THE ASTROPHYSICAL JOURNAL, 470:115–130, 1996 October 10

BUILDING A COSMOLOGICAL HYDRODYNAMIC CODE: CONSISTENCY CONDITION, MOVING MESH GRAVITY, AND SLH-P^{3}M

NICKOLAY Y. GNEDIN[1,2] AND EDMUND BERTSCHINGER[1]
Received 1996 February 12; accepted 1996 May 2

ABSTRACT

Building a self-gravitating hydrodynamic code as a combination of a hydrodynamic solver and a gravity solver is discussed. We show that straightforward combining of those two solvers generally leads to a code that does not conserve energy locally, and instead a special gravitational consistency condition ought to be satisfied. A particular example of combining softened Lagrangian hydrodynamics (SLH) with a P^{3}M gravity solver is used to demonstrate the effect of the gravitational consistency condition for a self-gravitating hydrodynamic code. The need to supplement the SLH method with the P^{3}M gravity solver arose because the moving mesh gravity solver, used in conjunction with the SLH method previously, was found to produce inaccurate results. We also show that most existing cosmological hydrodynamic codes implicitly satisfy the gravitational consistency condition.

Subject headings: cosmology: theory — dark matter — hydrodynamics — large-scale structure of universe — numerical methods

1. INTRODUCTION

Numerical cosmological simulations are now widely used to assess properties of various cosmological models and to study theoretical aspects of large-scale structure evolution. The variety of numerical methods employed ranges from high-resolution collisionless N-body simulations (see, e.g., Davis et al. 1985; Park 1990; Bertschinger & Gelb 1991; Gelb & Bertschinger 1994a, 1994b; Xu 1995) to sophisticated gasdynamics methods (see, e.g., Cen & Ostriker 1992a; Katz, Hernquist, & Weinberg 1992; Evrard, Summers, & Davis 1994; Bryan et al. 1994; Summers, Davis, & Evrard 1995) to numerical algorithms that at some level include galaxies together with collisionless dark matter and intergalactic gas as a distinct third component of a simulation (Cen & Ostriker 1992b, 1993a, 1993b; Frenk et al. 1995; Gnedin 1996a, 1996b).

However, not all of these simulation methods are reliably tested and well understood. Historically, collisionless N-body methods have been subjected to the most detailed study and testing. The theory of N-body methods is well developed, and their errors and limitations are well understood (Hockney & Eastwood 1981; Efstathiou et al. 1985). Cosmological hydrodynamics methods range from Eulerian hydrodynamic techniques borrowed from engineering applications (Cen et al. 1990; Ryu et al. 1993; Bryan, Norman, & Ostriker 1995) to quasi-Lagrangian methods that include the "smooth particle hydrodynamics" (SPH) method (Evrard 1988; Hernquist & Katz 1989) and "moving mesh approach" (MMA) (Gnedin 1995; Pen 1996). Unfortunately, currently there is no full understanding of errors and specifics of various hydrodynamic methods. The first comparisons between various cosmological methods (Kang et al. 1994; see also Gnedin 1995) demonstrated that differences between various methods can be so substantial that only a few statistical quantitative results are consistent between different methods. Further work on comparison of various numerical techniques is therefore required to support quantitative results of cosmological hydrodynamical simulations.

Since the moving mesh approach is the most recent and therefore the least understood, it requires special attention in comparing it with other existing numerical techniques. In this paper, we concentrate on comparing an implementation of the moving mesh approach called softened Lagrangian hydrodynamics (SLH; Gnedin 1995) with existing N-body techniques. One possible advantage of the moving mesh approach is that it offers a way to calculate gravitational forces with high resolution without employing a "particle" approach like P^{3}M or TREE. Since the moving mesh represents a single-valued coordinate transformation from quasi-Lagrangian space into real (physical) space, the Poisson equation in real space can be reduced to an elliptic partial differential equation in quasi-Lagrangian space, which can then be solved by the standard numerical techniques. In the SLH method thus calculated, the gravity force has also the virtue of having *exactly* the same resolution as the hydrodynamic solver has (we elaborate below what this actually means). There are also efforts under way to develop a purely N-body version of the moving mesh approach (Pen 1995).

The moving mesh approach looks very promising since it has higher spatial resolution in dense regions than Eulerian codes for the same amount of computational resources and is substantially faster than SPH methods for the same spatial resolution. We therefore undertook to investigate its accuracy by comparing SLH with the P^{3}M gravity solver. Our detailed tests, however, have uncovered two serious problems with the moving mesh gravity solver that have not been discussed in the literature; the problems we found may appear in a wide range of algorithms, so we report them in this paper. The first problem is the discreteness noise (shot noise) in high-density regions. It is well known that the shot noise does not represent a serious problem in traditional gravity solvers (Hernquist, Hut, & Makino 1993), where force errors effectively average out in high-density regions; however, in the moving mesh approach, the relationship between the density and the gravitational force is nonlinear, with the nonlinearity coming from the nonlinear dependence of the volume of a

[1] Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139; gnedin@arcturus.mit.edu, bertschinger@mit.edu.
[2] Princeton University Observatory, Peyton Hall, Princeton, NJ 08544; gnedin@astro.princeton.edu.

mesh element on the force acting on it. In this case, the shot noise does not necessarily average out as in the linear case, which leads to spurious numerical heating of high-density regions. Second, a solution to the finite-difference realization of the Poisson equation on a strongly deformed mesh does not necessarily converge to the true solution, giving the incorrect force law. We propose a solution to the first problem in § 2, but we cannot resolve the second one. We have therefore been forced to abandon the moving mesh gravity solver and have turned instead to a P^3M gravity solver that is known to work correctly and accurately. However, while developing the new SLH-P^3M code, we encountered another problem that is generic for (and potentially present in) any self-gravitating hydrodynamic code, not only a moving mesh code: since the gravity force solver and the hydrodynamic solver may treat spatial gradients differently, they are not a priori consistent with each other. This inconsistency may lead to a serious local nonconservation or imbalance of energy even if, globally, energy is well conserved. We discuss this effect in detail in § 3, where we also derive the "gravitational consistency condition" that every cosmological hydrodynamic code must obey in order to be energy conserving and apply it in § 4 to develop a new SLH-P^3M code. Finally, we repeat some of the Kang et al. (1994) tests for the new code in § 5. We summarize our conclusions in § 6.

2. FAILURE OF MOVING MESH GRAVITY

The softened Lagrangian hydrodynamics method was extensively tested in Gnedin (1995); however, the main emphasis there was on the hydrodynamic properties of the code. Later, we proceeded to test the gravitational solver of the SLH code and discovered interesting problems with the moving mesh gravity solver and potentially with self-gravitating hydrodynamic solvers in general. The purpose of this paper is to report the results of those tests and our solutions to the problems.

We have chosen a CDM+Λ model as a framework for our tests. The cosmological parameters we use are as follows: $\Omega_0 = 0.4$, $h = 0.65$, and $\sigma_8 = 0.85$. We use the BBKS transfer function (Bardeen et al. 1986) to set up initial conditions. The model is *COBE* normalized and is in reasonable agreement with all existing large- and intermediate-scale observations (Kofman, Gnedin, & Bahcall 1993; Peacock & Dodds 1994; Ostriker & Steinhardt 1995). We have run several simulations with 32^3 and 64^3 grids, all of them having the initial (Eulerian) comoving cell size (or, equivalently, the mean interparticle separation) equal to 0.5 h^{-1} Mpc, and, thus, a 32^3 simulation has a 16 h^{-1} Mpc box size, and a 64^3 simulation has a 32 h^{-1} Mpc box size. We set both the P^3M and SLH softening parameter to 1/10 (the softening length is 50 h^{-1} kpc) and keep it constant throughout our tests. Our simulations have box sizes too small to be of interest for large-scale structure formation at the current epoch, but they have sufficient resolution to push simulations into the highly nonlinear regime to test the SLH method in challenging conditions. In addition, we assume that the gas is "adiabatic," i.e., no radiative processes such as cooling or ionization are taken into account. Our current goal is to investigate in detail gravitational properties of the SLH code, and, therefore, we do not need to complicate our tests with radiative physics.

We mostly concentrate on properties of the largest clumps found in our simulations, unlike Kang et al. (1994), who mainly concentrated on some average statistical properties of gas simulations with different sizes.[3] First, we have run SLH and P^3M 32^3 simulations with identical initial conditions and with $\Omega_b = 0$ for the SLH code (the SLH code is thus used as a collisionless N-body solver; however, the hydrodynamic part is used to follow the mesh deformation as the mesh is tied to the baryonic gas in the SLH method). Figure 1 shows dark matter particles in a subvolume of these simulations containing the largest clump at current epoch. The lower left-hand panel shows particle positions for a SLH simulation, and the lower right-hand position shows P^3M particles. One can easily see that the SLH method fails to produce a clump nearly as dense as the P^3M clump. In our comparison, we explicitly assume that the P^3M gives results that are correct for dark matter ($\Omega_b = 0$) up to the softening length, and we use it as a template against which to compare the SLH results.

For comparison, we have also run the same SLH simulation but with $\Omega_b = \Omega_0$, which implies that the dark matter does not gravitate. The corresponding subvolume is plotted in Figure 1 (*upper left-hand panel*), and, as can be easily seen, there is much better agreement between the SLH and P^3M simulations in that case.[4] We must therefore ask ourselves what is the difference between the SLH $\Omega_b = 0$ and SLH $\Omega_X = 0$ cases that causes such a significant deviation.

There are two principal differences between dark matter–only and gas-only simulations with SLH. First, the dark matter density assignment uses the cloud-in-cell method to assign the dark matter density on a quasi-Lagrangian mesh (as explained below), and this assignment suffers from discreteness noise (or, in other words, shot noise). Second, the baryonic pressure tensor is isotropic, whereas the dark matter pressure tensor is not. However, the second reason cannot explain why a cluster failed to form with $\Omega_b = 0$ SLH, since in the P^3M simulation, the dark matter has an anisotropic pressure tensor as well, but a dense cluster forms. We conclude therefore that it is the SLH density assignment scheme that is responsible for the cluster diffusion problem.[5]

In order to elaborate on this effect, we recall the basic ingredients of the moving mesh approach in general and the SLH method in particular (the reader should refer to Gnedin 1995 for a complete explanation of notation and terminology). Let us consider a general coordinate transformation, which connects the real space coordinates, x^i, with some other coordinate system q^k, which we will call "quasi-Lagrangian" hereafter,

$$x^i = x^i(t, q^k) . \tag{1}$$

There are no restrictions on q^k up to now, but we will assume in the following that the new coordinate system q^k has the property that it redistributes numerical resolution into high-density regions. The important quantity that controls the degree of deformation between coordinate systems x^i and q^k is the *deformation tensor*,

$$A^i_k \equiv \frac{\partial x^i}{\partial q^k} . \tag{2}$$

[3] However, we repeat some of the tests of Kang et al. (1994) in § 5.

[4] The SLH gas feels isotropic pressure while the dark matter has an anisotropic pressure tensor, so some differences are expected.

[5] Note, that both dark matter–only and gas-only simulations are truly one-component simulations, and artificial interaction between the dark matter and the gas can not explain the cluster diffusion problem.

