THE ASTROPHYSICAL JOURNAL SUPPLEMENT SERIES, 70:419–446, 1989 June

TREESPH: A UNIFICATION OF SPH WITH THE HIERARCHICAL TREE METHOD

LARS HERNQUIST
Institute for Advanced Study

AND

NEAL KATZ
Princeton University Observatory
Received 1988 September 19; accepted 1988 November 23

ABSTRACT

A new, general-purpose code for evolving three-dimensional self-gravitating fluids in astrophysics, both with and without collisionless matter, is described. In TREESPH hydrodynamic properties are determined using a Monte Carlo–like approach known as smoothed particle hydrodynamics (SPH). However, unlike most previous implementations of SPH, gravitational forces are computed with a hierarchical tree algorithm. A unification of these two techniques is attractive for a number of reasons, suggesting that TREESPH will be considerably more flexible than other similar numerical schemes.

First, since SPH and the hierarchical tree method are both gridless, TREESPH is fully Lagrangian and does not impose any artificial restrictions on the global geometry of the systems under study or any mesh-related limitations on the dynamic range in spatial resolution. In addition, TREESPH will be able to handle a variety of boundary conditions, depending upon the application.

Second, the dynamic nature of the data structures used to manipulate the particle distributions naturally permits the use of individual particle smoothing lengths and individual particle time steps. Both of these features have been incorporated into TREESPH, so it is adaptive in both space and time. As a result, the development of large density inhomogeneities can be resolved dynamically and TREESPH is not limited globally by local stability criteria such as the Courant condition.

Third, since the thermodynamic state of the gas is completely specified, processes such as radiative cooling and star formation can be prescribed in a physical manner.

Finally, SPH and the hierarchical tree method are both efficient for large particle number, N, with costs per step scaling as $\sim O(N)$ and $\sim O(N \log N)$, respectively. Consequently, simulations with large N, and correspondingly high resolution, are now feasible.

Subject headings: hydrodynamics — numerical methods

I. INTRODUCTION

Gas dynamical processes are observed to play an important role in the evolution of astrophysical systems on virtually all length scales. Individual stars are well-described as fluid spheroids held together by their own self-gravity. The evolution of close binaries is driven by viscous dissipation resulting from tidal perturbations. The interstellar medium consists largely of gas, being continually depleted and replenished by the births and deaths of stars. Hydrodynamic flows onto neutron stars and black holes may provide the energy input necessary to power compact X-ray and γ-ray sources as well as active galactic nuclei. On larger scales, gas has been detected in galactic cooling flows, X-ray coronae surrounding many cluster ellipticals, and in the cosmic jets frequently associated with extragalactic radio sources. Finally, it is widely believed that dissipation was an important ingredient in the formation of galaxies and large-scale structure in the universe.

Analytic treatments of hydrodynamical effects in astrophysics are typically restricted to problems possessing a high degree of symmetry with other simplifying assumptions (e.g., planar shocks or spherical blast waves) or to situations in which linear perturbation theory can be applied. More generally, numerical techniques are required.

To date, the most detailed computations have been made with codes which solve the hydrodynamical equations on a grid using finite-difference methods. This approach has a number of important advantages, including the capability of handling arbitrarily large gradients in the fluid. However, owing to memory and CPU limitations most finite-difference codes have been restricted to two spatial dimensions or have suffered from poor resolution and have typically not computed the gravitational potential self-consistently.

An alternate scheme, smoothed particle hydrodynamics (SPH), was introduced by Lucy (1977) and Gingold and Monaghan (1977) to circumvent the limitations of grid-based codes. In SPH, fluid elements constituting the system are sampled and represented by particles, and dynamical equations are obtained from the Lagrangian form of the hydrodynamic conservation laws. Owing to its particulate nature, SPH is fully Lagrangian and is most naturally implemented in three dimensions. There is no grid to constrain the dynamic range in spatial resolution or the global geometry of the

systems being studied. Finally, numerical tests suggest that SPH is less diffusive than comparable finite-difference methods (e.g., Lucy 1977; Gingold and Monaghan 1980; Nolthenius and Katz 1982; Durisen *et al.* 1986).

However, SPH is not without disadvantages. In its basic form it handles shocks using an artificial viscosity, imposing restrictions on the local spatial resolution. In addition, it is not possible to represent an arbitrarily large density gradient with a finite number of particles. Nevertheless, SPH has been successfully applied to a wide variety of astrophysical problems, being primarily limited in scope by the small number of particles used; typically $N \lesssim 1000$. Most previous implementations have computed gravitational forces by directly summing over all particles, resulting in a cost per step scaling as $\sim O(N^2)$. The CPU efficiency can be improved considerably through the use of a grid to evaluate the gravitational potential (e.g., Monaghan 1983; Evrard 1988), but not without compromising the Lagrangian nature of SPH.

Recently, a new class of N-body algorithms has been proposed which appears to be ideally suited for SPH. In the hierarchical tree method, multipole expansions are used to approximate the potential of distant groups of particles, reducing the cost per step to $\sim O(N \log N)$ and ultimately to $\sim O(N)$ in a more refined approach. More significantly, the improvement in efficiency has been achieved without the introduction of a grid. A unification of SPH with the hierarchical tree method is a natural way of allowing for larger N within a Lagrangian framework. Furthermore, the data structures used to manipulate the grouping of particles can be applied directly to certain aspects of the SPH calculation.

In this paper we describe one such unification. The basic methodology is summarized in § II. Efforts to optimize and vectorize the resulting code are discussed in § III. Simple numerical tests are given in § IV. Finally, conclusions and future prospects are outlined in § V.

II. METHODOLOGY

a) Smoothed Particle Hydrodynamics

i) *Basic Principles*

In SPH, a Lagrangian representation is adopted and the fluid is modeled as a collection of fluid elements. A real fluid consists of an infinite number of such elements, each infinitesimal in extent. It is not practical to model such a system computationally, however, and instead several approximations must be made. The standard procedure is to select a set of N fluid elements which are then represented by particles. The selection process assumes that the particle mass density is proportional to the mass density of the fluid, ρ. Thus, if the system is evolved according to the laws of hydrodynamics, ρ can be estimated from the local density of particles.

Since the computational model consists of a finite number of fluid elements, local averages must be performed over volumes of nonzero extent. This is conveniently accomplished by introducing a systematic procedure for smoothing out local statistical fluctuations in the particle number. The mean value of a physical field, $f(\mathbf{r})$, within a given interval can be determined through kernel estimation according to

$$\langle f(\mathbf{r})\rangle = \int W(\mathbf{r}-\mathbf{r}';h) f(\mathbf{r}')\, d\mathbf{r}', \tag{2.1}$$

where $W(\mathbf{r})$ is known as the smoothing kernel, the smoothing length, h, specifies the extent of the averaging volume, and the integration is over all space (e.g., Lucy 1977; Gingold and Monaghan 1977, 1982; Monaghan 1982). The smoothing kernel is normalized according to $\int W(\mathbf{r};h)\, d\mathbf{r} = 1$ as part of the condition resulting from the requirement that $\langle f(\mathbf{r})\rangle \to f(\mathbf{r})$ as $h \to 0$. (Note that many of these concepts are similar to those underlying the theory of distributions [e.g., Gel'fand and Shilov 1964].)

Typically, $W(\mathbf{r})$ is sharply peaked about $\mathbf{r} = 0$, in order that it resemble a delta-function as $h \to 0$. In such cases the error made in approximating $f(\mathbf{r})$ by its smoothed estimate can be determined by expanding the integrand of equation (2.1) as a power series in h. If $W(\mathbf{r})$ is an even function, then the leading term vanishes and $\langle f(\mathbf{r})\rangle = f(\mathbf{r}) + O(h^2)$. For the special case of a spherical kernel, the interpolation errors are

$$\langle f(\mathbf{r})\rangle = f(\mathbf{r}) + c\frac{h^2}{6}\nabla^2 f + O(h^3), \tag{2.2}$$

where $c = \int u^2 h^3 W(u)\, d\mathbf{u}$ is independent of h (e.g., Gingold and Monaghan 1982). Consequently, $f(\mathbf{r})$ can always be replaced by its smoothed equivalent to within the order of accuracy of the smoothing process itself. This implies, for example, that

$$\left\langle \frac{A(\mathbf{r})}{B(\mathbf{r})} \right\rangle = \frac{\langle A(\mathbf{r})\rangle}{\langle B(\mathbf{r})\rangle} + O(h^2). \tag{2.3}$$

In principle, the quadratic error term in equation (2.2) can be eliminated if W is negative over some range of its arguments. This would appear to be dangerous if the sampling is coarse, and the discussion below will always assume that $W \geq 0$ everywhere and that the leading error is consequently $\sim O(h^2)$.

If the values of $f(\mathbf{r})$ are known only at a finite number of discrete points, distributed with number density $n(\mathbf{r}) = \sum_{j=1}^{N} \delta(\mathbf{r}-\mathbf{r}_j)$, then equation (2.1) can be evaluated by multiplying the integrand by $n(\mathbf{r}')/\langle n(\mathbf{r}')\rangle$, using relation (2.3), and integrating to give

$$\langle f(\mathbf{r})\rangle = \sum_{j=1}^{N} \frac{f(\mathbf{r}_j)}{\langle n(\mathbf{r}_j)\rangle} W(\mathbf{r}-\mathbf{r}_j;h). \tag{2.4}$$

In particular, if a mass m_j is associated with each fluid element then

$$\langle \rho(\mathbf{r})\rangle = \sum_{j=1}^{N} m_j W(\mathbf{r}-\mathbf{r}_j;h). \tag{2.5}$$

Equation (2.5) is subject to two distinct interpretations, which are analogous to the meanings of computational gather-scatter

operations. The more traditional point of view, which we choose to refer to as the "scatter" interpretation, assumes that each particle has a mass which is smeared out in space according to W and h. The density at any point in space is then found by summing the contributions from the density profiles of neighboring particles, as in Figure 1. Alternatively, in the "gather" interpretation, particles can be regarded as point markers in the fluid. Local properties at any point in space are obtained by sampling all neighboring particles and weighting the contribution of each according to W, also as in Figure 1. The distinction between these viewpoints vanishes if the smoothing length is the same for all particles, but becomes relevant if h is spatially variable.

The smoothing formalism provides a natural means for estimating gradients of the local fluid properties. By definition, $\langle \nabla f(\mathbf{r}) \rangle = \int W(\mathbf{r}-\mathbf{r}'; h) \nabla f(\mathbf{r}')\, d\mathbf{r}'$. Integrating by parts gives

$$\langle \nabla f(\mathbf{r}) \rangle = \int f(\mathbf{r}') \nabla W(\mathbf{r}-\mathbf{r}'; h)\, d\mathbf{r}', \qquad (2.6)$$

ignoring surface terms. Finally, for a finite sampling, equation (2.6) can be evaluated in a manner similar to that leading to equation (2.4), resulting in

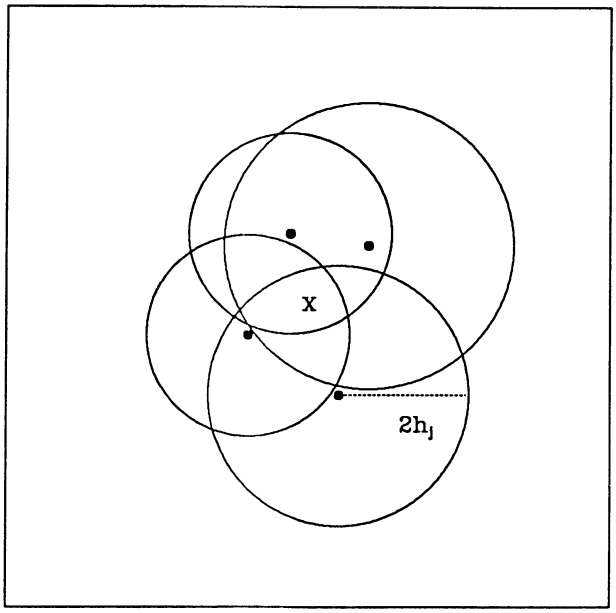
$$\langle \nabla f(\mathbf{r}) \rangle = \sum_{j=1}^{N} \frac{f(\mathbf{r}_j)}{\langle n(\mathbf{r}_j) \rangle} \nabla W(\mathbf{r}-\mathbf{r}_j; h). \qquad (2.7)$$

Higher order derivatives are computed in an analogous manner. Thus, gradients of the physical variables are replaced by derivatives of the smoothing kernel, thereby constraining $W(\mathbf{r})$ to be differentiable to at least the same order as that of the terms present in the dynamical equations. In practice, ∇W must exist unless, perhaps, the medium is pressureless, and $\nabla^2 W$ must exist if diffusion is included.

In order that the smoothed estimates retain a local character $W(\mathbf{r})$ should be sharply peaked and approach a delta function as $h \to 0$. As will be discussed in the section describing the calculation of the gravitational potential, a smoothing kernel with compact support has a number of advantages in the context of the hierarchical tree method. A natural choice is the spherically symmetric spline kernel proposed by Monaghan and Lattanzio (1985), defined by

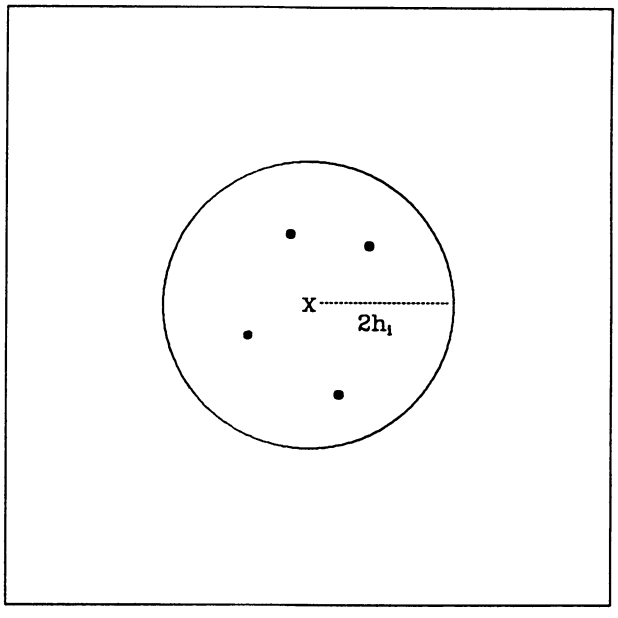
$$W(r,h) = \frac{1}{\pi h^3} \begin{cases} 1-(3/2)(r/h)^2 & \\ \quad +(3/4)(r/h)^3, & 0 \le r/h \le 1, \\ (1/4)[2-(r/h)]^3, & 1 \le r/h \le 2, \\ 0, & r/h \ge 2. \end{cases} \qquad (2.8)$$

In this case only those particles within $2h$ of a given point will contribute to the smoothed estimates. This kernel has continuous first and second derivatives and is second-order accurate, in the sense that $\langle f(\mathbf{r}) \rangle = f(\mathbf{r}) + O(h^2)$.



Scatter

FIG. 1a



Gather

FIG. 1b

FIG. 1.—Depiction of the sampling procedure in SPH. In (*a*) the "scatter" interpretation is illustrated, where the density at X is computed from the overlap of all particles within a distance $2h$, assuming the spline kernel defined by equation (2.8). In (*b*) the "gather" interpretation is demonstrated. Here, an observer at X is assumed to be active and to estimate the density by interrogating neighbors within a distance $2h$, weighting the contribution of each according to the smoothing kernel.

Schüssler and Schmitt (1981) note that kernels having the property $\nabla W \to 0$ as $r \to 0$, such as that in equation (2.8), may lead to unphysical clustering in non–self-gravitating systems. If this possibility becomes a concern, then the kernel (2.8) can simply be modified so that ∇W is finite at $r = 0$ by relaxing the constraint that ∇W be continuous through the origin. This obviates the need for kernels with divergent gradients at $r = 0$.

ii) *Smoothing Lengths*

The local spatial resolution is determined by the smoothing length, h. In its fundamental form, SPH assumes that h is the same for all particles. However, the local statistical fluctuations resulting from the kernel estimates are determined by the number of particles within $2h$ of a given point in space (assuming the smoothing kernel defined by eq. [2.8]). Thus a constant h would yield relatively more accurate estimates in regions with a high density of particles than in lower density regions and would furthermore not take full advantage of the particle distribution to resolve local structures. From the standpoints of both consistency and efficiency it is desirable to compute smoothed quantities to the same level of accuracy at all points in the fluid.

The SPH formalism can be generalized to allow for a spatially variable smoothing length. If one adopts the scatter interpretation of SPH then equation (2.1) is replaced by

$$\langle f(\boldsymbol{r})\rangle = \int W[\boldsymbol{r}-\boldsymbol{r}', h(\boldsymbol{r}')]\, f(\boldsymbol{r}')\, d\boldsymbol{r}' \tag{2.9}$$

implying that the discrete estimate of the density is computed according to

$$\langle \rho(\boldsymbol{r}_i)\rangle = \sum_{j=1}^{N} m_j W(r_{ij}, h_j), \tag{2.10}$$

where $r_{ij} = |\boldsymbol{r}_i - \boldsymbol{r}_j|$ (e.g., Gingold and Monaghan 1982; Nagasawa and Miyama 1987).

For an even, spherical kernel the interpolation errors in this case are

$$\langle f(\boldsymbol{r})\rangle = f(\boldsymbol{r}) + c_1 \frac{h^2(\boldsymbol{r})}{6}\nabla^2 f + c_2 \frac{\nabla h\cdot\nabla h}{6} f(\boldsymbol{r}) + c_3 \frac{h(\boldsymbol{r})}{3}\nabla f\cdot\nabla h + c_3 \frac{h(\boldsymbol{r})\nabla^2 h}{6} f(\boldsymbol{r}) + O(h^3), \tag{2.11}$$

where

$$c_1 = h^{-2}\int_0^\infty 4\pi u^4 h^5 W(u)\, du,$$

$$c_2 = \int_0^\infty 4\pi u^4 h^5 (\partial^2 W/\partial h^2)\, du,$$

$$c_3 = h^{-1}\int_0^\infty 4\pi u^4 h^5 (\partial W/\partial h)\, du,$$

and assuming $W(u) \equiv W(hu, h)$ (Monaghan 1987).

