Title
Vector Field Visual Data Analysis Technologies for Petascale Computational Science

Author
Garth, Christoph

Publication Date
2009-12-14

Peer reviewed

Vector Field Visual Data Analysis Technologies for Petascale Computational Science

State-of-the-art computational science simulations generate large-scale vector field data sets. Visualization and analysis is a key aspect of obtaining insight into these data sets and represents an important challenge. This article discusses possibilities and challenges of modern vector field visualization and focuses on methods and techniques developed in the SciDAC Visualization and Analytics Center for Enabling Technologies (VACET) and deployed in the open-source VisIt tool.

Introduction

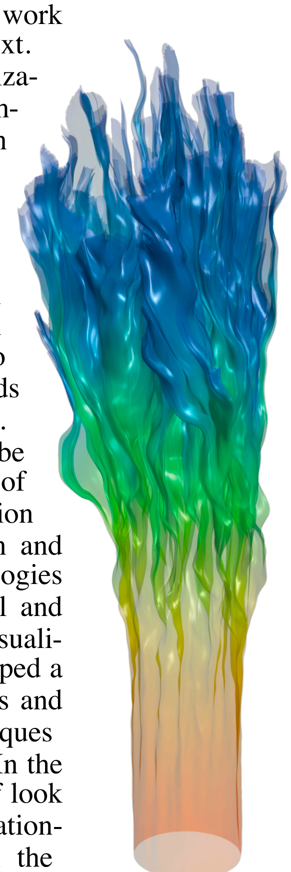
Modern scientific visualization plays a significant role in contemporary science. Visualization is the transformation of abstract data, whether it is observed, simulated, or both, into readily comprehensible images, and has proven to be an indispensable part of the scientific discovery process in many fields of science.

The visualization of vector fields is one of the more complex areas of scientific visualization. For example, the analysis of fluid flow that governs natural phenomena on all scales from the smallest (e.g. Rayleigh-Taylor mixing of fluids) to the largest (e.g. supernovae explosions) relies crucially on visualization to elucidate the patterns exhibited by flows and the dynamical aspects driving them. Processes typically described by vector fields – such as transport, circulation and mixing – are prevalently non-local in nature. Due to this specific property, methods and techniques developed and proven successful for scalar data visualization are not readily generalized to the study of vector fields. Hence, while it is technically feasible to apply such methods to directly derived scalar quantities such as vector magnitude, the resulting visualizations often fall short of providing an explanation of the mechanisms underlying the scientific problem.

Broadly, vector field visualization techniques fall into three categories. Glyph-based methods depict the behavior of a field by mapping vectors to a geometric representation (such as e.g arrows) and hence convey only a very local perspective. Feature-based techniques typically aim at identifying regions of interest by comparing a vector field locally to an empiric definition of a pattern of interest, for example vortices and shocks. Both classes are limited in their use as general visualization tools since they fail to capture essential aspects of vector fields or work in a very narrow, problem-specific context.

The third class of vector field visualization techniques, integration-based methods ,is focused on deriving visualization from the study of integral curves of a vector field. Intuitively interpreted as the trajectories of massless particles transported by a vector field, such curves capture the evolution of material quantities over variable time-scales and are thus a broadly applicable tool to study, visualize and illustrate vector fields and their non-local and dynamic aspects.

The focus of this article is to describe recent research efforts in the field of integration-based vector field visualization within the DOE SciDAC Visualization and Analytics Center for Enabling Technologies (VACET). Working on both theoretical and practical aspects of integration-based visualization, VACET researchers have developed a number of new visualization techniques and developed tools to make these techniques accessible to the scientific community. In the following article, after providing a brief look at some of the fundamentals of integration-based visualization, we describe both the basic research centered on novel visualization methods as well as the current state of and future plans for integration into VisIt, the open-source visualization tool VACET uses to deploy its research results.



A stream surface illustrates turbulent flow into a tank.

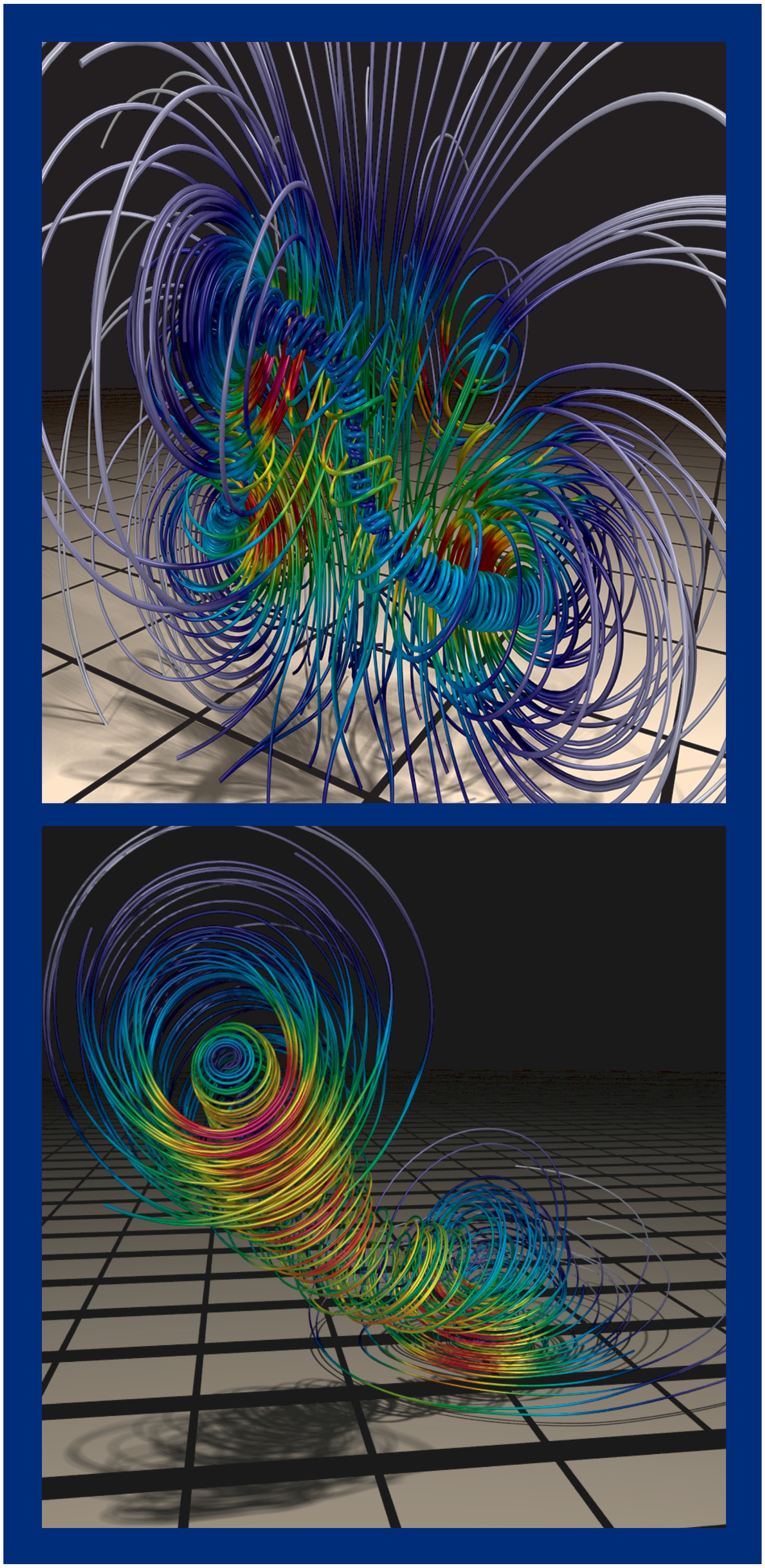


Figure 1. *This image shows streamlines in different time steps of a vortex merger data set. Data courtesy Dan Martin, LBNL Applied Numerical Algorithms Group / SciDAC Applied Partial Differential Equations Center for Enabling Technologies (APDEC).*

Integral Curves and Visualization

Integral curves are one of the oldest and most prevalent vector field visualization techniques, and were originally developed to reproduce physical flow visualization experiments based on small, neutrally buoyant particles.

