

Recent Advances in VisIt: Parallel Crack-free Isosurface Extraction

Gunther H. Weber^{1,2}, Hank Childs^{1,2}, and Jeremy S. Meredith³

¹*Computational Research Division, Lawrence Berkeley National Laboratory,
One Cyclotron Road, MS50F, Berkeley, CA 94720.*

²*Department of Computer Science, University of California, Davis, One
Shields Avenue, Davis, CA 95616.*

³*Computer Science and Mathematics Division, Oak Ridge National
Laboratory, One Bethel Valley Road, Oak Ridge, TN 37831.*

Abstract. We present a new method for extracting crack-free isosurfaces from adaptive mesh refinement (AMR) data. This method builds on prior work, which utilizes dual grids and stitch cells that fill the resulting gaps. To simplify implementation of stitch cell generation, we propose a case-table-based approach. The most significant benefit of our new technique is that it uses ghost data to handle parallel isosurface extraction in a distributed memory environment efficiently.

1. Introduction

Many physical phenomena, such as star formation, occupy vast regions of space. These areas have extreme variances in spatiotemporal scales, i.e., some areas are fairly homogeneous while other areas may change rapidly. To represent these domains efficiently, simulations must adapt the resolution to local features. The block-structured adaptive mesh refinement (AMR) approach introduced by Berger & Colella (1989) addresses this challenge by creating a hierarchy of axis-aligned rectilinear grids, which are commonly referred to as *boxes*. This representation requires less storage overhead than unstructured grids—it is only necessary to store the layout of all grids with respect to each other since connectivity within each rectilinear grid is implicit—and makes it possible to represent different parts of the domain at varying resolutions. Due to its effectiveness, an increasing number of application domains use this simulation technique.

This hierarchical representation of AMR data makes data analysis particularly challenging. It is necessary to take into account that a finer box may invalidate and replace the value of a given grid cell. A greater challenge is in handling transitions between hierarchy levels such that no discontinuities appear at the boundaries between refinement levels. Like in many hierarchical data representations, cracks in an extracted isosurface arise due to T-junctions between levels. These cracks distract from a visualization’s exploration- or communication-oriented objective and introduce questions of correctness. Furthermore, they affect the accuracy of quantities—such as surface area—derived from an isosurface.

Our goal was to design an algorithm for crack-free isosurface extraction that works in a distributed-memory setting. We view the ability to run analysis on the system

that generates the data as a key requirement. Although some sufficiently high-memory systems do exist to handle large data sets, the transfer time from the distributed-memory supercomputer to this remote shared-memory system is generally prohibitive.

2. Related Work

Isosurface extraction using the marching cubes (MC) method—developed by Lorensen & Cline (1987)—is a common building block for the visualization and data analysis of three-dimensional (3D) scalar fields. A correct MC implementation, such as the work by Nielson (2003), produces watertight, closed isosurfaces when applied to single grids. However, it often exhibits cracks in the resulting isosurface when applied to hierarchical data representations. This problem was first observed and fixed for octree-based multiresolution data representations, e.g., by Shu et al. (1995) and Westermann et al. (1999).

Most AMR simulations produce cell-centered data, and only re-sampling to a vertex-centered representation leads to T-junctions. The approach developed by Weber et al. (2003) changes the grids to the dual grids formed by the cell centers of AMR boxes, thus making it possible to utilize original values from the simulation and to avoid T-junctions. Moran & Ellsworth (2011) generalized stitch cell generation by removing the restriction requiring a distance of at least one cell between the child level and parent level boundary, making crack-free isosurfaces available to a wider range of AMR simulations, such as Enzo astrophysics simulations. Fang et al. (2004) developed a crack-free isosurface extraction approach that preserves the original grids, which is useful for debugging purposes. These approaches all operate in a serial setting and assume that they have access to data values of all boxes in the simulation at the same time. This assumption makes them unsuitable on massively parallel distributed memory machines predominantly used by current simulations.

3. Implementation

Our new approach, described in greater detail in Weber et al. (2012), extends from prior work using dual grids and stitch cells to define continuous interpolation and isosurface extraction simplifying its implementation by using a case table. The main design goals for improving this approach were: (i) enabling data parallel stitch cell generation and isosurface extraction that operate on individual AMR boxes separately, and (ii) maintaining a rectilinear grid representation as long as possible.

Like in prior work, we use dual grids—connecting the cell centered values produced by the simulation—and stitch cells filling the gaps between those dual grids—see Figure 1. Stitch cells connect a box to its neighboring boxes in the same level, or containing/adjacent boxes in the coarser parent level. To construct stitch cells and assign values to their vertices, we require access to adjacent samples in boxes. To facilitate parallelization, we utilize *ghost cells*, a concept originating from simulation. By extending grids with a one-cell wide layer of cells and filling these cells with values from adjacent grids or the next coarser level, all data required to construct stitch cells are available locally, and it becomes possible to process AMR data on a per-box basis. Ghost cell generation is the only step in our approach that requires global communication between distributed memory nodes.

