High Performance Computing In Astrophysics: Parallel Gasdynamics And Gasoline

James Wadsley[a]

[a]Physics and Astronomy & SHARCNET, McMaster University, Ontario, Canada
wadsley@mcmaster.ca

High Performance Computing is steadily increasing in importance for theoretical and observational astronomy. Largely through CFI, Canadian astrophysicists have unprecedented access to near world-class facilities. We review the contributions of computational astrophysics as a leading-edge player in the development of highly parallel applications and scalable algorithms with a particular focus on self-gravitating gasdynamical simulations. We present Gasoline, our portable parallel tree code for gasdynamical simulations. We review current applications in theoretical astrophysics and chart the future development of the field in the direction of increasing physical complexity and larger dynamic ranges in time and space.

L'importance du calcul de haute performance en astronomie d'observation et en astronomie théorique croît constamment. En bonne partie grâce à la FCI, les astrophysiciens canadiens ont un accès sans précédent à une infrastructure de calcul de classe mondiale. Nous présentons une revue de la contribution de l'astrophysique numérique au développement d'applications et d'algorithmes parallèles, en insistant sur les simulations de gaz gravitationnels. Nous présentons Gasoline*, notre code portable parallèle de dynamique gravitationnelle des gaz. Nous décrivons des applications courantes à l'astrophysique théorique et dressons un plan du développement futur de ce domaine vers une complexité physique croissante et des échelles de temps et d'espace plus vastes.*

Introduction

Astronomy and astrophysics are very computer intensive. With the widespread adoption of the CCD camera, even traditional observational astronomy has become dependent on computer image processing. New wide-area surveys will involve enormous data rates. For example, the upcoming Atacama Large Millimeter Array[1] will require routine processing of three-dimensional imaging datasets arriving at rates of 1 TB per day. The Cosmic Microwave Background (CMB) has embedded fluctuations that ultimately lead to all structure in the present universe. High Performance Computing is essential to manipulate CMB data coming from new satellite experiments such as PLANCK[3], where a full analysis would require 10^{24} FLOP.

Theoretical astrophysics has always been at the forefront of technology, with N-body simulations predating the use of digital computers, such the use of light bulbs and light intensity measurements as an analog of gravity for manual simulations of a many-body self-gravitating system [2]. Astrophysical objects including planets, individual stars, interstellar clouds, star clusters, galaxies, accretion disks, clusters of galaxies through to large scale structure have all the been the subject of numerical investigations. The most challenging extreme is probably the evolution of space time itself in computational general relativistic simulations of colliding neutron stars and black holes. Since the advent of digital computers, improvements in storage and processing power have dramatically increased the scale of achievable simulations. This, in turn, has driven remarkable progress in algorithm development. Increasing problem sizes have forced simulators who were once content with $\mathcal{O}(N^2)$ algorithms to pursue more complex $\mathcal{O}(N)$ algorithms and adaptivity in space and time.

In this paper we begin by tracing past progress in computational astrophysics with a particular focus on self-gravitating systems (N-body) and the inclusion of hydrodynamics. Historically, many codes began as pure N-body solvers and later had gasdynamics and other physics added. The Gasoline parallel gasdynamics code discussed here evolved from the Pkdgrav N-body code. We describe aspects of hydrodynamics in astrophysical systems to motivate Smoothed Particle Hydrodynamics as our choice of method. We describe the current implementation in Gasoline in the context of parallel gravity plus gasdynamics codes generally. A more detailed technical discussion of the implementation is available elsewhere[33,42]. We explore how the need to perform higher resolution studies and to include additional physics is driving the development of algorithms. We give specific examples of how Gasoline is being adapted to future needs. We present examples of state-of-the-art simulations performed with Gasoline.

1 Simulating Self-Gravity

Gravity is the key driving force in most astrophysical systems. With assumptions of axisymmetry or perturbative approaches an impressive amount of progress has been made with analytical methods, particularly in the areas of solar system dynamics, stability of disks, stellar dynamics and quasi-linear aspects of the growth of large scale structure in the universe. In many systems of interest, however, non-linear interactions play a vital role. This ultimately requires the use of self-gravitating N-body simulations. Even in the apparently regular solar system numerical simulations have revealed the fundamentally chaotic nature of orbits and the long-term stability of the solar system is no longer taken for granted[5].

Fundamentally, solving gravity means solving Poisson's equation for the gravitational potential, ϕ, given a mass density, ρ: $\nabla^2\phi = 4\pi G\rho$ where G is the Newtonian gravitational constant. In a simulations with discrete bodies it is common to start from the explicit expression for the acceleration, $a_i = \nabla\phi$ on a given body in terms of the sum of the influence of all other bodies, $a_i = \sum_{i \neq j} GM_j/(r_i - r_j)^2$ where the r_i and M_i are the position and masses of the bodies respectively. In practical work it is essential to soften the gravitational force on some scale $r < \epsilon$ to avoid problems with the integration and to minimize two-body scattering as each simulated body represents many bodies in an effectively collisionless system.

