NUMERICAL MODELING OF SPACE PLASMA FLOWS// ASTRONUM-2009
Proceedings of the 4th International Conference
ASP Conference Series, Vol. 407, 2010
NAMES OF EDITORS*

Recent Advances in VisIt: AMR Streamlines and Query-driven Visualization

G. H. Weber,[1,2] S. Ahern,[3] E. W. Bethel,[1] S. Borovikov,[4]
H. R. Childs,[1,2] E. Deines,[2] C. Garth,[2] H. Hagen,[5,2] B. Hamann,[2,1]
K. I. Joy,[2,1] D. Martin,[1] J. Meredith,[3] Prabhat,[1] D. Pugmire,[3]
O. Rübel,[1,2,5] B. Van Straalen,[1] and K. Wu[1]

[1]*Computational Research Division, Lawrence Berkeley National Laboratory, One Cyclotron Road, Berkeley, CA 94720, USA*
[2]*Institute for Data Analysis and Visualization, Department of Computer Science, University of California, Davis, One Shields Avenue, Davis, CA 95616, USA*
[3]*Oak Ridge National Laboratory, PO Box 2008, Oak Ridge, TN 37831-6016, USA*
[4]*Center for Space Plasma and Aeronomic Research, The University of Alabama in Huntsville, 320 Sparkman Drive, Huntsville, AL 35899*
[5]*International Research Training Group 1131, Technische Universität Kaiserslautern, Erwin-Schroödinger Straße, D-67653 Kaiserslautern, Germany*

Abstract. Adaptive Mesh Refinement (AMR) is a highly effective method for simulations spanning a large range of spatiotemporal scales such as those encountered in astrophysical simulations. Combining research in novel AMR visualization algorithms and basic infrastructure work, the Department of Energy's (DOEs) Science Discovery through Advanced Computing (SciDAC) Visualization and Analytics Center for Enabling Technologies (VACET) has extended VisIt, an open source visualization tool that can handle AMR data without converting it to alternate representations. This paper focuses on two recent advances in the development of VisIt. First, we have developed streamline computation methods that properly handle multi-domain data sets and utilize effectively multiple processors on parallel machines. Furthermore, we are working on streamline calculation methods that consider an AMR hierarchy and detect transitions from a lower resolution patch into a finer patch and improve interpolation at level boundaries. Second, we focus on visualization of large-scale particle data sets. By integrating the DOE Scientific Data Management (SDM) Center's FastBit indexing technology into VisIt, we are able to reduce particle counts effectively by thresholding and by loading only those particles from disk that satisfy the thresholding criteria. Furthermore, using FastBit it becomes possible to compute parallel coordinate views efficiently, thus facilitating interactive data exploration of massive particle data sets.

1. Introduction

Adaptive Mesh Refinement (AMR) (Berger & Colella 1989) plays an increasingly important role in astrophysical simulations. In general, AMR techniques have proven useful in a wide variety of science and engineering application domains, and visualization researchers have started devoting efforts to the development of effective visualization and data analysis algorithms for AMR data. Furthermore, AMR support is becoming more common in visualization tools aimed at a wide user community in science and engineering. One of these tools is VisIt, which originated at the Lawrence Livermore National Laboratory, and which is now extended and developed by a larger number of outside groups. The Visualization and Analytics Center for Enabling Technologies (VACET) is working on extending VisIt with an emphasis of users in the Department of Energy Scientific Discovery through Advanced Computing (SciDAC) program, and improving VisIt's AMR handling capabilities is an important aspect of this effort.

Recently, our efforts focused on improving VisIt's streamline calculation algorithms (Section 2.). First, we consider multi-block data sets where blocks cover disjoint regions of the domain, i.e., blocks that tile the domain and do not overlap. Originally, VisIt computed streamlines independently within each block, and streamlines terminated when they reached a block boundary. We fixed this shortcoming with an implementation that continues streamlines at block boundaries. A main challenge in this context lies in effective parallelization and communication strategies.

AMR data poses additional challenges to streamline calculation (Deines et al. 2009). The domain does not just consist of multiple domains, but these blocks overlap and are ordered in levels of increasing resolution, where higher resolution representations of a region in the domain replace lower resolution versions. Thus, it becomes necessary to take the AMR hierarchy into account and ensure that interpolation always uses the highest resolution representation available. Furthermore, one needs to take care when interpolating close to level transitions.

