JOURNAL OF COMPUTATIONAL PHYSICS 109, 37–52 (1993)

On Particle–Grid Interpolation and Calculating Chemistry in Particle-in-Cell Methods*

PETER J. O'ROURKE AND J. U. BRACKBILL

Group T-3, Los Alamos National Laboratory, Los Alamos, New Mexico 87545

AND

BERNARD LARROUTUROU

CERMICS, INRIA, Sophia-Antipolis, 06560 Valbonne, France

Received October 21, 1991; revised November 18, 1992

A generalized Fisher's equation, which models chemically reacting fluid flow and for which there is an analytic solution for a one-dimensional premixed flame, is solved numerically using three particle-in-cell (PIC) methods and a finite-difference method that is second-order accurate in space. Two of the PIC methods introduce new particle–grid interpolation schemes that use mass matrix formulations. Through comparisons with the analytic solution and through truncation error analyses, it is shown that the two new interpolation schemes are superior to the standard PIC method and are as accurate as the finite-difference method when the fluid velocity is comparable to the flame velocity. All three PIC methods give superior results when the fluid velocity is greater than the flame velocity. Solving chemical rate equations on the grid instead of on each particle yields significant increases in computational efficiency with no significant increases in numerical error.

I. INTRODUCTION

In modeling chemically reacting flow, there is a need to maintain positive concentrations of chemical species and to minimize numerical diffusion that is not easily satisfied by standard numerical algorithms. In recent studies of trace chemical transport by tropospheric circulation, it is argued by some that particle-in-cell (PIC) methods meet these needs best [1, 2].

In PIC methods [3–6] for numerical fluid dynamics, all or part of the dynamical state of a fluid is stored as properties carried by Lagrangian computational particles, and a computational grid is used to help update particle properties. Generally, a three-stage calculation is used to update the particle properties in time. First, the particle properties are interpolated onto the computational grid. Second, the flow field on the grid is advanced in time. Third, the advanced-time flow field on the grid is interpolated back to the particles. The method used to interpolate between the particles and grid can have a significant effect on the accuracy of the calculation. Earlier PIC methods tended to assign to particles the properties of the fluid at the particles' locations on the grid [3]. Recently the numerical diffusion inherent in this earlier prescription has been reduced in the FLIP method [5], in which the *changes* in the fluid properties on the grid are interpolated to the particles in the third stage of the calculation.

Another particle method that has been proposed for chemically reacting flows is the discrete vortex method [7–9]. This is a grid-free method in which each Lagrangian particle carries with it an amount of vorticity and may undergo a random walk to simulate the effects of viscosity. More recently the method has been extended so that each particle carries gradients in temperature and composition [9]. The velocity at each particle's location is obtained using the Biot–Savart law, and since the vorticity of each particle affects the velocity of every other particle's location, computational times increase faster than linearly with the number of particles. In contrast, in PIC methods each particle only directly affects properties at grid points near the particle through the first stage interpolation, and since the number of particles used is proportional to the number of grid points, computational times only increase in proportion to the number of grid points.

The grid is used in PIC methods to calculate interactions between particles. Thus one cannot hope to obtain better

0021-9991/93 $5.00

than grid-scale resolution of physical effects involving fluid element interactions calculated on the grid. Typically, one uses the grid to obtain a velocity field that is used to move the particles [5], or to equilibrate the particle velocities [4], and a pressure field that is used to accelerate the particles. Nevertheless, there are two advantages of PIC methods over purely grid-based methods that have led to a recent resurgence in interest in PIC calculations of fluid flow [6]. First, for a given velocity field on the grid the calculation of convection in PIC methods is very accurate because of the Lagrangian nature of PIC methods. Second, because there are many more particles than computational grid cells, PIC methods offer the promise of subgrid scale resolution of those fluid processes or properties that are not updated on the grid. For example, by endowing each particle with a material type that does not change with time, one can sometimes follow material interfaces quite accurately. The ability to track material interfaces was probably the major motivation for the original development of the PIC method [3].

An extension of the idea of interface tracking with particles is to endow to each particle a chemical composition and solve chemical rate equations for each particle in a fashion that is largely independent of the grid. One can hope thereby to obtain subgrid scale resolution of chemical processes. For example, with this approach, Walton *et al.* seek to minimize perturbations to chemical equilibrium caused by advection [1]. The original motivation for this work was to exploit this idea in flame calculations. We first solved a simple model equation for a premixed laminar flame using an adaptation of the FLIP method. Surprisingly, we found that even though we solved the chemical source term for each particle, the computed flame speed was less accurate than the flame speed obtained with a commonly used second-order finite difference technique on the same grid used in the particle calculation. The reason for this paradox was eventually found to be partly due to truncation errors in the particle–grid interpolation scheme. Two alternative particle–grid interpolation schemes were tested, and these gave more accurate results than the FLIP method scheme. The truncation error analysis also showed that numerical accuracy was not gained by integrating the chemistry on the particles, a fact that was verified in calculations. Thus although we do have some important conclusions concerning the solution of chemical rate equations in PIC methods, a second contribution of this work is that we have devised two new, more accurate methods for interpolating between particles and the grid in PIC method calculations.

In what follows we first present the model flame problem that is to be solved numerically. The model problem, which has an analytic solution, is a generalized Fisher's equation [10]. Some properties of this solution are first discussed because they will help later in interpreting the computational results.

We next describe the particle methods used for solving the model problem. A single PIC method is used in conjunction with three different ways for interpolating between the particles and grid. One interpolation method, which we call Method I, is that used in the FLIP method [5]. Method I is explicit and globally conserves interpolated quantities, but gives rise to a finite difference approximation on the grid that is not locally conservative. The other two interpolation methods are new and are called Methods II and III. In Method II local conservation is obtained by solving implicitly a mass matrix problem for the changes in fluid properties on the grid. Method II gives rise to a standard finite difference approximation for advancing fluid properties on the grid. Method III requires the implicit solution of two mass matrix problems for each computational cycle and gives rise to a standard finite element approximation for advancing grid properties. The computational results obtained with these particle methods are to be compared with those of a standard finite difference scheme, which is next described.

We then give the computational results. Steady-state flame speeds are calculated while varying the grid resolution, number of particles, the interpolation method, and the method for approximating the chemical source term. The accuracy of the methods is first assessed by comparing the computed flame speeds with the exact solution. It is found that the two new interpolation schemes give more accurate flame speeds than Method I, but that the grid-based method gives more accurate flame speeds than all the particle methods for the steady-state problems. Surprisingly, the errors in all the PIC methods are somewhat reduced when the chemical reaction terms are calculated on the grid, instead of on the particles.

Calculations of the model flame in a moving frame of reference show that with all the PIC methods the computed flame speed is independent of frame of reference, but that with the grid-based method the flame speed changes with the frame of reference because of truncation errors associated with calculating convection on the grid. The reason that the grid-based method does so well relative to the particle methods on the above steady flow problem is that the flame is not moving relative to the mesh, and therefore truncation errors associated with differencing the convective terms on the grid are small. The Galilean invariance of PIC method calculations is a property that is highly desirable in numerical calculations of flow fields with moving flame fronts, such as occur in internal combustion engines and other unsteady combustors.

The results of a truncation error analysis of the particle methods are then presented. The analysis shows why there is no accuracy advantage in PIC methods to calculating chemical sources on the particles rather than on the grid. This is because the averaging errors incurred by calculating chemistry on the grid are generally no larger than temporal

truncation errors associated with integrating the chemical source terms. The analysis also shows that the two new interpolation schemes have comparable truncation errors, and that both have smaller truncation errors than Method I. Although neither of the two new schemes is to be preferred on the basis of truncation errors, Method II is more efficient computationally because it requires only one matrix inversion per computational cycle and is therefore recommended for use in PIC calculations.

II. MODEL PROBLEM

For the model flame problem a single transport equation is to be solved for function $Y(x, t)$, which can be thought of as the mass fraction of products of chemical reaction or as the dimensionless temperature:

$$Y_t + v(t)\, Y_x = \nu Y_{xx} + g(Y). \tag{1}$$

The boundary conditions are $Y(-\infty, t) = 0.0$ and $Y(+\infty, t) = 1.0$. The quantity v is the fluid velocity, which depends only on time, ν is a constant diffusivity, and the reaction rate g is given by

$$g(Y) = c(m+1)(1 - Y^m)\, Y^{m+1}. \tag{2}$$

Parameter m plays the role of an activation energy. As we show below, for large m the chemical reaction is confined to a narrow zone in which Y is nearly equal to 1.0, where the temperature is nearly the flame temperature. When $m = 1$, Eq. (1) is Fisher's equation [10].

The advantage of using Eq. (2) for the reaction rate, rather than a more realistic Arrhenius rate [11], is that exact traveling wave solutions to Eq. (1) are obtained [10],

$$Y = \hat{Y}_m = (1 + e^{-m\xi})^{-1/m}, \tag{3}$$

where

$$\xi = \left\{\frac{c}{\nu}\right\}^{1/2} [x - (v - u_f)t],$$

and the flame propagates relative to the fluid with speed $u_f = (\nu c)^{1/2}$. Just as in more realistic flame models [11], the flame speed is proportional to the square root of the diffusivity times the reaction rate, and the flame thickness is proportional to the square root of the diffusivity divided by the reaction rate. This shows ways in which numerical errors can influence computed flame speeds. For example, numerical diffusion augments physical diffusion and can significantly increase computed flame speeds. Numerical errors in approximating the reaction rate can also lead to errors in computed flame speeds.