Early N-body work such as studies of relatively small stellar systems were approached using a direct summation of the forces on each body due to every other body in the system[4]. This direct $\mathcal{O}(N^2)$ approach is impractical for large numbers of bodies, N, but has enjoyed a revival due to high throughput special purpose hardware such as GRAPE[6]. The GRAPE hardware performs the mutual force calculation for sets of bodies entirely in hardware and remains competitive with other methods on more standard floating hardware up to $N \sim 100,000$.

A popular alternative scheme is the Particle-Mesh (PM) method which has long been used in electrostatics and plasma physics. The adoption of PM was strongly related to the realization of the existence of $\mathcal{O}(N\,log\,N)$ Fast Fourier Transform (FFT) in the 1960's. The FFT is used to solve for the gravitational potential from the density distribution interpolated onto a regular mesh. In astrophysics sub-mesh resolution is often desired in which case the force can be corrected with local direct sums as in the Particle-Particle Particle-Mesh (P^{3}M) method. PM is popular in stellar disk dynamics and P^{3}M has seen widespread adoption in Cosmology[7]. PM is similar to Multigrid[13] and other iterative schemes. However, working in Fourier space is not only more efficient, but it also allows force error control through optimization of the Green's Function and smoothing. Fourier Methods are widely recognized as ideal for large, fairly homogeneous, periodic simulations. Multigrid has some advantages in parallel due to the local nature of the iterations. The Particle-Particle correction can get expensive when particles cluster in a few cells. Both Multigrid[8,9] and P^{3}M[10] can adapt to address this via a hierachy of submeshes. With this approach the slow down due to heavy mass clustering tends towards a fixed multiple of the unclustered run speed.

In applications such as galactic dynamics where high resolution in phase space is desirable and particle noise is problematic, the smoothed gravitational potentials provided by an expansion in modes is useful. PM does this with Fourier modes however a more elegant approach is the Self-Consistent Field method[11,12] (SCF). Using a basis set closely matched to the evolving system dramatically reduces the number of modes to be modeled however the system must remain close to axi-symmetric and similar to the basis. SCF parallelizes well and is also used to generate initial conditions such as individual galaxies within merging simulations.

A current popular choice is to use tree algorithms which are inherently $\mathcal{O}(NlogN)$. This approach recognises that details of the local mass distribution become less important for accurate gravity with increasing distance. Thus the remote mass distribution can be expanded in multipoles on the different size scales set by a tree-node hierachy. The appropriate scale to use is set by the *opening angle* subtended by the tree-node bounds relative to the point where the force is being calculated. The original Barnes-Hut[14] method employed oct-trees but this is not essential[15]. The tree approach can adapt to any topology and thus the speed of the method is somewhat insensitive to the degree of clustering. Once a tree is built it can also be re-used as an efficient search method for other physics such as particle based hydrodynamics.

A particularly useful property of tree-codes is the ability to efficiently calculate forces for a subset of the bodies. This is critical if there is a large range of time-scales and multiple independent timesteps are employed. At the cost of force calculations no longer being synchronized among the particles substantial gains in time-to-solution may be realized. Multiple timesteps are particularly important for current astrophysical applications where the interest and thus resolution tends to be focused on small regions within large simulated environments such as individual galaxies, stars or planets. Dynamical times can become very short for small numbers of particles. P^3M codes are faster for full force calculations but are difficult to adapt to calculate a subset of the forces

In order to treat periodic boundaries it is necessary to effectively infinitely replicate the simulation volume which may be approximated with an Ewald summation[16]. An efficient alternative which is seeing increasing use is to use trees in place of the direct Particle-Particle correction to a Particle-Mesh code, often called Tree-PM[17–19].

The Fast Multipole Method (FMM) recognizes that the applied force as well as the mass distribution may be expanded in multipoles. This leads to a force calculation step that is $\mathcal{O}(N)$ as each tree node interacts with a similar number of nodes independent of N. Building the tree is still $\mathcal{O}(NlogN)$ but this is a small cost for simulations up to $N \sim 10^7$[22]. The Greengard-Rohklin[20] method used spherical harmonic expansions where the desired accuracy is achieved solely by changing the order of the expansions. For the majority of astrophysical applications the allowable force accuracies make it much more efficient to use fixed order cartesian expansions and an opening angle criterion similar to standard tree codes[21,22]. This approach has the nice property of explicitly conserving momentum (as do PM and P^{3}M codes). The prefactor to cartesian FMM is quite small so that it can outperform tree codes even for small N[22]. It is a significantly more complex algorithm to implement, particularly in parallel. One key obstacle to its widespread adoption is the fact that the speed benefit over a tree-code is much reduced when small subsets of the particles are having forces calculated (i.e. for multiple timesteps).