Many astrophysical simulations also consider particle data sets consisting of a large number of particles. Recently, we developed a research prototype (see Section 3.) that integrates FastBit (Wu, Otoo, & Shoshani 2006) into VisIt, and the most recent VisIt release includes the functionality of this prototype. It is possible to perform thresholding operations based on the FastBit index and only load those particles from disk that are of interest. Furthermore, it is possible to utilize the index to effectively calculate 2D histograms that can be used for accelerating parallel coordinate-based visualizations. In the following we survey these recent developments and provide pointers to papers that describe them in more detail.

2. Improving VisIt's Streamline Calculation Capabilities

2.1. Background and Significance

For simulations that involve vector fields, integral curves, or streamlines, are one of the most illuminating techniques to obtain insight; they are a cornerstone of visualization and analysis across a great variety of application domains. Drawing on an intuitive interpretation in terms of particle movement, they are an ideal tool to illustrate and describe a wide range of phenomena encountered in the study of application-domain vector fields, such as transport and mixing in fluid flows.

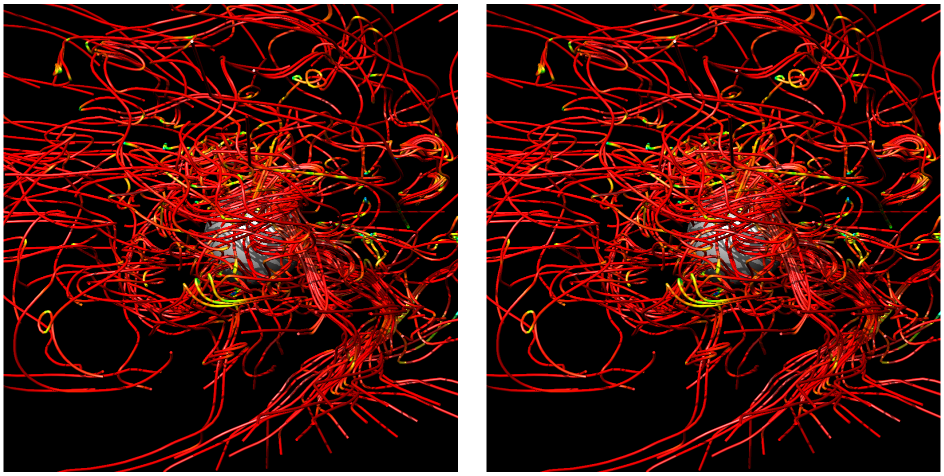

Figure 1. Streamlines extracted from simulated astrophysical data set using a parallel approach..

For a stationary vector field v that does not depend on time, an integral curve is called a *streamline* and is given by the ordinary differential equation

$$\dot{S}(t) \;=\; v(S(t,x)) \qquad \text{and} \qquad S(t_0) \;= x_0. \tag{1}$$

Hence, it describes a parameterized curve that starts at the *seed point* x_0 and is tangent to v over its parameter interval $[t_0, t_1]$ for $t_0 < t_1$.

In the discrete setting we are concerned with in this paper, streamlines are approximated using numerical integration methods to approximate the describing ordinary differential equations. There is an extensive body of work on this topic, and we refer the interested reader to Hairer, Nörsett, & Wanner (1993) for an overview. In our streamline implementation, we use an integration scheme of Runge-Kutta type with adaptive stepsize control as proposed by Prince, & Dormand (1981). The visualization and analysis of vector fields is an active research area, and so-called *integration-based* techniques that derive vector field visualization from integral curves have progressed well beyond the direct depiction of individual streamlines or a small subset of them (McLoughlin et al. 2009).

2.2. Efficient Parallel Streamline Calcuation

Compared to isosurface extraction or direct volume rendering, it is difficult to parallelize streamline generation. While it is possible to extract isosurfaces independently within each block of a multi-block data set, or to take samples along a ray within each block independently for volume rendering, it is not possible to extract the portion of a streamline in data blocks independently. This is due to the fact that the streamline depends on the seed point, and for blocks along the path of a streamline it is not known a-priori where on the boundary a streamline enters the block. As a consequence, it is necessary to compute a streamline piece-by-piece, communicating intermediate results between processors as the streamline passes from blocks owned by a processor to blocks owned by a different processor.

