FLUIDS IN THE UNIVERSE: ADAPTIVE MESH REFINEMENT IN COSMOLOGY

Using galactic clusters, galactic formation, and the first stars as examples, the author demonstrates how adaptive mesh refinement techniques can be successfully applied to cosmological research.

When computers first arrived on the desks of cosmologists some 20 years ago, the field had just undergone a revolution. For the first time, we had coherent, complete, and predictive theories of cosmological structure formation. The timing was fortuitous because these new models—built around the big bang—promised a detailed connection between the conditions of the early universe and the present.

As computer speed increased, so did the demands of cosmologists. Early machines could solve linearized versions of equations, and it slowly became clear that we could simulate the full evolution of large regions of the universe. While these simulations proved immensely useful, they usually approximated matter as a collection of collisionless particles—but their predictions broke down when forces besides gravity become important. One of the stiffest computational difficulties facing cosmologists today is how to include hydrodynamical and other nongravitational forces into simulations. Only when this happens will the comparisons between cosmological models and the real world be on even footing.

In this article, I describe a new approach to this difficult problem: one that uses the technique of *adaptive mesh refinement* (AMR).[1] This method combines adaptive refinement of interesting regions with modern grid-based methods for solving hydrodynamic equations.

Cosmic evolution

The universe began with a bang, but its early evolution was very smooth. Even 300,000 years later, when the cosmic microwave background last interacted with matter, the density distribution was uniform to one part in 100,000. These fluctuations, though small, were important. They grew under their own gravity and eventually formed the galaxies and stars we see around us today.

While early cosmic evolution can be described with perturbation theory, eventually nonlinear effects become important, and a numerical calculation is required. To date, the majority of such work has been with *N*-body methods, simulating collisionless fluids such as stars (which almost never directly collide) or dark matter. Substantial evidence points to the existence of this strange dark matter, which interacts only gravitationally and dominates the universe's matter

1521-9615/99/$10.00 © 1999 IEEE

GREG L. BRYAN
Massachusetts Institute of Technology

content. This matter's actual form is unknown, with suggestions ranging from black holes to elementary particles, and its identification remains one of the major challenges confronting cosmology today. Pending a direct detection, we study its gravitational influence. Although much progress can be made with only collisionless simulations (see the article by Simon White and Volker Springel in this issue), they neglect the presence of gaseous matter out of which stars and galaxies form.

The process of gravitational collapse from a smooth initial state can shrink an object's size by many orders of magnitude. This gives rise to the multiscale nature of cosmology and astrophysics—it also provides much of the field's excitement as well as its difficulty. A naïve attempt to model gravitational collapses with a fixed mesh will soon fail, because the object quickly collapses to the mesh spacing's scale. One solution that cosmologists often use is to shed the mesh entirely and model the gas's evolution with particles, as is done for dark matter. Although not popular in other fields that use computational fluid dynamics, this *smoothed particle hydrodynamics* approach has been the workhorse of cosmological CFD because it can adapt its resolution in response to the collapse. Because it is Lagrangian—particle trajectories describe fluid stream lines—the resolution at a point in space improves as the local density grows. This technique has been fairly successful; much of what we know about the hydrodynamics of cosmic structure formation stems from it.[2]

Despite SPH's relative success, a number of compelling reasons exist for searching for an alternative simulation technique. SPH does not have the history of error analysis that grid-based methods have, and—as with any approximation technique—having another, independent check is useful. Also, in many cases, a strictly Lagrangian method is not optimal, and resolution should not just follow the mass. Moreover, the CFD community has worked over the years to develop an accurate Eulerian family of methods that can represent shocks and other discontinuities in a few zones without resorting to using an artificial viscosity. Such viscosity tends to smear shocks (a common problem with SPH).

How does AMR work?

AMR's basic principle is straightforward. A uniform mesh covers the entire computational volume, and in regions that require higher resolution, a finer subgrid is added. If you need more resolution in a subgrid, an even finer mesh is added. This process can repeat recursively and results in a tree of grids like that shown in Figure 1. Obviously, this only helps if a relatively small volume needs to be refined at the highest resolution. While this does not hold for fields such as turbulence, it is very much true for cosmology. The uniform mesh is this hierarchy's root. Each level up the tree decreases the mesh spacing relative to its parent by a factor r, which is known as the *refinement factor*. Therefore, the resolution (defined as the ratio of the computational box's length to the cell size) at a level l is nr^l, where n is the root grid's size.