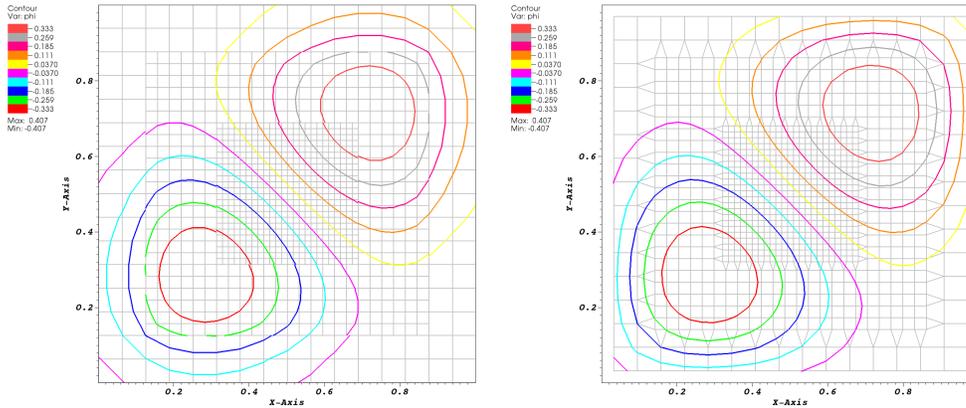


Figure 1. Extraction of crack-free isosurfaces from a 2D AMR data set. (Left) Extracting contours using the standard marching cubes approach after re-sampling cell centered AMR data to grid vertices (grid shown as light gray lines). The resulting contours show clear discontinuities at the boundaries between levels. (Right) Using original simulation data by using the dual grid connecting cell centers and filling resulting gaps between levels with procedurally generated stitch cells yields contours continuous at level boundaries.

Stitch cells correspond to linear VTK cells (hexahedron, pyramid, wedge and tetrahedron) with values specified at the defining vertices. To produce stitch cells, we consider the dual grid for a box formed by connecting the values at all cell centers—including ghost cells. The resulting dual grid contains two types of cells: *inner cells* that contain only vertices within the box and *boundary cells* that connect to at least one value in a ghost cell. To generate stitch cells, we iterate over all boundary cells and map them to the appropriate stitch cells using a single case table. This mapping to linear VTK cells is determined by the coarse/fine level configuration of vertices analogous to the way that the mapping to triangles in MC is determined by the above/below isovalue configuration of vertices. Using a single case table improves the original approach by making it easier to implement and maintain in visualization software.

In simulations, ghost data is normally used at the boundaries to support, e.g., the computation of gradients using only data locally available. For visualization purposes, these cells are typically blanked out, or any generated geometry corresponding to them is removed. It is possible to use a similar concept to handle cells in a box that are invalidated by a finer resolution box. Instead of removing individual cells from a box, by either converting boxes to unstructured meshes or a set of boxes that leaves out refined regions, it is more computationally and storage efficient to keep values in these cells and mark them as “ghost.” This gives rise to a new type of “ghost cell,” i.e., one ghost cell marked as invalid because there is a more accurate data representation available. We handle ghost data at the boundaries as well as ghost data due to invalidation by finer grids, by using an array of flags that specify for each grid cell whether it is a ghost cell or not. If a cell is flagged as ghost, the flag also specifies its type. All visualization algorithms operate on the original rectilinear grids and use the ghost array information to blank out ghost cells. This is the standard implementation in VisIt. In the dual grid, a cell is marked as ghost if any of the cells in the original grid corresponding to its

vertices is labeled as a ghost cell, i.e., if it connects any samples that are flagged as invalid.

4. Results

Figure 2 illustrates applying our method to extract an isosurface corresponding to the shape of the heliopause. The upper row shows the cracks that result from re-sampling to vertex centered data. The lower row shows the same isosurface extracted using our parallel stitch cell generation approach that eliminates discontinuities in the isosurface.

To test the scaling behavior of our algorithm, we examined results from two data sets on the Hopper system at the National Energy Research Scientific Computing Center (NERSC). The first data set is a relatively small 3D BoxLib AMR simulation of a hydrogen flame. It consists of 2,581 boxes in three hierarchy levels containing a total of 48,531,968 grid cells. The simulation data size for all 22 scalar variables is 8.1GB, which is approximately 377MB per scalar variable. Figure 3 highlights the difference between an isosurface extracted from a re-sampled data set and the continuous isosurface provided by our new approach. We stored this data set on the local scratch file system.