Another useful relation can be obtained from Eq. (1) by setting $v = u_f$ and integrating from $x = -\infty$ to $x = +\infty$:

$$u_f = \int_{-\infty}^{+\infty} g(Y(x))\, dx. \tag{4}$$

This is used to evaluate the flame speed in some of the numerical calculations.

It will also be helpful in interpreting the numerical results to know something of the structure of the steady-state solution to Eq. (1) in the limit of large activation energy m. Just as in models with Arrhenius reaction rates [11], the flame can be divided into two zones in the limit of large activation energy—one, called the preheat zone, in which there is a balance of convection and diffusion terms, and another, called the reaction zone, in which there is a balance of diffusion and reaction terms. First, from (3) one obtains

$$\lim_{m \to \infty} \hat{Y}_m(\xi) = \begin{cases} e^{\xi}, & \xi < 0 \\ 1, & \xi \geqslant 0 \end{cases}. \tag{5}$$

By direct substitution into Eq. (1), one can show for fixed $\xi < 0$ and m large enough, the reaction term is negligible. Second, by substituting (3) into (2) one obtains

$$g(\hat{Y}_m) = c\hat{Y}_m \frac{e^{m\xi}(m+1)}{(1 + e^{m\xi})^2}. \tag{6}$$

Thus the reaction rate $g = O(m)$ when $\xi = O(1/m)$ and vanishes exponentially outside this region. By writing the diffusion term using stretched variable $\eta = m\xi$, one can show

$$\nu(\hat{Y}_m)_{xx} = c\hat{Y}_m \frac{1 - me^{\eta}}{(1 + e^{\eta})^2}. \tag{7}$$

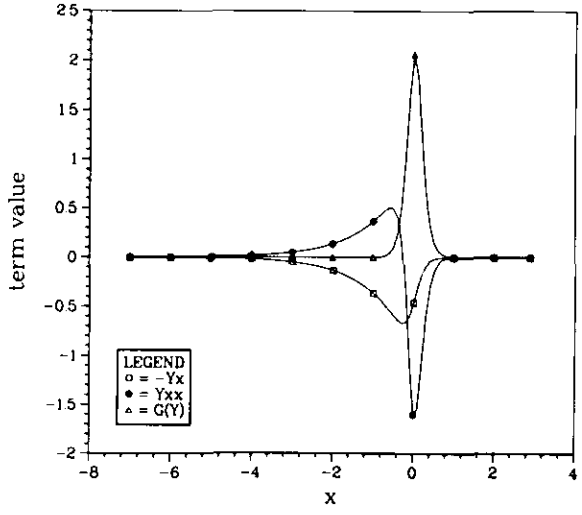

FIG. 1. Plots of $-Y_x$, Y_{xx}, and $G(Y)$ for the steady solution of Eq. (1) when $m = 8.0$ and $c = \nu = 1.0$.

Thus when $m \gg 1$ the diffusion and reaction terms nearly balance and the convection term is negligible in a thin reaction zone, where $\eta = O(1)$.

In Fig. 1 are plotted the convection, diffusion, and reaction terms for the steady solution of Eq. (1) when $m = 8.0$ and $c = v = 1.0$. The negative of the convection term is plotted so that the sum of the three terms is always zero. The convection term is thus always negative, the reaction term is always positive, and the diffusion term is at first positive and then negative. We make two observations concerning the plots in Fig. 1. First, in addition to the preheat and reaction zones, there is always a very narrow region in which the diffusion term is small and there is a convective-reactive balance. Second, the reaction zone thickness is of order $1/m$ with a proportionality constant between 4 and 6.

III. NUMERICAL FORMULATIONS

In this section we give the numerical methods that are used to solve Eq. (1). A single PIC method is used in conjunction with three different methods for interpolating between the particles and the grid. We first describe the PIC method, give the methods that are used in conjunction with PIC for calculating chemistry on the particles and on the grid, and tell how the flame speeds and fluid velocities are calculated. Then we give the three particle–grid interpolation schemes and discuss some of their properties. Finally, we briefly describe the grid-based method with which the PIC schemes are compared.

A. *The PIC Method*

All the methods in this paper use an Eulerian computational grid with uniform cell size δx and uniform timestep δt. Because of the thinness of the reaction zone, computational efficiency could have been improved by using an adaptive mesh. Computational times were so small, however, that using an adaptive mesh was deemed an unnecessary embellishment. The PIC method uses computational particles with uniform spacing δx_p at all times. This could be done because the fluid velocity v in our example problem was always constant in space. Each particle has location $x_p(t)$ and reaction product fraction $Y_p(t)$. The quantities x_p^n and Y_p^n denote the position and reaction product fraction of particle p at time $t = n\,\delta t$.

Before giving the finite difference approximations we introduce functions that are used to interpolate between the particles and the grid. Let $S(x)$ be an even function satisfying

$$S(i\,\delta x) = \begin{cases} 1, & i = 0, \\ 0, & i \neq 0 \end{cases} \tag{8a}$$

$$S(x) \geqslant 0, \tag{8b}$$

and,

$$\sum_i S(x - i\,\delta x) = 1, \qquad -\infty < x < \infty. \tag{8c}$$

In this paper we use the linear interpolation function

$$S(x) = \begin{cases} 1 - \dfrac{|x|}{\delta x}, & |x| < \delta x, \\ 0, & |x| > \delta x, \end{cases}$$

but other choices are possible [12, 13]. For brevity we write $S_{i,p}^n = S(x_p^n - i\,\delta x)$. The quantity $S_{i,p}^n$ can be thought of as the fraction of some property of particle p that is interpolated to grid point i at time $t = n\,\delta t$.

A three-stage computational cycle is used in the PIC method to update the particle product fractions in time. First, we interpolate the particle product fractions to the grid to obtain Y_i^n, the value of Y at spatial location $i\,\delta x$ and time $n\,\delta t$. Details of this first stage are given later.

Second, we calculate diffusion explicitly on the grid using the standard second-order accurate difference approximation:

$$\frac{\delta Y_i}{\delta t} = v\,\frac{Y_{i+1}^n - 2Y_i^n + Y_{i-1}^n}{\delta x^2}. \tag{9a}$$

Equation (9a) can be written in a slightly different way that will be used later. Defining the number of particles N_i^n associated with grid point i by

$$N_i^n = \sum_p S_{i,p}^n \tag{10}$$

and noting that $N_i^n = \delta x/\delta x_p$ when δx_p divides δx evenly, which is true in our calculations, we obtain the alternate form of Eq. (9a):

$$N_i^n\,\frac{\delta Y_i}{\delta t} = v\,\frac{Y_{i+1}^n - 2Y_i^n + Y_{i-1}^n}{\delta x\,\delta x_p}. \tag{9b}$$

Finally, we interpolate the grid changes back to the particles and integrate the chemical source term:

$$Y_p^{n+1} - Y_p^n = (\delta Y_p)_{\text{grid}} + (\delta Y_p)_{\text{chem}}. \tag{11}$$

The change $(\delta Y_p)_{\text{grid}}$ in Y_p interpolated from the grid will be given later. The approximation to $(\delta Y_p)_{\text{chem}}$ is obtained by operator splitting and solving the chemical rate equation

$$\frac{dY}{dt} = g(Y). \tag{12}$$

Equation (12) can be integrated by parts to obtain

$$H(Y_p^n + (\delta Y_p)_{\text{chem}}) = \exp[cm(m+1)\,\delta t]\, H(Y_p^n), \quad (13)$$

where

$$H(Y) = \frac{Y^m}{1 - Y^m} \exp\left(-\frac{1}{Y^m}\right).$$

The chemical source is then, in a sense, found exactly by solving (13) implicitly for $(\delta Y_p)_{\text{chem}}$. Use of Eq. (13), rather than an implicit difference approximation to (12), is more time-consuming computationally, but has the advantage that it eliminates a temporal truncation error that would otherwise obscure the interpolation errors that concern us here.

Low order implicit difference formulas are usually used when approximating chemical source terms in computations of reactive flow because of their robustness and the fact that exact integrals of chemical rate equations are rarely available. In preliminary calculations for this study we used a first-order implicit approximation to $(\delta Y_p)_{\text{chem}}$ and found that this resulted in errors in the computed flame speed that were comparable to errors associated with particle–grid interpolation. These temporal truncation errors will still be present, and may be the largest errors, in PIC calculations of realistic reactive flows.

Equation (13) is used when chemistry is calculated on the particles. When chemistry is calculated on the grid, another term $N_i^n(\delta Y_i)_{\text{chem}}$ is added to the right-hand side of (9b), where $(\delta Y_i)_{\text{chem}}$ is obtained by using Y_i^n in place of Y_p^n in (13). Then both the diffusion and chemical changes are contained in $(\delta Y_p)_{\text{grid}}$, and $(\delta Y_p)_{\text{chem}}$ is zero.

