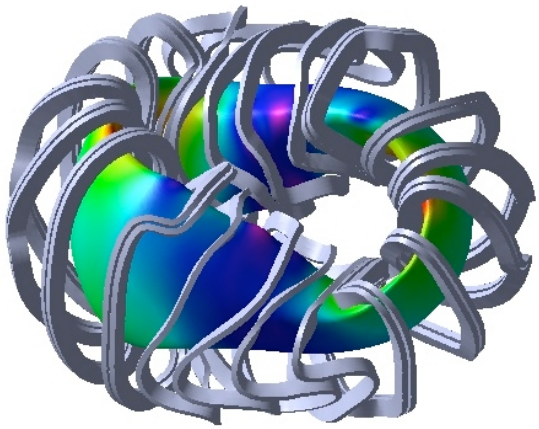
I. **DESCRIPTION OF SCIENTIFIC PROJECT – D. A. Spong**

Web pages:
http://www.ornl.gov/fed/Theory/stci/stellarator_theory.html
http://www.ornl.gov/fed/mhd/mhd.html

Our group works on the optimization and physics analysis of stellarators. These are non-axisymmmetric toroidal fusion devices that utilize optimal 3-dimensional 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 (see Fig. 1).



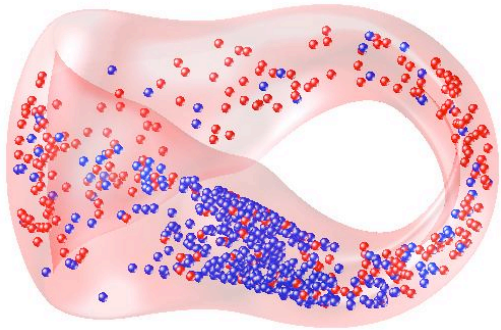


Figure 1 (left side) – Magnetic flux surface and modular coils for a compact quasi-poloidal stellarator (QPS). Color coding on the flux surface indicates the magnetic field strength (blue is low field, red is high field).
Figure 2 (right side) – One frame from a stellarator particle orbit simulation in a compact stellarator (red spheres are passing particles, blue spheres are locally trapped particles).

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.

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 (see 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 (see Figure 3) involves solving field equations in 3D have that have both local and long range couplings; these lead to large dense matrix problems.

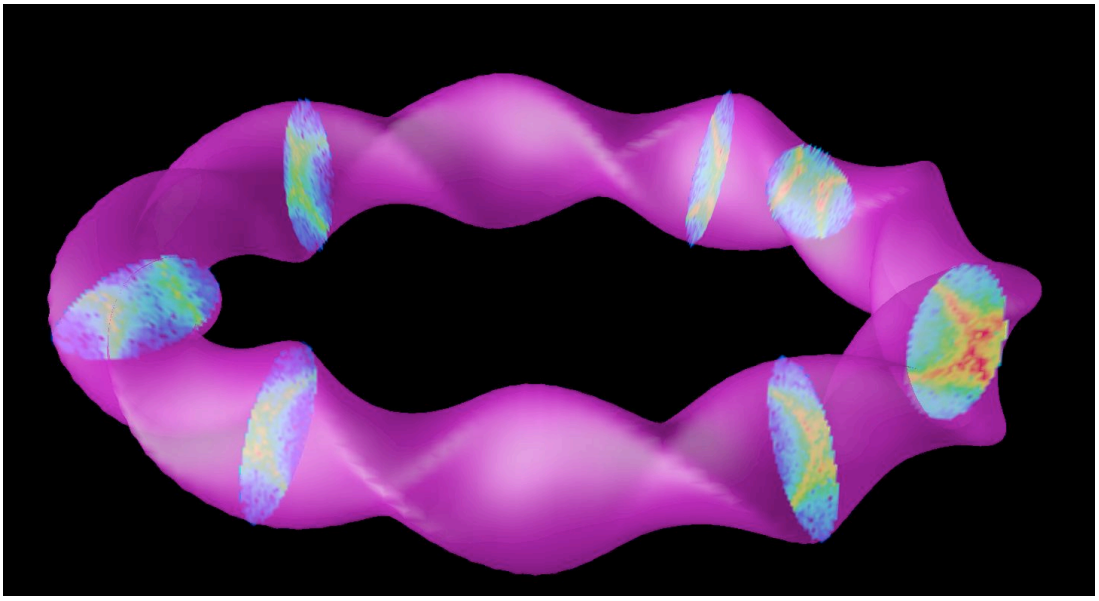


Figure 3 – RF wave field visualization in the LHD (large helical device) stellarator. The purple surface is the outermost magnetic flux surface. Color coded RF wave field strengths are shown on 7 cut planes at fixed toroidal angles.

II. **DATA REQUIREMENTS FOR SCIENTIFIC VISUALIZATION**

Most of the data we currently visualize involves scalar fields localized to a single 3D surface. The typical size of this data set 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 50Mb; 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 data sets. The limits are largely set by what is 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 data sets. Although such data sets 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 is 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 terabytes) that is well beyond the capacity of any resource we expect to be able to access at this time.

III. VISUALIZATION SOFTWARE/HARDWARE CURRENTLY IN USE

We do visualizations both on UNIX/Linux workstations and on Windows and Mac PC's. We use a variety of software, including AVS5 AVS/Express, OpenDX, Mathematica, IDL, as well as some direct OpenGL programming. This works well as long as we constrain the size of the datasets we work with, as indicated in section 2.

IV. VISUALIZATION ISSUES WHERE IMPROVEMENT IS NEEDED

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 (either too basic or too advanced) that are not optimal for us. 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 only ray trace within oddly shaped geometric regions (i.e., toroidally shaped devices) and not 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 data sets onto multi-resolution grids so that the user can view both the big picture as well as being 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 Terabytes) 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 5 dimensional spaces.

V. **FUTURE SCIENTIFIC VISUALIZATION NEEDS AND GOALS**

As indicated under sections 2 and 4, 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 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.