LA-UR-09-08278

*Approved for public release;
distribution is unlimited.*

Title: Verification of the Time Evolution of Cosmological Simulations via Hypothesis-Driven Comparative and Quantitative Visualization

Author(s): Chung-Hsing Hsu 198251 CCS-1
James P. Ahrens 113788 CCS-1
Katrin Heitmann 175878 ISR-1

Intended for: Pacific Visualization Conference March 3, 2010

Verification of the Time Evolution of Cosmological Simulations via Hypothesis-Driven Comparative and Quantitative Visualization

Chung-Hsing Hsu, James P. Ahrens, and Katrin Heitmann

Abstract

We describe a visualization-assisted process for the verification of cosmological simulation codes. The need for code verification stems from the requirement for very accurate predictions in order to interpret observational data confidently. We compare different simulation algorithms in order to reliably predict differences in simulation results and understand their dependence on input parameter settings. Our verification process consists of the integration of iterative hypothesis-verification with comparative, feature and quantitative visualization. We validate this process by verifying the time evolution results of three different cosmology simulation codes. The purpose of this verification is to study the accuracy of AMR methods versus other N-body simulation methods for cosmological simulations.

Keywords: Visualization in Earth, Space, and Environmental Sciences; Hypothesis Testing, Visual Evidence; Feature Detection and Tracking

Index Terms: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

1 Introduction

Understanding the physics of dark energy and dark matter – the two components dominating the content of the Universe – is the foremost challenge in cosmology today. This task requires very accurate predictions from high-performance simulation codes to interpret the observational data from ground and space based cosmology missions. Structure formation is very sensitive to the physics of dark energy. Current observations constrain the nature of dark energy at the 10% level. In order to go the next step, observations and modeling have to improve by an order of magnitude. To achieve this goal with respect to the modeling, extensive code verification exercises have to be carried out. Code verification in this context means that the accuracy of the algorithm used to solve the set of equations underlying our physical model is understood and controlled. Often this is achieved by numerically solving problems with a known analytic solution and evaluating how well the algorithm handles the problem. Cosmological simulations are highly nonlinear and while exactly solvable problems give some hints about the accuracy of the algorithms they do not capture the full problem to be solved. An important part of the verification process is therefore the comparison of different algorithms, along with characterization and understanding of the differences in the results.

The contribution of this paper is the description and real-world evaluation of a visualization-assisted process for code verification. We integrate *iterative hypothesis-verification* with *comparative, feature and quantitative visualization* to facilitate this objective. The first step is based on a qualitative comparison of the different code results through comparative visualization. In this way feature-based differences can be easily and efficiently identified. In the next step, a hypothesis for the cause of the differences is formulated in such a way that it can be tested in a quantitative manner. Quantitative comparative visualization is used to test the hypothesis. The visualization leads to new insights and questions, that then can be tested in qualitative ways – the process is iterative and allows us with every step to gain a deeper understanding of the underlying simulations. We believe this elucidation of specific visualization processes that help achieve a specific goal (in this case, code verification) is critical to understanding and advancing the science of visualization. Our metrics for success are the speed and ease we obtain verification results when we apply our process to a real-world verification problem. Specifically, we demonstrate with two cosmological code verification tasks how the process works.

This paper is organized as follows. In Section 2 we provide a brief overview of previous work. We describe in detail the code verification task and outline the visualization process in Section 3. A detailed account of how the process applies to two examples is presented in Section 4. In Section 5, we conclude and provide future directions.

2 Related Work

In this section, we describe related approaches. We classify these approaches into three categories: iterative hypothesis-verification, comparative visualization and feature extraction/quantitative analysis.

2.1 Iterative Hypothesis-Verification

Chen et al. [1] describe the visualization process as an iterative search process. Each iteration produces a visualization that may increase the user's information and knowledge. In the general case, the search space is extremely large. For example, parameters such as types of visualization algorithms, configuration of these algorithms and view positions are just some of the search space that can be considered.

Kehrer et al. [8] quickly explore this search space by rapidly generating promising hypotheses. Specifically, they seek to identify regions in climate data which are sensitive to atmospheric climate change and then statistically verify whether or not these regions are robust indicators of climate change. Iterative visualization and interaction was used with sensitivity metrics to narrow down regions of interest.

Our work is different from Kehrer et al.'s in that we narrow the search space by focusing on code verification and provide a structured process for traversing the space. Our iterative process builds up qualitative and quantitative evidence that highlights the differences between the codes being verified.

2.2 Comparative Visualization

Our work endeavors to understand the sources of inconsistency between different cosmology codes. It starts with perceiving the difference in the simulation results through comparative visualization. Comparative visualization supports the ability to visually study, multiple related visualizations.

Spreadsheets/small multiples are a common method to present visual differences [7, 9], and have been adopted in many visualization tools. Features can be enhanced to make user judgment easier [14, 22]. Recently, researchers started to address the potential "change blindness" in the side-by-side comparative visualization; for example, by visualizing results from different simulations in the same volume [5]. We attempted to use a visual differencing scheme similiar to [5] but because of the significant differences between the simulation codes visually cluttered results were generated. At this time, cosmologists prefer side-by-side comparsions instead of more advanced differencing methods because side-by-side comparisons offer direct visual confirmation about the properities of the source data as well as the ability to intuit differences. Side-by-side comparisons can easily be generated automatically. For example, "Show Me" [10] is an integrated set of user interface commands and defaults that incorporate automatic comparative presentations into a commercial visual analysis system called Tableau. Unfortunately, "Show Me" is designed to handle spreadsheet data, not large-scale spatial-based scientific data. Our process incorporates automated side-by-side comparsion of simulations results.