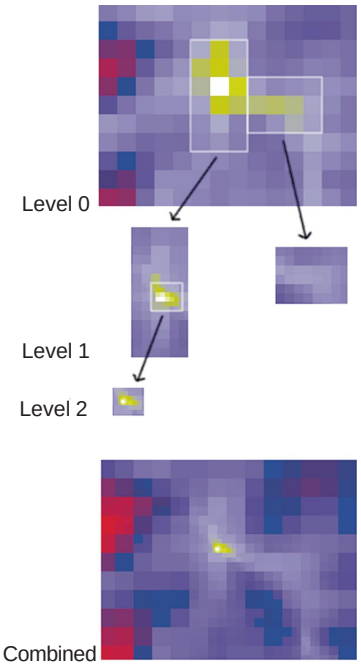


Figure 1. A 2D adaptive mesh refinement example showing a three-level, four-grid hierarchy.

The new meshes form in addition to the original "parent" mesh, so some redundancy exists; however, the reduction in bookkeeping usually makes up for this small wasted effort. Also for simplification, AMR imposes a number of restrictions on the new mesh:

- The ratio of the grid spacings between the coarse and fine meshes—the refinement factor—must be an integer. In principle, each dimension can have a different refinement factor, but in practice, they are the same (typically two to four).
- The subgrid must begin and end on a parent grid-cell's boundary. This means that all parent cells are either completely refined or completely unrefined.
- A subgrid must be rectangular, uniform, and aligned with (not rotated with respect to) its parent. It must also be completely contained within its parent.

These restrictions, although not strictly essential for the method, significantly decrease the resulting code's complexity without sacrificing much efficiency.

Figure 1 shows the hierarchy of grids at one instant in time. Of course, gas clumps move and merge, requiring regular modifications to the

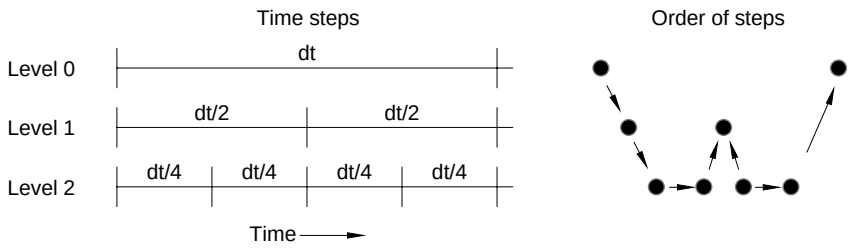


Figure 2. Finer grids on higher levels in the hierarchy must take smaller time steps, as demonstrated by the left diagram. The right diagram shows the order in which the steps are to be taken, starting with the root grid.

mesh layout. The overall control algorithm looks very much like one used in a single grid scheme, and can be written as

```
InitializeHierarchy
While (Time < StopTime)
begin
      dt = ComputeTimeStep(RootGrid)
      Time += dt
      EvolveLevel(0, Time)
      CheckForOutput(Time)
end
```

The size of the time step, `dt`, is set by both the equations we are solving and the solution itself.[3]

The AMR control algorithm is contained in the recursive `EvolveLevel` algorithm, which takes the level as an argument. It operates on all the grids at a given level of the hierarchy and resembles this pseudocode:

```
EvolveLevel(level, ParentTime)
begin
      SetBoundaryValues
      while (Time < ParentTime)
      begin
            dt = ComputeTimeStep(all grids)
            SolveHydroEquations(dt)
            Time += dt
            SetBoundaryValues
            EvolveLevel(level + 1, Time)
            FluxCorrection
            Projection
            RebuildHierarchy(level + 1)
      end
end
```

The algorithm's recursive nature keeps the code simple while ensuring the grids are advanced in the correct order. Because the time constraint for a wide class of equations requires that the time step decreases as the cell spacing decreases (usually linearly), grids at higher levels require more steps for the same time interval. Figure 2 shows this, where we have assumed $r = 2$. It also depicts the order in which the grids must advance: coarser levels go first and then finer meshes catch up. For those familiar with multigrid methods, this looks much like the W-cycle, and indeed, AMR shares many similarities with the multigrid method.

