

A Computational Steering System for Studying Microwave Interactions with Missile Bodies

J. Edward Swan II¹ Marco Lanzagorta^{1,4} Doug Maxwell^{1,5} Eddy Kuo^{1,6} Jeff Uhlmann³
Wendell Anderson² Haw-Jye Shyu² William Smith²

The Naval Research Laboratory
Washington D.C.

Abstract

This paper describes a computer modeling and simulation system that supports *computational steering*, which is an effort to make the typical simulation workflow more efficient. Our system provides an interface that allows scientists to perform all of the steps in the simulation process in parallel and online. It uses a standard network flow visualization package, which has been extended to display graphical output in an immersive virtual environment such as a CAVE. Our system allows scientists to interactively manipulate simulation parameters and observe the results. It also supports *inverse steering*, where the user specifies the desired simulation result, and the system searches for the simulation parameters that achieve this result. Taken together, these capabilities allow scientists to more efficiently and effectively understand model behavior, as well as to search through simulation parameter space.

This paper is also a case study of applying our system to the problem of simulating microwave interactions with missile bodies. Because these interactions are difficult to study experimentally, and have important effects on missile electronics, there is a strong desire to develop and validate simulation models of this phenomena.

Keywords: Modeling and Simulation, Scientific Visualization, Computational Steering, Inverse Steering, Virtual Reality

1 Introduction

This paper describes *computational steering*, which is an effort to make the typical modeling and simulation workflow more efficient. The particular modeling and simulation application where we have applied our efforts is the interaction of high-power microwaves with missile bodies. This application has the goals of 1) better understanding this complicated and hard-to-measure phenomenon, and 2) driving the design of both better missiles and potential anti-missile technologies.

1.1 Computational Steering

Computational steering is an evolution of the traditional batch sequence — 1) execute the model, 2) analyze the results, 3) modify the parameters — that computational scientists have engaged in for decades. This traditional batch process is not always effective,

¹Virtual Reality Laboratory.

²Systems Applications Section.

³Decision Making Technologies Section.

⁴Scientific Visualization Laboratory; Contractor from Scientific and Engineering Solutions.

⁵Louisiana Tech University.

⁶Contractor from ITT Systems & Sciences.

Work performed at US Naval Research Laboratory, Code 5580, 4555 Overlook Ave SW, Washington, DC, 20375-5320, USA. Contact Email: swan@acm.org

especially when executing the model is very time-consuming. In computational steering the same three steps all take place more or less concurrently and online. This places the scientist in the middle of a running experiment, and allows them to steer the parameters while the experiment is underway. The ultimate goal is to enable the scientist to achieve the objectives of the experiment more efficiently and effectively. An overview of computational steering is given by Parker et al. [7].

Computational steering can be done in two directions. In *forward steering*, the scientist directly manipulates the model steering parameters, based on the observed behavior of the model. If the model execution time is sufficiently small, and the graphical visualization sufficiently revealing, this can allow the scientist to efficiently explore the steering parameter space, and discover parameter settings that are optimal in some sense. However, when model behavior is very complex, when there are many steering parameters, or when it takes too long to execute the model, it becomes infeasible for the scientist to manually steer to the desired solution. For these situations *inverse steering* is needed. With inverse steering, the scientist defines desired model behavior, and the system automatically searches the steering parameter space, looking for parameter settings that produce the desired behavior. This effectively exchanges the input and output of the forward steering process.

1.2 Application Area

In this work, we are simulating the interaction of high-power microwaves with missile body casings. These interactions have potentially important consequences for the electronics of the missile. High power microwave radiation could come from either a nearby nuclear explosion or an anti-missile weapon system. Microwaves are just electromagnetic (EM) radiation in a certain frequency band. In our simulations, we set up a simplified model of a missile. We then define an EM radiation source, and begin the simulation such that at time step 0, radiation is just beginning to arrive at the model. The radiation encounters the model and begins a complex series of reflections. Typically, after several hundred time steps, the simulation approaches a *steady state*, where, in the vicinity of the model, the EM field does not change with additional time steps, except for an oscillation inherent in the frequency of the EM radiation source. The simulation typically reveals *hot zones*, which are regions of local maximum EM energy. The locations and intensities of these hot zones are an important application design parameter which our work aims to supply. With this knowledge, engineers can design the missile so that critical electronics are protected from the hot zones.