The second data set is a larger 3D BoxLib AMR simulation of a methane flame. This data set consists of 7,747 boxes in 3 levels containing a total of 1,525,420,032 grid cells. The data set has a total size of 592GB for 52 scalar variables, which corresponds to a size of approximately 11.4GB per scalar variable. We stored this data set on the global NERSC file system. Both data sets were provided by the Center for Computational Sciences and Engineering (CCSE) at the Lawrence Berkeley National Laboratory (LBNL).

In these experiments, we saw excellent scaling of the stitch cell algorithm, as well as of the following contouring operation, even on the small data set. Unfortunately, we reached the best overall performance with this visualization pipeline at only 120 cores. Although disk I/O is one factor limiting the scaling in this example, the main bottleneck is the ghost data generation phase, as we see its runtime increase with core count. Using a simple model that assumes two fixed transfer rates for intra- and inter-node communication time, we found that the increased amount of ghost data duplicated in the distributed-memory setting explains the increase in ghost communication time. We also examined performance of this same visualization pipeline on the larger methane flame data set. The results showed a significantly better scaling performance for total runtime on this larger data set than on the smaller hydrogen flame data set.

Conclusions

Our approach extracts crack-free isosurfaces in a distributed-memory setting encountered on many DOE high-performance computing platforms. It is deployed as part of the VisIt visualization tool, which makes it easily available on these platforms. AMR is used for a wide range of DOE relevant application domains (fusion, astrophysics, climate, carbon sequestration), and the ability to extract “correct” isosurfaces benefits data analysis in all these areas. Furthermore, our dual grid and stitch-cell based approach can be used to improve other visualization algorithms (e.g., volume rendering) as well.