In the gather interpretation the smoothed estimates become

$$\langle f(\boldsymbol{r})\rangle = \int W[\boldsymbol{r}-\boldsymbol{r}', h(\boldsymbol{r})]\, f(\boldsymbol{r}')\, d\boldsymbol{r}' \tag{2.12}$$

and

$$\langle \rho(\boldsymbol{r}_i)\rangle = \sum_{j=1}^{N} m_j W(r_{ij}, h_i) \tag{2.13}$$

(e.g., Wood 1981, 1982; Benz 1984; Loewenstein and Mathews 1986; Evrard 1988). The interpolation errors for an even, spherical kernel are now

$$\langle f(\boldsymbol{r})\rangle = f(\boldsymbol{r}) + c_4 \frac{h^2}{6}\nabla^2 f + O(h^3), \tag{2.14}$$

where

$$c_4 = \int_0^\infty 4\pi u^4 h^3 W(u)\, du$$

is independent of h.

Both prescriptions have a number of conceptual difficulties. For example, Monaghan (1985) points out that the total system mass will not be conserved if computed in the gather interpretation using equation (2.13). This is technically correct. Note, however, that according to equation (2.14) the resulting errors are $\sim O(h^2)$ and are consistent with those already present in the smoothing procedure. Similar reservations apply to the scatter interpretation. However, from equation (2.11) the errors are again of the same order as those inherent in the SPH formalism.

Given that the errors introduced by spatially variable smoothing are of the same order in the gather and scatter formalisms, there appears to be no *a priori* reason to favor one over the other. A significant drawback to either scheme is that the receiver and transmitter do not appear on an even footing in the definition of smoothed estimates. If dynamical equations are then derived in the standard manner, this asymmetry will be manifested as a violation of Newton's third law. In order to conserve total momentum the equations of motion must be symmetrized in h_i and h_j.

One approach, which has been used extensively, is to symmetrize the dynamical equations by substituting $(h_i + h_j)/2$ for h_i or h_j, but leave the smoothed estimate of the density in asymmetric form. We have experimented with this approach and found that this inconsistency can introduce errors into the integration of the system. This can be most easily visualized for a smoothing kernel having compact support, as that in equation (2.8), where replacing h_i or h_j by $(h_i + h_j)/2$ has the effect of altering the list of nearest neighbors. The various smoothed estimates will then not be compatible if the number of neighbors changes significantly or if large gradients are present locally. Alternatively, for an isentropic, nondissipative gas, dynamical equations can be derived from a Lagrangian variational principle in either the gather or scatter formulation (e.g., Gingold and Monaghan 1982). The equations of motion will conserve momentum

since the *total* energy of the system depends symmetrically on h_i and h_j. However, this procedure cannot easily be generalized. In order to avoid these complications we decided to symmetrize the smoothing procedure itself.

In our new formalism, which is a hybrid of the gather and scatter interpretations, smoothed estimates are computed according to

$$\langle f(\mathbf{r})\rangle = \int f(\mathbf{r}')\frac{1}{2}\{W[\mathbf{r}-\mathbf{r}', h(\mathbf{r}')] + W[\mathbf{r}-\mathbf{r}', h(\mathbf{r})]\}\, d\mathbf{r}'. \tag{2.15}$$

Since this expression is simply a linear combination of the gather and scatter formulations, the interpolation errors will be the mean of equations (2.11) and (2.14), scaling as $\sim O(h^2)$. The application of equation (2.15) to the thermodynamic variables proceeds as before. For example, the discrete form of the smoothed density is

$$\langle \rho(\mathbf{r}_i)\rangle = \sum_{j=1}^{N} m_j \frac{1}{2}\left[W(r_{ij}, h_i) + W(r_{ij}, h_j)\right], \tag{2.16}$$

which has the correct limiting behavior if $h_i = h_j$.

The optimal procedure for updating the individual particle smoothing lengths has yet to be determined. Ideally, h should be proportional to the local mean interparticle separation, scaling as $h \propto n^{-1/3}$ in three dimensions. In TREESPH a two-stage technique is used to update h such that each particle interacts with roughly a constant number of nearby neighbors. First, if h_i^{n-1} is the smoothing length for particle i at step $n-1$, and the corresponding number of neighbors is $\mathcal{N}_i^{n-1}$, then a prediction for h_i^n is made from

$$h_i^n = h_i^{n-1}\frac{1}{2}\left[1 + \left(\frac{\mathcal{N}_s}{\mathcal{N}_i^{n-1}}\right)^{1/3}\right], \tag{2.17}$$

where $\mathcal{N}_s$ is an input parameter. Equation (2.17), which averages the old and new smoothing lengths, was found to be more stable than extrapolating h_i^n from $h_i^n = h_i^{n-1}(\mathcal{N}_s/\mathcal{N}_i^{n-1})^{1/3}$.

The current number of neighbors, $\mathcal{N}_i^n$, is then determined using this predicted value of h_i^n. If $\mathcal{N}_i^n$ is found to differ from $\mathcal{N}_s$ by more than a prescribed tolerance, typically a few percent, then h_i^n is corrected in such a way that $\mathcal{N}_i^n$ falls within the allowed range surrounding $\mathcal{N}_s$. This method is stable, by construction, and compels all smoothed estimates to be made to roughly the same level of accuracy at all points in the fluid.

In principle, a formalism based on the mean smoothing length $(1/2)(h_i + h_j)$ can also be developed (e.g., Evrard 1988). The resulting dynamical equations will then be symmetric in h_i and h_j, and hence momentum will be conserved. We prefer the smoothing procedure defined by equation (2.15), however, since the number of neighbors contributing to smoothed estimates, as in equation (2.16), can easily be controlled. If smoothing lengths are updated as outlined above, then a minimum of $\mathcal{N}_s$ neighbors will always contribute to the discrete sums. (Some particles may contribute to the smoothed estimates even if $2h_i < r_{ij}$ as long as $2h_j \geq r_{ij}$.)

iii) *Dynamical Equations*

With the procedure outlined above it is possible to derive smoothed forms of the hydrodynamical conservation laws which can then serve as equations of motion for the particles. If the density is computed according to the kernel estimates in equations (2.5), (2.10), (2.13), or (2.16) then the continuity equation will be satisfied automatically to terms $\sim O(h^2)$ and it need not be integrated forward in time. The phase space evolution of particle i is given by Euler's equation

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i, \tag{2.18}$$

$$\frac{d\mathbf{v}_i}{dt} = -\frac{1}{\rho_i}\nabla P_i + \mathbf{a}_i^{\rm visc} - \nabla\Phi_i, \tag{2.19}$$

where Φ_i is the gravitational potential, P_i is the pressure, and $\mathbf{a}_i^{\rm visc}$ is an artificial viscosity term to allow for the presence of shock waves in the flow. Note that these equations are structurally similar to those seen in N-body experiments, with the addition of terms to describe the local properties of the fluid.

Several forms have been used for the smoothed estimate of $(\nabla P)/\rho$. A symmetric expression can be obtained from the identity $(\nabla P)/\rho = 2\sqrt{P}\,(\nabla\sqrt{P})/\rho$, yielding

$$\frac{\nabla P_i}{\rho_i} = \sum_j m_j \frac{\sqrt{P_i P_j}}{\rho_i\rho_j}\left[\nabla_i W(r_{ij}, h_i) + \nabla_i W(r_{ij}, h_j)\right]. \tag{2.20}$$

Here, the factor $\sqrt{P_i P_j}/\rho_i\rho_j$ represents the geometric mean of P/ρ^2 for particles i and j. We prefer this symmetrized expression for $(\nabla P)/\rho$ to the arithmetic mean $(P_i/\rho_i^2 + P_j/\rho_j^2)/2$, which has typically been used in previous implementations of SPH. If large gradients are present then the arithmetic mean form can lead to instability in the integration of the thermal energy equation when the leapfrog scheme is used, resulting in negative thermal energies. The geometric mean appears to be more robust in this regard. For relatively smooth density distributions the arithmetic and geometric means yield results of comparable accuracy.

The introduction of an artificial viscosity is required for an accurate treatment of shocks (e.g., Monaghan and Gingold 1983; Monaghan and Pongracic 1985). A form suggested by Monaghan (1988*a*) has been adopted. Specifically,

$$-\frac{\nabla P_i}{\rho_i} + \mathbf{a}_i^{\rm visc} = -\sum_j m_j\left(2\frac{\sqrt{P_i P_j}}{\rho_i\rho_j} + \Pi_{ij}\right) \times \frac{1}{2}\left[\nabla_i W(r_{ij}, h_i) + \nabla_i W(r_{ij}, h_j)\right], \tag{2.21}$$

where Π_{ij} is the viscous contribution to the pressure gradient. A number of expressions have been proposed for Π_{ij}. One which gives an excellent description of the flow near shocks is

that given by Monaghan (1988*a*):

$$\Pi_{ij} = \frac{-\alpha\mu_{ij}\bar{c}_{ij} + \beta\mu_{ij}^2}{\bar{\rho}_{ij}}, \tag{2.22}$$

where

$$\mu_{ij} = \begin{cases} \dfrac{\boldsymbol{v}_{ij}\cdot\boldsymbol{r}_{ij}}{h_{ij}\left(r_{ij}^2/h_{ij}^2 + \eta^2\right)}, & \text{for } \boldsymbol{v}_{ij}\cdot\boldsymbol{r}_{ij} < 0, \\ 0, & \text{for } \boldsymbol{v}_{ij}\cdot\boldsymbol{r}_{ij} \geq 0. \end{cases} \tag{2.23}$$

Here, $\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$, $\bar{c}_{ij} = (c_i + c_j)/2$ is the average speed of sound of particles i and j, $h_{ij} = (h_i + h_j)/2$ and $\bar{\rho}_{ij} = (\rho_i + \rho_j)/2$. The first term in equation (2.22) is analogous to a bulk viscosity, while the second term, which is intended to suppress particle interpenetration at high Mach number, is similar to a Von Neuman–Richtmyer artificial viscosity (e.g., Richtmyer and Morton 1967). Typically, $\alpha \sim 1$ and $\beta \sim 2$ (Monaghan 1988*a*; Evrard 1988). The factor $\eta^2 \sim 0.01$ is inserted to prevent numerical divergences.

Equations (2.21)–(2.23) were found to give results similar to those obtained using the artificial viscosity proposed by Lattanzio *et al.* (1985, 1986) in simple shock tube and piston experiments, but they are more easily extended to situations in which the pressure can be negative; e.g., the Tillotson equation of state for metals (Benz, Slattery, and Cameron 1986, 1987; Monaghan 1988*b*).

An undesirable aspect of artificial viscosities such as that defined above is that they can introduce considerable shear viscosity into the flow. This problem can be reduced by using a viscosity depending on the divergence of the velocity field. Possible choices include a bulk viscosity (e.g., Lucy 1977; Sanders 1977) or a Von Neuman-Richtmyer artificial viscosity (e.g., Wood 1981, 1982). In TREESPH, we use the combined form

$$\Pi_{ij} = \frac{q_i}{\rho_i^2} + \frac{q_j}{\rho_j^2}, \tag{2.24}$$

where

$$q_i = \begin{cases} \alpha h_i \rho_i c_i |\nabla\cdot\boldsymbol{v}|_i + \beta h_i^2 \rho_i |\nabla\cdot\boldsymbol{v}|_i^2, & \text{for } \nabla\cdot\boldsymbol{v} < 0, \\ 0, & \text{for } \nabla\cdot\boldsymbol{v} \geq 0, \end{cases} \tag{2.25}$$

and $\nabla\cdot\boldsymbol{v}$ is estimated according to

$$\nabla\cdot\boldsymbol{v}_i = -\frac{1}{\rho_i}\sum_{j=1}^{N} m_j \boldsymbol{v}_{ij}\cdot\frac{1}{2}\left[\nabla_i W(r_{ij}, h_i) + \nabla_i W(r_{ij}, h_j)\right] \tag{2.26}$$

(Monaghan 1988*a*). As defined here, this artificial viscosity can lead to spurious results if $\nabla\cdot\boldsymbol{v}_j < 0$, for some particle j, and $\boldsymbol{v}_{ij}\cdot\boldsymbol{r}_{ij} > 0$; i.e., if particles i and j are receding. Such terms will contribute viscous *cooling* to particle i. To avoid this problem, we set $\Pi_{ij} = 0$ if $\boldsymbol{v}_{ij}\cdot\boldsymbol{r}_{ij} > 0$; a decision which may, in some cases, add small amounts of shear viscosity. Determining Π_{ij} by referring to binary interactions effectively hybrids the "molecular" and bulk fluid descriptions. Another such hybridization has recently been proposed by Balsara *et al.* (1988).

Although equations (2.24)–(2.26) contribute less shear viscosity to the fluid than do equations (2.22) and (2.23), they provide a somewhat less accurate description of the flow near shocks, as discussed in § IV*a*. In order to be able to study the sensitivity of the results to the choice of Π_{ij}, both artificial viscosities (2.22) and (2.24) were included in TREESPH and a switch was added to allow one or the other to be chosen.

Finally, an equation of state must be added to close the system of equations describing the evolution of the fluid. A convenient form can be obtained by writing the ideal gas law as

$$P = (\gamma - 1)\rho u, \tag{2.27}$$

where u is the specific internal energy. If the input $\gamma = 1$, then TREESPH instead defaults to the isothermal equation of state

$$P = c_{\rm iso}^2 \rho, \tag{2.28}$$

where $c_{\rm iso}$ is the isothermal sound speed.

The dynamical evolution of u is determined by the thermal energy equation, which is obtained from the first law of thermodynamics, $du = -PdV + Tds$. Here, $V = 1/\rho$ is the specific volume, and all nonadiabatic effects are included in the change in specific entropy, ds. The smoothed version of the thermal energy equation can be written in a variety of forms equivalent to second order in h. In TREESPH we choose to express the thermal energy equation as

$$\frac{du_i}{dt} = \sum_{j=1}^{N} m_j \left(\frac{\sqrt{P_i P_j}}{\rho_i \rho_j} + \frac{1}{2}\Pi_{ij}\right)\boldsymbol{v}_{ij} \cdot \frac{1}{2}\left[\nabla_i W(r_{ij}, h_i) + \nabla_i W(r_{ij}, h_j)\right] + \frac{\Gamma - \Lambda}{\rho}, \tag{2.29}$$

where $(\Gamma - \Lambda)/\rho$ accounts for nonadiabatic processes not associated with the artificial viscosity (e.g., Field 1965; Spitzer 1978). For an isothermal gas, equation (2.29) is not required.

As we indicated earlier, the geometric mean of P/ρ^2 for particles i and j tends to suppress an instability in the integration of the thermal energy equation which may result in negative thermal energies. However, equation (2.29) is still not infallible. Consider an inviscid, dissipationless fluid, experiencing only adiabatic heating and cooling. Further, suppose that nearly all particles are concentrated into a single dense region. If one of the remaining particles is isolated and rapidly receding from this dense region, its thermal energy, u_i, could become negative owing to adiabatic cooling if $u_i \ll u_j$, where u_j refers to neighboring particles contributing to equation (2.29). It is not surprising that SPH can fail in such a situation since a single particle cannot adequately describe

large gradients on scales smaller than its own smoothing length.

If du_i/dt is sufficiently large for u_i to become negative over the particle's current time step, then TREESPH replaces $\sqrt{P_iP_j}/\rho_i\rho_j$ by P_i/ρ_i^2, as in Evrard (1988), reducing the adiabatic term to $du_i/dt|_{\text{adiabatic}} = -P_i(\nabla\cdot v)_i/\rho_i$, by equation (2.26). This procedure will introduce errors $\sim O(h^2)$ into the conservation of energy since the thermal energy equation and equations of motion are no longer strictly compatible. In practice these errors are confined to a negligible fraction of the mass and hence will be unimportant. We prefer this approach to that chosen by Nagasawa, Nakamura, and Miyama (1988) who prevent negative thermal energies by introducing an ad hoc modification to the thermal energy equation for *all* particles.

In principle SPH can be generalized to include any physical effect that can be expressed in global conservation form. Previous generalizations have included the effect of thermal diffusion (e.g., Lucy 1977; Monaghan 1988*a*) and the presence of magnetic fields (e.g., Gingold and Monaghan 1977; Benz 1984; Phillips and Monaghan 1985; Phillips 1986*a*, *b*).

iv) *Dynamical Consequences of Variable Smoothing*

In reality, if the smoothing length is spatially and/or temporally variable, then expressions involving smoothed estimates of derivatives, such as equations (2.19), (2.20), and (2.29) will include terms proportional to ∇h and dh/dt. In this case, the correct form of the smoothed dynamical equations is obtained only if the continuum equations are first multiplied by W, an integration is performed over all space, and the resulting expressions are then integrated by parts. This procedure can be simplified through the use of the following relations between the smoothed estimates of the derivatives of a quantity A, $\langle dA/dt\rangle$ or $\langle\nabla_r A\rangle$, and the derivatives of the smoothed form of A, $d\langle A\rangle/dt$ or $\nabla_r\langle A\rangle$. In the scatter formulation,

$$\left\langle\frac{dA}{dt}\right\rangle = \frac{d\langle A\rangle}{dt} - \int A\frac{\partial W}{\partial h}\frac{dh(r')}{dt}\,dr' + \int A\frac{\partial W}{\partial h}(v'-v)\cdot\nabla_{r'}h\,dr' + \int[(v'-v)\cdot\nabla_{r'}A]\,W\,dr', \quad (2.30)$$

$$\langle\nabla_r A\rangle = \nabla_r\langle A\rangle - \int A\frac{\partial W}{\partial h}\nabla_{r'}h(r')\,dr', \quad (2.31)$$

while in the gather interpretation

$$\left\langle\frac{dA}{dt}\right\rangle = \frac{d\langle A\rangle}{dt} - \frac{dh(r)}{dt}\int A\frac{\partial W}{\partial h}\,dr' + \int[(v'-v)\cdot\nabla_{r'}A]\,W\,dr', \quad (2.32)$$

$$\langle\nabla_r A\rangle = \nabla_r\langle A\rangle - \nabla_r h(r)\int A\frac{\partial W}{\partial h}\,dr'. \quad (2.33)$$

For a mixed formalism, such as ours, the corrections will be a linear combination of the corresponding terms from equations (2.30)–(2.33).

