JOURNAL OF COMPUTATIONAL PHYSICS 121, 357–372 (1995)

von Neumann Stability Analysis of Smoothed Particle Hydrodynamics—Suggestions for Optimal Algorithms

DINSHAW S. BALSARA*

Physics and Astronomy Department, Johns Hopkins University, Baltimore, Maryland 21218

Received November 4, 1992; revised April 6, 1995

We present a von Neumann stability analysis of the equations of smoothed particle hydrodynamics (SPH) along with a critical discussion of various parts of the algorithm. The stability analysis is done without any major restrictions and, hence, models the full Euler equations in one dimension. This then allows us to deduce optimal ranges for parameters that need to be used in SPH. Thus we show that for the commonly used M_5 spline the ratio of smoothing length to interparticle distance should range between 1.0 to 1.4. We also show that the linear artificial viscosity coefficient and the coefficient of spatial filtering have to be bounded. The results of this von Neumann stability analysis provide us with several suggestions for future algorithm improvement. Because the SPH method is so unique we provide, wherever possible, comparisons with more familiar and well-used high resolution finite difference methods. © 1995 Academic Press, Inc.

I. INTRODUCTION

The particle method known as smoothed particle hydrodynamics (SPH from here on) was first developed in [4, 7]. It has been used for a variety of astrophysical applications (although almost never for any applications outside of astrophysics). However, the mathematical underpinnings of the method have not yet been thoroughly explored. As a result, a large variety of formulations exist and a large number of parameter settings in any practical SPH implementation are based on the scientists' intuition and subjective feeling about various aspects of the algorithm. Needless to say, that is not a very satisfactory situation, particularly when the intuitions of different people can be vastly different. As a result, a large disparity exists on issues such as which interpolation kernel to use, what width of interpolation support is necessary before the fluid's continuum behavior is arrived at, how much numerical viscosity is necessary, how much spatial averaging is to be used, and which timestepping schemes are stable. The introduction of some mathematical rigor in resolving such issues could only improve the logical trusswork on which the scheme is founded. Moreover, it would serve as a guide in situations where more physics is added to the equations or when the equations are extended to larger systems of hyperbolic equations.

For finite difference methods such a logical framework has been built up over several decades by many workers. It is well known that the von Neumann stability analysis (see, for example, [14]) is one of the cornerstones for such a logical framework. Consistency and conservation are the other two. Taken together, they ensure the convergence of the discrete system to some weak solution of the system of hyperbolic equations. One still needs entropy enforcement in the sense of [13] to ensure that the convergence is to the physically correct entropy-satisfying weak solution. In its simplest form the von Neumann stability analysis requires analyzing the linear stability of sinusoidal modes in the interior part of the computational domain. For most equations and for most systems of equations, this interior mode analysis is perfectly adequate and leaves one with only the additional onus of constructing accurate and stable boundary conditions. In this paper, we present such a von Neumann stability analysis for SPH. We find that a detailed von Neumann stability analysis of the SPH equations allows one to resolve many of the issues that have bearing on the quality of an SPH simulation. These issues have been treated only subjectively so far. In [12] an effort was made to understand the problems associated with particle penetration in SPH. However, in that work the author was interested in the dissipation and dispersion of certain averaging terms he had added to the position update equations. In that work he restricted his analysis to the long wavelength limit. We find that much insight is lost when one makes approximations consistent with such limits. An analysis done in such limits cannot even begin to resolve the above issues because it does not adequately represent the entropy wave and the forward and backward propagating sound waves. In this paper we carry out the full analysis without any of the above stated restrictions and thoroughly explore the implications of such an analysis.

In Section II we give the mathematical background for the method. In Section III we give the results of the analysis and in Section IV we give the conclusions and a critical discussion of how SPH fares vis à vis current high resolution finite difference methods. A large appendix provides various mathematical

* Currently at National Center for Supercomputing Applications, Univ. of Illinois Urbana-Champaign

0021-9991/95 $12.00
Copyright © 1995 by Academic Press, Inc.
All rights of reproduction in any form reserved.

details which the reader can refer to according to need and interest.

II. LINEAR MODE ANALYSIS

II. a. *SPH Time Evolution Equations*

We start our analysis by considering the basic SPH equations in one dimension. von Neumann stability analyses for multidimensions have not yet been attempted. However, insights into the multidimensional behavior can be obtained by thinking of the multidimensional kernel as a tensor product of one-dimensional kernels in each of the principal directions. In theory a good SPH interpolation scheme that conserves global mass, linear momentum, and total energy can be obtained by using such a tensor product; however, in practical implementations one prefers the use of spherically symmetric kernels. The spherical symmetry guarantees angular momentum conservation. Yet thinking of the multidimensional kernel as tensor products of one-dimensional kernels yields valuable insight into wave propagation, the choice of kernels, and the choice of smoothing lengths for those kernels.

A clear derivation of the equations of SPH has been given in [11]. We assume that the particles are set down along the x direction at uniform intervals Δx from each other. For the purpose of simplicity, the particles all have equal mass and a fixed smoothing length h. The particles are labeled by integer subscripts "a" and "b" which range from minus infinity to plus infinity. The primitive variables assigned to each particle are those of mass, velocity, and specific thermal energy. The density at each (moving) particle position is then defined to be a suitable tapered average of the masses of neighboring particles within a certain radius. Thus the density at a point can be denoted by

$$\rho_a = \sum_{b=-\infty}^{+\infty} m_b W(x_b - x_a). \tag{1}$$

Here m_b is the mass of particle "b" and the function $W(x_b - x_a)$ supplies the tapered average and is known as the smoothing (or interpolation) kernel. The equations for momentum and energy update can also be derived by local tapered averages of the nonconservative forms of the Euler equations. The derivation is nontrivial and the interested reader is referred to [11]. Unlike modern day higher order total variation diminishing (henceforth TVD) schemes, SPH does not rely on Riemann solvers and local conservation to capture shocks. For a definition of the TVD property see [5]. In fact, in SPH the time update is not based on the construction of higher order fluxes and so SPH has no local conservation form. Instead, SPH relies on the older artificial viscosity formulation for capturing shocks. The equations for the time-evolution of the particle's velocity at a fiducial particle "a" can then be written as

$$\frac{dv_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right)(1 - \alpha\mu_{ab} + \beta\mu_{ab}^2)\,\nabla_a W_{ab} \tag{2}$$

and the equation for the time-evolution of the specific thermal energy can be written as

$$\frac{du_a}{dt} = \frac{1}{2}\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right)(1 - \alpha\mu_{ab} + \beta\mu_{ab}^2)v_{ab}\cdot\nabla_a W_{ab}. \tag{3}$$

In the equations above, the derivatives on the left-hand sides are total derivatives since the advection is represented by the particle's motion. P_a is the pressure at particle "a" and is evaluated using the energy u_a, the density ρ_a and the equation of state. μ_{ab} is a shock term and α and β are coefficients of linear and quadratic viscosity and v_{ab} is the difference in velocity between particle "a" and "b."

The shock term μ_{ab} that is used in SPH is different from the shock terms that are used in most parabolized finite difference codes. Such finite difference codes rely on a scalar artificial viscosity [17], or a tensor artificial viscosity [16]. A direct transcription of the von Neumann and Richtmeyer artificial viscosity to SPH gives poor results [9]. A better representation of one-dimensional shocks is obtained by treating the particles as molecules and defining a viscosity that is modelled after molecular interactions [8]. Needless to say, the addition of such a viscosity term in the discrete equations does not translate into parabolic terms (like the von Neumann Richtmeyer viscosity) when we make the transcription back again from the discrete equations to the original partial differential equations. In multidimensions even this formulation breaks down since it was shown in [2] that it generates a substantial amount of entropy in regions of strong shear even when there is no compression. A way of stopping the spurious entropy generation while retaining the good one-dimensional shock capturing was found in [2], where the form for the viscosity term

$$\begin{aligned}\mu_{ab} &= \frac{h v_{ab}\cdot x_{ab}}{(c_a + c_b)(x_{ab}^2 + \eta^2)}(f(a) + f(b)), \quad \text{if } v_{ab}\cdot x_{ab} < 0,\\ &= 0, \qquad\qquad\qquad\qquad\quad \text{otherwise,}\end{aligned} \tag{4}$$

was suggested, where

$$f(a) = \frac{|\langle\nabla\cdot\mathbf{v}\rangle_a|}{|\langle\nabla\cdot\mathbf{v}\rangle_a| + |\langle\nabla\times\mathbf{v}\rangle_a| + 0.0001c_a/h},$$

where $f(a)$ and $f(b)$ are form functions defined as each particle and carry information about the local flow around that particle. In particular, the form function is a suitably devised function of the local compression and the local vorticity. It is designed to approach unity in regions of strong compression and zero in regions of strong vorticity. Here c_a is the local sound speed

at particle "a"; $x_{ab} = x_b - x_a$ and $v_{ab} = v_b - b_a$ are the relative position and velocity, respectively, between particles "a" and "b"; $\langle \nabla \cdot \mathbf{v} \rangle_a$ and $\langle \nabla \times \mathbf{v} \rangle_a$ are the divergence and curl, respectively, of the velocity field defined at particle "a." Equation (4) is just a simple, logical extension of the molecular viscosity in [8]. The author believes that this is but a modest solution since important physical problems involving interactions of shocks with shear layers may prove intractable even with this formulation. However, since the present analysis is in one dimension there is no vorticity in the problem and Eq. (4) reduces to the one in [8] which is the best available for SPH. We, therefore, use it in our stability analysis. Since the derivation of the SPH equations of motion required a kernel averaging of the actual Euler equations in nonconservation form, the final forms obtained in Eqs. (2) and (3) are dependent on the order in which the integrals are evaluated in the averaging process. Thus the forms of the right-hand sides of Eqs. (2) and (3) are nonunique and other forms for the SPH time evolution equations are possible. The reader is referred to [11] for other forms, but we accept the forms given above as a starting point for this work. They are also most often used in SPH implementations owing to the greater symmetry of the terms for the particle "a" that is being updated and all the particles "b" that it is interacting with. The analysis of alternative forms for representing the source terms can be done in a way that is completely analogous to the method demonstrated in this paper.

