

Projecting 3-D Simulations Into Pseudo Observations

Alex Antunes, John Wallin

*School of Computational Sciences, George Mason University, Fairfax,
VA 22030*

Abstract. We present methods for converting particle method three-dimensional simulations into observationally verifiable projected column densities, channel maps, fluxes, and velocity contours. Such projections are suitable for direct comparison with radio data (such as produced by AIPS), X-ray observations (e.g., ximage), optical, and IR. Whereas modeling usually involves N-body, mesh, SPH, or LPR calculations upon an idealized 3-D space, our observational data is always limited to a single line of sight projection, observing only one ‘plane’ of the object, emission from which may or may not include extinction. For models to have any validity, we must be able to generate pseudo-observational data from the model, to compare with actual observations. This connects our modeling with the real universe we see; herein we discuss the methods for creating such projections.

1. Introduction

Pseudo observations provide a connection between data and modeling. In N-body simulations, we model bulk gas, heated gas, and stars as individual particles in three-dimensional space. We can run time forward or backward from ‘now’ to study evolution.

Yet from Earth, we see flat images. Therefore we must convert our three-dimensional simulations to two-dimensional surface densities in order to compare directly with observational results.

This is about pinning down the crime scene—getting our simulations to match observations. From that, we can then make theories. Is a given feature dynamically created or evolutionary (e.g., did existing stuff move to that position, or did new stuff get formed there due to the collision)? Is it stable, will it be long-lived?

The goal here is to replicate nature. We make assumptions based on what we see (‘most gas is near the radio-visible gas’), we run a model using that assumption, then we create results thinking “I’ll plot all the gas as if it were radio-visible, because gas congregates, right?” In some cases, modeling can point out conceptual flaws (e.g., a dark matter halo is required for stability; sims verify this), in other cases our model will simply regurgitate our expectations. So the feedback between data and mode is complex, and validation via pseudo observations is therefore crucial.

Nature \Rightarrow
 View from 1 location + limited wavelength range + instrument limitations \Rightarrow
 Theoretical model \Rightarrow Guess at evolutionary history \Rightarrow
 Input to computer model \Rightarrow Series of individual time frames \Rightarrow
 3-D+time movie

2. Pseudo-Observations

2.1. Data Space

In the real world, radio observations show cool bulk gas (abundances and velocities with no extinction) and basic kinematics (inc. rotation curve). Optical and near-IR observations show hotter gas, extinction (dust). Broadband optical shows hot and massive short-lived O/B stars (recent star formation) and stellar populations and evolution. X-ray data show very hot regions—typically due to singularly active objects that, at our N-body resolution of $10^5 - 10^7$ stars/computational point, are not sim-resolvable. But they do help show very hot gas regions as well.

In N-body Simulation Space, we have collisionless point-like particles, which we call ‘stars.’ Each represents a lot of stars. We also have gas blobs that represent local gas, with a somewhat unified temperature and group velocity. Each blob interacts with nearby blobs in a smooth way (hence the term Smoothed Particle Hydrodynamics). Some of the gas forms into stars, changing the overall particle total metallicity (and removing some of the gas), and also injecting energy (as supernovae) into the region.

This high level of abstraction has to be mapped to the real world observations in order to obtain meaningful results. Our short list of criteria includes being able to create a 2-D projection of the data set, creating surface brightnesses or density contours (hopefully with extinction), and allowing per-component tagging to create false color images similar to the processing done to real data.

Our code (**mass99**) is an N-body code with SPH, LPR, and various evolution and solid codes. Other packages (TreeSPH and its kin, NEMO’s N-body, etc.) do similar work. The basic approach is to simulate a galaxy with a lesser number of collisionless particles and some sort of gas surrogate. The higher the number of particles, N, the more accurate the simulation. The more (and more accurate) the physics included, the better the simulation. The better the simulation, the longer it takes to run on a supercomputer.

2.2. Analysis Space

To create a pseudo-observation, we must:

1. Run a 3-D simulation using an N-body tree, a mesh, SPH blobules, LPR gridless grid, or whatever method is in our code.
2. Choose a data frame (time slice) that matches ‘now.’
3. Project the 3-D data along the Earth-directed line of sight.
4. Sum up the surface brightness or fluxes or surface density or volumetric average or contour value for our given quantity. Some sums are extinguished across the volume, others are surface summations only, others are accumulated totals.

