Chapter 20

The VAPOR Visualization Application

Alan Norton

National Center for Atmospheric Research

John Clyne

National Center for Atmospheric Research

VAPOR is an open source visual data analysis package developed by the National Center for Atmospheric Research, with support from the National Science Foundation. VAPOR provides a highly interactive, platform independent, desktop exploration environment, capable of handling some of the largest numerical simulation outputs, yet requiring only commodity computing resources. The cornerstone of VAPOR's large data handling capability is a wavelet-based progressive access data model that enables the user to trade-off I/O, memory, and computing resource requirements for data fidelity. This chapter provides an overview of VAPOR, placing an emphasis on the VAPOR strategy for handling large data. To illustrate the effectiveness of this technique, the chapter concludes with a brief case study of two data sets: one from Magneto-Hydrodynamics, and another from numerical weather prediction.

20.1 Introduction

VAPOR is a visualization package that was designed from the outset as a means to enable the interactive exploration of massive, time-varying, gridded data sets, primarily those resulting from high-resolution numerical simulations [2, 1]. For numerous computational scientists, the greatest limitation in the visualization of large data is often the I/O. Other computational costs incurred during analysis (e.g. computer graphics rendering, flow integration, calculation of derived quantities) are frequently dwarfed by the time it takes to retrieve the data from a disk.

VAPOR addresses this issue by attempting to minimize the amount of data read from the secondary storage. This is accomplished through the use of both a progressive access data model, designed to support efficient region-of-interest (ROI) access, and a strong reliance on the caching of previous data retrievals in random access memory (RAM). Hence, VAPOR's advanced visualization capabilities, and the ability for the user to make speed/quality trade-offs to maintain interactivity, enable rapid feature identification or identification of spatio-temporal regions of interest. Once identified, these reduced-size ROIs can be quantitatively or qualitatively explored with progressively finer detail.

While suitable for use in numerous computational science domains, VAPOR is primarily designed to address the needs of the earth and space sciences communities, and, in particular, weather, solar, oceanic and climate science and related disciplines. As a result, VAPOR supports features typically not found in other packages (e.g., handling of geo-referenced data), but may lack capabilities found in more general visualization packages (e.g., support for unstructured computational grids).

Finally, VAPOR was designed to run on a commodity desktop (or laptop) computing platform with a shared memory architecture. As discussed below, interacting with sizeable data sets is enabled in VAPOR with only modest computing resources.

20.1.1 Features

This section provides a brief overview of some fundamental capabilities of the VAPOR GUI. These capabilities are aimed at enabling visualization guided-analysis. VAPOR is integrated with NumPy [3] environments to provide more quantitative analysis, and to provide the ability to manipulate variables and derive new quantities.

- The VAPOR GUI was designed to run and enable highly interactive performance using only a desktop or laptop computer equipped with hardware-accelerated graphics. Through the use of two user-specified data reduction parameters (20.2) and the reuse of previously computed and cached results, the system can provide interactive performance on commodity platforms.

- The GUI provides common advanced visualization capabilities associated with state-of-the-art interactive visualization, making use of the GPU for improved performance whenever possible. These capabilities include volume rendering, isosurfaces, contour planes, steady and unsteady flow lines, image-based flow visualization, inclusion of geometric models, transfer function editing, calculation of derived variables, coordinate axis annotation, color bars, etc.

- VAPOR supports an embedded NumPy calculation engine. Derived variables are easily expressed as Python expressions or Python scripts. These scripts are executed only on demand. When a derived variable is requested, Python calculates the derived variable at the desired accuracy and the calculation is constrained to the requested ROI. The new variable is immediately available to all data operators in the GUI. Moreover, the derived quantity is stored in cache, if space is available, to improve access speed for subsequent references.

- VAPOR provides several forms of steady and unsteady flow integration and visualization. The GUI offers numerous data-driven methods for seeding integration locations. For example, random seed locations may be biased towards high-magnitude spatial regions of an arbitrary variable. The flow integrator does not require uniform temporal or spatial sampling, and the GUI allows the selection of limited spatial and temporal extents.

20.1.2 Limitations

Several features that are commonly found in visualization packages, aimed for large data were deliberately not included in the VAPOR design. Many of these limitations or omissions stem from VAPOR's focus on the earth and space science communities, and its emphasis on desktop computing.

- Computational grids are limited to structured, regular tessellations, though the sampling between grid points need not be uniform.

- There is no support for distributed memory architectures. A shared-memory programming model is assumed. Thus, the resolution of a grid is constrained by the available shared-memory address space.