1.1 Pkdgrav

Pkdgrav is a state-of-the art parallel tree-code designed by Joachim Stadel[23] and coded with Thomas Quinn. The code was designed to very portable, using a compact machine dependent library (MDL) to encapsulate the architecture-dependent communication aspects. Ports exists for specialized shared memory (e.g. T3E), SMP (threads) and MPI (many architectures). MDL uses a scalable on-request system to acquire remote particle and tree-node data and maintains a local cache for efficiency. The overall design is modular and additional physics such as collisions between bodies (with Derek Richardson) and gasdynamics (with James Wadsley to create the Gasoline code) have been incorporated easily. Pkdgrav has been applied to a large range of astrophysical problems ranging from solar system formation to the evolution of large scale structure in the universe.

Some key features of Pkdgrav are 1) a binary tree for both domain decomposition and gravity calculations; 2) multipole expansions to hexadecapole order; 3) local caching for parallel gravity; and 4) Multiple timesteps. These features combine to make the code exceptionally flexible in handling large dynamic ranges in space and time.

In parallel the simulation volume is decomposed through recursive splits so that each processor ends up with a comparable workload and a box-shaped domain. The splitting procedure does not place any restrictions on the number of processors. It is efficient for relatively homogeneous simulations, even if the local level of clustering is high. In each processor domain a local binary tree is formed by recursive binary splits of the largest axis and the bounds are continuously shrunk to exactly enclose the particles. Thus the code adapts to flat two-dimensional topologies such as disks and the tree depth is controlled.

Hexadecapole (4th order) expansions of the mass distribution result in higher accuracy for a given value of the opening angle and thus less node interactions for a set level of accuracy than more typical quadrupole tree-codes. In parallel this reduces the communication overheads.

Multiple timesteps have been implemented using nested hierachy of time step *rungs* where each substep is performed with a Kick-Drift-Kick (KDK) operation[24]. This scheme would be symplectic in the absence of particles changing rungs. With KDK force calculations are synchronized across many rungs and the integration is more accurate than the similar Drift-Kick-Drift scheme. Multiple timesteps and the standard tree-code approximation to the forces prevent explicit momentum conservation however the conservativeness of the timestep criteria are more important for accurate integrations.

2 Computational Fluid dynamics in Astrophysics

Astrophysical systems are in predominantly at very low particle densities and experience wide-ranging temperature variations. Most of the material is in a highly compressible gaseous phase. In general this means that a perfect gas is an excellent approximation. Direct physical processes such as shear viscosity and diffusion can usually be neglected. High-energy processes and the action of gravity tend to create large velocities so that flows are both turbulent and supersonic: strong shocks and very high Mach numbers are common. Radiative cooling processes can be important, however the timescales can often be much longer or shorter than dynamical timescales. In the latter case isothermal gas is often assumed for simplicity. In many areas of computational astrophysics, particularly cosmology, gravity tends to be the dominant force driving the evolution of the system. Visible matter, often in the form of radiating gas, provides the essential link to observations. Radiative transfer is always present but may not significantly affect the energy and thus pressure of the gas during the simulation. Table 1 lists areas commonly explored with self-gravitating gas simulations and indicates the importance of physical processes to recover the basic observed properties. To match observations in detail involves more physics than is often practical. For example, a detailed model of solar system formation would include dust grains and full radiative transfer. Recent advances in computer power and algorithms are bringing real-time radiative transfer within reach of simulators[25].

Table 1 is also a guide to the range of physical scales present in representative simulations in astrophysics. The dynamical timescale for gas scales as $\mathrm{density}^{-1/3}\,\mathrm{temperature}^{-1/2}$ and for gravity it scales as $\mathrm{density}^{-1/2}$. For adiabatic temperature variations the gas dynamical time scales as $\mathrm{density}^{-2/3}$. With gas cooling (or the isothermal assumption) simulations can achieve very high gas densities. These two factors mean that gas sets the shortest dynamical timescales and that gas simulations are much more demanding (many more steps to completion) than corresponding gravity only simulations.

2.1 Methods

Fluid dynamical methods can be broadly classified into Eulerian or Lagrangian methods. Eulerian methods use a fixed computational mesh through which the fluid flows via explicit advection terms. Regular meshes provide for cleaner analysis and thus high order methods such as PPM[26] and TVD schemes[27,28] have been developed. The inner loops of mesh methods can often be pipelined for high performance. Lagrangian methods follow the evolution of fluid parcels via the full or comoving derivatives. This requires a deforming mesh or a meshless method such as Smoothed Particle Hydrodynamics[29] (SPH). Data management is more complex in these methods however advection is handled implicitly and the resolution elements naturally adapt to follow density contrasts.