The discrete dynamical equations are then obtained in the standard manner, as in the steps leading to equation (2.4), and, contrary to some existing claims, will generally contain terms proportional to dh/dt and ∇h in both formalisms. Typically, however, such terms are smaller than the dominant, physical terms by powers of $N^{-1/3}$ in three dimensions, and will, therefore, be unimportant for large N (e.g., Gingold and Monaghan 1982). Nevertheless, it should be emphasized that no systematic study of the effect of these terms exists and that their influence remains somewhat uncertain. (See, however, the discussion by Evrard [1988].)

b) The Hierarchical Tree Method

i) *Basic Principles*

The straightforward particle-particle (PP) technique for computing the gravitational attraction, in which the acceleration is determined by a direct sum, has a number of advantages over other potential solvers in the context of SPH. In particular, it is fully Lagrangian and does not use a grid. However, since the computing time per step scales as $\sim O(N^2)$ this approach is prohibitively expensive for large N.

Recently, a new class of N-body algorithms has been proposed that retains many of the advantages of the PP technique (e.g., Appel 1981, 1985; Jernigan 1985; Porter 1985; Barnes and Hut 1986). Rather than compromising the spatial resolution and/or imposing geometrical restrictions, the hierarchical tree method introduces explicit approximations into the calculation of the potential to improve efficiency.

Particles are first organized into a nested hierarchy of clusters or cells and multipole moments of each cluster or cell, up to a fixed order, are computed. The gravitational acceleration is obtained by allowing each individual particle to interact with various elements of the hierarchy, subject to a prescribed accuracy criterion. As a rule, the force from nearby particles is computed as a direct sum. The influence of remote particles is included by evaluating the multipole expansions of the clusters or cells which satisfy the accuracy requirement at the location of each particle. Typically, the number of terms in the expansions is small compared with the number of particles in the corresponding cluster or cell, and a significant gain in efficiency is realized.

The process of computing the potential from a truncated multipole expansion is equivalent to neglecting the detailed distribution of particles within a given cluster or cell, to a specified level of accuracy. This procedure is physically well-motivated. For example, the dynamics of the Earth-Moon system is relatively insensitive to the detailed mass distribution within each body. The approximation is also numerically well-motivated. Errors will always be present in particle simulations from round-off, truncation, and discreteness effects. Given these, it is not consistent to compute the force field to *arbitrarily high* precision.

If the accuracy criterion is such that the size of an unresolved cluster or cell increases in proportion to the distance from a given point, then a sum over N particles is replaced by

a sum over log N interactions, and the cost per step scales as $\sim O(N \log N)$. The CPU efficiency can be further improved by symmetrizing the force calculation between clusters or cells. In the fast multipole method (Greengard and Rokhlin 1987; Greengard 1987, 1988; Ambrosiano, Greengard, and Rokhlin 1988; Carrier, Greengard, and Rokhlin 1988) particles are organized into cells and then cell-cell interactions are computed prior to the force calculation. Once these have been determined the force on a single particle can be obtained in a time independent of N, resulting in an overall scaling $\sim O(N)$.

The construction and maintenance of the hierarchy and subsequent determination of interactions can be performed efficiently through the use of tree-structured data (e.g., Appel 1981, 1985; Jernigan 1985; Barnes and Hut 1986; Barnes 1986; Press 1986; Greengard 1987; Hernquist 1987; Porter and Jernigan 1987). Unlike fixed grids, tree structures are adaptive and do not restrict the global geometry or local spatial resolution. Thus the hierarchical tree method retains those aspects of the PP technique which are advantageous for SPH, but at a significant reduction in computing costs.

ii) *The Barnes-Hut Algorithm*

The Barnes-Hut method relies on a hierarchical subdivision of space into regular cubic cells. At each step, prior to the force evaluation, a tree structure is built to store the hierarchy. Each node in the tree is associated with a cubic volume of space containing a given number of particles. Each volume is subdivided into eight subunits (in three dimensions) of equal volume, which become the descendents of the original node in the tree structure. This process is continued recursively until each fundamental subcell contains either one or zero particles. Empty cells are not stored explicitly in the tree, thus the leaves represent volumes of space containing precisely one particle. As part of the tree-building procedure, the total mass, center-of-mass coordinates, and quadrupole moments are computed recursively for each cell.

The force on a given particle is determined by walking through the tree, beginning at the top of the hierarchy (i.e., the largest volume). At each step, the size of the current cell, s, is compared with its distance from the particle, d. If

$$\frac{s}{d} \leq \theta, \tag{2.34}$$

where θ is a fixed tolerance parameter, then the influence of all particles within the cell is computed as a single particle-cell interaction. Otherwise, the cell is subdivided by continuing the descent through the tree until either the tolerance criterion is satisfied or an elementary cell is reached. In this manner, all operations, including tree construction and force evaluation, can be performed in $O(N \log N)$ time.

c) *TREESPH*

i) *Basic Principles*

There appears to be a deep correspondence between SPH and the hierarchical tree method. Both techniques are fully Lagrangian and neither uses a grid. Tree codes are adaptive in nature since the tree structure is updated continuously according to the phase space evolution of the system. This property is shared by SPH and is most clearly demonstrated through the use of individual particle smoothing lengths.

The task of unifying SPH and the hierarchical tree method is relatively straightforward since their underlying principles are similar. This paper describes a code, TREESPH, which combines SPH with a version of the Barnes-Hut algorithm developed for supercomputers by Hernquist (1987). In so doing, it was possible to introduce refinements into each technique by exchanging concepts from one to the other, as had been advocated by Hernquist (1988). A number of these refinements are described below and in § III.

ii) *Nearest Neighbor Searching*

Since the kernel defined by equation (2.8) has compact support, only those particles lying within $2h$ of a given point in space will contribute to smoothed estimates there. Thus, a fundamental requirement of SPH is that nearest neighbor searching be performed efficiently. This has typically been achieved through the use of grids and linked lists (e.g., Hockney and Eastwood 1981; Monaghan 1985). This appears to be the optimal procedure if the search interval, $2h$, is the same for all particles and if the cell width in the search grid, Δ, satisfies $\Delta = 2h$. However, this approach will degrade if $\Delta < 2h$, which can occur if memory constraints are severe. Furthermore, the grid-linked list method cannot be generalized to the case of spatially variable h without violating the condition $\Delta = 2h_j$, for some particles j.

An alternative scheme is to use a tree structure to search for nearest neighbors through range searching (e.g., Sedgewick 1983, p. 335). The optimal data structures appear to be those with simple geometrical interpretations, such as the oct-tree used in the Barnes-Hut algorithm (e.g., Finkel and Bentley 1974; Bentley and Stanat 1975; Bentley and Friedman 1979).

The fundamentals of range searching using oct-trees can be summarized as follows. Suppose we want to find all neighbors within a sphere of radius $2h_j$ surrounding the particle j. First, enclose the sphere in a cube of length $4h$ per side. Then begin the tree descent as for the force calculation (§ IIa[ii]). At each level, check to see if the volume enclosed by the search cube overlaps that represented by the current node in the tree. If not, discontinue the descent down that particular path. Otherwise, subdivide the cell and proceed down to the next level. If the current node represents a particle, check to see if it lies within $2h$ of particle j, recording it as a neighbor if appropriate. Continue this procedure recursively until all paths in the tree are exhausted.

Several different searching algorithms were tested in the development of TREESPH. Typically, grid-based methods were a factor ~ 3–5 times faster than those using trees for constant smoothing lengths, in agreement with Benz (1988). A tree scheme such as that described above and in § III was ultimately selected, however, owing to its greater flexibility.

iii) *Softening of Gravitational Potential*

The process of "softening" the gravitational interaction has generally been introduced as an ad hoc modification to the Kepler potential at small separations. In most cases the

computed potential has been of the form $\Phi \propto (r^2+\varepsilon^2)^{-1}$, corresponding to a Plummer density profile, where ε is the softening parameter. However, it is far from clear that this is the best prescription since the acceleration implied by this choice converges slowly to the Kepler value, compromising the local spatial resolution.

An attractive alternative is to soften the gravitational interaction using the spline kernel (eq. [2.8]), since it has compact support. The acceleration will then be identical to the Kepler form for $r \geq 2\varepsilon$. This is highly advantageous for the hierarchical tree method. Strictly speaking, the multipole expansions used to represent the potential of distant clusters or cells assume that the bodies are point particles. If the particles are actually extended, then further error results from the multipole approximation, aside from truncation. Clearly it would be preferable for the softened acceleration to approach the Kepler form quickly. If spline softening is used then no additional error is introduced if the cluster or cell is further than 2ε from the particle on which to compute the force.

The gravity in TREESPH is softened according to the spline kernel (eq. [2.8]), and the relevant formulae can be found in the Appendix. The softening length is currently assumed to be constant in time and not necessarily equal to the smoothing length, h. For applications involving both gas and collisionless particles, each species can have its own softening length, $\varepsilon_{\rm gas}$ and $\varepsilon_{\rm coll}$, respectively, to improve accuracy in situations where segregation occurs. In such cases $\varepsilon_{\rm gas}$ and $\varepsilon_{\rm coll}$ are the same for all particles of a given type and the equations of motion are symmetrized as outlined in § II*a*(ii). We are currently exploring the possibility of allowing ε to be spatially variable. Note that the discussion here is not specific to SPH and that several of these considerations, e.g., different softening parameters for different components, may be useful for pure N-body simulations.

A final refinement could be made to improve the accuracy of the hierarchical tree computation of the potential. Bodies more distant than 2ε are equivalent to point particles if spline softening is assumed. This suggests that all clusters or cells within 2ε should be fully resolved into individual particles. Such an option has not yet been implemented into TREESPH, however, since there would then be a tendency for the code to degrade for highly clustered states as an increasingly large number of *particles* contribute to the list of interactions. This is not true of the standard implementation of the hierarchical tree method (e.g., Hernquist 1988; Bouchet and Hernquist 1988).

iv) *Time Integration*

In TREESPH, particle positions and velocities are updated using a standard leapfrog integrator, which is accurate to second order in the time step, Δt. For particle i

$$\boldsymbol{r}_i^{n+1/2} = \boldsymbol{r}_i^{n-1/2} + \Delta t \boldsymbol{v}_i^n + O(\Delta t^3), \quad (2.35a)$$

$$\boldsymbol{v}_i^{n+1} = \boldsymbol{v}_i^n + \Delta t \boldsymbol{a}_i^{n+1/2} + O(\Delta t^3), \quad (2.35b)$$

where the superscripts refer to the time step at which quantities are computed. For the SPH particles the acceleration depends on the velocity through the artificial viscosity, as in equations (2.22)–(2.23) or (2.24)–(2.25). In this case second-order accuracy is maintained only if the velocity is updated in two stages. First, a predicted estimate, $\tilde{\boldsymbol{v}}_i^{n+1/2}$ is obtained from

$$\tilde{\boldsymbol{v}}_i^{n+1/2} = \boldsymbol{v}_i^n + 0.5\Delta t \boldsymbol{a}_i^{n-1/2}. \quad (2.36)$$

This value of $\tilde{\boldsymbol{v}}_i^{n+1/2}$ is then used to compute the time-centered acceleration, $\boldsymbol{a}_i^{n+1/2}$, which, in turn, allows the velocity to be updated according to equation (2.35b).

If, as is assumed here, the long-range forces are computed approximately, then it is probably not advantageous to use a higher order integrator, since it would tend to amplify the random, noisy contribution to the acceleration. The leapfrog method does not suffer from this problem since the acceleration is computed only once per step. Generally, however, higher order integrators are more efficient (e.g., Benz 1986).

For SPH particles, the time step for explicit schemes such as the leapfrog integrator is limited by the Courant condition. If the time step is fixed, stability can be ensured by varying the smoothing length in accordance with this constraint (e.g., Evrard 1988). However, this is a limitation since large density contrasts can then not be resolved without resorting to a prohibitively small time step. Instead, TREESPH allows each particle to have its own time step, Δt_i, chosen to maintain stability, according to modified forms of the criteria derived by Monaghan (1988*a*). These read

$$\Delta t_i = \mathscr{C} \frac{h_i}{h_i|\nabla\cdot\boldsymbol{v}_i| + c_i + 1.2(\alpha c_i + \beta \max_j |\mu_{ij}|)}, \quad (2.37)$$

$$\Delta t_i = \mathscr{C} \frac{h_i}{h_i|\nabla\cdot\boldsymbol{v}_i| + c_i + 1.2(\alpha c_i + \beta h_i|\nabla\cdot\boldsymbol{v}_i|)} \quad (2.38)$$

for the artificial viscosities defined in equations (2.22)–(2.23) and (2.24)–(2.26), respectively, where $\mathscr{C} \sim 0.3$ is the Courant number. In equation (2.38), the term $\beta h_i|\nabla\cdot\boldsymbol{v}_i|$ is only included if $\nabla\cdot\boldsymbol{v}_i < 0$, whereas the additional term $h_i|\nabla\cdot\boldsymbol{v}_i|$ is used in both equations (2.37) and (2.38), irrespective of the sign of $\nabla\cdot\boldsymbol{v}_i$. This latter $h_i|\nabla\cdot\boldsymbol{v}_i|$ term, which is not demanded by the Courant condition, improves the accuracy of the time integration. Note that, in principle, the Courant criterion can be circumvented by using a fully implicit or barely implicit (e.g., Patnaik *et al.* 1987) integration scheme.

The thermal energy equation (2.29) is advanced along with the positions. In the absence of explicit sources or sinks of energy, the time integration of this equation is also limited by the Courant condition, and an explicit scheme is adequate. However, if Γ and Λ are nonzero, as is the case if radiative effects are important, then additional constraints are imposed on the time step. These will be much more severe than that resulting from the Courant condition if the time scales associated with Γ and Λ are much smaller than the dynamical time. To avoid such complications, the thermal energy equation is solved semi-implicitly in TREESPH using the trapezoidal rule,

$$u_i^{n+1/2} = u_i^{n-1/2} + \frac{\Delta t}{2}\left(\dot{u}_i^{n+1/2} + \dot{u}_i^{n-1/2}\right) + O(\Delta t^3), \quad (2.39)$$

in a manner similar to that adopted by Lucy (1977) and Monaghan and Varnas (1988).

In general, equation (2.39) is a nonlinear relation for $u_i^{n+1/2}$, depending upon the form of $\dot{u}_i^{n+1/2}$, which must be solved iteratively. If the right-hand side of equation (2.29) consisted solely of the source and sink terms, then this procedure would be completely straightforward since Γ and Λ only depend on u_i. Unfortunately, the first term on the right-hand side of equation (2.29) depends on $u_j^{n+1/2}$, for the neighboring particles. A completely rigorous inversion would then require the simultaneous solution of a coupled set of nonlinear equations. An approximate two-stage procedure approach, which proved to be stable, is instead used.

First, a prediction of the thermal energy, $\tilde{u}_i^{n+1/2}$, is obtained by solving equation (2.29) implicitly, but assuming the first term on the right-hand side of equation (2.29) at step $n+1/2$ to be equal to that at step $n-1/2$. The predicted value $\tilde{u}_i^{n+1/2}$ is then used, along with the predicted velocity $\tilde{\boldsymbol{v}}_i^{n+1/2}$ to estimate the adiabatic and viscous contributions to $\dot{u}_i^{n+1/2}$. Finally, equation (2.39) is again solved implicitly using both the predicted and old adiabatic and viscous terms. The explicit phase of this method is equivalent to a second-order accurate Runge-Kutta integrator. The solution of the nonlinear equation at both predictor and corrector phases is obtained using a hybrid of the Newton-Raphson and bisection techniques, and vectorization is maintained by looping over subsets of particles that have yet to achieve convergence. This semi-implicit treatment is similar to that suggested by Casulli and Greenspan (1984) for compressible fluids.

v) *Individual Particle Time Steps*

To make TREESPH more efficient in handling problems with multiple time scales, the equation of motion for each particle is integrated with its own individual time step, in a manner similar to that adopted by Porter (1985) and Ewell (1988). All individual particle time steps, Δt_i, are chosen to be a power-of-two subdivision of the largest system time step, Δt_s, such that

$$\Delta t_i = \frac{\Delta t_s}{2^{n_i}}, \tag{2.40}$$

where n_i refers to the time bin of particle i and Δt_s is an input parameter. No particle is allowed to have a time step larger than Δt_s, implying that $n_i \geq 0$. Particles can move to a larger time bin, i.e., smaller time step, at the end of their own time step, but are allowed to move to a smaller time bin, i.e., larger time step, only if that time bin is currently time synchronized with their own time bin. This ensures that the system will be time synchronized at the end of every largest time step.

The particle positions and the thermal energy equation are advanced with a time step

$$\Delta t_{\rm pos} = \frac{\Delta t_s}{2^{n_{\rm max}+1}}, \tag{2.41}$$

where $n_{\rm max}$ is the largest time bin currently occupied, and hence $\Delta t_{\rm pos}$ is half the smallest time step. This procedure ensures that the gas pressure and gravitational acceleration can be computed at the midpoint of each time step, as required by the time-centered nature of the leapfrog integration scheme. Also, to preserve time centering, the positions are advanced according to the most recently updated value of their velocity.