- All data in a VAPOR session are presumed to arise from a single numerical experiment and are constrained to a single sampling grid; variables sampled at different rates, or with different coordinates can not be mixed in a single session.

20.2 Progressive Data Access

VAPOR's method for handling gridded data sets, resulting from high resolution numerical simulations, differs from many other scientific visualization applications. Its approach is motivated by the widening gap between compute and I/O performance. For example, in many analysis operators, especially when time varying data are involved, the single biggest bottleneck is the rate at which data can be retrieved from a disk.

Another technology trend factored into VAPOR's design is the advancement in output display resolution. While very high resolution tiled display devices have been deployed at a number of research facilities, the number of pixels available to a typical researcher working in his or her office has not changed significantly over the years, relative to the rate of progress of other technologies.

With these thoughts in mind, the primary approach to enabling interactive processing of large data, employed by VAPOR, is progressive access. The aim of this data model is to allow the user to make trade-offs between speed and accuracy, providing a form of *focus-plus-context* [4]. Users are able to accelerate the scientific discovery process by formulating hypotheses in an interactive mode, using less resource-intensive approximations of their numerical model outputs. These hypotheses can later be validated on data, to increase resolution accuracy, at the cost of reduced interactivity.

The progressive access data model employed by VAPOR is based on the energy, or information, compaction properties of the discrete wavelet transform. An overview of wavelets, their suitability for compression of scientific data sets, and the mathematical framework for much of the discussion in this section is described in Chapter 8.

20.2.1 VAPOR Data Collection

To take advantage of VAPOR's progressive access capabilities, a data set must first be translated into VAPOR's progressive access data format—VAPOR Data Collection (VDC). A variety of command line tools supporting common file formats and user-callable libraries are provided to facilitate translation. Each variable is transformed, one time step at a time, from the spatial to the wavelet domain. The resulting wavelet coefficients are sorted based on their information content—coefficients with larger magnitudes contain more information—and distributed to a small, finite number of bins. The number of bins, as well as the number of coefficients stored in each bin, is determined by the user, subject to the constraint that the aggregate number of wavelet coefficients in the bins equals the number of coefficients output by the wavelet transform, which in turn equals the number of grid points.

The reconstruction of a variable from wavelet space requires the application of the inverse discrete wavelet transform. If all of the bins are used for reconstruction, no information is lost (up to a floating point round-off). The user may choose, however, to only include a subset of the bins in reconstruction, producing an approximation of the original data. While any combination of bins might be used to reconstruct an approximation of the original variable, the best results are obtained when the subset of bins used in reconstruction are those containing the largest magnitude coefficients. This sub-setting of wavelet coefficients is called the level-of-detail or simply LOD selection.

Each wavelet coefficient bin, for each variable, at each time step, is stored in a separate file. This distribution of bins into different files provides some flexibility in large data management. Files corresponding to bins with less information content might be kept on tertiary storage (e.g. tape), while smaller, but higher information content bins can be kept on secondary storage (e.g. rotating disk). By default, a VDC distributes wavelet coefficients into four LOD bins, that are sized to offer compression rates of 500:1, 100:1, 10:1, and 1:1, when the bins are combined for reconstruction. Thus, using lower-resolution LODs can result in significantly reduced I/O and storage requirements.

To improve both the computational performance of the forward wavelet transform (transformation to wavelet domain) and inverse wavelet transform (transformation to spatial domain) operations themselves, and the performance of data ROI subsetting, data volumes are decomposed into blocks prior to the forward transformation. Blocks are individually and independently transformed, sorted, and stored to files. The block dimensions are a user parameter. The default for the VDC of a 3D variable is 64^3, which empirically strikes a reasonable balance between computational efficiency on cache-based microprocessors, disk transfer rates, and provides enough degrees of freedom for high rates of compression. This latter point warrants further explanation. Because each block is treated independently, and the distribution of wavelet coefficient bins are fixed for all blocks, the maximum compression rate possible is a function of the block size. Larger blocks offer better a compression rate at the expense of an increased computational cost and storage access when a ROI is retrieved.

20.2.2 Multiresolution

The principal benefits of LOD selection are reduced I/O transfer times—which often dominate analysis—as well as the opportunity for a reduction in disk storage requirements if the lower information content wavelet coefficient bins are not stored. However, the reconstructed data contains the same number of grid points as the orginal data, whether generated from an approximating LOD, or losslessly reconstructed using all wavelet coefficients. Hence, the CPU or GPU computational cost of processing a variable, as well as the RAM requirements, are no less than those required for a conventional data representation, and may easily overwhelm the resources of a desktop computing platform regardless of the LOD.