The routine `SolveHydroEquations` is the solver that advances the solution on each grid. With a few restrictions, the AMR technique can solve nearly any set of hyperbolic equations, although I limit myself here to fluid dynamics. One of AMR's advantages is that because the basic component is a uniform grid, off-the-shelf solvers can be swapped in or out. Of course, some modifications are inevitable, because the solvers must receive information through a standard calling interface rather than through common blocks or global variables, which is common in older programs.

Hyperbolic solvers require boundary values. AMR provides these either by interpolating them from the parent grid, or by directly copying them from neighboring grids on the same level. The interpolation is done both in space and time as necessary. The routine `SetBoundaryValues` accomplishes this function.

The next two functions, `FluxCorrection` and `Projection`, keep the solution consistent among all hierarchy levels. Because a refined region is simulated with at least two different resolutions (one fine and one coarse), it has two or more "solutions" (values of density, velocity, and so on) that represent the same volume of space. To maintain consistency, the coarse grid is updated with the finer values. Figure 3 shows this process schematically.

The FluxCorrection step is slightly more subtle and requires a few details about the solution vector itself, which can always be expressed in terms of conserved quantities, such as density, momentum, or energy. The solver can be thought of as a technique for determining the flux of conserved quantities from cell to cell (plus possibly a source term). To stay strictly conservative in mass and momentum, all the flux must be accounted for, something most solvers do as a matter of course. Unfortunately, the projection step upsets this flux conservation because at the refined region's boundary, the coarse and fine cells on either side of the border are updated with different flux estimates across the boundary. On one side, this

comes from the coarse-grid computation; on the other, it comes from the fine-grid solution. Conservation can be restored by correcting all the abutting coarse cells (the dark-shaded region in Figure 3) with the difference between the coarse and fine flux estimates across the boundary.

The last routine, `RebuildHierarchy`, is quite important—it is the adaptive part of AMR. Although shifting and resizing the grids might seem more efficient, the strategy that the AMR community actually adopts is to completely rebuild the hierarchy (but only for the level specified in the argument and all finer levels). Remarkably, this can be done with enough computational efficiency that it consumes only about 10% of the CPU's time. The first step is to flag those cells that need to be refined, using one or more refinement criteria defined at the beginning of the simulation. The criteria can be based on an estimate of the local truncation error, on the presence of discontinuities or shocks, or even on the local density (to mimic a Lagrangian code). The next step is to define rectangular regions that cover all the flagged cells and as few unflagged cells as possible. A number of ways to do this have been suggested by various authors, including using edge-detection algorithms from machine-vision studies.[4] The last step is to create the new grids and initialize their physical variables (density, velocity, and so on), by interpolating the physical variables from each grid's parent, and then copying values—where available—from the old grids that are being replaced. This procedure ensures that no important data is lost and that the grids always cover all regions in need of refinement.

Adding dark matter and gravity

To simulate a cosmologically interesting system, we need to also include a collisionless fluid and self-gravity. One way to model a collisionless fluid is with the Boltzmann equation—a hyperbolic partial-differential equation. Unfortunately, this approach has two flaws: first, the equation is a function both of position and velocity, so it is 6D rather than 3D. Second, the system does not naturally form collapsed, compact regions, so it is not an appealing target for AMR. Fortunately, a much more efficient method is to follow a set of particles' trajectories. This is the *N*-body technique and has been successfully used to model dark matter. However, it introduces a new type of data structure, requiring a few changes to the AMR method.