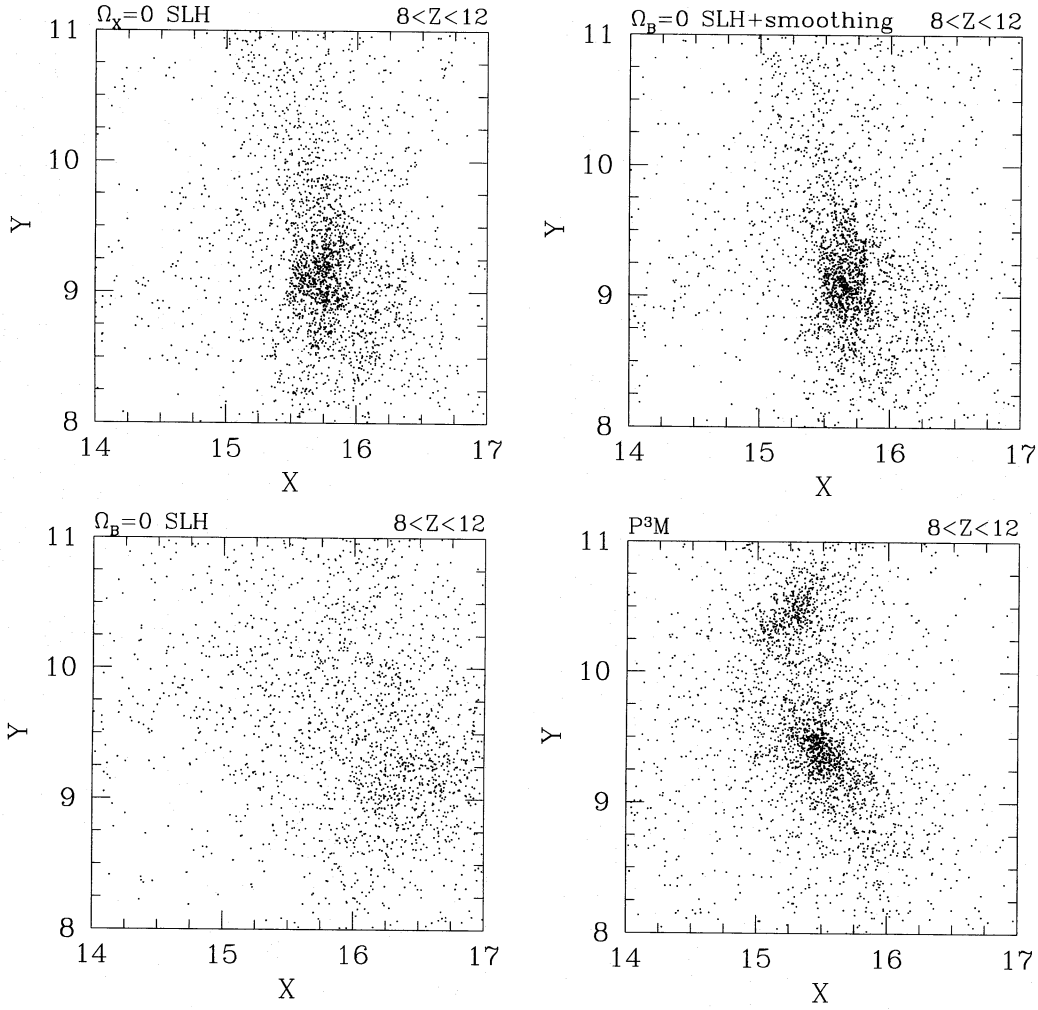

FIG. 1.—Dark matter particles in a subvolume of four 32^3 simulations: the original SLH $\Omega_b = 0$ (dark matter only) simulation (*lower left-hand panel*), the pure P^{3}M simulation (*lower right-hand panel*), the original $\Omega_X = 0$ (baryons only) simulation (*upper left-hand panel*), and the $\Omega_b = 0$ SLH simulations with dark matter density smoothed in quasi-Lagrangian space (*upper right-hand panel*).

Then the density of gas or dark matter can be represented as

$$\rho = \rho_0/\mathscr{A} , \tag{3}$$

where $\mathscr{A}$ is the determinant of the deformation tensor A^i_k,

$$\mathscr{A} \equiv \det A^i_k , \tag{4}$$

and $\rho_0 \equiv \rho_0(q^k)$ is the quasi-Lagrangian density or, in other words, the mass of a fluid element having a unit volume in quasi-Lagrangian space q^k. Note that quasi-Lagrangian coordinates may deviate significantly from exactly Lagrangian ones; i.e., for a given mass element, q^k need not be constant.

In a moving mesh approach (and, therefore, in the SLH code as well), coordinates x^i are real space images of vertices of a uniform mesh in quasi-Lagrangian space, $q^1_{ijk} = i$, $q^2_{ijk} = j$, and $q^3_{ijk} = k$. Since $\mathscr{A}$, as a function of the deformation tensor, is the property of the mesh, it is a smooth function of position (as smooth as the actual mesh equation allows it to be). For baryonic gas, $\rho_{0,B}$ is also a smooth function of position since it represents the mass of the gas in a cell; for a fully Lagrangian flow, $\rho_{0,B} \equiv 1$. However, this is not necessarily true for the dark matter density, which may, therefore, have large gradients in q-space. In the SLH approach, the dark matter density that sources the right-hand side of the Poisson equation is determined by assigning dark matter mass to cells in the quasi-Lagrangian space with the cloud-in-cell (CIC) technique:

$$\rho_{0,X}(q^l) = \sum_{i,j,k} m_{i,j,k}\, W(q^l - q^l_{i,j,k}) , \tag{5}$$

where W is the CIC weighting function,

$$W(\Delta q^l) = W_1(\Delta q^1) W_1(\Delta q^2) W_1(\Delta q^3) ,$$

and

$$W_1(\Delta q) = \begin{cases} 1 - |\Delta q| , & \text{if } |\Delta q| < 1 , \\ 0 , & \text{otherwise} \end{cases} .$$

In voids where, by construction, there are always almost the same number of dark matter particles per cell (since in voids the baryonic flow is Lagrangian and the pressure is negligible, the dark matter velocity is equal to the gas velocity at the same position and, therefore, both the dark matter and baryons move together in voids), and, therefore, $\rho_{0,X}$ is a smooth function of position and is almost constant. (It would be exactly constant before dark matter shell crossing and before gas shocking if the pressure in voids was zero and quasi-Lagrangian space was exactly Lagrangian.) This is not so in high-density regions. Dark matter particles

wander around in a cluster, while the mesh does not move substantially.

In Eulerian PM codes, the density fluctuations are a less severe (while still existing) problem since there are many dark matter particles per one PM cell in a high-density region and the dark matter density is determined accurately with the cloud-in-cell method. However, in the moving mesh approach, if the number of dark matter particles is equal to the number of baryonic cells, there is on average one (or slightly more than one if the dark matter is more concentrated than the baryons) dark matter particle per cell even in the highest density region since the high density is achieved not by collecting many particles in one immobile cell but by shrinking a cell to small volume while keeping its mass approximately constant. If these particles are distributed randomly with respect to the mesh, i.e., if all $q^l_{i,j,k}$ are random positions, equation (5) would predict that $\rho_{0,X}$ fluctuates on a scale of one cell with the amplitude comparable with its mean value. That means that the density ρ_X in clusters would fluctuate with the amplitude comparable with its mean value even in the cluster center. Since particles move, the dark matter density will fluctuate both spatially and temporally. The temporal fluctuations, however, seem to be less serious since the gravitational time scale in clusters is usually larger than the hydrodynamic Courant timescale.

Such enormous fluctuations will produce fluctuations in the gravitational potential; it has been shown (Hernquist et al. 1993) that those fluctuations effectively average out for traditional gravity solvers. However, in the moving mesh approach, the situation is different. Since the mesh responds to mass flows that are driven by gravity and the gravity force itself is calculated on the mesh, there is a nonlinear dynamical relation between the gravitational force and, say, the density defined on the mesh (with nonlinearity coming from the nonlinear dependence of the deformation tensor on the gravitational force). Therefore, the traditional analysis (Hernquist et al. 1993), in which the relation between the force and the density is linear, does not apply, and force errors do not necessarily average out as in the linear case. The nonlinearity between the density and the force also serves as a dissipative numerical mechanism, which leads to eventual blowing up of a cluster. We would like to stress here that this effect is a direct consequence of the shot noise effect in clusters, when in the moving mesh approach, there are only a few dark matter particles per cell even in the highest density regions.

It is easy to demonstrate that this shot noise effect is indeed present in moving mesh gravity solvers. First, we noticed above that this effect is absent when the baryons instead of the dark matter are gravitationally dominant, and $\rho_{0,B}$ is always a smooth function of position. Second, we can define $\rho_{0,X}$ as a smooth function of position if we smooth it in quasi-Lagrangian space. This procedure will not significantly degrade the resolution since it is $\mathscr{A}$, not $\rho_{0,X}$, that carries the high resolution. The smoothing will not affect voids since $\rho_{0,X}$ is almost constant in voids; it will affect places where the dark matter density is physically different from the baryon density, i.e., near shocks. We can hope, however, that these regions occupy a small fraction of the total volume and, what is more important, that they do not dominate the gravitational potential.

To test the shot noise hypothesis, we have run another SLH simulation in which we have smoothed the quasi-Lagrangian dark matter density $\rho_{0,X}$ entering the right-hand side of the Poisson equation with a Gaussian filter with width equal to three cell sizes in quasi-Lagrangian space (we would like to stress that this is *not* equivalent to smoothing the dark matter density, even in the highest density regions, since the factor $\mathscr{A}$ remains unsmoothed). The resulting particle positions are plotted in Figure 1 (*upper right-hand panel*). One can easily see that a cluster is now formed. Figure 2 shows density profiles for the clusters in these four cases after the cluster was identified with the DENMAX algorithm (Bertschinger & Gelb 1991). The SLH gravity with smoothing is perhaps a factor of 1.5 softer than the Plummer law for these simulations.[6]

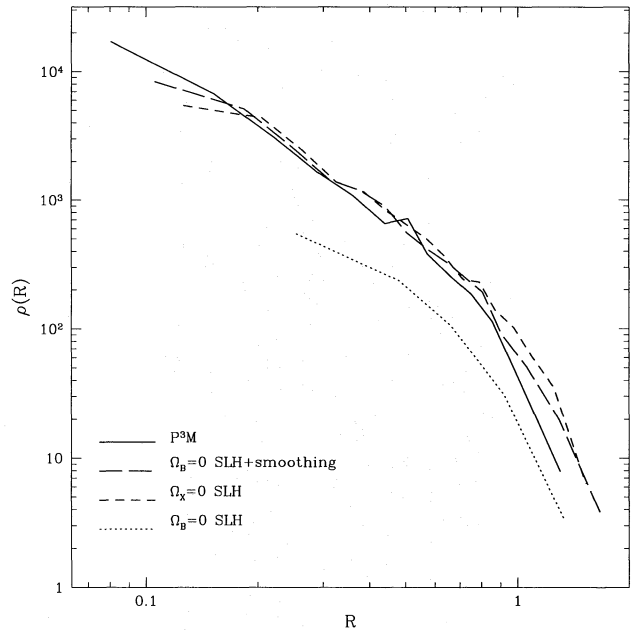

FIG. 2.—Density profiles for four clusters shown in Fig. 1: P³M only (*solid line*), $\Omega_b = 0$ SLH (*dotted line*), $\Omega_X = 0$ SLH (*short-dashed line*), and $\Omega_b = 0$ SLH with smoothing (*long-dashed line*). The radius has units of the mean interparticle spacing; the force softening is $R = 0.1$.

In order to test the moving mesh gravity further, we have performed two 64^3 simulations with the parameters described above: one is a P³M simulation, and the other one is a SLH simulation with $\Omega_b = 0$ and dark matter smoothing as explained in the previous paragraph. Particles around the heaviest cluster in the two simulations are plotted in Figure 3 in the right-hand and left-hand panels, respectively. One can see that at higher resolution (64^3 vs. 32^3 in Fig. 1) the SLH cluster again diffuses outward. Figure 4 shows the density profile for the cluster. Obviously, the SLH gravity effectively softens at a scale a factor of 6 or 7 larger than the formal softening length. Since the effect of shot noise in high-density regions is eliminated in the SLH simulation, one has to admit the existence of yet another problem with the SLH gravity.

