

Visualization and Analysis of Numerical Simulation Data

Physics Department, University of California Santa Barbara, and
Thermonuclear Applications (X-2) Division, Los Alamos National Laboratory
C.A.R.E. Program: July 7 - September 23, 2003

Ronald M. Caplan

Under Direction of Dr. Kimberly New, Dr. Crystal Martin, and Dr. Akimi Fujita

Introduction

When trying to understand large dynamical systems, it is almost always necessary to run numerical computer simulations. This is due to the highly non-linear evolution of such systems, which makes it practically impossible to calculate analytically. A result of having to use simulations is an often enormous amount of data output. Visualization packages must be written to take such data and display it in a useful way for analysis.

At Los Alamos National Laboratory (LANL), a program named RAGE has been developed to simulate large hydro-dynamical systems. One main feature of RAGE is its use of real-time Adaptive Mesh Refinement (AMR), which continuously alters the grid of data in order to have higher resolution gridding at areas of importance in the simulation, and lower resolution gridding at large unchanging areas. This dramatically decreases the space needed for the data, and the size of outputs. In order to visualize the output data from RAGE, routines were written to output the data in a format that can be read by Ensight, a very professional, very expensive visualization software package.

Although RAGE is used heavily at LANL, researchers around the country also use it. These researchers, especially the academic ones, do not have the money for software such as Ensight to do their visualization. Thus, a problem arises as to find an alternate software solution for visualizing RAGE output.

As a participant in the Cooperative Research, Assistance, and Education (C.A.R.E) program, between LANL and the University of California Santa Barbara, my task was to explore such alternatives.

Visualization Goals

Some features desired in a numerical simulation visualization are the ability to view and save animation movies of the evolution of the system, be able to zoom and rotate the image, and to have tools that make it easy to see the different values of the data and where those values are located.

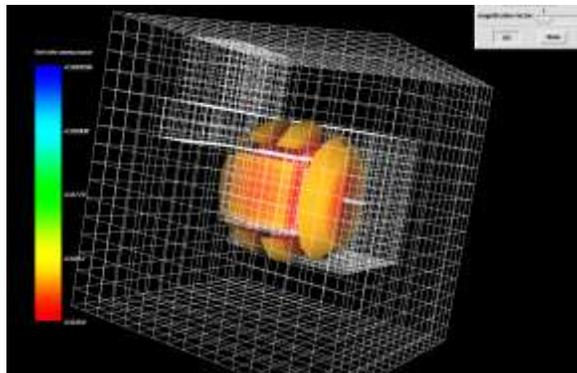
For RAGE output, a major aspect of visualization desired is to be able to view the data cube in its original AMR format, without having to convert it into a uniform cube of a specified resolution. This would be desirable so that rendering times and memory space can be cut to a minimum.

After first searching the Internet for visualization software to fulfill these requirements, I found a promising possible solution. A team at Lawrence Berkeley Laboratory has written a program called ChomboVis, capable of viewing AMR directly.

ChomboVis

ChomboVis was developed to visualize data outputted from Chombo, a numerical simulation software. It is extremely fast, and easy to use. Another attractive feature is that it is completely open-source and free for use by anyone. One difficulty is that it requires the data to be in a Chombo-specific hdf5 format, which is not compatible with anything else. ChomboVis tries to get around this problem by providing a script called `ascii2hdf5` which takes data in a specifically formatted ASCII text file and converts it into a hdf5 file readable by ChomboVis, including AMR formatting. However, getting any other data into this format takes some doing.

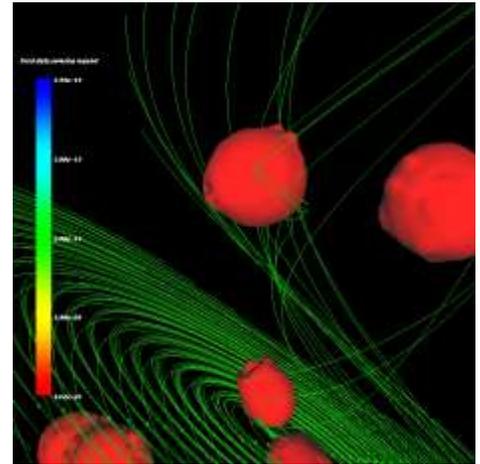
To test this out, I wrote an IDL program called `hdf42ascii.pro`, which takes a uniform hdf4 data cube, and outputs an ASCII file that can be used with `ascii2hdf5`. With this, I was able to view



ChomboVis: Display of Block AMR

some sample data given to me in hdf4 format (taken from a raster image from RAGE) in ChomboVis. The procedure is very smooth and produced good images including velocity streamlines. The problem is that the procedure takes a very long time and a very large amount of disk space.