Technically, an integral curve is a curve whose tangent is parallel to the vector field at every curve point. Among all such admissible curves, individual curves are determined by selecting an initial condition, i.e. a seed point that the curve originates from or passes through. The mathematical interpretation of integral curves as the solutions of ordinary differential equations coincides with that of massless point particles that are advected along the vector field. This intuitive interpretation shows that integral curves are ideal tools to study non-local phenomena as transport ("where do particles go?"), mixing ("how strongly do particles mingle?") and circulation ("are trajectories closed?"). Moreover, this type of analysis can be performed both qualitatively, by directly visualizing integral curves as particle trajectories, or quantitatively, by computing many trajectories and performing e.g. statistical analysis.

Integral curves are applicable in many different settings, since the only theoretical requirement on existence and uniqueness is the continuity and boundedness of the vector field they are defined over. This condition can be fulfilled for virtually all kinds of discrete vector fields from simulation or measurement using interpolation. Hence, integral curves are broadly useful as a vector field visualization primitive and apply to both stationary and time-varying vector fields.

The ability to study transport through integral curves also offers an interesting change of perspective. Instead of considering the so-called *Eulerian* perspective, i.e. the value and evolution of vector field-transported scalar quantities at fixed locations in space, the *Lagrangian* view examines the evolution from the point-of-view of an observer attached to a particle moving with the vector field. While both approaches are technically on equal footing, the Lagrangian perspective allows a more natural and intuitive description and analysis of the overall behavior. Consider for example the case of combustion: fuel burns while it is advected by surrounding flow; hence the burning process and its governing equations are primarily Lagrangian in nature.

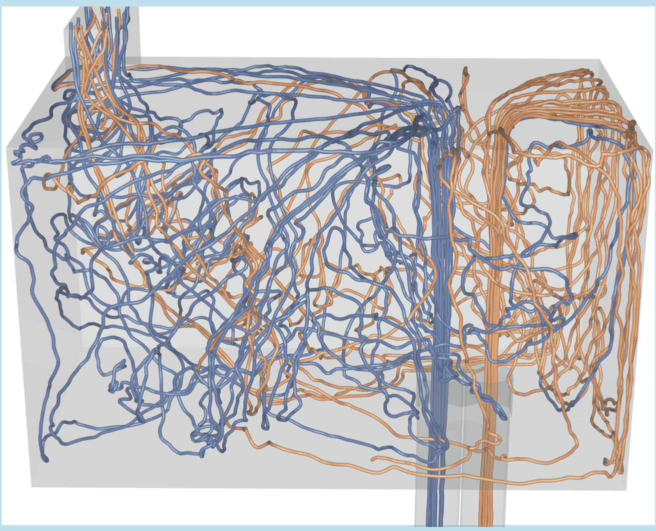
Thermal Convection Analysis

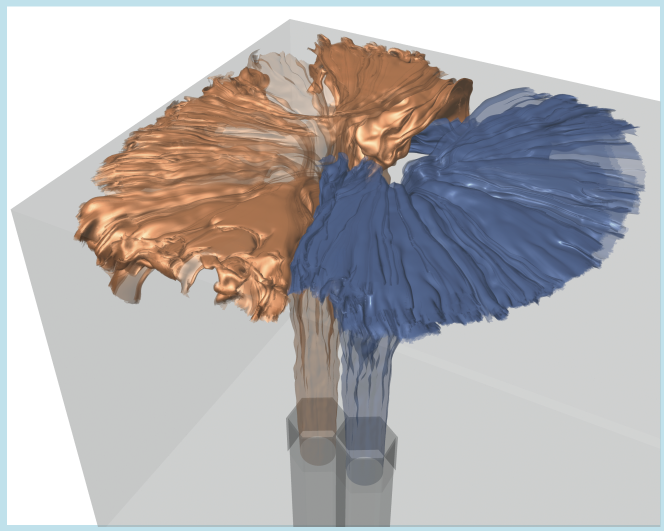
Streamlines and stream surfaces thermal convection data set illustrate the mixing of air in a box. Warm (orange) and cold (blue) air are entering the box through two inlets. Although the warm air dominates one side of the box, both temperatures mix before ultimately exiting through the outlet in the upper right.

From the below streamline visualization, it becomes apparent that while hot and cold air do mix, the mixing is not entirely optimal since no hot air penetrates into the lower left corner of the box.

A stream surface visualization (bottom right) shows that the streams stay separated for some time after entering the box.

The time-varying simulation was conducted by Paul Fischer and Alex Obabko (ANL) as part of an INCITE run using the Nek5000 spectral-element code. It consists of 23 million unstructured elements over 3600 time steps (only one timestep was used for the images below).





A secondary but nevertheless important benefit of the Lagrangian perspective is an increased independence from a specific frame of reference.

Computationally, integral curves are approximated using numerical integration schemes. These schemes construct a curve in successive pieces: starting from the seed point, the vector field is queried in the vicinity of the current point, and an additional curve piece is determined and appended to the existing curve. This is repeated until the curve has reached its desired length or leaves the vector field domain. While it is trivial to parallelize the calculation of multiple integral curves on a small data set (by replicating the data and parallelizing over the integral curves), it is very difficult to parallelize the calculation of one or more integral curves on a large data set. The non-local, non-linear nature of particle advection implies that particles can traverse almost arbitrary parts of a data set which must be brought into memory. VACET researchers have evaluated existing and developed new parallelization strategies that allow for scalability of integral curve computation, allowing the application even on very large data sets.

Concerning visualization, a number of visualization algorithms have been developed over the past decades that are built on integral curves. The simplest method, the direct depiction of trajectories as *streamlines* (in the stationary case) and *pathlines* (in the time-varying case) is often able to quickly yield a visual impression of the basic vector field structure. In a dynamically changing vector field, pathlines reflect the time-dependent nature of the field better than streamlines. However, the increased amount of data processing required to compute them can be a reason to limit the visualization to streamlines. While these direct depictions are often sufficient for a first inspection, there are some inherent limitations. If too many integral curves are depicted simultaneously,

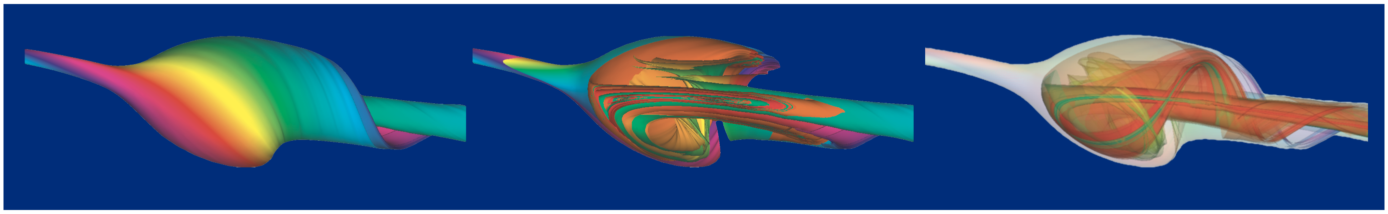
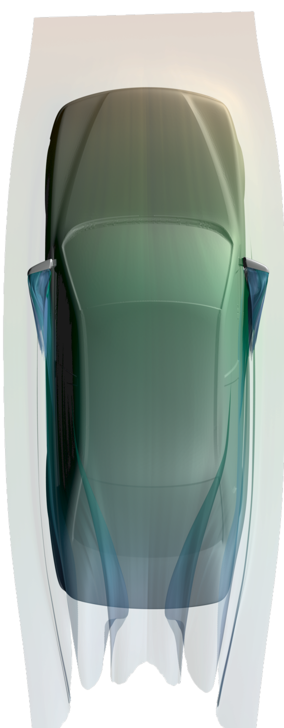


Figure 2. *This image shows a stream surface flowing through a so-called vortex breakdown bubble. After entering the bubble, the flow recirculates, stretches and folds many times before leaving the bubble at the rear end. While the opaque surface (left) only conveys the basic shape, apply a clipping plane (middle) or transparency (right) reveals the complex inner structure of the bubble. Sample data courtesy M. Rütten (DLR Germany).*

the resulting image is visually ambiguous since individual lines are not well separated. VACET researchers have addressed these visual problems by developing methods to compute *integral surfaces*, as described below. For some problems, integral curves take on the role of an elementary building block to deliver more abstract visualizations, such as for example topological methods (see Sidebar "Visualizing Magnetically Confined Fusion").