[6] We have also performed the same SLH simulation without smoothing but with 8 and 64 times more dark matter particles; there is a definite improvement as the number of dark matter particles increases, but the rate of improvement is very slow, and even with 64 times more dark matter particles than baryonic cells, the cluster under consideration is still significantly diffused out.

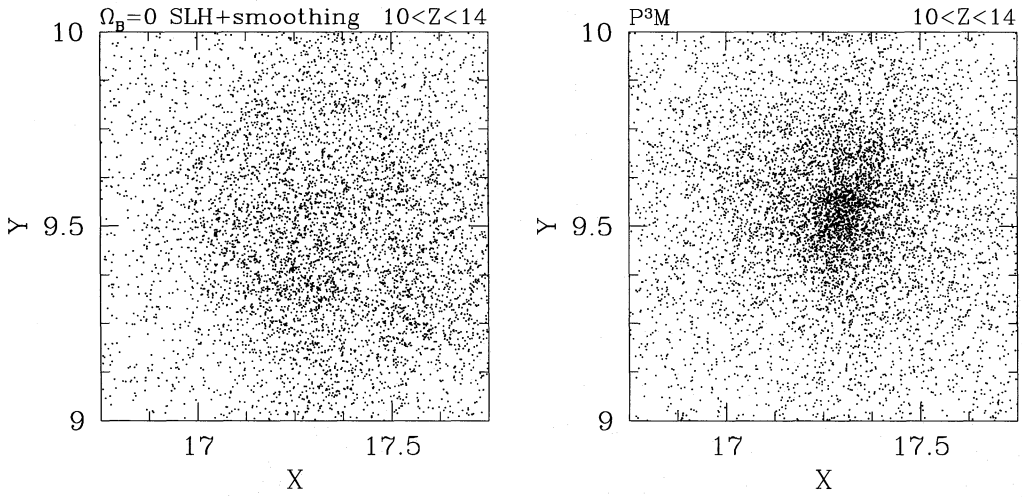

FIG. 3.—Dark matter particles in a subvolume of two 64^3 simulations: the $\Omega_b = 0$ SLH simulation with smoothing (*left-hand panel*) and the pure N-body simulation (*right-hand panel*).

In order to investigate the accuracy of the SLH Poisson solver, we calculated point-mass gravitational force placing a point mass at a random position on the SLH mesh taken from the 64^3 SLH simulation at $z = 0$. The corresponding direction-averaged force laws for 10 different random positions are plotted in Figure 5 (*left-hand panel*). Since points are chosen randomly on a mesh, their effective smoothing lengths are different (for example, in a void, gravitational smoothing is larger than the original cell size since a cell in a void has expanded during the evolution); in general, however, there is an acceptable agreement between the SLH point-mass force and $1/R^2$ law (Fig. 5, *dotted line*) on scales larger than the local softening length and smaller than the box size, where the periodic boundary conditions alter the force law. Next, instead of 10 randomly chosen points, we picked up two points that lay close to the center of the biggest cluster in the SLH simulation. Their point-mass force laws are plotted in Figure 5 with solid lines. These laws deviate significantly from the $1/r^2$ law at 20–30 softening lengths even if these cells have sizes of the order of a softening length.

What causes these errors in gravity in a cluster? In order to test if it is due to the loss of accuracy in the finite differencing of the gravitational potential, instead of solving the

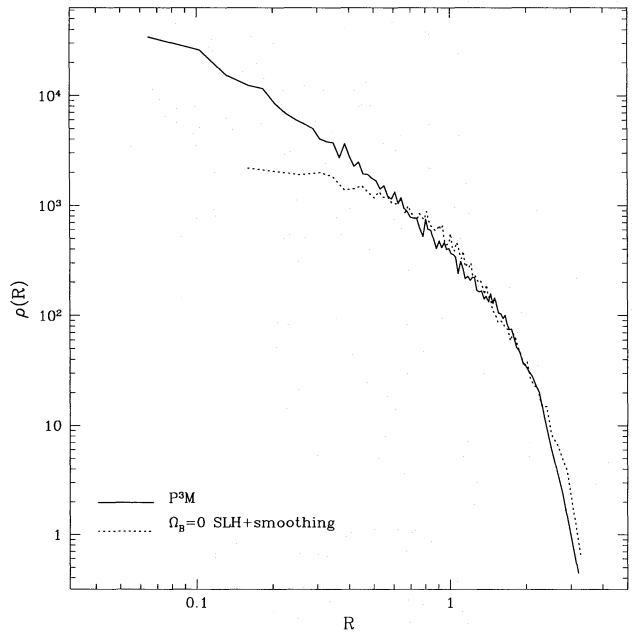

FIG. 4.—Density profiles for two simulations shown in Fig. 3: P^{3}M only (*solid line*) and $\Omega_b = 0$ SLH with smoothing (*dotted line*).

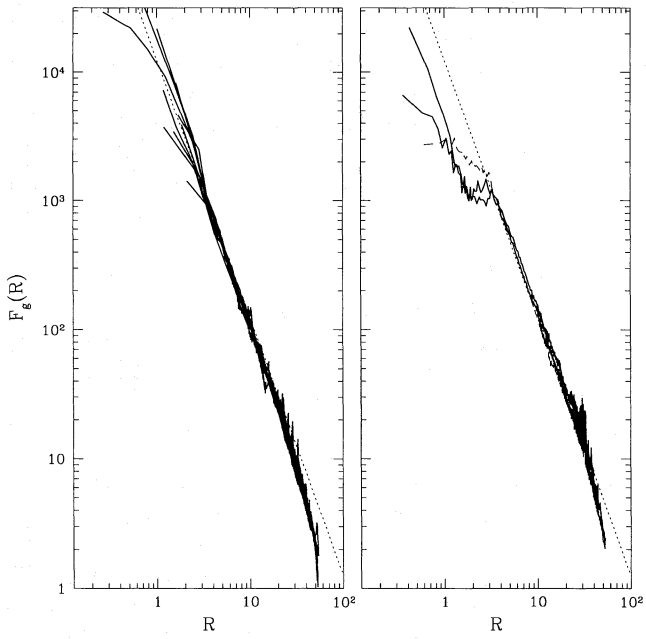

FIG. 5.—*Left*: Green's functions for 10 randomly chosen points in the 64^3 SLH simulation (*solid lines*) and the inverse square law (*thin dotted line*). *Right*: Green's functions for two points chosen near the center of the cluster from the left panel of Fig. 3 (*solid lines*) and the Green's function for one of those points calculated by solving a Poisson equation for three components of the gravitational force on the deformed mesh (*dashed line*); shown with the thin dotted line is the inverse square law. The radius has units of the mean interparticle spacing; the minimum force softening is $R = 0.1$.

Poisson equation

$$\Delta\phi = \frac{3\Omega_0 a}{2}\delta\,, \tag{6}$$

we solved three Poisson equations for each of force component separately,

$$\Delta f_i = \frac{3\Omega_0 a}{2}\frac{\partial\delta}{\partial x^i}\,, \tag{7}$$

which eliminates the need to differentiate the potential but requires at least 3 times the CPU time used by the potential solver. The corresponding point mass force law is plotted with the thin dashed line in the right panel of Figure 5. While there is some improvement in solving for the force compared to solving for the potential, it is obviously insufficient to improve the moving mesh gravity solver. We specifically stress here that in this test, the specifics of the SLH algorithm come only with the strongly deformed mesh; the Poisson solver is a *generic* feature of any moving mesh approach, and it is this that fails. We, thus, have encountered another problem with the moving mesh gravity: when the mesh is strongly deformed, the Poisson solver occasionally gives incorrect results. It seems quite difficult to find out what specifically is wrong with the Poisson solver or what properties of the mesh cause such a big error. It is also not clear if the less deformed mesh has the same problem but is less pronounced so that it avoids clear recognition.

Even if the problem we have highlighted appears only in small volume of the whole simulation, there is no guarantee that the rest of a simulation is correct or has a specified resolution. Until the specific problem with the moving mesh gravity is identified and resolved, we feel that it is unsafe to use the moving mesh gravity solver for cosmological simulations. The full analysis of Poisson solver errors on strongly deformed meshes is beyond the scope of this paper, and, therefore, we have decided to abandon the moving mesh gravity solver and proceed into combining the SLH hydrodynamic method with the P^{3}M solver which is known to give correct results (Bertschinger 1991). We shall see that merging these codes introduces another problem to solve.

3. GRAVITATIONAL CONSISTENCY CONDITION

We now imagine building a cosmological hydrodynamic code by combining separate hydrodynamic and gravity solvers together. Both methods can calculate hydrodynamic and gravitational forces on a resolution element scale quite differently, and there is a priori no guarantee that these two methods would be consistent with each other.

Let us consider an example. Let us imagine that we constructed an isothermal gas sphere; we utilize so many resolution elements (cells or particles) that we do not have to worry about relaxation processes. We now insert this object into a hydro + gravity code that has gravitational force softening much shorter than the pressure force softening.[7] The object, which would be in real physical equilibrium (i.e., in equilibrium with exact force laws for both gravity and pressure force), would collapse further since the softened gravity force would be stronger than the softened pressure force because of larger pressure force softening. The object would thus form a small and dense clump that is a purely numerical artifact. This process is an example of a more general inconsistency between hydro and gravity solvers that we discuss in detail below.

Let us consider the hydrodynamic equations with self-gravity in a general coordinate system q^k as defined above (at this point, we do not require that the coordinate system q^k be close to Lagrangian; in particular, it may be fully Eulerian when $x^i = q^i$). Let ρ be the density, v^i be the velocity, P be the pressure, $E \equiv \rho v_i^2/2 + P/(\gamma - 1)$ be the total energy density, and f_i be the gravitational force per unit mass of the fluid with the polytropic index γ. The Euler equations in a coordinate system q^k read:

$$\frac{\partial\rho_0}{\partial t} + \frac{\partial}{\partial q^k}(\rho_0 w^k) = 0\,, \tag{8a}$$

$$\frac{\partial\rho_0 v_i}{\partial t} + \frac{\partial}{\partial q^k}(\rho_0 v_i w^k + \mathscr{A}B_i^k P) = \rho_0 f_i\,, \tag{8b}$$

and

$$\frac{\partial E_0}{\partial t} + \frac{\partial}{\partial q^k}(E_0 w^k + v^i \mathscr{A}B_i^k P) = \rho_0 v^i f_i\,, \tag{8c}$$

where the deformation tensor A_k^i has been defined above; B_k^i is the matrix inverse to A_k^i,

$$B_j^i A_k^j = A_j^i B_k^j = \delta_k^i\,;$$

$\mathscr{A} \equiv \det(A_k^i)$ is the determinant of the deformation tensor, which is equal to the volume in real space of a fluid element with the unit volume in quasi-Lagrangian space q^k, $d^3x = \mathscr{A}d^3q$; ρ_0 is the quasi-Lagrangian mass density,

$$\rho_0 \equiv \rho\mathscr{A}\,; \tag{9}$$

E_0 is the quasi-Lagrangian total energy density,

$$E_0 \equiv E\mathscr{A}\,; \tag{10}$$

and

$$w^k \equiv B_i^k(v^i - \dot{x}^i)\,, \tag{11}$$

with the dot standing for the partial time derivative, $\dot{x}^i(t, q^k) = \partial x^i(t, q^k)/\partial t$.

