (Douglas.swesty@sunysb.edu).

Linda Sugiyama, Massachusetts Institute of Technology (linda@psfc.mit.edu).
Lin-Wang Wang, Lawrence Berkeley National Laboratory (lwwang@lbl.gov).

7 Workshop Participants

Joonhee An, Lawrence Berkeley National Laboratory (jman@lbl.gov).
Alan Aspuru-Guzik, University of California at Berkeley (aspuru@okra.cchem.berkeley.edu).
Wes Bethel, Lawrence Berkeley National Laboratory (ewbethel@lbl.gov).
Mark Brewer, University of California at Berkeley (mlbrewer@uclink.berkeley.edu).
Greg Butler, Lawrence Berkeley National Laboratory (gbutler@nersc.gov).
Christopher Cantalupo, Lawrence Berkeley National Laboratory (cmc@nersc.gov).
Jonathan Carter, Lawrence Berkeley National Laboratory (jtcarter@lbl.gov).
Christine Celata, Lawrence Berkeley National Laboratory (CMCelata@lbl.gov).
Michele Ceotto, University of California at Berkeley (ceotto@neon.cchem.berkeley.edu).
Tom DeBoni, Lawrence Berkeley National Laboratory (TMDeBoni@lbl.gov).
Ouafae El Akramine, University of California at Berkeley (ouafae@uclink4.berkeley.edu).
Stephane Ethier, Princeton Plasma Physics Laboratory (ethier@pppl.gov).
Randy Frank, Lawrence Livermore National Laboratory (rjfrank@llnl.gov).
Alex Friedman, Lawrence Berkeley National Laboratory (afriedman@lbl.gov).
Miguel Furman, Lawrence Berkeley National Laboratory (mafurman@lbl.gov).
David Grote, Lawrence Berkeley National Laboratory (DPGrote@lbl.gov).
Bernd Hamann, University of California at Davis (hamann@cs.ucdavis.edu).
Chuck Hansen, University of Utah (hansen@cs.utah.edu).
Brian Hingerty, Oak Ridge National Laboratory (beh@ornl.gov).
Donald Jones, Pacific Northwest National Laboratory (dr.jones@pnl.gov).
Ricky Kendall, Ames Laboratory (rickyk@ameslab.gov).
Scott Klasky, Princeton Plasma Physics Laboratory (sklasky@pppl.gov).
James Kohl, Oak Ridge National Laboratory (kohlja@ornl.gov).
Alexander Kollias, University of California at Berkeley (sasha@okra.cchem.berkeley.edu).
Terry Ligocki, Lawrence Berkeley National Laboratory (tjligocki@lbl.gov).
Nelson Max, Lawrence Livermore National Laboratory (max2@llnl.gov).
Juan Meza, Lawrence Berkeley National Laboratory (JCMeza@lbl.gov).
Esmond Ng, Lawrence Berkeley National Laboratory (EGNg@lbl.gov).
Doug Olson, Lawrence Berkeley National Laboratory (dlolson@lbl.gov).
Michael Papka, Argonne National Laboratory (papka@mcs.anl.gov).
Sung-Ho Park, Lawrence Berkeley National Laboratory (Sungho_Park@lbl.gov).
Ji Qiang, Lawrence Berkeley National Laboratory (jqiang@lbl.gov).
Romelia Salomon (romesalomon@hotmail.com).
Ken Schwartz, Lawrence Berkeley National Laboratory (KSSchwartz@lbl.gov).
Cristina Siegerist, Lawrence Berkeley National Laboratory (cesiegerist@lbl.gov).
Horst Simon, Lawrence Berkeley National Laboratory (hdsimon@lbl.gov).
Oliver Staadt, University of California at Davis (staadt@cs.ucdavis.edu).

John Staples, Lawrence Berkeley National Laboratory (staples@lbl.gov).
Theodore Sternberg, Lawrence Berkeley National Laboratory (tdsternberg@lbl.gov).
John Ashley Taylor, Argonne National Laboratory (jtaylor@mcs.anl.gov).
John Tran, Stanford Linear Accelerator Center (jtran@SLAC.Stanford.EDU).
Samuel Uselton, Lawrence Livermore National Laboratory (uselton1@llnl.gov).
Michel Van Hove, Lawrence Berkeley National Laboratory (vanhove@lbl.gov).
Francesca Verdier, Lawrence Berkeley National Laboratory (fverdier@lbl.gov).
Vince Wayland, University Corporation for Atmospheric Research (wayland@ucar.edu).
Michael Welcome, Lawrence Berkeley National Laboratory (mlwelcome@lbl.gov).
Dean Williams, Lawrence Livermore National Laboratory (williams13@llnl.gov).
Yushu Wu, Lawrence Berkeley National Laboratory (yswu@lbl.gov).
Keni Zhang, Lawrence Berkeley National Laboratory (kzhang@lbl.gov).

8 Acknowledgements

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Sciences Division, of the U.S. Department of Energy under Contract No: DE-AC03-76SF00098.