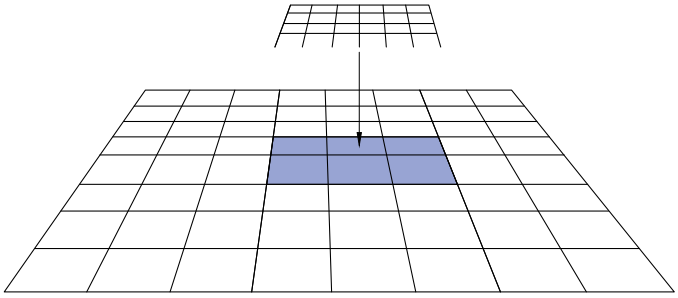


Figure 3. A schematic representation of the projection step, in which the more accurate solution on the fine grid (shown on top) replaces that on the coarser level (dark-shaded region). The flux-correction step affects the coarse cells in the light-shaded region.

Particles are associated with the most refined grid that covers their location, so they are advanced with the same time step as that grid. They also contribute to that grid's density field and use its potential field to calculate the local gravitational acceleration. Because particles move, they can change grid associations; this is handled during the hierarchy-rebuild stage. Finally, we compute the self-gravity of the combined gas and collisionless matter field by solving Poisson's equation for the potential.[5,6]

The code is written in C++ with Fortran routines for computationally intensive sections such as the solver and interpolation functions. An object-oriented approach is very natural for this application because a grid is the basic unit of work. All grid interactions can be phrased as either unary (for example, `SolveHydroEquations`) or binary (for example, interpolation) operations, which both is very appealing and helps to ensure data encapsulation.

Application examples

To give a better idea of how the method works in practice (and to demonstrate its achievements to date), I briefly describe three examples from cosmology.

Galactic clusters

Clusters of galaxies are the largest and most massive collapsed structures in the universe: light takes more than 10 million years just to cross their diameter. They also emit copious X rays and so can be observed from great distances. This makes them important cosmological probes because their abundance and evolution is a sensitive function of cosmology. However, to trans-