If we now consider a hydrodynamic code that solves these equations, we must interpret the spatial derivative $\partial/\partial q^k$ as a numerical derivative (for example, a finite difference derivative for a mesh-based code), and we denote it as $\partial_N/\partial q^k$, and we must include numerical dissipation terms responsible for shock handling and stability. Let us introduce a numerical density flux correction $\mathscr{F}^k$, a numerical momentum flux correction $\mathscr{G}_i^k$, a numerical total energy flux correction $\mathscr{H}^k$, and a numerical gravitational energy correction Λ through the following formulae:

$$\frac{\partial\rho_0}{\partial t} + \frac{\partial_N}{\partial q^k}(\rho_0 w^k + \mathscr{F}^k) = 0\,, \tag{12a}$$

$$\frac{\partial\rho_0 v_i}{\partial t} + \frac{\partial_N}{\partial q^k}(\rho_0 v_i w^k + \mathscr{A}B_i^k P + \mathscr{G}_i^k) = \rho_0 f_i\,, \tag{12b}$$

[7] This should be considered only as a thought experiment; in reality, hydrodynamic codes have pressure softening comparable to the resolution element size and, thus, our assumption of having many resolution elements per pressure or gravity softening length is unrealistic. To implement it in practice, one would require to smooth both pressure and gravity force over a scale encompassing many resolution elements.

and

$$\frac{\partial E_0}{\partial t} + \frac{\partial_N}{\partial q^k}(E_0 w^k + v^i \mathcal{A} B_i^k P + \mathcal{H}^k) = \rho_0 v^i f_i + \Lambda . \quad (12c)$$

Among those four, only $\mathcal{F}^k$ and Λ will be of interest to us. We now want to establish the total energy conservation. The total energy can be represented as the sum of the total gas energy,

$$\mathcal{E} = \int_N E_0 \, d^3q ,$$

and the gravitational energy,

$$\mathcal{W} = \tfrac{1}{2} \int_N \rho_0 \phi \, d^3q ,$$

where we understand integrals in a numerical sense and denote this by index N, and ϕ is a gravitational potential and is determined by the following equation:

$$\phi(q^k) = \int_N G[\,|x^i(q^k) - x^i(q_1^k)|\,]\rho_0(q_1^k) d^3q_1 , \quad (13)$$

and $G(r)$ is a Green's function for a gravity solver. It is important to note here that this function is arbitrary, and, therefore, our consideration is equally applicable to a softened gravity law as well as to the $1/r^2$ law. Moreover, we do not need to specify a softening length for the gravity. It will be shown below that the problem of inconsistency does not reduce to the problem of different resolution scales for gravity and gas and is more generic.

We now demand (neglecting cosmological expansion for the moment) that

$$\frac{d}{dt}(\mathcal{E} + \mathcal{W}) = 0 . \quad (14)$$

It can be easily seen that

$$\frac{d}{dt}\mathcal{E} = \int_N \frac{\partial E_0}{\partial t} d^3q = \int_N (\rho_0 v^i f_i + \Lambda) d^3q , \quad (15)$$

provided the numerical integral of a numerical divergence is zero. The latter condition is satisfied exactly for any hydrodynamic scheme that conserves energy exactly in the absence of a gravitational field and heating or cooling, since in that case the rate of local energy density change,

$$\frac{\partial E_0}{\partial t} = -\frac{\partial_N}{\partial q^k}(E_0 w^k + v^i \mathcal{A} B_i^k P + \mathcal{H}^k) ,$$

is a numerical divergence and must vanish when integrated over a sufficiently large volume (we ignore a flux over boundaries here for simplicity and because cosmological simulations usually have trivial periodic boundary conditions), whereas details of a particular numerical scheme are hidden into $\mathcal{H}^k$ and are of no relevance for us here.

For the time derivative of the gravitational energy, we obtain

$$\frac{d}{dt}\mathcal{W} = \int_N \frac{\partial \rho_0}{\partial t} \phi d^3q + \int_N \rho_0 \dot{x}^i \frac{\partial}{\partial x^i} \times \left\{ \int_N G[\,|x^i - x^i(q_1^k)|\,]\rho_0(q_1^k) d^3q_1 \right\} d^3q . \quad (16)$$

It is important to note that the derivative with respect to x^i that enters the last equation is a *full derivative*, not the numerical one. We can introduce the new quantity, $\hat{f}_i$, defined as

$$\hat{f}_i \equiv -\frac{\partial \phi}{\partial x^i} = -\frac{\partial}{\partial x^i}\left\{ \int_N G[\,|x^i - x^i(q_1^k)|\,]\rho_0(q_1^k) d^3q_1 \right\} d^3q , \quad (17)$$

which is the true gradient (with the minus sign) of a numerical gravitational potential, i.e., the gradient that will be computed in a simulation, in which all spatial gradients are computed mathematically exactly (for example, a P^{3}M code possesses this property). Note that, because of the finite number of resolution elements in a real simulation, f_i is not necessarily equal to $\hat{f}_i$ but, rather, is a numerical approximation to it. Equation (16) can now be written as follows:

$$\frac{d}{dt}\mathcal{W} = \int_N \frac{\partial \rho_0}{\partial t} \phi d^3q - \int_N \rho_0 \dot{x}^i \hat{f}_i d^3q . \quad (18)$$

We now assume that the numerical derivative ∂_N obeys the product rule, i.e.,

$$\frac{\partial_N}{\partial q^k}(fg) = f \frac{\partial_N g}{\partial q^k} + g \frac{\partial_N f}{\partial q^k} . \quad (19)$$

This is true for dimensionally split mesh codes with central differences and can be true in SPH codes with appropriate choice of weighting scheme; it is generally not true in the SLH code since it is not dimensionally split, and, therefore, the SLH code has additional energy errors compared to dimensionally split mesh codes. (These errors are typically small and random and seem not to contribute significantly to the total energy error, although, in principle, nothing prevents them from causing a significant energy error.) Then we can use equation (12a) to reduce equation (18) to the following expression:

$$\frac{d}{dt}\mathcal{W} = \int_N (\rho_0 w^k + \mathcal{F}^k) \frac{\partial_N \phi}{\partial q^k} d^3q - \int_N \rho_0 \dot{x}^i \hat{f}_i d^3q . \quad (20)$$

Combining both equation (15) and equation (20) with equation (14), we finally obtain

$$\int_N \left(\Lambda + \mathcal{F}^k \frac{\partial_N \phi}{\partial q^k} \right) d^3q + \int_N \rho_0 \left[v^i f_i - \dot{x}^i \hat{f}_i + (v^i - \dot{x}^i) B_i^k \frac{\partial_N \phi}{\partial q^k} \right] d^3q = 0 . \quad (21)$$

We can immediately derive the form for Λ from the equation (21) taking into account that Λ is the numerical correction to the gravitational energy change and it must vanish when $\mathcal{F}^k$ vanishes:

$$\Lambda = -\mathcal{F}^k \frac{\partial_N \phi}{\partial q^k} . \quad (22)$$

Now we are left with the following equation, which must be satisfied if the energy is to be conserved:

$$\int_N \rho_0 \left[v^i f_i - \dot{x}^i \hat{f}_i + (v^i - \dot{x}^i) B_i^k \frac{\partial_N \phi}{\partial q^k} \right] d^3q = 0 . \quad (23)$$

Let us examine this carefully. There are three gravitational forces that enter this equation: the original force that a full gravity solver provides, $\hat{f}_i$; the force that the hydrodynamic code feels, f_i; and the numerical gradient of the gravitational potential, $B_i^k \partial_N \phi / \partial q^k$. The last expression is the gradient of a scalar function as the hydrodynamic code understands it;

for a mesh-based code, it is a finite difference, while for a SPH-like code it is the function values weighted by the gradients of the kernel, etc.

If we require that the self-gravitating hydrodynamic code should conserve energy everywhere and for all possible cases, we must satisfy the gravitational consistency condition at every resolution element:

$$v^i f_i = \dot{x}^i \hat{f}_i - (v^i - \dot{x}^i) B_i^k \frac{\partial_N \phi}{\partial q^k} . \quad (24)$$

Strictly speaking, we may add a divergence term, which always can be hidden into $\mathscr{H}^k$. This condition should be considered as a restriction that the gravitational force f_i should satisfy in order for the self-gravitating hydrodynamic code to be energy conserving.

It is instructive to consider some special cases:

1. A fully Lagrangian code (in particular, SPH). For a fully Lagrangian code, $\dot{x}^i = v^i$ and equation (24) is satisfied if $f_i = \hat{f}_i$. Thus, any gravitational solver is consistent with the fully Lagrangian code; in other words, a fully Lagrangian code is a priori consistent with a gravity solver, and no special care should be taken to satisfy the gravitational consistency condition.

2. A fixed mesh finite-difference code (in particular, a fully Eulerian finite-difference code). For this case, we can assume $\dot{x}^i = 0$, and the gravitational consistency condition is satisfied if

$$f_i = -B_i^k \frac{\partial_N \phi}{\partial q^k} . \quad (25)$$

We conclude, therefore, that it is essential that the gravitational force in a fixed mesh code be a finite difference gradient of the gravitational potential. In particular, in a PM + Eulerian finite-difference code combination, it is the gravitational potential that should be calculated in the PM part and not the force.

3. A Riemann solver. In this case the numerical gradient $\partial_N/\partial q^k$ is actually a Riemann solver; it is not straightforward to represent the gravitational force as the gradient of a potential in this case and our notation for a numerical derivative, $\partial_N/\partial q^k$ becomes obscure, so that equation (25) should then be understood only symbolically. The gravitational consistency is achieved in this case by appropriately incorporating the gravitational force into a Riemann solver itself; in particular, this is the way the KRONOS code works (Bryan et al. 1995).

4. A moving mesh gravity code (i.e., a code that uses the moving mesh gravity as a gravity solver). Strictly speaking, for this case, equation (13) does not hold, but we can still consider it here, recalling that

$$\hat{f}_i = -B_i^k \frac{\partial_N \phi}{\partial q^k}$$

for a moving mesh gravity code. In that case, f_i is also equal to both $\hat{f}_i$ and $-B_i^k \partial_N \phi/\partial q^k$, and the gravitational consistency condition holds.

Thus, the gravitational consistency condition is usually satisfied for most of the currently used numerical techniques.

Let us consider here a simple example. Let us imagine that we have a one-dimensional Eulerian finite-difference hydrodynamic code (so that the product rule [eq. (19)] is always satisfied) that keeps all hydrodynamic quantities at the centers of equal cells numbered by index $i = 1, 2, \ldots, N$ (where N is the number of cells and the periodic boundary condition is assumed), so that the gas density at the cell i is ρ_i, the energy density is E_i, etc. This code is combined with the PM gravity solver that uses the gas density ρ_i to solve the Poisson equation (using, say, the FFT technique)

$$\phi_{i+1} + \phi_{i-1} - 2\phi_i = 4\pi G \rho_i \quad (26)$$

(we assume here $\Delta x = 1$ for simplicity) to obtain the gravitational potential ϕ_i at the cell centers, derives the gravitational force f_i acting at the cell center by finite differencing the potential,

$$f_i = (\phi_{i-1} - \phi_{i+1})/2 , \quad (27)$$

and uses it to update the gas velocity v_i. The code constructed this way conserves total energy exactly (provided the hydrodynamic part conserves the kinetic plus thermal energy exactly, as is the case for conservative schemes) as explained above, i.e., it satisfies the gravitational consistency condition. Now let us imagine that one decides to replace the PM gravity solver with the exact gravity solver, which calculates the exact gravitational potential $\hat{\phi}$ and force $\hat{f}$ given the density distribution:

$$\rho(x) = \begin{cases} \rho_1 , & 0 < x \le 1 , \\ \rho_2 , & 1 < x \le 2 , \\ \ldots \\ \rho_i , & i - 1 < x \le i , \\ \ldots \\ \rho_N , & N - 1 < x \le N . \end{cases}$$

(One can achieve this, for example, by using the PM gravity solver with a much finer mesh.) If one now uses the exact gravitational force $\hat{f}_i$ to update the gas velocity v_i, the gravitational consistency condition will not be satisfied, and local energy errors will be introduced. However, if one uses the exact gravitational potential $\hat{\phi}$ and calculates the gravitational force f using equation (27) with the new potential, the energy conservation will be restored even if the exact gravitational potential $\hat{\phi}$ is different from the potential ϕ obtained by solving the Poisson equation on the hydrodynamic mesh. Thus, the gravitational consistency condition requires a consistent treatment of the gravitational potential and forces in the dark matter and gas.