Fortunately, an intrinsic property of the wavelet transform is multiresolution; an N^d signal is decomposed into dyadic hierarchy, where each level in the hierarchy approximates the next, finer level using only the fraction $\frac{1}{2^d}$ of the sample points (see 8.4 for a detailed discussion). This property affords VAPOR users two forms of quality control: the first form is LOD selection, based on wavelet coefficient prioritization; and, second, resolution or *refinement* control, based on the grid sampling rate. LOD selection primarily impacts I/O, while refinement control has implications for both primary storage (RAM) and computation. Coarser grids have a smaller memory footprint, and can substantially reduce memory requirements, as well as the computational and graphical expense of many visual and non-visual analysis operations, whose costs are proportional to the number of grid points.

The next concrete example illustrates the concepts of LOD and multiresolution, and their respective impacts on computing resources. Assume a computing mesh with 1024^3 grid points that are transformed into a VDC, resulting in three levels of detail corresponding to compression rates of 1:1 (no compression), 10:1, and 100:1. The wavelet coefficients for each LOD reside in separate files on a disk named *lod2*, *lod1*, and *lod0*, respectively. Moreover, the number of coefficients stored in each file would be approximately $1024^3 - \frac{1024^3}{10} - \frac{1024^3}{100}$, $\frac{1024^3}{10} - \frac{1024^3}{100}$, and $\frac{1024^3}{100}$, respectively. As described earlier, reconstruction of our data using the coarsest approximation (100:1) requires reading the coefficients from *lod0*, while the second coarsest approximation is reconstructed from the coefficients from both *lod1* and *lod0*, and so on. The choice of LOD will determine how much data are read from a disk.

Due to the multiresolution properties of wavelets, a second form of data reduction can be had by performing an incomplete wavelet reconstruction, halting the inverse transform after the grid has been reconstructed to 512^3, 256^3, or 128^3 grid points, for example. A multiresolution approximation contains fewer grid points than the original data, thus, resulting in reduced memory and compute resources required to store and operate on the approximation. Note that the grid resolution refinement selection is independent of the LOD. However, regardless of the refinement level, higher level LODs will contribute more information, leading to more a more accurate approximation.

20.3 Visualization-Guided Analysis

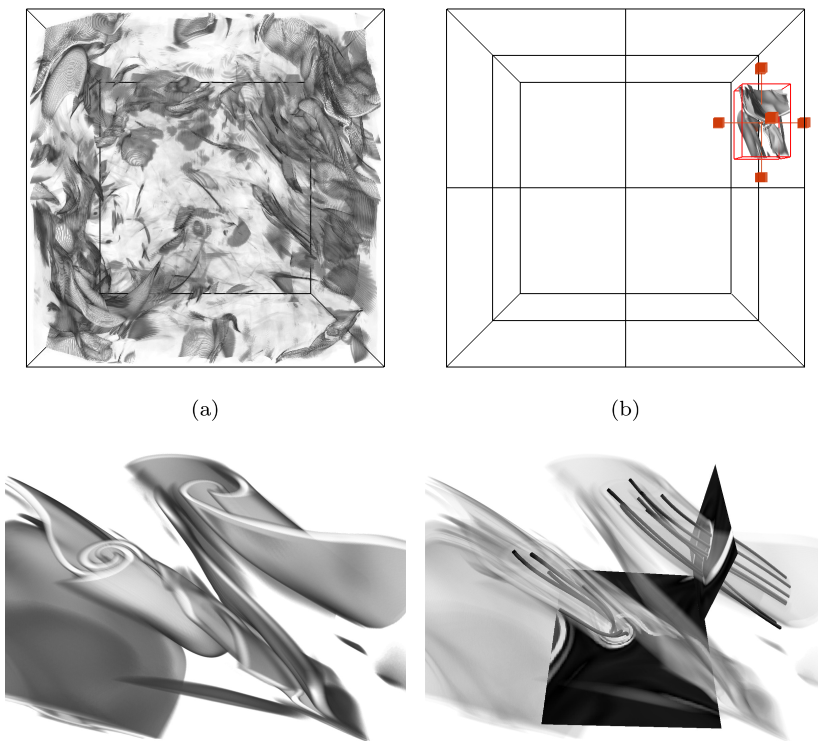
As an illustration of the visualization-guided analysis capabilities of VAPOR, this section describes the workflow used in addressing a research problem in Magneto-Hydrodynamics (MHD) [6]. The data set explored is output from an MHD simulation with a high Reynolds number, computed on a 1536^3 grid, with 16 variables, requiring 216GB per time step. It was expected that, at this resolution, geometric structures, known as current sheets, would form. A current sheet is characterized by the magnitude of the electrical current achieving local maxima along a 2D surface. While current sheets were expected to appear, it was not known exactly what shapes these surfaces would take. There are theoretical reasons to expect that the current sheets could wind into a rolled-up structure, i.e., a current roll; however, this phenomenon was not observed in previous simulations. Direct volume rendering of one scalar variable from the data, at full-resolution, without data reduction, would be possible, but the demands on computing resources would be substantial. The image in Figure 20.1(a) is a full-domain volume rendering of the current field reconstructed from the VDC by reading only $\frac{1}{100}$ of the available wavelet coefficients and performing a partial inverse wavelet transform to produce a grid with $\frac{1}{64}$ the resolution of the original mesh (384^3 grid points). Both the reduction in grid resolution (saving computation and memory costs), and in the wavelet coefficients used to reconstruct the variable from wavelet space (saving I/O costs), were necessary for interactivity.