2 System Design

2.1 System Architecture

Figure 1 illustrates the basic architecture of our system. The scientist interacts with the system using a desktop system or a CAVE.

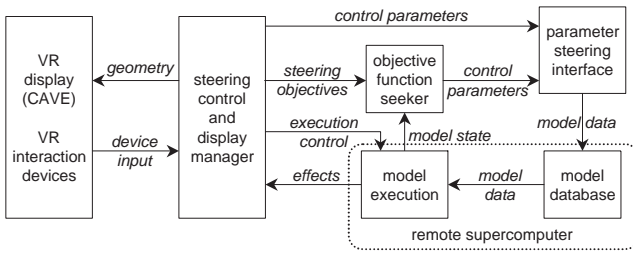


Figure 1: Overall system architecture.

The *steering control and display manager* contains the interface between the scientist and the rest of the system components. This process is implemented within AVS, a standard pipeline-based scientific visualization package. Color plate 1 shows an image of the AVS control panel. *Model execution* interfaces with the other system components, providing both user control and observation. The scientist controls the model execution — starting, stopping, parameter adjustment, etc.— directly from the AVS control panel. The *parameter steering interface* converts these control parameter adjustments into appropriate model updates within the *model database*. These updates could be as simple as setting a single variable, or as complex as setting a number of arrays. The *objective function seeker* provides inverse steering. It obtains the current model state from the executing process, evaluates that state according to the desired objective function, and then adjusts control parameters accordingly.

Our system runs on a high-end Silicon Graphics Onyx 2, with 6 processors and 2 graphics pipes. This machine also runs the CAVE, and we can use it to execute smaller models locally. As shown in Figure 1, we use a remote supercomputer (described in Section 2.2 below), for moderate to larger models. A high-speed ATM network connects the two machines.

2.2 Computational Issues

Simulating EM field interactions requires solving Maxwell’s equations for the propagation of electromagnetic fields. This has been a common simulation application for 30 years, and many standard algorithms for solving Maxwell’s equations exist [9]. However, the limited accuracy of standard algorithms demand a dense computational grid, which consequently makes the execution time untenable for even moderately-sized models. In the early 1990’s James Cole developed the HASP (High-Accuracy Scattering and Propagation) algorithm [1, 2, 3], at the Naval Research Lab (NRL). The HASP algorithm is up to 10,000 times more accurate than standard approaches [3]. This increase in accuracy permits the use of a relatively coarse grid, which greatly improves memory use and execution time. In addition, greater algorithmic stability allows a reduction in the number of necessary iterations. Furthermore, the structure of the algorithm makes it very amenable to parallel processing.

As described in Section 1.2 above, we are studying the effects of high-power microwaves on missile bodies. Because experimental studies of EM interactions with actual missile structures are very difficult to arrange, cooperating researchers have developed a standard test body, termed the *standard cylinder* [6]. The standard cylinder allows multiple research groups to experiment with a common, unclassified, test body, and to easily compare results. Figure 2 is a schematic drawing of the standard cylinder. When tested experimentally, the field strength probe measures the EM field at various places along the center axis. This probe must be included in the model, as it effects the EM field inside the cylinder. While we have

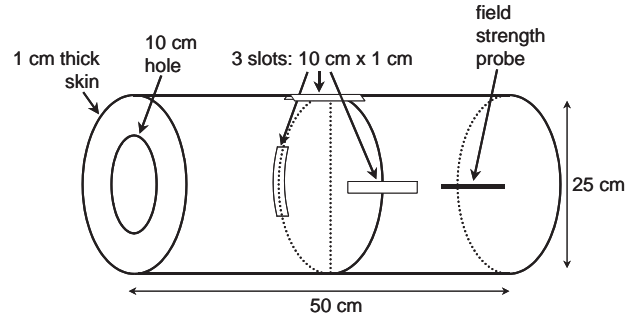


Figure 2: The standard cylinder, a commonly-used model for testing microwave interactions with missile bodies.

used a variety of models in our system, we have most extensively studied the standard cylinder model. All of the Color Plate images in this paper are of the standard cylinder.