2.3 Feature Extraction and Quantitative Analysis

One difficulty in the visualization and quantitative analysis of cosmological datasets is the massive size of the cosmological dataset. Directly visualizing every particle in these datasets via a glyph, such as a point or sphere, introduces visual clutter, making important structures not easily discernible in the resulting images.

To address this problem researchers have pioneered the use of feature-based approaches to extract meaningful higher-level structure from scientific datasets [18, 19]. A feature is, by definition, a region in the dataset that satisfies certain constraints [18]. The exact definition of a feature varies by domains. For example, an ocean eddy, a feature of interest in oceanography, is defined as a flow with semi-closed circulation pattern [23]. Navrátil et al. [12] convert particle data from a cosmological simulation into a grid-based representation from which it is easier to extract structural information. In general, a feature describes a coherent structure – an "effect" that persists for a significant amount of time [20]. In the cosmological context, the halo (a cluster of particles) is a common feature of interest. A feature-based approach allows users to extract features and to visualize, track, isolate, and quantify their evolution. A key to success is the efficient extraction of the features of interest. In this paper, we present efficient feature analysis methods for particle-based cosmological data.

Quantitative visualization, by definition, allows the user to extract quantitative information directly from the visual presentation. Peskin et al. started to advocate for its interactive form [15] to enhance scientific and engineering computational simulation prototyping. Recent efforts include query-driven visualization [3, 4, 17], and quantitative visualization programming languages [11]. We support the combination of feature-extraction and quantitative visualization by emphasizing the creation and use of measurable features to help quantify the differences between simulations.

3 Problem: Verification of Cosmology Codes

We propose a feature-based visualization-assisted process to simplify the verification of cosmology codes. In this section, we provide a description of the verification task and the new process to solve it. We also describe the tool support needed to instantiate the process.

3.1 Code Verification Task

The aim in this paper is to characterize differences in the results from three cosmology codes started from the same initial conditions at redshift $z = 50$ and to identify the causes underlying these differences. [1]

3.1.1 Cosmology Codes

The three cosmology codes under investigation are GADGET-2 [21], MC^2 [16], and Enzo [13]. Each of these codes were independently developed by different groups at institutions around the world, GADGET-2 at the Max Planck Institute, MC^2 at Los Alamos and Enzo at UC San Diego. While all three codes are solving the same N-body problem – the evolution of the dark matter distribution in an expanding universe – their underlying algorithms are rather different. Simulation algorithms employed to investigate structure formation range from grid-based methods (particle mesh (PM) and adaptive mesh refinement (AMR) methods) to tree-based methods to mixed methods such as tree-PM. MC^2 and Enzo fall in the first category (MC^2 being a pure PM code and Enzo being an AMR code) while GADGET-2 is a tree-PM code. All three codes are well-established and have been used extensively for cosmological simulations.

3.1.2 Initial Conditions

Maintaining the accuracy of cosmological simulations is challenging due to the multi-scale complexity in the time and spatial range. The spatial dynamics range of realistic problems is anywhere between two to five orders of magnitude. Therefore, we can design an experiment in a regime that demands high resolution to verify whether these codes reach the desired accuracy for gravitational interactions alone.

We investigate the simulation of the Standard Model of cosmology in a box with 90.14 Megaparsec (Mpc) in each dimension. [2] Due to the small size of the simulation box, the force resolution of all codes is, in principle, sufficient to capture properties of individual features. Each simulation was run with 256^3 particles.

3.2 Problem Solution: A Code Verification Process

Our specific code verification process is as follows:

Registration of the codes: First the codes are registered making sure all measurable axes, such as time steps and spatial coordinates, are in agreement.

- **Step 1: Define/refine features:** In this step, a measurable feature is defined and an algorithm is coded that identifies the feature in the scientific data. This may be as simple as a density metric or as complex as a precise structural feature.

- **Step 2: Formulate/refine hypothesis about a measurable difference between the simulation codes:** Given the feature defined in Step 1, the scientist makes a hypothesis about the simulation codes. The feature-extraction algorithm is applied to all codes to create a measurement for each code and these measurements are compared.

- **Step 3: Qualitative comparative visualization:**[3] The 3D visualizations created in this step are side-by-side comparisons that visually highlight the differences between the simulations. This step requires some visualization expertise to effectively identify simulation differences.

- **Step 4: Quantitative comparative visualization:** The 2D plots created in this step are side-by-side comparisons that highlight the quantitative differences between the simulations. Quantitative results reinforce the hypothesis verification.

[1] In cosmological simulation, redshift z is often used to specify time points. In contrast to typical time-point progression that increases time-point values, redshift progression decreases redshift values. For these simulations, time runs from z=50 (the big bang) to z=0 (present day).

[2] Each Mpc is equal to 3,262,000 light years or 2×10^{19} miles.

[3] We use the word *qualitative* to describe 3D visualizations and the word quantitative to describe 2D plots. Clearly, 3D visualizations present quantitative data. However, we use these words to identify how 3D visualization (for exploration and intuition) and 2D plots (for measurements and analysis) are typically used by cosmologists in the visualization process.

Repeat starting at step 1 until the codes are verified.

There are number of benefits to this process. The first is that the verification process now has a well defined structure. As features are identified and the feature extraction method is coded, these advances can be reused to identify more sophisticated features and improved measurements. The scientific method demands quantitative evidence to support or refute a given hypothesis; our process contains the generation of this evidence as a key step. We recognize that the definition of the features and the development of extraction algorithms is the longest step in the process, however, the planned integration of these algorithms into the visualization process and planned reuse of these feature extraction methods should reduce the overall coding burden.

3.3 Application of the Verification Process

Registration of the Cosmology Codes

In order to register the cosmology codes, two technical issues need to be resolved. The first issue is that different codes use different time steps. As a consequence, the result from each code contains a set of snapshots of the simulation box for a different set of time points. We selected a sequence of reference time points. The results from each code are interpolated linearly, if necessary, to match with the sequence of reference time points. This way, we can compare snapshots of results from all three codes at the same time scale.

