Interactive Visualization of High-Order Finite Element Fields Using a GPU-Based Ray Tracer

Blake Nelson, Robert Haimes, and Robert M. Kirby

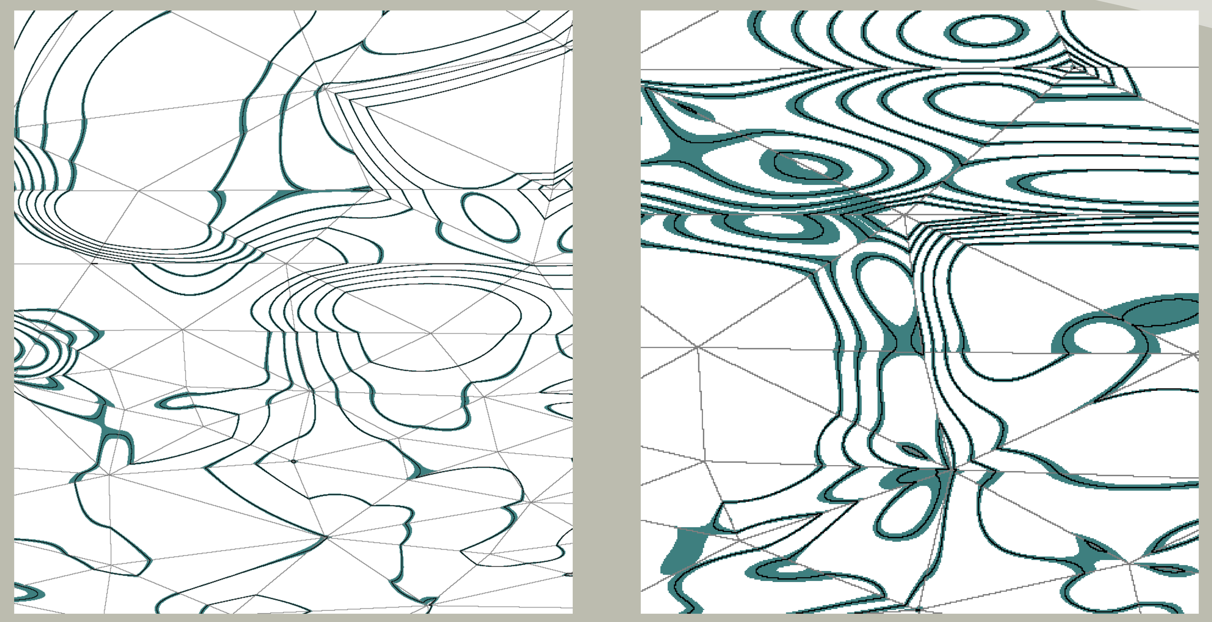
Abstract

We present a GPU-based ray-tracing system for the accurate and interactive visualization of 3D simulations of physical processes created from high-order finite element methods. In order to study physical processes, the ability of highly complex simulation imagery to precisely reflect the data is critical. In practice, scientists use an array of visualization tools to construct scenes that can answer a query of the simulation data. This is effective as long as image quality adequately reflects the underlying simulation data.

Current software libraries used for rendering scientific visualizations, such as OpenGL, work remarkably well for exploratory visualization for most simulation results. This is due to the consistency between the use of first-order representations in the simulation and the linear assumptions inherent in libraries such as OpenGL.

The fidelity of visualizations generated from common techniques are not adequate when simulations are created using higher-order elements. There have been attempts to mitigate this through the use of spatial adaptation and/or texture mapping. These methods do a better job of approximating what the imagery should be but are not exact and tend to be view-dependent.

This work introduces new rendering mechanisms that specifically deal with the kinds of native data generated by simulations based on high-order finite elements.



Finding every contour is not always possible. Pixels that may contain a contour (teal) are compared between a 2nd-order (left) and 6th-order (right) data set.