Overall, integral curves are a fundamental building block of vector field visualization. As described below, the VisIt visualization system offers a robust implementation of integral curves over a variety of mesh types and data formats and offers a ready-to-use, scalable vector field visualization capabilities in this context.



An integral surface illustrates the flow around a car.

Integral Surfaces

Integral surfaces represent a natural generalization of integral curves. Instead of depicting an integral curve emanating from a single seed point, integral surfaces capture the simultaneous evolution of a continuum of particles that emanate from one- or two-dimensional seed sets such as lines or surfaces. Depending on the type of seed set, and the chosen visualization, these surface have different names.

Path surfaces represent a set of integral curves which are seeded on a one-dimensional curve at a single point in time. Intuitively, this corresponds to the simultaneous release and evolution of infinitely many particles forming the seed curve. The surface traversed by these particles then forms the path surface. Hence, path surfaces represent the advection of the seed curve through the field.

A slightly different approach underlies *streak surfaces*. Particles are seeded from a curve, but in contrast to path surfaces, they are seeded continually over a specified time interval. The intuitive interpretation is that of dye released into a flow at the locations of the seed curve. The dye forms a surface sheet – a streak surface. For such surfaces, it is not feasible to depict the entire volume traversed by the particles in a single frame. However, streak surfaces lend themselves well to animated visualization that illustrate the general time-dependent behavior of a flow.

Integral surfaces are preferable over integral curves in a many situations. For the case of presentation visualization, i.e. visualization with the intent of conveying specific aspects of a data set, they provide an ideal tool since they have strong illustrative character. Furthermore, the visual quality of images generated from such surfaces is often vastly superior to those obtained using a large number of integral curves. If used in large numbers, the latter tend to result in cluttered visualization that do not convey the three-dimensional nature of vector field structures. Typical phenomena such as folding, curling and divergence of integral curves aggravate these problems. Integral surfaces, on the other hand, can provide spatial cues by means of physically correct shading. The natural visual coherence of the seed set points is inherited by the surfaces. Within VACET, researchers have developed and evaluated a variety of approaches to further augment the visualization quality of integral surfaces through the application high-quality transparent rendering and surface textures that encode such information as seed point location, and seeding time into the surface color. Through this, expressive visualizations of even very complex flows are easily obtained.

While powerful as visualization primitives, integral surfaces are hard to approximate correctly. This is a consequence of the often strongly non-linear nature of advection as it induces divergence, shearing, twisting and folding in the surfaces. Typically, an integral surface is discretized by a finite set of integral curves that form a "skeleton" of the surface; failing to adapt this skeleton results in a surface that contains extreme rendering artifacts. VACET researchers have built on existing approaches to enable the computation of integral surfaces over very large vector fields. This is achieved by propagating the integral curve skeleton in such fashion to keep vector field evaluations closely localized in space and time. This enables a distributed treatment of large vector fields. The corresponding algorithms are planned for inclusion into the VisIt visualization tool in the near future.

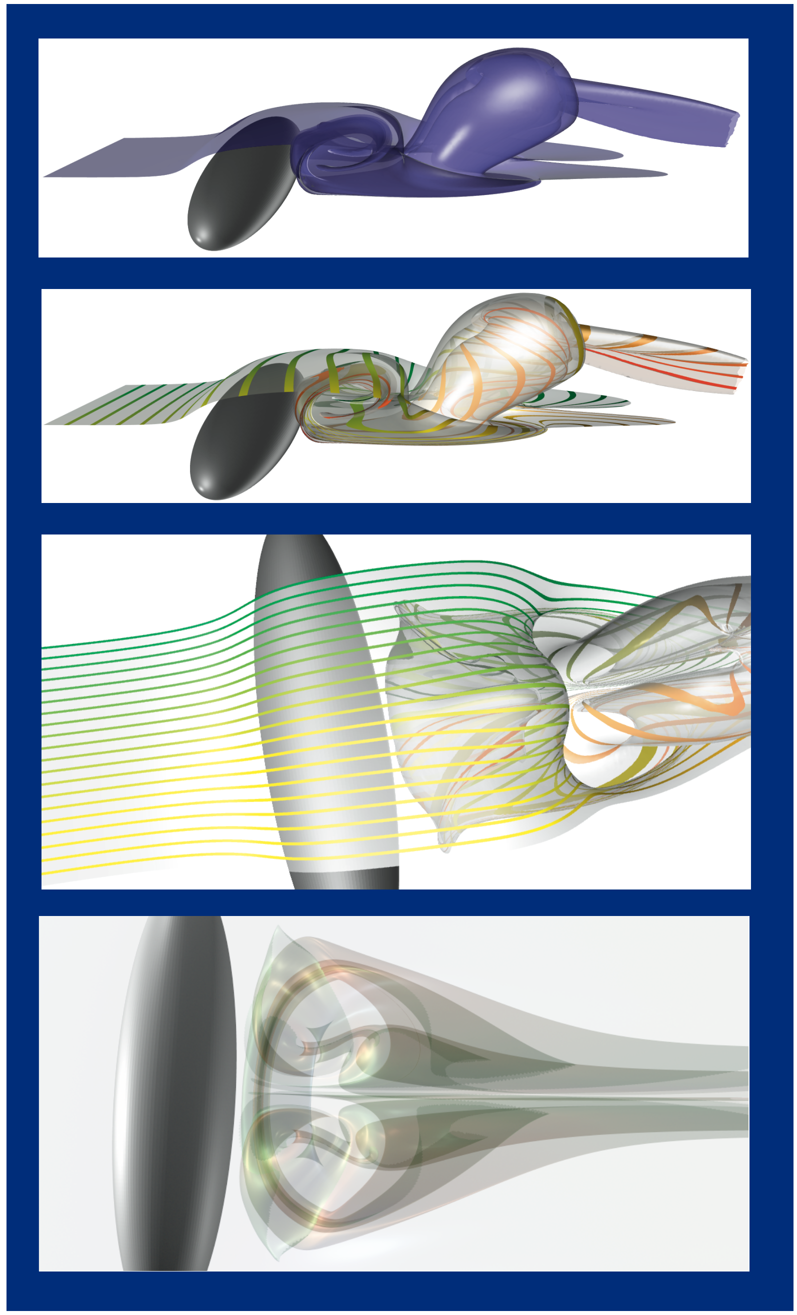


Figure 3. *This image shows various integral surfaces depicting the formation of a vortex system behind an ellipsoid that is surrounded by the flow of water. Textures (middle images) highlight different dynamic aspects of the flow by highlighting particles time lines and streak lines. Sample data courtesy M. Rütten (DLR Germany).*

Lagrangian Visualization & Analysis

The visualization techniques discussed above are centered on directly representing the evolution of a set of particles as they are advected throughout a vector field. A different class of visualization techniques, also built on integral curves as the elementary primitive, focuses on Lagrangian analysis and visualization by deriving information from the movement of *all* particles under advection in a vector field. In this context, the recently introduced notion of so-called *Finite-Time Lyapunov Exponents* (abbreviated FTLE) has demonstrated great potential towards insightful visualization of the dynamic structure of vector fields. The basic idea consists of transporting particles over a short time interval, and measuring an average exponential separation rate between closely neighboring particles over a finite time.

For particles transported forward in time, a locally large exponent at a given point indicates strong divergence or separation of the particles contained in the neighborhood of this point. Conversely, transporting particles backward in time (which amounts to identifying where the particles in the point neighborhood originated from in the past) yields a large exponent in those locations that exhibit strong particle convergence. The two scalar fields of exponents for forward- and backward-time advection can be directly visualized using traditional scalar visualization techniques, such as volume rendering.