The velocity v is specified in one of two ways. In calculations in which the flame is in a moving frame of reference, v is a prescribed constant. These will be referred to as unsteady calculations. In calculations in the frame of reference of the flame, the velocity v is calculated so that the total amount of reaction product in the mesh is maintained at a constant value [14]. These will be referred to as steady-state calculations because the steady flame solution is then obtained as the long-time limit of a transient calculation. The procedure for determining $v(t)$ in steady-state calculations will now be described. The computational region extends from $x = 0$ to $x = L$, where L is the length of the computational mesh. After the three-stage calculation described above, the total amount of reaction product in the mesh has increased because of chemical reaction. (It will be shown later that the diffusion calculation conserves the total amount of reaction product.) Thus there is an $x_R < L$ such that

$$\int_0^{x_R} \mathbf{Y}(x)\, dx = I_0, \quad (15)$$

where $\mathbf{Y}(x)$ is the product distribution determined from the particles:

$$\mathbf{Y}(x) = \sum_p Y_p^{n+1} \chi\left\{\frac{x - x_p^n}{\delta x_p}\right\}, \qquad \chi(\phi) = \begin{cases} 1, & |\phi| < \frac{1}{2} \\ 0, & |\phi| \geqslant \frac{1}{2}, \end{cases} \quad (16)$$

and I_0 is the amount of reaction product initially in the mesh. We choose v^{n+1} in order to flux enough product out of the mesh to maintain the total amount of product equal to I_0. Since particles enter the upstream boundary with $Y_p = 0$, as required by the boundary condition given after Eq. (1),

$$v^{n+1} = \frac{L - x_R}{\delta t}. \quad (17)$$

After determining v^{n+1}, the particle positions are updated by

$$x_p^{n+1} = x_p^n + v^{n+1}\,\delta t. \quad (18)$$

B. *The Interpolation Schemes*

We now describe the interpolation methods used in the first and third stages of the PIC calculations. Method I is the explicit scheme that is used by the FLIP method [6]. In stage one the grid values are determined by

$$N_i^n Y_i^n = \sum_p S_{i,p}^n Y_p^n, \quad (19)$$

where N_i^n are given by Eq. (10). In stage three the particle changes are obtained by

$$(\delta Y_p)_{\text{grid}} = \sum_i S_{i,p}^n\, \delta Y_i, \quad (20)$$

where the δY_i are given by (9b).

It is interesting to derive the finite difference approximation that results from Method I for updating the grid values Y_i^n. For simplicity we neglect convection, so that N_i and $S_{i,n}$ are constant in time. We deduce from (19),

$$N_i Y_i^{n+1} = \sum_p S_{i,p} Y_p^{n+1},$$

from (11),

$$N_i Y_i^{n+1} = \sum_p S_{i,p}\{Y_p^n + (\delta Y_p)_{\text{grid}} + (\delta Y_p)_{\text{chem}}\},$$

and finally from (19) and (20),

$$N_i Y_i^{n+1} = N_i Y_i^n + \sum_j \left\{ \sum_p S_{i,p} S_{j,p} \right\} \delta Y_j + \sum_p S_{i,p} (\delta Y_p)_{\text{chem}}. \quad (21)$$

Defining $N_{i,j}$,

$$N_{i,j} = \sum_p S_{i,p} S_{j,p},$$

and denoting the right-hand side of (9b) by R_i, we obtain

$$N_i \frac{Y_i^{n+1} - Y_i^n}{\delta t} = \sum_j \frac{N_{i,j}}{N_j} R_j + \frac{\sum_p S_{i,p} (\delta Y_p)_{\text{chem}}}{\delta t}. \quad (22)$$

The symmetric matrix $N_{i,j}$ will be called the mass matrix because in the limit of a continuous particle distribution

$$\lim_{\delta x_p \to 0} N_{i,j}\, \delta x_p = \int_0^L S(x - i\,\delta x)\, S(x - j\,\delta x)\, dx. \quad (23)$$

The integral in (23) is similar to the mass matrix in Galerkin finite element methods [15]. In PIC fluid dynamics calculations, this mass matrix has the form

$$M_{i,j} = \sum_p m_p S_{i,p} S_{j,p},$$

where m_p is the fluid mass associated with particle p, and thus $M_{i,j}$ has the units of mass. (The reformulation of the FLIP equations using a mass matrix is discussed in Burgess *et al.* [16].) In the calculations of this paper we are taking $m_p \equiv 1$, so that $M_{i,j} = N_{i,j}$. For later reference we note that in the limit of a continuous distribution, our linear interpolation functions give

$$\lim_{\delta x_p \to 0} N_{i,j} \frac{\delta x_p}{\delta x} = \begin{cases} \frac{2}{3}, & i = j \\ \frac{1}{6}, & |i-j| = 1 \\ 0, & |i-j| > 1. \end{cases} \quad (24)$$

From Eq. (21) we see that the spatial difference operator used to update Y_i is an average of the R_j at surrounding grid points. Because of this, the spatial difference approximation is globally, but not locally, conservative. It is globally conservative because in the absence of chemistry

$$\sum_i N_i \frac{Y_i^{n+1} - Y_i^n}{\delta t} = \sum_i \sum_j \frac{N_{i,j}}{N_j} R_j = \sum_j R_j \quad (25)$$

and the R_j sum to zero because each is the difference of fluxes $R_j = f_{j+1/2} - f_{j-1/2}$. To have conservation cell-by-cell, however, we need

$$N_i \frac{Y_i^{n+1} - Y_i^n}{\delta t} = \sum_j f_{i,j},$$

where $f_{i,j} = -f_{j,i}$. The spatial approximation in (22) does not have this property because, in general, $N_{i,j} R_j / N_j$ is unequal to $-N_{j,i} R_i / N_i$.

The motivation for interpolation method II is to obtain the standard difference approximation to the diffusion term in the equation for Y_i and, thereby, make the difference approximation locally conservative. To accomplish this we replace the stage-three interpolation of Eq. (20) by

$$(\delta Y_p)_{\text{grid}} = \sum_i S_{i,p}^n (\delta Y_i)', \quad (26)$$

where

$$\sum_j N_{i,j} (\delta Y_j)' = N_i\, \delta Y_i. \quad (27)$$

The stage-one interpolation of Method II remains the same as in Method I; that is, the Y_i^n are given by Eq. (19). Using similar reasoning to that of Eq. (21) we deduce that Method II gives rise to the following difference approximation for Y_i^n:

$$N_i \frac{Y_i^{n+1} - Y_i^n}{\delta t} = R_i + \left\{ \frac{\sum_p S_{i,p} (\delta Y_p)_{\text{chem}}}{\delta t} \right\}. \quad (28)$$

Note that Method II requires the implicit solution of Eq. (27) for the $(\delta Y_j)'$ at each timestep. In the calculations of this paper a direct tridiagonal matrix solver is used, but in multidimensional PIC calculations an iterative procedure is used [16].

The motivation for interpolation method III is to obtain the finite element approximation to (1), in which the mass matrix multiplies the approximation to $\partial Y / \partial t$ [15]. To accomplish this, we replace the stage-one interpolation of Eq. (19) with

$$\sum_j N_{i,j} Y_j^n = \sum_p S_{i,p} Y_p^n. \quad (29)$$

The stage-three interpolation remains the same as in Method II; that is, the $(\delta Y_p)_{\text{grid}}$ are given by Eqs. (26) and (27). From (9b), (11), (26), (27), and (29) one can show that

$$\sum_j N_{i,j} \frac{Y_j^{n+1} - Y_j^n}{\delta t} = R_i + \left\{ \frac{\sum_p S_{i,p} (\delta Y_p)_{\text{chem}}}{\delta t} \right\}. \quad (30)$$

Because of (27) and (29), Method III requires the solution of two implicit equations for each computational timestep.

A summary of the equations used in stages one and three to interpolate between the particles and grid are given in Table I.

C. *The Grid-Based Method*

The PIC method calculations will be compared with calculations using the following finite difference approximation to Eq. (1):

$$\frac{Y_i^{n+1}-Y_i^n}{\delta t}+v^n\frac{Y'_{i+1/2}-Y'_{i-1/2}}{\delta x}=\nu\frac{Y_{i+1}^n-2Y_i^n+Y_{i-1}^n}{\delta x^2}+\frac{(\delta Y_i)_{\text{chem}}}{\delta t}, \tag{31}$$

where

$$Y'_{i+1/2}=\frac{1}{2}\left[Y_i^n+Y_{i+1}^n+\frac{v^n\,\delta t}{\delta x}\{Y_i^n-Y_{i+1}^n\}\right].$$

The approximation to the convection term is Leith's method [17], which is second-order accurate in space and time. We use the standard explicit diffusion approximation, which is second-order accurate in space, and the approximation to the reaction rate that is used by the PIC methods when the chemical source term is integrated on the grid. At the upstream boundary we specify $Y'_{1/2}=0.0$, and at the downstream boundary upwind differencing is used, so that $Y'_{NI+1/2}=Y_{NI}$, where NI is the number of grid points.

In unsteady calculations the velocity is a specified constant. When the steady-state flame solution is desired, the velocity v^n for the grid-based method is chosen so that the total amount of reaction product in the mesh approaches a prescribed value I_0 (cf. Eqs. (15)–(18) for the particle method),

$$v^{n+1}=v^n+\frac{2I^{n+1}-I_0-I^n}{\delta t\ Y_{NI}^{n+1}}, \tag{32}$$

where

$$I^n=\delta x\sum_{i=1}^{NI}Y_i^n.$$

TABLE I

	Stage one	Stage three
Method I	(19)	(20)
Method I	(19)	(26) and (27)
Method III	(29)	(26) and (27)

We note that Y_{NI} is in the post-flame region and is very nearly equal to unity. Equation (32) is derived in the following manner. By summing (31) over all grid points, multiplying by δx, and using the boundary values of Y', we obtain

$$\frac{I^{n+1}-I^n}{\delta t}+v^nY_{NI}^n=S_{\text{chem}}^n, \tag{33}$$

where S_{chem}^n is the integrated chemical source. Thus to make $I^{n+2}=I_0$, we need to take v^{n+1} so that

$$\frac{I_0-I^{n+1}}{\delta t}+v^{n+1}Y_{NI}^{n+1}=S_{\text{chem}}^{n+1}. \tag{34}$$

Equation (32) is obtained by assuming that S_{chem} and Y_{NI} are independent of time and by subtracting (33) from (34).