Another technical issue is that different codes assign particle IDs differently. As a result, we may need an additional analysis step to correlate particle IDs across multiple cosmology codes in order to track the motion of a particle in different codes.

Step 1: Define/refine measurement feature

In the cosmology community, the halo paradigm is central to the analysis of dark-matter simulations. Unfortunately there is no unique algorithmic definition for identifying a dark matter halo. In this paper we use exclusively a common cosmological definition, the Friends-of-Friends (FOF) metric [2]. In this definition, a pair of particles are designated as friends if their distance in space is within a certain threshold called *linking length*. A halo is defined as a set of particles connected by one or more friendship relations, i.e., friends-of-friends.

The linking length is measured in units of the mean inter-particle spacing. In this paper, we use 0.2 as the linking length. We require that the objects we find have at least ten particles before we call them "halos". The purpose of this threshold is to reject spurious halos, i.e., groups of friends that do not form graviationally-bound objects in the simulation. Finally, we distinguish between the mass and the size of a halo. The *mass* is the number of particles forming the halo whereas the *size* is reserved for spatial dimensionality. The specific linking length, particle threshold, and FOF algorithm provide a community accepted density threshold estimator to find gravitationally-bound objects that is considered reliable to compare simulation codes.

A straight-forward halo-finding algorithm based on FOF is to check each and every pair of particles to determine whether they are friends or not. Mathematically, the algorithm can be described as follows:

$$\forall (i,j), i \neq j, \text{ if } d(i,j) < l \text{ then } \mathbf{union}(i{,}j)$$

where i and j are particle IDs, $d(i,j)$ is the Euclidean distance between the two particles, l is the user-defined threshold value, and **union** is a book-keeping operation that indicates that the two particles are in the same group. Given n particles, a baseline time complexity of the algorithm is $O(n^2)$ – if each union operation has an $O(1)$ average cost. Obviously, such an implementation will hinder interactivity when n becomes large.

We improve the performance of the above algorithm by reducing the number of pairs (i,j) examined. This reduction is facilitated via a balanced kd-tree. A balanced kd-tree is a data structure that organizes points in a k-dimensional space in such a way that the number of points in a subtree at each level are equal. Building a balanced kd-tree from n points takes $O(n \log n)$ operations. A recursive algorithm starts at the leaf nodes (single particles) and merges nodes into halos by checking if the particles are within range of each other. As particles are merged, subtree bounding boxes are used to skip the testing and merging of spatially distant subtrees. The cost of the algorithm, primarily in the merge step, is data dependent based on the number of particles in a halo. Our new algorithm only takes *two minutes* to analyze 256^3 particles, and efficiently processes the hundreds of data files necessary to perform this verification study. It is now in daily use by the cosmologists to analyze their results.

Step 2: Formulate/refine hypothesis about a measurable difference between the simulation codes

Specific hypotheses will be explored in the case studies presented in Section 4.

Step 3, 4: Qualitative and quantitative comparative visualization

Qualitative and quantitative visualization support is offered in ParaView. ParaView is an open-source, multi-platform data analysis and visualization application. In ParaView the user can automatically set up a collection of related 3D visualizations or 2D plots in a one or two dimensional configuration. The user identifies the parameter that will change along each axis and the number of frames horizontally and vertically. The visualizations in the frames are linked so that the same interactively selected view angle is shown. We ran our verification process within ParaView. The use of ParaView is not required, as any visualization tool that supports the integration of new code and offers comparative qualitative and quantitative visualization tools can be used.

4 Case Studies: Evolution of the Halo Population

In the following we demonstrate with two examples how the feature-based, visualization-assisted process can be used in a scientific setting to answer specific questions in an efficient and seamless way. We will document the specific scientific process we went through in detail to verify the cosmology codes and highlight how the process led us quickly to key scientific results. We consider the speed and ease that the cosmologists were able to verify their simulations to be one of the best tests of the effectiveness of our visualization approach. In the first example we study the evolution of the entire halo population in the simulation with time. This example addresses questions about statistical measures from the simulations, such as halo counts as function of mass and time and therefore global results of the simulations. In the second example we carry out a focused study of the formation of one particular halo, a localized feature in the simulation. This halo was chosen due to its anomalous shape which complicates very accurate simulation. Specific small scale features of the halo and the ability of the simulation codes to track and resolve these features are of central interest.

4.1 The Evolution of the Halo Population

The number of halos as a function of their mass is called the halo mass function (or short: mass function). The mass function and its time evolution contains a lot of information about the formation of cosmological structures. The precise measurement of the mass function from observations and its prediction from theory are a field of very active research in cosmology. In the following we will show how our approach can be utilized to find a reliable and accurate prediction of the mass function and its evolution. The features of interest in this case are the halos themselves and their masses. As explained in the last section, an efficient halo finder has been developed to extract these features from the simulation – Step 1 in our