Large variations in physical length scales in astrophysics have limited the usefulness of Eulerian grid codes. Adaptive Mesh Refinement[8,30] (AMR) overcomes this at the cost of data management overheads and increased code complexity. In the cosmological context there is the added complication of dark matter, which must be modelled with particles. There is more dark matter than gas in the universe so it dominates the gravitational potential. Perturbations present on all scales in the dark matter guide the formation of gaseous structures including galaxies and the first stars. A fundamental limit to AMR in computational cosmology is matching the AMR resolution to the underlying dark matter resolution. Particle based hydrodynamics (e.g. SPH) is well matched to this constraint. A useful feature of Lagrangian simulations is that bulk flows (which can be highly supersonic in the simulation frame) do not limit the timesteps. Particle methods are well suited to rapidly rotating systems such as astrophysical disks where arbitrarily many rotation periods may have to be simulated (e.g. SPH explicitly conserves angular momentum). A key concern for all methods is correct angular momentum transport.

2.2 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics is a mesh-less particle method. It has been applied to astrophysical problems since it was first conceived[31,32]. It relies on a spatially second-order smoothing interpolation procedure over nearby neighbours to determine fluid forces on each particle. The SPH pairwise forces may be symmetrized resulting in exact energy, momentum and angular momentum conservation to the accuracy of the integration scheme. There are several effective schemes for converting the Navier-Stokes fluid flow equations to SPH interpolation formulas. SPH can be formulated in terms of energy or entropy integration[34]. We implemented the standard asymmetric energy form in Gasoline[42,33]. SPH velocities (and often pressure) are needed for the SPH force calculation. This tends to favour second order predictor-corrector time-integration schemes. The timesteps are limited by the Courant condition and often an acceleration based limiter. Ongoing issues with SPH include: the effects of artificial viscosity for shock capturing on angular momentum transport; and energy integration, particularly in strong cooling regimes.

3 Parallel SPH-Gravity codes

The components of parallel SPH plus Gravity codes tends to be as follows: a spatial domain decomposition; a softened gravity solver tuned for a force accuracy of around 1%; a wide variety of SPH formulations making detailed code comparisons difficult; implicit radiative cooling on a per-particle basis; and second order time-integration with multiple-time steps where practical.

Codes established in the literature can be separated on the basis of the gravity method into mesh (P^3M) codes (e.g. HYDRA[35,36], P^{3}MSPH[37]) and Tree-like codes (e.g. TreeSPH[38,39], GADGET[40], Hashed-Tree[21], GRAPESPH[41] and Gasoline). The key trade-off in this separation is the raw speed of the gravity solution versus the ease with which multi-stepping can be applied. Tree based codes are significantly slower on bulk gravity calculations: timesteps involving most of the particles tend to be entirely dominated by the gravity calculation. Any method with individual particle gravity force calculations such as GRAPE, direct summation or tree-codes may use multiple timesteps and make gravity solution costs much smaller for reduced numbers of active particles. For P^3M codes the FFT is a large fixed cost and it is harder to realize benefits from multiple-timesteps.

Typical applications such as those shown in table 1 tend to have a distribution of timesteps with the bulk of the simulation on larger timesteps and a long tail going to the smallest timesteps (Gasoline simulations have had a range of 2^{18} in timesteps). This situation is ideal for multiple-timesteps. Maximizing the throughput from multiple timesteps requires that operations scale as the number of active particles, N_{active}. A typical gravity tree-walk dominates code time in single-stepping but scales as $\mathcal{O}(N_{active} logN)$ and becomes less important when N_{active} is small. Tree-building from scratch is $\mathcal{O}(NlogN)$. Historically tree building has been a relatively small cost (roughly 1% of the gravity calculation time for single stepping with Gasoline) however as the range of simulated dynamical time expands new developments must consider tree-repair operations that partly reconstruct the tree at $\mathcal{O}(N_{active} logN)$ cost or extrapolate from old tree node data. With re-use of old trees overlapping node boundaries will result in slightly more expensive tree operations. For gravity, care must also be taken when dealing with old multipole moment expansions so that the accuracy of the time integration is not compromised. Tree repair is a successful feature of some current codes[40].