IV. RESULTS

Calculations were performed to compare the relative accuracy of interpolation Methods I, II, and III and the grid-based method for the steady-state and unsteady problems, while integrating the chemical source term on the particles and on the grid. Fixed parameters in the calculations were reaction rate coefficient $c=1.0$, diffusivity $\nu=1.0$, activation energy $m=8.0$, length of the computational region $L=10.0$, and $\delta t/\delta x^2=\frac{1}{12}$. This small timestep was used because for the grid difference approximation Eq. (30) associated with Method III, the stability criterion was $\delta t/\delta x^2\leqslant\frac{1}{6}$. Because $c=\nu=1.0$ the exact solution for the flame speed is unity, and the flame thickness is also approximately one. Thus the computational region spanned approximately 10 flame thicknesses, enough so that computational boundaries could be far removed from regions of steep gradients and chemical reaction. In calculations of the steady flame solution, the initial conditions were

$$Y(x,0)=\begin{cases}e^{x-x_0}, & x<x_0\\ 1, & x\geqslant x_0,\end{cases} \tag{35}$$

where $x_0=7$. The value of I_0, the integral used to determine the velocity v in steady flow calculations, was just the integral of this initial distribution:

$$I_0=\int_0^L Y(x,0)\,dx.$$

Varying x_0 between six and eight made no difference to the computational results. Calculations of the steady flame were all integrated to a time $t=5.0$, which is approximately five fluid residence times in the flame. The fluid residence time is

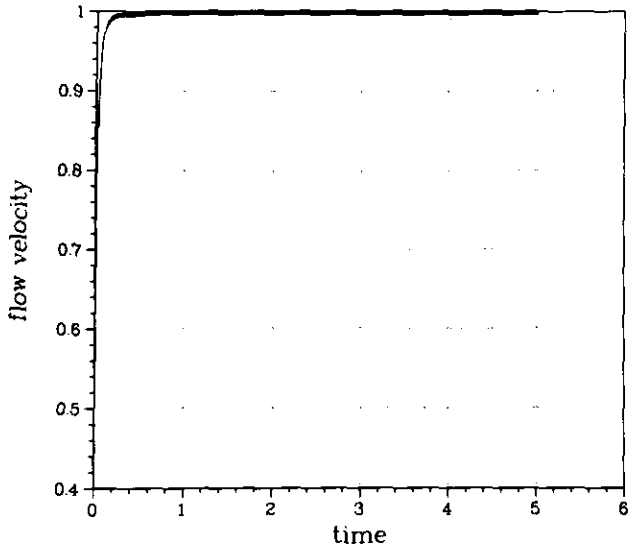

FIG. 2. Flow velocity versus time for the steady flame calculation using Method II, $\delta x = 0.125$, and $\delta x_p = 0.25 * \delta x$.

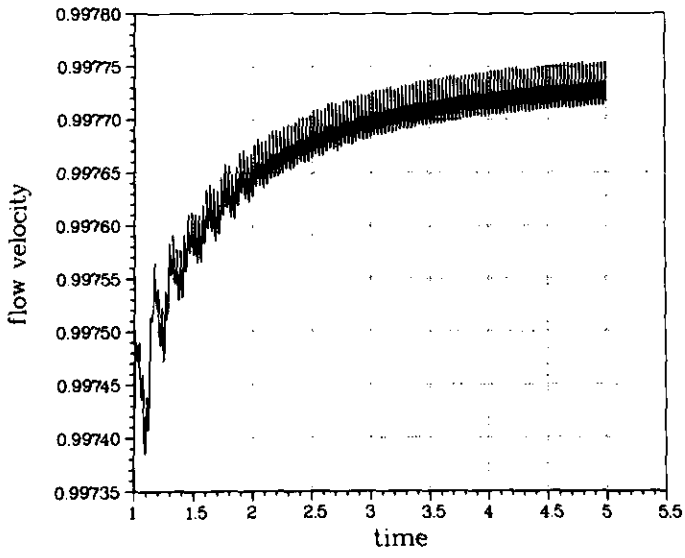

FIG. 3. Flow velocity of Fig. 1 between times $t = 1.0$ and $t = 5.0$ using Method II, $\delta x = 0.125$, and $\delta x_p = 0.25 * \delta x$.

the longest characteristic time associated with this flame problem (see Section II), and thus we expect the computed solution to be approximately steady after several residence times. At time $t = 5.0$ the steady flame speed was determined in a manner described below.

Three groups of calculations were performed. The purpose of the first group of calculations was to assess the relative accuracies of the particle and grid-based methods in the steady flame problem. We computed steady flame solutions using Methods I, II, and III and integrated the chemical source term on the particles. The computational cell size δx was taken to be 0.25, 0.125, and 0.0625, and for each value of δx, three values of δx_p were used: 0.25 δx, 0.125 δx, and 0.0625 δx. These particle calculations were compared with calculations using the grid-based method and the same three values of δx.

The purpose of the second group of calculations was to determine if numerical accuracy was improved by integrating the chemical source terms on the particles, rather than on the grid. We computed steady flame solutions using Methods I, II, and III and integrated the chemical source term on the particles and on the grid. The computational cell sizes were 0.25 and 0.125.

Finally, we performed several unsteady flow calculations

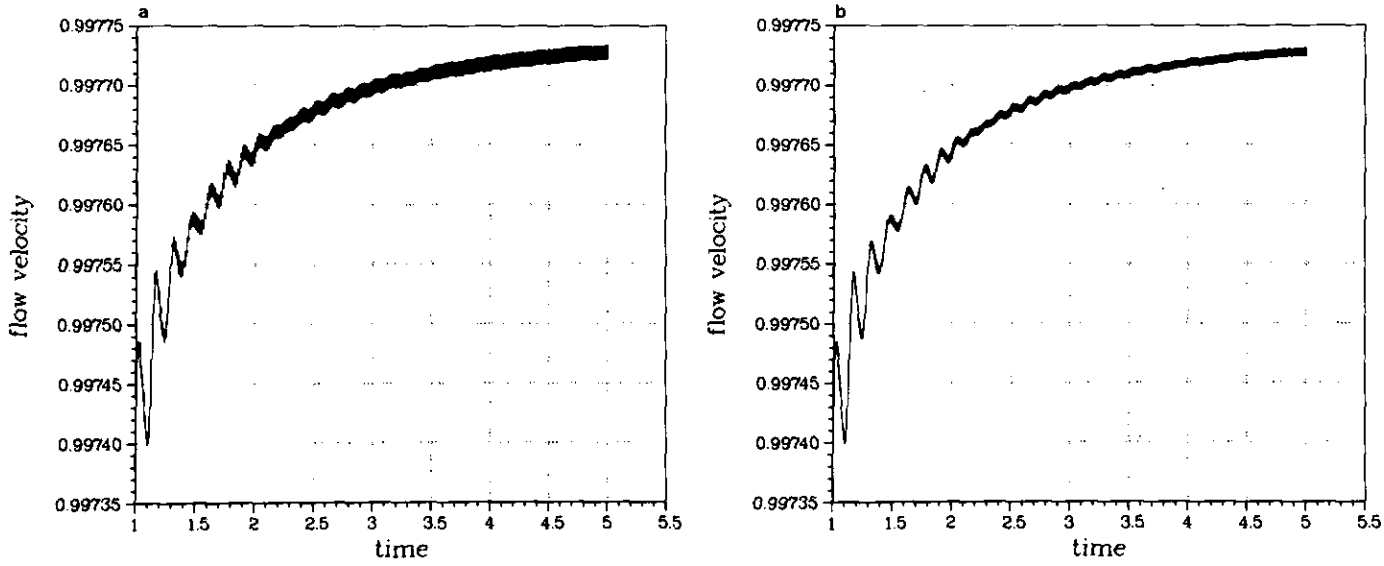

FIG. 4. Flow velocity versus time for the conditions of Fig. 1 with (a) $\delta x_p = 0.125 * \delta x$ and (b) $\delta x_p = 0.0625 * \delta x$, showing the effects of varying particle spacing.

to assess the relative accuracies of the particle and grid-based methods on problems in which the flame was moving relative to the grid. The chemical source term was integrated on the particles and the cell size δx was 0.125. The inlet velocity was taken to be 5.0, 10.0, and 20.0, with the initial conditions taken to be the steady flow solution. The computed flame speed was evaluated by approximating the integral in Eq. (4) with the total chemical source on the particles:

$$u_f = \frac{\delta x_p}{\delta t} \sum_p (\delta Y_p)_{\text{chem}}. \tag{36}$$

TABLE II

	Mesh spacing		
	$\delta x = 0.25$	$\delta x = 0.125$	$\delta x = 0.0625$
Method I	−0.04424	−0.00762	−0.00179
Method II	−0.00685	−0.00226	−0.00056
Method III	0.01418	0.00252	0.00063
Grid	0.01164	0.00118	0.00029