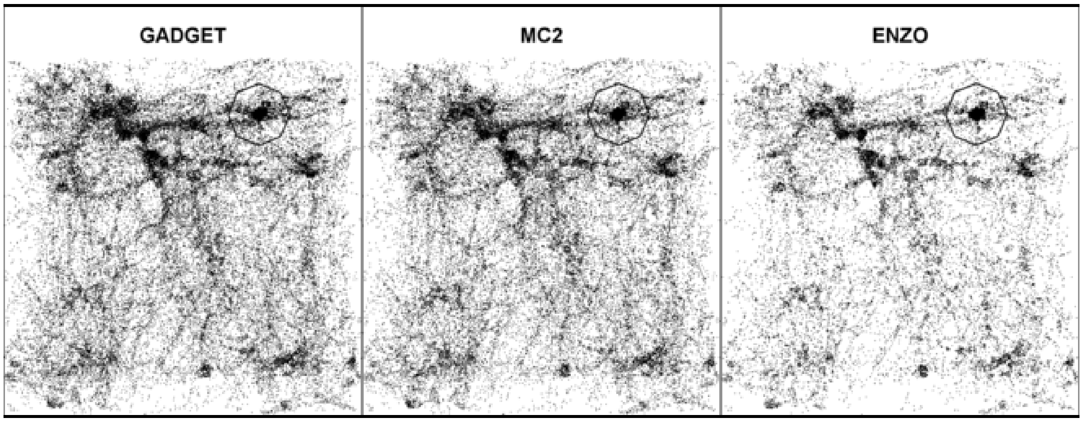

Figure 1: Comparative visualization of the distribution of the halos for the simulation results at $z = 0$. Each halo is represented by a dot. It is immediately obvious from these images that Enzo has fewer halos throughout the simulation volume and that therefore our Hypothesis 1 is false. The black circles mark the position of Halo 3. The formation history of Halo 3 will be discussed in detail in our second case study.

process.

A popular N-body algorithm for large-scale structure formation simulations is the adaptive mesh refinement (AMR) method. In AMR codes the simulation starts out with a uniform grid (the base grid) and refines specific regions in the simulation at higher resolution as time evolves. In cosmological simulations, the refinement criterion is usually set by a density threshold: if a certain density is reached, the grid is refined. We refer to the highest level of refinement as peak resolution. The AMR method has previously successfully been used in the simulation of single objects, e.g., supernovae. In cosmology, the simulation starts with a smooth Gaussian density field and over time high density region evolves. Therefore, AMR methods seem to be a natural way to perform such simulations: the hope is that higher resolution is only needed in high density regions and at later times. This would make the simulation requirements much easier to handle and save computational costs. These benefits of AMR methods need to be tested, leading to Step 2 in the process – the formulation of

Hypothesis 1: An AMR code with a peak resolution equivalent to a uniform grid code should resolve all halos of interest.

In order to test this hypothesis, three simulations are carried out: one with Enzo, an AMR code, with a 256^3 base mesh and two levels of refinement leading to a 1024^3 mesh peak resolution. For comparison, the second simulation is carried out with a pure mesh code, MC^2, with a 1024^3 mesh all throughout the evolution. To provide a second reference simulation, we also perform a third run with a high-resolution tree-PM code, GADGET-2. We store from each simulation 100 time snapshots.

Following Section 3, in Step 3 we test our hypothesis by performing a qualitative visualization comparison. Since the feature extraction algorithm (in this case the halo finder) is already integrated in the visualization tool, this step is now very easy and efficient. We visualize the halos from the three simulations at the last time step. The results are shown in Figure 1. In this case the visual interpretation of the result is very easy: our hypothesis is false. Visually Enzo has obviously far fewer halos. In Step 4 we find that Enzo has 29099 halos, MC^2 49293 halos, and GADGET-2 has 55512 halos. Therefore, we need to refine our hypothesis. One could argue that the pure halo count would have been enough to falsify our hypothesis. The visualization of the distribution of the halos contains much more information than a pure number though, it has *spatial* information about the halos. Figure 1 already hints at the fact that the halo count is different in the entire simulation volume and not just in specific regions where the AMR code might not have refined to the highest level of resolution.

4.1.1 Discussion

- An alternative to incoporating the feature extraction code within the visualization tool as we require in Step 1 of our process is the creation of the feature extraction code in the simulation. In practice, the simulations have their own unique analysis methods as part of the simulation code making the comparisons of these methods difficult. In addition, there is a sigificant time lag in the visual analysis workflow if the user needs to re-run each simulation in order to produce feature outputs for the comparative visual analysis step.

- An alternative for Step 3 is to produce the side-by-side visualizations by hand by varying the parameters to the visualization tool. For example for Figure 1, the scientists would need to input each different simulation names and use the same visualization pipeline to generate seperate visualization windows. Managing these parameter changes and windows by hand is tedious task. Our process supports the automatic generation and management of comparative visualizations. In practice, this automation reduces the time to complete Step 3 to seconds versus minutes or hours when the scientist is forced to do this by hand.

- One alternative is to rely on purely analyticial methods. In our process, we require visual inspection of the data as well (i.e. Step 3), and we have found that this typically provides siginificant additional insight (in this case, the data's spatial distribution) which drives the verification process forward quicker than a pure analtyical approach.

Next, we aim to refine our hypothesis in such a way that we gain a clear understanding why Hypothesis 1 was false. Two possible, related explanations for the deficit of halos could be:

Hypothesis 2a: The halos do not form at early times when the base resolution is still very low and cannot be recovered later.

Hypothesis 2b: Only halos of a certain size – dictated by the base grid and *not* by the peak resolution – can be captured correctly.

With the refinement of the first hypothesis, we also focus now on an additional feature in the simulation: the halo mass itself. This "feature" is automatically provided by the halo finder and will be very important to understand our results in more detail. We can test our two new hypotheses in one step by (i) dividing the halos into mass bins, (ii) tracking the number of halos forming in the simulation over time in these separate mass bins.