When particles change time bins their positions must be updated using an estimated midpoint velocity in order to preserve the second-order accuracy of the leapfrog integrator. Since this estimate need only be first-order accurate it can be obtained from the last value of the particle's acceleration. If the correction to the positions is made at the midpoint of the new time step, then

$$\boldsymbol{r}_{i,\rm corrected} = \boldsymbol{r}_{i,\rm uncorrected} - \left(1-\frac{1}{\chi}\right)\left(1+\frac{1}{\chi}\right)\frac{\Delta t_{i,\rm old}^2}{8}\boldsymbol{a}_i, \tag{2.42}$$

where

$$\chi = \frac{\Delta t_{i,\rm old}}{\Delta t_{i,\rm new}} \tag{2.43}$$

is the ratio of the old to new time step.

The technique to preserve second-order accuracy, introduced above, can also be used to make the leapfrog integrator self-starting. If, on entry, the positions and velocities are given at time step n, then a second-order accurate estimate for $\boldsymbol{r}_i^{n+1/2}$ follows from

$$\boldsymbol{r}_i^{n+1/2} = \boldsymbol{r}_i^n + \frac{1}{2}\Delta t \boldsymbol{v}_i^n + \frac{1}{8}\Delta t^2 \boldsymbol{a}_i^n. \tag{2.44}$$

Similarly, for diagnostic purposes, the positions and velocities can be time synchronized at time step $n+1$ with the correction

$$\tilde{\boldsymbol{r}}_i^n = \boldsymbol{r}_i^{n+1/2} + \frac{1}{2}\Delta t \boldsymbol{v}_i^{n+1} - \frac{1}{8}\Delta t^2 \boldsymbol{a}_i^{n+1/2}. \tag{2.45}$$

These additional terms guarantee that all operations will be performed to second-order accuracy, *without* the need for an additional higher order integration method.

vi) *Miscellaneous*

Since SPH is formally similar to a simple N-body method, TREESPH is naturally suited for applications involving both gas and collisionless matter. Examples include the formation of galaxies and large-scale structure in the universe and the evolution of disk galaxies, where the collisionless material consists of both stars and dark matter. In such cases, only a subset of the particles represent gas. Phase space coordinates for *all* particles are advanced using equations (2.35a) and (2.35b). For the collisionless matter, the acceleration includes only the gravitational contribution, and, of course, the thermal energy equation is not needed.

Since TREESPH is fully Lagrangian and does not require a grid it can naturally be adapted to handle a variety of boundary conditions. The version of TREESPH described

here assumes that the region surrounding the simulation volume is a vacuum, but we are currently generalizing the formulation to include both the quasi-periodic and fully periodic boundary conditions developed for tree codes by Bouchet and Hernquist (1989) and by Bouchet, Hernquist, and Suto (1989), respectively.

Unlike other codes which simulate gas dynamical effects in astrophysics by introducing an ad hoc prescription to describe dissipation, as in the so-called sticky particle method, TREESPH solves the hydrodynamic equations ab initio. Consequently, the correct continuum limit will be approximated, and physical processes depending upon the thermodynamic state of the gas can be specified rigorously. As an example, we are currently modifying TREESPH to include the effects of star formation for applications to galaxy formation using a simple Jean's criterion.

III. OPTIMIZATION

a) *Vectorization*

i) *Background*

In TREESPH a tree structure is used to compute the gravitational interaction and to find nearest neighbors for the SPH calculations. It is conceptually useful to regard the gravitational tree walk as a device for establishing a list of interactions; i.e., a collection of cells and particles with which a given particle interacts gravitationally. The procedures for determining interaction lists and nearest neighbor lists are similar, with the primary difference being in the criterion used to decide whether or not a descent is to be made to the next lower level in the hierarchy.

In the implementation of the Barnes-Hut hierarchical tree algorithm designed for vectorizing supercomputers by Hernquist (1987), interaction lists were established for single particles by walking through the tree from node to node. Given an interaction list, the subsequent force calculation was vectorized and an overall factor of ~ 3–4 improvement in efficiency was realized from vectorization and compiler optimization. However, the CPU usage was dominated by the tree descent which was not at all vectorized.

Contrary to popular belief, tree descents can be fully vectorized. All algorithms proposed thus far rely on the possibility of vectorizing gather and scatter operations, i.e., indirect addressing. This feature was not available to Hernquist (1987) on the MFECC Cray XMP, but exists on newer XMPs as well as on many other supercomputers.

ii) *Tree Descents*

Several different methods for vectorizing tree descents were incorporated at various times into TREESPH. Initially, a scheme suggested by Makino (1989*a*) was tried. His trick is to perform tree descents for many particles simultaneously and vectorize loops over particles. This scheme allows for the processing of long vectors, and on machines such as the Cyber 205 overall factor of ~ 5 improvements in performance can be obtained.

For present purposes, i.e., including individual particle time steps, an alternate approach (Hernquist 1989) was found to be preferable. Particles are processed individually, as in Hernquist (1987), but the walk through the tree no longer proceeds node by node but rather level by level. That is, the tolerance criterion, equation (2.34), is simultaneously applied to all relevant cells at a given level. Those cells at the current level satisfying the accuracy requirements are added to the list of interactions. The remainder are subdivided and their descendants are placed on the list of cells to be visited on the next level further down.

This procedure retains the structure of the original algorithm since the force on single particles is computed, which is an advantage when each particle has a unique time step. The vectors are typically short, and hence this approach would probably not be optimal for a Cyber 205. However, the amount of overhead required is negligible and a significant reduction in CPU costs resulted on a Cray XMP. The tree descent routines were sped up by a factor of ≈ 4–5, resulting in an overall gain of ≈ 2–3, depending on the number of terms retained in the multipole expansions (Hernquist 1989). Note that this technique is general and is not, in any way, specific to the SPH implementation of the hierarchical tree method.

The tree search for nearest neighbors can be vectorized in an analogous manner and can be further refined by noting that particles close together in space will have similar lists of neighbors (see Barnes 1989). Thus, it is efficient to locate neighbors for *groups* of particles. In this case, the searches are performed using range searching, as outlined in § II*c*(ii), but with a search volume chosen to enclose the smoothing intervals of all particles in the group.

In passing, we note that nearest neighbor searching using grids and linked lists can be vectorized through a generalization of the suggestion made by Makino (1988*a*) and by locating neighbors for groups of particles, as above. Thus all neighbors of a given cell containing many particles are determined, and vectorization is achieved by processing a large number or such cells simultaneously. This appears to be the optimal method if the search interval is the same for all particles and is equal in size to the grid spacing.

iii) *Tree Construction*

For geometric trees, such as those assumed here, the routines initializing the tree structure are fully vectorizable. The process of setting up pointers is vectorized by looping over all particles not yet placed in the tree (Makino 1989). At a given level in the hierarchy the appropriate direction in the tree for each remaining particle is determined from its spatial coordinates. Those found to be the sole occupants of a cell at the next lower level are added to the tree and removed from the list of particles in the loop. The procedure is repeated until all particles are in place.

Given the pointers, the calculation of the mass, center-of-mass coordinates, and quadrupole moments of each node can be vectorized by looping over all cells at the same level in the tree, starting with the level immediately above the particles (Hernquist 1989). Since the properties of a node depend only on the properties of its descendants, there will be no conflicts if the cells being processed are at the same level and this procedure starts at the deepest level. In this case, vectoriza-

tion gained a factor of ≈ 4 in the routine that establishes the node properties. Again, this procedure is not specific to the SPH implementation of the hierarchical tree method.

b) *Miscellaneous*

Several minor refinements were incorporated into TREESPH to improve efficiency. The smoothing kernel, equation (2.8) and other expressions derived from it, such as equations (A1) and (A2), are evaluated using look-up tables. Cray intrinsic functions are used to perform simple vector operations, such as the selection of elements of a vector according to a mask.

Finally, nearest neighbors are required at various stages during each time step. Rather than determining neighbors at each separate instance, the requisite searches are performed only once per step, and a data base of neighbors is maintained. The CPU costs are thereby reduced at the expense of additional core memory. Many of the arrays used by TREESPH are specific to SPH. Hence, memory requirements will be less severe for applications in which only a fraction of the particles represent gas, as in simulations involving collisionless matter. In fact, the total number of machine words, $N_{\rm words}$, used by TREESPH in three dimensions is

$$N_{\rm words} = 23N + 22N_{\rm nodes} + 17N_{\rm SPH} + 0.5\mathcal{N}_{\rm avg}N_{\rm SPH} + N_{\rm overhead}. \quad (3.1)$$

In this expression, $N_{\rm nodes}$ is the number of internal nodes in the tree structure, $N_{\rm SPH}$ is the number of SPH particles, $\mathcal{N}_{\rm avg}$ is the average number of neighbors per SPH particle, and the overhead, $N_{\rm overhead}$, is of order 250,000 words. As an example, for cases in which roughly half of the particles represent gas, approximately $(50–60)N$ words of memory are required. Typically, $\mathcal{N}_{\rm avg} \sim 30–40$, in which case the data base of neighbors will consume $10N$ words. In this context, spatially variable smoothing lengths offer the advantage of allowing $\mathcal{N}_{\rm avg}$ to be strictly controlled, even in situations involving large changes in density. In any event, TREESPH will probably still be CPU- rather than memory-limited.

IV. EMPIRICAL STUDIES

a) *Initialization*

Given the masses, phase space coordinates, and thermal energies of all particles, TREESPH optionally initializes densities and smoothing lengths. The latter are computed to satisfy the constraint that all particles interact roughly with $\mathcal{N}_s$ near neighbors, as discussed in § IIa(ii). In addition, TREESPH also smooths the initial velocity field and thermal energy profile of SPH particles according to

$$\langle \mathbf{v}_i \rangle = \frac{1}{\rho_i}\sum_{j=1}^{N} m_j \mathbf{v}_j \frac{1}{2}\left[W(r_{ij}, h_i) + W(r_{ij}, h_j)\right], \quad (4.1)$$

$$\langle u_i \rangle = \frac{1}{\rho_i}\sum_{j=1}^{N} m_j u_j \frac{1}{2}\left[W(r_{ij}, h_i) + W(r_{ij}, h_j)\right]. \quad (4.2)$$

The application of equations (4.1) and (4.2) to the initial realization ensures that the resulting evolution will not be adversely affected by sharp discontinuities or fluctuations present in the input data. Furthermore, smoothing the velocity field is necessary to remove unphysical dispersion on small scales. (Note, however, that the initial velocities of collisionless particles are *not* smoothed.) These considerations have typically been ignored in previous implementations of SPH (see, however, Monaghan 1988a). If TREESPH is to evolve previously smoothed data, as in a restart, then these initialization procedures are not invoked.

Initial conditions for SPH simulations can be generated using several procedures. The simplest method distributes particles randomly, according to a known density profile. Alternatively, particles can be placed on a uniform Cartesian grid. The density profile is then reproduced by applying an appropriate radial transformation to the coordinates of the particles (e.g., Evrard 1988).

Neither of these prescriptions will produce precise equilibrium initial conditions, owing to discreteness effects. Clearly, this difficulty is only a concern if the physical system to be modeled is time stationary. In such cases, the influence of fluctuations can be minimized by evolving the computational system with a frictional damping term in the equations of motion, as suggested first by Lucy (1977). The resulting models can then be used to study equilibrium systems. This procedure has been used together with TREESPH to construct fluid polytropes. The density profiles computed numerically agree with the analytic profiles to an accuracy similar to that achieved by Evrard (1988). Results of these tests will be reported elsewhere.

b) *Tests in One Dimension*

TREESPH was applied to a number of one-dimensional systems to demonstrate its ability to reproduce known analytic solutions. Extensive tests were performed with simple shock tube and piston problems.

For the shock tube, the initial conditions introduced by Monaghan and Gingold (1983) were adopted (see Sod 1978):

$$\begin{array}{llll} \rho = 1, & P = 1, & v = 0, & \text{for } x < 0; \\ \rho = 0.25, & P = 0.1795, & v = 0, & \text{for } x \geq 0. \end{array} \quad (4.3)$$

Dissipational effects, other than those associated with the artificial viscosity, were ignored. The ratio of specific heats was $\gamma = 1.4$.

The density, pressure, velocity field, and entropic function, $A(s)$, are shown in Figures $2a–2d$, using the artificial viscosity from equations (2.22)—(2.23), and in Figures $3a–3d$ for the artificial viscosity from equations (2.24)–(2.26). The function $A(s)$ is related to the specific thermal energy, u, by $A(s) = (\gamma - 1)u\rho^{1-\gamma}$. In both cases, 400 equal-mass particles, each with $m_i = 0.75/400$, were initially distributed in the range $-1 \leq x \leq 1$, in such a way as to satisfy equation (4.3). The smoothing length and time step were both fixed and equal to $h = 0.015$ and $\Delta t = 0.005$, respectively. (Examples with varying h and Δt are presented below.) For the artificial viscosity in equations (2.22)–(2.23), the parameters α, β, and η were chosen to be $\alpha = 1$, $\beta = 1$, and $\eta^2 = 0.1$, while for the

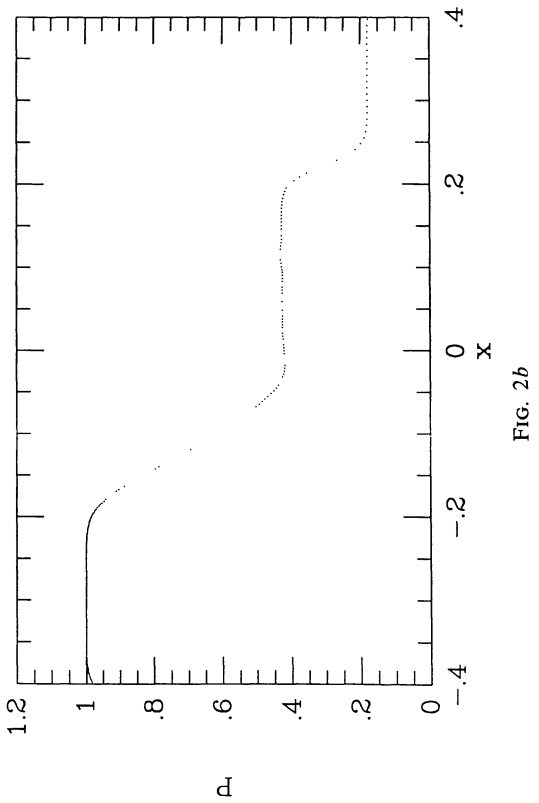


FIG. 2b

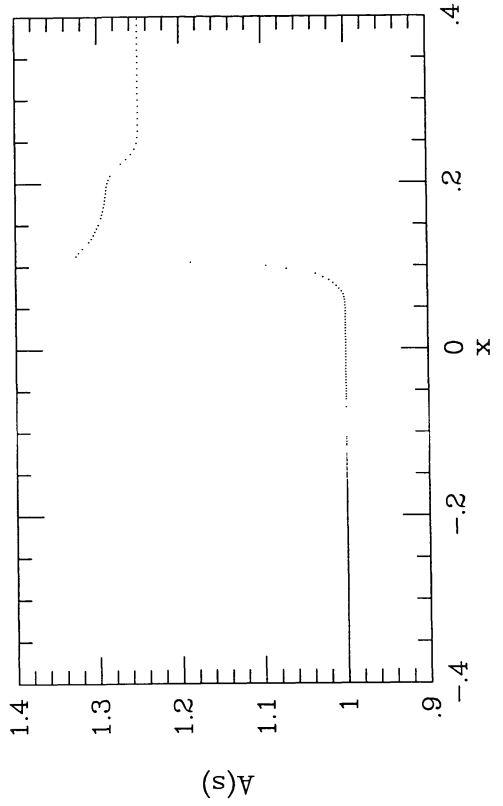


FIG. 2d

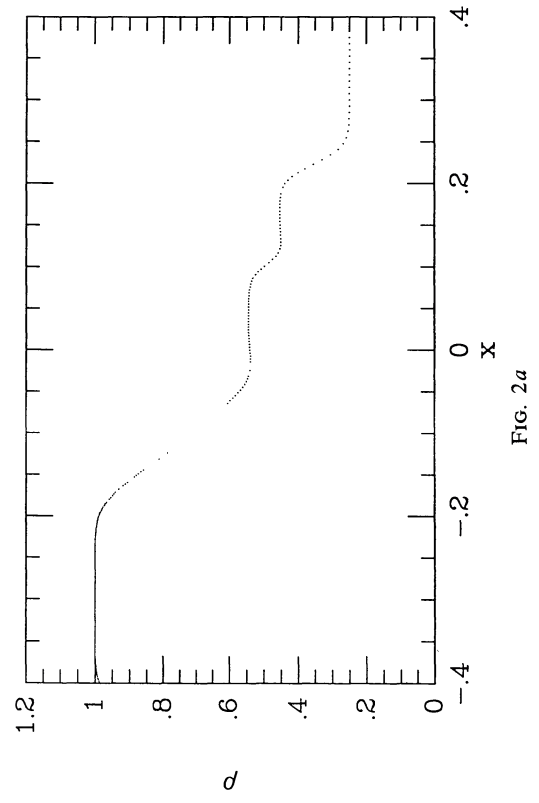


FIG. 2a

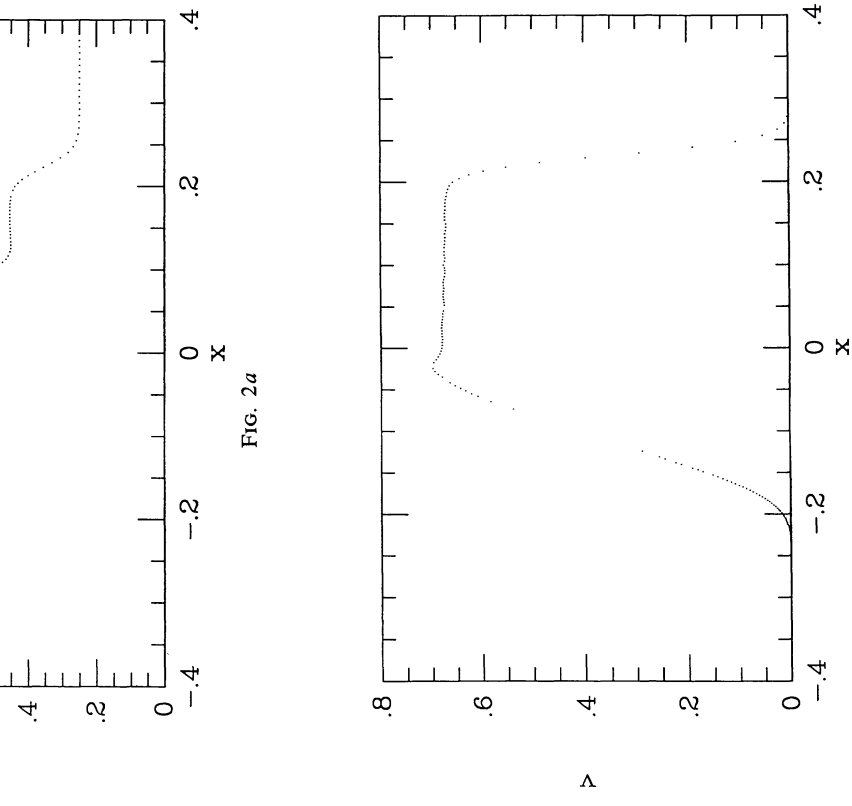


FIG. 2c

FIG. 2.—(*a*) Density, (*b*) pressure, (*c*) velocity, and (*d*) entropic profiles in a one-dimensional shock tube, using the standard SPH artificial viscosity defined by eqs. (2.22)–(2.23).