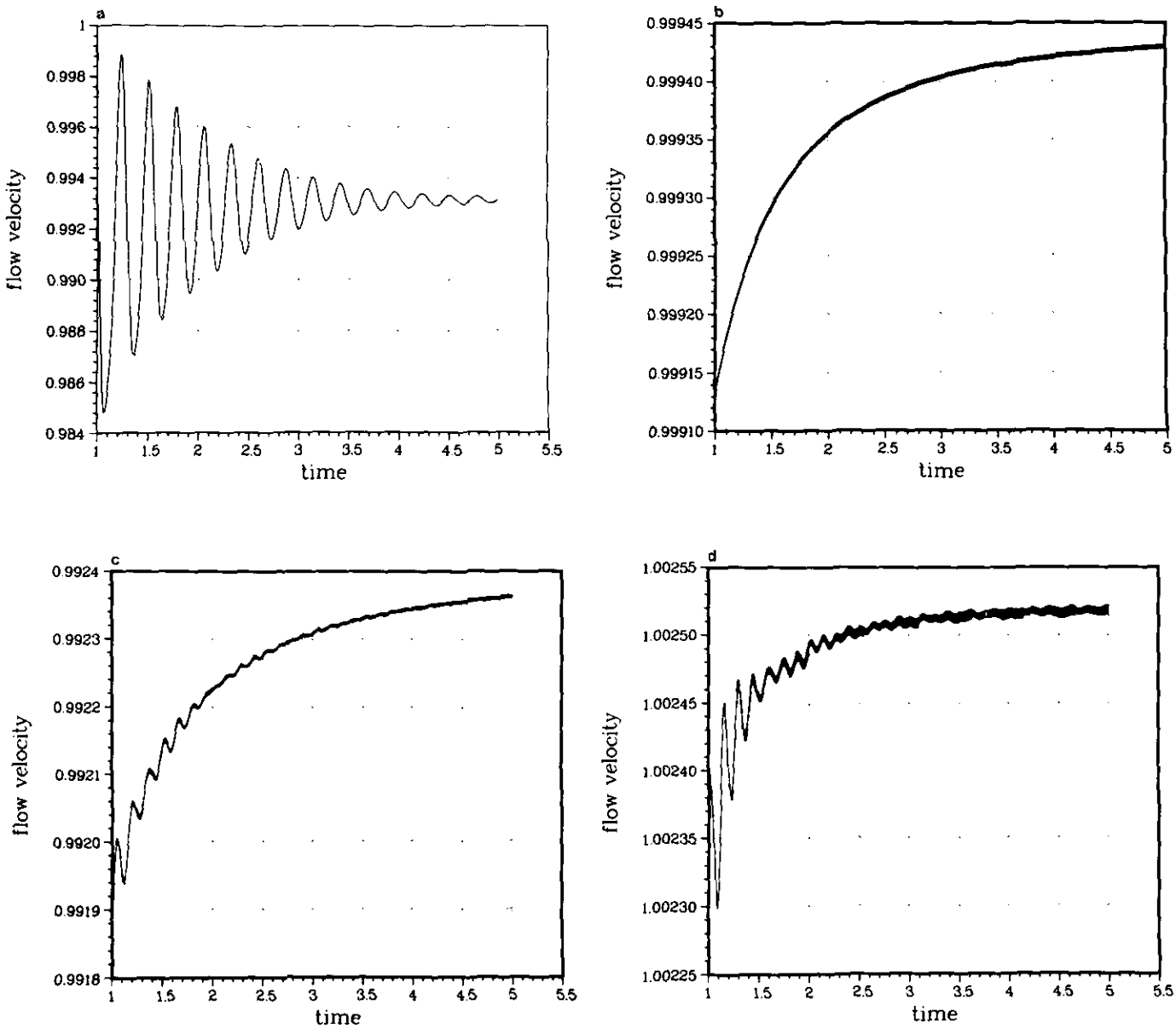

FIG. 5. Flow velocity versus time for the steady flame PIC calculations showing the effects of varying grid spacing and interpolation method: (a) Method II with $\delta x = 0.25$; (b) Method II with $\delta x = 0.0625$; (c) Method I with $\delta x = 0.125$; and (d) Method III with $\delta x = 0.125$.

We first give the computational results and postpone the interpretation and discussion of the results until the next section. Figure 2 shows a typical plot of flow velocity versus time for a steady flow calculation with particles. This is for the case using interpolation Method II with $\delta x = 0.125$ and $\delta x_p = 0.25 * \delta x$. The velocity appears to attain a steady value—the computed flame velocity—within a time of 1.0. Two types of small amplitude, high frequency fluctuations are revealed, however, by plotting the same data, as in Fig. 3, with a greatly enlarged vertical scale. Further, when averaged over the high frequencies, the flow velocity continues to change slowly on a much longer time scale. Thus a truly steady solution is never reached, and the flame speeds we report later are obtained by assuming an exponential approach of the computed flow velocity $v(t)$ to its steady value. After filtering out the high frequency fluctuations, we used the computed flow velocities at times 3.0, 4.0, and 5.0 to determine the three parameters u_f, a, and t_r in the equation

$$u(t) = u_f\left[1 - a\exp\left(-\frac{t}{t_r}\right)\right]. \tag{37}$$

The highest frequency corresponds to the particle injection frequency, $v/\delta x_p$. Increasing the number of particles, as in Fig. 4, reduces the amplitude of these fluctuations, but does not significantly change the flame speed. The next higher frequency is approximately the mesh frequency $v/\delta x$. These oscillations are strongly damped with increasing grid resolution, as shown in Fig. 5. The high frequency oscillations are similar in nature for Methods I, II, and III.

Table II shows the errors in computed flame speeds for the first group of calculations modeling the steady flame problem. The results are independent of δx_p. The most

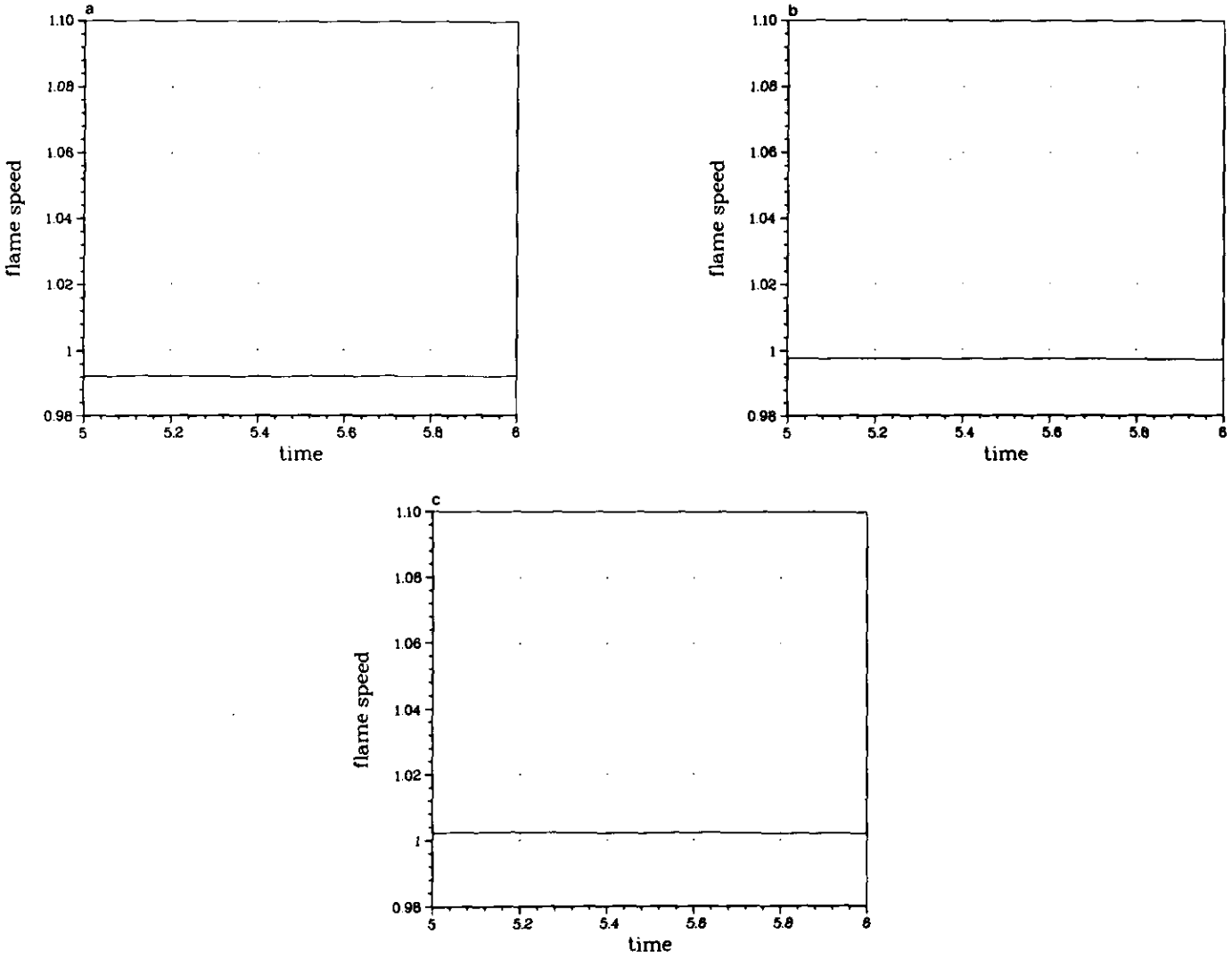

FIG. 6. Flame speed versus time for the unsteady flow calculations using particle Methods I, II, and III (resp. a, b, and c) with flow velocity equal to 20.0.