The efficiency of the HASP algorithm allows the work reported here. To achieve acceptable computation times, we implemented the HASP algorithm on NRL’s 128 processor Silicon Graphics Origin 2000 supercomputer, using both the automatic parallelization option of the SGI MISPRO Fortran90 compiler, as well as aggressive optimization techniques to speed the scalar code running on each processor. For the standard cylinder, this implementation calculates each time step in a few tenths of a second, and a steady state result (~ 10 billion operations) in a few minutes.

2.3 Visualization Interface

Color Plate 1 shows the user interface for the system. The control panel to the left controls both the simulation execution and the steering parameters. Scientists can start, stop, and resume the simulation at any time step, and can save the system state to disk. They also use this control panel to modify the five scalar *steering parameters*: microwave source frequency, source direction angle (θ, ϕ), polarization angle, and probe insertion distance (note, in Color Plate 1, the five dial controls in the steering control panel). The control panel on the top controls various display parameters, such as slice plane values and glyphs. The bottom of Color Plate 1 shows the AVS network that controls receiving and viewing the results. To the right are two views generated by the two *geometry viewer* modules; a pair of blue arrows indicate which viewer produces each view. Also shown is another module, termed the *VR Viewer*, which generates output for a CAVE display. This module, developed at NRL [5], replaces the AVS geometry viewer and displays information on any VR display. The VR Viewer uses the OpenGL Volumizer library to render volumes using 3D texture memory, and allows the scientist to interact with their dataset via a tracked, hand-held 3D wand. Color Plate 3 shows a scientist interacting with the data in our CAVE facility. It also shows how we sometimes use one or two CAVE walls to display steering parameters, color maps, and other AVS widgets. We mounted a keyboard and mouse on a portable stand to allow users to manipulate these interface elements.

2.4 Inverse Steering

In general, the scientists and designers using our system are interested in two different types of phenomena: the evolution of the EM field over time as it approaches a steady state, and the steady state solution. As described in Section 2.2 above, our system computes EM field evolution in interactive time (several updates per second), and we found the computational steering controls to be an effective

way of exploring parameter space for field evolution. However, the steady state computation is on the order of minutes, which is too slow for interactive parameter exploration. This motivated the development of the *objective function seeker* module (see Figure 1). This module allows the scientist to define an *objective function*, which is a formal way of describing the particular simulation result which is desired. An example of an objective function from our application area is: “*maximize the value of the EM field intensity, for all points inside the cylinder, over one source frequency cycle*”. The module automatically adjusts steering parameters and iterates, converging on the objective function. This module uses various optimization techniques, which are not covered in this paper, but are described in Smith et al. [10].

3 Results and Lessons Learned

3.1 Visualizations

The microwave simulations produce a pair of vector fields, an electrical field and a magnetic field. Visualizing vector fields is one of the most challenging problems in scientific visualization, because — unlike a simple scalar field of amplitude values — there is no direct method for visualizing a vector field. Furthermore, both the field domain and range are multi-dimensional, and so understanding the underlying topological and geometrical structures is a very challenging task.

Visualization Techniques: We tried a number of techniques to visualize the two vector fields. In particular, we tried hedgehogs, streamlines, particle systems, slice planes, and volume rendering. Each of these methods has various pros and cons.

The *hedgehog* method uses arrow icons to represent the 3D vector field at each point in space. The length, direction, and color of the arrow can visualize all the parameters of a 3D vector field. We found that, although this visualization method is frequently good for understanding global structures, it does not easily depict local structures. In particular, when we applied hedgehogs to a dense and locally varying field structure, the result was not very informative and even confusing, due to the clustering and overlapping of vectors in a small region. *Streamlines* treat the 3D vector as a velocity vector, and the resulting displacement is visualized by using continuous lines connecting vectors of constant velocity. Since the simulated EM fields are frequently discontinuous and periodic, disjointed and repeating patterns of the streamlines create a clustering problem similar to that of hedgehog techniques. A *particle system* is a variant of streamlines, which visually animates small particles that flow along the path of the displacement described by the streamlines. This technique provides a better visualization by avoiding the crowding effect found in both hedgehogs and streamlines, but we found that it was not suitable for visualizing our time-varying vector fields. We also experimented with *isosurfaces*, but we found them unable to deal well with the complex multimodal field structures.