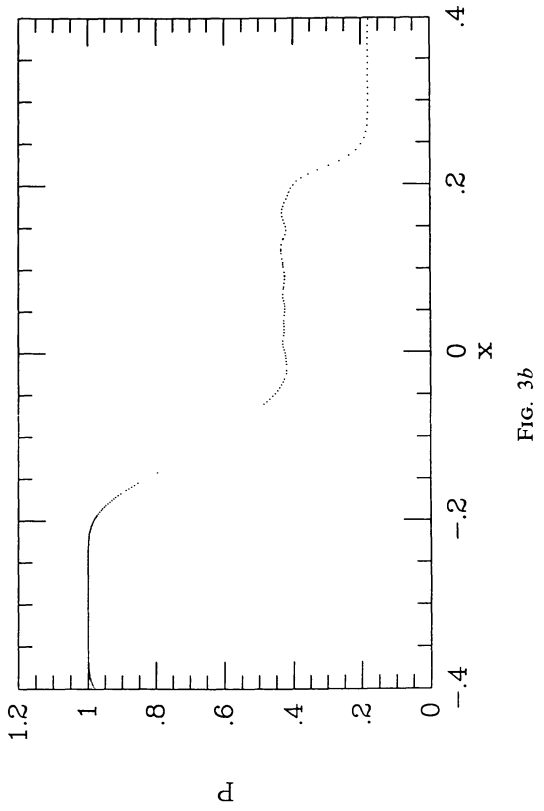


FIG. 3b

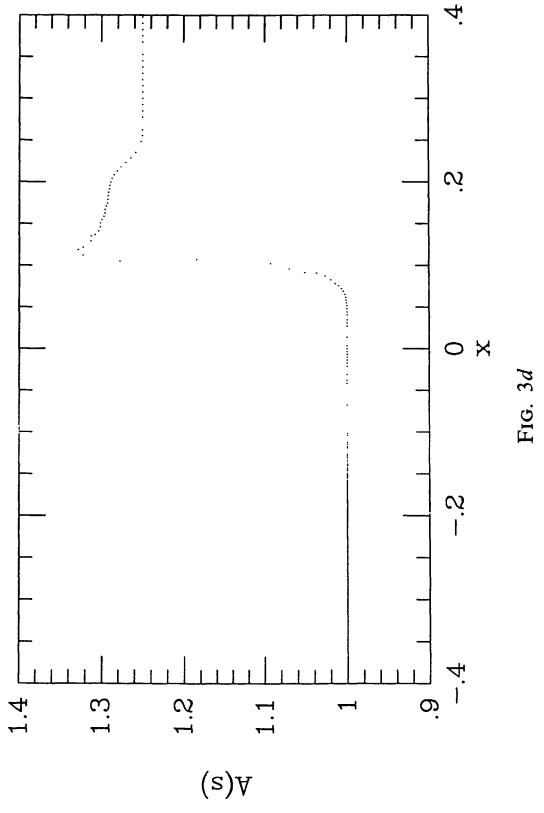


FIG. 3d

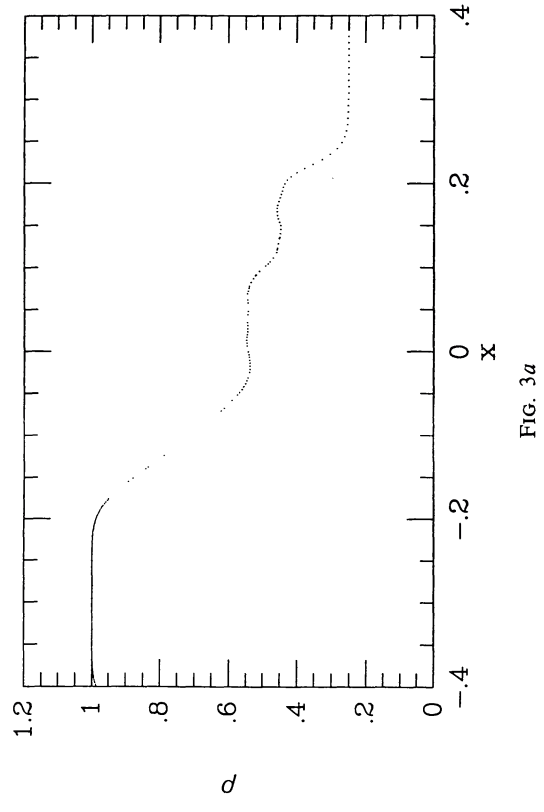


FIG. 3a

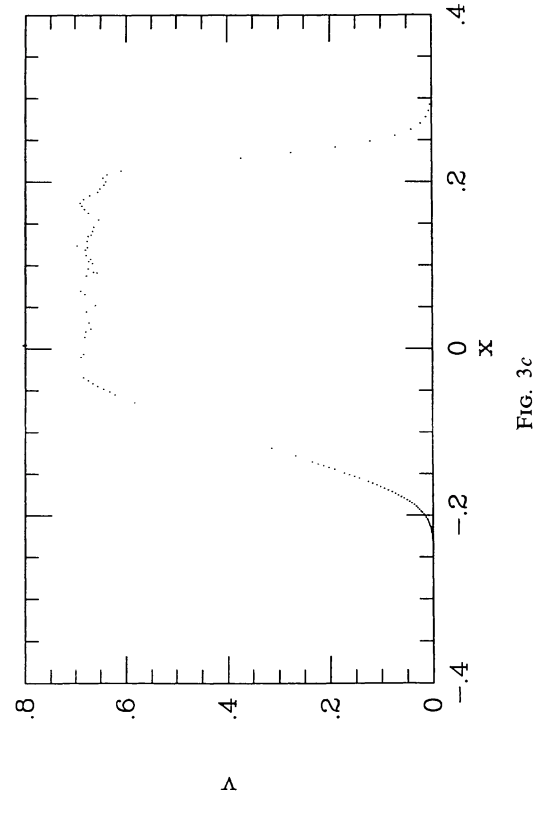


FIG. 3c

FIG. 3.—Same as Fig. 2, but using the bulk artificial viscosity, eqs. (2.24)–(2.26).

artificial viscosity in equations (2.24)–(2.26), $\alpha = 0.5$ and $\beta = 0.5$. Figures 2 and 3 show the system state at a time $t = 0.15$.

In both simulations, the shock, located between $x \approx 0.2$ and $x \approx 0.25$, is broadened over a range $\Delta x \approx 3h$. The contact discontinuity, centered around $x \approx 0.1$ in the density profile, has a similar width. This resolution is comparable to that which can be achieved with finite-difference codes using artificial viscosities (see Sod 1978). Better resolution can be obtained with smaller α and β, but at the expense of introducing additional postshock oscillations. The structure of the rarefaction wave, located between $x \approx -0.2$ and $x \approx 0$, is physical since the viscous terms do not contribute in expanding regions.

The results shown in Figure 3 are somewhat less satisfactory than those in Figure 2. The resolution is comparable in these two examples, but larger postshock oscillations can be seen in Figure 3, especially in the velocity field. As noted by Monaghan and Gingold (1983) artificial viscosities based on smoothed estimates of $\nabla \cdot v$ are incapable of completely damping these high-frequency vibrations. Note, however, that the results presented here are less noisy than similar previous calculations owing to the initial smoothing procedure (eqs. [4.1] and [4.2]) and our decision to turn off the viscous interaction between receding particles (§ II*a*[iii]). For situations in which an accurate resolution of shocks is critical, the artificial viscosity from equations (2.22)–(2.23) is preferable unless, as noted in § II*a*(iii), the implied unphysical contribution to the shear viscosity becomes a serious problem. In such cases it may be necessary to use the bulk viscosity from equations (2.24)–(2.26) and accept either lower resolution or slightly noisy velocity fields behind shocks.

Although reassuring, the examples in Figures 2 and 3 are not completely conclusive since the shocks are rather weak. For the parameters chosen above, the shock Mach number, $\mathcal{M}$, is approximately $\mathcal{M} \approx 1.4$. To examine the robustness of SPH in the high Mach number limit, another set of tests was performed by compressing a tube of gas with a piston.

As before, the gas was assumed to be dissipationless, aside from the artificial viscosity, with a ratio of specific heats $\gamma = 1.4$. The preshock density, ρ_1, and velocity, v_1, were $\rho_1 = 1$ and $v_1 = 0$, respectively, and the velocity of the piston, u_p, was $u_p = 1$. The Mach number of the shock was varied by adjusting the preshock pressure, P_1, through the relation

$$P_1 = \frac{(1+\gamma)^2}{4\gamma} \frac{\mathcal{M}^2}{(\mathcal{M}^2 - 1)^2} \tag{4.4}$$

(e.g., Liepmann and Roshko 1957, p. 64).

In order to facilitate application of the boundary conditions at the piston interface, the piston was represented by a small number of particles which were given a constant velocity equal to u_p. The properties of these particles, i.e., density and thermal energy, were selected to satisfy the jump conditions across the shock, as given by the Rankine-Hugoniot relations.

An example involving a strong shock of Mach number $\mathcal{M} = 100$ is shown in Figures 4*a*–4*d* using the SPH artificial viscosity in equations (2.22)–(2.23). Initially, 150 equal-mass particles, each with mass $m_i = 0.01$, were distributed in the range $0 \le x \le 1.5$. The smoothing length was $h = 0.02$ and the time step was $\Delta t = 0.001$. The parameters appearing in the artificial viscosity were $\alpha = 2.5$, $\beta = 2.5$, and $\eta^2 = 0.1$. The system state is shown at a time $t = 0.9$.

The shock, located between $x = 1.05$ and $x = 1.15$ in Figures 4*a*–4*d*, is broadened over approximately $4h$, and postshock oscillations are negligible. The jump conditions across the shock, as measured from the properties of the particles, are satisfied to a high degree of accuracy. The ratios of postshock to preshock density and pressure are predicted to be $\rho_2/\rho_1 = 5.997$ and $P_2/P_1 = 11{,}667$, while the corresponding numerical values are $\rho_2/\rho_1 = 5.95$ and $P_2/P_1 = 11{,}664$. Furthermore, the predicted shock speed is $u_s = 1.20$, while that found numerically, determined from the position of the median particle in the shock front, is $u_s = 1.21$. Evidently, SPH is fully capable of handling even rather strong shocks.

The behavior of the entropic function, $A(s)$, near the piston in Figure 4*d* is partly an artifact of the crude handling of the boundary conditions. The phenomenon of "wall heating" at rigid boundaries is commonly seen in numerical treatments of hydrodynamics and can be suppressed, if necessary, by introducing artificial thermal diffusion (see Monaghan 1988*a*).

Initial conditions identical to those above were also evolved using the bulk viscosity, equations (2.24)–(2.26); the results are shown in Figure 5. The parameters α and β, chosen to give roughly the same resolution as in Figure 4, were $\alpha = 1.0$ and $\beta = 1.0$. In this case the bulk viscosity gives a surprisingly accurate description of the flow, in view of the results obtained for low Mach number. Small-amplitude oscillations can be seen in the velocity field, but the jump conditions are satisfied to the same degree of precision as in Figure 4. For example, the measured jumps in density and pressure are now $\rho_2/\rho_1 = 5.94$ and $P_2/P_1 = 11{,}682$, while the inferred shock speed is $u_s = 1.21$. These results are encouraging, suggesting that the bulk viscosity will be completely adequate in many situations. We stress, however, that this conclusion follows largely from our decision to turn off the viscous interaction between receding particles, as outlined in § II*a*(iii). Simulations without this refinement gave substantially noisier fields behind the shock fronts and, even more disturbing, did not accurately reproduce the jump conditions or shock speed.

The tests shown in Figures 2–5 were repeated using various formulations of SPH and a wide range of parameters. Several different choices for the smoothing kernel, W, were tried, including Gaussians, exponentials, and splines of higher than cubic order. None gave results significantly improving upon those obtained with the cubic spline [eq. (2.8)]. A similar conclusion held for symmetrized forms of $\nabla P/\rho$ other than that in equation (2.20), aside from stability problems discussed earlier, and for other artificial viscosities optimized for SPH such as that proposed by Lattanzio *et al.* (1985, 1986).

Of somewhat greater interest is the dependence of the shock evolution on the artificial viscosity parameters α and β. For equations (2.22)–(2.23) there typically exists optimal values for α and β above which postshock oscillations are negligible. The best choices increase slowly with increasing Mach number, $\mathcal{M}$, from $\alpha \sim \beta \sim 1$ for $\mathcal{M} \sim 1.5$ to $\alpha \sim \beta \sim 2.5$

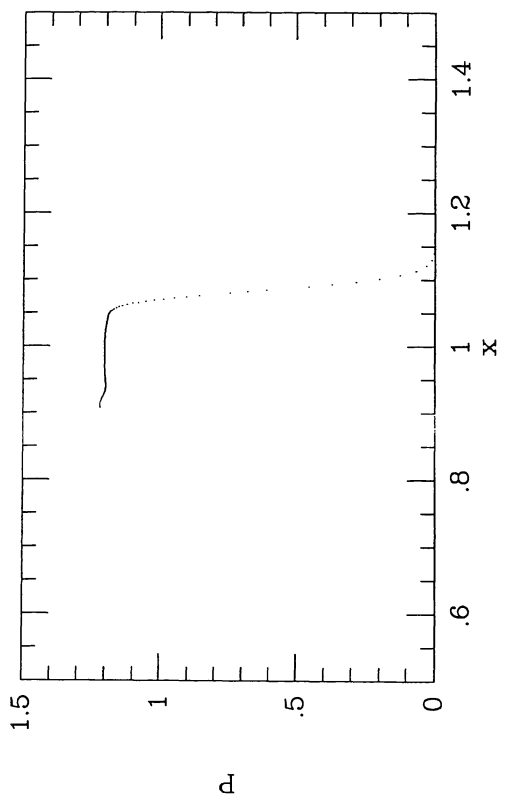


FIG. 4a

FIG. 4b

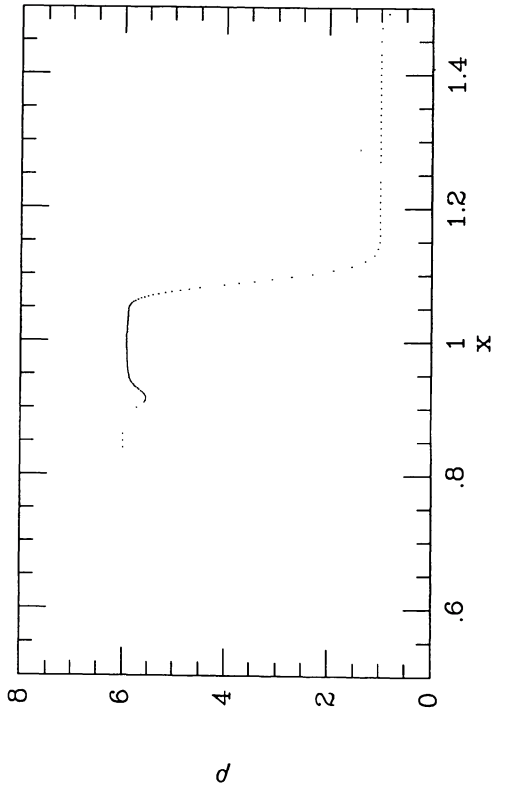


FIG. 4c

FIG. 4d

FIG. 4.—(*a*) Density, (*b*) pressure, (*c*) velocity, and (*d*) entropic profiles in a one-dimensional piston problem, using the standard SPH artificial viscosity defined by eqs. (2.22)–(2.23). The piston is represented by the leftmost 10 equally spaced particles.