We would like to stress here again that the gravitational consistency condition is not equivalent to having gravity and pressure force softening commensurate; the Green's function in equation (13) allows for any value for the gravitational softening parameter; more than that, since the shape of the Green's function is not specified, it is not required that the force f_i be actually a gravitational force; any potential force solver (for example, molecular forces) should satisfy a consistency condition similar to equation (24). This comment would imply that the requirement of having gravitational and pressure softenings commensurate is an *additional* requirement that a self-gravitational hydrodynamic code must satisfy.

4. A SLH-P^{3}M CODE

We now turn to constructing a SLH-P^{3}M code, where the gravity force is calculated by a P^{3}M solver and the mesh

velocity $\dot{x}^i$ is not necessarily equal to the fluid velocity v^i. In this case, it is impossible to solve equation (24) exactly, and only an approximate solution can be proposed.

We start by noting that the SLH mesh equation, i.e., the equation that connects $\dot{x}^i$ and v^i in the SLH approach,

$$\frac{\partial^2 \dot{x}^i}{\partial q_m \partial q^m} = \frac{\partial}{\partial q_m}\left[\left(\delta^i_j - \sigma^i_j\right)\frac{\partial v^j}{\partial q^m}\right], \tag{28}$$

is a Galilean invariant modification of the following equation:

$$\dot{x}^i = (\delta^i_j - \sigma^i_j)v^j \tag{29}$$

(here σ^i_j is the softening tensor that vanishes in the Lagrangian limit and approaches δ^i_j in the Eulerian limit; for full definitions see Gnedin 1995 or the Appendix). If we consider $\dot{x}^i$ from equation (29) as an approximation to $\dot{x}^i$ derived from equation (28) and substitute equation (29) into equation (24), we obtain the following (Galilean-invariant) equation:

$$f_i = \hat{f}_i - \sigma^j_i\left(\hat{f}_j + B^k_j \frac{\partial_N \phi}{\partial q^k}\right). \tag{30}$$

This equation will serve as our definition for the gravity force on the gas in the SLH-P^{3}M code. Both $\hat{f}_i$ and ϕ are obtained by means of a P^{3}M code, and f_i is the gravitational force applied to the baryonic gas. The forces on the dark matter particles are given instead by $\hat{f}_i$. Tests also show that the following simplified version of equation (30) is a good choice as well:

$$f_i = \hat{f}_i - \bar{\sigma}\left(\hat{f}_i + B^k_i \frac{\partial_N \phi}{\partial q^k}\right), \tag{31}$$

where $\bar{\sigma}$ is the maximum eigenvalue of the softening tensor σ^j_i. One can note that, again, in the fully Lagrangian ($\bar{\sigma} = 0$) case, $f_i = \hat{f}_i$, while in the fully Eulerian ($\bar{\sigma} = 1$) regime, $f_i = -B^k_i \partial_N \phi/\partial q^k$.

Equations (30) and (31) have a remarkable feature that they *guarantee the commensurate softenings for both gravity and pressure* for the SLH code, since the SLH code switches from the Lagrangian to the Eulerian description as density increases, and, therefore, the gravitational force at the resolution limit is the gradient of a scalar function in the way the hydrodynamic solver understands a gradient; gravitational and pressure resolutions for the gas are again *exactly* equal as with the original SLH code. However, since the dark matter particles are acted upon by the force $\hat{f}_i$, the gravitational resolution for the dark matter component need not be (but, of course, may be set) equal to the gravitational resolution for the gas component.

Now we turn to the details of implementing SLH-P^{3}M. There exist two distinct ways to combine SLH hydro and P^{3}M gravity codes (and many intermediate variants).

The first one, which we call a "maximally coupled" code, implies that dark matter "lives" in quasi-Lagrangian space as in the original SLH code; then the P^{3}M part is used to calculate forces acting on vertices of the mesh, where the mass in a vertex is found by linear interpolation from the cell centers and the mass of a cell is determined by the total mass of dark matter, gas, and, possibly, galaxies in the cell. This approach requires calculating the P^{3}M force only on N_B mesh vertices and does not suffer from the shot noise problem since the relation between the density and the gravitational force is, contrary to the moving mesh approach, linear, and the Hernquist et al. (1993) analysis fully applies. However, since the mesh is tied to baryons, the maximally coupled version fails to follow accurately places where the dark matter density differs significantly from the gas density, like cores of big clusters.

The second variant, which we call a "minimally coupled" code, allows dark matter to evolve in real space similar to existing SPH approaches. The mesh is then considered to represent the baryonic gas only, and only the gas mass is included in computing the cell mass. This variant of the SLH-P^{3}M code does not suffer from the limitations of the minimally coupled mode but requires substantially more CPU time since computing of the gravitational force acting on both N_B grid cells and N_X dark matter particles is required. For the most common case, $N_X = N_B$, this would require twice the PM time and 4 times the PP time of the maximally coupled code. When combined with the hydrodynamic part, this averages out to a minimally coupled mode being approximately 2.5 times slower than the maximally coupled mode.

We now proceed to testing the SLH-P^{3}M code. Since, in addition to the mesh softening of SLH, the P^{3}M gravity solver introduces a Plummer softening, the new SLH-P^{3}M code has two softening lengths. In what follows, we set these two softenings equal to 1/10 so that our simulation resolution will not exceed that of a pure N-body template run; in principle, the gravitational force acting on the gas can be computed with vanishing Plummer softening since the mesh softening will prevent exceedingly small scales from being resolved. Therefore, in the maximally coupled mode, when the dark matter, being placed on the baryonic mesh, is subject to the mesh softening, the Plummer softening can be assumed arbitrary small, thus squeezing the last drop of resolution from the simulation. One must note, however, that in that case, the resolution becomes even more anisotropic and nonuniform across the simulation volume.

First, we would like to demonstrate the importance of the gravitational consistency condition. Figure 6 shows the gas distribution of the biggest clusters in two 32^3 baryons-only ($\Omega_X = 0$) simulations (see Fig. 1): the left-hand panel shows the results for the simulation in which no gravitational consistency condition was applied and the gravitational force acting on the gas was directly provided by the P^{3}M part of the code; the right-hand panel shows the same cluster in the simulation in which the gravitational consistency condition of the form of equation (31) was incorporated in the code (we remind the reader that the corresponding N-body result is plotted in the lower right-hand panel of Fig. 1). It is easily seen that without the gravitational consistency condition, most of the gas sinks to the very center of the cluster, thus violating energy conservation. This effect of gas sinking toward the cluster center can be solved without the gravitational consistency condition by simply smoothing the gravitational force on a scale of a cell size; in that case, the gravity force would have approximately the same softening as the pressure force. This, nevertheless, would *not* solve the problem since the cluster profile would still be significantly different from the N-body results because of local nonconservation of energy.

Since the original SLH method failed substantially only at the resolution of a 64^3 mesh, we ran a 64^3 $\Omega_X = 0$ minimally coupled SLH-P^{3}M simulation (the case $\Omega_B = 0$ coin-

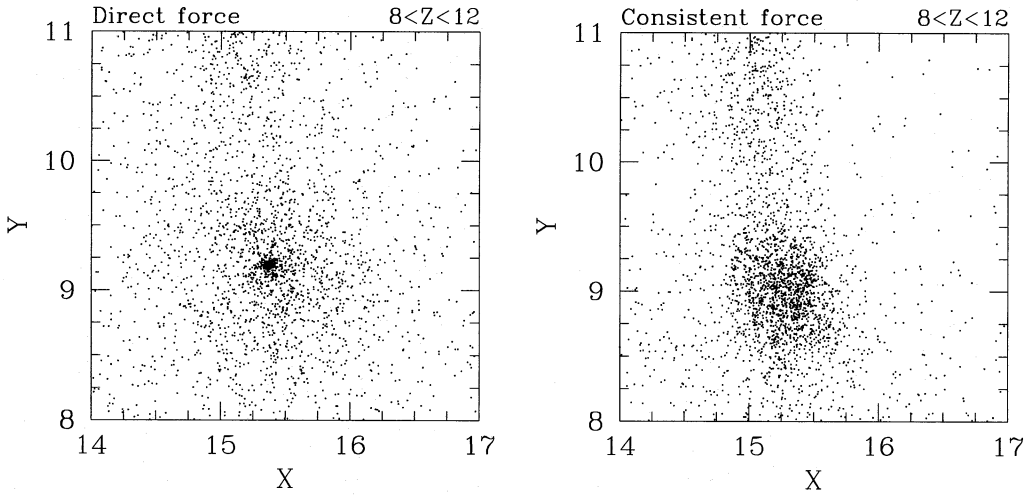

FIG. 6.—Baryonic particles (cell centers) for two 32^3 SLH-P^{3}M simulations: without the gravitational consistency condition (*left*) and with the gravitational consistency condition (*right*).

cides with the pure N-body result). The profiles for the two largest clusters in the simulation are shown in Figure 7 together with the pure N-body result (*solid line*). The dashed line shows the dark matter profile and the dotted line shows the gas profile. While the dark matter is fairly consistent with the pure N-body profiles, the gas distribution has much larger cores. However, we cannot attribute this purely to lack of resolution: recent high-resolution simulations have also found this effect (Navarro, Frenk, & White 1995).[8]

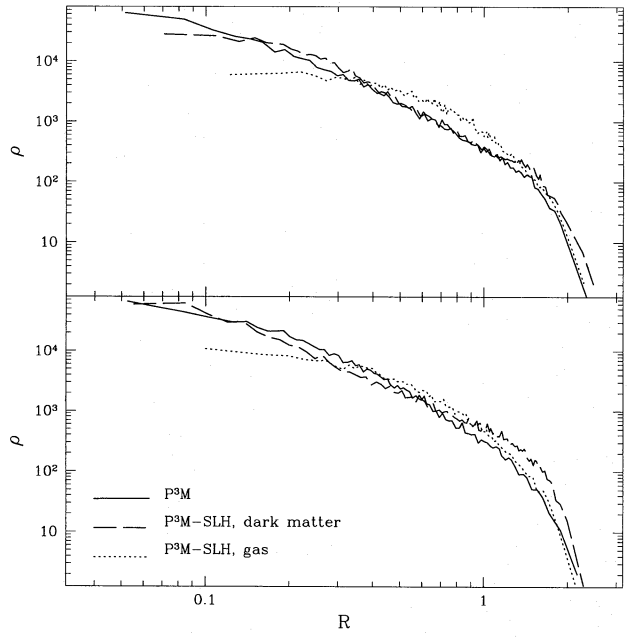

FIG. 7.—Density profiles for two largest clusters in the 64^3 SLH-P^{3}M $\Omega_X = 0$ simulation for dark matter (*dashed line*) and baryonic gas (*dotted line*); shown with the solid line is the density profile for the same cluster from the pure N-body (P^{3}M only) simulation with the same initial conditions.