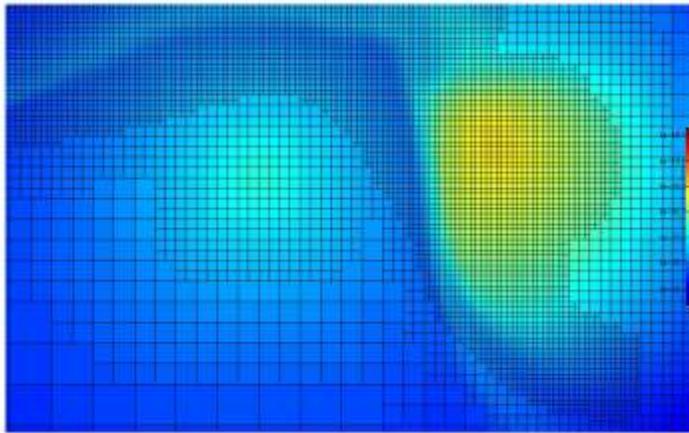
Since ChomboVis displays AMR so well, this time could be worth it. However, the way RAGE does AMR is with continuous adaptive meshing, while hdf5 uses block AMR (which has specified sections of different resolutions). This isn't much of a problem because RAGE's AMR can be ported to block AMR, simply by making a block size of 1. In order to do this RAGE itself would have to output the ASCII file. The RAGE programmers did not have the desire to do this, because they already had someone working on visualization software specifically designed for RAGE, called X/M/WSHOW. This fact ruled out using other programs similar to ChomboVis such as LCAVision, because all had to have some kind of hdf5 as an input.



ChomboVis: Display of converted hdf4 data w/ velocity streamlines

X/M/WSHOW

This software (MShow for Mac, Xshow for unix, and Wshow for windows) is very useful in looking at RAGE data. It can read RAGE's dump files directly without having to convert anything. When looking at 2D slices through the data, the original AMR gridding is displayed.



Xshow: 2D slice of data with gridlines shown to display RAGE's AMR

As for 3D, the program lets the user create a 3d image by taking sets of 2D slices and rendering a model from them. This creates a uniform resolution volume, losing the AMR advantages.

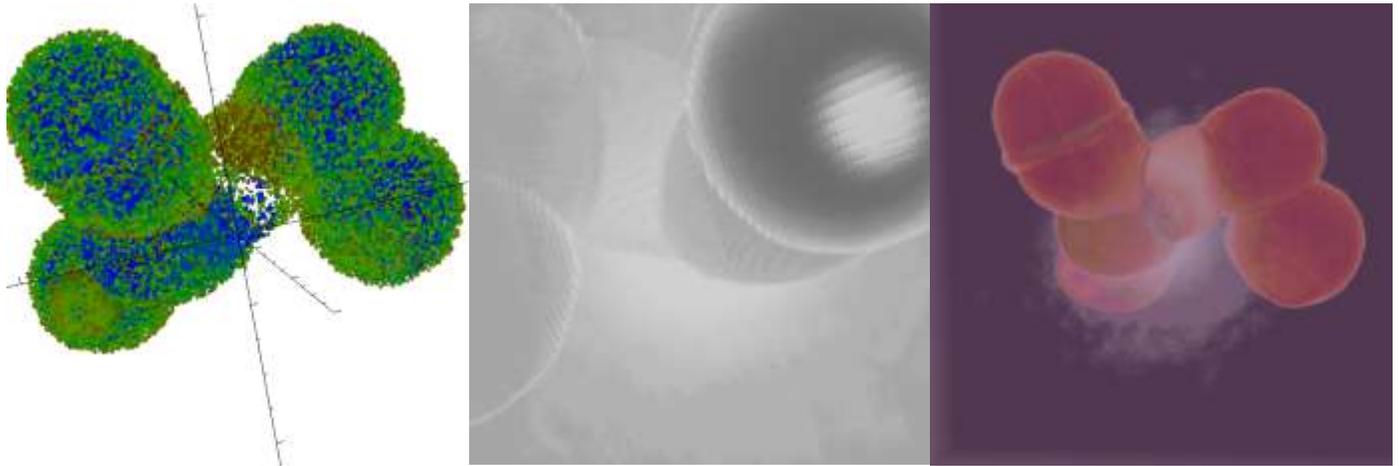
Some other difficulties with X/M/Wshow is that it is very difficult to use unless one takes the time to learn the ins and outs of the interface. Not only that, but certain features such as multiple iso level viewing automatically are not implemented yet. The software does have movies, but can only save them to standard formats on the Mac using Mshow. Xshow, instead creates a series of JPG images to save a movie. This will undoubtedly be updated some day, but for now, I wrote an IDL program called **xmmaker.pro**, which takes a series of images (in either JPG, BMP, or TIF) and creates a MPG movie out of them.