An additional cost in parallel is the domain decomposition. A popular and obvious choice is large rectangular domains or straightforward aggregates of rectangular domains. A more complex alternative is the partitioning of a space filling curve that threads a full oct-tree of the particles[21]. The common principle is always to divide up the work evenly to achieve load balance. The constraint is to have enough memory to hold all the assigned particles. Work load balance usually implies evenly distributed active particles however spatial partitions can result in very uneven distributions of the inactive particles. A domain decomposition that moves away from localized or simply connected domains to attempt to be better balanced for a wider range of time-step rungs will incur larger communication overheads during the force calculations. This idea is yet to be explored in depth for astrophysical particle codes. The FLASH[8] parallel AMR code makes no attempt to keep domains connected. The associated cost is a multiplication of the simulation time and the parallel scaling is otherwise unaffected. Full domain decompositions are a significant cost due to the large communication requirements. For tree codes the cost is much smaller than solving gravity if N_{active} is a significant fraction of N. A common practice is to re-use old decompositions for small N_{active} however this tends to create load imbalance.

The increasing availability of cheap SMP systems makes the idea of domains per node instead of per processor appealing. This allows any problem with the domain decompositon and load balancing to be postponed to higher processor numbers. An idea related to per-node domains is per-processor caching of remote data so that multiple processes can amortize the overhead of the original communication.

SPH is an entirely local operation so, in principle, it is well adapted to multiple timesteps and parallel processing. The non-local data aspects can be addressed through a cache that reports back to the original particle owner (combiner cache) or one-sided force evaluations which may require some duplication of effort but a single communication of data per particle.

In a multistepping SPH calculation as N_{active} decreases, the number of inactive neighbours tends to increase so that the work involved in the SPH calculation decreases more slowly than $\mathcal{O}(N_{active})$. The limiting case is that every neighbour is inactive. Tree-searching operations scale as $\mathcal{O}(N_{active} logN)$ and are thus well suited for this regime. The original SPH formulation calls for repeated operations on neighbours for a single force calculation. In practice, most SPH quantities can be predicted or extrapolated so that only one neighbour interaction is absolutely required per step[35]. Following this route has marginal impacts on accuracy but no systematic problems have been reported and the cost of the SPH calculation can be reduced by a large factor.

3.1 Gasoline: Current status and Future developments

Gasoline is the result of the addition of gas dynamics to pkdgrav using the SPH method. It is a robust code and performs well on comprehensive sets of test problems[33,42]. Gasoline has been applied to a wide range of problems in astrophysics and some recent examples are shown in section 3.2. Its efficient modular parallel design, finely tuned algorithms and the resulting ability to deal efficiently with large dynamic ranges in space and time make Gasoline one of the highest performing codes used in astrophysics.

Gasoline, like Pkdgrav, achieves a big part of its throughput (faster time to solution) through multiple timesteps. Large portions of the code scale in proportion to N_{active} to facilitate this, including the gravity walk and the SPH. In parallel there are some extra costs associated with caching that are discussed below. As the dynamic range in time of our target simulations increases the cost of the $\mathcal{O}(N, log, N)$ tree-build is becoming an issue. Current development ideas focus on repairing the trees as discussed above.

Gasoline relies upon efficient tree-based search operations to identify particle neighbours for SPH and a combiner cache. Each particle interacts with the nearest N_s particles to itself (gather neighbours) plus any particle for which it numbers among that particle's nearest N_s (scatter neighbours). With multiple timesteps the scatter neighbours may not be actively calculating a new acceleration and therefore we perform an inverse k-th nearest neighbour search explicitly using interaction distances stored on tree-nodes to avoid the $\mathcal{O}(N)$ work of an SPH force evaluation over all particles.

It is standard practice to implicitly integrate the energy equations so as to avoid limiting the dynamical timesteps to stable cooling integration timescales. With multiple timesteps we need intermediate values of the pressures for neighbour particles. We do this by interpolating after taking a large implicit (predictor) step. A side-effect of the multistepping is the exact conservation properties of SPH are lost however our timesteps are sufficiently conservative that there are no significant differences between multistepping and single-stepping runs. This makes the integration costs scale as $\mathcal{O}(N_{active})$

In parallel the MDL layer hides the latency of fetching off-processor particle information using a cache. For large numbers of active particles the cost of filling the cache is hidden. For small numbers the early cache misses dominate and prevent the work decreasing in proportion to N_{active}. Techniques such as bulk cache refreshes based on previous fetching histories or re-use of old cached data will be used to address this.

For homogeneous simulations Gasoline scales very well in parallel. For example, a single gravity step on 96 nodes of the McMaster AlphaServer (Idra) scales with 80% efficiency from a single node step for a large cosmological volume. The issue of scaling for less homogeneous simulations (such as renormalized galaxy formation simulations) relates mainly to load balancing in the face of multiple timesteps. The issue is not whether a good decomposition can be found for $N_{active} << N$, but that the cost of doing so is large compared to the rest of the work required. The significant load imbalance related to re-using an old decomposition may not result in a large change to the *overall* work balance once all costs in the code scale as N_{active} *if* N_{active} is sufficiently small for these cases. This problem is the subject of active research and development by the code authors.

