Visualization Techniques for Studying Large-Scale Flow Fields from Fusion Simulations

Franz Sauer, Yubo Zhang, Weixing Wang, Stephane Ethier, and Kwan-Liu Ma | University of California, Davis and Princeton Plasma Physics Laboratory

A joint study between computer scientists and fusion scientists resulted in the development of visual tools for studying patterns in flow fields from large-scale magnetic confinement fusion simulations. These tools aid scientists in managing the visual complexity of large trajectory datasets and are crucial in locating and understanding subtle features of interest.

Fusion energy research is a complex field dedicated to developing low-cost sustainable energy sources. Scientists at the Princeton Plasma Physics Laboratory (PPPL) are currently focusing on studying magnetic confinement in an attempt to develop practical fusion energy. The most promising current confinement method is that of the toroidal-shaped device known as a Tokamak. While effective, it's still subject to significant energy loss via transport out of the hot plasma core due to magneto-hydrodynamic instabilities and turbulence.

Consequently, a great deal of effort has been put into the development of numerical simulations of fusion experiments. Specifically, scientists have developed the Gyrokinetic Tokamak Simulation (GTS) code[1,2] to study microturbulence and transport in Tokamak devices. Depending on the type of simulation, this code has the ability to produce multiple data types (such as scalar and vector fields or particle data). Our work focuses on visualizing and exploring properties found within flow velocity fields, vector fields describing the fluid system's motion. By gaining a better understanding of underlying patterns in flow data, scientists can continue to make new contributions to the field of plasma physics.

Particle advection is known to be an effective tool for visualizing and interpreting flow fields. By injecting some form of "matter" into key points of the flow field, scientists can visualize how it's moved, or advected, to gain an understanding of its potentially complex properties. While an effective tool for visualizing velocity fields, particle advection on its own has its limits in being able to highlight specific features of interest to fusion scientists. To capture the highly detailed, turbulent nature of these large-scale datasets, a high density of seeded particles is required to visualize any subtle patterns in the flow. This leads to a number of perceptual issues such as occlusion and overplotting. The toroidal-shaped chamber used in fusion research presents additional problems, as the shape itself is self-occluding unless viewed directly from above or below. As a result, we focus on developing a set of mapping, filtering, and projection methods to aid scientists in locating and understanding important elements in their simulation data.

Our contributions are characterized by the identification of which methods are useful in close consultation with domain scientists and their integration into a usable system. We demonstrate the effectiveness of our methods by using real simulation data produced by the GTS code.

Gyrokinetic Fusion Simulations

To develop practical fusion energy, fusion devices need to sustain the high temperatures required for reactions to take place. Magnetic confinement techniques are used in an attempt to contain the plasma and prevent it from coming in contact with chamber walls and cooling. Instabilities caused by

 1521-9615/16/$33.00 Copublished by the IEEE CS and the AIP

Related Work in Visualizing Fusion Simulations

Several efforts in visualization have focused on data types produced by the Gyrokinetic Tokamak Simulation (GTS) code. Chad Jones and colleagues[1] implemented an exploration system for particle data from gyrokinetic simulations, developing a user interface that enables interactive selection of particles with multiple ranges through parallel coordinates. David Crawford and colleagues[2] focused on visualization of the scalar field data from the GTS code, implementing a hardware-accelerated volume visualization of the scalar Maxwell potential data produced by the simulations. In contrast, our work focuses on visualizing the vector flow field data produced by the GTS code.

Kesheng Wu and colleagues[3] took a more quantitative approach in developing a scalable way of extracting regions from the toroidal meshes produced by the GTS code. By taking advantage of the specific mesh structure, they were able to develop an efficient connected component-labeling algorithm to construct regions of interest from mesh points. Ningning Dang and colleagues[4] also developed a Web-based visualization tool for analyzing waveform data produced by the Experimental Advanced Superconducting Tokamak (EAST).

In addition, related work has been done in visualizing data from other fusion simulations. Greg Schussman and colleagues[5] focused on visualizing magnetic field data, using a set of illuminated and haloed field lines to represent the shape of the magnetic field, leading to insights about magnetic island chains, which are points of instability within the field. Allen Sanderson and Xavier Tricoche and their colleagues[6,7] have done extensive work on visualizing features using Poincaré maps, deriving a set of functions to identify and characterize flux surfaces and island chains found in the magnetic field and integrating these into an interactive visualization tool.