[8] In real clusters, gas cores are also much larger than the dark matter cores as shown by comparison of X-ray pictures and lensing reconstruction (see Miralda-Escudé 1995); we do not use this fact as an argument, since it is not proved that our tests possess adequate resolution and physics to simulate real clusters.

We thus conclude that the agreement is acceptable and proceed to test the new SLH-P^{3}M code further.

5. COMPARISON WITH OTHER CODES

In this section, we repeat all the major tests of Kang et al. (1994) to assess differences between the repaired SLH code and other hydrodynamic codes. We performed three simulation with 32^3, 64^3, and 128^3 mesh sizes with initial conditions of Kang et al. (1994).[9] The total energy error in terms of the quantity R defined in Ryu et al. (1993) for the SLH-P^{3}M 64 run is plotted in Figure 8. One can see that the total energy error fluctuates around 1%, which is better

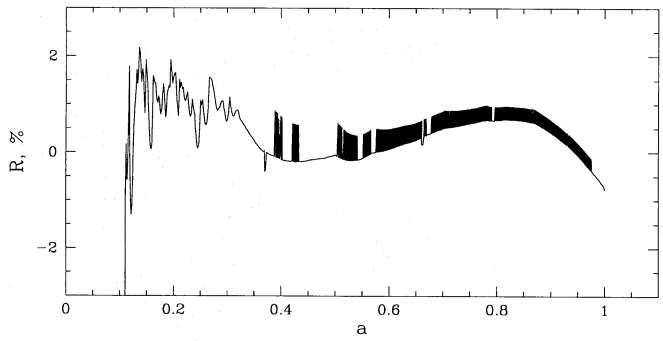

FIG. 8.—Total energy error for the SLH-P^{3}M 64 run as a function of scale factor a; the total error fluctuates around zero with amplitude 1% (except for $a < 0.2$ where the energy error is not computed accurately enough).

[9] These are as follows: Standard CDM model, $\Omega_b = 1$, 64 h^{-1} Mpc box size; initial conditions were defined on a 64^3 mesh; for the 32^3 simulation, every other value along each dimension was taken, and for 128^3 simulation, linear interpolation was used to determine initial values on a 128^3 mesh. Note that the latter procedure in fact adds some small-scale power to the initial conditions and that it is incorrect to say that the 128^3 SLH-P^{3}M simulation and the 128^3 and 256^3 Eulerian simulations of Kang et al. (1994) have the same initial conditions as the 64^3 simulations. In particular, the lack of convergence in the Kang et al. (1994) Eulerian simulations is due, at least partially, to this effect. However, we show results of a 128^3 simulation as a comparison with larger Eulerian simulations since the initial conditions for our 128^3 simulation are similar to 128^3 Eulerian simulations of Kang et al. (1994).

than most existing cosmological hydrodynamic codes (see Table 1 of Kang et al. 1994 for comparison).

The final data for each simulation were rebinned onto a uniform 16^3 mesh, and integral properties of the resulting 16^3 solution were calculated. Figure 9 compares these against each other and against other codes shown in earlier work (Kang et al. 1994; Gnedin 1995). Plotted on four panels are the volume-averaged temperature $\langle T\rangle$, the mass-averaged temperature $\langle T\rangle_\rho \equiv \langle \rho T\rangle/\langle\rho\rangle$, the rms density fluctuation $\sigma \equiv \langle\rho^2\rangle/\langle\rho\rangle^2 - 1$, and the total X-ray luminosity $L_X = \sum \rho^2 T^{1/2}$ in some arbitrary units. Four other cosmological codes are presented together with the SLH code described in this paper: two Eulerian hydrodynamic codes, the Cen-Ostriker-Jameson code (Cen et al. 1990; *filled squares*) and the TVD code (Ryu et al. 1993; *filled circles*), and two SPH codes, Tree-SPH of Hernquist & Katz (1989; *open triangles*) and P^{3}M-SPH of Evrard (1988; *open circles*). The self-evident notation used to designate these codes is fully explained in Kang et al. (1994). The bold solid line with open stars tracks the results of the original SLH code (which we shall hereafter call SLH-MMG as an abbreviation for SLH–moving mesh gravity), and the bold solid line with solid circles shows the results of the new SLH-P^{3}M code. One can easily see that the SLH-P^{3}M code provides a major improvement in resolution over the original SLH-MMG code, with resolution similar to or exceeding that of the SPH codes for a 64^3 grid of particles, while the temperature is similar to Eulerian codes. The SLH-P^{3}M code, therefore, manages to reach the high resolution of SPH codes while introducing less numerical heating. The SLH-P^{3}M code is also a factor of 2–3 faster than a SPH code for the same resolution.

Let us now compare original (unrebinned) data for different codes. Figure 10*a* shows a slice of original data one cell thick of the SLH-P^{3}M 64 run rebinned onto a 256^3 mesh (to facilitate the comparison with the TVD 256 run); the slice has 256^2 points and is 0.25 h^{-1} Mpc thick. Figure 10*b* shows the same slice for SLH-P^{3}M 128. Also shown are subslices containing one of the largest clusters found in the whole slice.

These slices are to be compared with corresponding plots from Gnedin (1995); however, in order to ease comparison, we plot subslices shown on the right-hand side of Figures 10*a* and 10*b* together with corresponding subslices for TVD

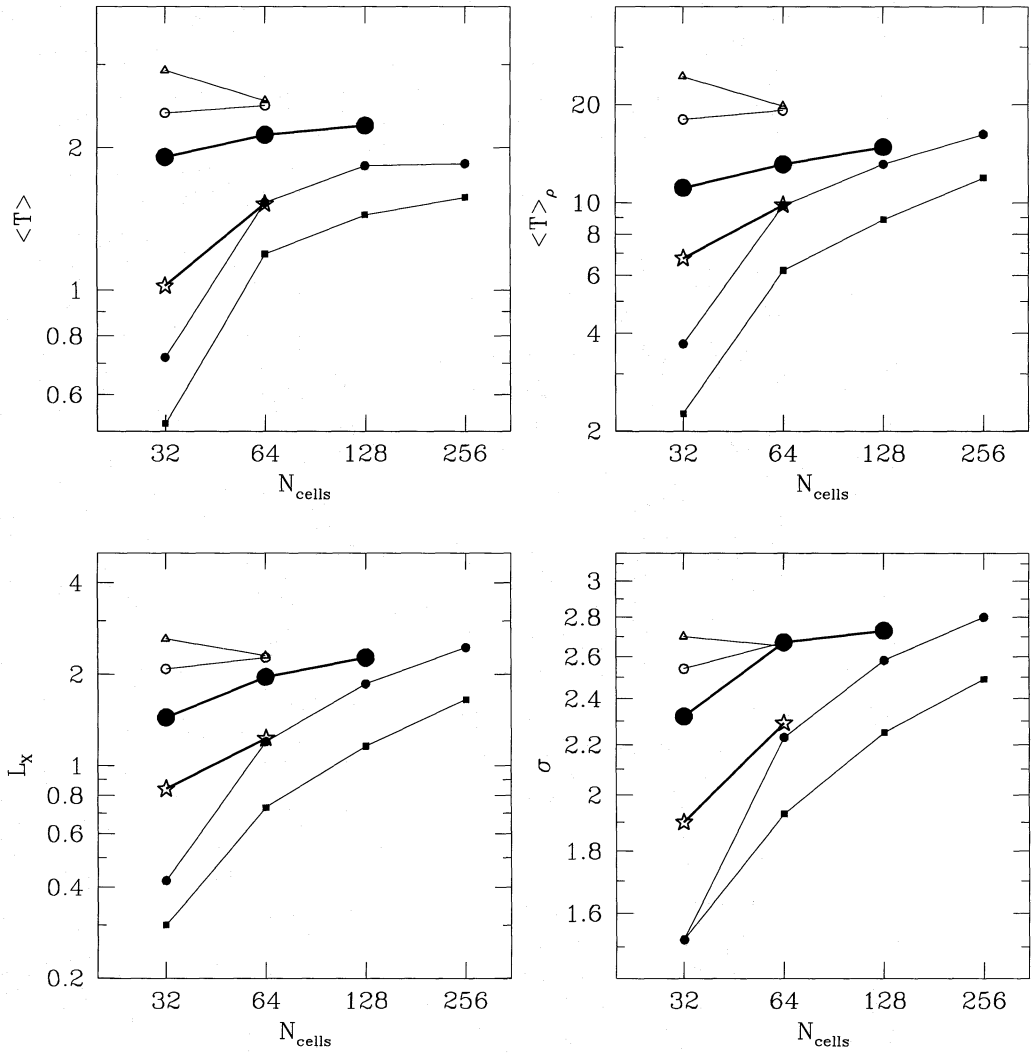

FIG. 9.—Comparison of integrated quantities: volume-averaged temperature $\langle T\rangle$ (*upper left-hand panel*), mass-averaged temperature $\langle T\rangle_\rho$ (*upper right-hand panel*), X-ray luminosity L_x (*lower left-hand panel*, in arbitrary units), and rms density fluctuations σ (*lower right-hand panel*), for the SLH-P^{3}M approach (*solid circles connected with bold solid line*) to the other existing codes: original SLH (*stars connected with bold solid line*), COJ (*filled squares*), TVD (*small filled circles*), PSPH (*open circles*), and TSPH (*open triangles*).