Interactive and Accurate Rendering:

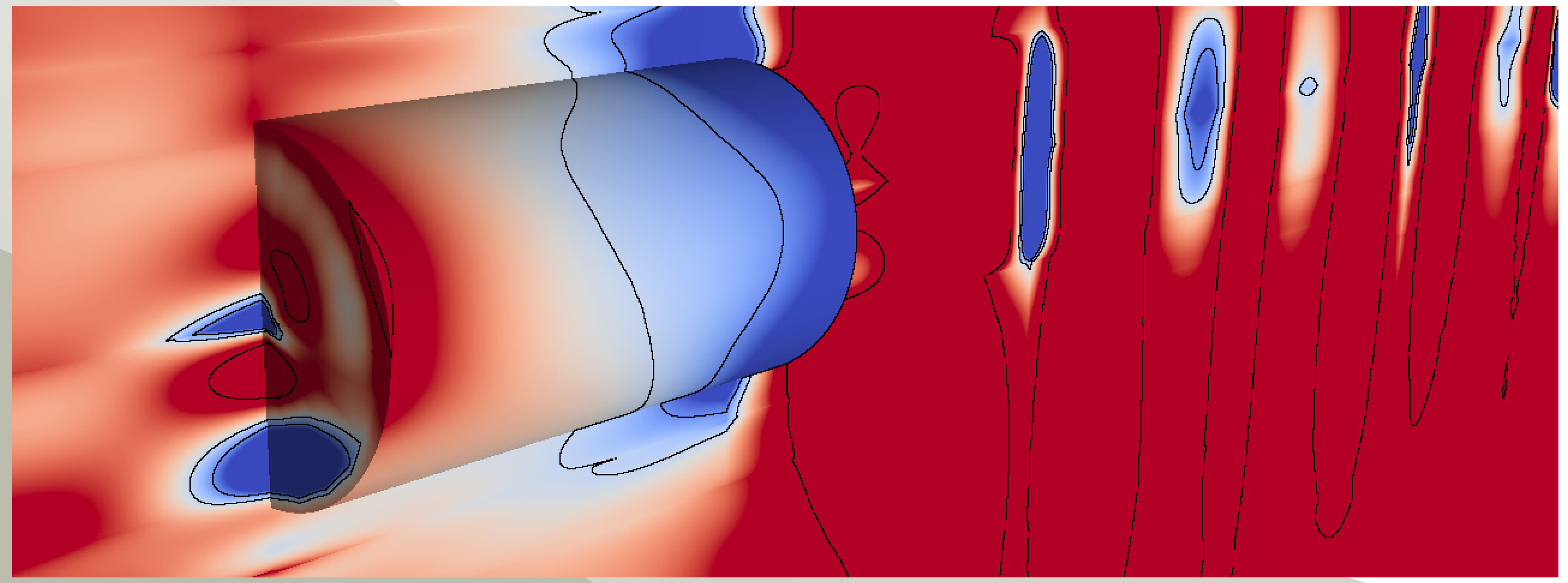
The need for interactivity and a desire to optimize high-order element visualization accuracy prompted the use of a ray-tracer for rendering. Ray-tracers operate on each pixel directly which makes this type of renderer a natural choice in this scenario. Assuming appropriately accurate intersection routines are available, the error associated with locating a point on the cut-surface and calculating the field value is reduced to machine precision.

We have implemented CPU-based high-order ray-tracing systems in the past, and have found them to be too slow to be useful for exploratory visualization. Therefore, we have implemented this system on the GPU, leveraging the highly parallel nature of the GPU to process rays simultaneously. Rather than implement our own ray-tracer, we have decided to use the OptiX framework from NVIDIA, enabling us to provide an implementation that can run interactively on a typical workstation.

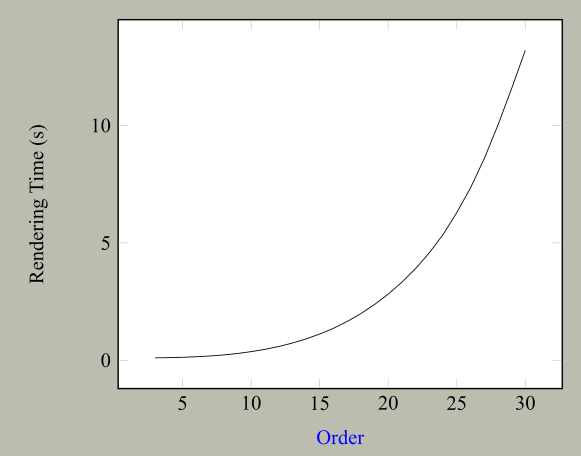
Finite element methods are widely used for solving scientific and engineering problems. They are characterized by the discretization of the problem domain into a collection of elements, followed by the construction of an approximate global solution that is specified in terms of a series of local approximations. Many versions of the finite element method use linear interpolation to represent solution values; other versions, such as those considered in this work, represent solutions using higher-order approximating polynomials.

