Large-Scale Simulations of Clusters of Galaxies

P. M. Ricker[a], A. C. Calder[a], L. J. Dursi[a], B. Fryxell[a], D. Q. Lamb[a], P. MacNeice[b], K. Olson[a,b], R. Rosner[a], F. X. Timmes[a], J. W. Truran[a], H. M. Tufo[a], and M. Zingale[a]

[a]*ASCI Flash Center, University of Chicago, Chicago, IL 60637*
[b]*NASA/Goddard Space Flight Center, Greenbelt, MD 20771*

Abstract. We discuss some of the computational challenges encountered in simulating the evolution of clusters of galaxies. Eulerian adaptive mesh refinement (AMR) techniques can successfully address these challenges but are currently being used by only a few groups. We describe our publicly available AMR code, FLASH, which uses an object-oriented framework to manage its AMR library, physics modules, and automated verification. We outline the development of the FLASH framework to include collisionless particles, permitting it to be used for cluster simulation.

SIMULATING CLUSTERS

Clusters of galaxies are the largest gravitationally bound objects in the universe. They consist mainly of dark matter and diffuse, hot plasma, with galaxies themselves contributing only a few percent of the total mass. Clusters have attracted attention in recent years because they are large enough to serve as a representative sample of the universe; they provide strong contraints on cosmological models. Clusters are also interesting from an astrophysical point of view. The intracluster medium (ICM), at densities $\sim 10^{-4} - 10^{-2}$ cm^{-3} and temperatures $\sim 10^{7-8}$ K, is collisionally ionized and emits X-rays, primarily via bremsstrahlung [21]. The radiative cooling time can be short enough to produce cooling flows [11]. Observations of Faraday rotation show that the ICM is magnetized, with $B \gtrsim 1$ μG [7]. With the diffuse, nonthermal radio [15] and X-ray [14] emission seen in some clusters, this suggests that clusters are sites for cosmic-ray acceleration [2]. Magnetic fields may also help to suppress diffusion in the ICM [6]. Galaxies orbiting in cluster potentials experience tidal and ram-pressure stripping and help to stir the ICM [22]. Star formation and supernovae may affect the abundances of heavy elements in the ICM as well as its global energetics [10, 17]. Many elements of a complete model for the ICM are known, but we still cannot answer such questions as: what is the source of entropy and metals in the ICM? what happens to the gas that cools below X-ray temperatures in cooling flows? how robust are cooling flows? how is energy partitioned among thermal and nonthermal particle populations, magnetic fields, and turbulent motions?

Cluster mergers play a key role in these phenomena. Mergers strongly affect the ICM, producing long-lived distortions in X-ray images and temperature maps. The energies they release ($\sim 10^{64}$ erg) are easily enough to heat the ICM to 10^8 K. These events are complex, nonlinear, multiphysical, and three-dimensional; thus numerical simulations are appropriate tools for studying them.

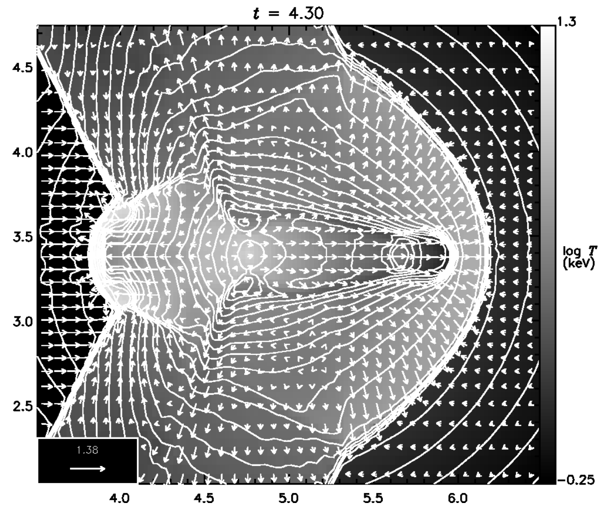

FIGURE 1. Gas density (log contours), temperature (shading), and velocity (arrows) in a merger of clusters with mass ratio 1:3 (Ricker & Sarazin 2000, in prep.). Times and lengths are in Gyr and h^{-1} Mpc, respectively, with Hubble constant $h = 0.6$.

Cluster simulations with multiphysics (at least hydrodynamics in addition to dark matter) have reached $\gtrsim 10^6$ particles or zones, simulating a cluster and its environment down to kpc scales [12, 16]. Cosmological simulations with 10^9 particles, yielding 10^5 clusters, have been performed [8], but thus far these have only included dark matter. To understand recent observations, we must resolve scales needed for cosmological context ($\gtrsim$10 Mpc) and galaxies ($\lesssim$1 kpc) – a dynamic range of $> 10^4$ – with hydrodynamics, cooling, magnetic fields, and nonequilibrium plasma effects. Star formation and supernova feedback will remain unresolved for the forseeable future and must be included phenomenologically.