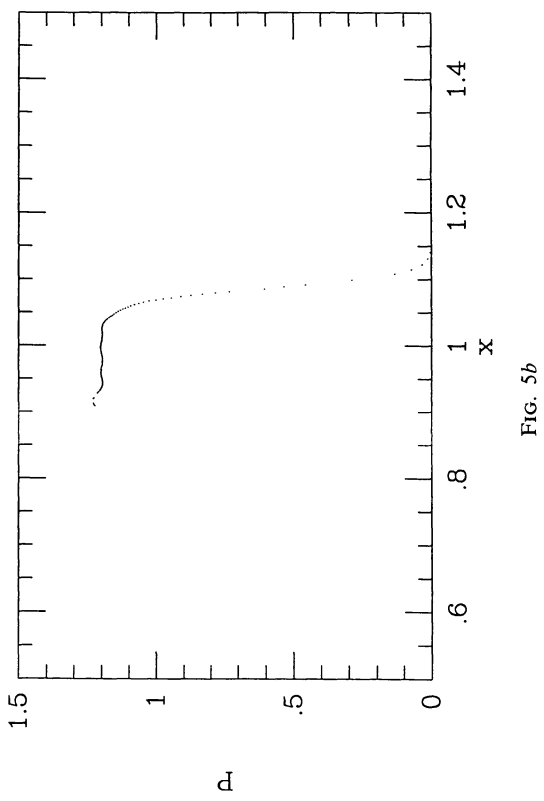


FIG. 5b

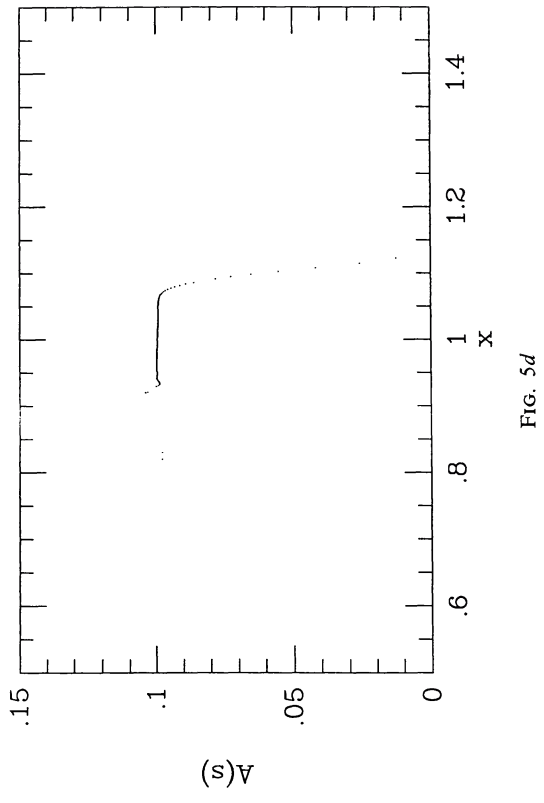


FIG. 5d

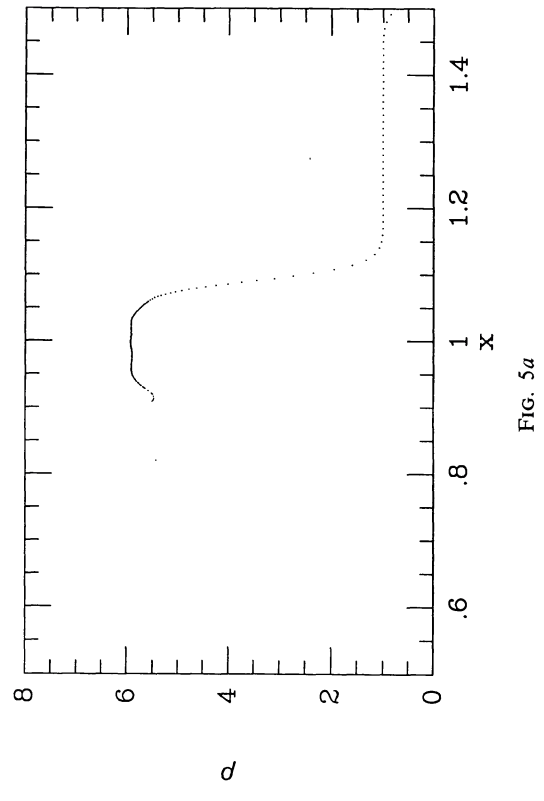


FIG. 5a

FIG. 5c

FIG. 5.—Same as Fig. 4, but using the bulk artificial viscosity, eqs. (2.24)–(2.26).

for $\mathcal{M} \sim 100$. Intermediate to these, $\alpha \sim \beta \sim 1.7$ at $\mathcal{M} \sim 3$, $\alpha \sim \beta \sim 2.1$ at $\mathcal{M} \sim 5$, and $\alpha \sim \beta \sim 2.4$ for $\mathcal{M} \sim 10$. For these optimal values the shock width is $\sim(3\text{–}4)h$, independent of the Mach number, providing further evidence that SPH does not degrade seriously with increasing $\mathcal{M}$. Since the term in the artificial viscosity involving α dominates for low Mach number, while that depending on β is more important at high Mach number, our results suggest that the choice $\alpha \sim 1$ and $\beta \sim 2$ will be a reasonable compromise for all $\mathcal{M}$, in agreement with Monaghan (1988*a*) and Evrard (1988). Similar conclusions apply to Lattanzio *et al.*'s (1985, 1986) artificial viscosity, but with α and β reduced by a factor $2/\gamma$.

Optimal values of α and β for the bulk viscosity from equations (2.24)–(2.26) do not exist, in the sense that postshock oscillations will always be present at some level, unless the resolution is excessively degraded. In order to achieve the same resolution as with equations (2.22)–(2.23), α and β must be reduced by a factor ~ 2 relative to the values given above.

c) *Tests in Three Dimensions*

In addition to the one-dimensional tests discussed above, TREESPH has been applied to a number of three-dimensional problems having known analytic solutions or which have been studied using independent numerical techniques. Here, we summarize results of experiments simulating the adiabatic collapse of initially isothermal gas spheres. The initial conditions were chosen to match those of Evrard (1988) to facilitate comparison with his P3MSPH code and with finite-difference methods (e.g., Thomas 1987). Other three-dimensional tests, such as the normal mode oscillations of gas polytropes, will be presented elsewhere.

Following Evrard (1988), the initial state consists of gas spheres of radius R and total mass M_T, with density profile

$$\rho = \frac{M_T}{2\pi R^2}\frac{1}{r}. \tag{4.5}$$

The gas is initially isothermal with internal energy per unit mass $u = 0.05GM_T/R$. The ratio of specific heats is $\gamma = 5/3$.

An example of the resulting evolution is shown in Figures 6–12, for a simulation with $N = 4096$. The particles were initially distributed randomly using an acceptance-rejection procedure. The SPH artificial viscosity defined by equations (2.21)–(2.23) was selected, with $\eta^2 = 0.01$, $\alpha = 1$, and $\beta = 2$. The gravitational potential was computed with tolerance parameter $\theta = 0.8$ (eq. [2.34]), including quadrupole terms in the multipole expansions. The gravitational softening parameter was $\epsilon = 0.1$; approximately equal to the initial mean interparticle separation evaluated at the half-mass radius of the sphere. Individual particle smoothing lengths and time steps were used. The smoothing length for each particle was chosen as outlined in § II*a*(ii), with $\mathcal{N}_s = 32$. No constraints were imposed on the variation in h_i. Time steps were selected on the basis of the modified Courant condition, equations (2.37)–(2.38), with $\mathcal{C} = 0.3$. The largest system time step, Δt_s, was $\Delta t_s = 0.022$.

Total energies are shown in Figure 6, in a system of units such that $G = M_T = R = 1$. The gravitational potential energy, Ω, is initially equal to -0.669, close to the value $-2/3$ expected for a $1/r$ density profile. During the maximum compression between $t \approx 0.8$ and $t \approx 1.2$, most of the kinetic energy is converted into heat, and then the system expands slowly. At late times, $t \gtrsim 3$, the system is nearly in equilibrium with total thermal energy $U \approx -\Omega/2$.

The detailed system profiles are shown in Figures 7–12, at times matching those given by Evrard (1988). Note that our normalization of the time, t, differs from Evrard's by a factor $(\pi^2/8)^{1/2} \approx 1.1$. The outward propagating shock is most clearly visible at $t = 0.88$, when it is located between $0.18 \lesssim r/R \lesssim 0.25$. The shock is broadened by the artificial viscosity over roughly 2–3 smoothing lengths, and postshock oscillations are negligible. At late times a core-halo structure develops, with nearly isothermal inner regions and the outer regions cooling adiabatically.

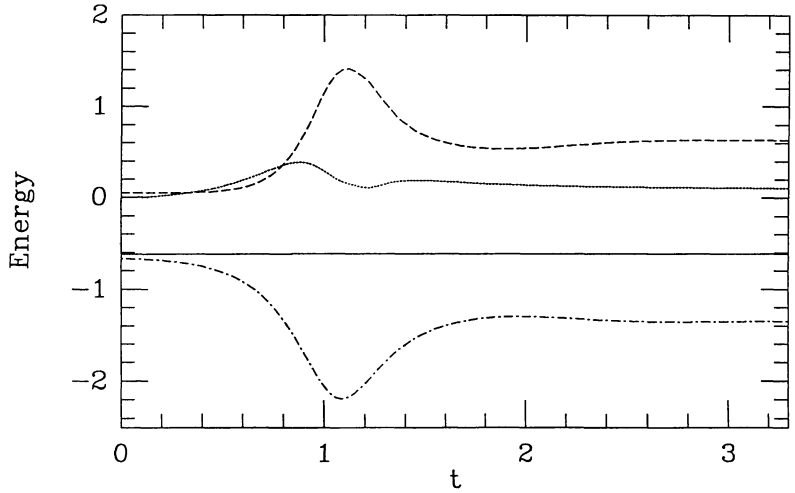


FIG. 6.—Energies in the adiabatic collapse of an initially isothermal gas sphere. Solid line is the total system energy, dashed line is the thermal energy, dotted line is the kinetic energy, and dot-dashed line is the potential energy.

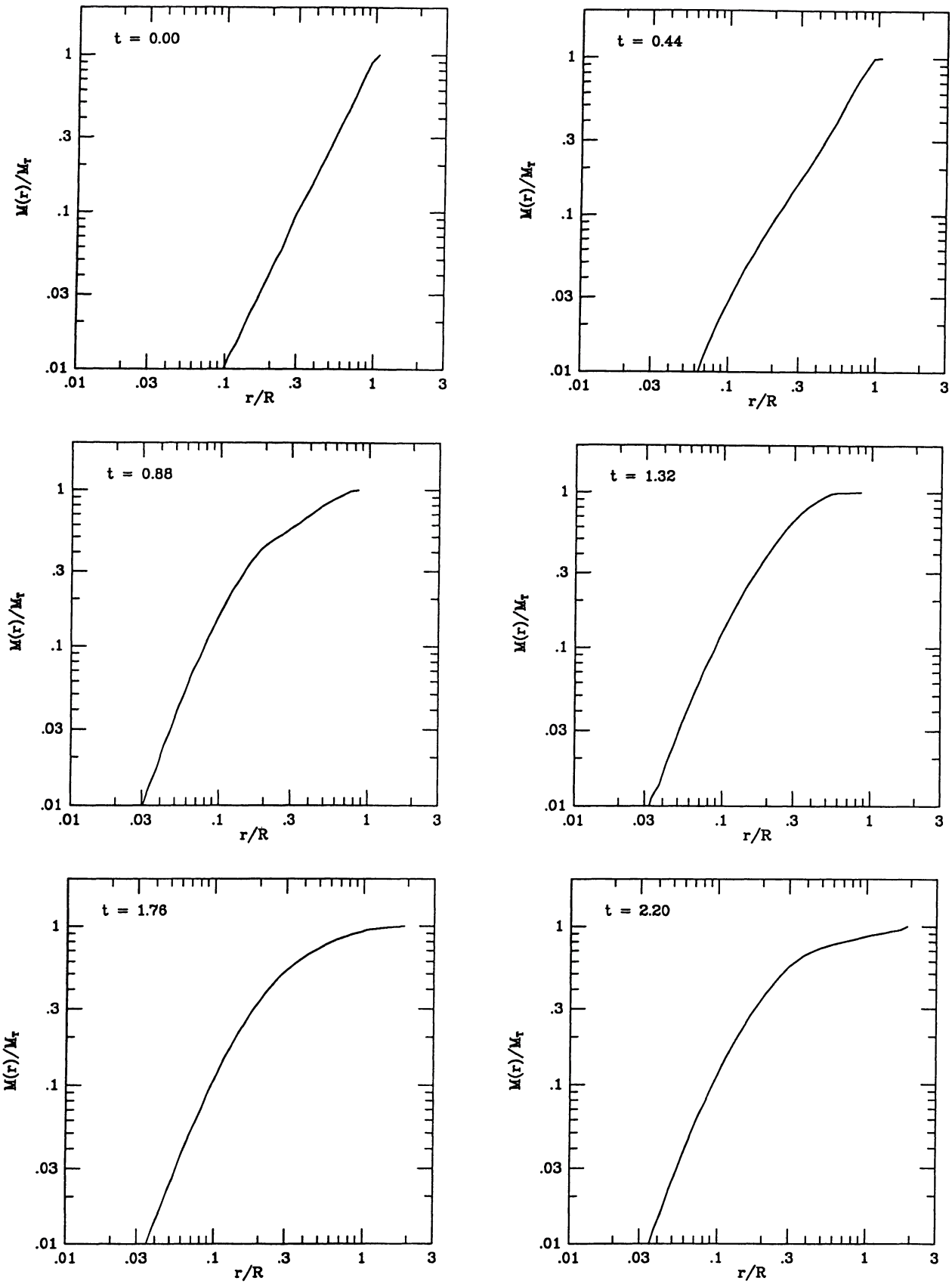


FIG. 7.—Cumulative mass function during the adiabatic collapse of a gas sphere, initially having an isothermal internal energy distribution and a $1/r$ density profile. The abscissa measures the spherical radius, r, in units of the initial system radius, R. The ordinate denotes the cumulative mass distribution, $M(r)$, in units of the total system mass, M_T. Dimensionless time, i.e., assuming $G = M_T = R = 1$, is shown in the upper left corner of each frame.

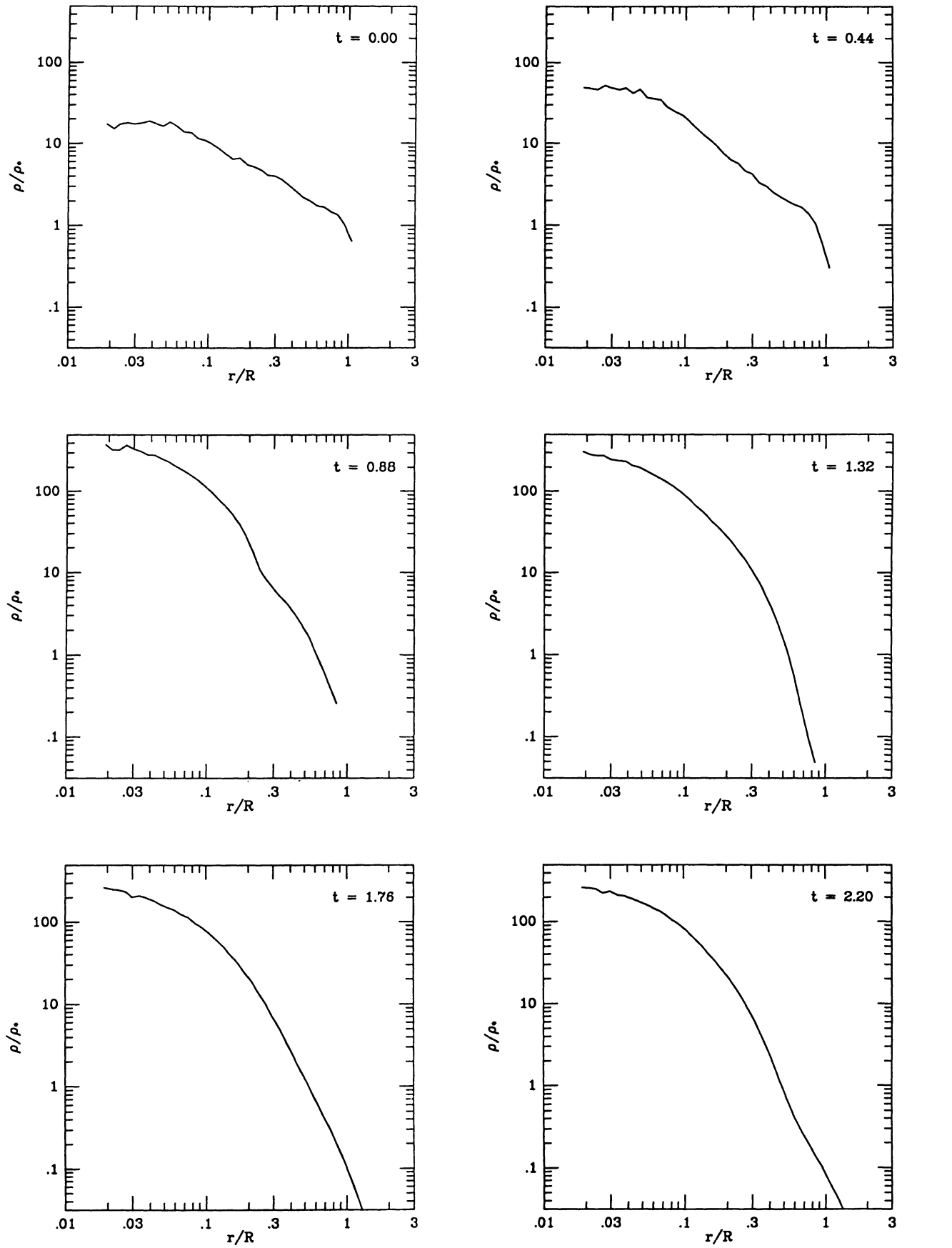


FIG. 8.—Density profile during the adiabatic collapse of the gas sphere described in Fig. 7 and the text. Following Evrard (1988), density is measured in units of $\rho^* = 3M_T/4\pi R^2$. Dimensionless time, as normalized in Fig. 7, is shown in the upper right corner of each frame.