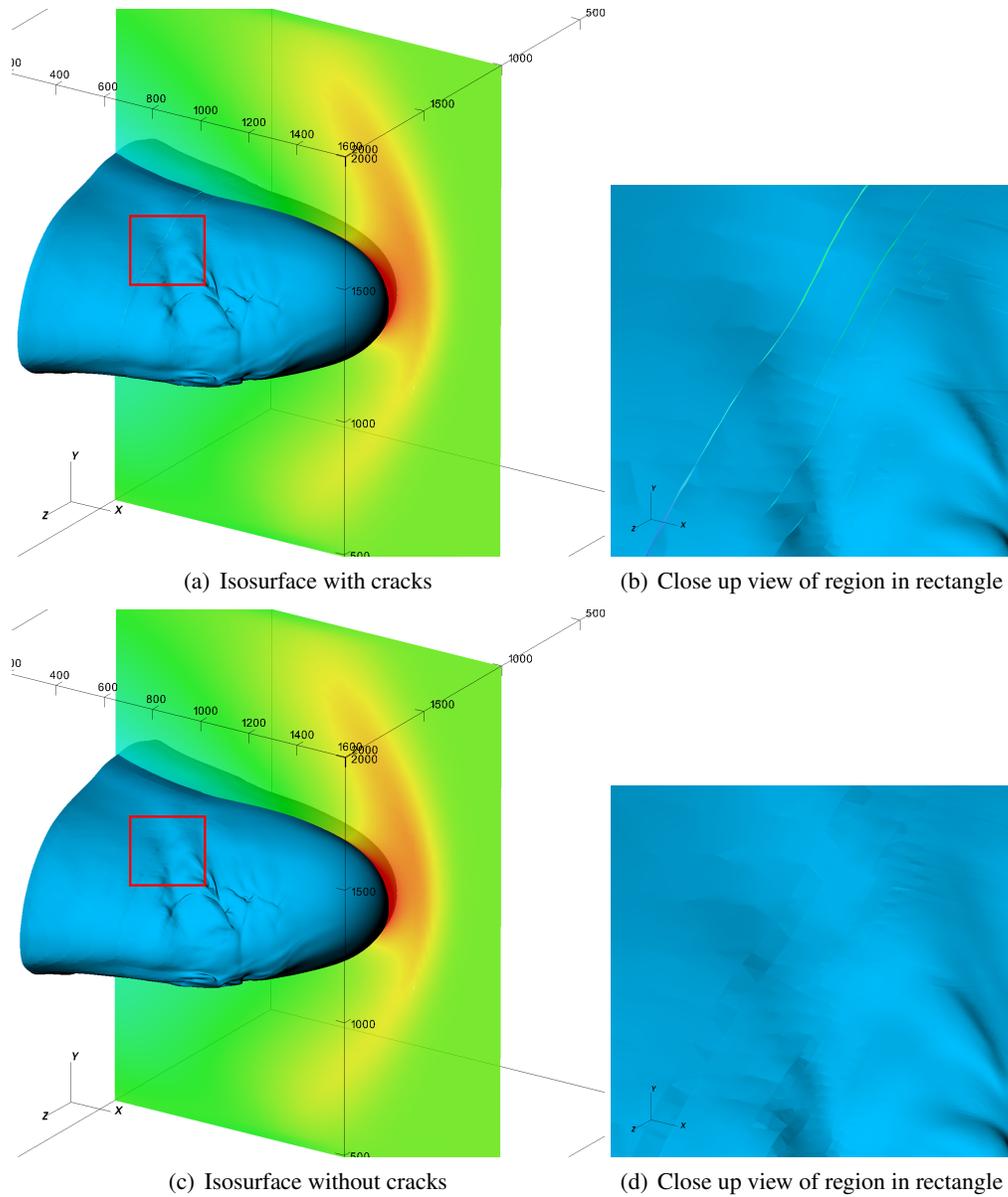


Figure 2. Crack-free isosurface example for interstellar magnetic field strength and shape of the heliopause. The colored slice shows interstellar magnetic field strength, and the isosurface illustrates the shape of the heliopause as the result of the solar wind interaction with the local interstellar medium. (Data courtesy of Sergey Borovikov, UA Huntsville)

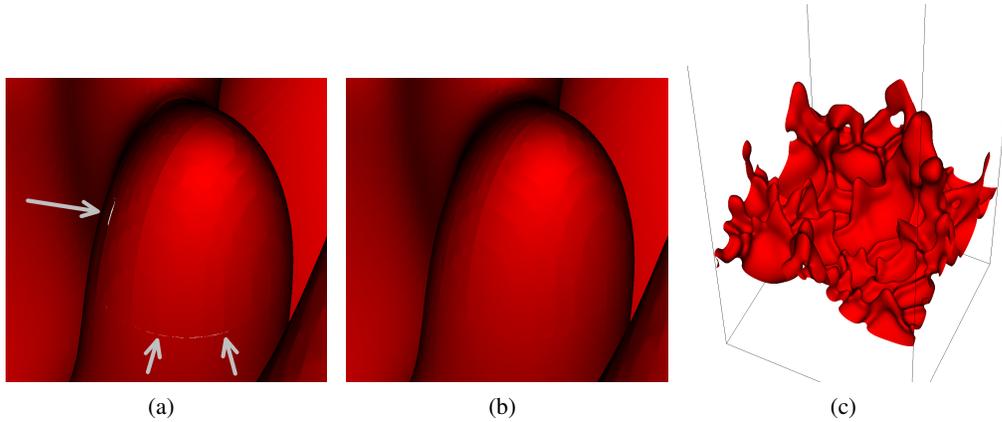


Figure 3. Isosurface (temperature of 1225K) for the hydrogen flame data set. (i) Close-up view of the isosurface extracted via re-sampling to a vertex-centered format. Cracks are easily visible. (ii) Close-up view of the same region extracted using our new method. There are no discontinuities in the isosurface. (iii) View of the entire crack-free isosurface extracted using our new approach. The isosurface consists of approximately 2.2 million triangles.

Finally, this approach could form the basis for deriving topological structures like the contour tree from AMR data.

Acknowledgments. This work was supported by the Director, Office of Science, Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 and DE-AC05-00OR22725 and the use of resources at the National Energy Research Scientific Computing (NERSC) Center. We thank Terry Ligocki from the LBNL Applied Numerical Algorithms Group (ANAG) for generating data sets used to debug and test our method. We also thank the members of the LBNL CCSE for providing the data sets used to benchmark our method. We thank Sergey Borovikov of the University of Alabama at Huntsville for the heliopause AMR data set used to illustrate our method.

Disclaimer

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

References

- Berger, M., & Colella, P. 1989, *Journal of Computational Physics*, 82, 64
- Fang, D. C., Weber, G. H., Childs, H., Brugger, E., Hamann, B., & Joy, K. 2004, in *Proc. 2004 Hawaii International Conference on Computer Sciences*, 216
- Lorensen, W. E., & Cline, H. E. 1987, *Computer Graphics (Proc. ACM SIGGRAPH 87)*, 21, 163
- Moran, P., & Ellsworth, D. 2011, *IEEE Transactions on Visualization and Computer Graphics*, 17, 1862
- Nielson, G. M. 2003, *IEEE Transactions on Visualization and Computer Graphics*, 9, 341
- Shu, R., Zhou, C., & Kankanhalli, M. S. 1995, *The Visual Computer*, 11, 202
- Weber, G. H., Childs, H., & Meredith, J. S. 2012, in *Proceedings of IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, 31. LBNL-5799E
- Weber, G. H., Kreylos, O., Ligocki, T. J., Shalf, J. M., Hagen, H., Hamann, B., & Joy, K. I. 2003, in *Hierarchical and Geometrical Methods in Scientific Visualization (Springer Verlag)*, 19–40
- Westermann, R., Kobbelt, L., & Ertl, T. 1999, *The Visual Computer*, 15, 100