II. b. *Interpolation Kernels in SPH*

Equations (1), (2), and (3) have been stated without any consideration of their order of accuracy. We now address this important issue. It turns out that we cannot give a wholly conclusive answer to that question. The reason is that accuracy estimates for interpolation formulae can only be arrived at when the points one is interpolating between are uniformly spaced. In SPH the particles move and an initial set of uniformly spaced particles may soon become randomly spaced. This makes general accuracy estimates impossible and reliable convergence testing impossible for general multidimensional problems. (An extreme viewpoint may then be that SPH can then only converge with the convergence rate of a Poisson process. The number of particles within the kernel's support then provide the large number statistics. This is an extremely slow convergence rate.) However, we can relate the question of accuracy of the SPH scheme when all the particles are equidistant to the order of accuracy of kernel interpolation. In [10] an effort was made to understand the order of accuracy of kernel interpolation. There are a variety of interpolation kernels that are popularly used in SPH. We catalog one-dimensional forms of several popular interpolation kernels below. Here x is the interparticle distance and h is a characteristic length, called a smoothing length, over which the kernel has substantial support.

For the one-dimensional kernels we normalize the kernels with the smooth length as

$$W(x; h) = \frac{1}{h}\tilde{W}(x; h) = \frac{1}{h}\tilde{W}\left(\frac{x}{h}\right). \tag{5}$$

Here a generic W will stand for M_n, W_n, G, or S kernels defined below. The values we will be interested in are fourier transforms of the normalized kernels of the form

$$\hat{W}(k; h) \equiv \int_{-\infty}^{\infty} \tilde{W}(x; h)e^{ikx}\, dx \tag{6}$$

and in the usual way we have

$$\int_{-\infty}^{\infty} \frac{\partial^n}{\partial x^n}\tilde{W}(x; h)e^{ikx}\, dx = (-ik)^n \hat{W}(k; h). \tag{7}$$

The explicit x-space representation can be obtained trivially by taking the inverse Fourier transform of Eq. (6).

The Gaussian and super-Gaussian kernels denoted by $G(x; h)$ and $S(x; h)$, respectively, have often been used in SPH. They have interesting mathematical properties but are rarely used in practice because of their infinite extent. The Gaussian and super-Gaussian interpolants are given by

$$\tilde{G}(x; h) \equiv \frac{1}{\sqrt{\pi}}e^{-x^2/h^2}; \quad \tilde{S}(x; h) \equiv \frac{1}{\sqrt{\pi}}e^{-x^2/h^2}\left(\frac{3}{2} - \frac{x^2}{h^2}\right) \tag{8}$$

and their Fourier transforms are given by

$$\hat{G}(k; h) = he^{h^2k^2/4}; \hat{S}(k; h) = he^{h^2k^2/4}\left(1 + \frac{h^2k^2}{4}\right). \tag{9}$$

Most practical work relies on the use of a family of splines known as monotone splines (not the same as monotonicity in finite difference methods) and denoted by $M_n(x; h)$. Their interpolation and smoothing properties have been thoroughly studied in [15]. The lower members have rather small support, i.e., a small integral or half-integral multiple of h. For example, M_4 has a support of 2 h, M_5 has a support of 2.5 h, and M_6 has a support of 3 h. Note that the support can still range over a large number of particles since the ratio of h to the interparticle distance Δx can be rather large. The $M_4(x; h)$, $M_5(x; h)$, and $M_6(x; h)$ are the three most popularly used ones. All monotone splines make a positive definite interpolation of positive definite data. Hence, their obvious use in evaluating the density variable in Eq. (1). The Fourier space representation of the monotone splines is given by:

$$\hat{M}_n(k; h) \equiv \int_{-\infty}^{\infty} \tilde{M}_n(x; h)e^{ikx}\, dx = h\left[2\sin\left(\frac{kh}{2}\right)\Big/ kh\right]^n. \tag{10}$$

The monotone splines only interpolate uniformly spaced data points with second-order accuracy. By applying Richardson

extrapolation to the $M_n(x; h)$ family of splines a new family of splines, denoted by $W_n(x; h)$, was derived in [10]. The individual members in this family of splines have the same interpolation support as the corresponding monotone splines. However, they do not have the positive definite interpolation property. This limits their usefulness in SPH although they may be selectively used in regions of the flow where all the quantities vary gradually. Their key advantage is that $W_n(x; h)$ with $n > 4$ provide fourth-order accurate interpolation of uniformly spaced data,

$$\hat{W}_n(k; h) \equiv \int_{-\infty}^{\infty} \tilde{W}_n(x; h) e^{ikx}\, dx = h\left[2\sin\left(\frac{kh}{2}\right)\Big/ kh\right]^{n-1} \left[\left(1+\frac{n}{2}\right)\left(2\sin\left(\frac{kh}{2}\right)\Big/ kh\right) - \frac{n}{2}\cos\left(\frac{kh}{2}\right)\right]. \tag{11}$$

The coefficients for M_4, M_5, M_6, W_4, W_5, and W_6 for use in two and three-dimensional problems have been explicitly tabulated in [2].

II. c. *Some Comparisons of the SPH Method with Finite Difference Methods*

To clearly lay down points that readers familiar with high quality finite difference methods might have guessed by now: (1) SPH does not have local conservation which is available in finite difference methods by discretizing the equations in flux form. The time evolution equations are not derived from the conserved forms of the Euler equations but rather from an averaged version of the Euler equations in nonconservative form. (2) It does have global conservation (proof given in [11]). (3) Stability and accuracy restrict the form of the time update in finite difference calculations. Accuracy considerations have been used in constructing SPH time-integration schemes. However, prior to the present paper, stability considerations do not seem to have been used in designing SPH time update strategies. (4) The method relies on an artificial viscosity formulation for shock capturing. (5) The shock capturing is not of the familiar scalar or tensor viscosity type. The SPH equations, therefore, do not reduce to the parabolized form of the Euler equations. For the parabolized form of the artificial viscosity, we are guaranteed that the smallest wavelengths in the system will dissipate most efficiently because of the very structure of the ∇^2 operator. That is not guaranteed for the SPH viscosity terms. (6) There is no TVD property in the SPH formulation. (7) No discrete entropy functional and E-fluxes, in the sense of [13], have so far been constructed, so the proof of entropy enforcement does not exist. (8) The order of interpolation can only be guaranteed when the particles are equidistant. The initial configuration in an SPH simulation may indeed be set up to have this property. However, once the particles have moved to a new set of random positions in an SPH simulation, the order property does not obtain. (9) The requirement of positive definite interpolation restricts SPH to no better than second order in space and that is achieved only when the particles are equidistant.

II. d. *Stability Analysis*

Because of the Lagrangian nature of the equations of motion, doing a stability analysis around an unperturbed state with constant velocity is equivalent to doing a stability analysis around an unperturbed state with zero velocity and then applying a translation to the modes. Thus we take a state with zero velocity as our unperturbed state. This gives us the expressions for the unperturbed states plus the linear perturbations with wavenumber k,

$$\begin{aligned} &x_a = a\,\Delta x + q(t)e^{ika\Delta x}, \quad v_a = v(t)e^{ika\Delta x}, \quad \rho_a = \rho_0 + r(t)e^{ika\Delta x}, \\ &P_a = P_0 + p(t)e^{ika\Delta x}, \quad T_a = T_0 + t(t)e^{ika\Delta x}, \end{aligned} \tag{12}$$

and we impose the usual assumptions about ideal polytropic gas law (in this work we have taken $\gamma = \frac{5}{3}$ throughout)

$$\frac{P}{\rho} = RT, \quad u = \frac{P}{\rho(\gamma - 1)}. \tag{13}$$

For the rest of this paper we will also make the normalizations

$$\begin{aligned} &\rho_0 = \frac{m}{\Delta x}, \quad c_0 = \gamma R T_0, \quad \tilde{k} = k\,\Delta x, \\ &\tilde{h} = \frac{h}{\Delta x}, \quad \widetilde{\Delta t} = \frac{c_0\,\Delta t}{\Delta x}, \quad \tilde{v}(t) = \frac{v(t)}{c_0}, \\ &\tilde{t}(t) = \frac{t(t)}{T_0}, \quad \tilde{q}(t) = \frac{q(t)}{\Delta x}, \quad \dot{\tilde{v}}(t) = \frac{\Delta x \dot{v}(t)}{c_0^2}, \\ &\dot{\tilde{u}}(t) = \frac{\gamma(\gamma-1)\,\Delta x}{c_0^3}\,\dot{u}(t). \end{aligned} \tag{14}$$