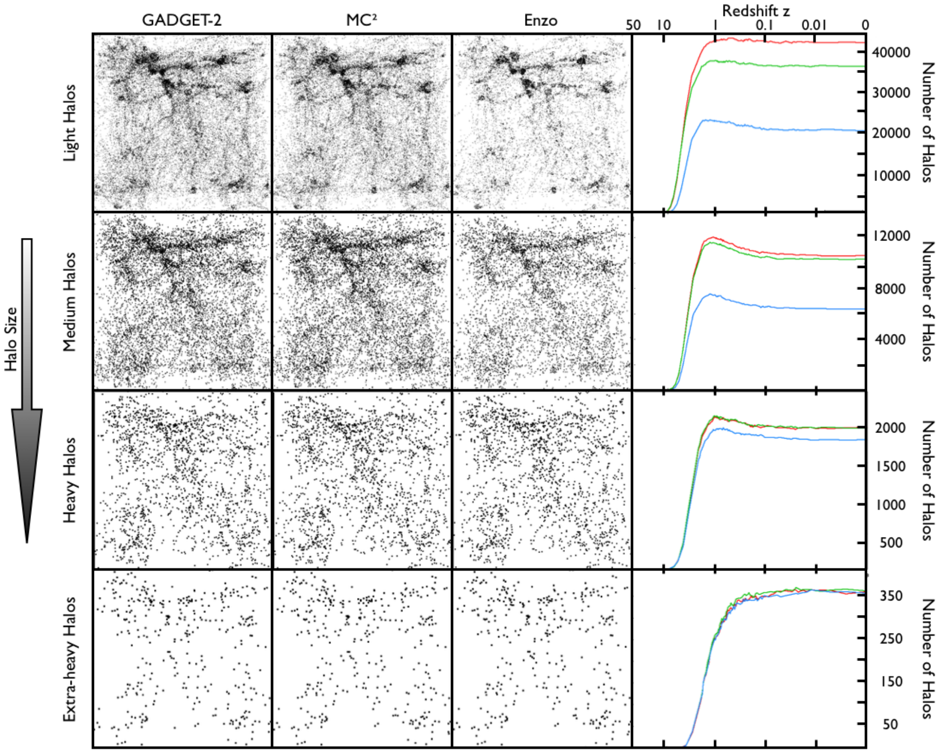

Figure 2: First three columns: comparative qualitative visualization of the distribution of the halos in different mass ranges (light to extra-heavy from top to bottom) at a single time step $z = 0$. Throughout the paper we present results from GADGET-2 in red, from MC^2 in green, and from Enzo in blue. A light halo contains 10-40 particles, a medium halo contains 41-300 particles, a heavy halo contains 301-2500 and an extra-heavy halo contains more than 2500 particles. Fourth column: comparative quantitative visualization of the number of halos in different mass ranges. In addition, information about the time evolution of the number of halos is shown. While the qualitative approach contains spatial information, the quantitative plots have extra temporal information. The combination of both allow the assessment of the validity of hypothesis 2a and 2b.

In [6] a simple criterion was derived for the force resolution required to resolve a halo of a certain size. The force resolution of Enzo's base grid would allow to capture halos with more than 2500 particles, the first refinement level would allow the resolution of halos with more than 300 particles, and the highest refinement grid would allow the resolution of halos as light as 40 particles. Since MC^2 uniform base resolution is equivalent to Enzo's peak resolution, MC^2 should reliably capture halos with at least 40 particles. While halos with fewer particles are often unphysical (random particles which are by chance close to each other get linked together even though they do not form a bound structure) we also investigate halos with less than 40 particles to demonstrate the robustness of our hypothesis 2b. The final results of our comparative qualitative and quantitative visualization analysis are summarized in Figure 2. We would like to emphasize that this figure only shows the summary of an interactive process in which the linked-view comparative visualization techniques were used to get a better intuitive understanding of the simulations to answer the scientific question.

From top to bottom, Figure 2 shows the different mass ranges described above, starting with the lightest halo and ending with the heaviest halo. The first three columns show the results for the different simulations at the final redshift and column four presents the quantitative results including time evolution. The qualitative inspection of the simulation results through visualization (column 1 - 3) in this case does not lead to an immediate answer as it did in the previous case. It is clear visually in the figure that Enzo has fewer light halos and most likely also fewer medium halos. A judgment about the number of heavy and extra-heavy halos by inspection of the qualitative plots is not possible and this time Step 4 in our process becomes essential. In column 4 in Figure 2 the results from column 1 - 3 are displayed in a quantitative manner: we show the number of halos and the evolution over time in four different mass bins. The colors of the curves correspond to the coloring scheme of the visualization of the simulations. Now it is easy to see that Enzo has a large deficit in the halo count for the light and medium halos and also a lack of heavy halos. Only the very heavy halos are captured by Enzo correctly. We therefore have confirmed that our hypothesis 2b is correct. Figure 2 also contains information about Hypothesis 2a: it is very clear that the deficit of halos in the Enzo simulation is rooted at early times: halos are missing from the beginning and cannot be recovered over time (e.g. the curve for the medium halos is not catching up with the MC^2 result at later times).

At this point, the verification process is successfully completed: we now have a good understanding of what halos can be captured reliably by AMR codes. We verified a criterion for the halo masses that can be resolved given a certain force resolution. Of course the performance of AMR codes can easily be improved by either starting the simulation with a much larger base grid or setting the refinement criteria much more aggressive so that the refinement starts long before the first halos form.