3.2 Gasoline Applications

Gasoline has recently been used to provide new insight into clusters of galaxies, galaxy formation, gas-giant planet formation and large scale structure.

Galaxy clusters contain hot gas with temperatures of up to $10^8 K$ which emits X-rays. Gasoline simulations[43] have demonstrated that shock-heating during infall is insufficient to explain the temperature profiles implied by X-ray observations as shown in figure 1.

Galaxy formation is a particularly challenging subject for numerical work. Gasoline has been used to generate state-of-the-art realization of disk galaxies (see figure 2) with properties approaching those of observed galaxies[45].

Some commonly held theories of gas-giant planet formation relied on slow forming rocky cores as a seed for forming planets such as Jupiter. Gasoline simulations[44] shown in figure 3 have demonstrated that self-gravitating gaseous disks around new stars can form gas-giants on timescales of hundreds of years.

Cosmic Microwave Background (CMB) experiments are increasing in sensitivity and angular resolution. In this regime the small perturbations to the temperature of the CMB via the Sunyaev-Zeldovich effect due to gas in and around clusters of galaxies are critically important if the primordial CMB signal is to be recovered. We have simulated these effects with the largest cosmological SPH simulation to date[46] (270 million particles). A slice of the simulation is shown in figure 4.

Common themes in these simulations are large dynamic ranges in space and time. This is the cutting edge for gasdynamical simulations of self-gravitating astrophysical systems. The challenge to simulators is to adapt techniques developed only recently for large scale homogeneous simulations to this challenging new regime.

References

1. Atacama Large Millimeter Array : `http://www.alma.nrao.edu/`
2. Homberg, Erik 1947, *Astrophysical Journal*, 94, 385
3. PLANCK surveyor satellite `http://astro.estec.esa.nl-/SA-general/Projects/Planck/`
4. Aarseth, S., Lecar, M. 1975, ARAA, 13, 1
5. Sussman, G.J. and Wisdom, J. 1992, Science, vol. 257, 56
6. Hut, P. and Makino, J. 1999, Science,283, 501
7. Efstathiou, G., Davis, M., Frenk, C.S., White, S.D.M. 1985, ApJS, 57, 241
8. FLASH AMR Code, ASCI, `http://flash.uchicago.edu`
9. Kravtsov, A.V., Klypin, A.A. and Khokhlov, A.M.J. 1997, ApJS, 111, 73
10. Couchman, H.M.P. 1991, ApJ, 368, 23
11. Hernquist, L. and Ostriker, J. 1992, ApJ, 386, 375
12. Weinberg, M., 1999, Astron. J., 117, 629
13. Press, W.H., Teukolsky, S.A., Vetterling W.T., Flannery, B.P. 1995, Numerical Recipes in C, Cambridge University Press
14. Barnes, J., and Hut, P. 1986, *Nature*, 324:446
15. Jernigan, J.G. and Porter, D.H. 1989, ApJS, 71, 871
16. Hernquist, L., Bouchet, F., and Suto, Y. 1991, ApJS, **75**, 231–240.
17. Wadsley, J.W., 1998, Ph.D. Thesis, University of Toronto
18. Bode, P., Ostriker, J.P., Xu G., 2000 ApJS, 128, 561
19. Bagla, J.S., 1999, preprint astro-ph/9710043.
20. Greengard, L. and Rokhlin, V. 1987, J. Comput. Phys., 73, 325
21. Salmon, J. and Warren, M. 1994, J. Comput. Phys., 111, 136
22. Dehnen, W. 2000, ApJ, 536, 39
23. Stadel, J. 2001, Ph.D. Thesis, University of Washington
24. Quinn, T., Katz, N., Stadel, J., and Lake, G. 1997, preprint, astro-ph/9710043.
25. Razoumov, A, Norman, M., Abel, T., Scott, D. 2002, ApJ, 572, 695
26. Woodward, P. & Collela, P. 1984, J. Comput. Phys, 54, 115
27. Harten, A., Enquist, B., Osher, S. and Charkravarthy, S., 1987, J. Comput. Phys., 71, 231-303
28. Kurganov, A. and Tadmor, E, 2000, J. Comput. Phys., 160, 241
29. Monaghan, J.J. 1992, ARAA, 30, 543
30. Bryan, G. & Norman, M. 1997, Proc. 12th Kingston Conference, Halifax, Oct. 1996 (PASP), 363
31. Gingold, R.A. & Monaghan, J.J. 1977, MNRAS, 181, 375
32. Lucy, L. 1977, AJ, 82, 1013
33. Stadel, J., Wadsley, J. and Richardson, D.C. 2000, HPCS 2000, Kluwer Academic Publishers
34. Springel, V. and Hernquist, L. 2001, preprint astro-ph/0111016
35. Couchman, H. M. P., Thomas, P. A., Pearce, F. R. 1995, ApJ, 452, 797
36. Thacker *et al*, “Hydra MPI”, *in preparation*
37. Evrard, A. 1988, MNRAS, 235, 911
38. Hernquist, L. & Katz, N. 1989, ApJS, 70, 419
39. Dave, R., Dubinski, J., Hernquist, L. 1997, New Astronomy, 2, 277
40. Springel, V., Yoshida, N., White, S. D. M. 2001, New Astronomy, 6, 79
41. Steinmetz, M. 1996, MNRAS, 278, 1005
42. Wadsley, J., Stadel, J. and Quinn, T., *in preparation*
43. Borgani, S., Governato, F., Wadsley, J., Menci, N., Tozzi, P., Quinn, T., Stadel, J., Lake, G. 2002, MNRAS, 336, 409
44. Mayer, L., Quinn, T., Wadsley, J, Stadel, J. 2002, Science, 298, 1756
45. Governato, F., Mayer, L., Wadsley, J., Gardner, J.P., Willman, B., Hayashi, E., Quinn, T., Stadel, J., Lake, G. 2002, preprint astro-ph/0207044
46. Contaldi, *et al*, Proc. XVIII IAP Colloquium ‘On the nature of dark energy’, Paris, 1-5 July 2002, astro-ph/0210303