Putting the expressions for x_a and v_a into Eq. (1) above and, linearizing around the unperturbed positions of particles "a" and "b," we get

$$\begin{aligned} \rho_a &= \rho_0 + r(t)e^{ika\Delta x} \\ &= \sum_b m_b W((b-a)\,\Delta x) \\ &\quad + q(t)e^{ika\Delta x} \sum_b m_b \frac{\partial W}{\partial x}\,(x = (b-a)\,\Delta x)(e^{ik(b-a)\Delta x} - 1). \end{aligned} \tag{15}$$

Thus we get an expression relating $r(t)$ to $q(t)$ as

$$r(t) = -q(t)\rho_0 ik F_{vt}(k; h), \tag{16}$$

where $F_{vt}(k; h)$ is a function of the wavenumber k and the smoothing length h. It is explicitly given by

$$F_{vt}(\tilde{k};\tilde{h}) \equiv \frac{-2}{\tilde{k}\tilde{h}^2} \sum_{b=1}^{\infty} \frac{\partial}{\partial v} \tilde{W}\left(v = \frac{b}{\tilde{h}}\right) \sin(\tilde{k}b). \tag{17}$$

At this point it helps to point out an important and useful approximation that is often made in the spatial part of von Neumann stability analyses. Notice that Eq. (17) entails a summation over the spatial variable "b." We, therefore, call it a "discrete in space" form of the equation. For kernels with finite support this summation will be a finite sum. The "discrete in space" forms have no limiting approximations built into them. Many of the summands in this work are most easily evaluated and even obtained in closed form if we make the "continuous in space" approximation where

$$\sum_{b=-\infty}^{\infty} \Delta x \to \int_{-\infty}^{\infty} dx. \tag{18}$$

The error in making this transcription can be explicitly evaluated using the Poisson summation formula and the process for doing that is given in [10]. The approximation is accurate when the wavelength is substantially larger than the smoothing length. Numerical comparisons with the "discrete in space" forms done in the next section will show the "continuous in space" approximation to be rather good, even when the wavelength is comparable to the smoothing length. On making that approximation one has

$$F_{vt}(k;h) \equiv \frac{\hat{W}(k;h)}{h}. \tag{19}$$

Having thus explained the method for doing the spatial part of the von Neumann analysis for one SPH equation, we give the expressions for the others. It must be noted that Eq. (4) cannot be accurately linearized because the "switch" $v_{ab} \cdot x_{ab} < 0$ introduces a fundamental nonlinearity in the equations. In such a case one has two options. The first is to linearize around a time-steady unperturbed flow that has an overall compressive or expansive character to see the effect of this term in either situation. Nozzle flow in engineering or accretion flow in astrophysics could serve as good examples of such flows. We do not do that there since it is equivalent to doing the stability analysis with one or another specified value of the viscosity term, with the entire viscosity term becoming independent of the wavenumber. We find this option unsatisfying. The other option, which we do adopt here, is to linearize without the use of the "switch" and then to require three things of the analysis. First, it must give behavior at long wavelengths that is not too different from the equations without shock terms. Second, at shorter wavelengths it should give no unphysical sound speeds. Third, since the viscosity terms are unique, the parts of the viscosity terms that contribute to the linear stability analysis should give no unphysical growing modes at any wavelength. In fact, it would be nice to require that the SPH equations with the SPH artificial viscosity included show the same strong damping of the smallest wave modes that is guaranteed by the parabolized form of the finite difference equations. It must also be noted that only the linear artificial viscosity term contributes to the linear stability analysis. Thus, without making any adiabatic approximations for the pressure we have to linearize around the pressure terms and around the particle positions in Eq. (2). We do this to get

$$\begin{aligned} \dot{v}(t^n) = {} & \frac{2RT_0\alpha h}{c_0\,\Delta x^2} F_{vs}(k;h)v(t^n) \\ & - RT_0k^2F_{vq}q(t^n) \\ & - RikF_{vt}(k;h)t(t^n). \end{aligned} \tag{20}$$

The linearization of the energy equation (3) now gives

$$\dot{u}(t^n) = -T_0RikF_{vt}(k;h)v(t^n). \tag{21}$$

For the "continuous in space" analysis one has

$$F_{vq}(k;h) \equiv \frac{2\hat{W}(k;h)}{h} - \frac{\hat{W}(k;h)^2}{h^2} \tag{22}$$

$$F_{vs}(k;h) \equiv \frac{k}{h}\frac{\partial}{\partial k}\hat{W}(k;h) + \frac{1}{h}\hat{W}(k;h) - 1. \tag{23}$$

For the "discrete in space" analysis, because we cannot flip derivatives, we have one auxiliary definition,

$$F_{vr}(\tilde{k};\tilde{h}) \equiv \frac{-1}{\tilde{k}^2\tilde{h}^3} \sum_{b=-\infty}^{\infty} \frac{\partial^2}{\partial v^2} \tilde{W}\left(v = \frac{b}{\tilde{h}}\right) (\cos(\tilde{k}b) - 1), \tag{24}$$

and so in the "discrete in space" case we have

$$F_{vq}(\tilde{k};\tilde{h}) \equiv 2F_{vr}(\tilde{k};\tilde{h}) - F_{vt}(\tilde{k};\tilde{h})^2 \tag{25}$$

$$F_{vs}(\tilde{k};\tilde{h}) \equiv \frac{-2}{\tilde{h}^2} \sum_{b=1}^{\infty} \frac{\partial}{\partial v} \tilde{W}\left(v = \frac{b}{\tilde{h}}\right) (\cos(\tilde{k}b) - 1)(b). \tag{26}$$

II. e. *Spatial Filtering*

The position update in SPH is usually done as

$$\frac{dx_a}{dt} = v_{a\ \mathrm{up}}, \tag{27}$$

where it has become clear (see [12, 2]), that the update velocity $v_{a\ \mathrm{up}}$ should be of a form so as to minimize the buildup of small-scale noise in numerical simulations. A simple spatial filter of the form

$$v_{a\,\mathrm{up}} = v_a + \varepsilon \left(\frac{\sum_b m_b v_b W^*_{ab}}{\sum_b m_b W^*_{ab}} - v_a \right) \tag{28}$$

has been proposed in [2]. Alternatively, we can make a slight modification of the XSPH formalism in [12] to get

$$v_{a\,\mathrm{up}} = v_a + \varepsilon \left(\sum_b \frac{2m_b}{(\rho_a + \rho_b)} (v_b - v_a) W^*_{ab} \right). \tag{29}$$

Here ε is a small number between 0 and 1, but preferably kept as close to 0 as possible. The original choice in [12] was to set $\varepsilon = 1$. The kernel in Eqs. (28) and (29) may be different from that used in the dynamics, i.e., Eqs. (1) to (4). The simple XSPH update given by Eq. (2.6) of [12] is likely to filter too much as will be demonstrated later. The form given in Eq. (29) is to be preferred because it has better conservation properties. Both forms reduce in the linear limit to the same form given by

$$v_{a\,\mathrm{up}} = v_0 + v(t)(1 - \varepsilon + \varepsilon F^*_{vt}(k; h))e^{ika\Delta x}. \tag{30}$$

The asterisk in $F^*_{vt}(k; h)$ denotes that it is evaluated using the kernel W^* which may be different from the kernel W used in Eqs. (1) to (4) to evaluate the dynamics.

It is from the above equation that one begins to see why ε should be kept as small as possible. With $\varepsilon = 1$, as one would have in the straight XSPH formulation, all wavenumbers would be modified substantially while our purpose is to modify the propagation as little as possible. This we do by keeping ε as small as one possibly can without building up small scale noise in a given problem. The formulation of some form of nonlinear update that preferentially filters on the smallest scales would be of great value in a situation of this type but it doesn't seem to have been attempted in the literature. In the next section we shall analyze this term further.

At this time it helps to recapitulate for the reader the steps that have gone before. Thus we take a set of particles of mass m and center them at equidistant positions with interparticle distance Δx. This gives the particles a mean unperturbed density $\rho_0 = m/\Delta x$. The unperturbed particles all carry a temperature T_0, a pressure P_0, and an initial sound speed c_0, where $c_0^2 = \gamma R T_0$. The particles are set into oscillation using Eq. (12). Thus the problem reduces to finding the time evolution of the five oscillation amplitudes $q(t)$, $v(t)$, $r(t)$, $p(t)$, and $t(t)$. The equation of state (13) provides one constraint relating $r(t)$, $p(t)$, and $t(t)$. The equation for the density (1) provides another constraint relating $r(t)$ to $q(t)$ in Eq. (16). The remaining evolution of the system is determined by the three equations of evolution, (2), (3), and (27). Thus there are three quantities whose time evolution needs to be written down as a function of their initial values. In this work we take them to be $v(t)$, $q(t)$, and $u(t)$.

Because the perturbations introduced in Eq. (12) are oscillatory, one needs the representation of the interpolation kernels in Fourier (i.e., wavenumber) space. This is provided in Subsection II.b. The particles are located at discrete positions. The Fourier space representation developed in Subsection II.b is strictly valid only in the "continuous in space" approximation, i.e., when the wavelength of the perturbation is much larger than the interparticle spacing. The "continuous in space" approximation is analytically very convenient and in practice it has been seen to give reasonably good answers even beyond the strict domain of its applicability. Thus in Subsection II.d we distinguish between the "continuous in space" approximation and the "discrete in space" formulation. The "discrete in space" formulation is exact in all cases and is developed in Subsection II.d. The equations are normalized using Δx as a unit of length and $\Delta x/c_0$ as a unit of time. The normalized quantities are denoted with a tilde on top to distinguish them from unnormalized quantities. The full normalization of all the quantities is given in Eq. (14).