5. Optionally, degrade by the appropriate interstellar medium, telescope response, or atmospheric observation. Or, if our observation data analysis already removed these effects, skip this.
6. Repeat for all wavelengths that we have data. This means make column densities, velocity contours, and channel maps for radio data, fluxes and direct image projections for *HST* and *CHANDRA* work (optical and X-ray), et cetera.
7. Display our assumptions. For example, our sim false color images assume star formation produces blue/white stars, and older stars are redder, and all gas is radio-visible or distributed where the radio-visible gas is.

Reduction items must therefore include projected particle/gas data as well as binned and/or averaged contours and/or surface densities, z -buffer sorted and with any extinction applied. Radial binning and radial profiles, phase plots, and velocity slices are also needed. Any data should be ‘sliceable’ as observational techniques typically ‘catch’ only a specific type or temperature of particle. All of this is best handled at the data digestion stage, before creating the pseudo-observation. First select the possible data, then filter and massage that to get the associated physically related output.

3. Keyframing Techniques

The issue of how to match a simulation to an observation runs into an interesting limitation, as finding the slice of time that corresponds to ‘now’ is a necessary step in creating the pseudo-observation. High performance computing is almost becoming i/o limited, creating more data than we can effectively manage. As data sets go from 10^8 to 10^9 in size, individual data frames of several gigabytes are common. So we deliberately degrade data, analyze during runtime to save on storage, and use Disney-esque keyframing to pre-focus on times of interest.

- Half a million star+gas particles = 720MB
- Projected subset of data = 500MB
- FITS datafile of above subset = 100MB
- Contour plot of data = under 1MB

Keyframing lets us do on-the-fly reduction. To create a simulation matching observations, we first do a small sim (10k particles) and create a movie. This lets us define keyframes—significant points of interaction crucial to defining the system. For example, we mark six frames (of 60+ run) that define the evolution best, from approach through merging and departure: six unevenly spaced frames that define the system.

We can then do a large scale (million-particle) simulation. At each keyframe, we first create intermediate data products using the many saved output files since the last keyframe. This includes a movie, radial velocity evolution, etc. Having created this, we can delete the massive intermediate output files and simply keep our bracketing keyframes.

After the run, we need only transfer the keyframes and the pre-reduced products to our workstation for analysis. This saves time and trouble: only 7GB of files are needed, not 150GB. And, our initial guess at reduced files are already created.

If we need to investigate sim results other than at the keyframes or in the pre-reduced set, we simply rerun the sim from the nearest keyframe (rather than from time zero).

4. Codes Available

Our own N-body+kitchensink code, **mass99**, includes the **reduce99** package for creating useful scientific output from the fairly abstract simulation space. **reduce99** handles projections of entire or per-component elements, with various summations to create surface densities, volumetric averages, and contours (as well as simple movie-making).

Nemo is a large set of tools that can handle individual frames (snapshots), plot angular momentum, and create conservation plots. Nemo is both flexible and extensible. Topsy is a good freeware 4-D data analysis package, with a GUI that includes the ability to move around your simulation data space.

Other packages include AstroMD, NVisF (N-body Visualization Framework, formerly NVision), and Partiview (also used within Nemo). In addition, packages like IRAF, MIDAS, IDL, Geomview, QDP, plplot, gnuplot, and other raw plotting tools and scripts are often useful.

It is useful to be able to create pseudo-observations, as they can be directly loaded into your favorite data analysis (not modeling) package to cross-compare with real (observational) data. For example, a pseudo-radio contour of a 10^9 particle simulation imported into AIPS allows direct numerical comparison with the actual observation.

5. Conclusion

For our current work, we study AM0644-741 as our sample case. It is a wonderful example of a Lindsay-Shapley Ring, a beautiful off-axis collision for which we have a good orientation angle for viewing. We see an outer ring of presumed star formation and an older interior, a possible double ring, uneven star formation, blobby gas, a break, a possible ejecta or jet (or tidal feature).

A lot of people do 3-D visualization. Ironically, we want lossy 2-D visualizations of our perfect 4-Dimensional N-body+SPH simulations. Our goal is to just be handed a file and not know if it is an observation or a simulation. This is the antithesis of data mining. The key is to use our simulations to recreate observations. Only then can ‘forensic data mining’ have meaning. We welcome feedback and suggestions on further tools to support this. All work will go up at <http://science.gmu.edu/~aantunes/>.

References

- Antunes, A. & Wallin, J. 2001, AAS, 199, 8706A
- Borne, K. 1988, ApJ, 330, 38
- Higdon, J. & Wallin, J. 1997, ApJ, 474, 686