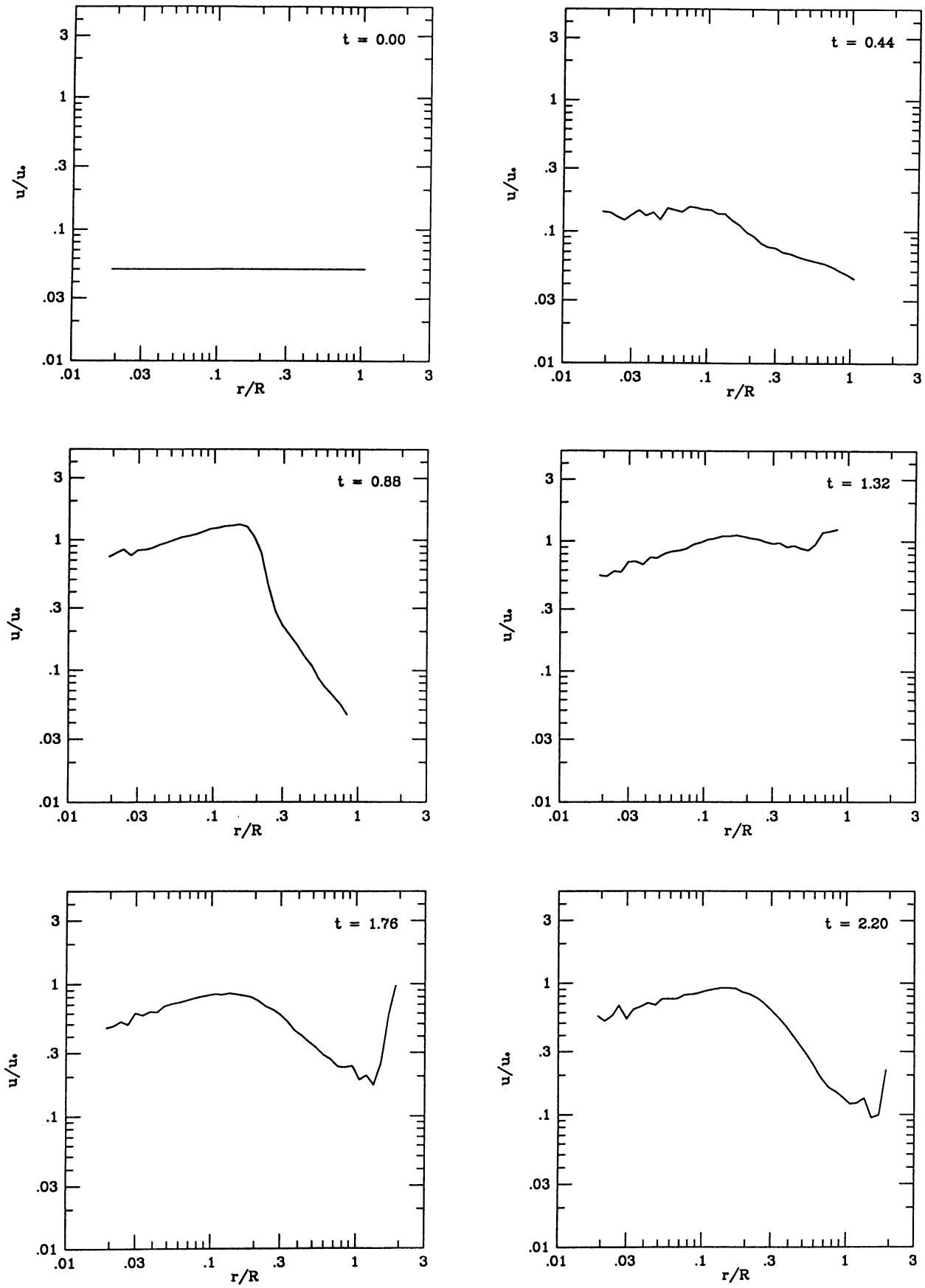


FIG. 9.—Thermal energy distribution, normalized to $u^* = GM_T/R$, during the adiabatic collapse of the gas sphere described in Fig. 7 and the text. Dimensionless time, as defined in Fig. 7, is shown in the upper right corner of each frame.

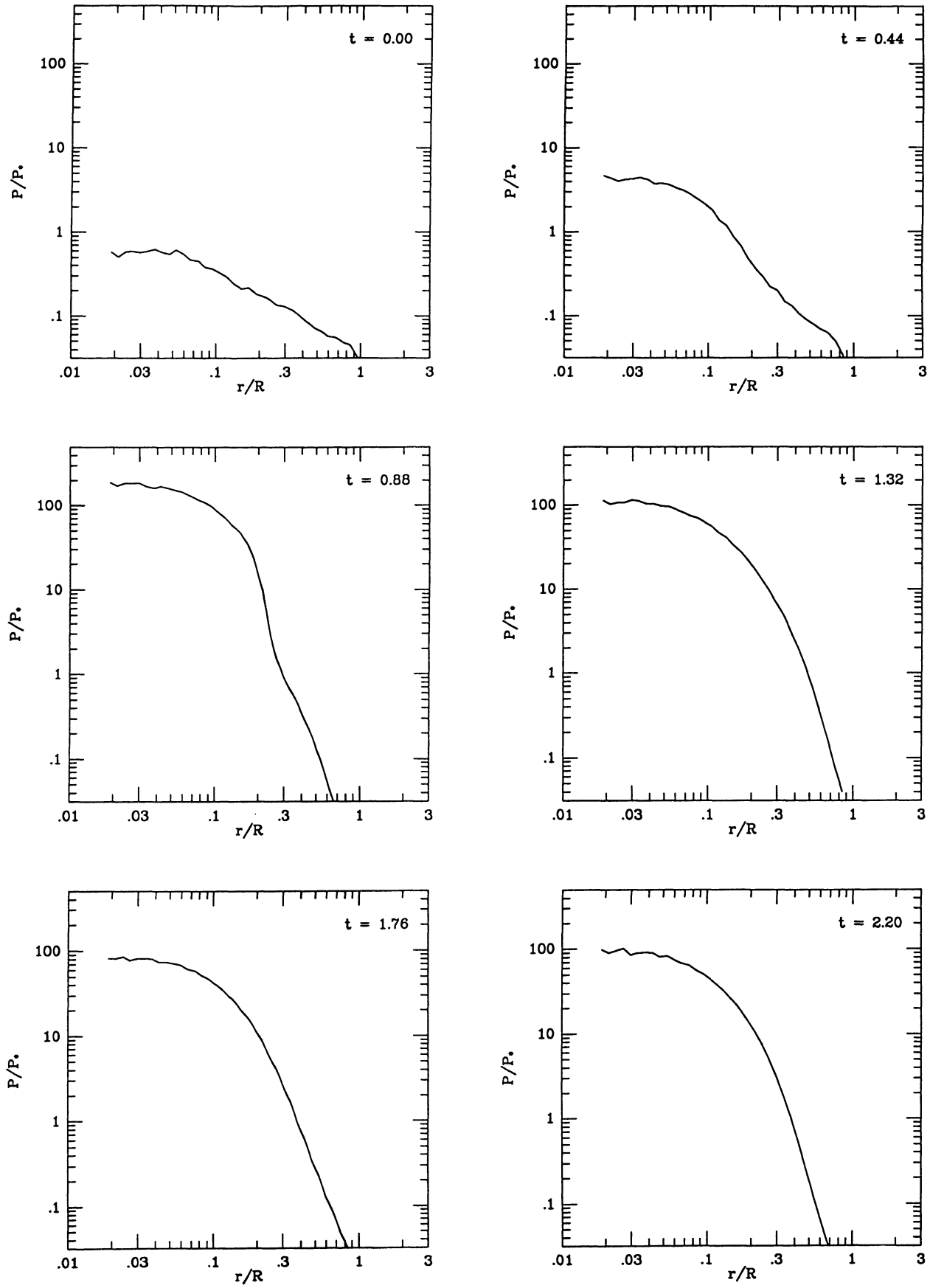


FIG. 10.—Pressure, normalized to $P^* = \rho^* u^*$, during the adiabatic collapse of the gas sphere described in Fig. 7 and the text. Dimensionless time, as defined in Fig. 7, is shown in the upper right corner of each frmae.

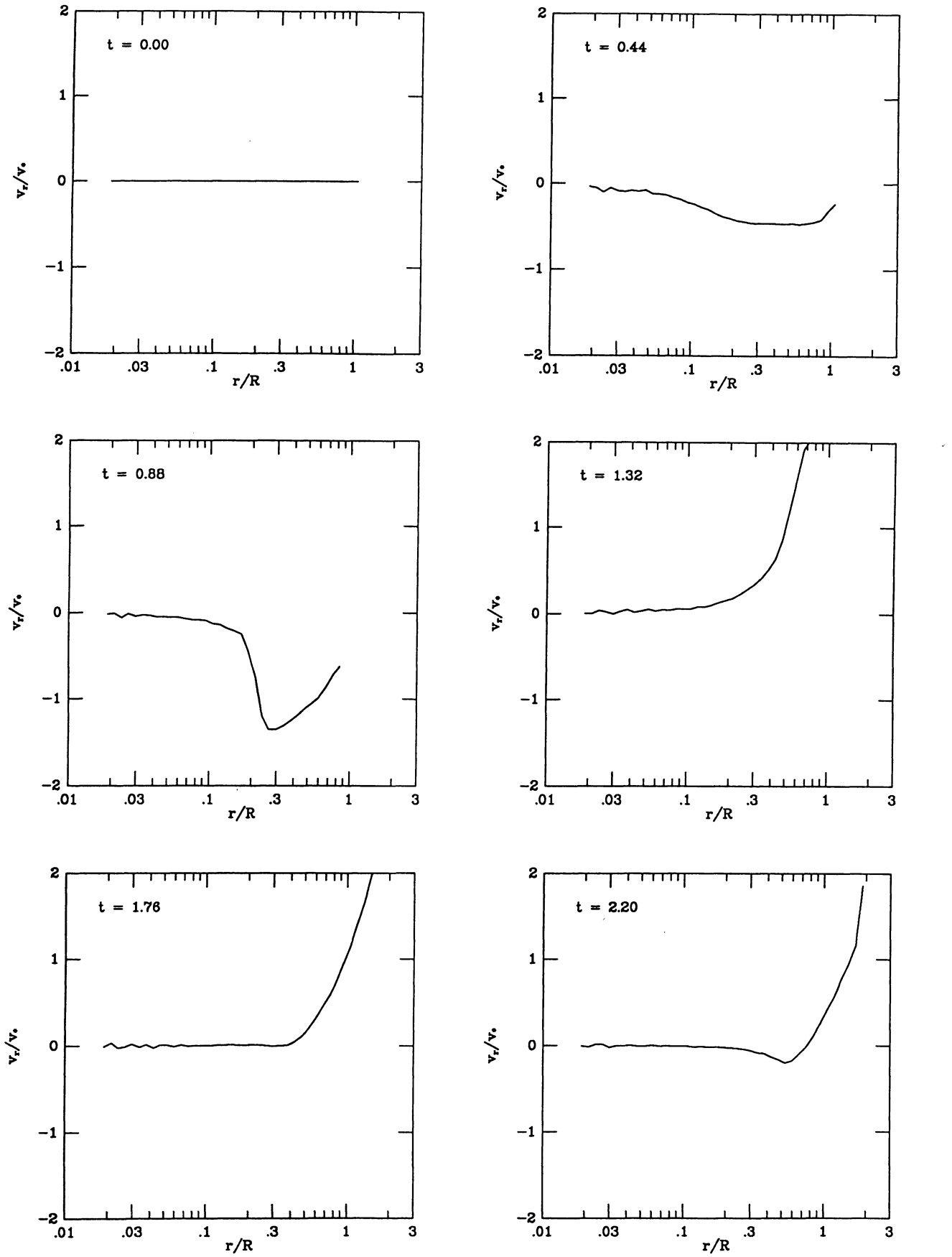


FIG. 11.—Radial velocity profiles, normalized to $v^* = (GM_T/R)^{1/2}$, during the adiabatic collapse of the gas sphere described in Fig. 7 and the text. Dimensionless time, as defined in Fig. 7, is shown in the upper left corner of each frame.

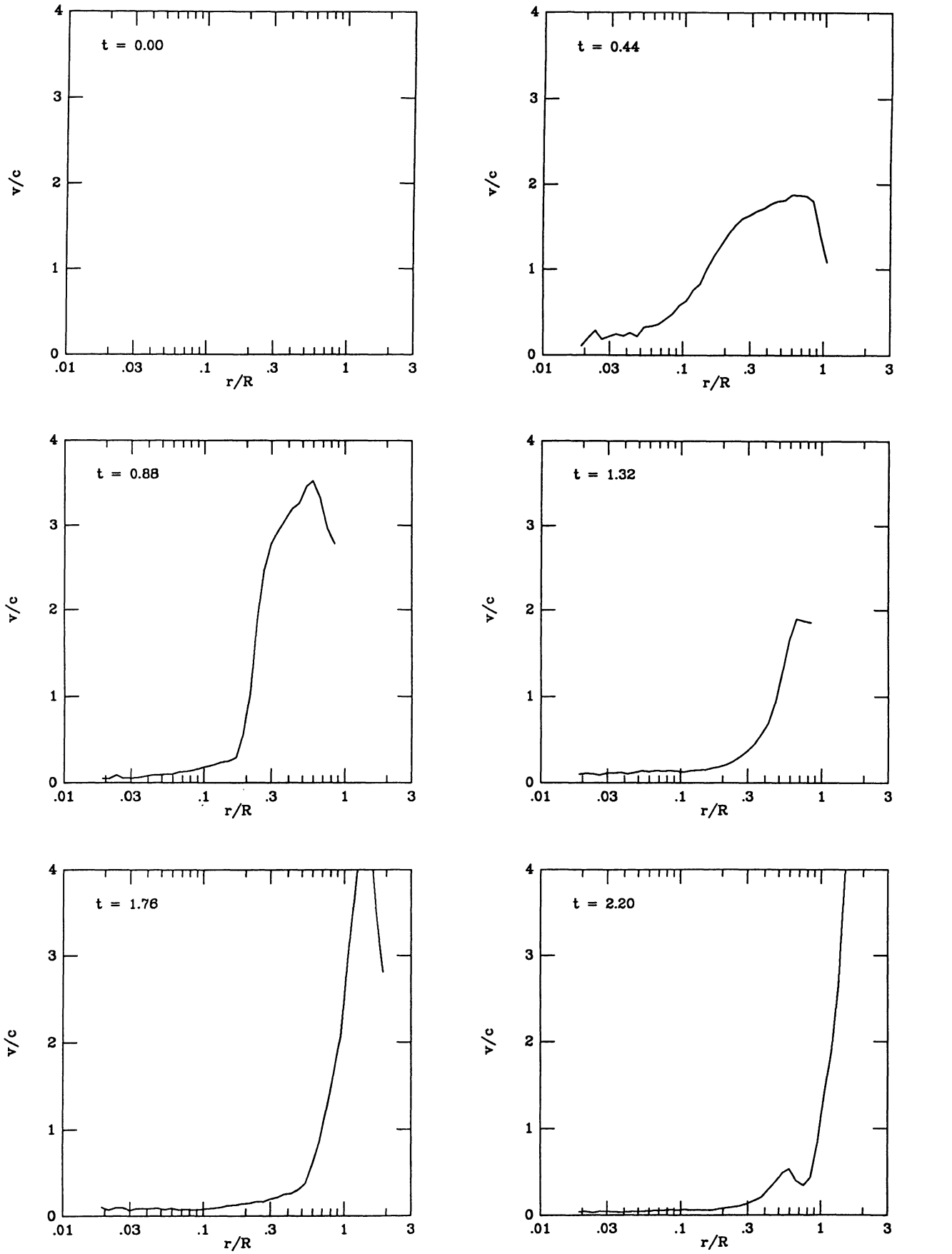


FIG. 12.—Mach number, v/c, during the adiabatic collapse of the gas sphere described in Fig. 7 and the text. Dimensionless time, as defined in Fig. 7, is shown in the upper right corner of each frame.

The results are in excellent agreement with those presented by Evrard (1988), particularly those obtained with a one-dimensional finite-difference code (Thomas 1987). (Differences between our results and Evrard's SPH calculations can be attributed to discreteness effects.) The location and strength of the shock are accurately reproduced by our simulation. For example, at the time equivalent to our $t = 0.88$ frame, the shock in Evrard's one-dimensional calculation is located at $r/R \approx 0.2$. The structure at late times also agrees well. For example, the thermal energy distribution shown in Figure 9 has a slow rise from $r/R = 0$ to $r/R \approx 0.1$ in both cases. Small differences at intermediate times result from the lower resolution in our simulation and the fact that the artificial viscosities are not identical.

The agreement we obtain in comparison with Evrard's finite-difference calculation is especially encouraging since a large range in smoothing lengths and time steps was present throughout the course of the simulation shown in Figures 6–12. For example, the dynamic range in time-scales was $\approx$32:1 at the point of maximum compression. We tentatively conclude that the dynamic nature of TREESPH does not adversely affect the evolution of at least some simple adiabatic flows. Obviously, however, this statement cannot be generalized and the numerical effects must be examined on a case-by-case basis.

d) *Timing Analysis*

The gain in efficiency from vectorization of the tree traversals and tree construction was first investigated by evolving pressureless Plummer models, using just the gravitational sections of TREESPH, and comparing these results to those obtained with Hernquist's (1987) original FORTRAN TREECODE. A substantial improvement is realized, amounting to factors ~4–5 in the tree descent and tree initialization subroutines. Overall, CPU costs are reduced by factors $\approx$2.5–3 or $\approx$2–2.5 if the potential of cell is evaluated to monopole or quadrupole order, respectively, with a weak dependence on N and θ (Hernquist 1989). The gravitational computation is now essentially fully vectorized, in the sense that $\gtrsim$95% of the operations consist of vector arithmetic.