The key property that provides meaning to FTLE fields beyond a simple visual encoding of particle divergence and convergence is the fact that these fields indicate so-called *Lagrangian Coherent Structures* (LCS) in the form of ridges, i.e. lines and surfaces along which the exponents take on locally maximal values. It can be shown that LCS are strongly connected to the mathematical interpretation of vector fields as dynamical systems by taking on the role of separatrices of the vector field. The identification and visualization of such separatrices admits direct insight into

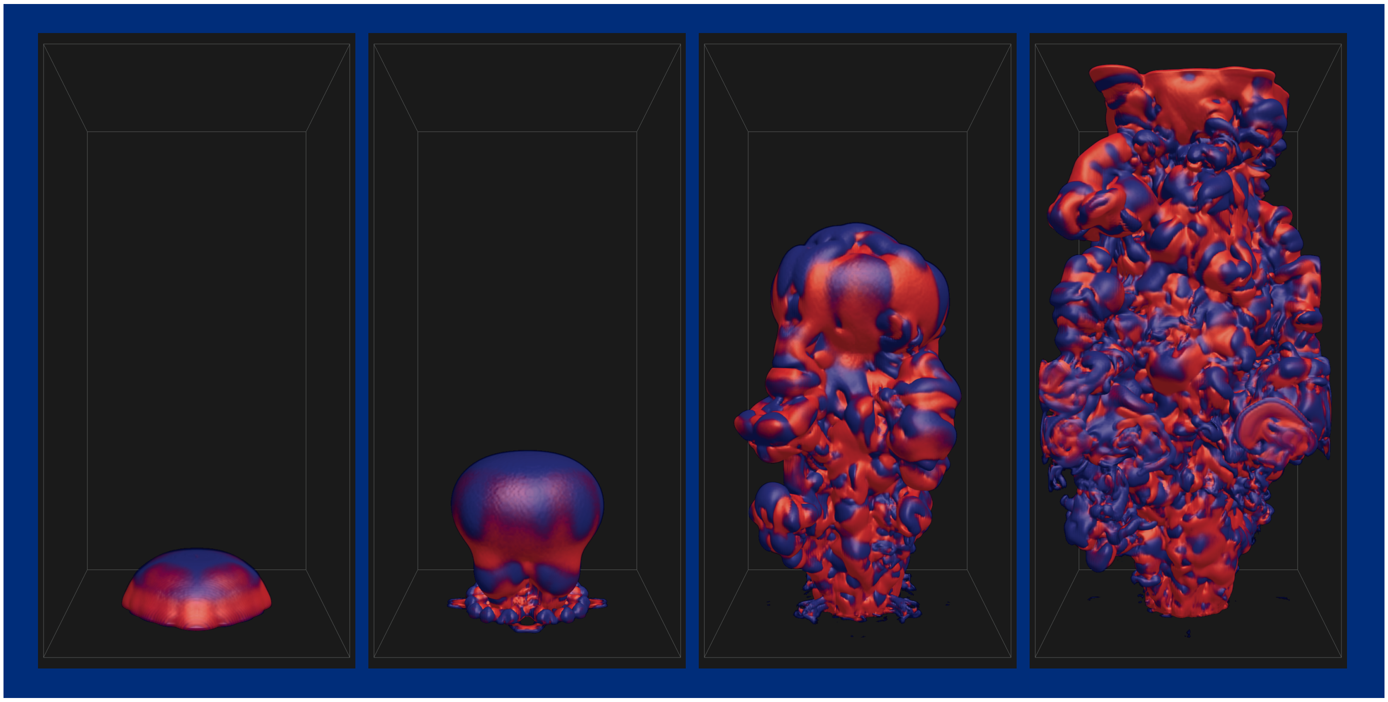


Figure 4. *This image shows direct volume rendering of the time-varying Finite-Time Lyapunov Exponent fields (red indicates the forward-time exponent, blue shows the backward-time exponent) for four time steps, illustrating the formation of turbulence in a high-speed jet of entering a domain of stationary fluid. Individual turbulent structures and structure size and distribution can be observed directly from the volume rendering. Data set: C. Garth (UC Davis).*

the global, time-varying dynamics of the flow by visually separating regions of differing particle behavior.

For example, particles that are circulating in a vortex close to its boundary take a different path than those particles that are flowing past the vortex. Consequently, the exhibit strong divergence, reflected in a large finite-time exponent. Thus, the FTLE field encodes the vortex boundary – where particles separate – directly. Thus, direct depiction of these fields often yields an excellent overview of the vector field behavior. Beyond straightforward visualization through direct volume rendering, FTLE fields can also serve to study a vector field indirectly through the application of readily available visualization methods developed for scalar fields.

The definition and computation of FTLE fields are deceptively simple. Overall, all that is required is the advection of a sufficiently dense set of particles that span the domain of interest and the successive measurement of the separation exponent. The requirement of sufficient density, however, often requires that millions or even billions of particles are advected through the vector field, and hence a corresponding number of integral curves must be computed. Thus, Lagrangian visualization borders on the infeasible even for vector fields of medium size and complexity. Yet, the potential for insight from this type of visualization is a crucial aspect of vector field visualization.

To remediate this situation, VACET researchers have developed novel algorithms that allow the incremental adaptive approximation of FTLE fields over a domain of interest. By adapting the resolution of the computation to the complexity of the FTLE fields themselves, the number of required integral curves is significantly reduced. Additionally taking advantage of distributed computation brings the computational effort of FTLE visualization to a feasible level and allows scaling to very large time-dependent data sets.

AMR Vector Fields

Adaptive Mesh Refinement (AMR) is a highly effective discretization method for a variety of physical simulation problems and has recently been applied to the study of vector fields in flow and magnetohydrodynamic applications. AMR represents the domain as a set of nested rectilinear grids at increasing resolutions. Grids are ordered in levels of resolution, and finer levels replace informa-

Visualizing Magnetically Confined Fusion

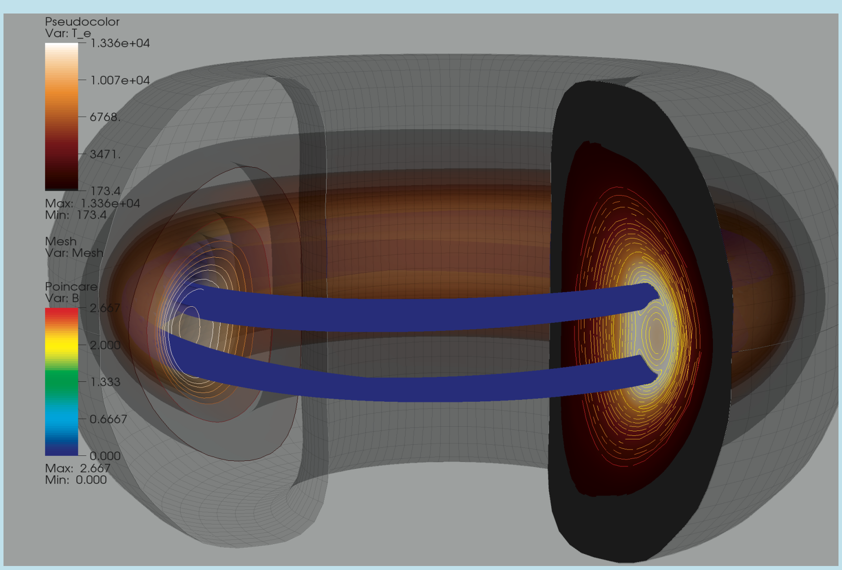
Fusion energy has the potential to provide an environmentally-acceptable, virtually inexhaustible source of energy for the future. Fusion energy plants would harnesses the power that fuels the stars by fusing light elements within a hot plasma. Creating an environment amenable to fusion, where temperatures of 100 million degrees are required, is an enormous challenge. Tokamaks such as the one being built by ITER is a type of fusion reactor that use toroidal magnetic fields to confine the burning plasma within the reactor core.