4.1.2 Discussion

- Our visualization process structures the visual analysis so that it produces a chain of documentary evidence that supports the verification process. The hypothesis text and Figure 1 and 2 were used to present the verification issues to the Enzo scientific team.
- Our visualization process requires interactive exploration. It

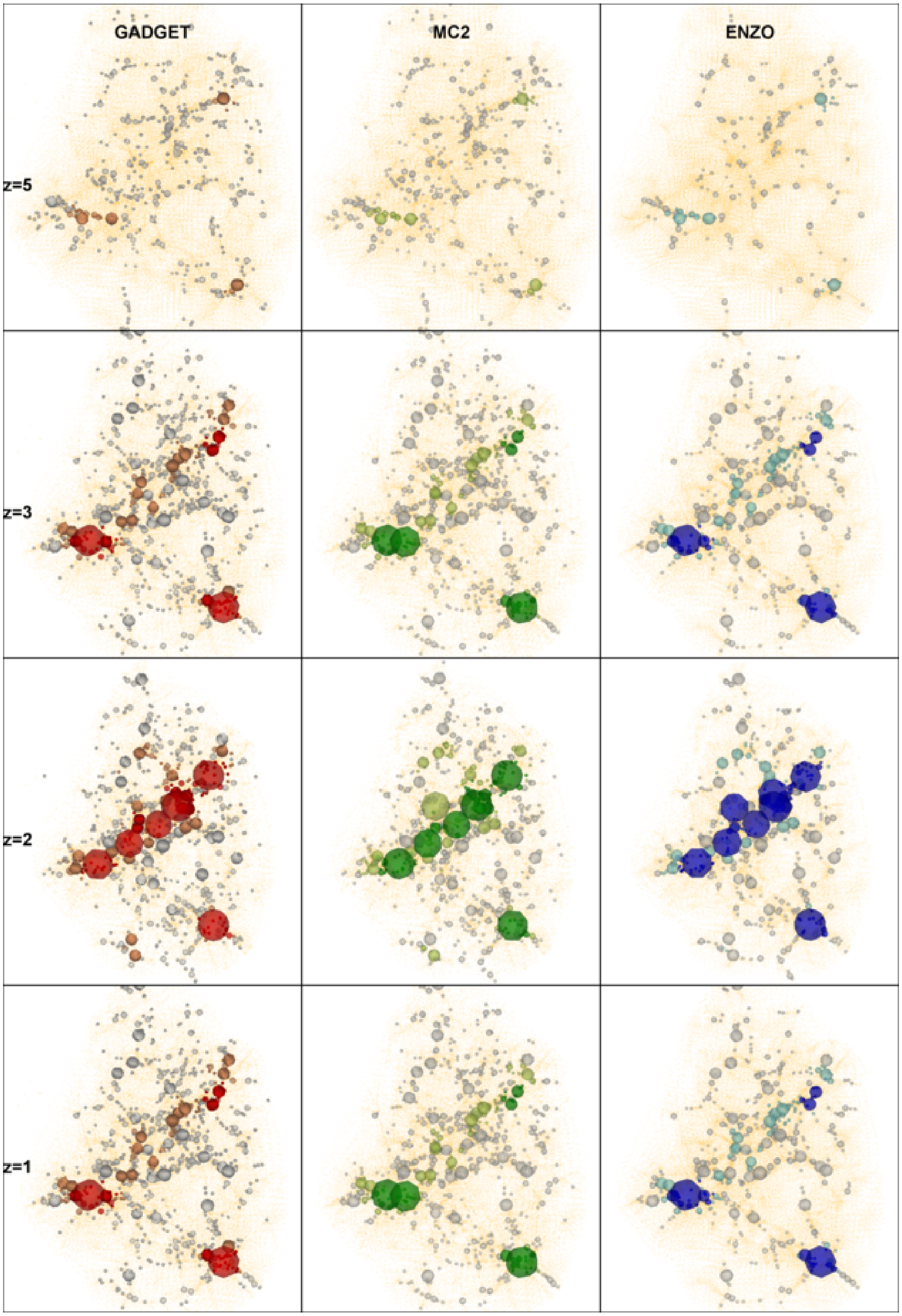

Figure 3: Comparative qualitative visualization: Four different time snapshots of the formation of Halo 3 at $z = 5,\ 3,\ 2$ and $z = 1$. The spheres represent halos in four different mass bins (light, medium, heavy, extra-heavy), the size of the spheres is scaled with respect to the mass bin the halo belongs to. The intensity of color contains information about the density environment in which the halos live: three intensity levels are used to reflect three density levels: low, medium, and high, and the darker the color, the higher the density. Particles which do not belong to any halo are shown as yellow dots. At the final time step, not shown here, all particles and halos will belong to the final Halo 3. As in the previous case study, red is reserved for GADGET-2, green for MC^2, and blue for Enzo. The region we are investigating is approximately 5 Mpc wide.

is hard to convey in a static publication the "search" aspect of this process (i.e. working to identify a qualitiative and quantitative measure that would distinguish between the simulations). Alternative approaches that did not include this interactive visualization "trial and error" process might have missed the key aspects (i.e. formation issues based on density regions) that we are able to find through our comparative visualization exploration process.

4.2 Halo Formation Process

In the last section, we have established that the ability of AMR codes with standard refinement settings to resolve halos of a certain size is dictated by the size of the base grid and not by the peak resolution. The next obvious question is: are the structures within the extra-heavy halos resolved at the level of the peak resolution or will details be missing which were not seeded at early times where the base grid dominated the simulation? If the AMR code would not be able to resolve features in a high-density region and would resolve features at the base grid resolution, the use of AMR for cosmological gravity only simulations would become less attractive. We beleive that applying our process to this different problem (small-scale vs. large-scale structure formation) we further document the benefits of our structured visualization process for verification.

To answer these questions, we present a second case study in which we closely follow the formation of the third heaviest halo (Halo 3) in the simulation. Its position within the full simulation is marked in Figure 1 by black circles. We choose this particular halo because of its anomalous shape. A large fraction of halos ($\sim 80\%$) are relatively spherical and relaxed. They form through accretion of mass and lighter halos from the surrounding area. Halo 3 is slightly unusual in that it forms through the merger of several major halos. This leads to a slightly elongated shape. The formation of Halo 3 is most likely not complete at $z = 0$ and in the future this halo would become more spherical and relaxed. Being still in the formation phase at the final redshift makes this halo particularly suited for our aim: we can study the evolution of many small substructures and investigate if these substructures are reliably resolved by the AMR code.