The hydrodynamic sections of TREESPH are vectorized to a similar extent. The overall gain in speed on a Cray compared with other smaller machines is impressive. Typically, TREESPH is a factor $\approx$350–500 times faster on a Cray XMP-48, in single-processor mode, than on a Sun 3/50, depending primarily on θ. For example, the performance ratio Cray/Sun is $\approx$415 if $\theta = 0.8$ and $\approx$440 if $\theta = 0.4$, for an $n = 3/2$ fluid polytrope with $N = 2048$ and including quadrupole terms in the gravitational force computation. This is in good agreement with timing tests using linear algebra packages (e.g., Dongarra 1988). The Sun 3/50 has a floating point rating of roughly 90 kflops, implying that the Cray XMP-48 is being driven at $\approx$35–50 Mflops, which is still considerably less than the theoretical peak performance of 210 Mflops per processor. This is probably the result of inefficient memory addressing, lack of chaining, operations involving vectors of nonoptimal length, and extensive use of gather-scatter operations.

If $\theta \gtrsim 0.2$–0.3, then the CPU time per step in TREESPH scales as $c_1 N \log N + c_2 N$, where c_1 and c_2 are constants. For a fixed value of $\mathcal{N}_{\rm avg}$, all SPH functions are $\sim O(N)$, as are the nearest neighbor searches and most aspects of tree construction and maintenance. The only routines scaling as $\sim O(N \log N)$ are those in the gravity computation, which establish interaction lists and perform force summation, and that which initializes the pointers in the tree structure.

Unfortunately, the CPU time per step required by the gravitational sections of TREESPH relative to the hydrodynamic contribution is sensitive to the density profile, to the dynamic range in time scales present, and to the relative mixture of particles representing gas and collisionless matter. Consequently, any estimate of the efficiency of TREESPH will be highly model-dependent. In the limit that all particles represent gas and have the same time step, empirical tests demonstrate that the CPU time per step is mostly dominated by the gravitational computation. An example of the CPU usage in various sections of TREESPH is given in Table 1 for an $n = 3/2$ polytrope, with $N = 16{,}384$, $\mathcal{N}_s = 32$ (see eq. [2.17]), and $\theta = 0.8$, including quadrupole terms in the multipole expansions. Here, a single fixed time step was used for all particles. For these parameters, the ratio of CPU times in the gravitational and SPH sections of the code is $\approx$3:2.

As is evident from Table 1, the time spent constructing the tree is negligible, despite the fact that it is rebuilt at each step. Furthermore, the relative cost of tree maintenance is lower than that found by Hernquist (1987), owing to vectorization. This property of the Barnes-Hut algorithm is especially advantageous if multiple time scales are present and only a small fraction of the particles are advanced on the shortest time step. The CPU time devoted to the gravitational computation is roughly divided evenly between the subroutine which establishes the interaction lists and that which performs force summation (Hernquist 1989).

An example of the CPU distribution in a problem involving a relatively large range in time scales, using individual particle time steps, is shown in Table 2, for the adiabatic collapse of a nonrotating gas sphere with $N = 4096$ described in § IV*c*. The cost is now more evenly divided between the gravitational and hydrodynamic computations, and tree construction is relatively more expensive than in Table 1. In this context, we note that in its current form TREESPH is not completely optimized since some aspects of the calculation, such as the integration of the thermal energy equation and nearest neighbor searching, are performed for all particles on the shortest time scale. Furthermore, the tree data structure is reconstructed *from scratch* on this same time interval. It would presumably be more efficient to reform the tree by locally

TABLE 1

CPU PERCENTAGES FOR EQUILIBRIUM POLYTROPE

Section of Code	Percentage of CPU Time
Gravitational computation ...	56.20
SPH computations	21.65
Nearest neighbor searching ...	18.60
Tree construction	3.28
Miscellaneous	0.27

TABLE 2

CPU PERCENTAGES FOR ADIABATIC COLLAPSE

Section of Code	Percentage of CPU Time
Gravitational computation ...	43.28
SPH computations	27.63
Nearest neighbor searching ...	24.43
Tree construction	4.50
Miscellaneous	0.16

modifying the copy from the previous step. Obviously, room for improvement still exists and a number of these options are currently under investigation.

Considerations of the accuracy of the gravitational force computation suggest that a reasonable choice of θ will be $\theta \sim 0.6$–1.0. A simulation such as that described by Table 1 with $N = 16{,}384$, a single time step, and θ in this range will require $\approx$10–20 CPU seconds per step on a Cray XMP, if quadrupole terms are included in the multipole expansions, implying a total cost of $\approx$3–6 CPU hours for 1000 steps. Calculations involving a large range in time scales, as in radiatively cooling flows, will be more costly.

V. SUMMARY

In spite of the obvious importance of gas dynamical processes in astrophysics, a detailed understanding of these phenomena remains elusive, owing to their complexity. Hydrodynamic flows are frequently nonlinear and three-dimensional in nature, possessing no obvious symmetries. Large dynamic ranges in both spatial and mass scales are often present, resulting from the interplay of self-gravity with short-range hydrodynamic forces. Finally, microscopic effects, such as radiative cooling, complicate matters further by introducing large ranges in temporal scales.

A complete physical description of the dynamics in such situations can only be obtained numerically. Smoothed particle hydrodynamics (SPH) was introduced as an alternative to grid-based schemes, which have historically not been sufficiently flexible to model three-dimensional fluids with self-gravity. Another advantage of SPH is the ease with which it can handle multiple time scales, through the use of individual particle time steps. In the past, owing to inefficiencies in the computation of the gravitational potential, SPH has been limited to problems where a large dynamic range in spatial resolution was not required or to others in which self-gravity could be neglected. However, a number of new potential solvers have now been developed which offer many advantages for SPH. Among these, hierarchical tree methods appear to be most promising, since they are gridless and adapt naturally to the SPH formalism.

In this paper, we have described a version of SPH, TREESPH, in which gravitational forces are computed using the Barnes-Hut tree algorithm. Our approach can be contrasted with other efficient SPH–based codes, which have recently been proposed. In P3MSPH (Evrard 1988), hydrodynamic effects are computed in the usual manner, but self-gravity is determined using the P^3M technique (e.g., Hockney and Eastwood 1981), naturally permitting the use of a large particle number, N, and a large dynamic range in spatial resolution. P3MSPH will certainly be a useful tool for studying the growth of large-scale structure in the early universe, but is likely to be rather limited in its present form. Historically, P^3M has been used only in conjunction with periodic boundary conditions, a restriction which is not shared by the hierarchical tree method (Bouchet, Hernquist, and Suto 1989). P3MSPH will also not be useful for evolving flows in which the cooling time is much shorter than the dynamical time. Most important, any implementation of SPH will be severely constrained by the Courant condition, unless multiple time scales are included.

TREESPH is most closely related to Benz's algorithm (Benz 1988; Benz *et al.* 1988), in which gravitational forces are computed with a hierarchical *binary* tree (Press 1986). We expect that these two techniques will offer similar advantages, and it will be interesting to see how they compare. For now, the major advantages of TREESPH lie in the ease with which the Barnes-Hut potential solver can be fully vectorized and in the implementation of individual particle time steps. Similar refinements have yet to be included in the binary tree algorithm. Note, also, that numerical tests performed by Makino (1989) do not support the contention that the binary scheme is significantly more efficient at fixed accuracy in the force calculation than Eulerian tree algorithms. In fact, if a large dynamic range in time scales is present, Eulerian tree codes should be faster than their Lagrangian counterparts owing to less costly tree construction.

Although we have emphasized astrophysical applications, TREESPH will obviously be useful for studying a much broader class of problems and is best suited to those involving both long- and short-range forces. An obvious, but completely unrelated example, is that of the computer modeling of nuclear dynamics (e.g., Boal 1987). Others include molecular dynamics and screened plasmas (e.g., Greengard 1987).

TREESPH is not the final word on this subject, however, since a number of refinements immediately suggest themselves. The CPU efficiency could, in principle, be improved further still through the use of the fast multipole method (e.g., Greengard and Rokhlin 1987) to compute the gravitational potential. More important, unlike finite-difference codes, SPH cannot resolve arbitrarily large density contrasts. Some progress can be made in this direction by dynamically creating and destroying particles (e.g., Monaghan and Varnas 1988), but a complete solution is likely to require major revisions to the SPH formalism. One possibility might be to switch to logarithmic variables and advance the continuity equation explicitly, which would then imply that the smoothing procedure be altered. Similar complexities would arise if it were necessary to resolve shocks more precisely than is possible with an artificial viscosity treatment (e.g., Woodward and Colella 1984). In some cases, there is apparently no good substitute for the finite-difference method.

Nevertheless, we anticipate that TREESPH will enable us to study many astrophysical phenomena that have not been approachable with more traditional analytic or numerical techniques. The most obvious systems are those which are inherently three dimensional, self-gravitating, which possess a large dynamic range in temporal scales, and which involve both gas and collisionless matter.

We would like to thank Joe Monaghan for advice concerning the implementation of variable smoothing lengths in SPH and for numerous comments on the manuscript, Bernard Chazelle and Robert Sedgewick for suggestions regarding nearest neighbor searching, Josh Barnes and Piet Hut for discussion about the hierarchical tree method, Willy Benz, Gus Evrard and Leon Lucy for discussion about SPH, and Jim Gunn for continued interest and support. In addition, L. H. is grateful to Avery Meiksin for assistance in performing one-dimensional tests of SPH, and for the hospitality of the Astronomy Department at the University of California at Berkeley and Institute of Geophysics and Planetary Physics, Livermore, where this project was initiated. This work was supported in part by grants from the San Diego Supercomputer Center and the Pittsburgh Supercomputing Center and by New Jersey High Technology grant No. 88-240090-2.

APPENDIX

SOFTENED FORM OF GRAVITATIONAL FIELD

The spline-softened form of the gravitational potential, $\Phi = -mf(r)$, and acceleration, $a = -mrg(r)$, using equation (2.8) and the procedure outlined in Gingold and Monaghan (1977), are

$$f(r) = \begin{cases} -\frac{2}{\varepsilon}\left[(1/3)u^2 - (3/20)u^4 + (1/20)u^5\right] + 7/5\varepsilon, & 0 \le u \le 1, \\ -1/15r - \frac{1}{\varepsilon}\left[(4/3)u^2 - u^3 + (3/10)u^4 - (1/30)u^5\right] + 8/5\varepsilon, & 1 \le u \le 2, \\ 1/r, & u \ge 2, \end{cases} \tag{A1}$$

$$g(r) = \begin{cases} 1/\varepsilon^3\left[4/3 - (6/5)u^2 + (1/2)u^3\right], & 0 \le u \le 1, \\ 1/r^3\left[-1/15 + (8/3)u^3 - 3u^4 + (6/5)u^5 - (1/6)u^6\right], & 1 \le u \le 2, \\ 1/r^3, & u \ge 2, \end{cases} \tag{A2}$$

where $u = r/\varepsilon$.

REFERENCES

Ambrosiano, J., Greengard, L., and Rokhlin, V. 1988, *Computer Phys. Comm.*, **48**, 117.
Appel, A. W. 1981, Undergraduate thesis, Princeton University.
_____. 1985, *SIAM J. Sci. Stat. Comput.* **6**, 85.
Balsara, D. S., Norman, M. L., Bicknell, G. V., and Gingold, R. A. 1988, preprint.
Barnes, J. 1986, in *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan (Berlin: Springer), p. 175.
_____. 1989, *J. Comput. Phys.*, in press.
Barnes, J., and Hut, P. 1986, *Nature*, **324**, 446.
Bentley, J. L., and Friedman, J. H. 1979, *Computing Surveys*, **11**, 397.
Bentley, J. L., and Stanat, D. F. 1975, *Information Processing Letters*, **3**, 170.
Benz, W. 1984, *Astr. Ap.*, **139**, 378.
_____. 1986, in *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan (Berlin: Springer-Verlag), p. 117.
_____. 1988, *Computer Phys. Comm.*, **48**, 97.
Benz, W., Bowers, R. L., Cameron, A. G. W., and Press, W. H. 1988, preprint.
Benz, W., Slattery, W. L., and Cameron, A. G. W. 1986, *Icarus*, **66**, 515.
_____. 1987, *Icarus*, **71**, 30.
Boal, D. H. 1987, *Ann. Rev. Nucl. Particle Sci.*, **37**, 1.
Bouchet, F. R., and Hernquist, L. 1988, *Ap. J. Suppl.* **68**, 521.
Bouchet, F. R., Hernquist, L., and Suto, Y. 1989, in preparation.
Carrier, J., Greengard, L., and Rokhlin, V. 1988, *SIAM J. Sci. Stat. Comput.*, **9**, 669.
Casulli, V., and Greenspan, D. 1984, *Internat. J. Numerical Methods Fluids*, **4**, 1001.
Dongarra, J. J. 1988, Tech. Memo. No. 23, Argonne National Laboratory.
Durisen, R. H., Gingold, R. A., Tohline, J. E., and Boss, A. P. 1986, *Ap. J.*, **305**, 281.
Ewell, M. W. 1988, Ph.D. thesis, Princeton University.
Evrard, A. E. 1988, *M.N.R.A.S.*, **235**, 911.
Field, G. B. 1965, *Ap. J.*, **142**, 531.
Finkel, R. A., and Bentley, J. L. 1974, *Acta Informatica*, **4**, 1.
Gel'fand, I. M., and Shilov, G. E. 1964, *Generalized Functions* (New York: Academic).
Gingold, R. A., and Monaghan, J. J. 1977, *M.N.R.A.S.*, **181**, 375.
_____. 1980, *M.N.R.A.S.*, **191**, 897.
_____. 1982, *J. Comput. Phys.*, **46**, 429.
Greengard, L. 1987, Ph.D. thesis, Yale University.
_____. 1988, *The Rapid Evaluation of Potential Fields in Particle Systems* (Cambridge: MIT Press).
Greengard, L., and Rokhlin, V. 1987, *J. Comput. Phys.*, **73**, 325.
Hernquist, L. 1987, *Ap. J. Suppl.*, **64**, 715.
_____. 1988, *Computer Phys. Comm.*, **48**, 107.
_____. 1989, *J. Comput. Phys.*, in press.
Hockney, R. W., and Eastwood, J. W. 1981, *Computer Simulation Using Particles* (New York: McGraw-Hill).
Jernigan, J. G. 1985, in *IAU Symposium 127, Proceedings: Dynamics of Star Clusters*, ed. J. Goodman and P. Hut (Dordrecht: Reidel).
Lattanzio, J. C., Monaghan, J. J., Pongracic, H., and Schwarz, M. P. 1985, *M.N.R.A.S.*, **215**, 125.
_____. 1986, *SIAM J. Sci. Stat. Comput.*, **7**, 591.
Liepmann, H. W., and Roshko, A. 1957, *Elements of Gasdynamics* (New York: McGraw-Hill).
Loewenstein, M., and Mathews, W. G. 1986, *J. Comput. Phys.*, **62**, 414.
Lucy, L. 1977, *A.J.*, **82**, 1013.
Makino, J. 1989*a*, *J. Comput. Phys.*, in press.
_____. 1989*b*, preprint.
Monaghan, J. J. 1982, *SIAM J. Sci. Stat. Comput.*, **3**, 422.
_____. 1983, *Proc. Ast. Soc. Australia*, **5**, 182.
_____. 1985, *Computer Phys. Rept.*, **3**, 71.
_____. 1987, private communication.
_____. 1988*a*, preprint.
_____. 1988*b*, *Computer Phys. Comm.*, **48**, 89.
Monaghan, J. J., and Gingold, R. A. 1983, *J. Comput. Phys.*, **52**, 374.
Monaghan, J. J., and Lattanzio, J. C. 1985, *Astr. Ap.*, **149**, 135.
Monaghan, J. J., and Pongracic, H. 1985, *Applied Numerical Math.*, **1**, 187.
Monaghan, J. J., and Varnas, S. R. 1988, *M.N.R.A.S.*, **231**, 515.
Nagasawa, M., and Miyama, S. M. 1987, *Progr. Theor. Phys.*, **78**, 1250.

Nagasawa, M., Nakamura, T., and Miyama, S. M. 1988, *Pub. Astr. Soc. Japan*, in press.
Nolthenius, R. A., and Katz, J. I. 1982, *Ap. J.*, **263**, 377.
Patnaik, G., Guirguis, R. H., Boris, J. P., and Oran, E. S. 1987, *J. Comput. Phys.*, **71**, 1.
Phillips, G. J. 1986*a*, *M.N.R.A.S.*, **221**, 571.
_____. 1986*b*, *M.N.R.A.S.*, **222**, 111.
Phillips, G. J., and Monaghan, J. J. 1985, *M.N.R.A.S.*, **216**, 883.
Porter, D. 1985, Ph.D. thesis, University of California, Berkeley.
Porter, D., and Jernigan, J. G. 1987, preprint.
Press, W. H. 1986, in *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan (Berlin: Springer), p. 184.
Richtmyer, R. D., and Morton, K. W. 1967, *Difference Methods for Initial-Value Problems* (New York: Wiley).
Sanders, R. H. 1977, *Ap. J.*, **217**, 916.
Schüssler, M., and Schmitt, D. 1981, *Astr. Ap.*, **97**, 373.
Sedgewick, R. 1983, *Algorithms* (Reading: Addison-Wesley).
Sod, G. A. 1978, *J. Comput. Phys.*, **27**, 1.
Spitzer, L. 1978, *Physical Processes in the Interstellar Medium* (New York: Wiley).
Thomas, P. 1987, Ph.D. thesis, Cambridge University.
Wood, D. 1981, *M.N.R.A.S.*, **194**, 201.
_____. 1982, *M.N.R.A.S.*, **199**, 331.
Woodward, P., and Colella, P. 1984, *J. Comput. Phys.*, **54**, 115.

LARS HERNQUIST: The Institute for Advanced Study, Princeton, NJ 08540 [hernquist@iassns.bitnet]

NEAL KATZ: Princeton University Observatory, Princeton, NJ 08540