Application	Gas	Density Range, (cm^{-3})	Temperature Range (K)	Additional physics
Cosmological Large Scale structure	Tracer	$10^{-7} - 10^{-1}$	$1000 - 10^8$	-
Clusters of Galaxies	Secondary	$10^{-7} - 1$	$1000 - 10^8$	-
Galaxy Formation	Primary	$10^{-7} - 100$	$100 - 10^7$	Radiative Cooling, Star Formation
Interstellar Medium	Primary	$10^{-2} - 10^5$	$10 - 10^6$	Magnetic fields?
Accretion disks	Primary	$10^5 - 10^{13}$	$100 - 10^6$	Magnetic fields
Planet Formation	Primary	$10^5 - 10^{18}$	$10 - 10^4$	Aggregating Solids?

Table 1
Gas Applications and minimum physics requirements. The density and temperatures ranges present in representative high-resolution simulation are listed.

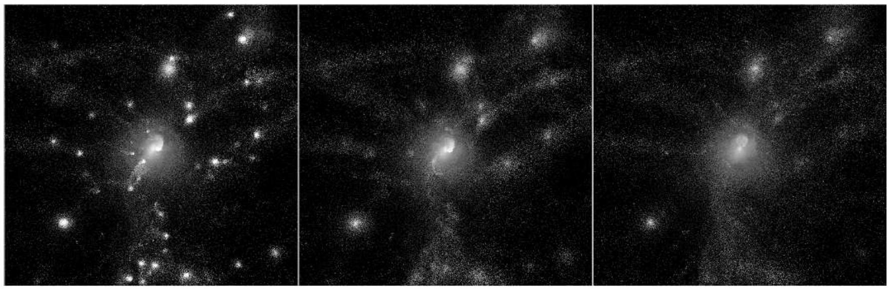

Figure 1. A qualitative look at the effect of extra heating on X-ray clusters. Entropy maps are shown for the same Virgo cluster analogue at the current epoch, within a box of 12.5 Mpc (41 million light years) where the nature of the extra-heating was varied for each simulation. Left panel: gravitational heating only; central panel: Supernova feedback with unity (high) efficiency; right panel: entropy floor $S_{\rm fl} = 50$ keV cm^2 imposed at early times ($z = 3$). Brighter regions indicate gas with lower entropy. The associated extremes of temperature are around $10^8 K$.

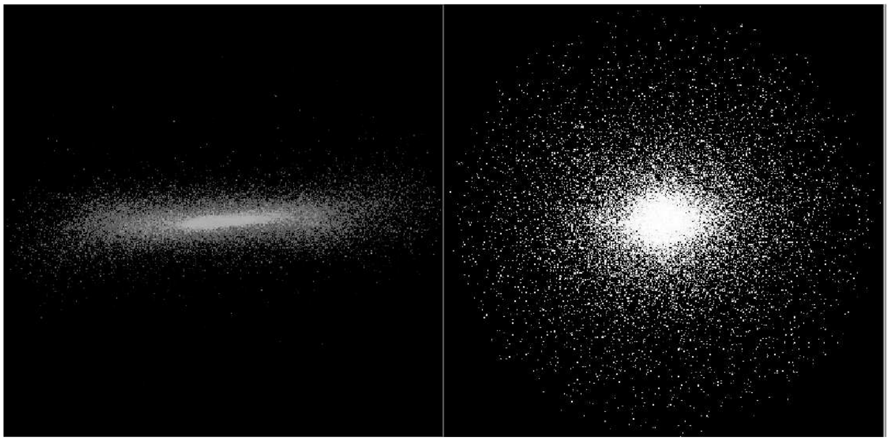

Figure 2. A spiral disk galaxy of comparable size to the Milky Way seen edge on at the current epoch. The left panel shows the younger stars that form the galactic disk and the right panel shows the old or spheroid component. The galaxy was formed from random initial conditions assuming the current preferred Lambda-CDM cosmological model.