High-order finite element simulations are often visualized using low-order (typically linear) primitives. Performing visualizations in this way is compelling for several reasons: first, there exists an extensive collection of visualization techniques that expect linear primitives as input, and second, modern graphical processing units (GPUs) can render highly complex scenes composed of linear primitives at interactive speeds. While linear approximations can be rendered efficiently, they do not, in general, faithfully represent the high-order data.

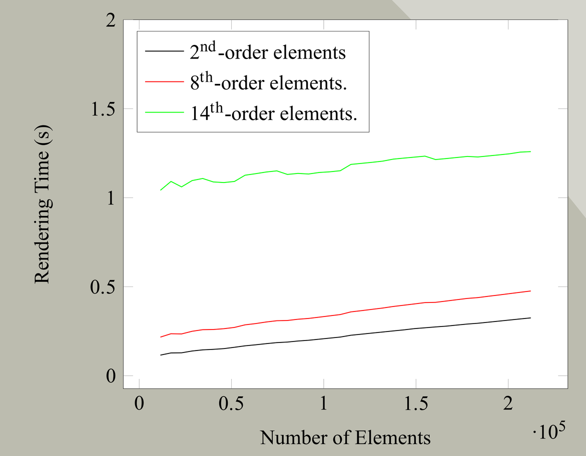
Our system's goal is to generate accurate and interactive images, allowing users to debug the simulation's code, accurately interpret the image, and perform general exploratory visualization. The system uses the high-order data in its native state, without the need for low-order approximation, and uses the knowledge of the structure and mathematical properties of the underlying fields to provide accurate images.



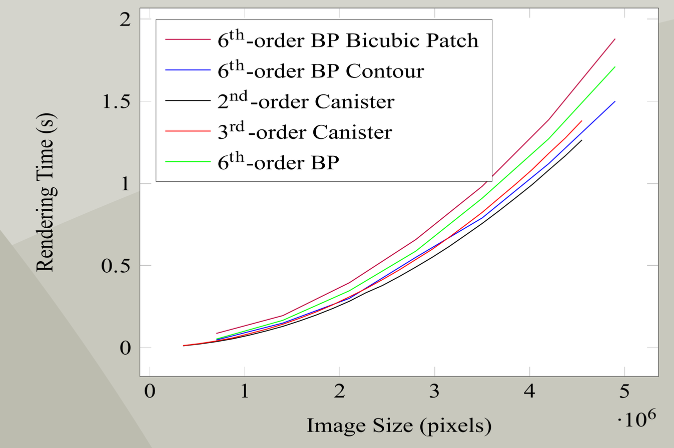
Using the software system developed for natively viewing data from high-order finite element-based simulations we show a view of the pressure field of a rotating canister moving through an incompressible fluid. A color map of the pressure field, along with contours of constant pressure, have been applied to the cylinder and the cut-plane.



(a) Time in seconds to render volumes with increasing polynomial order.

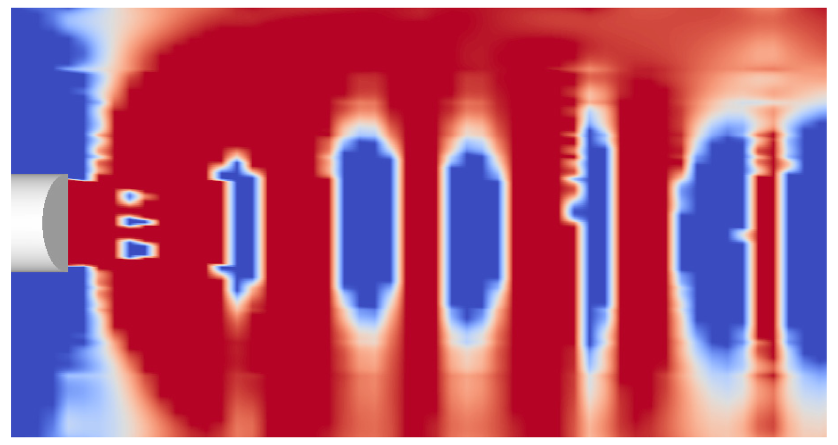


(b) Time in seconds to render volumes with a varying number of elements.