TABLE III

	Mesh spacing	
	$\delta x = 0.25$	$\delta x = 0.125$
Method I	−0.0233	−0.0052
Method II	−0.0139	−0.0023
Method III	−0.0048	0.0005

surprising result of these calculations is that the particle methods are less accurate than the grid-based method. With the exception of $\delta x = 0.25$, for a given grid cell size the largest errors were associated with Method I and the smallest with the grid-based method. When $\delta x = 0.25$ Method II was most accurate. Except at the largest cell size, the errors of Methods II and III were similar in magnitude and about twice the size of the errors of the grid-based method. All methods showed close to a factor of four decrease in error in reducing δx from 0.125 to 0.0625, which is what one would expect if they were second-order accurate in space. In reducing δx from 0.25 to 0.125, however, relative error reduction varied considerably. Results using the mesh spacing $\delta x = 0.25$ were exceptional in this and subsequent groups of calculations. This might be due to the fact that marginal resolution of the flame is provided by $\delta x = 0.25$, since the reaction zone thickness is of order $1/m$. In fact, we attempted calculations with $\delta x = 0.5$, identical to the above calculations in all other respects, but the com-

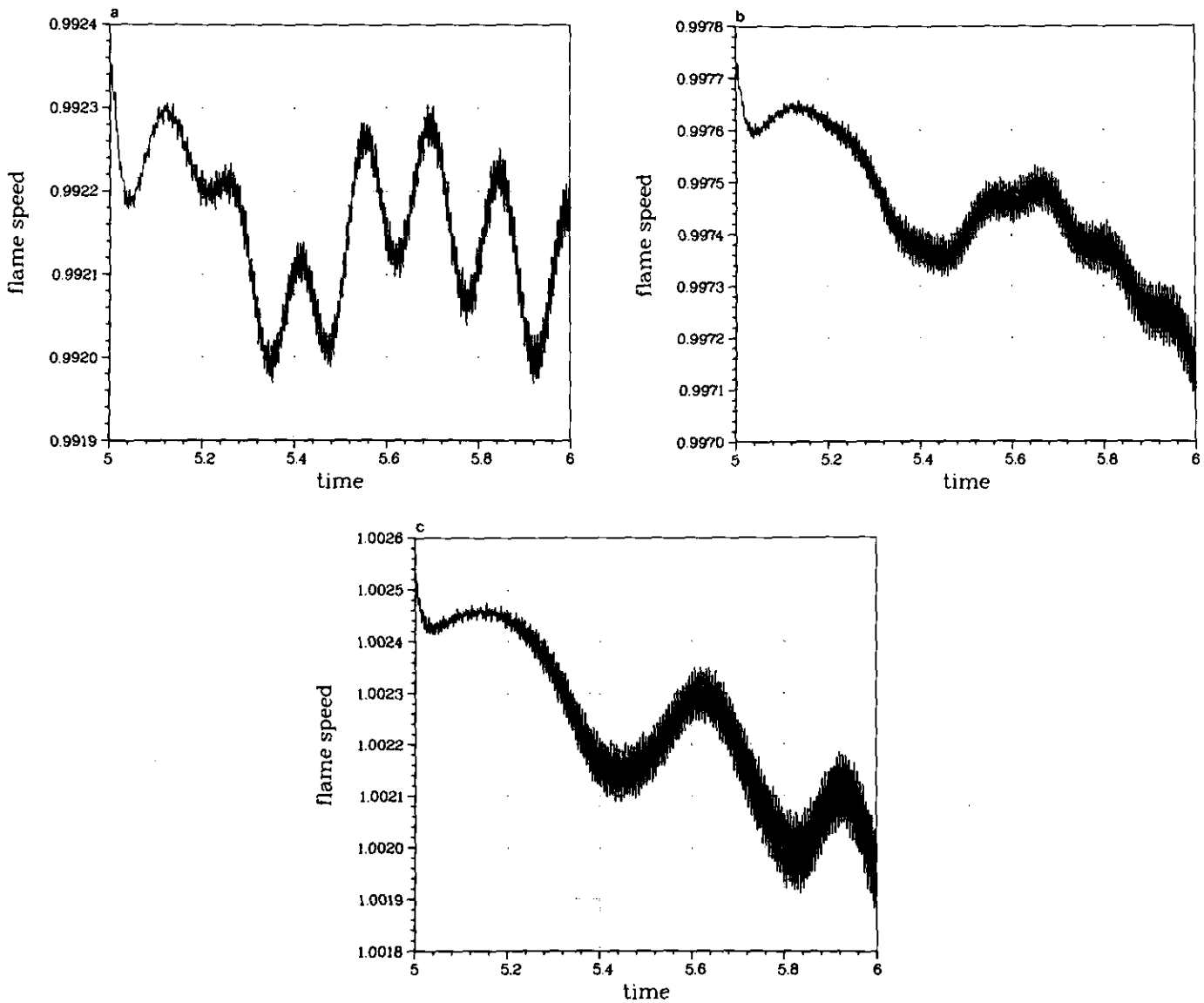

FIG. 7. Flame speeds of Fig. 6 plotted using a greatly expanded vertical scale.

puted value of v dropped to zero, indicating a failure to calculate a sustained burn.

Another surprising result is seen in Table III, which gives the errors in computed flame speeds in the second group of calculations, in which the chemistry terms were integrated on the grid. Comparing Tables II and III shows that in every case the flame speed was calculated more accurately when the chemistry terms were integrated on the grid. Again Method I was least accurate, but now Method III was most accurate, more accurate than the grid-based method. Computing the chemical source term on the grid did have one deleterious effect on the solutions. In this second group of calculations the particle oscillations in the computed flame speed were significantly increased. For example, when Method II was used with $\delta x = 0.125$ and $\delta x_p = 0.125 * \delta x$, the particle oscillation amplitudes were about $5 * 10^{-4}$ with chemistry on the grid and only $1 * 10^{-5}$ with chemistry on the particles. To reduce these particle oscillations to comparable levels to those in the steady flame problem we used 80 particles per cell in the second group of calculations. We point out that this did not result in exorbitant computer times because the chemistry was being integrated on the grid.

The third group of calculations, in which the flame moved relative to the mesh, showed an expected major advantage of all the particle methods over the grid-based method. Once again the results were independent of δx_p. In the particle method calculations, the computed flame speed was nearly Galilean invariant, but in the grid-based method calculations the flame speed changed with changing frame of reference. The computed flame speeds versus time are shown plotted in Figs. 6–8. In each case, these unsteady calculations were initialized with the computed steady solution at time $t = 5.0$ and continued to time $t = 6.0$. Figure 6

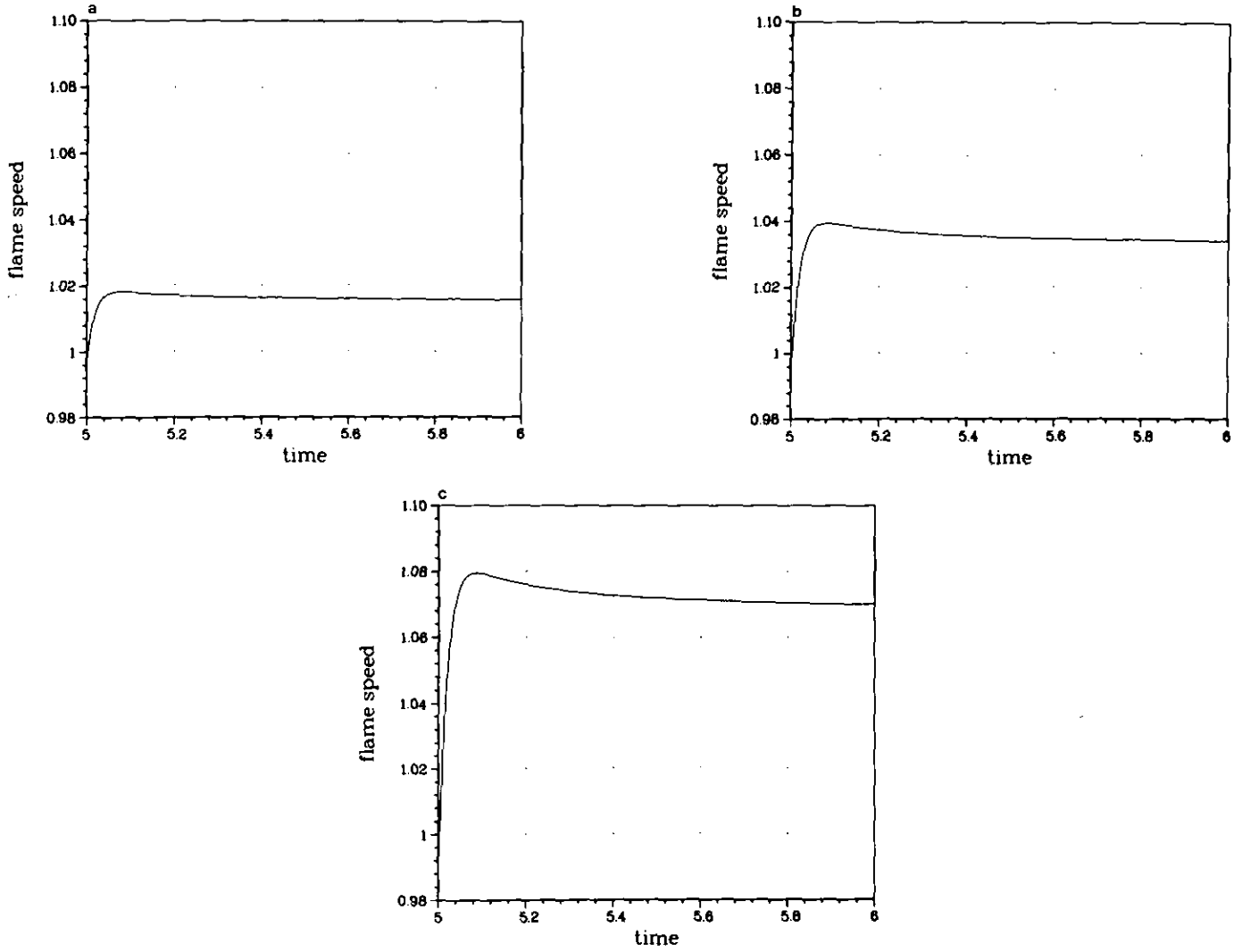

FIG. 8. Flame speed versus time for the unsteady flow calculations using the grid-based method and flow velocities equal to 5.0, 10.0, and 20.0 (resp. a, b, and c).