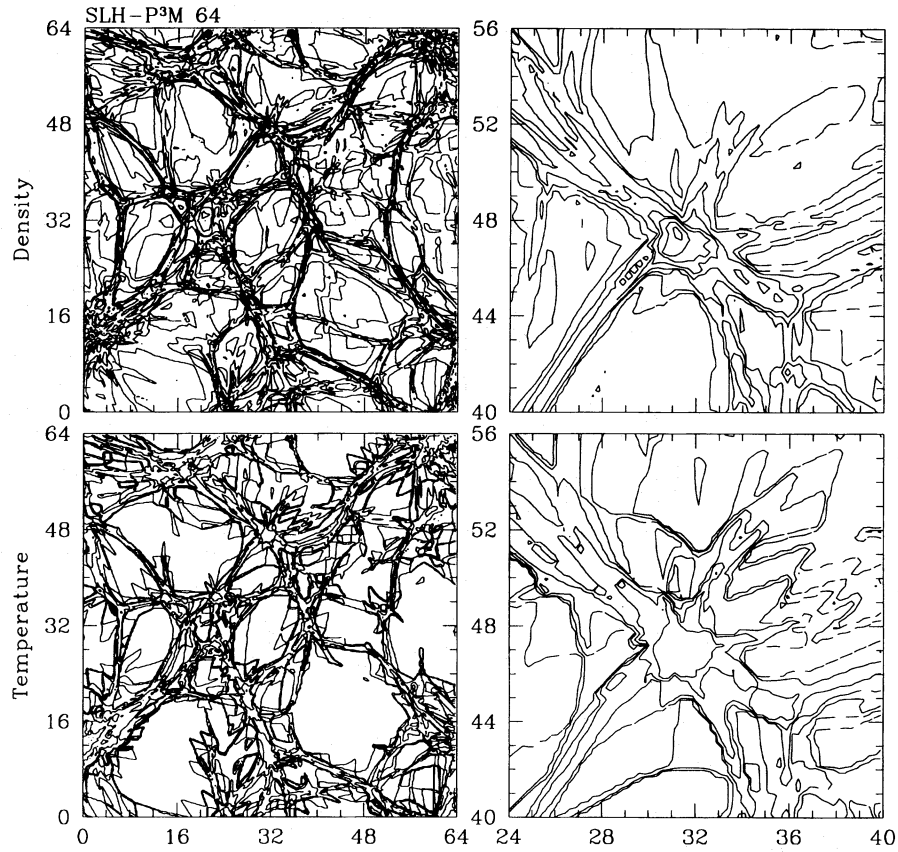

FIG. 10*a*

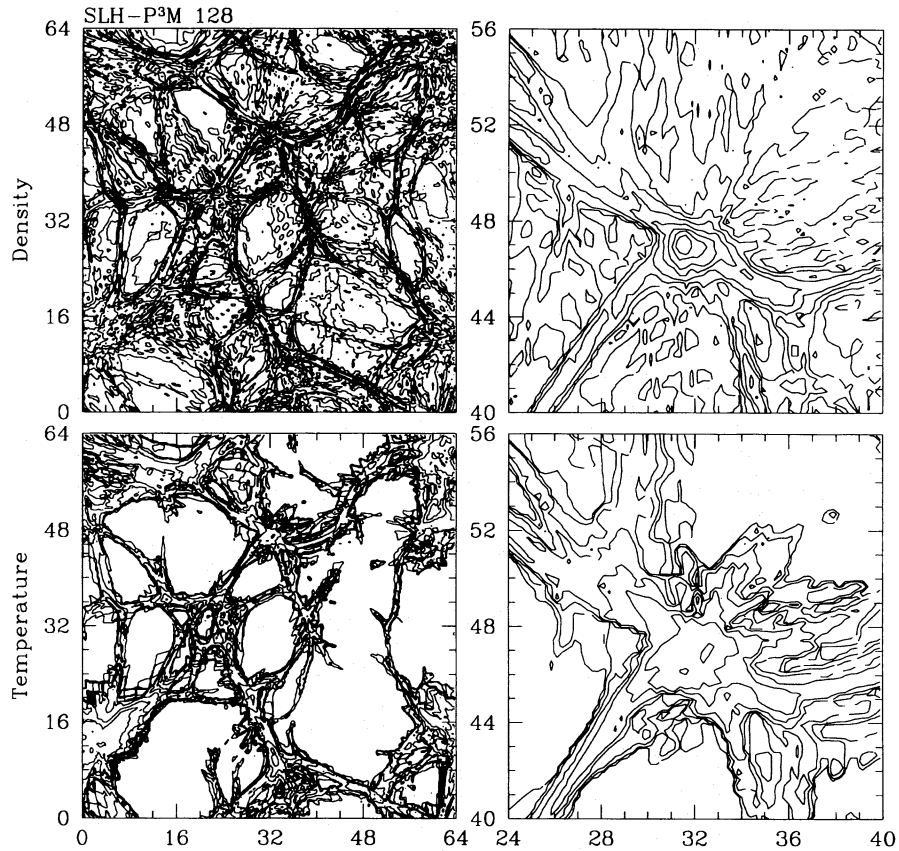

FIG. 10*b*

FIG. 10.—A slice of original (256^3 rebinned) data for SLH-P^3M 64 (*a*) and SLH-P^3M 128 (*b*) runs. The top row shows density, and the bottom row shows temperature; the right column shows a zoom of 1/16 of the whole slice ($24 < x < 40, 40 < y < 56$) containing one of the largest clusters present in the slice.

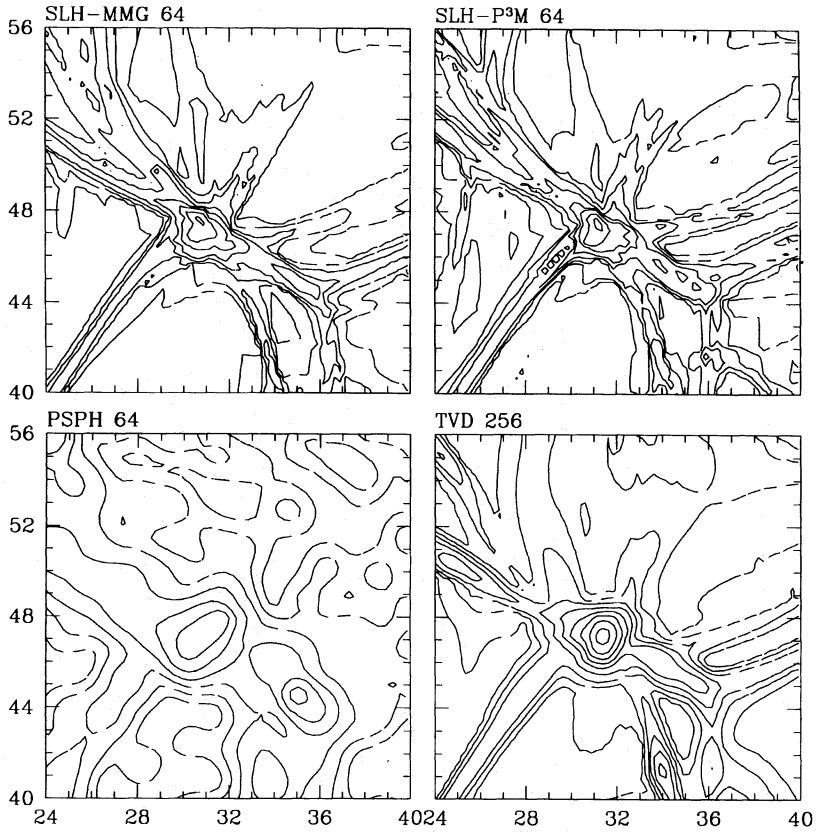

FIG. 11*a*

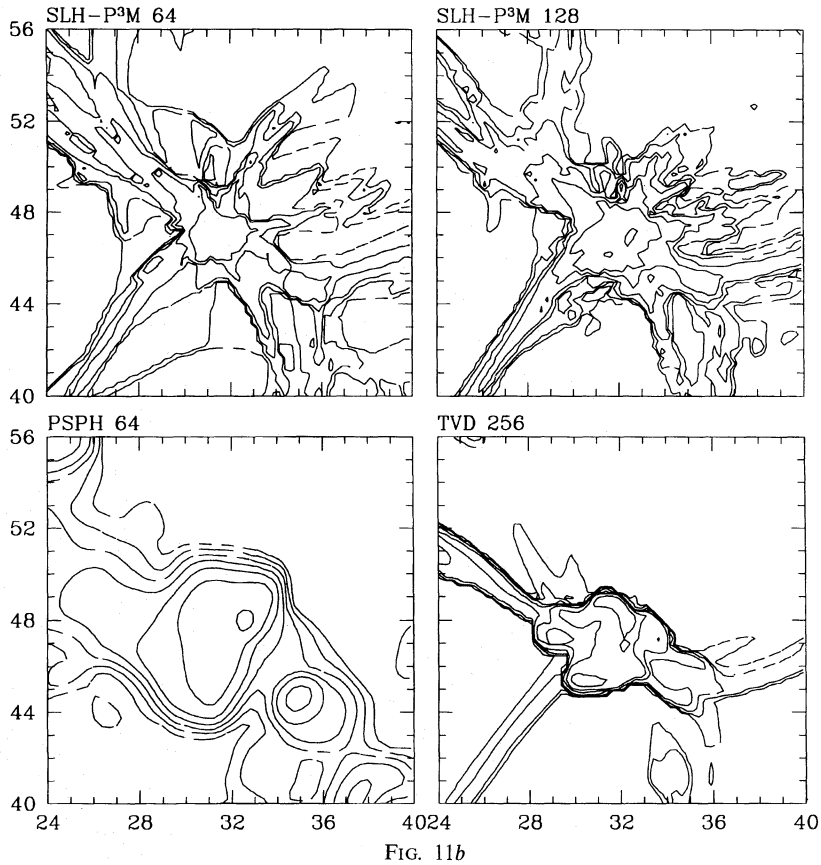

FIG. 11*b*

FIG. 11.—Zooms of the density contours of 1/16 of the whole 256 × 256 slice ($16 < x < 32$, $16 < y < 32$) for four different codes: SLH-MMG, 64 (*upper left-hand panel*), PSPH 64 (*lower left-hand panel*), SLH-P³M 64 (*upper right-hand panel*), and TVD 256 (*lower right-hand panel*): (*a*) the density distribution; (*b*) the temperature distribution.

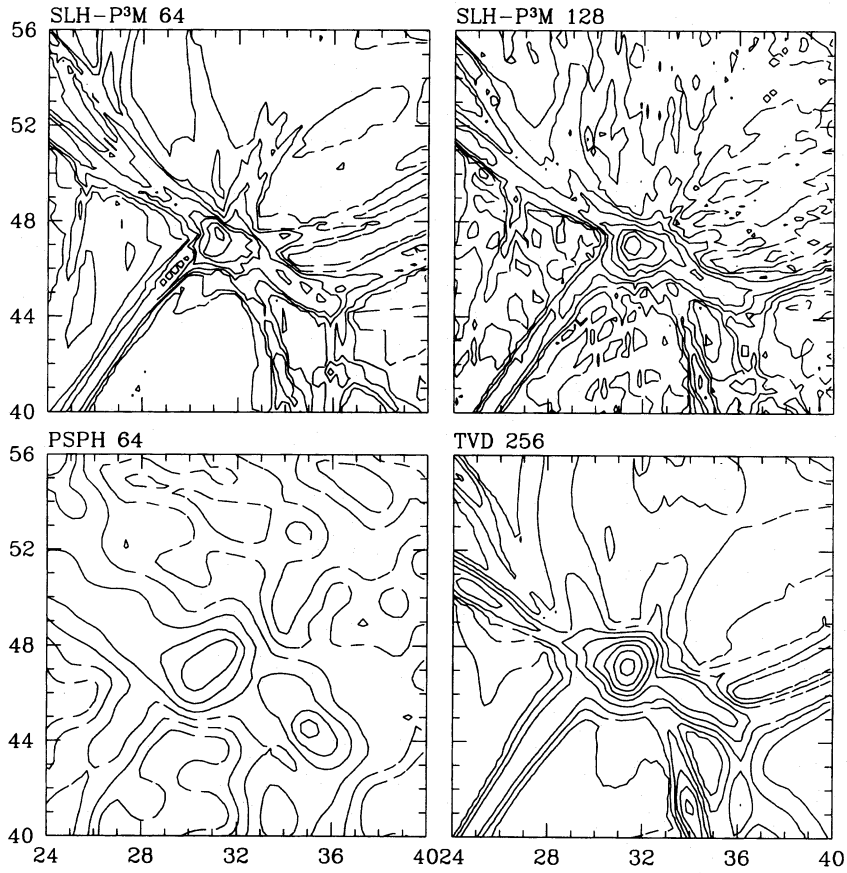

FIG. 12*a*

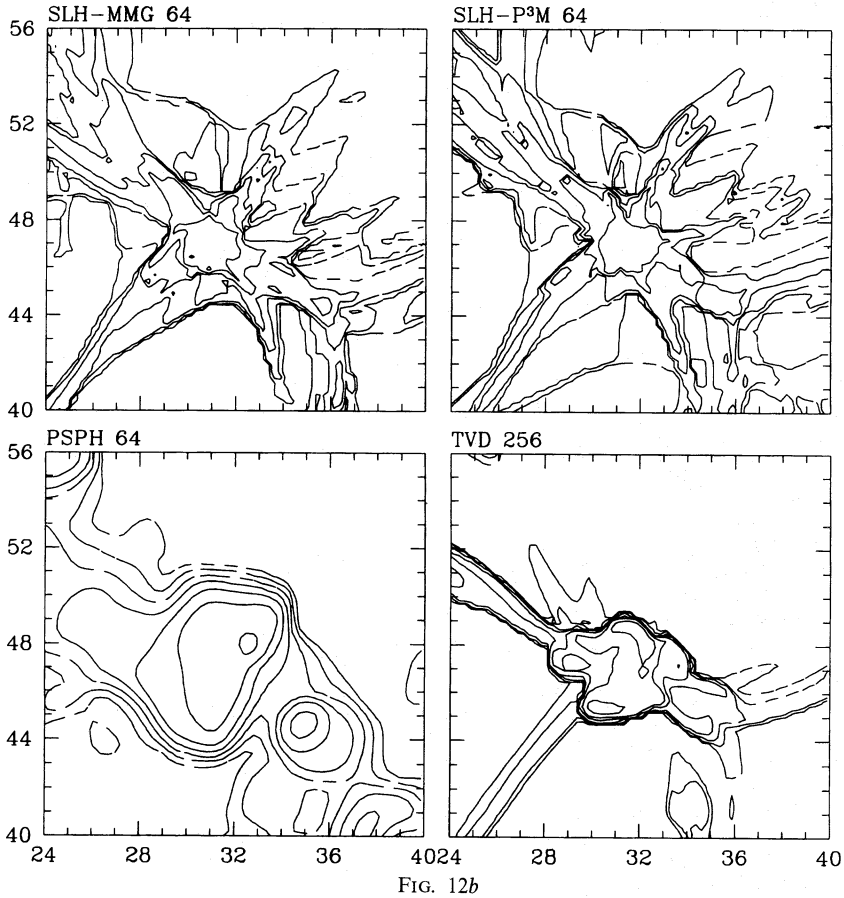

FIG. 12*b*

FIG. 12.—The same as Fig. 11 except SLH-MMG,64 and SLH-P³M 64 are replaced by SLH-P³M 64 and SLH-P³M 128, respectively