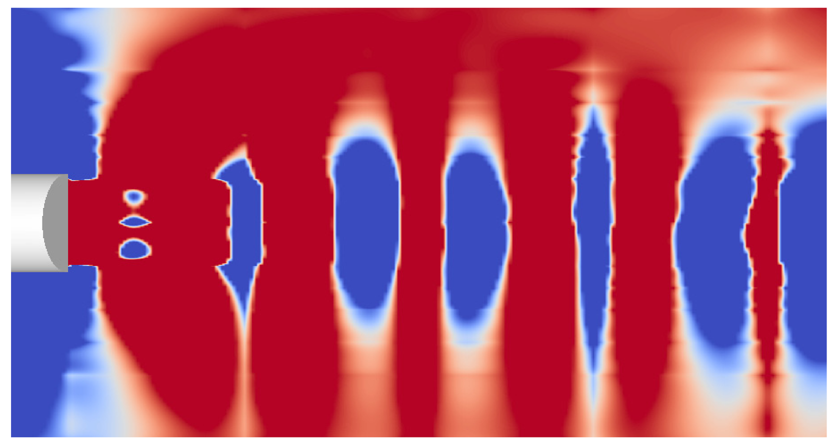


(c) Time in seconds to render high-order data sets for a variety of image sizes and cut-surfaces, measured in pixels.

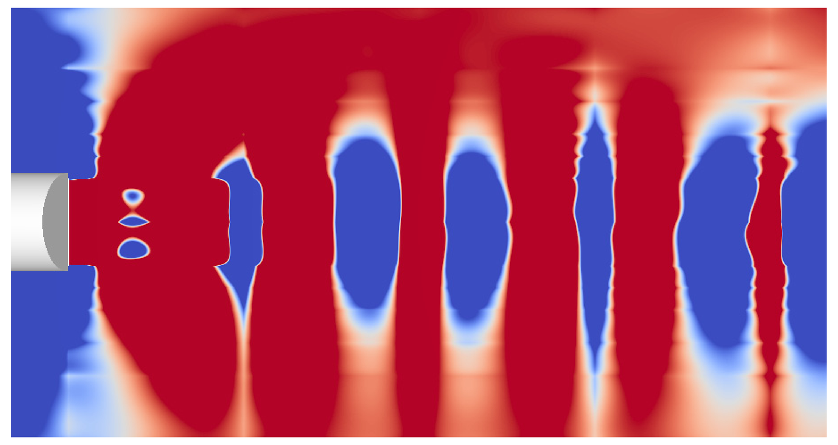
Performance results and memory usage for rendering high-order data using our system. For typical usage scenarios, the parameter that impacts performance the most is image size.



Cut-plane with 902,289 triangles, VTK Rendering Time = 0.08 seconds.



Cut-plane with 8,388,608 triangles, VTK Rendering Time = 2.0 seconds.



Pixel-exact cut-plane color map. Rendering time is 0.015 seconds for a 1800 x 800 image.

Color maps for a cut-plane through the canister data set with a coarse sampling (top), a fine sampling (middle), and pixel-exact using our method (bottom).

Summary:

Our system was motivated by the lack of existing visualization techniques capable of interactively and accurately rendering color-maps and contour lines on cut-surfaces using high-order finite elements. We have described a new system that is capable of rendering these surfaces interactively and accurately while using the high-order data in its native form. We have also shown that the most important factor for determining rendering speed is the size of the final image, indicating that our system can efficiently handle high-order data sets with a large number of elements with a large number of modes. An additional benefit is that these interactive frame rates are achievable on commodity GPUs, meaning simulation scientists can easily perform even the most demanding visualizations at their workstations.

While our system is useful in its current form, additional functionality, such as isosurface generation and volume rendering would move this research system toward serving as a general-purpose high-order visualization system.

Acknowledgments: This work is supported under ARO W911NF-08-1-0517 (Program Manager Dr. Mike Coyle) and Department of Energy (DOE NET DE-EE0004449). Infrastructure support provided through NSF-IIS-0751152

SCI INSTITUTE • EXHIBIT • EXPLORE • EXCITE • EXPERIENCE • EXCHANGE

THE UNIVERSITY OF UTAH

www.sci.utah.edu