shows plots of flame speed for the three particle methods when the flow velocity was 20.0, and Fig. 7 shows the same plots with greatly expanded vertical scales. In each case fluctuations in the flame speed were less than $7.0 * 10^{-4}$, despite the fact that the flame translated over about 150 computational cells during the calculations. Note that at time $t = 5.0$, the computed flame speeds, which used Eq. (36), agree with the steady values reported in Table II, which were computed using Eq. (17).

Figure 8 shows the computed flame speeds in the grid-based method calculations when the flow velocities were 5.0, 10.0, and 20.0. Here the changes in computed flame speeds were proportional to the flow velocity and were about 0.07 when the flow velocity was 20.0, two orders of magnitude larger than in the particle methods.

V. INTERPRETATION OF RESULTS

Truncation error analyses [18] were performed for the grid-based and particle methods. The derivation for the grid-based method is straightforward. Outlines of the derivations for the particle methods are given in Appendix A, and here we give only the final results:

Particle Methods-Chemistry on Particles.

$$Y_t + vY_x = vY_{xx} + g - v\frac{\delta t}{2}\left[vY_{xxxx} + 2\frac{dg}{dY}Y_{xx} + \frac{d^2g}{dY^2}Y_x^2\right] + v\,\delta x^2\left[\frac{k}{6}Y_{xxxx} + \frac{(x_{i+1}-x)(x-x_i)}{2\delta x^2}Y_{xxxx}\right], \tag{38}$$

where $x_i \leqslant x \leqslant x_{i+1}$ and

$$k = \begin{cases} 1, \text{ Method I} \\ 0, \text{ Method II} \\ -1, \text{ Method III.} \end{cases}$$

Grid-based Method.

$$Y_t + vY_x = vY_{xx} + g - v\frac{\delta t}{2}\left[vY_{xxxx} + 2\frac{dg}{dY}Y_{xx} + \frac{d^2g}{dY^2}Y_x^2\right] + v\,\delta x^2\left[\frac{1}{12}Y_{xxxx}\right] + v\left\{\delta t\left[vY_{xxx} + \frac{dg}{dY}Y_x\right] - \delta x^2\frac{1}{6}Y_{xxx}\right\}. \tag{39}$$

Particle Methods-Chemistry on Grid.

$$Y_t + vY_x = vY_{xx} + g - v\,\delta t\left[vY_{xxxx} + 2\frac{dg}{dY}Y_{xx} + \frac{d^2g}{dY^2}Y_x^2\right] + v\,\delta x^2\left[\frac{k}{6} + \frac{(x_{i+1}-x)(x-x_i)}{2\delta x^2}\right]Y_{xxxx} + \delta x^2\left[\frac{(x_{i+1}-x)(x-x_i)}{2\delta x^2}G + \frac{1}{12}Y_{xx}\frac{dg}{dY} - \frac{m}{6}G\right], \tag{40}$$

where

$$G = \left\{\frac{d^2g}{dY^2}Y_x^2 + \frac{dg}{dY}Y_{xx}\right\},$$

$x_i \leqslant x \leqslant x_{i+1}$,

$$k = \begin{cases} 1, \text{ Method I} \\ 0, \text{ Method II} \\ -1, \text{ Method III} \end{cases}$$

and

$$m = \begin{cases} 0, \text{ Method I} \\ 1, \text{ Methods II, III.} \end{cases}$$

To perform the truncation error analyses for the particle methods, we assumed for simplicity a continuous particle distribution, in which case Eq. (24) pertains, and expanded the particle solution in a Taylor series on the grid. Only the lowest order truncation errors in space and time were retained, and for all the methods these were second order in space and first order in time. Since $\delta t \sim \delta x^2$ in our calculations, it is clear why second-order convergence of the computed flame speed was obtained.

It is difficult to relate these truncation errors exactly to errors in the computed flame speeds u_f. We can obtain equations for the computed u_f of each method by integrating the steady forms of (38)–(40) and using the boundary conditions imposed on Y. These equations give u_f to lowest order in terms of integrals of truncation errors evaluated using the exact solution *and* in terms of

$$I = \int \frac{dg}{dY}\varepsilon(x)\,dx,$$

where $\varepsilon(x)$ is the difference between the computed and exact solutions. A difficulty is that the computed flame speed is an integrated quantity. This means that not only can truncation errors cancel each other locally, but the effects of a

truncation error on u_f can be offset by the effects of another truncation error at a different place in the mesh. At best we can use the truncation error analyses to give plausible explanations of how errors in u_f have varied. We can also use the analyses as a guide in selecting methods for more general reactive flow problems. This is what we now proceed to do.

We first point out two major differences between the truncation errors of the particle methods and those of the grid-based method. First, in the second-order spatial errors the coefficient of $\delta x^2 Y_{xxxx}$ in the grid-based method is $\frac{1}{12}$ and in the particle methods $k/6 + (x_{i+1} - x)(x - x_i)/(2\delta x^2)$. In the grid-based method this error arises from the difference approximation to the diffusion term. In the particle methods this source of error is still present but is modified by the particle–grid interpolation, and an additional term is added that depends on a particle's position relative to the grid points. For the particle methods the average value of the coefficient is $k/6 + 1/12$, which gives a plausible explanation for why the calculated flame speeds of Method I ($k = 1$) are less accurate than those of Methods II ($k = 0$) and III ($k = -1$). It does not explain why the grid-based method always does better than the particle methods on the steady flow problem. Perhaps there are compensating errors in the grid-based method.

The second major difference between the truncation errors of the grid-based and particle methods, is the presence of errors proportional to the fluid velocity v in Eq. (39) that are not present in Eqs. (38) and (40). These terms arise from the differencing of the convection term and are almost certainly responsible for the grid-based method performing worse than the particle methods in the unsteady problems. To see this we compare the magnitudes of the errors proportional to Y_{xxxx}, which occur in both methods, and vY_{xxx}, which occur in just the grid-based method. In the flame's preheat zone, which has thickness of order unity, $Y_{xxxx} \approx 1$ and $vY_{xxx} \approx v$, and thus the latter term will only dominate the former when $v > 1$. In the reaction zone, which has thickness of order $1/m$, where $m = 8$, $Y_{xxxx} \approx m^3$ and $vY_{xxx} \approx vm^2$. The latter term will only become significant when $v \approx m$. Thus the convective truncation errors are comparable to or smaller than other errors in the steady flow problems, but in the unsteady flow problems they dominate in the preheat zone.

We also point out an important difference between the particle methods with chemistry integrated on the particles and on the grid. The truncation errors of Eq. (40) contain as a subset all those of Eq. (38) and several more second-order spatial errors that arise due to the averaging of Y and of $g(Y)$ that occur when interpolating between the particles and the grid. Since $\delta t = v\,\delta x^2/12$ in our calculations, it can be seen that these averaging errors are comparable in magnitude to the δt errors arising from not time-centering the difference approximations to the diffusion and reaction terms. It is probable that some cancellation occurred between these averaging errors and the δt errors, and this explains the surprising result that the flame speed was more accurately calculated when the chemistry was integrated on the grid.

In fact, the averaging errors will usually be comparable to or smaller than the δt errors in numerical calculations of combustion because we usually have

$$v\,\delta t \gtrsim \delta x^2,$$

since implicit approximations are used for the diffusion term [19]. Since the δt errors are difficult to avoid in integrating stiff chemical rate equations, integrating chemical sources on the grid is preferable to integrating them on particles because accuracy is not degraded but computational efficiency is improved.

VI. CONCLUSIONS

Particle-in-cell (PIC) calculations of a model flame problem have shown that while there are advantages to calculating convective transport using particles there is no advantage to integrating chemical source terms on the particles in flame problems. This is because the averaging errors that are eliminated by integrating the chemistry on the particles are comparable to, or smaller than, temporal truncation errors arising from the diffusion and reaction terms. This conclusion may not be true for atmospheric chemistry problems. When, for example, physical diffusion is small and chemical source terms are accurately integrated in time, the averaging errors incurred by integrating chemistry on the grid may be the dominant numerical truncation errors, and clearly then it would be advantageous to calculate chemistry on the particles.

Two new schemes have been devised that reduce spatial truncation errors that arise because of the particle-grid interpolation in PIC methods. These schemes involve implicit solution of mass matrix problems on the grid and thus are more costly than the original FLIP interpolation scheme, but our experience in two dimensions has been that a 30% increase in cost yields the same increase in accuracy as a calculation with four times as many grid cells [16].