References

1. C. Jones et al., "An Integrated Exploration Approach to Visualizing Multivariate Particle Data," *Computing in Science & Eng.*, vol. 10, no. 4, 2008, pp. 20–29.
2. D. Crawford et al., "Visualizing Gyrokinetic Simulations," *IEEE Visualization*, 2004, pp. 59–66.
3. K. Wu et al., "Finding Regions of Interest on Toroidal Meshes," *Computational Science and Discovery*, vol. 4, no. 1, 2011.
4. N. Dang et al., "WebScope: A New Tool for Fusion Data Analysis and Visualization," *Real Time Conf.*, 2009, pp. 141–143.
5. G. Schussman et al., "Visualizing DIII-D Tokamak Magnetic Field Lines," *IEEE Visualization*, 2000, pp. 501–504.
6. A.R. Sanderson et al., "Analysis of Recurrent Patterns in Toroidal Magnetic Fields," *IEEE Trans. Visualization and Computer Graphics*, vol. 16, no. 6, 2010, pp. 1431–1440.
7. X. Tricoche, C. Garth and A. Sanderson, "Visualization of Topological Structures in Area-Preserving Maps," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 12, 2011, pp. 1765–1774.

microturbulence within fusion devices are detrimental to magnetic confinement. Scientists therefore use complex simulations to study the origin of such instabilities.

The GTS Code

Scientists at PPPL developed the GTS code,[1,2] a massively parallel particle-in-cell code, to solve modern gyrokinetic equations and simulate the turbulence and transport that occurs in physical plasma experiments. It's a highly scalable code designed to run on petaflop systems on hundreds of thousands of processor cores. Moreover, it has already been successfully applied to several studies regarding transport and confinement.

The specific simulation that we explore in this article, known as MODEL_KE95_TiTeOMP2, was carried out to study spontaneous plasma flow generation in the electron turbulence regime, which is highly relevant to physical burning plasma experiments. Specifically, it looks at how collision-less trapped electron mode (CTEM) turbulence drives plasma rotation in Tokamaks. Such rotations have key impacts on the performance of plasma confinement in fusion devices.

Simulation Data

The simulation data lies on an unstructured mesh separated into a set of poloidal slices that are equally spaced around the torus. A vector at each mesh point location represents the flow velocity field, with 32 slices consisting of more than 200,000 mesh points per slice each for a total of over 6 million points of data. The data is also time-varying and available through a per timestep format. The simulation as produced by the GTS code runs through more than 40,000 timesteps. However, due to storage concerns, it's often convention to only save a smaller subset of output steps. In this particular simulation, every 50th timestep is saved to disk, resulting in 830 timesteps that the visualization system can use.

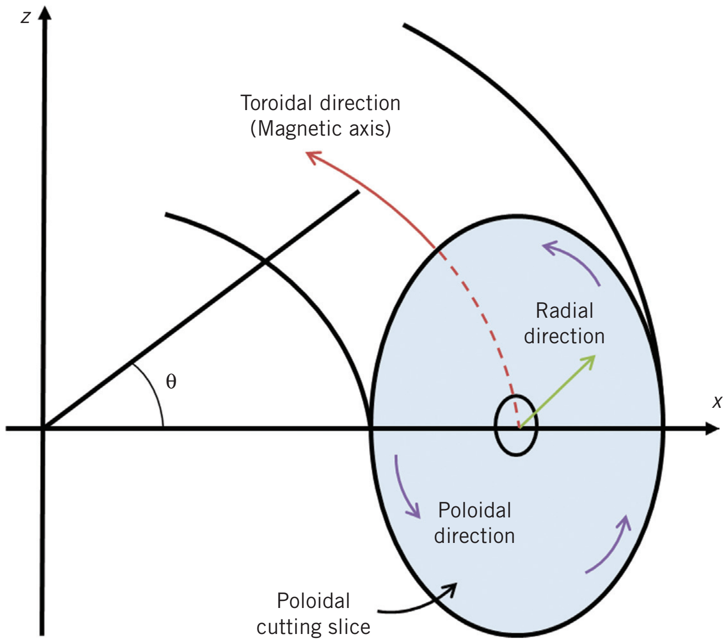


Figure 1. Reference image for geometric terminology within the torus volume. Portions of the torus can be cut away using a "poloidal cutting slice." The poloidal and radial directions run parallel to the slice while the toroidal direction runs through the center of the Tokamak chamber and is always perpendicular to a slice.

Figure 1 describes the toroidal shape of the simulation domain and the geometric terminology used throughout this article.

Implementation

Our system is implemented in two main portions: a preprocessing phase, in which trajectories are generated from the raw simulation data, and the visualization system itself, which allows users to interact with the data in real time.

Trajectory Generation

We generate trajectories by advecting 30,400 particles initially distributed on 32 slices evenly spaced throughout the torus (with 950 particles per slice). These are further distributed around 19 concentric noncircular rings that represent the shape of the magnetic flux surface running through the torus. Seeding particles in this way is beneficial because it makes areas of low turbulence very obvious. In low-turbulence areas, the particles will flow along the magnetic flux surface: if particle trajectories are projected onto such a slice, they'll draw out the exact concentric rings used to seed those particles. To accelerate the particle advection, we implement this step on the GPU using fourth-order Runge-Kutta in CUDA. We also calculate filtering information from each trajectory based on its geometric shape, using this information to isolate groups of trajectories with similar properties.