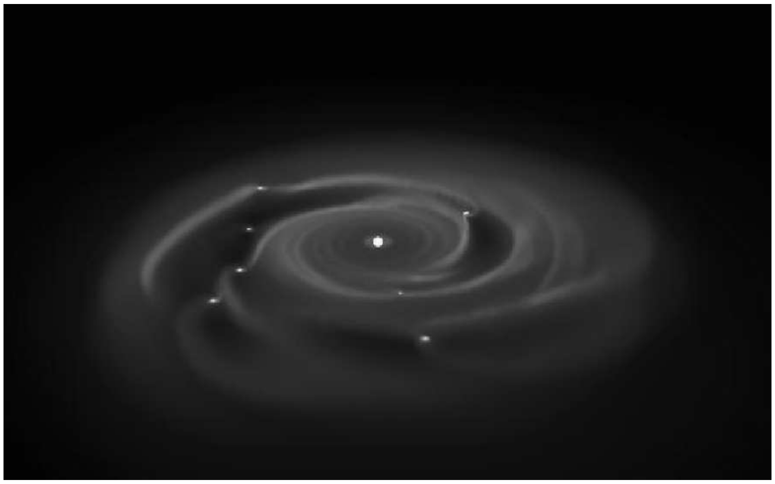

Figure 3. Formation of gas-giant planets in a self-gravitating gaseous disk 300 years after the start of the simulation. The bright points are large Jupiter-like planets. The initial condition was a million particle 0.1 solar mass disk extending from 4 to 20 A.U. with a $r^{-3/2}$ surface density profile. The gas was evolved isothermally from an initial temperature profile set by a standard radiative transfer model for disks around very young stars.

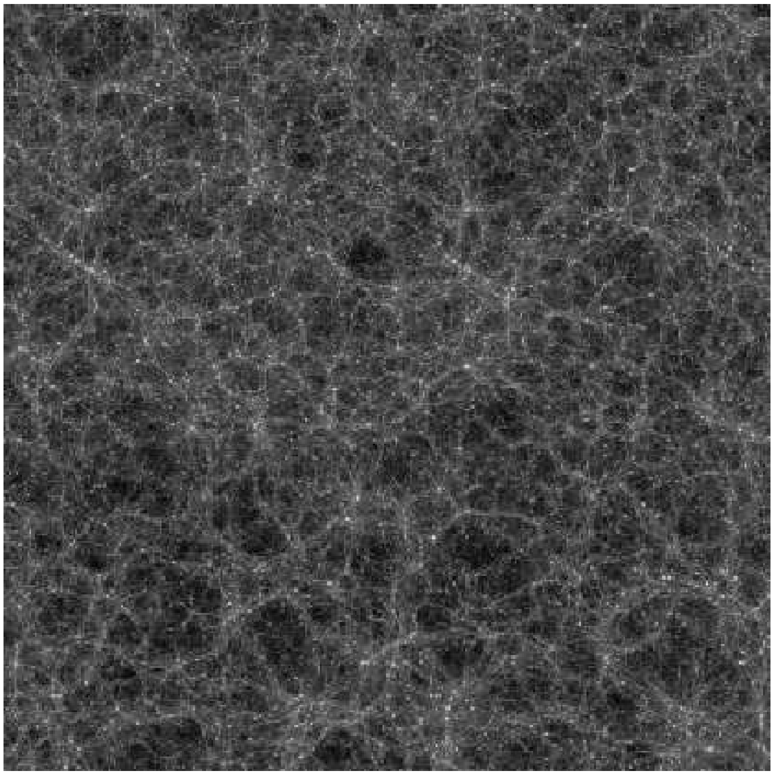

Figure 4. The gas distribution in the Sunyaev-Zeldovich simulation unrolled as a 290 million lightyear thick periodically extended 2.9 x 2.6 billion light year sheet. The bright filaments are dotted with bright points representing hot gas in cluster of galaxies that distort the primary signal present in the Cosmic Microwave Background. The volume of the 2×512^3 simulation is $(400Mpc)^3$ comoving and the initial perturbation spectrum was taken from a currently preferred Λ-CDM cosmological model.