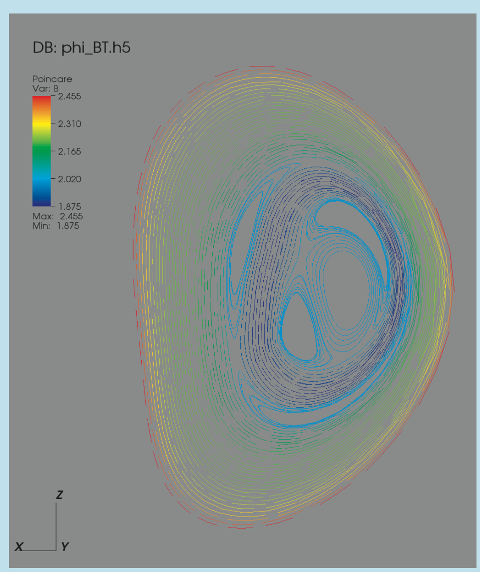
Simulations of magnetically confined fusion systems must model a variety of different plasma regions, including the core plasma, edge plasma and the edge-wall interactions. Each region of the plasma is subject to anomalous transport driven by turbulence, which can lead to large-scale disruptions and instabilities. Many of these processes must be computed on short time and space scales, while the results of integrated modeling are needed for the whole device on long time scales. The mix of complexity and widely differing scales in integrated modeling results in a unique computational challenge.

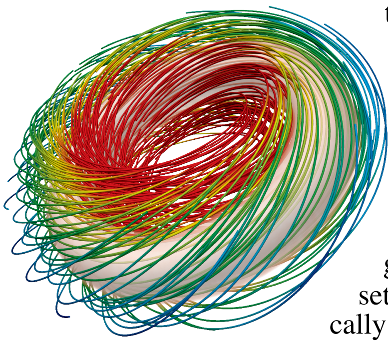
One key area to understanding the behavior of the plasma is the topology of the magnetic field. One effective method for visualizing and understanding the magnetic field's topology is via a Poincare map. This map is the intersection of a plane with the magnetic field and gives a cross-sectional view of the magnetic surfaces within the field. A Poincare map is formed by intersecting multiple fieldlines, each of which lies on a magnetic surface, with a plane that is transverse to the magnetic field. As the fieldline traverses the field a series of “puncture” points will be accumulated. These points form the outline of the magnetic surfaces with the field.

Traditionally, forming a Poincare Map is computationally very expensive because in order to analyze the entire magnetic field, a large number of fieldlines must be considered. Further, each fieldline can require a large number (thousands) of “puncture” points to accurately outline each magnetic surface.

To reduce this computational expense a new Poincare analysis tool has recently been added to VisIt. This new tool, developed by VACET researchers, uses the parallel streamline functionality to compute the magnetic field lines and then creates the Poincare map using only a few hundred puncture points. The Poincare tool is able to leverage the parallel streamlines in two ways. First, it enables analysis of simulations too large to fit into memory on a single compute node. Second, it allows the large number of field lines to be computed in parallel, irrespective of the size of the simulation. The ability to perform increasingly detailed analysis on a larger number of field lines on larger computational simulations is a powerful enabling tool for plasma fusion researchers. Simulation performed by Scott Kruger using the NIMROD code on the IBM SP RS/6000 computer at the National Energy Research Center. Visualization and Analysis performed using the VisIt.



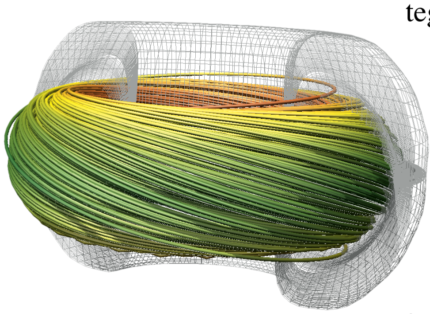




tion at locations of the domain they overlap. Thus, AMR combines the adaptivity of unstructured meshes with the implicit connectivity of regular grids, adding only little overhead in form of a box layout description.

Vector field visualization using integral curves faces two challenges in this setting. First, AMR vector fields are typically represented in cell-centered fashion with one vector per cell; thus, an interpolation scheme must be chosen to make the vector field continuous. The commonly used approach of averaging cell information onto the mesh nodes and then interpolating trilinearly, however, effects a strong smoothing of the data. Second, resolution level boundaries, if treated naively and combined with cell-averaged interpolation, induces discontinuities of interpolation that can lead to significant errors during integral curve propagation and reduces the overall correctness and accuracy of the visualization.

VACET researchers have examined these problems and performed a comparative analysis of a several interpolation schemes on AMR meshes. Here, it was determined that the so-called *dual-mesh interpolant* that interpolates the cell-centered values directly without averaging yields results of improved accuracy. While conceptually simple, this interpolant however requires additional information on the boundaries of resolution blocks (i.e. *ghost cells*) that, if not present in existing data, must be derived from coarser levels. Furthermore, the explicit treatment of resolution level boundaries during the integration process can further improve the accuracy and correctness of the generated integral curves, resulting in a robust and accurate integral curve algorithm over AMR vector fields. These techniques have been implemented to augment the existing VisIt integral curve framework and will be deployed in a future release, allowing to apply generic integral curve visualization without special considerations on AMR data.



Streamlines in a Tokamak simulation

Scalable Integral Curves

The visualization techniques discussed above represent a significant step toward elucidating vector field visualization for many scientific problems. However, the insight provided by these methods comes at a high computational price. While geometric techniques such as integral surfaces require the computation of hundreds to thousands of integral curves, for Lagrangian visualization, these number in the millions to billions. Each integral curve propagation requires many vector field evaluations, which are typically expensive. Hence, the visualization of large data – or even of medium sized data with many integral curves – with such methods mandates the use of parallel computation for acceptable performance and user experience.

The propagation of individual integral curves is mutually independent; thus integrating many curves lends itself to parallelization. Based on this insight, a simple parallelization strategy statically distributes the set of seed points (each spawning one integral curve) over the available processors, with vector field data loaded on demand on each processor independently. This approach works well for small vector fields, since all processors have approximately the same workload, and hence parallel efficiency is good. In the case of large data, however, the performance of this approach strongly depends on the size and distribution of the seed point set. Some integral curves may traverse more parts of the data than others, requiring more time-consuming loading of data, while others finish propagating quickly, leaving their processors idle.

A second approach distributes the entire vector field data statically among processors. Integral curve computation is then started on the processor that contains the seed point in its resident subset of the vector field and is passed to other processors depending on the course of the trajectory, i.e. the computation follows the data. This procedure has the advantage of consolidating data loading at the beginning of the run, removing a typical bottleneck from the computation. However, if the seed points are chosen such that only a small fraction of the domain is traversed, only a small subset of processors participates in the computation, resulting in sub-optimal efficiency.

Overall, it can be observed that static decomposition of seed set, data, or both does not perform efficiently over the typical range of use cases presented by integration-based visualization. VACET researchers have investigated the benefits and problems inherent in static decomposition strategies and developed a novel parallelization algorithm that aims at providing better efficiency of parallel resources. Rather than decompose statically, the algorithm performs dynamic distribution of both seed points and data parts to individual processors. To achieve this, one processor acts as a *master* that assigns other processors (*slaves*) data parts to load and integral curves to propagate. The master maintains an overview of which data parts are resident on the individual processors and how much work in the form of integral curves is assigned to processors. It can thus quickly identify if processors are idle, and in that case redistribute the workload more evenly by telling other processors to load corresponding data and take over some part of the overall workload. The master derives these assignment from a set of heuristic rules that are designed to identify and avoid overloaded or idle processors and I/O bottlenecks. This *hybrid master/slave* algorithm avoids the worst cases of static parallelization strategies while retaining best-case comparable performance.