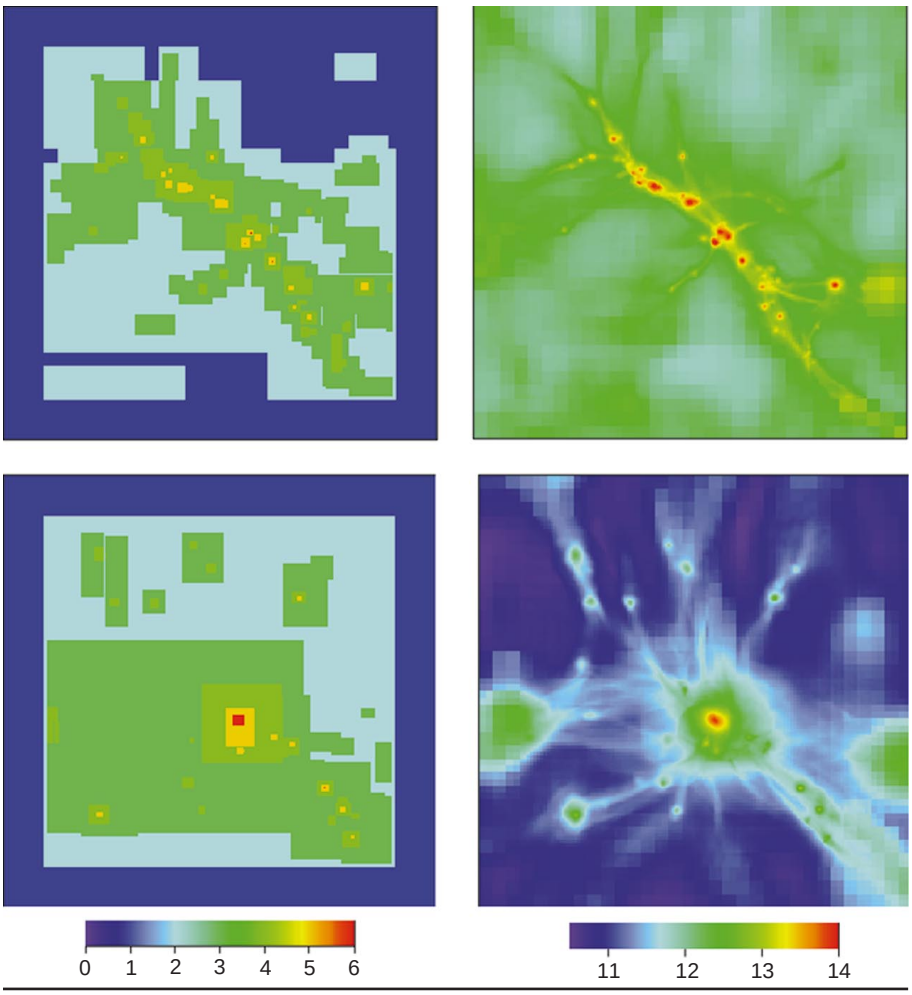


Figure 4. Two stages in the evolution of a cluster of galaxies. The top right shows the density field early in its evolution, while the top left depicts the grid-distribution projection (color-coded by level). The highest-resolution grids cover a very small region. The bottom two figures show the same quantities in the present day.

late these observations to constraints on theory requires a good understanding of cluster physics. Fortunately, clusters are relatively simple, consisting mostly of dark matter and hot gas, along with a relatively small fraction of their mass in the form of stars. Still, they are nonlinear, aspherical, and highly centrally concentrated, making them challenging to model.

Clusters are relatively rare, with typical separations of 50 to 100 megaparsecs (a parsec—pc—is 3.26 light years, or approximately 3×10^{18} cm). Therefore, obtaining enough clusters for an interesting sample requires a large computational volume. At the same time, modeling the whole volume at high resolution, even with AMR is not feasible. The matter surrounding a cluster can influence its evolution through long-range gravitational interactions, but its detailed structure is unimportant. Therefore, my colleagues and I simulate a large region at low resolution and allow AMR to create refined grids only near the cluster of interest.

Figure 4 shows two evolution stages: at early times (top) to the present day (bottom). On the right is the hot, X-ray emitting gas's projected density, while on the left is a "shadow" diagram: a 2D projection of the grids that the AMR routine has generated. The grids are color-coded by level (and hence resolution); clearly the densest regions are allotted the highest resolution, reflecting the Lagrangian refinement criterion used here. The collapse's hierarchical nature is evident first as a network of small objects aligned along a filament at early times; these then merge into steadily larger objects until the cluster itself forms at the present epoch. The merger generates strong shocks in the gas. One prediction of simulations such as this is that the continuous merging produces steady subsonic turbulence throughout the cluster, as well as coherent flows. These gas motions are of interest because they could lead to magnetic field amplification and might be responsible for the large magnetic fields observed in many clusters.

Galaxy formation

Galaxies are the traditional heart of observational cosmology. They were the first objects identified outside our own Milky Way. They remain an essential probe of cosmology with projects such as the Sloan Digital Sky Survey (see the article by Alex Szalay in this issue), which plans to map out the 3D structure of a million galaxies. However, even more so than with clusters of galaxies, the link between dark-matter distribution—a direct prediction of cosmological models—and galaxies is unclear because galaxy formation is not well-understood. Cosmologists cannot yet include all the physical processes necessary to fully model this complicated process. However, we

can simulate a simplified system and ask the more restricted question: Do gas disks with the observed properties of spiral galaxies (mass, size, angular momentum) form in simulations that include gas and radiative cooling but not star formation?

Again, to concentrate the available computational power on the area of interest while including the surrounding material's long-range gravitational effects, we employ the same focusing technique as for galaxy-cluster simulations. Figure 5's first panel clearly demonstrates this; it shows a projection of the gas density through much of the computational box. Most of the region has visibly large grid cells, while a small region, approximately 1% of the total volume, is adaptively refined. To show this refined region more clearly, the remaining frames zoom into a smaller and smaller region, each time magnifying the previous image by a factor of eight.

The many small clumps are regions of very dense cold gas that would probably form stars if allowed. At this point in the evolution—about half-way between the big bang and present day—the cold clumps are arrayed in filamentary structures and are beginning to merge and form larger systems such as the one shown in the final frame. By this time, 10 levels of refinement have been employed; with a refinement factor of two and an initial grid of 64^3, this leads to an effective resolution, in small regions, of $65{,}536^3$ and to a 500-pc cell size.