Among the most effective were simple color-coded *slice planes* (see the lower right-hand window in Color Plate 1), which did a good job of visualizing field intensity. Since the more interesting aspects of our simulation occur in specific regions — for example along the center axis and around the wall slots of the standard cylinder — we can place a slice plane through these regions of the visualization space. With slice planes we can visualize both local and global field features, and we can avoid clustering effects by using only one plane for visualization through a region.

Finally, we *volume rendered* the field energy density throughout the computational space. We found transparency mapping to be the key parameter for creating understandable visualizations — when set correctly, internal structures were easily observable inside

the field volume. Most of the field features, both global and local, can be displayed by volume rendering. While there are many advantages to volume rendering, we found that details of the internal geometric structure of the field were difficult to examine if parts of the outer layers of the field were rendered opaque by the mapping function that we used. Color Plate 4 shows a volume rendering of the field energy density, with the standard cylinder rendered as a wireframe to aid in visualization. Note the two hot spots which can be seen in the field on each side of one of the slits in the cylinder wall.

Visualization Issues: As mentioned above, our simulation actually produces two 3D vector fields — an electric field, and a magnetic field. If we wish to visualize both fields concurrently, we have to utilize two different techniques. We experimented with combinations of the techniques described above. In particular, we had good results with a combination of hedgehogs and slicing planes. We found that, although volume rendering is good for visualizing field density, it does not blend well with other visualization techniques, because it fills the whole computational space and obscures additional geometric objects. A slice plane, on the other hand, occupies a significantly smaller space, and thus allows other geometric objects to be seen.

In addition to the 3D field structure at a given time step, we are also interested in the behavior of the field over time. Temporal information comes from the transient behavior of the field structure as it evolves from time step 0 to a steady state. At steady state the EM field exhibits a stable standing wave pattern, which is easily seen by observing the field at every n th time step, where n is the number of time steps per cycle of the EM source; at steady state the pattern is stable from observation to observation. Another approach is to compute and display the average magnitude of each point in the field every n time steps. This presents a direct view of the time average power pattern as well as an easy way to assess steady state conditions.

Virtual Reality Environments: As described in Section 2.3 above, in addition to viewing our results via the AVS geometry viewer on our computer monitor, we used our VR Viewer module to view the results in our CAVE facility (Color Plate 3). We found several advantages here. First, since the CAVE screens are much larger than normal computer monitors, we were able to see images that completely surround us with a very large field of view. In addition, we could render the visualization in stereo, which made 3D effects more prominent. We could interact with a hand-held 3D wand, which allowed us to probe and manipulate the field images directly. We believe the CAVE environment helped us to better understand our data.

We also found advantages to driving the CAVE display from our VR Viewer. The main advantage is that the VR rendered image is part of an AVS visualization network. Thus we can interactively perform all the visualization functions provided by AVS, as well as our other custom modules. The same advantage extends to end-user scientists as well: they can visualize their scientific information in a virtual environment (VE) with the same visualization toolkit (AVS) that they are already using. In addition, VE environments such as CAVEs are still relatively uncommon, and may not be located near a scientist’s office. The VR Viewer allows scientists to develop their visualization networks at their own desktops, and only visit the VE facility when they have data to visualize.

Automatic Viewing Parameter Tuning: We have experimented with automatically tuning viewing parameters for complex visualization applications. In particular, we have developed a module which tunes hedgehog displays of vector fields [4]. If the scale of the vector arrows is too small, or the density of arrows is too low, the vector information may not appear in the display. Conversely, if

the scale is too large, or the density of arrows is too high, then the display will be saturated with arrows and all other information will be obliterated. Without the automated parameter selection, the user must test various parameter values by hand.