After imposing the normalizations given in Eq. (14) we get a general matrix equation giving the time rate of update in terms of the field variables at that point.

$$\begin{pmatrix} \dot{\tilde{v}}(t^n) \\ \dot{\tilde{q}}(t^n) \\ \dot{\tilde{u}}(t^n) \end{pmatrix} = \begin{pmatrix} \frac{2\alpha\tilde{h}}{\gamma} F_{vs}(\tilde{k}; h) & \frac{-\tilde{k}^2}{\gamma} F_{vq}(\tilde{k}; \tilde{h}) & \frac{-i\tilde{k}}{\gamma} F_{vt}(\tilde{k}; \tilde{h}) \\ \tilde{F}^*(\tilde{k}, \tilde{h}, \varepsilon) & 0 & 0 \\ -i(\gamma - 1)\tilde{k} F_{vt}(\tilde{k}; \tilde{h}) & 0 & 0 \end{pmatrix} \begin{pmatrix} \tilde{v}(t^n) \\ \tilde{q}(t^n) \\ \tilde{u}(t^n) \end{pmatrix} \tag{31}$$

where $\tilde{F}^*(\tilde{k}, \tilde{h}, \varepsilon)$ is given by

$$\tilde{F}^*(k, h, \varepsilon) \equiv 1 - \varepsilon + \varepsilon F^*_{vt}(k; h). \tag{32}$$

The above equation holds, irrespective of the time-stepping scheme, and can be used to understand the "continuous in time" behavior of SPH. It can also be used to understand the effects of various time-discretizations. Thus we now turn our attention to different schemes for updating the equations in time. It might be noted that Eq. (31) can be used as is to show why the backward Euler update is unstable. We do not do it here because backward Euler has only first-order accuracy in time and is never used in practice.

II. f. *Time Discretization*

We now outline the predictor–corrector scheme. It is second-order accurate in time. The predictor–corrector scheme is derived by realizing that values at $t^n + 1/2$ are intermediate values (superscript T). Thus we predict with

$$\begin{aligned} v_a^{Tn+1/2} &= v_a^n + \dot{v}_a^n \frac{\Delta t}{2} \\ x_a^{Tn+1/2} &= x_a^n + v_{a\,\mathrm{up}}^n \frac{\Delta t}{2} \\ u_a^{Tn+1/2} &= u_a^n + \dot{u}_a^n \frac{\Delta t}{2} \end{aligned} \tag{33}$$

and correct with

$$\begin{aligned} v_a^{n+1} &= v_a^n + \dot{v}_a^{n+1/2}\,\Delta t \\ x_a^{n+1} &= x_a^n + \frac{1}{2}(v_{a\,\mathrm{up}}^{n+1} + v_{a\,\mathrm{up}}^n)\,\Delta t \\ u_a^{n+1} &= u_a^n + \dot{u}_a^{n+1/2}\,\Delta t. \end{aligned} \tag{34}$$

The method as written in the above two equations is a one-level scheme if and only if the two force and energy updates on the right-hand sides of (33) and (34) at times n and $n + \frac{1}{2}$ are carried out explicitly. The von Neumann stability analysis for that case can be rather easily done and will not be spelled out in detail here. The double evaluation of the force and energy update terms increase the computational cost by a factor of 2 and is usually avoided. Thus, as suggested in [1], one does only one force and energy update at time $n + \frac{1}{2}$, obtaining the forces and energy update terms at time $n + 1$ by simple extrapolation as follows:

$$\begin{aligned} \dot{v}_a^{n+1} &= \dot{v}_a^{n+1/2} \\ \dot{u}_a^{n+1} &= \dot{u}_a^{n+1/2}. \end{aligned} \tag{35}$$

This now yields a scheme that is not a one-level scheme. The von Neumann stability analysis for this important, stable, and useful case is not easy to work out and has been done for the reader in some detail in Appendix A. It turns out that the analysis cannot be done, except by using two of the intermediate values and this yields two computational modes. This is typical of time evolution schemes which are higher than first order in time; i.e., they usually (although not always as the example of the predictor–corrector with two updates above shows) introduce computational modes in the scheme and one then has to be very careful to ensure that the computational modes are rapidly damped out.

The other popular update scheme is the leapfrog scheme for which we have the following update equations:

$$\begin{aligned} x_a^{n+1/2} &= x_a^{n-2} + v_{a\,\mathrm{up}}^n\,\Delta t \\ v_a^{Tn+1/2} &= v_a^n + \dot{v}_a^{n-1/2}\frac{\Delta t}{2} \\ u_a^{Tn+1/2} &= u_a^{n-1/2} + \dot{u}_a^{n-1/2}\,\Delta t \\ v_a^{n+1} &= v_a^n + \dot{v}_a^{n+1/2}\,\Delta t \\ u_a^{n+1/2} &= u_a^{n-1/2} + (\dot{u}_a^{n+1/2} + \dot{u}_a^{n-1/2})\frac{\Delta t}{2}. \end{aligned} \tag{36}$$

This scheme, too, introduces two additional computational modes. In the next section we will give the results of doing the von Neumann stability analysis for all the above-mentioned schemes.

III. RESULTS

We now present the results of the analysis carried out in the previous section and in the Appendix. Most of the results will be drawn from an analysis of the predictor–corrector with two force evaluations per timestep. Although this is not how predictor–corrector is usually used, in this form it yields a one-level system and one can cleanly see the results. Later in the section we do, however, discuss in detail the predictor–corrector with one force evaluation and the leapfrog methods.

For the benefit of readers who haven't followed the previous section or the Appendix closely, we provide here some essential background material. Thus, first realize that $\widetilde{\Delta t}$, using the normalization spelled out in Eq. (15), is equivalent to the Courant number. Since most simulations are done at or around a Courant number of 0.5, we shall, unless otherwise stated, display the results for a Courant number of 0.5. $\tilde{h}$ is the ratio of the smoothing length h to Δx the interparticle distance; $\tilde{k}$ is the wavenumber multiplied by Δx the interparticle distance. It ranges from 0 for the longest wavelengths to π for the shortest wavelengths which are $2\,\Delta x$ in length. ε is a parameter defined in Eq. (28) and (29) which determines the amount of spatial filtering; α is a coefficient of numerical viscosity. The von Neumann stability analysis boils down to finding a complex eigenvalue to an eigenvalue problem; see the end of Appendix A. The amplitude of this eigenvalue tells us how much dissipation there is in our system of equations. If it is unity, it implies that the system is essentially dissipationless; if it is greater than unity, it means that the system is unstable; and if it is less than unity, it means that there is some form of dissipation introduced in the system, either by the artificial viscosity or by the discretization. The ideal is to be as close to unity for as large a range of wavenumbers as possible, as is consistent with the requirement of stability. The phase of this eigenvalue gives us information about the propagation speed of the linearized perturbations that we have imposed. The equations of fluid mechanics allow for modes to be propagating along C^+ $(u + c)$, $C^0(u)$, and C^- $(u - c)$ characteristics. Here u is the unperturbed fluid speed

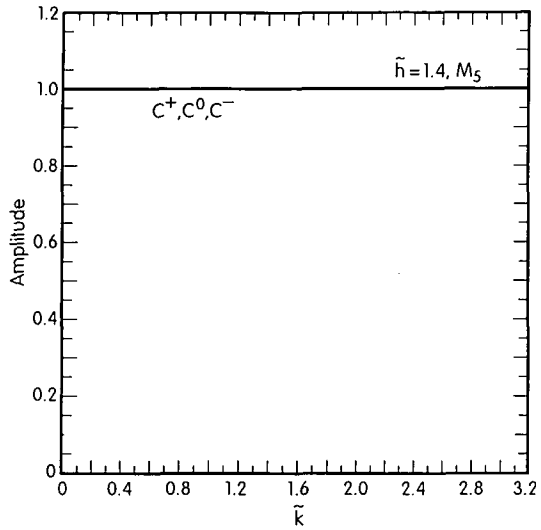


FIG. 1. "Discrete in space" amplitude for the M_5 spline with Courant number of 0.5 and $\tilde{h} = 1.4$.

ourselves in the wavenumbers between 0 and $\pi/2$ since the higher wavenumbers will be substantially modified by the shock and the spatial averaging terms. For all other values of $\tilde{h}$, too, we would find that without artificial viscosity the modes would be dissipationless. Hence we omit the graphs demonstrating so.