By the present day, most of these clumps have merged into a rotationally supported disk with many of our own galaxy's characteristics. However, in at least one respect, the disk is not a good match: the disk has a smaller radius, so it is too centrally condensed compared to most spiral galaxies. This seems to be a general problem with the gas disks produced in these sorts of simulations:[7] the final angular momentum is too low, probably because during the collapse, energy has transferred from gas to dark matter through a process called dynamical friction.[8] It is not obvious whether this is a problem with the cosmological model or with the simplified assumption of ignoring stars. One well-known effect of star formation is the generation of a class of very large stars that undergo massive explosions. These supernovae heat the gas and might substantially change the way the system evolves. Including this process is both important and technologically challenging because the resolution required to self-consistently include star formation is even greater than that demonstrated here, and currently unfeasible. Nevertheless, parametric modeling of this crucial effect is possible, and work along these lines is ongoing.

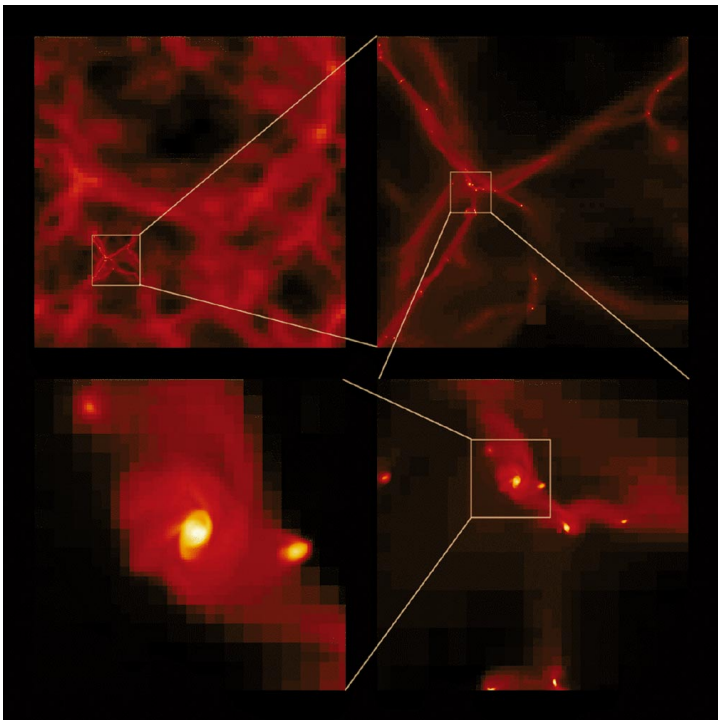


Figure 5. The formation of a disk-like galaxy in an AMR simulation. The upper left panel shows the gas density on the root grid, and each panel in clockwise order shows an increasingly smaller region at higher resolution.

The first stars

Perhaps the most exciting and certainly the newest frontier for cosmological hydrodynamics is one that is currently beyond observational range. Many cosmological models predict that the first objects to collapse do so shortly after the big bang, before the universe is 100 million years old (assuming a current age of 15 billion years). The mass of these objects depends on their ability to cool and is predicted to be approximately a million times the mass of our own sun.

These are the first stars, and their presence drastically changes the universe. Before first-generation star formation, there are essentially no elements heavier than lithium, no ionizing radiation, and no magnetic fields. From a computational standpoint, this makes the simulation a member of that rare class: a well-posed problem. The equations are well-known, there is almost no missing physics, and the initial conditions are clear. Of course, the resulting system is computationally challenging nevertheless.