As a major advantage, this hybrid strategy alleviates the requirement on a visualization user to choose a specific parallelization approach in response to specific data or visualization method characteristics. Rather, it delivers good performance for a wide range of cases, and thereby significantly reduces the barrier of entry to parallel vector field visualization; this makes it an ideal candidate for a general-purpose, robust end-user tool. Along with the discussed static distribution strategies, the hybrid approach has been implemented and deployed in the VisIt tool.

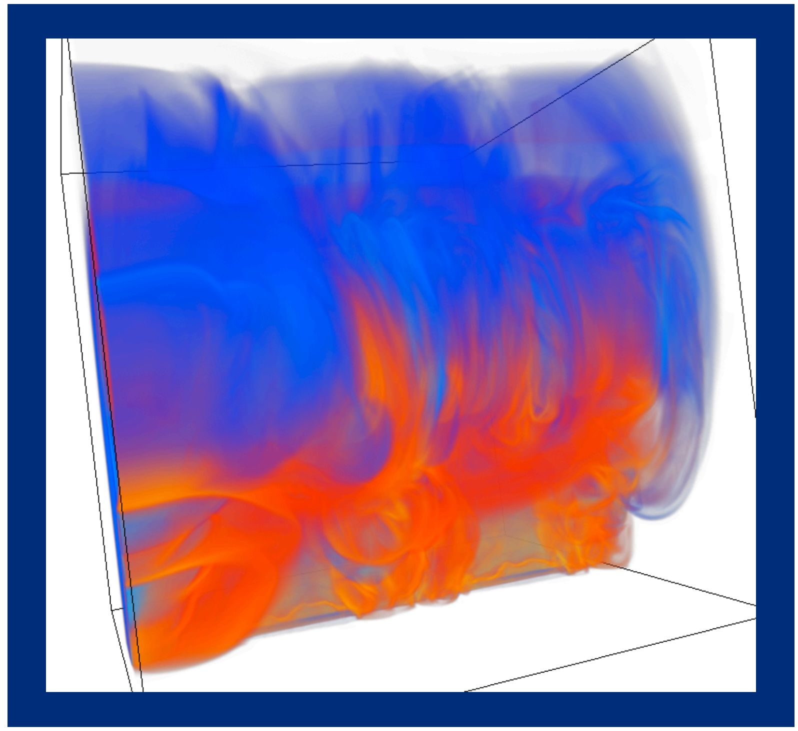


Figure 5. *This image shows translucent volume rendering of the time-varying Finite-Time Lyapunov Exponent fields (red: forward-time, blue: backward time). Data set: C. Garth (UC Davis).*

Vector Field Visualization using VisIt

VisIt is an open source, turnkey application for large-scale simulated and experimental data sets. Its charter goes beyond pretty pictures; the application is an infrastructure for parallelized, general post-processing of extremely massive data sets. Target use cases include data exploration, comparative analysis, visual debugging, quantitative analysis, and presentation graphics.

The VisIt product delivers the efforts of many software developers in a single package, leveraging several third party libraries: the Qt widget library for its user interface, the Python programming language for a command line interpreter, and the Visualization ToolKit (VTK) library for its data model and many of its visualization algorithms. In addition, an estimated fifty man-years worth of effort have been devoted to the development of VisIt itself. This effort has largely been focused on parallelization for large data sets, user interface, implementing custom data analysis routines, addressing non-standard data models (such as AMR meshes and mixed materials zones), and creating a robust overall product. VisIt consists over one and a half million lines of code, and its third party libraries have an additional million lines of code. It has been ported to Windows, OS X, and many UNIX variants, including AIX, IRIX, Solaris, Tru64, and, of course, Linux, including ports for SGI's Altix, Cray's XT4 and XT5, and many commodity clusters.

The basic design is a client-server model, where the server is parallelized. The client-server aspect allows for effective visualization in a remote setting, while the parallelization of the server allows for the largest data sets to be processed reasonably interactively. The tool has been used to visualize many large data sets, including a 216 billion point structured grid, a one billion point particle simulation, and curvilinear, unstructured, and AMR meshes with hundreds of millions to billions of elements. Additionally, VisIt re-

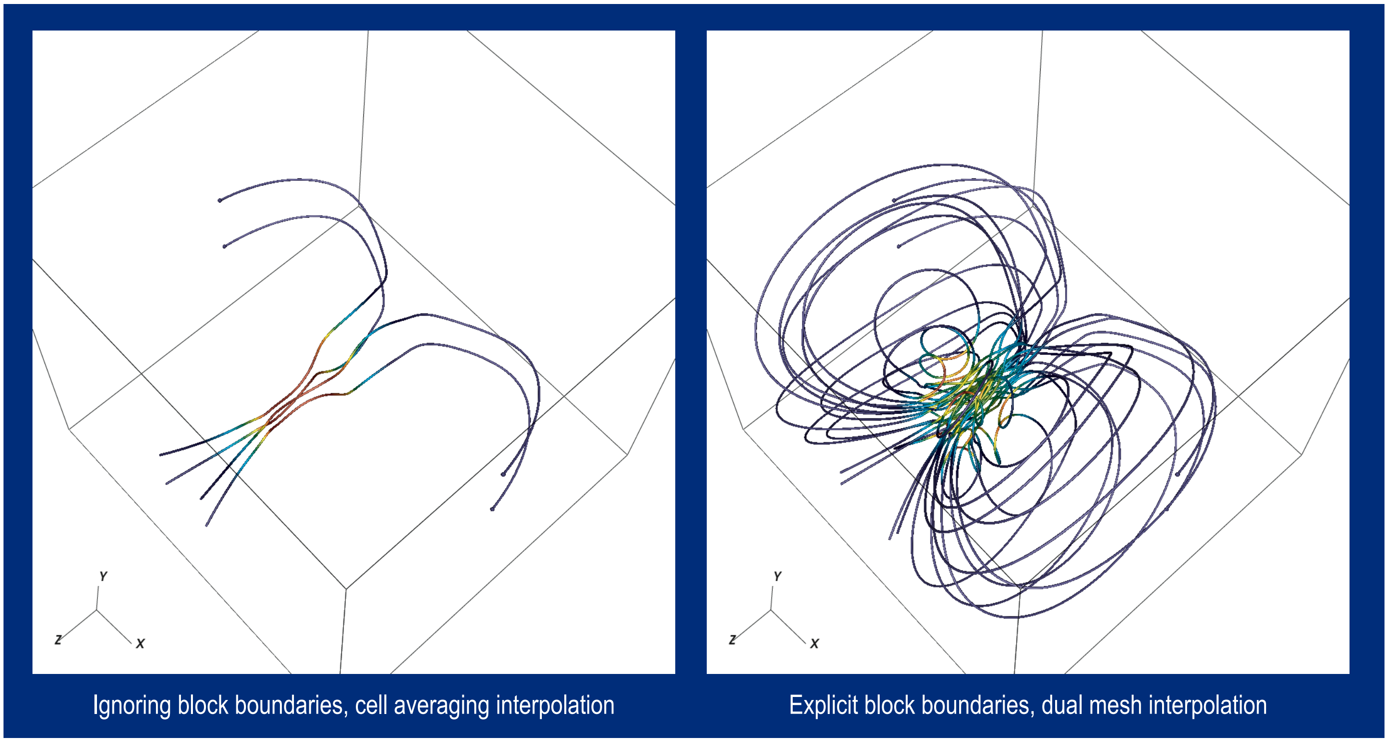


Figure 6. *This image shows an illustration of the qualitative difference between integral curve results using a naive integration approach (left) and dual-mesh interpolation with explicit block boundary treatment (right). The streamlines in the left image accumulate enough error that they wrongly exit the domain early. The right image shows the more accurate reproduction of the vortex structure present in the data set. Data courtesy Dan Martin, LBNL Applied Numerical Algorithms Group / SciDAC Applied Partial Differential Equations Center for Enabling Technologies (APDEC).*