In Step 1, we continue to track the formation and evolution of the halo population that will lead finally to the formation of Halo 3. We also added a new feature of interest, density. The density is determined by the number of particles inside a sphere of a fixed diameter (2.82 Mpc) of the halo center. To compute density we were able to quickly modify the halo finder code to reuse the kd-tree to count all particles within a given radius from the halo center. Following our process (Step 2) we formulate the hypothesis:

Hypothesis 1: The AMR code should resolve substructures in a highly overdense region reliably.

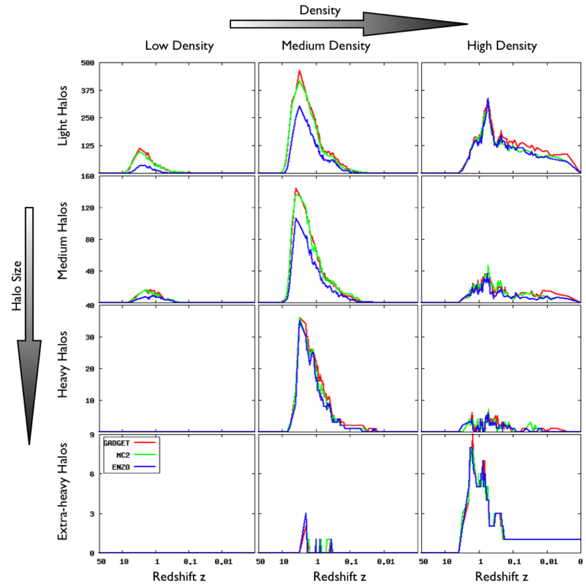

Figure 4: Comparative quantitative visualization: Comparison of different halo sizes (top to bottom: light to extra-heavy halos) and of halo counts in different density regions (left to right: low to high density region). Each panel shows the time evolution of the halo count in a specific mass bin and a specific density region. As in the previous case study, Enzo loses light halos. This time we have the additional information, that most light halos are lost in lower density regions. At the final time step, only one extra-heavy halo will remain, which has absorbed all lighter halos from earlier times. This explains why the panels in the first three rows all end up at zero at $z = 0$ and the last panel in the bottom right corner ends up at one.

In order to test this hypothesis, we track the evolution of Halo 3 over time. At the final time step, the mass of Halo 3 is the same for all three codes within a few percent. We identify all particles which belong to Halo 3 in the final time step and trace them back via their particle id to earlier times. For each of these earlier time steps we then identify all the halos in the particle distribution which will become finally Halo 3. A qualitative comparison for four of the 100 time snapshots is shown in Figure 3. Each halo that was identified is displayed by a sphere. The spheres are shown in four sizes, corresponding to light, medium, heavy and extra-heavy halos (the mass definitions here are the same as in case study I). In addition, the halos are colored with respect to their surrounding density: the darker the color, the higher the density. Low density ranges from 0 to 499 particles, medium density ranges from 500-5000 particles and high density is more than 5000 particles within the sphere. The visualization presents information about: the masses of the halos, the environment they live in, and temporal and spatial information. Following Step 3, we can now verify or falsify our hypothesis. In the first row at redshift $z = 5$, Enzo is again missing a large number of light halos. Especially in regions without heavier halos, this is very apparent. Over time, the discrepancy seems to decrease, though even at $z = 1$ it appears that some of the lighter halos are missing. The visualization of the halos contains spatial information that allows us to compare all three simulations in great detail.

The next step, Step 4, is to cast these observations into more quantitative results. Before we do so, the visual inspection of the formation history leads to a second hypothesis:

Hypothesis 2: The AMR code loses more light halos in lower density regions.

Figure 4 holds the definite answer to both of our questions. Here we show the halo population evolving with time and separated from top to bottom into light to extra-heavy halos and from left to right into low, medium, and high density environments. It is very interesting to note that in the high density region (third column) all codes resolve halos of all sizes reliably. The AMR code also resolves heavy halos in all environments reliably (remember, that in the global case heavy halos were missing in the AMR simulation). In the low and medium density region, Enzo is not able to capture all light and medium halos. The great advantage of combining both quantitative and qualitative comparison becomes very clear if we compare the results from GADGET-2 and MC^2. If we would have only the quantitative results from Figure 4 we would conclude that GADGET-2 and MC^2 are in almost perfect agreement with respect to Halo 3. The additional visual spatial information in Figure 3 shows quite a few small differences, which purely statistical measures such as halo counts would not have revealed.

To summarize the results from this second case study: We falsified Hypothesis 1: without more aggressive code settings the AMR code will not be able to resolve all small scale features reliably.

As for the global case, the resolution of the light halos is also a problem in the local study. Nevertheless, the refinement did help in capturing medium halos in higher density regions. Following our code verification visualization process, we were able to draw the following conclusion about AMR codes and their capabilities to track structure formation correctly. We found that (i) the base grid has to be large enough from the start to capture features that evolve much later, (ii) the refinement criteria (e.g. density thresholds) need to be chosen in such a way, that they capture structure formation processes early. Once halos are formed, it is usually too late. Appropriate usage guidance for AMR simulation codes for accurate output is a significant result for the cosmological simulation community that was found via our visualization process.

4.2.1 Discussion

- Cosmologists found our structured process useful since it enforced quantitative evidence gathering and guided their next exploratory steps in the massive search space of possibilities. In the past, exploratory visualization and quantitiative analysis were typically separate tasks. Analysis codes (such as halo identification) were not integrated within the visualization process. By following our process, of generating 3D visualizations, gaining insight, and then producing 2D quantitative plots the cosmologists were more productive in their code verification work.