Because shocks are important in the ICM, Eulerian shock-capturing methods such as the Piecewise-Parabolic Method (PPM) [9] are very desirable. Fig. 1 shows an example merger calculation performed using the COSMOS *N*-body/hydro code [20]. COSMOS uses PPM on a single nonuniform grid. This 3D calculation covered a dynamic range of $\sim$ 300 on 128 processors of the San Diego Cray T3E, requiring 10,000 node-hours. To add new physics and increase dynamic range, the cost of such calculations must be reduced significantly while retaining the shock-capturing properties of single-grid Eulerian schemes.

ADAPTIVE MESH REFINEMENT

The computational issues involved in studying gravitational clustering with multiphysics involve the coupling of small and large scales through gravity and hydrodynamics, requiring large dynamic range, and the presence of short-range source terms that upset load balancing.

Block-structured adaptive mesh refinement (AMR) methods address these issues by placing fine grids only where they are needed to resolve fine features [3]. An example of a freely available AMR package is PARAMESH [18]. PARAMESH manages an octree (in 3D) data structure whose nodes are uniformly gridded meshes (‘blocks’); Fig. 2 shows an example. Each block is a factor of two more refined than its parent. Refined blocks are placed according to user-defined criteria; interpolation is used to obtain their initial and boundary data from coarser blocks. PARAMESH distributes blocks among processors using a work-weighted space-filling curve, keeping spatially adjacent blocks on the same processor when possible and balancing the computational load.

Long-range coupling can be handled in AMR using multilevel relaxation techniques [4]. The coarse-grid solution is obtained on one processor, while finer levels use neighbor-to-neighbor communication.

Short-range forces produce highly clustered distributions of work. The mixed material representations in PPM-based cluster simulations (Lagrangian particles, Eulerian gas) require different domain decompositions. AMR can solve both of these problems by weighting blocks appropriately, e.g., by source terms or particle content. Blocks also can be evolved on different timesteps and weighted inversely by their timestep.

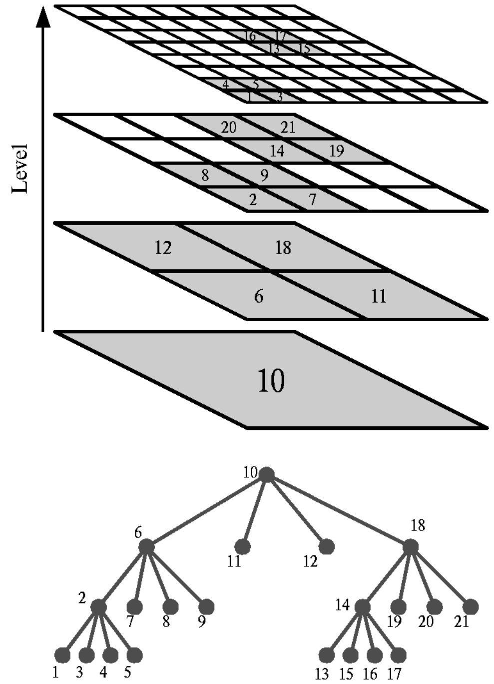

FIGURE 2. Example 2D mesh managed by PARAMESH [18].

AMR techniques are widely used in cosmological *N*-body and smoothed-particle hydrodynamics codes, but few groups have used them with Eulerian schemes [19]. AMR codes are difficult to construct, and most cosmological codes are proprietary. However, during the coming year we expect to see several AMR codes useful for cosmology emerge, some of which are freely available.

THE FLASH CODE

We are developing FLASH, an adaptive-mesh astrophysical simulation code based on PARAMESH [23, 13]. FLASH is coded mainly in Fortran 90 and uses the Message-Passing Interface (MPI). It is highly portable and scales to thousands of processors. We have recently been awarded the Gordon Bell Prize for achieving 0.24 TFlops with FLASH using 6,420 processors of ASCI Red on a cellular detonation problem relevant to Type Ia supernovae [5]. We intend for FLASH to evolve into a community simulation framework; the code is publicly available at `http://flash.uchicago.edu/`.

Many astrophysical problems require multiple physical processes and a wide range of scales. Each physical process requires a different numerical method and different tests. Exploiting AMR also requires complicated mesh management libraries. Such complex software is best managed using a framework.

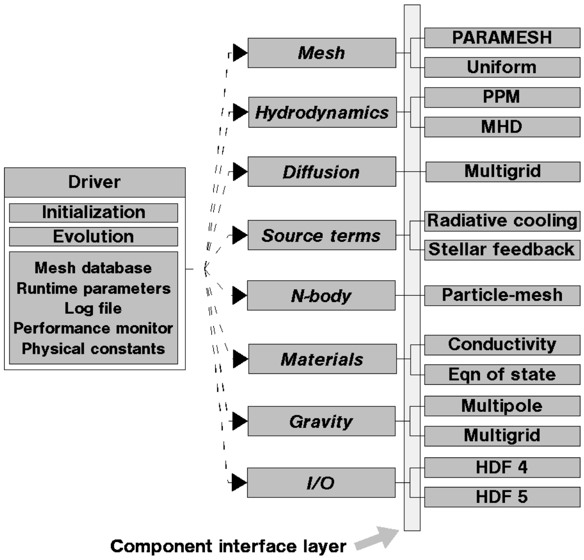

FIGURE 3. The framework of the FLASH code [13], showing components useful for cluster simulation.