Since X/M/WShow does not solve the 3D AMR visualization problem (yet), my focus shifted to finding a way to easily view and work with 3D uniform raster images. To do this, I wrote **rage3d.pro**.

Rage3D

Although there are many programs written that are free source for viewing 3D data cubes, each had their own problems and difficulties, some with certain features but not others, installation and compatibility problems, etc. To write my own program I decided to use IDL because it is a cross-platform and fairly inexpensive language. Rage3D allows users to view hdf4 uniform-gridded data in 3D and has the following features:

- An easy to use graphical interface, including a trackball-style point-and-drag rotation of the 3D image.
- Can view iso-surfaces specified by the user, or over a specified range.
- Can do volumetric rendering using transparency with user-specified opacity scale (for full data viewing).
- Provides easy switching from different data types, and different time steps.
- Easily saves images to BMP and animations (both time-step and rotations) into MPG.
- Has an automatic rename-extension feature to rename all data file names to have an .hdf extension. (Required for program to read the files, works for Unix only)
- Log and Linear scale switching, manual and automatic data range settings, axis display options, render quality settings, color palette switching, and ability to zoom into the image.



Rage3D: Iso-Surfaces, Zoomed Greyscale Volumetric, and Haze Palette Volumetric Render.

Data Analysis in 2D

Along with 3D calculations, there are also 2D simulations that need to be viewed and analyzed. Specifically, time-step movies need to be produced and basic raytracing analysis done. This was to be done with some star-burst galaxy blow-out calculations.

The analysis that is desired is to be able to compare the simulation to actual observations. One way to do this is to calculate the column density and velocities along a line of sight for specific temperatures for comparisons with observed emission and absorption lines in the spectrum.

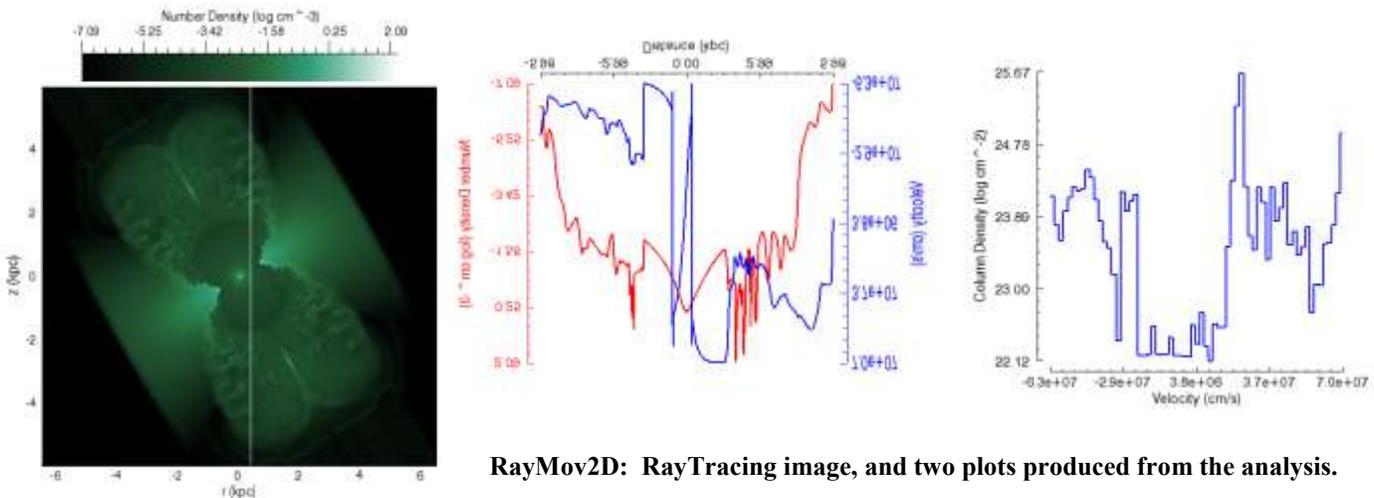
For this task I again used IDL, and wrote RayMov2D.

RayMov2D

Like Rage3D, RayMov2D also does easy movie animations, images, zoom, data range, etc. with an easy to use GUI. The main difference is that RayMov2D also has some analysis tools.