Through visual inspection of the highly compressed data involving the interactive manipulation of viewpoints, transfer functions, and cutting planes, a small ROI was identified, containing a current roll shown isolated in Figures 20.1(b) and **??**(c). Once isolated, additional visualization and analysis tools can be interactively applied in this smaller region, using both increased grid resolution and LOD quality. In Figure **??**(d), two contour planes are shown, along with the magnetic field lines passing through the center of the roll. The seeding location for these field lines was selected by picking locations on the contour planes in regions of high current magnitude, near the center of the current roll cross-section.

20.4 Progressive Access Examination

To further illustrate the effectiveness of VAPOR's progressive access data model, a qualitative comparison of VAPOR's two data reduction techniques—LOD and refinement level selection—are presented below. Two different data sets are used for comparison: the MHD data discussed in the previous section, and a numerical weather simulation of the severe atlantic storm, Erica [5], computed on a $1300 \times 950 \times 50$ grid. The MHD data set is used to demonstrate the impact of data reduction on direct volume rendering, while the Erica data set is used to demonstrate the impact to pathline integration in an unsteady velocity flow field. Figure 20.2 shows a volume rendering of the original MHD data (a), and data reduced by both LOD and resolution (b). Similarly, Figure 20.4 shows the integration over 20 time steps of five randomly seeded pathlines, computed from the original Erica data (a), and reduced data (b).

Figures 20.3 and 20.5 compare images generated with reduced MHD and

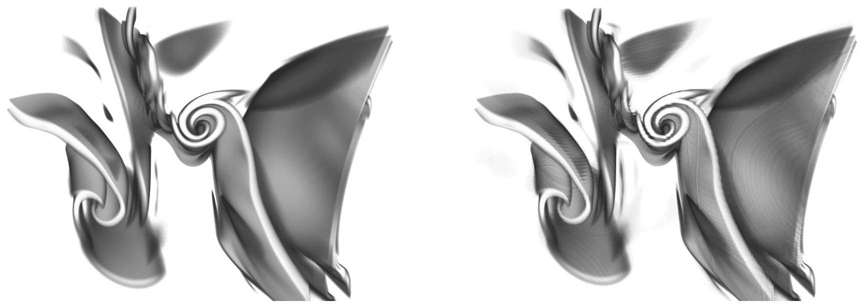


(a) (b) (c) (d)

FIGURE 20.1: Exploration of the current field of a 1536^3 MHD simulation. A volume rendering of current magnitude generated using reduced data (a), an isolated ROI exhibiting a current "roll-up" (b), a close up of the phenomenon (c), and magnetic field lines passing through the center of the roll-up (d). The images in (b)-(d) were generated with the highest refinement level and LOD data.

Erica data, respectively. The top rows show images generated from data at the highest refinement level, but with LODs corresponding to compression rates of 10:1 (a), 100:1 (b), and 500:1 (c). Similarly, the bottom rows show images generated from data at the highest LOD, but with refinement levels corresponding to grids at $\frac{1}{8}$ (d), $\frac{1}{64}$ (e), and $\frac{1}{512}$ (f) of the original resolution.