While it's beneficial to be able to trace a single particle's trajectory throughout the entire simulation, such a representation has some disadvantages because particles tend to quickly move away from parts of the velocity field with high divergence. This leads to portions of the torus with little to no trajectories, making it difficult to understand the velocity field in its entirety. To address this issue, we also allow the user to generate trajectories in which particles are constantly reseeded after a set number of timesteps. Older particles are removed while new particles are generated, filling in the empty gaps.

Visualization System

The visualization itself is implemented using OpenGL and consists of trajectories rendered as a set of illuminated pathlines. The visualization can be animated, allowing users to follow trajectories and obtain an understanding of how the flow field changes throughout the simulation. Each line represents a trajectory's backward history between the current step of the animation and a set number of steps in the past (as defined by the user). Adjusting this value not only allows the user to view trajectories between key simulation times but also adjusts the length of rendered lines to help reduce pathline clutter.

Users have full control of the visualization and can adjust parameters in real time. Intuitive keyboard and mouse controls can adjust the view camera, and play or pause the animation. Other visualization settings are found on a sidebar widget that allows users to select any of the data exploration tools described in the next section. Users can adjust the color function, apply any number of filters, or select view projections in real time as the animation plays.

Data Exploration Techniques

To aid in understanding the data, we use exploration techniques to highlight features represented by the trajectories. Useful ways of coloring, filtering, and projecting trajectories provide scientists with numerous options for verifying and interpreting their simulation results.

Color Functions

By changing the color of rendered trajectories, we can help scientists isolate certain flow patterns. Moreover, because properties vary along each trajectory, its color will vary as well. For example, if a trajectory's color is mapped to its velocity, then that will also vary as the particle speeds up or slows down. This can provide users with insight from just a single frame of the animation.

Toroidal velocity. Users can color trajectories based on their toroidal velocity (a projection of the particle velocity in the toroidal direction): a blue-white-red color scheme represents this velocity, with blue portions representing a high positive toroidal velocity (the particle was moving quickly around the torus in a counterclockwise fashion), white portions representing a very small toroidal velocity (the particle was either stationary or moving perpendicular to the toroidal direction), and red portions representing a high negative toroidal velocity (the particle was moving quickly around the torus in a clockwise fashion).

From Figure 2, it's clear that most trajectories have either a high positive or a high negative toroidal velocity. It's also easy to see that trajectories moving in different directions tend to occupy different portions of the torus. Figure 3 shows a zoomed image of a few trajectories that change color between blue and red, indicating a change in the particle's toroidal direction. By studying how quickly the trajectory fades from blue to white to red, scientists can gain an understanding of how quickly the particle decelerated and accelerated before and after changing direction. In this image, the trajectories transition between blue and red quickly, indicating a fast acceleration and deceleration when changing direction.

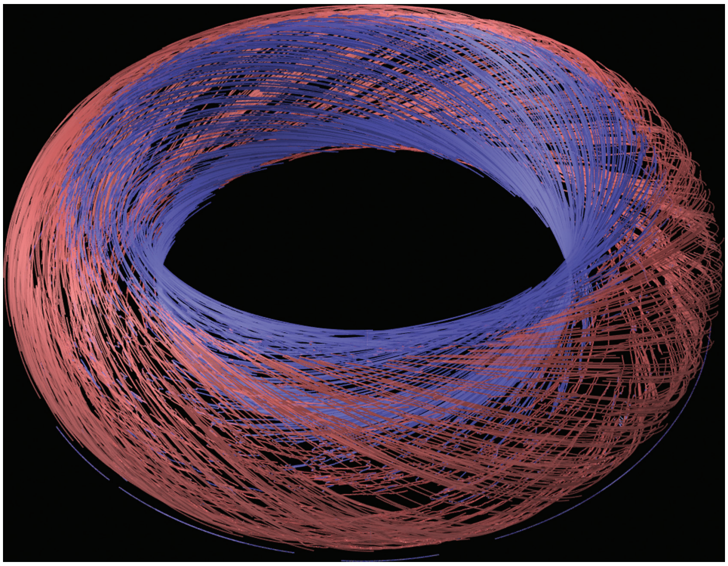


Figure 2. A view of trajectories in the torus colored according to their toroidal velocity. Trajectories in blue move counterclockwise around the torus, while trajectories in red move clockwise.