VolumePro Visualization: In a recent development, we obtained a VolumePro real-time volume rendering board [8], which fits in a high-end PC. We have begun experimenting with visualizing our data using this system; Color Plate 5 shows a preliminary result. The advantages of the VolumePro board are that it provides high-quality volume rendering at very interactive rates, allows interactive rendering not only of still volumes, but also of animated volumes, and of course that it costs an order of magnitude less than an equivalent high-end graphics computer. We have several future plans for the VolumePro board in this project. First, we are developing software that will provide real-time, mouse-based updating of color maps, while the volume dataset is being animated. As finding the color map that best displays the relevant information can be quite difficult and time-consuming, this should speed up this aspect of using a visualization system. Second, we hope to integrate the VolumePro-enabled PC into our overall computation steering system. This will involve writing another custom AVS module, which will drive a volume rendering application on the VolumePro-enabled PC.

3.2 Inverse Steering

We implemented the inverse steering module discussed in Section 2.4 above, and used it to explore a number of objective functions. However, because our supercomputer required about a minute to achieve a steady state condition for each steering parameter tuple, even an optimal search algorithm required several hours to find a solution over the full parameter 5-space. In addition, we found it common for this solution to exhibit unforeseen and undesirable characteristics. Although we could refine the objective function to avoid solutions with these characteristics, at several hours per iteration we found ourselves back in the realm of batch computing. We wanted to find a way to make the inverse steering more interactive.

We therefore performed the following experiment. We pre-computed the steady state field result for 22,638 parameter points uniformly sampled from the 4-space of source frequency (33 samples), source direction angle (7×14 samples), and polarization angle (7 samples). We left out probe insertion distance to make the problem tractable, and because we are most interested in the no-probe condition. This computation took about 25 hours of supercomputer time and resulted in a 6 gigabyte database. We developed a simple interface to search the database, and return the field result for any particular 4-tuple of steering parameters. This gives us a simulated HASP computation, which takes only about 10 milliseconds to return a steady state field result. Using this database, we can run the inverse steering module interactively. This gives us all the usual benefits of interactivity as compared to batch processing, although of course we are limited to the particular quantization of the parameter values and field results which we used to create the database.

An interesting result from this pre-computed database is that we can compute any objective function over the entire parameter space, and visualize the results using our standard, forward steering interface. Color Plate 2 is an example of this. Here we are visualizing the peak EM power intensity (the same objective function specified in Section 2.4 above), as a function of three of the five steering parameters. Notice in Color Plate 2a that, in addition to local “hot spots” representing higher values in the parameter space, there is a global bimodal structure with a “cloud” of higher peak power values occurring in the lower frequency region (around 1 GHz), and a second smaller, less intense cloud of peak power values in the

higher frequency region (around 2 GHz). This bimodal structure is consistent with the analytic observation that the standing wave pattern in a closed cylinder would tend to peak at wavelengths which are submultiples of each other.

4 Conclusions

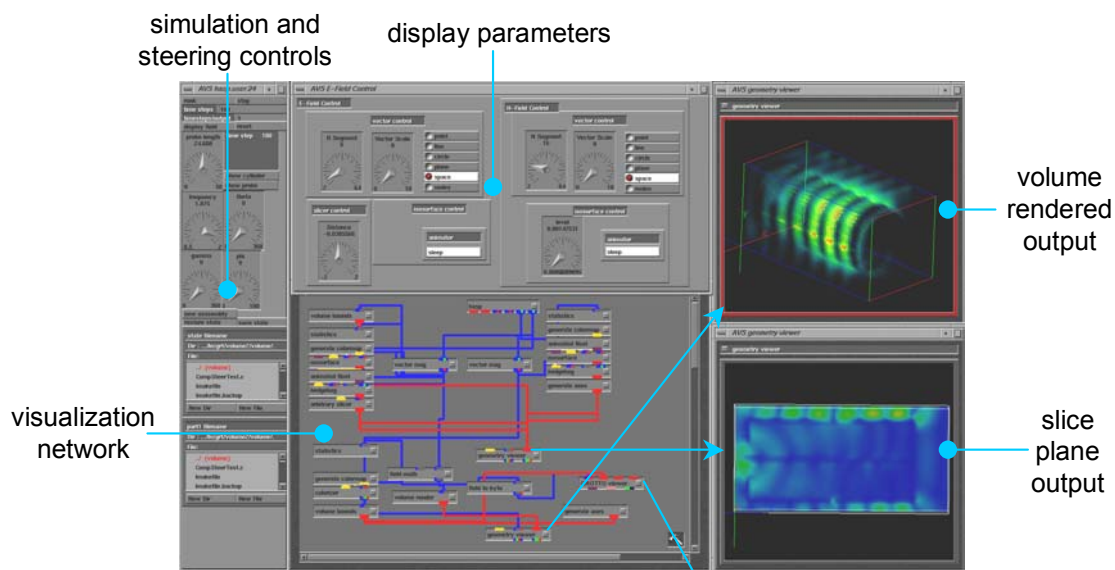
In this paper, we have described a computational steering system architecture, which we developed to support the execution and visualization of microwave interactions with missile bodies. We found the primary advantages of our system were that 1) it is interactive, which allows visualization parameters to be more thoroughly explored, 2) its interface is implemented within the standard visualization package AVS, 3) it allows scientists to view and manipulate their visual output in a large-format CAVE environment, and 4) it supports inverse steering, which enhances the ability of scientists to find simulation parameters which produce desired results.