Object-oriented languages provide several features useful for building simulation frameworks. Encapsulation allows us to interchange solvers that need conflicting internal data structures; inheritance allows us to abstract common features of different types of solvers; and polymorphism allows us to switch between solvers.

Component frameworks scale better with increasing complexity by providing standard ways for components to describe themselves to each other. Such frameworks are commonly used in business, but they have not seen wide use in science, because they impose unacceptable overhead and lack features needed for scientific applications. An appropriate scientific component standard, such as that being developed by the Common Component Architecture (CCA) Forum [1], is still several years away.

The FLASH framework is object-oriented and makes use of some component ideas. Its class structure appears in Fig. 3. The driver maintains mesh data in a static container class and instantiates objects from various classes of physics solvers. The solvers are divided into different classes by their level of coupling and by differences in solution method (e.g., hyperbolic solvers for hydrodynamics, elliptic solvers for radiation and gravity). The AMR library is also treated as a class. Solver and mesh objects access mesh data through methods supplied by the mesh container class. The component interface layer, for which we are developing a standard, will consist of F90 module wrappers implementing an interface that is abstractly specified in an interface definition language (IDL).

FLASH includes hydrodynamics using PPM, self-gravity using multigrid and multipole methods, and modules appropriate for supernova problems, including a partially degenerate equation of state and nuclear reaction networks. Modules for front tracking, implicit diffusion, magnetohydrodynamics, and collisionless particles are under active development by our group. With these new components FLASH will be capable of simulating individual clusters with multiphysics and a dynamic range of $\gtrsim 2,000$ per dimension during the coming year.

This work was supported by DOE under Grant No. B341495 to the ASCI Flash Center at the University of Chicago. Calculations were performed using the resources of the San Diego Supercomputer Center.

REFERENCES

1. Armstrong, R., et al., Argonne Natl. Lab. MCS preprint P759-0699, 1999
2. Berezinsky, V. S., Blasi, P., & Ptuskin, V. S., *Astroph. J.* **487**, 529-535 (1997)
3. Berger, M. J., & Oliger, J., *J. Comput. Phys.* **53**, 484-512 (1984)
4. Briggs, W. L., Henson, V. E., & McCormick, S. F., *A Multigrid Tutorial*, 2d ed., SIAM, Philadelphia, 2000
5. Calder, A. C., et al., in *Proc. Supercomputing 2000*
6. Chandran, B. D. G., et al., *Astroph. J.* **525**, 638-650 (1999)
7. Clarke, T. E., Kronberg, P. P., & Böhringer, H. B., in *Cluster Mergers and their Connection to Radio Sources*, 24th IAU, Joint Discussion 10, 2000
8. Colberg, J. M., et al., *Mon. Not. R. Astron. Soc.* **319**, 209 (2000)
9. Colella, P., & Woodward, P. R., *J. Comput. Phys.* **54**, 174-201 (1984)
10. Dupke, R. A., & White, R. E., *Astroph. J.* **537**, 123-133 (2000)
11. Fabian, A. C., *Ann. Rev. Astr. Ap.* **32**, 277-318 (1994)
12. Frenk, C. S., et al., *Astroph. J.* **525**, 554-582 (1999)
13. Fryxell, B., et al., *Astroph. J. Suppl.* in press (2000)
14. Fusco-Femiano, R., et al., *Astroph. J. Lett.* **513**, 21-24 (1999)
15. Kempner, J. C., & Sarazin, C. L., *Astroph. J.* accepted (2000)
16. Lewis, G. F., et al., *Astroph. J.* **536**, 623-644 (2000)
17. Lloyd-Davies, E. J., Ponman, T. J., & Cannon, D. B., *Mon. Not. R. Astron. Soc.* **315**, 689-702 (2000)
18. MacNeice, P., et al., *Comput. Phys. Comm.* **126**, 330-354 (2000)
19. Norman, M. L., & Bryan, G. L., in *Numerical Astrophysics*, eds. S. M. Miyama et al., Kluwer, Boston, 1999, p. 19
20. Ricker, P. M., Dodelson, S., & Lamb, D. Q., *Astroph. J.* **536**, 122-143 (2000)
21. Sarazin, C. L., *X-Ray Emission from Clusters of Galaxies*, Cambridge U. P., Cambridge, 1988
22. Stevens, I. R., Acreman, D. M., & Ponman, T. J., *Mon. Not. R. Astron. Soc.* **310**, 663-676 (1999)
23. Rosner, R., et al., *Comput. Sci. Eng.* **2**, 33-41 (2000)