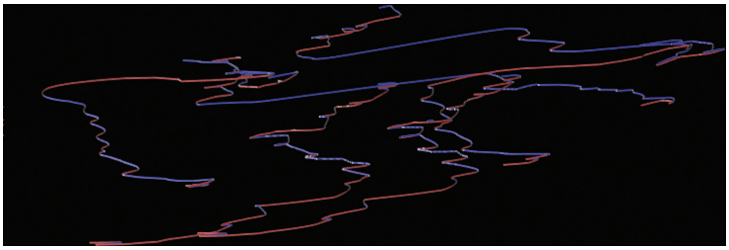
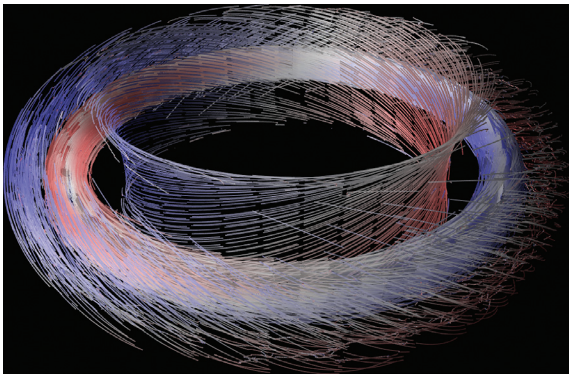


Figure 3. A zoomed view of trajectories from Figure 2 with a high number of turns colored according to their toroidal velocity. By studying how quickly the trajectory fades from blue to white to red, scientists can gain an understanding of how quickly the particle decelerated and accelerated before and after changing direction.

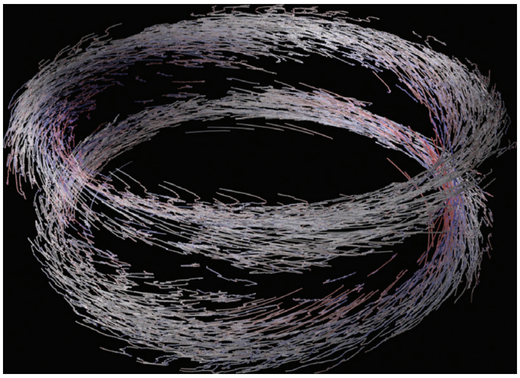
Flux normal velocity. In addition to the toroidal velocity projection, we also provide a method for users to color trajectories based on flux normal velocity (a projection of the particle velocity along magnetic flux surface normals). Blue portions represent a high positive velocity along the flux surface normals (the particle was moving quickly towards the outer walls of the torus), white portions represent a very small flux normal velocity (the particle was either stationary or moving perpendicular to surface normals), and red portions represent a high velocity against the surface normals (the particle was moving quickly toward the magnetic axis). Note that, while close, this motion isn't quite the same as motion in the radial direction as the chamber isn't a circular toroid. In addition, animating the trajectories shows that areas of positive and negative flux normal velocity tend to swap positions throughout the simulation.

Filtering Techniques

Visualizing a large amount of trajectories at once can become overwhelming and often results in too much occlusion to be able to visualize subtle features in the velocity field. This is especially an issue for fusion simulations because occlusion is inherently present in the torus shape itself when viewed from certain angles. Although we allow the user to



(a)



(b)

Figure 4. The difference between showing only trajectories with (a) a low number of turns and (b) a high number of turns. Trajectories are colored according to their flux normal velocity.

display fewer trajectories indiscriminately (that is, displaying only every *n*th particle), we also implement several filters that can hide or show trajectories based on their geometrical properties. We chose these filters based on the properties that fusion scientists are looking for in their data.

Number of turns. We implement a turns filter, which can show or hide trajectories based on the number directional changes throughout its trajectory. Such a filter is very useful for studying various features of the velocity field. The turns filter can easily highlight portions of the field that have numerous changes in direction (high turbulence) or few changes in direction (zonal flows).

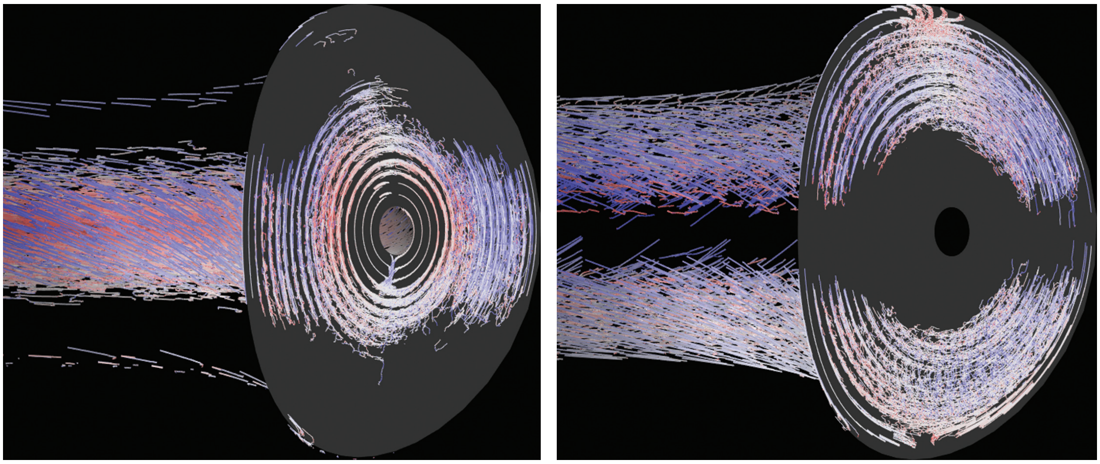
In our implementation, we characterize one "turn" as a directional change that's greater than or equal to 90 degrees. We implement this by computing the dot product between the direction vector of the start of the trajectory and the direction vector of subsequent positions. We iterate through these subsequent vectors until the dot product is negative (the angle is greater than 90 degrees). We then increment a turns counter, set this position as our new start position, and continue to iterate along the remaining positions along the trajectory. This allows us to include both sudden and gradual directional changes. Once this is computed, the visualization system can then choose to hide trajectories whose total number of turns is below or above a user-defined threshold.