For Raytracing, the image is rotated to a certain degree pertaining to the line of sight angle desired. The values for density, and velocity are taken from that line of sight. For the specific data that I tested with, the data was axis-symmetric so the program takes x/y-velocity values along with z-velocity values and calculates the line-of-sight velocity. With these values, column density is calculated with velocity ranges specified by the user, and adding up the density values within those ranges. After this is calculated, three plots are available for viewing: column density vs velocity, number density vs velocity, and number density and velocity vs distance.

Another useful feature are two analysis animations. One sweeps through the image with a specific angle line of sight, and creates an animation of how the plot being viewed changes with line position. The other animation is at a fixed line position, but the angle is changed from 0 to 360 degrees.



RayMov2D: RayTracing image, and two plots produced from the analysis.

Current Status

In the programs I have written, I have tried to make them as general as possible, and have tried to make the code easily readable, thus editable. Rage3D is very general, while RayMov2D is less so, (it is set up for specific units, axis-symmetric data, etc). Each of these two codes has manuals with contact information so they can be easy to use. RayMov2D does not use temperature data to select density values in the column density calculation, but probably should for data comparison purposes.

W/M/Xshow is continuing to be developed in San Diego and should be improving to the point where it can be used for everything that my programs do not do (mainly AMR 3D visualization).

Contact Information and Links

People

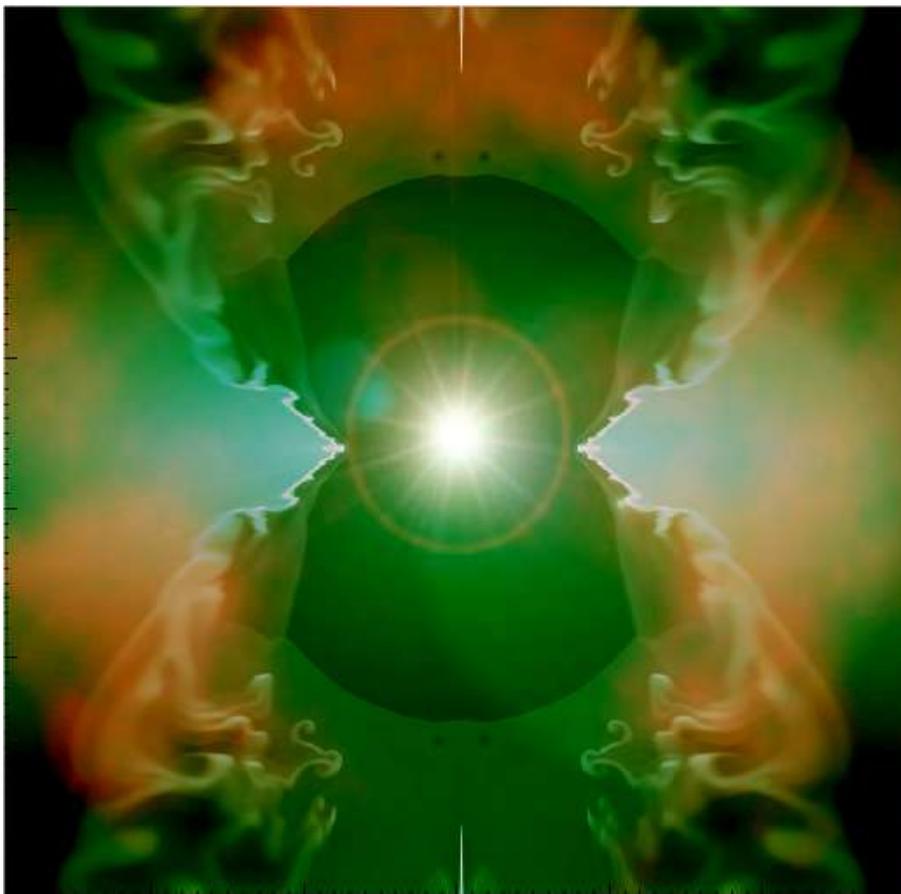
Ronald M. Caplan sumseq@hotmail.com
Kimberly C. B. New knew@lanl.gov
Crystal Martin cmartin@physics.ucsb.edu
Akimi Fujita fujita@physics.ucsb.edu

Software

ChomboVis <http://seesar.lbl.gov/anag/chombo/chombovis.html>
IDL <http://www.rsinc.com/>

Places

LANL X-2 Division <http://www-xdiv.lanl.gov/XTA/>
UCSB Physics <http://www.physics.ucsb.edu>



Simulation data meets Paint Shop Pro ©