In Fig. 2 we show the phases for $\tilde{h} = 1.0$ and $\tilde{h} = 1.2$, again for the M_5 spline, and observe that the results actually seem to improve with decreasing $\tilde{h}$. The amplitudes are exactly as in Fig. 1 and are not shown. Since the computational cost of a SPH scheme depends on how many particles are interpolated over and that it decreases with decreasing $\tilde{h}$, we might even be tempted to think that decreasing $\tilde{h}$ even further might bring further benefit. That is not true as Fig. 2 for $\tilde{h} = 0.65$ shows. Here we notice that the very essential property that we require, i.e., that at small wavenumbers the phases tend to $+1$ and -1, is not met. This means that an SPH code run with $\tilde{h} = 0.65$ will not even propagate sound waves accurately in the long wavelength limit! This value of $\tilde{h} = 0.65$ is of especially great interest because it is the recommended value for the TREESPH code described in [6]. The choice of $\tilde{h} = 0.65$ is just what one obtains if in three dimensions one requires about 32 neighboring particles (the suggested value in TREESPH and several other

quite substantial and we might deem that unacceptable. Thus from the above graphs we conclude that keeping $\tilde{h}$ somewhere between 1.0 and 1.4 interparticle distances is the right range when the M_5 spline is used. Using values of $\tilde{h}$ much smaller than that or much larger than that is unacceptable. This also allows us to deduce certain general and extremely important properties that an interpolating kernel should have in order to be a good kernel for SPH work. First, it should keep the phase reasonably constant over as wide a range of small wavenumbers as possible. Second, it should allow for a reasonable latitude of $\tilde{h}$ values over which the first property holds. Third, and this is more of a practical constraint, it should have as small a domain of support consistent with the first two properties. This third property has to do with computational efficiency. The use of the Gaussian and super-Gaussian kernels in practical calculations is ruled out as a consequence of the third property. This was proved by applying the analysis presented here to the Gaussian and super-Gaussian kernels. Hence we will not explore their properties further here. For the sake of completeness although these kernels have been tabulated in Subsection II.b.

Next we explore the use of the M_4 spline. Figure 3 shows the phases for $\tilde{h} = 0.8$, $\tilde{h} = 1.4$, and $\tilde{h} = 1.6$ for the M_4 spline. Notice two things. First, the deterioration of good long wavelength behavior is present even at $\tilde{h} = 1.4$ and that the falloff in the phases for long wavelengths even at $\tilde{h} = 1.6$ is somewhat more rapid than it was in the case of the acceptable range of $\tilde{h}$ for the M_5 spline. The deeper reason for that is that while both M_4 and M_5 are second-order accurate, M_5 has better

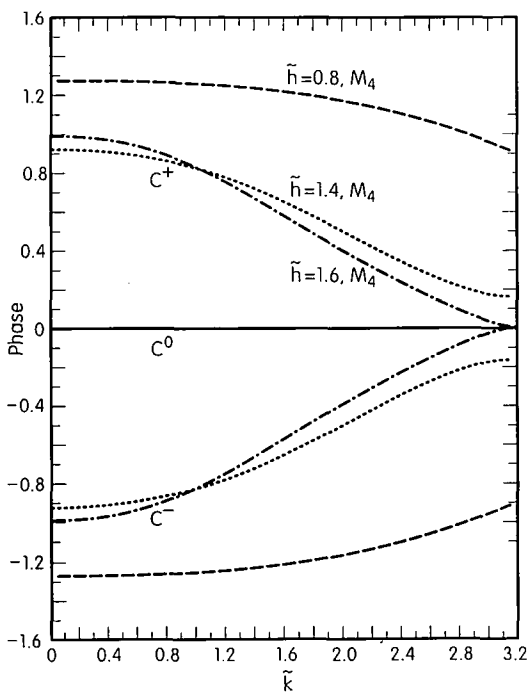


FIG. 3. "Discrete in space" phases for the M_4 spline with Courant number of 0.5 and (a) $\tilde{h} = 0.8$; (b) $\tilde{h} = 1.4$; (c) $\tilde{h} = 1.6$.

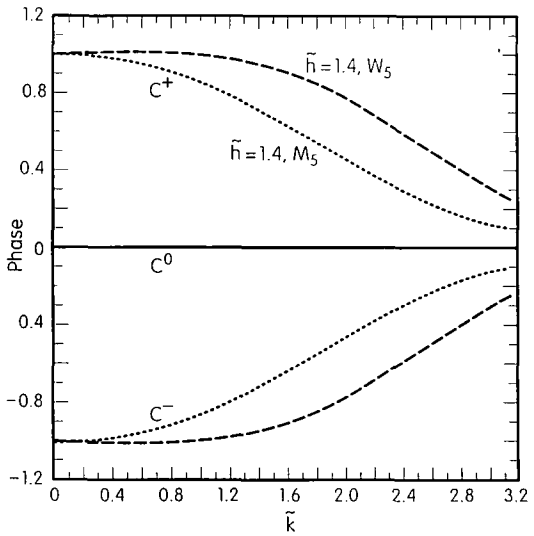


FIG. 4. "Continuous in space" phase for (a) W_5 spline with Courant number of 0.5 and $\tilde{h} = 1.4$; (b) M_5 spline with Courant number of 0.5 and $\tilde{h} = 1.4$.

smoothing properties. For a discussion of smoothing see [15]. Second, at $\tilde{h} = 0.8$, which in three dimensions would correspond to having about 35 particles under the interpolation kernel, the M_4 spline still gives the unfortunate behavior we observed for the M_5 spline when the smoothing length is made too small.

There is no essential problem with using splines like M_6 etc., which have even smoother properties, and carrying out the von Neumann analysis for those cases. It just turns out that, because M_6 has a larger domain of support, its use becomes more computationally expensive.

Now we explore the W_5 spline worked out by [10]. This is not a positive definite spline, but it still has some very good behavior in those situations where it can be used (not all problems have such steep density gradients as to make a spline that does not have the positive-definite property unusable). The reason for its excellence can be made manifest in Fig. 4 which is a "continuous in space" result. Notice there that the phases remain almost flat for an extremely wide range of small wavenumbers. Thus one sees that this von Neumann stability analysis does not just tell us what is good and bad, but it also guides the direction of future effort. Here we realize that construction of splines which have more of the W_5-like character, i.e., third moment preserving, but which are, unlike W_5, positive-definite, would be a very valuable thing. Another valuable suggestion that comes out of this analysis is that in regions of smooth flow one should use the W_5 spline and have some way of smoothly switching from W_5 to M_5 as the flow gets more discontinuous. In finite difference schemes, great benefits have been derived from having local "steepeners" and "flatteners" to locally modify the order of representation of the fluid flow and the present analysis indicates very clearly that SPH would benefit in a similar way. In the next paragraph we take up the issue

of accuracy. In the paragraph following it we take up the issue of efficiency.

The analysis presented above shows us clearly that there is only a limited range in $\tilde{h}$, the ratio of smoothing length to interparticle distance, for which SPH gives acceptable propagation of sonic disturbances. This is true for all the popular kernels used in SPH. Thus none of the SPH kernels has the second important property that we required of SPH kernels in one of the preceding paragraphs in this subsection. We also see quite clearly that a change in this ratio causes a change in the propagation speeds. This is where SPH runs into a fundamental difficulty which we explain as follows. Because of the lagrangian nature of SPH a high density region will be represented as a local condensation of particles and a low density region as a local thinning out of particles. Thus if we keep the smoothing length fixed for all the particles, the propagation of a sound wave will change dramatically as it passes from a dense region to a rarefied region adjoining it. This is because the ratio of smoothing length to interparticle distance changes as the wave propagates across the two regions. This change in propagation is a purely numerical attribute. It should not happen in a good numerical scheme but it does in SPH. An alternative would be to change the local values of the smoothing length [6]. But this is problematic especially at strong shocks. The shocks in [6] are poorer in quality than those obtained with SPH using a constant smoothing length. Moreover, the shocks in [6] are substantially worse than those obtainable using a second-order accurate high-resolution finite difference method [3]. Such a comparison has been made in [2].

While the above paragraph dealt with accuracy, there is also the question of efficiency that needs to be dealt with. For the M_5 spline, which has an extent of 2.5 smoothing lengths, $\tilde{h} = 1.4$ corresponds to having about seven particles to interpolate over in one dimension. Thus for two-dimensional and three-dimensional calculations one needs about 49 and 343 particles, respectively. The use of spherical kernels may reduce these numbers a bit. Regardless of that, this is a lot of particles to be interpolating over. In the language of finite difference methods it means that we need about 49 and 343 zones in two and three dimensions, respectively, to achieve second-order accuracy!

III. b. *The Issue of Resolution in SPH Simulations*

At this point we also need to resolve the issue of resolution in SPH. In almost any method, one has to have a good estimate of what range of wavenumbers are accurately represented. The rigorous mathematical statement of that fact is that all the modes that the scheme claims to resolve have to be propagated with speeds that are within some fraction of their theoretical propagation speed. Say we choose to require that all modes for which we can claim that our SPH scheme is doing a good job are propagated within 10% of the theoretical propagation speed then, if we are using the M_5 spline; we mark out a phase of 0.9 (or -0.9 for the backward propagating sound mode) on the $\tilde{h} = 1.4$ plot in Fig. 2 and ask which wavenumbers are propagated with phases of 0.9 or greater. On doing this we see that this happens at a normalized wavenumber of about 0.9 which corresponds to about 7 Δx. Thus only waves with wavelength longer than seven interparticle spacings are propagated with better than 10% accuracy. This is usually the kind of resolution one obtains in finite difference schemes too. Thus from a fluid dynamical point of view it is wrong to claim that SPH offers resolution down to two interparticle spacings. One must, however, note that the advection mode is accurately propagated for all wavenumbers so that SPH does offer that one advantage.