The reader can subjectively evaluate the quality of these images, keeping in mind the substantial reductions in data involved. While the leftmost images would obviously be preferable for publication purposes, many analysis operations might be suitable for enabling qualitative understanding, using highly compressed data for visualization.



(a) (b)

FIGURE 20.2: Volume rendering of an isolated ROI showing the magnitude of the current field from a 1536^3 simulation. The original, unreduced data are shown (left) along with data reduced by a combination of both LOD and resolution (right). The corresponding reduction factors for LOD and resolution coarsening are 10:1 and 8:1, respectively.

20.4.1 Discussion

In general, the accuracy of the reduced data depends strongly on the data's properties, in particular the degree of coherence between neighboring samples. However, the two examples presented above illustrate some principles generally applicable in the visualization of large data sets. When performing volume rendering (or similar visualizations, such as isosurface rendering, that map data values to graphics primitives), there is little benefit to having grid resolutions whose screen projection sampling rate exceeds, or even approaches, that of the display device itself. Visual quality rarely improves by increasing the refinement level once the projected voxels subtend a screen area smaller than the screen pixel size. In such a case, little information is lost if resolution is reduced, to afford interactivity.

Similarly, when performing an unsteady flow integration (or other operations requiring significant CPU processing on multiple volumes of data) it is again valuable to perform the initial analysis and visualization interactively. Interactivity can be obtained by using both lowered resolution and level of detail, and also by reducing the time sampling rate. By performing visualization and analysis on a small subregion, the quality impact of resolution, time sampling and compression level can be assessed. The desired visualizations can be previewed interactively at a lower accuracy by setting up the appropriate parameters for a subsequent noninteractive session of sufficient accuracy, to precisely illustrate the features of interest.

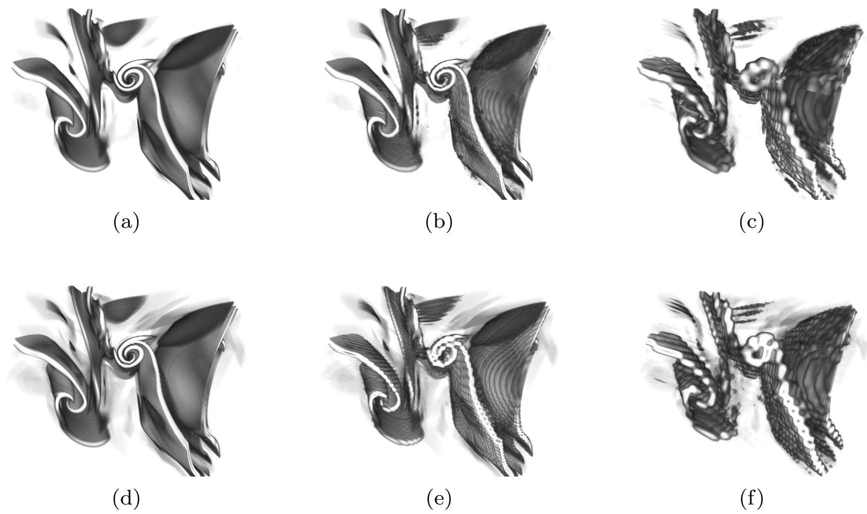
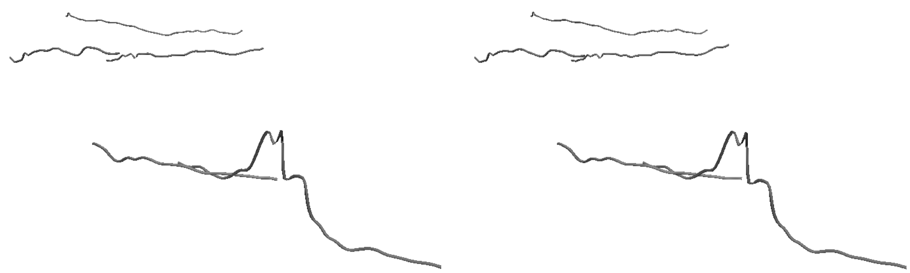


FIGURE 20.3: Direct Volume Rendering of reduced MHD enstrophy data (volume rendering of original data shown in Figure 20.2a). The images in the top row were produced with the native grid resolution, but varying the LODs with reduction factors of 10 : 1 (a), 100 : 1 (b), and 500 : 1 (c). The bottom row used the highest LOD for all images, but varies the grid resolutions with reduction factors of 8:1 (d), 64:1 (e), and 512:1 (f). Reduced LOD primarily benefits I/O performance, while reduced grid resolution primarily benefits memory, computation, and graphics.