Figure 4 highlights which portions of the torus are dominated by trajectories with low and high amounts of turns respectively. The locations of these trajectory types are confined to different parts of the torus and are representative of how the velocity field behaves in those parts. The turns filter can be a quick tool for identifying parts of the torus with high amounts of turbulence as those parts are likely to consist of trajectories with a higher number of turns.

Radial displacement. We also implement a radial displacement filter that can show or hide trajectories based on the amount of motion in the radial direction. Scientists are especially interested in the radial motion of particles because this highlights energy transport into and out of the hot plasma core. We calculate this value by projecting the displacement vector of the particle for each step in the radial direction. We accumulate the magnitude of this displacement over the entire length of the trajectory so that both motion toward and away from the toroidal axis increases this filtering value. The visualization system can then choose to hide trajectories whose total radial displacement is below or above a certain threshold.

Figure 5 highlights which portions of the torus are dominated by trajectories with low and high amounts of radial displacement, respectively. These trajectories are projected onto a poloidal cutting slice for easier viewing. Clearly, trajectories with different amounts of radial displacement dominate entirely different parts of the chamber—for example, trajectories with higher amounts of radial displacement aren't found anywhere near the toroidal axis.

Entropy. We also implement a way to filter trajectories based on their entropy. Stéphane Marchesin and colleagues[3] used the entropy measure for



(a) (b)

Figure 5. The difference between showing only trajectories with (a) a low radial displacement and (b) a high radial displacement. Trajectories are colored according to their flux normal velocity.

aesthetic purposes in streamline selection. Specifically, they use the following definition of angular entropy in their selection process:

$$E_A = -\frac{1}{\log_2(m)} \sum_{j=0}^{m-1} \log_2 \frac{A_j}{L_A}, \tag{1}$$

where m is the number of streamline segments, A_j is the absolute value of the angle at the jth streamline joint, and L_A is the total angular variation along the streamline (the sum of the absolute values of A_j). The entropy measure is meant to quantify the amount of angular variation between segments making up the streamline. Using this definition, a streamline forming a near perfect circle or a straight line will have a small entropy measure because there's little angular variation. Trajectories with a changing angle along segments will have a larger entropy measure.

To provide users with an alternate way of trajectory filtering, we extend this methodology and apply it to our particle trajectories. We calculate an entropy measure for each trajectory in a similar manner and then allow the visualization system to filter trajectories below or above a user-defined threshold. Although originally designed as an aesthetic tool, the entropy filter can also give physical insights, such as an alternate way of identifying areas of high turbulence.

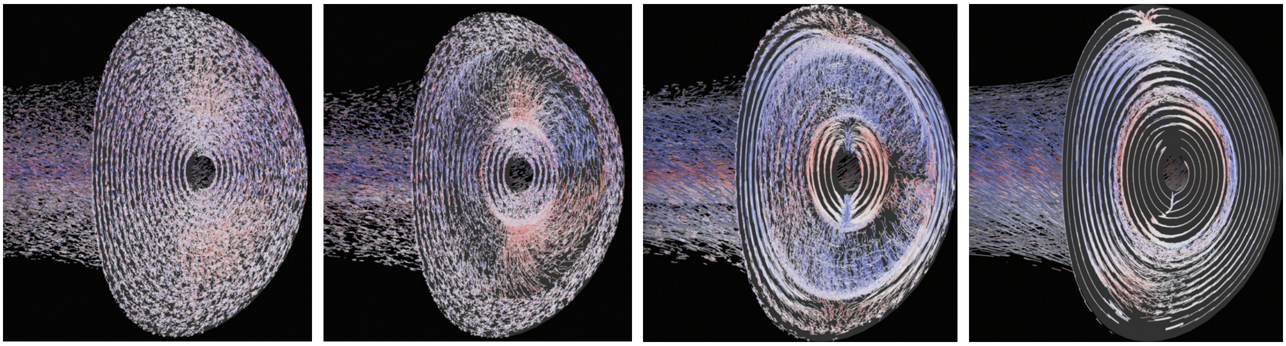
The entropy and turns filter tend to detect similar properties, but there are some subtle differences between the two. For example, a trajectory looping around in a perfect circle could potentially consist of a large number of turns, although its entropy will still be zero because the angle is constant throughout the trajectory. Such differences allow the entropy filter to reveal or hide information that the turns filter cannot.