Acknowledgements

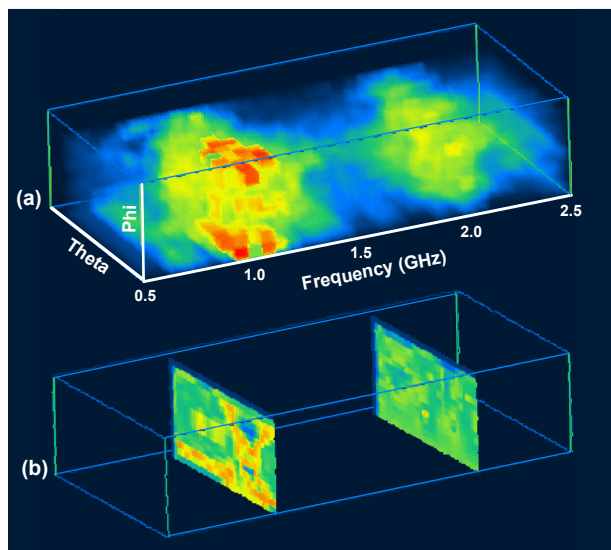
This work was supported by the Naval Research Laboratory and the Office of Naval Research. Access to the SGI Origin2000 was made available under the DoD High Performance Computer Modernization Program. We acknowledge our collaborators Merv Rosen and Mike Haftel from NRL’s Directed Energy Effects Branch, and Larry Rosenblum for leading scientific visualization efforts at NRL. And, we give special thanks to our system administrators Gary Samuels and Toni Miller.