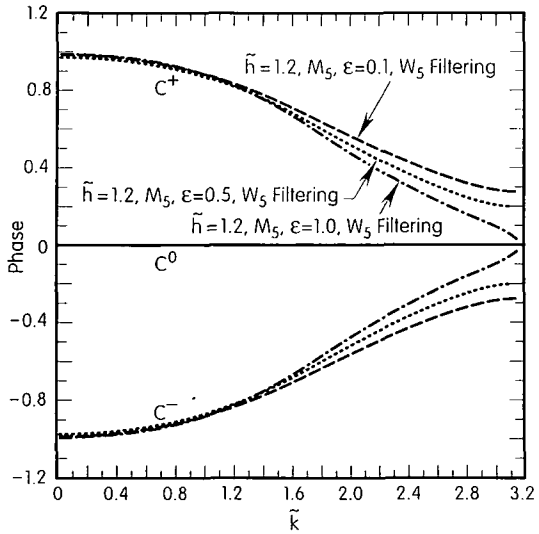


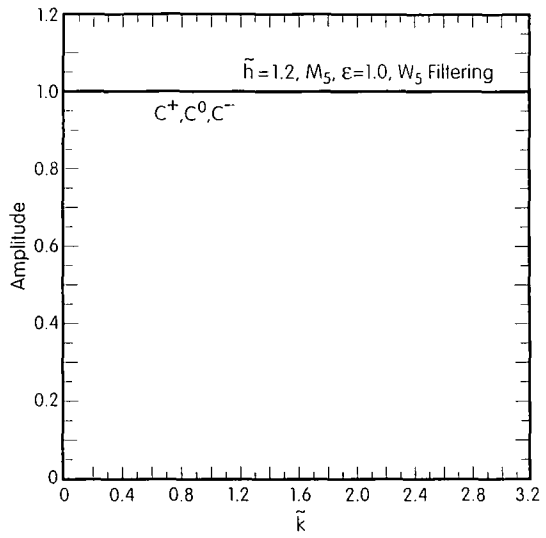
FIG. 5. "Discrete in space" phases for the M_5 spline with Courant number of 0.5, $\tilde{h} = 1.2$, and (a) $\varepsilon = 0.1$; (b) $\varepsilon = 0.5$; (c) $\varepsilon = 1.0$; filtering done with the W_5 spline.

III. c. *Discrete versus Continuous in Space Approximations*

Since the "continuous in space" approximation might be easier to implement on a computer, we briefly display an example to show that it indeed works very well. Thus Fig. 4 shows the phases for $\tilde{h} = 1.4$ for the M_5 spline. Comparing that to the "discrete in space" phases for $\tilde{h} = 1.4$ in Fig. 2 one notices that the differences are very small indeed.

III. d. *The Effect of Spatial Averaging*

Figure 5 gives the phase information, still with no artificial viscosity, but with $\varepsilon = 0.1$, 0.5, and 1.0 filtering done with the W_5 spline. Figure 6 shows the amplitudes for $\varepsilon = 1.0$ filtering done with the W_5 spline. The dynamics is done with the M_5 spline. The amplitudes for the other two cases with $\varepsilon = 0.1$ and 0.5 are exactly like Fig. 6. For the purposes of this analytical work we are not constrained to do a positive-definite interpolation and so we use the W_5 spline which as seen from the



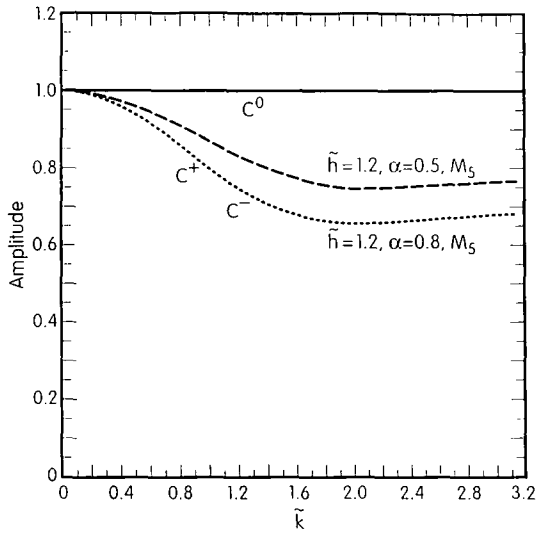


FIG. 7. "Discrete in space" amplitudes for the M_5 spline with Courant number of 0.5, $\tilde{h}$ = 1.2, and ε = 1.0 filtering done with the W_5 spline.

FIG. 7. "Discrete in space" amplitudes for the M_5 spline with Courant number of 0.5, h = 1.2, and (a) α = 0.5; (b) α = 0.8.

discussion above and, also in the references cited there, has better ability to represent a larger range of Fourier modes. From the amplitudes we see that the spatial averaging terms (without the shock terms) are indeed as was deduced, even in [12] for the long wavelength limit, truly nondissipative. Here we verify that that is indeed so for all wavelengths. They do, however, change the propagation speed as seen from Fig. 5. We can see from Figs. 2 and 5 that the spatial averaging with $\varepsilon = 0.1$ is only mildly dispersive, even $\varepsilon = 0.5$ is not so strongly dispersive but that $\varepsilon = 1.0$ is indeed strongly dispersive, especially in the

The result of this is that, at wavelengths comparable to the value of h, the shock terms yield an average measure, tempered by the kernel, of the local dissipation. Thus our assumption leads us to underrepresent the local dissipation at the smallest scales. This explains why the amplitudes seem to flatten out in Fig. 7. The graphs for the phases in Fig. 8 also show that the propagation speeds are slowed down which is as we might expect. We see that the shock terms in Eq. (4) do give us propagation on the largest wavelengths. With no filtering included we get no unphysical sound speeds and no unphysical growing modes at any wavelengths. Thus all the requirements

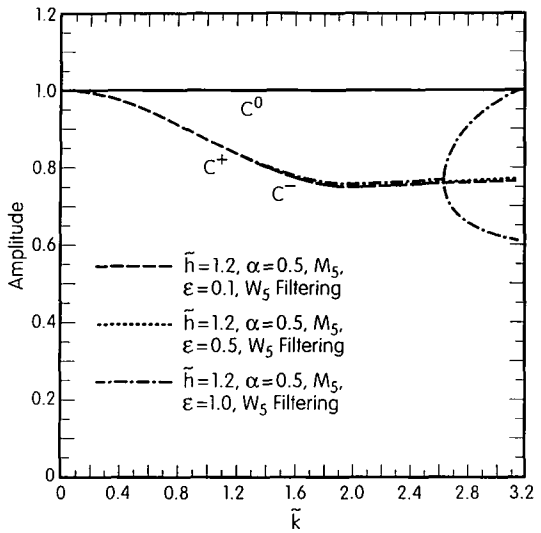


FIG. 9. "Discrete in space" amplitudes for the M_5 spline with Courant number of 0.5, $\tilde{h}$ = 1.2, α = 0.5, and (a) ε = 0.1; (b) ε = 0.5; (c) ε = 1.0; filtering done with the W_5 spline. The dotted and dashed curves for both C^- and C^- almost overlie each other. The dot-dash curves for C^+ and C^- show a bifurcation at large wavenumbers.

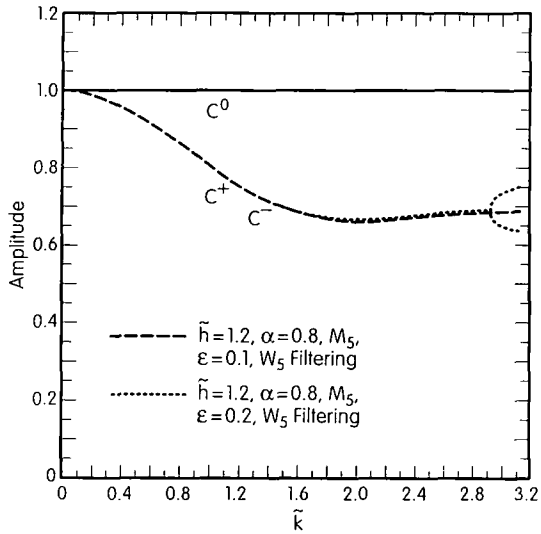


FIG. 11. "Discrete in space" amplitudes for the M_5 spline with Courant number of 0.5, $\tilde{h}$ = 1.2, α = 0.8, (a) ε = 0.1; (b) ε = 0.2; filtering done with the W_5 spline.

III. f. *Interplay between Shock Terms and Spatial Averaging*

We saw in a previous paragraph that the filtering terms couple with the shock terms. See Subsection III(e) and the third term in Eq. (A.7) to explicitly verify this. It turns out that for any second order in time, time-centered scheme this is so. Figures 9 and 10 show amplitudes and phases respectively for α = 0.5 and ε = 0.1, ε = 0.5, and ε = 1.0 filtering done with the W_5 spline. The M_5 spline was used for the dynamics. Figures 11 and 12 show amplitudes and phases respectively for α = 0.8 and ε = 0.1 and ε = 0.2 filtering done with the W_5 spline. One sees that for the smaller value of α there is a fairly large range of ε for which nothing unphysical happens. But for ε = 1.0, notice that the modes change character. The forward and backward propagating modes become completely nonpropagating for a whole range of large wavenumbers. At that point one has to stop distinguishing them as C^+ and C^- waves. This is not a very good thing because it means that information transfer

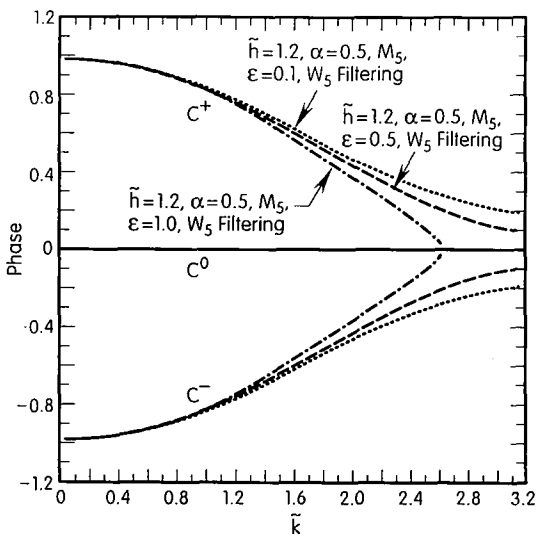


FIG. 10. "Discrete in space" phases for the M_5 spline with Courant number of 0.5, $\tilde{h}$ = 1.2, α = 0.5, and (a) ε = 0.1; (b) ε = 0.5; (c) ε = 1.0; filtering done with the W_5 spline.