Projections

Visualizing the absolute positions of the trajectories in the torus shape isn't always the best method for highlighting and understanding patterns present in the simulation data. We find that projecting the trajectories onto several cutting planes or into a different volumetric region offers different perspectives and can help reveal new scientific insights.

Poloidal cutting slice. The torus shape naturally makes it difficult to visualize particle motion in the radial direction, a property of significant interest to scientists. We therefore insert a cutting plane into the torus perpendicular to the toroidal axis. Projecting particles onto this slice presents a view as if we're looking through the torus along the toroidal direction.

Moreover, we can project trajectories from different toroidal positions to represent different parts of the torus on the slice. For example, by projecting only trajectories near the slice, scientists can gain an understanding of the velocity field in that local vicinity. Projecting the trajectories throughout all of half of the torus gives a more overall picture of the velocity field.



(a) (b) (c) (d)

Figure 6. A view of the poloidal cutting slice with a projection of trajectories located in half of the torus. Trajectories are colored according to their flux normal velocity, with each image representing a different phase of the fusion simulation: (a) phase 1, low initial turbulence; (b) phase 2, exponential growth in turbulence; (c) phase 3, turbulence becomes saturated and peaks; and (d) phase 4, turbulence oscillates and decreases.

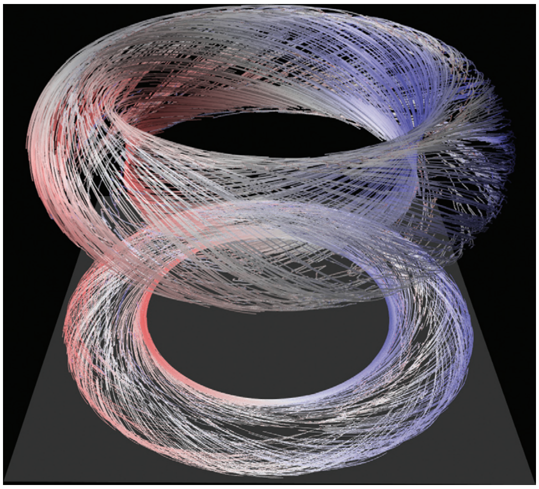


Figure 7. A floor projection of trajectories colored according to their flux normal velocity. The floor projection makes it clear that trajectories with a very high positive or a high negative flux normal velocity occur near the ring's inner and outer edges.

Figure 6 not only shows an example of this projection but also reveals how this view can be used to identify the different phases of turbulence in this particular simulation. Phase 1 starts out with a small but growing turbulence. Next, phase 2 consists of a continuing exponential growth in turbulence. Phase 3 is where the turbulence becomes saturated and peaks. Finally, phase 4 consists of an oscillating decrease of the turbulence to reach a steady state. Note that in the last image, most of the trajectories follow along the magnetic flux surfaces as we would expect during low turbulence.

Floor projection. Although the poloidal cutting slice is an effective method for visualizing the radial motion of trajectories, it makes it difficult to understand differences in the field at different toroidal positions. We therefore provide a floor projection of the trajectories onto a plane placed below the torus. Such a projection is useful for visualizing patterns that occur along the toroidal direction.

Figure 7 shows an example of the floor projection, highlighting new patterns in the flux normal velocity for trajectories. It's already evident from the full torus view that the flux normal velocity switches sign between the left and right halves of the torus. However, the floor projection makes it clear that trajectories with a very high positive or a high negative flux normal velocity occur near the ring's inner and outer edges. Trajectories in other portions are colored white and travel perpendicular to flux surface normals. Note that the particles in the torus can be hidden, thus showing only the floor projection as occlusion-free.

Expanded view. Although projecting the trajectories onto a 2D plane can eliminate some occlusion and provides new perspectives, it always result in a loss of some form of spatial information. We therefore use an "expanded view" to remove the inherent occlusion present in the torus shape without the loss of a spatial dimension. To generate this view, we project the trajectories into a rectangular volume representing an unfolded version of the original torus. Figure 8 shows the manner in which this volume is created. The torus is cut and unfolded twice into a 3D rectangular volume with trajectories projected accordingly.