20.5 Conclusion

VAPOR provides a desktop environment for the interactive exploration of high-resolution numerical simulation outputs. The interactive exploration of data sets, whose size would otherwise overwhelm desktop computing resources, is enabled by the use of a variety of user-controllable data reduction techniques. These include:

- compression ratio (LOD selection), which is in direct proportion to the I/O performed;
- refinement level (resolution), which is proportional to processing time (including graphics rendering time), and also, it can have a significant impact on memory overhead, and;
- a time sampling rate and region size, both of which are in direct proportion to both processing time and I/O time.



(a) (b)

FIGURE 20.4: Pathlines from the time-varying velocity field of a simulation of the atlantic storm Erica. Pathlines generated from the original, unreduced data are shown (a), along with data reduced by a combination of both LOD and resolution (b). The corresponding reduction factors for LOD and resolution coarsening are 10:1 and 8:1, respectively.

To maintain interactivity, users can explicitly manipulate the parameters associated with all of these data reduction offerings provided by VAPOR. When final, high-quality results are required, the parameters can be set for possibly non-interactive visualization and performed without user supervision, provided the computing platform has sufficient resources to handle the full fidelity data.

In addition to these user-controllable mechanisms, VAPOR also makes extensive use of data caching to avoid the unnecessary recalculation of previous results, and, more significantly, to minimize the reading of data from secondary storage.

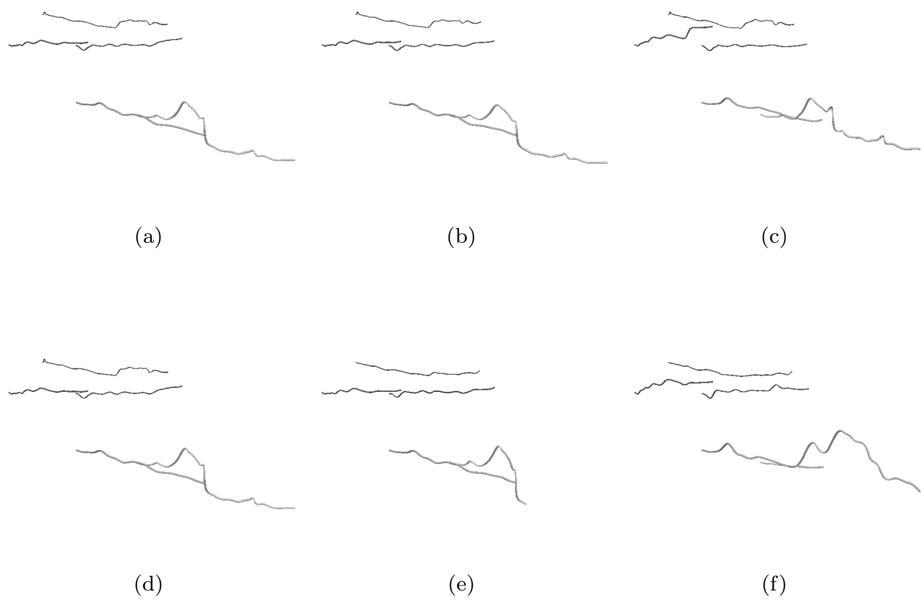


FIGURE 20.5: Pathline integration of five randomly seeded pathlines using reduced storm simulation data. Pathlines generated with original data shown in Figure **??**. The images in the top row were produced with the native grid resolution, but varying the LODs with reduction factors of 10:1 (a), 100:1 (b), and 500:1 (c). The bottom row used the highest LOD for all images, but varies the grid resolutions with reduction factors of 8:1 (d), 64:1 (e), and 512:1 (f).

Bibliography

[1] John Clyne, Pablo Mininni, Alan Norton, and Mark Rast. Interactive desktop analysis of high resolution simulations: application to turbulent plume dynamics and current sheet formation. *New J. Phys*, 9(301), 2007.

[2] John Clyne and Mark Rast. A prototype discovery environment for analyzing and visualizing terascale turbulent fluid flow simulations. In *Proceedings of Visualization and Data Analysis 2005*, January 2005.

[3] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[4] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '95, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[5] Ching-Hwang Liu and Roger M. Wakimoto. Observations of mesoscale circulations within extratropical cyclones over the north atlantic ocean during erica.

[6] P. D. Mininni, A. Pouquet, and D. C. Montgomery. Small scale structures in three-dimensional magnetohydrodynamic turbulence. 2006.