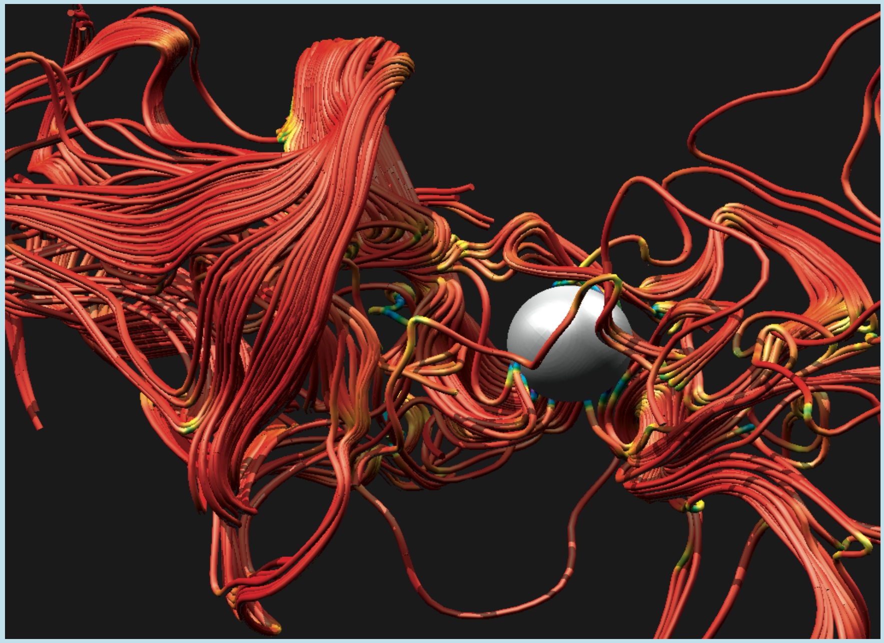
cently processed hero-sized synthetic data sets, ranging from one trillion to four trillion cells. Capabilities exist to link simulation code with Visit to allow *in situ* visualization and analysis during a simulation run.

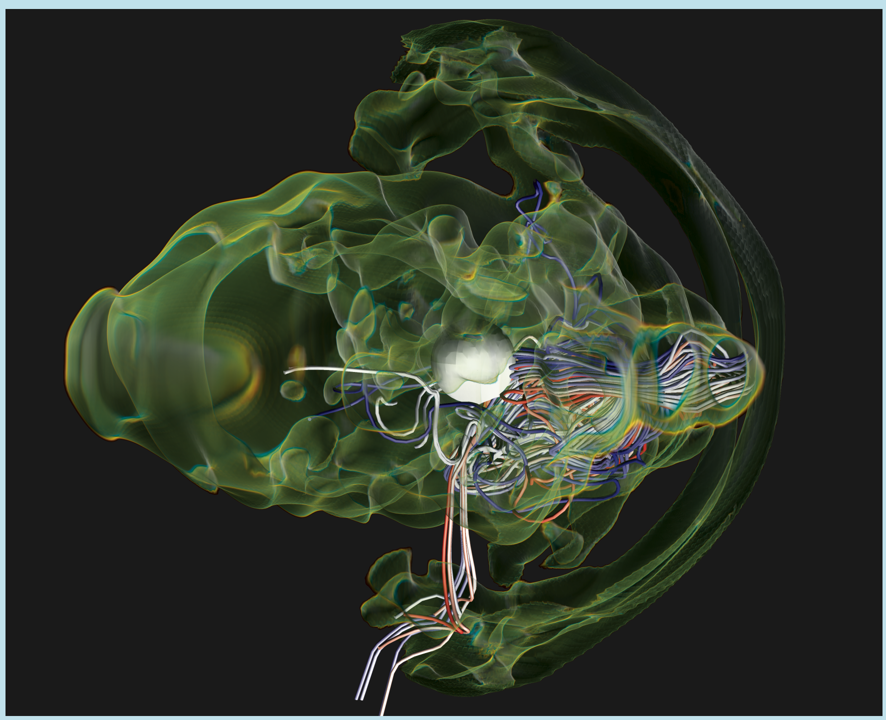
The VisIt project originated at Lawrence Livermore National Laboratory and currently has twenty-five developers from many organizations and universities, including five DOE Laboratories. VisIt received an R&D 100 award in 2005 and is downloaded approximately fifty thousand times per year. It is currently being used to visualize and analyze the hero runs on eight of the world's twelve fastest machines.

Recently, VACET R&D activities have enhanced VisIt's capabilities towards vector field visualization, resulting in a modern framework for integral curve computation that enables out-of-the-box application of corresponding techniques on all supported data formats and mesh types. Using state-of-the-art numerical schemes, VisIt integral curve methods are applicable to the highly complex vector fields resulting from current- and next-generation simulations. In combination with VisIt's interaction and scripting capabilities that allow the interactive determination of seed sets and integration parameters, integration-based visualization is applicable over a wide range of use cases, ranging from rapid interactive exploratory visualization to high-fidelity presentation visualization.

As most visualization algorithms of VisIt, the updated integration framework supports distributed computation to enable vector field visualization on large-scale data. It supports existing distributed data models and formats as typically generated from corresponding simulations; hence, no data conversion is necessary prior to visualization. Furthermore, scalability is achieved through a choice of distributed computation paradigms that allows visualization users to maximize parallel efficiency for specific problem types. A newly developed, hybrid approach, however, automatically adapts to the problem characteristics in terms of data size and seed set distribution to achieve near-optimal parallel performance for a wide range of visualization problems, thus freeing users of having to choose a specific parallelization strategy.

Supernova Magnetic Field Visualization





The search for the explosion mechanism of core-collapse supernovae and the computation of the nucleosynthesis in these spectacular stellar explosions is one of the most important and most challenging problems in computational nuclear astrophysics. Core collapse supernovae are the most energetic explosions in the Universe, releasing 10^{53} erg of energy in the form of neutrinos of all flavors at a staggering rate of 10^{57} neutrinos per second and 10^{45} Watts, disrupting, almost entirely, stars more massive than ten Solar masses and producing and disseminating many of the elements in the Periodic Table, without which life would not exist. They are a nexus for nuclear physics, particle physics, fluid dynamics, radiation transport, and general relativity, and serve as cosmic laboratories for matter at extremes of density, temperature, and neutronization that cannot be produced in terrestrial laboratories and physics beyond the Standard Model.

The collapse of the massive star's core results in the formation of an outgoing shock wave that eventually disrupts the entire star, resulting in a supernova. Along the way, the shock temporarily stalls and experiences the "stationary accretion shock instability" (SASI), which causes large deviations from spherical symmetry. This appears to be important to the supernova explosion mechanism, and may be responsible for spinning up the collapsed core, a nascent neutron star, into a pulsar. One open research question is the extent to which the SASI may generate magnetic fields, and the role of magnetic fields in the creation of supernovae.

The images above illustrate the nature of the magnetic field around the core using streamlines. The simulation was performed using GenASIS, a multi-physics code used for the simulation of astrophysical systems involving dense matter and run on Jaguar at NCCS by Eirik Endeve, Christian Cardall, and Reuben Budiardja (ORNL and University of Tennessee, Knoxville). The visualization was generated using the parallel streamline functionality in VisIt on 512 processors.