As described earlier, scientists are especially interested in the radial motion of particles. However, they're also concerned with how this motion varies as a function of the poloidal direction. The main motivation behind the expanded view is to lay the curved torus coordinate system onto a set of flat Cartesian coordinates. From Figure 8, the *x*, *y*, and *z* directions in the expanded view represent the radial, toroidal, and poloidal directions, respectively, allowing scientists to easily follow and compare radial motion as a function of poloidal angle as each coordinate now points in a constant direction. Figure 9 shows an angled view of the 3D volume along with trajectories projected onto a slice. The image also shows clearer patterns in the toroidal direction that are no longer obscured by parts of the torus shape.

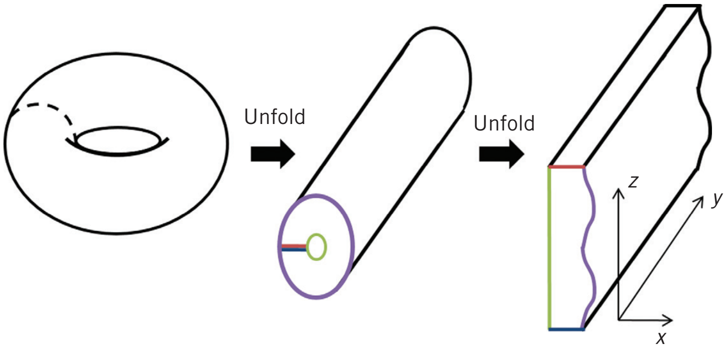


Figure 8. The manner in which the torus is unfolded to create the expanded projection. The *x*, *y*, and *z* directions in the expanded view represent the radial, toroidal, and poloidal directions, respectively.

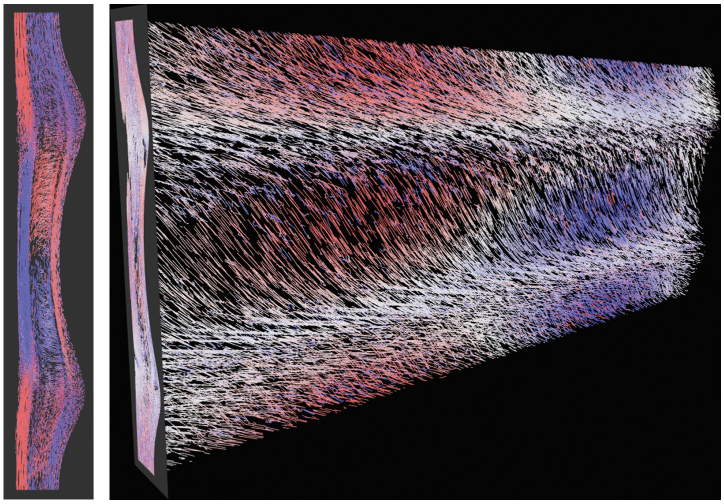


Figure 9. An angled view of the expanded 3D volume with trajectories colored according to (right) their flux normal velocity and (left) a slice of the volume with trajectories colored according to their toroidal velocity. The image also shows clearer patterns in the toroidal direction that are no longer obscured by parts of the torus shape.

Discussion

Our collaborators at PPPL gained some specific insights from this visualization tool.

Evaluation of Usability

As the direction parallel to the magnetic field lines is the fastest motion in the simulation, the toroidal color function helps identify interesting patterns in the flows. Specifically, it shows the opposite direction of the flows between the low- and high-field side of the Tokamak. The flux normal color function is very helpful in seeing the evolution of radial flows, namely, how different regions have opposite flow directions. Toward the end of the simulation, it's evident that a very small radial component on the low-field side can reflect a well-developed zonal flow.

The turns filter as well as the entropy filter are very useful in separating the more regular flows from the turbulent ones. This is important because regular flows can be associated with well-known zonal flows and long-wavelength modes. The radial displacement filter is also very useful in isolating regions of high and low radial motion, making patterns much clearer.

The full torus view is a great way of getting a global sense of the flows. In addition, watching the time-evolved animation is even more informative because it's easy to see the initial slow random motion of tracer particles develop into a fast and complex flow profile. The poloidal cutting slice makes it similarly easy to observe radial and poloidal motion, and is a view commonly used in fusion research. The floor projection led to some surprising results as it shows an interesting radial flow pattern in areas between the two halves of the torus. Finally, although it's sometimes initially difficult to keep in mind the direction that each axis represents, the expanded view offers new perspectives. It's also useful because the side that's normally occluded by the front part of the torus is now always visible.

Scalability

Our techniques address two types of scalability: interactive and perceptual. Interactive scalability describes a system's ability to keep up with user input by displaying new results in real time. As the simulation data becomes larger (such as through a

Although the motivations behind our tools are specific to studying physical properties in fusion simulations, our visualization techniques and system can be applied to other datasets as well.

denser mesh), it becomes more difficult to handle. However, our system lets us represent the field by visualizing the trajectories rather than the raw field data. We can easily organize trajectories by using indexing or levels of detail to ensure interactivity and responsiveness. Viewing even larger numbers of trajectories can also be achieved by splitting the particles across multiple GPUs and will be investigated in a future study.