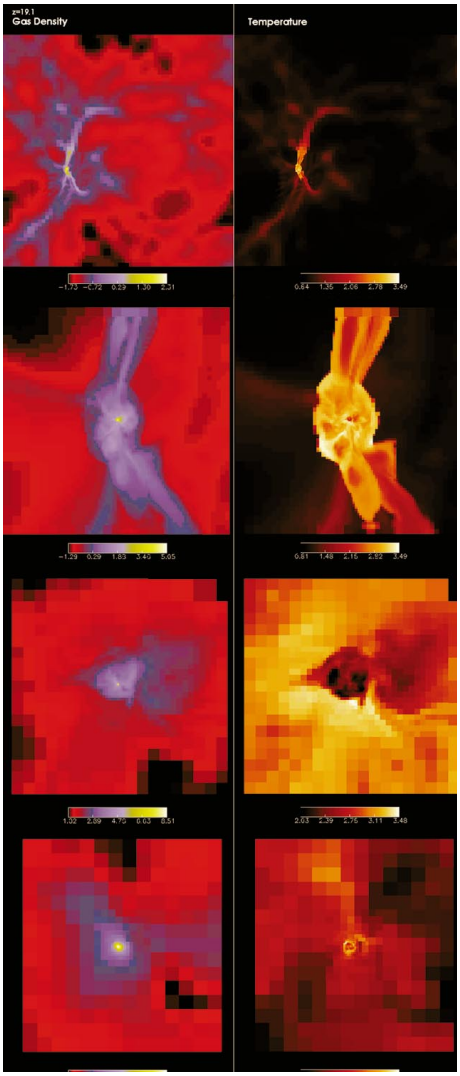


Figure 6. The first stars in current cosmological theories are predicted to form in structures like the one shown here. The four panels show successive factor-of-10 zooms on the central density peak, in gas density on the left and temperature on the right. (Figure courtesy of Tom Abel.)

Because there are no heavy elements—the primary coolants in astrophysical systems today—the main source of cooling is the molecular form of hydrogen (H_2). The formation of H_2 is a slow, nonequilibrium process, so we need to include a chemical reaction network in our simulations that follows nine species (including hydrogen, helium, and molecular hydrogen) with 28 reactions.[9] This adds a set of ordinary differential equations that must be solved at each zone, substantially increasing the computation time.

An outstanding question is the mass of the resulting stars. Although the whole object's size is much larger than our own sun, it is possible, in fact likely, that the entire system will not collapse into a single star but instead will fragment into smaller clumps. The size of these objects is important because their influence on their environment is quite sensitive to the output from these stars: large objects (greater than 10 solar masses) will produce much more radiation and heavier elements than smaller stars.

One of AMR's advantages over a purely Lagrangian method is the flexibility of specifying the refinement criteria. In the previous two simulations, I have primarily used a density-based threshold, which mimics a Lagrangian code. Here I use, in addition, a criterion designed to ensure that all possible fragmentation is resolved. More precisely, this means that the cell size must be kept smaller than a Jean's length (the size of a perturbation that is marginally gravitationally stable).

Once the gas cloud collapses, it begins to synthesize molecular hydrogen and cool. This is a two-step process: there is a first collapse that halts because cooling is relatively inefficient. Only when a sufficient mass of H_2-enriched gas is accumulated does a runaway cooling catastrophe occur, and a small fragment continues to collapse to a very small size and high density.

Figure 6 shows the system just after the second collapse occurs. Again, a zoom is necessary to encompass the range of scales that the problem involves: this time, a thin slice through the center of the object depicts both density and temperature. In the first frame, only the cloud's broad outlines are visible, while the second frame shows the aspherical shocked gas with a small, cool region barely visible in the core. The third frame shows this core, which is the product of the first collapse, in more detail. The small density spike in the center, shown at nearly full resolution in the final frame, is our protostar. At this point in the evolution, even the 12 levels of refinement that AMR provides are insufficient and the small central core collapses to unresolved scales. However, substantial information has been gained: the mass of this object is about 100 solar masses, setting an upper limit to the mass of the first stars.

These three examples show AMR's success story in cosmology. The last example used 12 levels of refinement, or a maximum resolution of more than 10^5. An equivalent single-grid simulation, if it could be accomplished at all, would need 10^{16} zones, or 1,000 Pbytes, and would require runtimes of a thousand years on a petaflop machine. Because this example ran in a few days on four processors of an SGI Origin2000 system, this represents considerable savings.

Still, there is much work left to be done. An important challenge is to include the new physical processes that were unimportant at low resolution. For example, in our first-structure simulation, the primary reason for halting the calculation was not excessive CPU time, but that the density became so high that a neglected chemical reaction had become important.