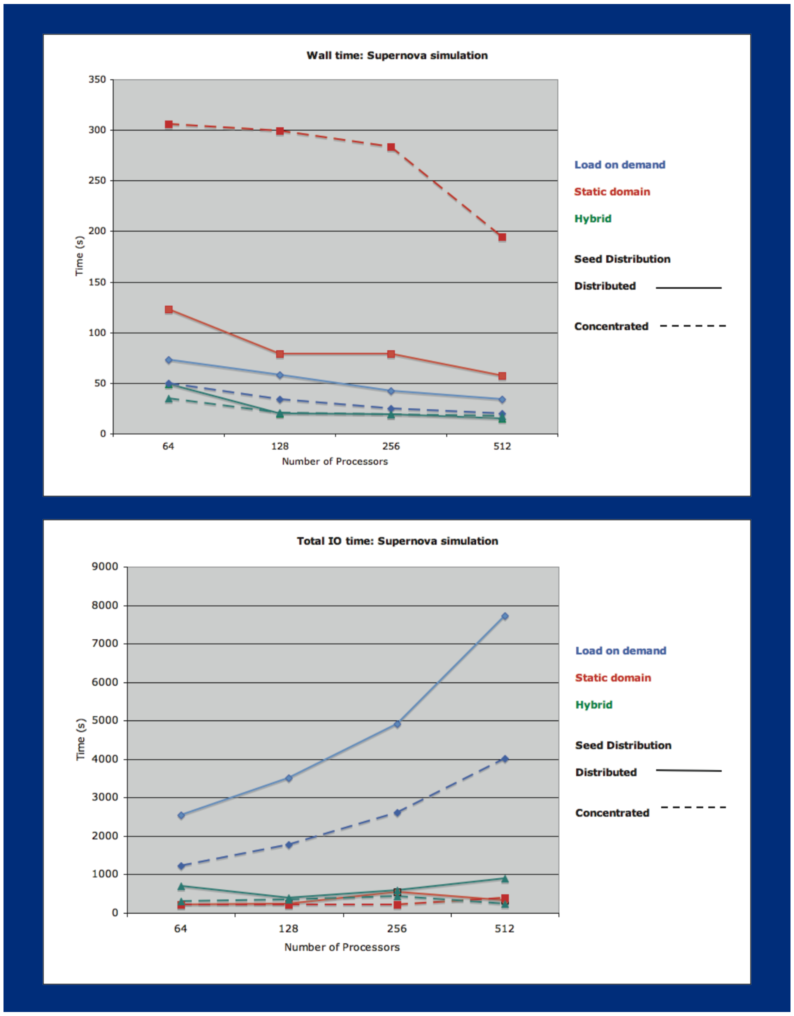


Figure 7. *These images document scalability results for computation time and IO time of different parallelization strategies on a Supernova simulation. The hybrid master/slave algorithm (green) retains the best-case characteristics of static decomposition w.r.t. to data (red) and seed points (blue).*

From a software engineering perspective, the integration framework is designed with future visualization requirements in mind. It allows fine-grained access at different levels of complexity, and hence facilitates the development of novel visualization techniques as well as the customization of existing algorithms with comparatively little effort. Leveraging VisIt's existing development infrastructure further simplifies this process. For example, this has enabled the easy incorporation of a third-party library for Poincaré analysis for fusion visualization into the existing infrastructure.

In the current release, VisIt offers basic capabilities for integral curve visualization using streamlines and pathlines, and advanced capabilities for Poincaré analysis. Integral surface support and Lagrangian analysis methods such as Finite-Time Lyapunov Exponents are currently being worked on for inclusion in an upcoming release. Overall, the goal is to provide through VisIt a modern, robust and widely applicable tool for vector field visualization applications and research.

Conclusion

Modern vector field visualization algorithm are a key component of successful vector field visualization in scientific applications. VACET researchers have developed and implemented state-of-the-art vector visualization capabilities, based on integral curve methods. In deploying the resulting algorithms in the open-source VisIt visualization tool, VACET provides a robust, stable and scalable toolset for vector field visual data analysis within SciDAC.

Contributors Christoph Garth, Eduard Deines and Kenneth I. Joy, all from the Institute for Data Analysis and Visualization at the University of California, Davis; E. Wes Bethel, Hank Childs and Gunther Weber, all at the Visualization Group at LBNL; Sean Ahern and Dave Pugmire from ORNL; and Allen Sanderson and Chris Johnson at the Scientific Computing and Imaging Institute, University of Utah. All contributors are VACET collaborators.

Acknowledgements This work was supported by the Director, Office of Science, Office and Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 through the Scientific Discovery through Advanced Computing (SciDAC) program's Visualization and Analytics Center for Enabling Technologies (VACET). This research used resources of the National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the Oak Ridge Leadership Computing Facility (OLCF), which is supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Further Reading

VACET
http://www.vacet.org/

VisIt
https://wci.llnl.gov/codes/visit/

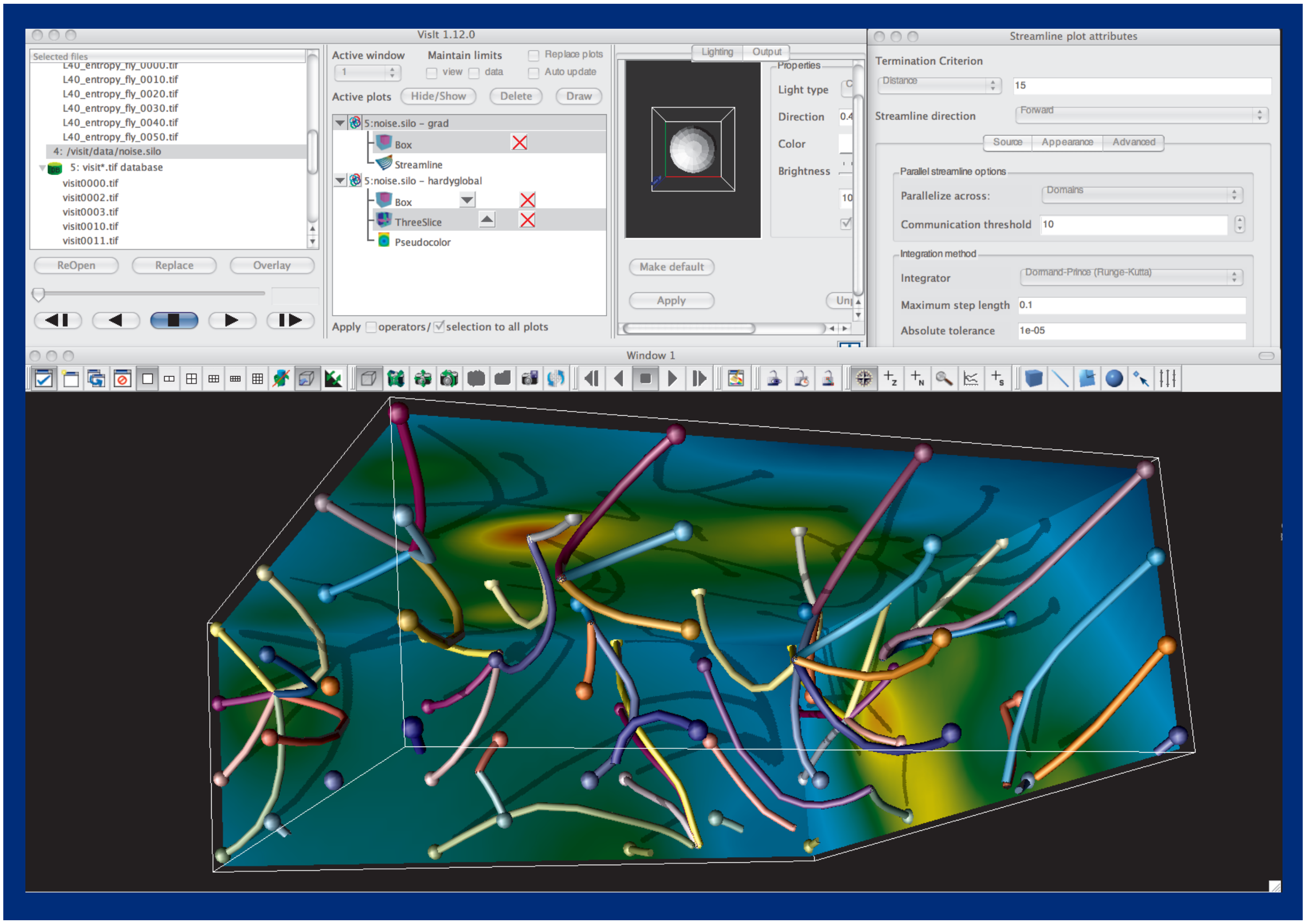


Figure 8. *VACET has deployed streamline techniques (serial and parallel) inside of VisIt, the popular visualization tool used heavily in the Office of Science. This image shows a screenshot of the VisIt user interface and the streamline interaction capabilities.*