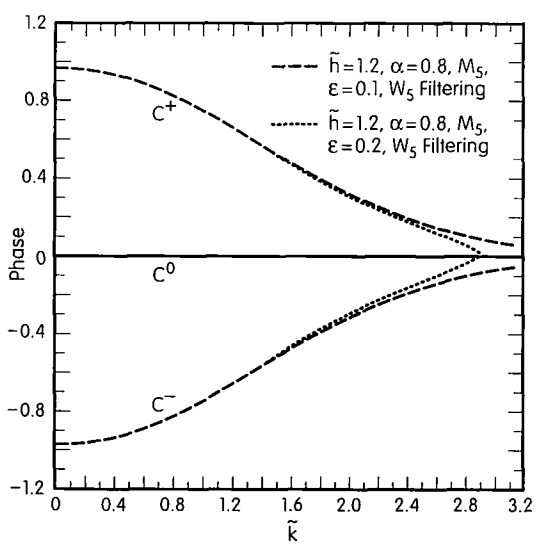


FIG. 12. "Discrete in space" phases for the M_5 spline with Courant number of 0.5, $\tilde{h}$ = 1.2, α = 0.8, and (a) ε = 0.1; (b) ε = 0.2; filtering done with the W_5 spline.

along waves for small wavelengths will be completely wrong. The information along all three will then tend to move with just the fluid flow speed, a situation which is unphysical. Were these modes damped out rapidly, the situation might have been acceptable but Fig. 9 makes it very clear that as ε, i.e., the filtering parameter, increases and, especially as it tends to unity, one of the modes tends to go undamped. At $\varepsilon = 1.0$, it is totally undamped, i.e., its amplitude becomes unity. In a realistic code, one actually wants waves with $\tilde{k}$ greater than $\pi/2$ to be efficiently damped. Of course, in a realistic code the quadratic viscosity also contributes to the damping of small wavelengths and so some of the damping at the highest wavenumbers will also be provided by the quadratic viscosity. The quadratic viscosity cannot be treated in a linear formalism. However, we see that there is a need to treat the filtering terms with caution and at all times to use only the minimum needed to ensure smooth flow. In that regard the comment we made about ε in Section II becomes especially relevant; ε could be made a function of the local attributes of the flow and the position update would, as a consequence, become nonlinear. Figures 11 and 12 show that as α is made larger this deleterious property of the modes sets in at smaller values of ε. Thus, ε must always be held to the minimum needed in every problem.

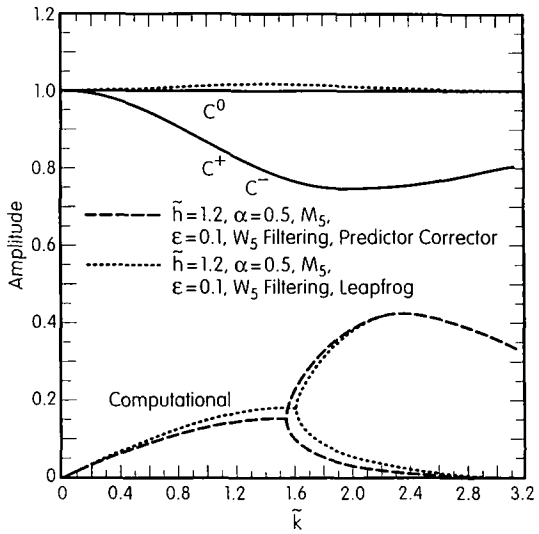


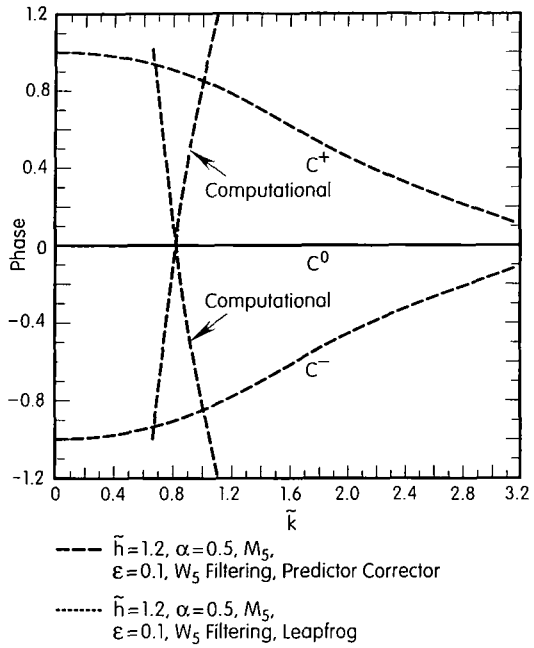
FIG. 13. "Continuous in space" amplitudes for the M_5 spline with Courant number of 0.5, $\tilde{h} = 1.2$, $\alpha = 0.5$, and $\varepsilon = 0.1$; filtering done with the W_5 spline. We have used (a) a predictor–corrector scheme with only one force update per timestep; (b) a leapfrog scheme.

III. g. *Other Timestepping Schemes*

In the above several paragraphs we addressed several issues using the predictor–corrector scheme with two force evaluations per timestep. This yields a one-level system and all the modes are physical modes. We saw that we obtained three such physical modes corresponding to the three characteristics along which information is carried in one-dimensional flow. Now we consider very briefly the other second-order accurate timestepping variants, the predictor–corrector with one force evaluation per timestep and the leapfrog schemes. Both these result in a 5×5 matrix whose eigenvalues give us five growth factors. For the predictor–corrector with one force update, we have given the derivation in Appendix A. As one might expect, one gets two surplus eigenvalues and these are the computational modes that these schemes have. The computational modes should always remain sufficiently small. This is guaranteed if they have sufficiently small growth rates for all possible wavenumbers. Since we have already established in the preceeding paragraphs that certain restrictions and guidelines have to be followed in the choice of $\tilde{h}$, α, and ε, we only demonstrate that use of those guidelines results in nice well-behaved behavior for both the predictor–corrector with one force evaluation per timestep and the leapfrog schemes. We have also demonstrated that the "continuous in space" approximation is good enough and so we use it here. In Figures 13 and 14 we show the amplitudes and phases for the predictor–corrector and leapfrog schemes, respectively. Notice that the computational modes, while moving quite fast (in fact, much faster than the real modes) are indeed very rapidly damped out at all wavelengths. This is as we wanted. Also notice that the propagation of the physical modes in these two schemes is almost as good as the predictor–corrector with two force evaluations that we have previously discussed in such great detail.

IV. CONCLUSIONS

We have shown that only a small range of ratios of smoothing length to interparticle distance for a specified choice of spline lead to optimal continuum behavior. Based on that finding we deduce that none of the currently used SPH kernels represents a particularly good choice. We have shown that the M_5 spline is a better choice than the M_4. For the M_5 spline we have shown that the ratio of smoothing length to interparticle distance should lie between 1.0 and 1.4. When this ratio falls below 1.0 the asymptotic behavior for long wavelengths is physically incorrect. This is commonly taken to be unacceptable behavior in a hydrodynamics algorithm. When this ratio exceeds 1.4 a large range of medium wavelengths have unacceptably large dispersion. Maintaining the ratio between the smoothing length to the interparticle spacing within the acceptable range requires us to have about seven particles to interpolate over in each dimension. In two and three dimensions the number of particles that one needs to interpolate over to achieve the fluid's continuum behavior with second-order accuracy is truly large indeed! We have also been able to quantify the very important issue of resolution in SPH. We see that just as finite difference schemes do not have a true resolving power of two zones, SPH, too, does not have a true resolving power of two times an interparticle distance. In fact, the resolution of sound waves that SPH offers is definitely not better than high rersolution



been able to make algorithmic comparisons between SPH and modern day higher order finite difference schemes. We see that the SPH formulation has no local mass, momentum, and energy conservation, the way all good finite difference schemes do. This lack of local conservation should strongly affect SPH's ability to track down a steady state and iterate it to convergence just as it does for finite difference schemes that are not in the locally conservative form. So far applications of SPH have not included problems where the numerical code was required to converge to a steady state and hold it for several thousand timesteps so no real data is available. No discrete entropy condition has been proved for SPH which is a severe disadvantage in the algorithmic evolution of SPH. There is no TVD property in SPH nor a strong entropy enforcement. The method relies on an artificial viscosity formulation which makes it similar in that respect to finite difference algorithms that are over 30 years old. The order of spatial accuracy in an SPH calculation cannot be guaranteed after the particle positions have randomized to a substantial extent. We also see that for the propagation of sound waves, SPH offers no special advan-

$$-\tilde{t}(t^{n+1}) + \tilde{t}^T(t^{n+3/2}) = -\tfrac{1}{2}(\gamma - 1)\,\widetilde{\Delta t}\, i\tilde{k}F_{vt}(\tilde{k};\tilde{h})\tilde{v}^T(t^{n+1/2}) \quad \text{(A.3)}$$