Perceptual scalability represents the amount of information that humans can perceive from their display. Factors such as clutter or overplotting are common causes of such limits. Our filtering techniques and view projections help deal with these limits. In this case, we can choose to present specific trajectories of interest from a large dataset without overwhelming the user with too much information.

Applications to Other Flow Field Datasets

Although the motivations behind our tools are specific to studying physical properties in fusion simulations, our visualization techniques and system can be applied to other datasets as well. Combustion scientists are especially concerned with the presence of turbulence and have developed large-scale simulations to understand the key reactions taking place. Because these simulations produce large amounts of particle data, we can apply our turns and entropy filters to highlight portions of high turbulence. Another example is understanding flow around vehicles to reduce aerodynamic drag. Scientists often utilize simulations to study velocity fields that form around bodies of interest. Our visualization techniques would help spot areas of turbulence and identify faulty design areas.

This study presents a joint collaboration between computer scientists and fusion scientists in developing a set of mapping, filtering, and projection methods for studying patterns in flow fields from fusion simulations. Through our collaborative effort, we make contributions in identifying which techniques are useful and integrate them into a coherent usable system.

This work also presents a useful stepping stone to further address scalability concerns by implementing our visualization techniques in situ. By utilizing recently published algorithms,[4,5] we can construct high-resolution flow lines and apply our techniques to the trajectory data before intermediate timesteps are discarded, helping us achieve a level of postprocessing analysis that was previously impossible. We plan to incorporate corresponding scalar field data alongside our visualization and hope to apply our tools to other flow datasets.

Acknowledgments

This research has been sponsored in part by the US National Science Foundation via grants NSF IIS-1320229, and also by the US Department of Energy through grants DE-FC02-06ER25777, DE-SC0005334, and DE-FC02-12ER26072. We also acknowledge the support of PPPL DOE contract number DE-AC02-09CH11466 and NERSC, which is where we performed the GTS simulations.

References

1. W.X. Wang et al., "Gyro-Kinetic Simulation of Global Turbulent Transport Properties in Tokamak," *Physics of Plasmas*, vol. 13, no. 9, 2006.
2. W.X. Wang et al., "Nonlinear Flow Generation by Electrostatic Turbulence in Tokamaks," *Physics of Plasmas*, vol. 17, no. 7, 2010.
3. S. Marchesin et al., "View-Dependent Streamlines for 3D Vector Fields," *IEEE Trans. Visualization and Computer Graphics*, vol. 16, no. 6, 2010, pp. 1578–1586.
4. H. Yu, C. Wang and K.-L. Ma, "Parallel Hierarchical Visualization of Large Time-Varying 3D Vector Fields," *Proc. ACM/IEEE Conf. Supercomputing*, 2007, p. 24.
5. D. Camp et al., "GPU Acceleration of Particle Advection Workloads in a Parallel, Distributed Memory Setting," *Eurographics*, 2013, pp. 1–8.

Franz Sauer is a fourth-year graduate student at the University of California, Davis, studying computer science and scientific visualization under Kwan-Liu Ma. His research interests include data visualization, large-scale scientific simulations, computer graphics, and physics. Sauer received a BS in physics from the California Institute of Technology. Contact him at fasauer@ucdavis.edu.

Yubo Zhang is a senior graphics architect at Nvidia. His current research interests include numerical methods, GPU architecture, computer graphics, and visualization. Zhang received a PhD in computer science from the University of California, Davis. Contact him at ybzhang@ucdavis.edu.

Weixing Wang is a principal research physicist at the US Department of Energy Princeton Plasma Physics Laboratory. His research interests include theory and computation of plasma micro-instabilities, turbulent and collisonal transport in magnetic fusion experiments, gyrokinetic simulation, and advanced simulation algorithms. Wang received a PhD in physics from the National Institute for Fusion Science in Japan. Contact him at wwang@pppl.gov.

Stephane Ethier is deputy head of the Computational Plasma Physics Group at the US Department of Energy Princeton Plasma Physics Laboratory. His research interests include high-performance computing and large-scale particle simulations of microturbulence in magnetic confinement fusion devices. Ethier has a PhD in physics from the Institut National de la Recherche Scientifique: Energie et Materiaux in Montreal, Canada. He's a member of the American Physical Society (APS) and ACM. Contact him at ethier@pppl.gov.

Kwan-Liu Ma is a professor of computer science and the chair of the Graduate Group in Computer Science (GGCS) at the University of California, Davis, where he leads the VIDi research group and directs the UC Davis Center for Visualization. His research interests include visualization, high-performance computing, and user interface design. Ma received a PhD in computer science from the University of Utah. He's an IEEE Fellow. Contact him at ma@cs.ucdavis.edu.

cn *Selected articles and columns from IEEE Computer Society publications are also available for free at http://ComputingNow.computer.org.*