5 Conclusions and Future Work

In this paper we described a visualization-assisted process for the verification of cosmological simulation codes. The need for code verification stems from the requirement for very accurate predictions in order to interpret observational data confidently. An important part of this task is the comparison of different algorithms so that we are able to reliably predict the differences in the simulation results and understand their dependence on the code parameter settings.

We presented two case studies that verified the accuracy of AMR methods versus other N-body simulation methods. We found that visualization facilitated the process of code verification. Specifically, we gained insight in the appropriate parameter settings for AMR cosmology codes. AMR is a common optimization strategy in many scientific simulation endeavors and our verification process can provide guidance for similiar verification projects in other fields. We found that combining comparative, feature and quantitative visualization together greatly improved the efficiency and insight in performing code verification.

In the future, we plan to improve our visualization process by making the steps of feature extraction and comparative visualization easier to use and more effective and efficient. We will explore the integration of a quantitative visualization language to simplify the definition of measurable features. We will include advanced comparative visualization techniques that incorporate differences in a single visualization to augment our current side-by-side presentation.

Acknowledgements

We thank Tamara Munzner, David Laidlaw, and the anonymous reviewers for helpful comments on previous drafts of this paper.

References

[1] M. Chen, D. Ebert, H. Hagen, R. Laramee, R. van Liere, K.-L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver. Data, information, and knowledge in visualization. *IEEE Computer Graphics and Applications*, 29(1):12–19, Jan–Feb 2009.

[2] M. Davis, G. Efstathiou, C. Frenk, and S. White. The evolution of large-scale structure in a universe dominated by cold dark matter. *The Astrophysical Journal*, 292:371–394, May 1985.

[3] M. Glatter, J. Huang, J. Gao, and C. Mollenhour. Scalable data servers for large multivariate volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1291–1298, Sep/Oct 2006.

[4] L. Gosink, J. Anderson, E. Bethel, and K. Joy. Query-driven visualization of time-varying adaptive mesh refinement data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1715–1722, Nov/Dec 2008.

[5] S. Haroz, K.-L. Ma, and K. Heitmann. Multiple uncertainties in time-variant cosmological particle data. In *IEEE Pacific Visualization Symposium*, pages 207–214, Mar. 2008.

[6] K. Heitmann, Z. Lukić, S. Habib, and P. Ricker. Capturing halos at high redshifts. *The Astrophysical Journal*, 642:L85–L88, May 2006.

[7] T. Jankun-Kelly and K.-L. Ma. Visualization exploration and encapsulation via a spreadsheet-like interface. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):275–287, July 2001.

[8] J. Kehrer, F. Ladstädter, P. Muigg, H. Doleisch, A. Steiner, and H. Hauser. Hypothesis generation in climate research with interactive visual data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1579–1586, Nov/Dec 2008.

[9] M. Levoy. Spreadsheets for images. In *ACM SIGGRAPH Conference*, pages 139–146, July 1994.

[10] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, Nov/Dec 2007.

[11] P. McCormick, E. Anderson, S. Martin, C. Brownlee, J. Inman, M. Maltrud, M. Kim, J. Ahrens, and L. Nau. Quantitatively driven visualization and analysis on emerging architectures. *Journal of Physics: Conference Series*, 125:012095 (10pp), 2008.

[12] P. Navrátil, J. Johnson, and V. Bromm. Visualization of cosmological particle-based datasets. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1712–1718, Nov/Dec 2007.

[13] B. O'shea, G. Bryan, J. Bordner, M. Norman, T. Abel, R. Harkness, and A. Kritsuk. Introducing Enzo, an AMR cosmology application. In T. Plewa, T. Linde, and V. Weirs, editors, *Adaptive Mesh Refinement - Theory and Applications*, pages 341–349. Springer Berlin, 2005.

[14] S. Park, B. Budge, L. Linsen, B. Hamann, and K. Joy. Multi-dimensional transfer functions for interactive 3D flow visualization. In *Pacific Conference on Computer Graphics and Applications*, pages 177–185, Oct. 2004.

[15] R. Peskin, S. Walther, A. Froncioni, and T. Boubez. Interactive quantitative visualization. *IBM Journal of Research and Development*, 35(1–2):205–226, Jan/Mar 1991.

[16] J. Qiang, R. Ryne, S. Habib, and V. Decyk. An object-oriented parallel particle-in-cell code for beam dynamics simulation in linear accelerators. *Journal of Computational Physics*, 163(2):434–451, Sept. 2000.

[17] O. Rübel, Prabhat, K. Wu, H. Childs, J. Meredith, C. Geddes, E. Cormier-Michel, S. Ahern, G. Weber, P. Messmer, H. Hagen, B. Hamann, and E. Bethel. High performance multivariate visual data exploration for extremely large data. In *ACM/IEEE Conference on Supercomputing*, pages 1–12, Nov. 2008.

[18] R. Samtaney, D. Silver, N. Zabusky, and J. Cao. Visualizing features and tracking their evolution. *IEEE Computer Magazine*, 27(7):20–27, July 1994.

[19] D. Silver. Feature visualization. In G. Nielson, H. Hagen, and H. Müller, editors, *Scientific Visualization, Overviews, Methodologies, and Techniques*, pages 279–293. IEEE Computer Society, 1997.

[20] D. Silver and X. Wang. Tracking and visualizing turbulent 3D features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, Apr-Jun 1997.

[21] V. Springel. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364(4):1105–1134, Dec. 2005.

[22] N. Svakhine, Y. Jang, D. Ebert, and K. Gaither. Illustration and photography inspired visualization of flows and volumes. In *IEEE Visualization Conference*, pages 687–694, Oct. 2005.

[23] Z. Zhu and R. Moorhead II. Extracting and visualizing ocean eddies in time-varying flow fields. In *International Symposium on Flow Visualization*, pages 206–211, Sept. 1995.