Now for the velocity update in the corrector step equation (34) we have

$$\begin{aligned}\tilde{v}(t^{n+1}) = \tilde{v}(t^n) &+ \frac{2}{\gamma}\widetilde{\Delta t}\,\alpha\tilde{h}F_{vs}(\tilde{k};\tilde{h})\tilde{v}^T(t^{n+1/2}) \\ &- \frac{1}{\gamma}\widetilde{\Delta t}\,\tilde{k}^2F_{vq}(\tilde{k};\tilde{h})\tilde{q}^T(t^{n+1/2}) - \frac{1}{\gamma}\widetilde{\Delta t}\, i\tilde{k}F_{vt}(\tilde{k};\tilde{h})\tilde{t}^T(t^{n+1/2})\end{aligned} \quad \text{(A.4)}$$

and on substituting Eq. (A.1) in it we get

$$\begin{aligned}\tilde{v}(t^{n+1}) = &\left[1 - \frac{1}{2\gamma}\widetilde{\Delta t}^2\,\tilde{k}^2F_{vq}(\tilde{k};\tilde{h})\tilde{F}^*(\tilde{k},\tilde{h},\varepsilon)\right]\tilde{v}(t^n) \\ &- \frac{1}{\gamma}\widetilde{\Delta t}\,\tilde{k}^2F_{vq}(\tilde{k};\tilde{h})\tilde{q}(t^n) \\ &+ \frac{2}{\gamma}\widetilde{\Delta t}\,\alpha\tilde{h}F_{vs}(\tilde{k};\tilde{h})\tilde{v}^T(t^{n+1/2}) - \frac{1}{\gamma}\widetilde{\Delta t}\, i\tilde{k}F_{vt}(\tilde{k};\tilde{h})\tilde{t}^T(t^{n+1/2}).\end{aligned} \quad \text{(A.5)}$$

From the position update in the corrector step, Eq. (34) we get

$$\tilde{q}(t^{n+1}) - \frac{\widetilde{\Delta t}}{2}\tilde{F}^*(\tilde{k},\tilde{h},\varepsilon)\tilde{v}(t^{n+1}) = \frac{\widetilde{\Delta t}}{2}\tilde{F}^*(\tilde{k},\tilde{h},\varepsilon)\tilde{v}(t^n) + \tilde{q}(t^n). \quad \text{(A.6)}$$

Substituting (A.5) into (A.6) we get

$$\begin{aligned}\tilde{q}(t^{n+1}) = &\left[\widetilde{\Delta t}\,\tilde{F}^*(\tilde{k},\tilde{h},\varepsilon) - \frac{1}{4\gamma}\widetilde{\Delta t}^3\,\tilde{k}^2F_{vq}(\tilde{k};\tilde{h})\tilde{F}^*(\tilde{k},\tilde{h},\varepsilon)^2\right]\tilde{v}(t^n) \\ &\times\left[1 - \frac{1}{2\gamma}\widetilde{\Delta t}^2\,\tilde{k}^2F_{vq}(\tilde{k};\tilde{h})\tilde{F}^*(\tilde{k},\tilde{h},\varepsilon)\right]\tilde{q}(t^n) \\ &+ \frac{1}{\gamma}\widetilde{\Delta t}^2\,\alpha\tilde{h}F_{vs}(\tilde{k};\tilde{h})\tilde{F}^*(\tilde{k},\tilde{h},\varepsilon)\tilde{v}^T(t^{n+1/2}) \\ &- \frac{1}{2\gamma}\widetilde{\Delta t}^2\, i\tilde{k}F_{vt}(\tilde{k};\tilde{h})\tilde{F}^*(\tilde{k},\tilde{h},\varepsilon)\tilde{t}^T(t^{n+1/2}).\end{aligned} \quad \text{(A.7)}$$

From the energy update in the corrector step we get

$$\tilde{t}(t^{n+1}) = \tilde{t}(t^n) - (\gamma - 1)\,\widetilde{\Delta t}\, i\tilde{k}F_{vt}(\tilde{k};\tilde{h})\tilde{v}^T(t^{n+1/2}). \quad \text{(A.8)}$$

Substituting (A.5) and (A.8) into (A.2) and (A.3) we get

$$\begin{aligned}\tilde{v}^T(t^{n+3/2}) = &\left[1 - \frac{3}{4\gamma}\widetilde{\Delta t}^2\,\tilde{k}^2F_{vq}(\tilde{k};\tilde{h})\tilde{F}^*(\tilde{k},\tilde{h},\varepsilon)\right]\tilde{v}(t^n) \\ &- \frac{3}{2\gamma}\widetilde{\Delta t}\,\tilde{k}^2F_{vq}(\tilde{k};\tilde{h})\tilde{q}(t^n) \\ &+ \frac{3}{\gamma}\widetilde{\Delta t}\,\alpha\tilde{h}F_{vs}(\tilde{k};\tilde{h})\tilde{v}^T(t^{n+1/2}) \\ &- \frac{3}{2\gamma}\widetilde{\Delta t}\, i\tilde{k}F_{vt}(\tilde{k};\tilde{h})\tilde{t}^T(t^{n+1/2})\end{aligned} \quad \text{(A.9)}$$

and

$$\tilde{t}^T(t^{n+3/2}) = \tilde{t}(t^n) - \tfrac{3}{2}(\gamma - 1)\,\widetilde{\Delta t}\, i\tilde{k}F_{vt}(\tilde{k};\tilde{h})\tilde{v}^T(t^{n+1/2}). \quad \text{(A.10)}$$

Now realize that (A.5), (A.7), (A.8), along with (A.9) and (A.10), form a system of equations relating $\tilde{v}(t^{n+1})$, $\tilde{q}(t^{n+1})$, $\tilde{t}(t^{n+1})$, $\tilde{v}^T(t^{n+3/2})$, and $\tilde{t}^T(t^{n+3/2})$ to their corresponding quantities a unit time earlier. This allows us to impose a growth factor between the two sets of values as shown in

$$\begin{pmatrix}\tilde{v}(t^{n+1})\\ \tilde{q}(t^{n+1})\\ \tilde{t}(t^{n+1})\\ \tilde{v}^T(t^{n+3/2})\\ \tilde{t}^T(t^{n+3/2})\end{pmatrix} = \lambda\begin{pmatrix}\tilde{v}(t^n)\\ \tilde{q}(t^n)\\ \tilde{t}(t^n)\\ \tilde{v}^T(t^{n+1/2})\\ \tilde{t}^T(t^{n+1/2})\end{pmatrix}.$$

Substituting this in (A.5), (A.7), (A.8), (A.9), and (A.10) gives an eigensystem for the growth factor. This eigensystem needs to be solved and its solution yields, in general, complex values for λ. The amplitude of λ then gives the growth or decay of the mode and the phase of λ gives the propagation speed of the mode.

ACKNOWLEDGMENTS

The author thanks Michael Norman for continued interest, encouragement, and support in doing this work. He also thanks Achi Brandt for a prescient one line comment without which this work would not be what it is. This work was done on NCSA's Cray Supercomputer. NASA support on Grant NAGW-2012 is also acknowledged.

REFERENCES

1. U. Anzer, G. Borner and J. J. Monaghan, *Astron. Astrophys.* **176,** 235 (1987).
2. D. S. Balsara, thesis, University of Illinois, 1990.
3. P. Colella and P. R. Woodward, *J. Comput. Phys.* **54,** 174 (1984).
4. R. A. Gingold and J. J. Monaghan, *Mon. Not. R. Astron. Soc.* **181,** 375 (1977).
5. A. Harten, On second order accurate Godunov-type schemes (unpublished).
6. L. Hernquist and N. Katz, *Astrophys. J. Suppl.* **70,** 419 (1989).
7. L. B. Lucy, *Astronom. J.* **83,** 1013 (1977).
8. J. C. Lattanzio, J. J. Monaghan, H. Pongracic, and M. P. Schwartz, *SIAM J. Sci. Stat. Comput.* **7,** 591 (1986).

9. J. J. Monaghan and R. A. Gingold, *J. Comput. Phys.* **52,** 374 (1983).
10. J. J. Monaghan, *J. Comput. Phys.* **60,** 253 (1985).
11. J. J. Monaghan, *Comput. Phys. Rep.* **3,** 71 (1985).
12. J. J. Monaghan, *J. Comput. Phys.* **82,** 1 (1989).
13. S. Osher, *SIAM J. Numer. Math.* **21,** 217 (1984).
14. R. D. Richtmeyer and K. W. Morton, *Difference Methods for Initial-Value Problems* (Interscience, New York, 1957).
15. I. J. Schoenberg, *Quart. J. Appl. Math.* **4,** 45 (1945).
16. J. M. Stone and M. L. Norman, *Astrophys. J. Suppl. Ser.* **80,** 753 (1992).
17. J. von Neumann and R. D. Richtmeyer, *J. Appl. Phys.* **21,** 232 (1950).