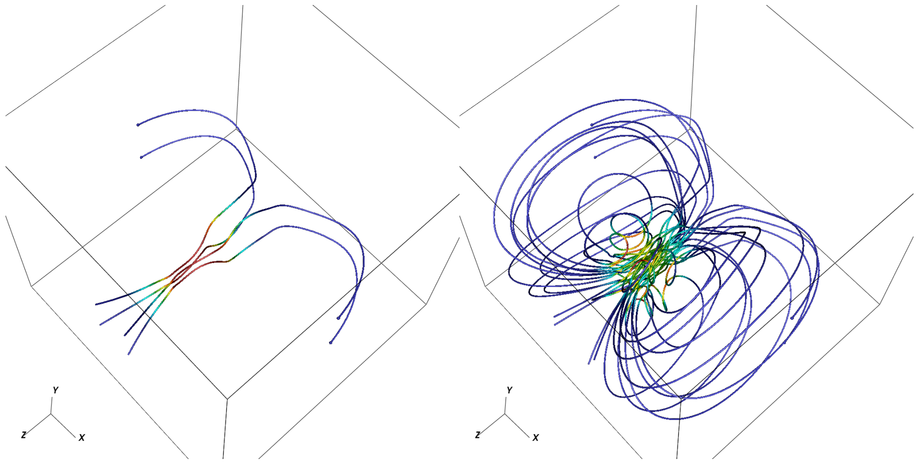

Figure 2. Streamlines extracted from an AMR vortex core merger simulation. When one does not explicitly check whether one enters a refined level, the simulation effectively considers only the root level, leading to a significant loss in streamline accuracy.

Two extreme parallelization strategies are possible. The first method distributes streamlines to processors. Each streamline remains on the same processor, which load data blocks dynamically as a streamline enters them. This strategy avoids communication and keeps a balanced work load among processors (each processor gets assigned roughly the same number of streamlines). However, in most cases it leads to significant data duplication.

The second method distributes data blocks evenly among processors. Each processors "owns" those streamlines that are currently located in one of its data blocks. This strategy avoids data duplication. However, streamlines need to be communicated from processor to processor as they enter and leave data blocks stored on different processors. Furthermore, this strategy may lead to severe load imbalance if streamlines are distributed unevenly among the data blocks belonging to each processor. VisIt 1.12 implements a third method, which is enabled by default, that uses heuristics to transition its behavior between both methods and attempts minimize data duplication while maintaining an even load balance among processors. In Pugmire et al. (2009) we performed an extensive analysis on the trade offs between the various schemes. In many cases, it is safe to use VisIt's new defaults. However, for a more in-depth understanding, we refer the reader to that paper.

2.3. AMR-aware Streamline Calculation

Another improvement to VisIt's streamline calculation lies in proper handling of AMR hierarchies. This work is in a research prototype stage, and we plan to make this functionality available in a later VisIt release. VisIt's current streamline implementation continues streamlines as they transition from one block to another and can trace a streamline through an AMR data set, see Fig. 2 (left). However, this implementation does not detect when a streamline enters a region in a patch that is also available at higher resolution. Instead, a streamline is calculated based on the samples of a patch until it exits its bounding box.

To remedy this problem, we are working on an implementation that detects when streamlines enter a refined region (Deines et al. 2009). This algorithm terminates streamline calculations in a lower resolution patch when the streamline enters a region that is available at a higher resolution and restarts calculations in the refining patch, see Fig. 2 (right). For the purpose of detecting this transition, we are considering two schemes. The first scheme does not use patch layout information explicitly. Instead, it checks during streamline integration whether the currently considered cell is superseded by a finer representation. If it is, it uses the adaptive step size control of the integration scheme to reduce terminate the streamline close to the level boundary and restarts calculation in a second patch. Our second scheme explicitly uses the bounding boxes of levels to identify when a streamline enters a refining level. It turns out that this approach leads to more accurate results. For both schemes, we use ghost cells to interpolate values at level boundaries. Currently, we assume that the simulation produces meaningful ghost data. For a more detailed discussion, we refer the reader to Deines et al. (2009).

3. Visualization of Large-scale Particle Simulations with FastBit

In last year's Astronum proceedings, we gave examples how the combination of VisIt and FastBit facilitates data discovery in large scale particle data sets (Bethel et al. 2009). In this prototype, the H5Part file reader (Adelmann et al. 2007), uses FastBit (Wu, Otoo, & Shoshani 2006) to create an index for the dependent variables in large scale particle simulations. The underlying idea is that when considering billions of particles, not all of these particles are of interest. Instead, one usually focuses only on a subset of these particles, e.g., based on thresholding, and analyzes the behavior of these particles.