References

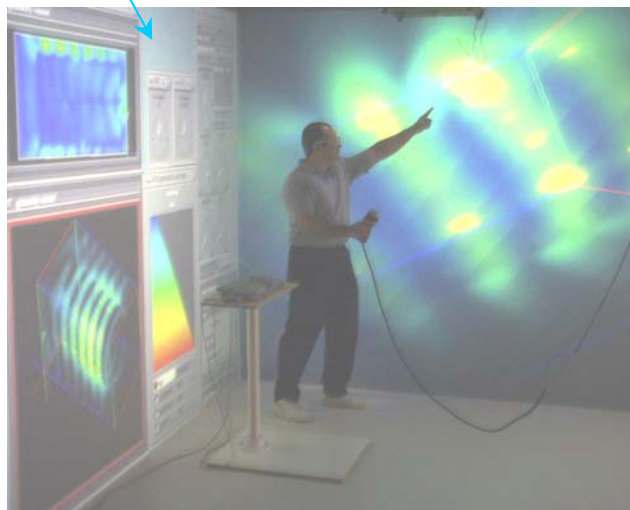
- [1] J. B. Cole, “A Nearly Exact Second-Order Finite-Difference Time-Domain Wave Propagation Algorithm on a Coarse Grid”, *Computers in Physics*, Vol. 8, No. 6, Nov/Dec 1994, pp. 730–734.
- [2] J. B. Cole, “High Accuracy Solution of Maxwell’s Equations Using Nonstandard Finite Differences”, *Computers in Physics*, Vol. 11, No. 3, May/June 1997, pp. 287–292.
- [3] J. B. Cole, “A High Accuracy Realization of the Yee Algorithm Using Non-Standard Finite Differences”, *IEEE Transactions on Microwave Theory and Techniques*, Vol. 45, No. 6, June 1997, pp. 991–996.
- [4] E. Kuo and J. Uhlmann, “Automated Tuning of Visualization Parameters,” Code 5580, US Naval Research Lab, Memo Report (in preparation), 2000.
- [5] E. Kuo, M. Lanzagorta, R. Rosenberg, and S. Julier, “The AVS VR Viewer”, *Proceedings of IEEE Visualization 1999 Late Breaking Hot Topics*, October 1999, San Francisco, California, USA, pp. 89–91.
- [6] L. F. Libelo and T. J. Wieting, “Local Field Distributions for an Excited Finite Cylindrical Cavity Containing a Set of Apertures”, *Proceedings of the International Symposium on Electromagnetic Theory*, International Union of Radio Science, May 1998, Thessaloniki, Greece, pages 657–658.
- [7] S. G. Parker, C. R. Johnson, and D. Beazley, “Computational Steering Software Systems and Strategies”, *IEEE Computational Science & Engineering*, Oct/Dec 1997, pp. 50–59.
- [8] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler, “The VolumePro Real-Time Ray-casting System”, *Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series*, August 1999, Los Angeles, California, USA, pp. 251–260.
- [9] K. L. Shlager and J. B. Schneider, “A Survey of the Finite-Difference Time-Domain Literature”, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Allen Taflov (ed.), Artech House, 1998.
- [10] W. R. Smith, W. L. Anderson, M. I. Haftel, E. Kuo, M. Rosen, and J. K. Uhlmann, “Goal-Oriented Computational Steering”, *Proceedings of the SPIE Conference on Enabling Technology for Simulation Science III*, Orlando, Florida, USA, April 1999.



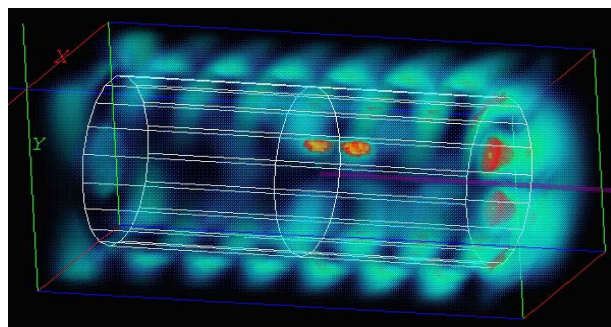
Color Plate 1: Computational steering control panel.



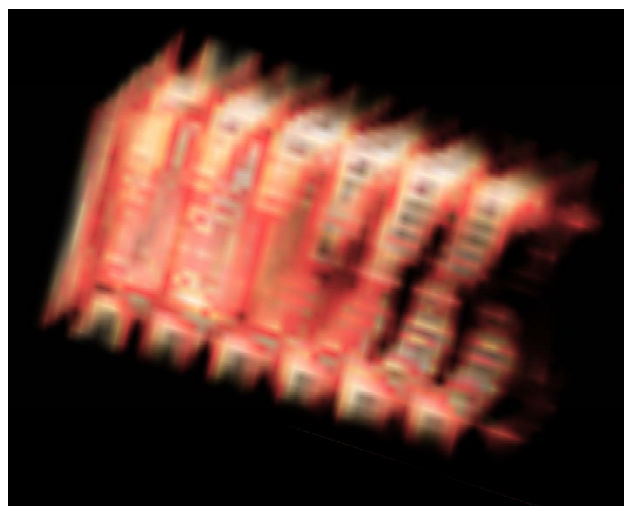
Color Plate 2: (a) Volume rendering of the standard cylinder peak power intensity as a function of three steering parameters: EM source frequency, EM source location (θ , ϕ). (b) Slices through the two resonant regions at 1.0 and 2.0 GHz.



Color Plate 3: Viewing the simulation in the CAVE.



Color Plate 4: Volume rendering of the field together with a wireframe rendering of the standard cylinder.



Color Plate 5: The dataset rendered with the VolumePro board.