More seriously, radiation plays a larger and larger role as densities become higher. So far, cosmologists have usually assumed that photons,

once emitted, can either be ignored or treated simply as a homogeneous radiation bath. However, the dense gas clumps that these new simulations produce can block photons, either preventing external radiation fields from penetrating their centers or trapping and reprocesssing light emitted by the gas clouds themselves, or the stars that reside within them. The equations of radiative transfer are 6D, so solving this problem in its entirety is extremely challenging—we will need approximate methods.

There are also new computational issues. Traditional high-performance computing used simple, static data structures—such as multidimensional arrays—that were allocated at the beginning of the run and never freed until the program finished. AMR is one of a new breed of algorithms that can stress the features of modern computational languages. For example, because a large AMR run might involve thousands of grids (which each have to be allocated and deallocated over hundreds of thousands of time steps), memory can become fragmented. Toward the end of a long run, the operating system has trouble finding sufficiently large, contiguous blocks of memory for the larger grids (even though there might be many smaller regions available). The system is forced to allocate more and more memory to the process, and in extreme cases, it can exhaust all available memory.

Another important area of research is the development of a parallel version of AMR that can run on distributed-memory supercomputers. This task is made substantially more challenging by the complicated data structures and interactions of AMR. Other researches (myself included) are exploring two approaches: the first is to add parallelism by hand, with explicit calls to message-passing libraries. While the grid class's encapsulated nature helps this, it is still a fair amount of work. The other—hopefully less labor-intensive—tactic is to use more sophisticated tools, such as high-performance C++, to recognize and exploit parallelism. In either case, it is clear that while new computational techniques such as AMR can help answer questions about the universe's past, they also raise new challenges for future generations of computational scientists. CiSE

Acknowledgments

This work is done under the auspices of the Grand Challenge Cosmology Consortium and supported in part by NSF grant ASC-9318185 and NASA Astrophysics Theory Program grant NAG5-3923. NASA also supported this work through Hubble Fellowship grant HF-01104.01-98A from the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy Inc. under NASA contract NAS5-26555. I thank Michael Norman, Jeremy Kepner, and Tom Abel for their permission to discuss collaborative work. The calculations were performed at the National Center for Supercomputing Applications.

References

1. M.J. Berger and P. Colella, "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *J. Computational Physics*, Vol. 82, No. 1, May 1989, pp. 64–84.
2. D.H. Weinberg, N. Katz, and L. Hernquist, "Simulating Cosmic Structure Formation," *Origins*, Vol. 148, Astronomical Society of the Pacific, San Francisco, 1998, pp. 21–40.
3. G.L. Bryan et al., "A Piecewise Parabolic Method for Cosmological Hydrodynamics," *Computer Physics Comm.*, Vol. 89, No. 1, Dec. 1995, p. 1.
4. M.J. Berger and I. Rigoutsos, "An Algorithm for Point Clustering and Grid Generation," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 21, No. 5, Sept. 1991, pp. 1278–1286.
5. G.L. Bryan and M.L. Norman, "A Hybrid AMR Application for Cosmology and Astrophysics," to appear in *Structured Adaptive Mesh Refinement Methods*, Springer-Verlag, New York, 1999.
6. M.L. Norman and G.L. Bryan, "Cosmological Adaptive Mesh Refinement," to appear in *Numerical Astrophysics 1998*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.
7. J.F. Navarro and M. Steinmetz, "The Effects of a Photoionizing Ultraviolet Background on the Formation of Disk Galaxies," *Astrophysical J.*, Vol. 478, No. 1, Mar. 1997, p. 13.
8. J. Binney and S. Tremain, *Galactic Dynamics*, Princeton Univ. Press, Princeton, N.J., 1987.
9. T. Abel et al., "Modeling Primordial Gas in Numerical Cosmology," *New Astronomy*, Vol. 2, No. 3, Aug. 1997, pp. 181–208.

Greg L. Bryan is a research associate at the Massachusetts Institute of Technology. His research focus is cosmological fluid dynamics and the large-scale structure of the universe, which has lead to an interest in high-performance computation and visualization. After earning a BS in physics from the University of Calgary, he obtained his PhD in astrophysics from the University of Illinois. He is a member of the American Astronomical Society. Contact him at the MIT Physics Dept., 77 Massachusetts Ave., Cambridge, MA 02139; gbryan@mit.edu.