256, PSPH 64, and SLH-MMG 64 simulations separately in Figure 11 and Figure 12; both figures show PSPH 64 and TVD 256 results on the bottom; Figure 11 shows comparison between the original SLH-MMG 64 simulation (*upper left-hand panel*) and the SLH-P^{3}M simulation (*upper right-hand panel*), and Figure 12 plots SLH-P^{3}M 64 and SLH-P^{3}M 128 simulations in the upper row to demonstrate improvement and convergence.

One first notes that there are no significant differences visible between the original SLH-MMG 64 and SLH-P^{3}M 64 results; the SLH-P^{3}M 64 density distribution resembles that of TVD 256 slightly better, but shocks seem to be more blurry since the higher resolution of the SLH-P^{3}M code leads to fewer cells being left in voids and in the vicinity of shocks. The comparison between SLH-P^{3}M 64 and SLH-P^{3}M 128 shows little difference in the density distribution but substantial improvement in the shock thickness. Also, the SLH-P^{3}M 128 result has somewhat higher temperatures and shock extent because the gas was heated to higher entropy by the extra small-scale power.

It is quite instructive to compare the SLH-P^{3}M results to SPH and TVD slices. One can notice here that, except in the highest density regions, the real resolution of an SPH simulation is low;[10] it is hardly possible to identify filaments at all, whereas they are easily visible in the SLH and TVD plots. In order to understand this somewhat unexpected conclusion, one must recall that the hydrodynamic quantities in the SPH method are computed by averaging over, say, 15 nearest neighbors,[11] while in the SLH approach, they are defined in every cell and do not require any averaging to compute. Since in the vicinity of filaments the SLH mesh closely follows the gas flow and, therefore, the hydrodynamical description is close to Lagrangian, the SPH description of the gas is similar to the SLH description being smoothed over 15 neighboring cells, which roughly corresponds to one-fifth to 1/10 of the whole slice size depending on the density. More than that, only at overdensities above 15 does the SPH method acquire higher resolution than the Eulerian hydrodynamic code with the same number of cells, and the TVD 256 run becomes inferior in spatial resolution to the SPH 64 run only at overdensities in excess of $15 \times 64 \approx 1000$.

In general, the SLH-P^{3}M 128 run compares better to the TVD 256 run than both SLH-P^{3}M 64 and PSPH 64. In some places, the correspondence between the former two runs is quite good, but there are also places where the two codes differ substantially. We therefore conclude that in detail, differences between existing hydrodynamic codes are still substantial, and much work is required in order to achieve the same level of computational accuracy and consistency as is observed within collisionless N-body codes.

[10] The SPH resolution in the highest density regions may be somewhat reduced by remeshing, which was used to produce these plots; however, this note is only applicable to the very centers of large lumps and does change our conclusions about the SPH resolution in the most of the volume.

[11] In real SPH implementations, the number of neighbors is usually larger, from 30 to 60, but the averaging is weighted over the kernel, and not all neighbors contribute equally; we arbitrarily describe an SPH with 15-neighbor averaging just to emphasize the differences in the approaches.

6. SUMMARY

We have demonstrated that moving mesh gravity solvers suffer from two serious limitations: the shot noise in high-density regions and incorrect Green's functions. We have proposed a way to solve the former problem and have failed to identify the source of the latter completely. Since our moving mesh gravity solver cannot be trusted now, we have proceeded further by building a combined SLH-P^{3}M code, in which the SLH hydrodynamic solver is combined with the P^{3}M method for computing gravity.

In order for a self-gravitating hydrodynamic code to conserve energy, a special "gravitational consistency condition" ought to be satisfied. We have derived the gravitational consistency condition for the SLH-P^{3}M code and have showed what errors would appear should this condition be violated. Ignoring this condition can lead to serious errors in dense regions.

Finally, we have compared the SLH-P^{3}M code with existing cosmological hydrodynamic codes including the original SLH code. The SLH-P^{3}M approach offers a substantial increase in resolution, making it comparable to the SPH codes in the highest density regions, while maintaining reasonable accuracy in resolving shocks and hydrodynamic features. In particular, we have repaired the biggest problem of the original SLH method: failure to simulate clusters accurately, which stem out of a faulty moving mesh gravity solver.

We therefore conclude that the new SLH-P^{3}M code is a good numerical tool to investigate the formation of nonlinear structure in the universe including both gas and dark matter.

Finally, we note here that the gas-only tests like that of Kang et al. (1994), even if they are far from "realistic" cosmological simulating including the dark matter, are very useful since they emphasize problems of different hydrodynamic schemes; "realistic" cosmological simulations will highlight new possible problems, such as artificial momentum transfer between the dark matter and the gas, but they will simultaneously shadow purely hydrodynamic differences, in particular, the lack of the gravitational consistency, since usually the gas contributes a small fraction, less than 10%, of the total mass and, therefore, is not a dominant source of the total energy.

We are thankful to J. P. Ostriker for many fruitful discussions and to the referee Greg Bryan for many important clarifications; parallelization of the SLH-P^{3}M code for a shared memory multiprocessor was improved with suggestions from F. Summers. Supercomputer time was provided by the National Center for Supercomputing Applications. This work was supported by NSF grant AST-9318185 awarded to the Grand Challenge Cosmology Consortium.

APPENDIX

SOFTENING TENSOR

At a general point in the flow, the softening tensor is a function of the deformation tensor A^i_k;[12] let

$$\bar{g}^{ij} \equiv A^i_k \delta^{kl} A^j_l \tag{32}$$

be the conjugate metric of quasi-Lagrangian space and let $\lambda_g(\bar{g}^{ij})$ be its eigenvalues. Since $\bar{g}^{ij}$ is a real symmetric matrix, one can introduce a unitary matrix C^i_a such that

$$\bar{g}^{ij} = C^i_a \text{ diag } [\lambda_g]^{ab} C^j_b \,, \tag{33}$$

where diag $[\lambda_g]^{ab}$ denotes a diagonal matrix with λ_g entries.

We can now introduce the softening tensor σ^{ij} built according to the following formula:

$$\sigma^{ij} = C^i_a \text{ diag } [\sigma(\sqrt{\lambda_g})]^{ab} C^j_b \,, \tag{34}$$

where $\sigma(\lambda)$ is defined by the following equation:

$$\sigma(\lambda) = \frac{1}{1 + (\lambda/\lambda_*)^2} \,, \tag{35}$$

where λ_* is a softening parameter. In two opposite limits, when all three λ_g approach zero or infinity, the softening tensor tends to zero or δ^{ij} correspondingly.

[12] Note that the definition of the deformation tensor used here differs from that in the original SLH code of Gnedin (1995); both definitions are similar, but the one we used here generally leads to a somewhat smoother mesh and to a slightly smaller number of time steps in a simulation; the resolution provided by both definitions is the same.

REFERENCES

Bardeen, J. M., Bond, J. R., Kaiser, N., & Szalay, A. S. 1986, ApJ, 304, 15
Bertschinger, E. 1991, in After the First Three Minutes, ed. S. Holt, V. Trimble, & C. Benett (New York: AIP), 297
Bertschinger, E., & Gelb, J. 1991, Computers in Phys., 5, 164
Bryan, G. L., Cen, R., Norman, M. L., Ostriker, J. P., & Stone, J. M. 1994, ApJ, 428, 405
Bryan, G. L., Norman, M. L., & Ostriker, J. P. 1995, Comput. Phys. Commun., 89, 149
Cen, R., Jameson, A., Liu, F., & Ostriker, J. P. 1990, ApJ, 362, L41
Cen, R., & Ostriker, J. P. 1992a, ApJ, 393, 22
———. 1992b, ApJ, 399, L113
———. 1993a, ApJ, 417, 404
———. 1993b, ApJ, 417, 415
Davis, M., Efstathiou, G., Frenk, C. S., & White, S. D. M. 1985, ApJ, 292, 371
Efstathiou, G., Davis, M., Frenk, C. S., & White, S. D. M. 1985, ApJS, 57, 241
Evrard, A. E. 1988, MNRAS, 235, 911
Evrard, A. E., Summers, F. J., & Davis, M. 1994, ApJ, 422, 11
Frenk, C. S., Evrard, A. E., White, S. D. M., & Summers, F. J. 1995, ApJ, submitted (astro-ph/9504020)
Gelb, J. M., & Bertschinger, E. 1994a, ApJ, 436, 467
Gelb, J. M., & Bertschinger, E. 1994b, ApJ, 436, 491
Gnedin, N. Y. 1995, ApJS, 97, 231
———. 1996a, ApJ, 456, 1
———. 1996b, ApJ, 456, 34
Hernquist, L., Hut, P., & Makino, J. 1993, ApJ, 402, L85
Hernquist, L., & Katz, N. S. 1989, ApJS, 64, 715
Hockney, R. W., & Eastwood, J. W. 1981, Computer Simulations Using Particles (New York: McGraw Hill)
Kang, H., Ostriker, J. P., Cen, R., Ryu, D., Hernquist, L., Evrard, A. E., Bryan, G. L., & Norman, M. L. 1994, ApJ, 430, 83
Katz, N., Hernquist, L., & Weinberg, D. H. 1992, ApJ, 399, L109
Kofman, L. A., Gnedin, N. Y., & Bahcall, N. A. 1993, ApJ, 413, 1
Miralda-Escudé, J. 1995, preprint astro-ph/9509077
Navarro, J. F., Frenk, C. S., & White, S. D. M. 1995, MNRAS, 275, 720
Ostriker, J. P., & Steinhardt, P. J. 1995, Nature, 377, 600
Park, C. 1990, MNRAS, 242, 59P
Peacock, J. A., & Dodds, S. J. 1994, MNRAS, 270, 1020
Pen, U. 1995, ApJS, 100, 269
———. 1996, in preparation
Ryu, D., Ostriker, J. P., Kang, H., & Cen, R. 1993, ApJ, 414, 1
Summers, F. J., Davis, M., & Evrard, A. E. 1995, ApJ, 454, 1
Xu, G. 1995, ApJS, 98, 355