The use case in last year's paper (Bethel et al. 2009), for example, focused on Laser-Wakefield particle acceleration. Here, particles of interest included those particles with an momentum in x-direction larger than a given threshold within a particular time step. Subsequently, one only considered these particles. If a particle file in H5Part format contains an index, VisIt will now communicate the settings of the threshold operator to the file reader via VisIt's contract based scheme (Childs et al. 2005). The reader can utilize the FastBit index to load only particles that satisfy the thresholding criteria. (While currently only the H5Part file reader supports FastBit indices, VisIt has the infrastructure in place to communicate thresholds to any custom file readers, so it is possible for users to write their own file format and possibly utilize FastBit. We are considering adding this functionality for particle data in Chombo files.)

Furthermore, VisIt supports now the concept of named selections. If one selects particles in a given time step, e.g., via the threshold operator, it is now possible to create a named selection comprising these particles. Using this named selection it is possible to restrict consideration in other time steps to the same set of particles as well. For example, one can use this functionality to compute the traces of particles that exceed a certain momentum in a given time step.

Finally, VisIt now uses the FastBit index to compute histograms for particles in H5Part files, and uses these histograms to accelerate the context display in parallel coordinate plots (Rübel et al. 2008).

Acknowledgments. This work was supported by the Director, Office of Advanced Scientific Computing Research, Office of Science, 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) nd also the support provided by the German Research Foundation (DFG) through the International Research Training Group 1131. We thank the members of the LBNL Visualization Group, the LBNL ANAG, the LBNL CCSE, and the VisIt development team.

References

Adelmann, A., Gsell, A., Oswald, B., Schietinger, T., Bethel, E. W., Shalf, J. M., Siegerist, C., & Stockinger, K. 2007. Progress on H5Part: a portable high performance parallel data interface for electromagnetics simulations. Particle Accelerator Conference. Albuquerque, NM, USA, June 2007, IEEE Computer Society Press.

Berger, M., & Colella, P. 1989. Local adaptive mesh refinement for shock hydrodynamics. In Journal of Computational Physics, 82, 64.

Bethel, E. W., Rübel, O., Prabhat, Wu, K., Weber, G. H., Pascucci, V., Childs, H., Mascarenhas, A., Meredith, J.i, & Ahern S. 2009. Modern scientific visualization is more than just pretty pictures. In Numerical Modeling of Space Plasma Flows: Astronum-2008 (Astronomical Society of the Pacific Conference Series). 406, 301.

Childs H., Brugger, E., Bonnell, K., Meredith, J., Miller, M., Whitlock, B., & Max, N. 2005. A contract based system for large data visualization. In IEEE Visualization 2005. Minneapolis, MN, USA. 25.

Deines, E., Weber, G. H., Garth, C., Van Straalen, B., Borovikov, S., Martin, D. F., & Joy, K. I. 2009, On the computation of integral curves in adaptive mesh refinement vector fields. Submitted to Dagstuhl Scientific Visualization 2009.

Hairer, E., Nørsett, S. P., & Wanner, G. 1993. Solving Ordinary Differential Equations I, second edition. In Springer Series in Comput. Mathematics. 8.

McLoughlin, T., Laramee R. S., Peikert R., Post, F. H., & Chen, M. 2009. Over two decades of integration-based, geometric flow visualization. In Eurographics STAR. 73.

Prince, P. J., & Dormand, J. R. 1981. High order embedded Runge-Kutta formulae. In Journal of Computational and Applied Mathematics. 7. 1.

Pugmire, D., Childs, H., Garth, C., Ahern, S., & Weber, G. H. 2009, Scalable computation of streamlines on very large datasets. In Supercomputing 2009 (SC09), Portland, OR, USA, November 2009.

Rübel, O., Prabhat, Wu, K., Childs, H., Meredith, J., Geddes, C.G.R., Cormier-Michel, E., Ahern, S., Weber, G. H., Messmer, P., Hagen, H., Hamann B., & Bethel, E.W. 2008. High Performance Multivariate Visual Data Exploration for Extremely Large Data. In Supercomputing 2008 (SC08), Austin, TX, November, 2008.

Wu, K., Otoo, E., & Shoshani, A. 2006. Optimizing bitmap indices with efficient compression. In ACM Transactions on Database Systems, 31, 1.