In flame calculations in which flow velocities are comparable to flame velocities, particle methods achieve comparable accuracy to second-order grid-based method if one of the two new schemes for particle–grid interpolation is used. However, PIC methods achieve comparable accuracy at greater cost than grid methods because of greater storage and time requirements.

In flame calculations in which flow velocities are significantly greater than flame velocities, as they often are when using general purpose hydrodynamics codes for

combustion calculations, PIC methos offer the possibility of significant increases in computational accuracy because they eliminate truncation errors that are dominant when flow velocities are large.

In the future application of particle methods to the calculations of combustion, several issues need to be addressed. There needs to be a careful comparison of the computational efficiency of particle methods relative to other methods. Are positivity, low numerical diffusion, and the ability to resolve contact discontinuities worth the extra cost of PIC in a particular problem? There also is a need for a way to extend the range of densities that can be represented by the particles. In real flame problems, unlike our model flame problem, the fluid density usually decreases by a factor of five to eight across the flame [11], and the Lagrangian particles may not provide adequate resolution in the low density post-flame gases without a method to add new particles. Nevertheless, our results indicate that the accuracy of PIC methods equals, and sometimes far exceeds that of grid-based methods, and that further development work on PIC combustion models is well justified.

APPENDIX A: TRUNCATION ERROR ANALYSIS

For the particle methods in the limit of a continuous distribution of particles, the equations solved are the following:

METHOD I.

Step 1. $Y_i = \int S_i(x)\, Y^n(x)\, dx/\delta x$

Step 2. $\delta Y_i = (v\,\delta t/\delta x^2)[Y_{i+1} - 2Y_i + Y_{i-1}] + \delta Y_{i,\text{chem}}$

Step 3. $Y^{n+1}(x) = Y^n(x) + \sum_i S_i(x)\,\delta Y_i + \delta Y(x)_{\text{chem}}$.

METHOD II.

Step 1. Same as Method I

Step 2. $\frac{2}{3}\delta Y_i + \frac{1}{6}(\delta Y_{i-1} + \delta Y_{i+1}) = (v\,\delta t/\delta x^2)(Y_{i+1} - 2Y_i + Y_{i-1}) + \delta Y_{i,\text{chem}}$.

Step 3. Same as Method I.

METHOD III.

Step 1. $\frac{2}{3}Y_i + \frac{1}{6}(Y_{i-1} + Y_{i+1}) = \int S_i(x)\, Y^n(x)\, dx/\delta x$

Steps 2 *and* 3. Same as Method II.

When chemistry is added on the particles, $\delta Y_{i,\text{chem}} = 0$, and

$$\delta Y(x)_{\text{chem}} = \int_{t^n}^{t^{n+1}} g(Y''(x,t))\, dt, \tag{A1}$$

where $\partial Y''/\partial t = g(Y'')$ and $Y''(x, t^n) = Y^n(x)$.

When chemistry is calculated on the grid, $\delta Y(x)_{\text{chem}} = 0$ and

$$\delta Y_{i,\text{chem}} = \int_{t^n}^{t^{n+1}} g(Y_i''(t))\, dt, \tag{A2}$$

where $dY_i''/dt = g(Y_i'')$ and $Y_i''(t^n) = Y_i$.

We now outline the truncation error analysis of the above equations and give some intermediate results that can be used to verify the analysis. First, we expand $Y^n(x)$ in a power series about $x = x_i$ and substitute this into the integral in Step 1 to obtain Y_i in terms of Y^n and its derivatives at x_i. For Methods I and II we obtain

$$Y_i = [Y + \tfrac{1}{12} Y_{xx}\,\delta x^2 + \tfrac{1}{360} Y_{xxxx}\,\delta x^4 + \cdots]_{x=x_i} \tag{A3a}$$

and for Method III,

$$Y_i = [Y - \tfrac{1}{12} Y_{xx}\,\delta x^2 + \tfrac{1}{360} Y_{xxxx}\,\delta x^4 + \cdots]_{x=x_i}. \tag{A3b}$$

In deriving these, use is made of

$$\int S_i(x)(x - x_i)^n\, dx = \begin{cases} 0, & n \text{ odd} \\ \dfrac{2(\delta x)^{n+1}}{(n+1)(n+2)}, & n \text{ even.} \end{cases}$$

Next, the expansions for Y_{i+1} and Y_{i-1} are written in terms of Y and its derivatives at $x = x_i$. Substituting the results into the Step 2 equations gives

$$\frac{\delta Y_i}{\delta t} = v\left[Y_{xx} + k\,\frac{\delta x^2}{6}\, Y_{xxxx} + O(\delta x^4) \right]_{x=x_i} + \frac{\delta Y_{i,\text{chem}}}{\delta t}, \tag{A4}$$

where

$$k = \begin{cases} 1, & \text{Method I} \\ 0, & \text{Method II} \\ -1, & \text{Method III.} \end{cases}$$

The chemical source term on the grid is treated as follows. The integrand in the defining Eq. (A2) is expanded in a power series in time, the time integration is performed, and then Eqs. (A3a) and (A3b) are used to obtain $g(Y_i)$ in terms of g and its derivatives at $x = x_i$. The result is

$$\frac{\delta Y_{i,\text{chem}}}{\delta t} = \left[g + \frac{dg}{dY}\frac{1}{12} Y_{xx}\,\delta x^2 + \frac{\delta t}{2} g \frac{dg}{dY} \right]_{x=x_i} \tag{A5a}$$

for Methods I and II, and

$$\frac{\delta Y_{i,\text{chem}}}{\delta t} = \left[g - \frac{dg}{dY}\frac{1}{12} Y_{xx}\,\delta x^2 + \frac{\delta t}{2} g \frac{dg}{dY} \right]_{x=x_i} \tag{A5b}$$

for Method III.

Before using Eq. (A4) in the equation for Step 3, we must first expand the right-hand side of (A4) about an arbitrary point x that may not coincide with the grid point x_i. After doing this and substituting the result in the Step 3 equation, we obtain for the case where chemistry is integrated on the grid

$$\frac{Y^{n+1}(x)-Y^n(x)}{\delta t} = \nu Y_{xx} + g + \nu\,\delta x^2\, Y_{xxxx}\left[\frac{k}{6}+\frac{(x_{i+1}-x)(x-x_i)}{2\delta x^2}\right] + \frac{\delta x^2}{12}\frac{dg}{dY}Y_{xx} + \frac{\delta t}{2}g\frac{dg}{dY} + \delta x^2\left[\frac{(x_{i+1}-x)(x-x_i)}{2\delta x^2}-\frac{m}{6}\right]G, \qquad \text{(A6a)}$$

where

$$G=\left\{\frac{d^2g}{dY^2}Y_x^2+\frac{dg}{dY}Y_{xx}\right\}$$

and

$$m=\begin{cases}0, & \text{Method I}\\ 1, & \text{Methods II and III,}\end{cases}$$

and when chemistry is integrated on the particles,

$$\frac{Y^{n+1}(x)-Y^n(x)}{\delta t} = \nu Y_{xx} + g + \nu\,\delta x^2\, Y_{xxxx}\left[\frac{k}{6}+\frac{(x_{i+1}-x)(x-x_i)}{2\delta x^2}\right] + \frac{\delta t}{2}g\frac{dg}{dY}. \qquad \text{(A6b)}$$

Finally, the right-hand sides of (A6a) and (A6b) are expanded about time t^n to obtain

$$\frac{Y^{n+1}(x)-Y^n(x)}{\delta t}=\frac{DY}{Dt}+\frac{\delta t}{2}\frac{D^2Y}{Dt^2}+\cdots,$$

where the material derivative is used because differentiation is along a particle path. The second-order time derivative is found in terms of the spatial derivatives by differentiating (A6a) and (A6b) with respect to time and substituting

$$\frac{DY}{Dt}=\nu Y_{xx}+g$$

in the resulting equations [18].

REFERENCES

1. J. J. Walton, M. C. MacCracken, and S. J. Ghan, *J. Geophys. Res.* **93**, 8339 (1988).
2. S. J. Ghan, M. C. MacCracken, and J. J. Walton, *J. Geophys. Res.* **93**, 8315 (1988).
3. F. H. Harlow, "The Particle-in-cell Computing Method for Fluid Dynamics," in *Fundamental Methods in Hydrodynamics*, edited by B. Alder, S. Fernbach, and M. Rotenberg (Academic Press, New York, 1964).
4. F. Brunel, J. N. Leboeuf, T. Tajima, and J. M. Dawson, *J. Comput. Phys.* **43**, 268 (1981).
5. J. U. Brackbill and H. M. Ruppel, *J. Comput. Phys.* **65**, 314 (1986).
6. J. U. Brackbill and J. J. Monaghan (Eds.), *Particle Methods in Fluid Dynamics and Plasma Physics* (North-Holland, Amsterdam, 1987).
7. A. J. Chorin, *J. Fluid Mech.* **57**, 785 (1973).
8. W. T. Ashurst and P. K. Barr, *Combust. Sci. Technol.* **34**, 227 (1983).
9. A. F. Ghoniem and A. Heidarinejad, *Combust. Sci. Technol.* **72**, 79 (1990).
10. R. D. Reitz, *SIAM J. Sci. Stat. Comput.* **2**, 95 (1981).
11. Y. B. Zeldovitch, G. I. Barenblatt, V. B. Librovich, and G. M. Makhviladze, *The Mathematical Theory of Combustion and Explosions* (Consultants Bureau, New York, 1985).
12. J. J. Monaghan, *Comput. Phys. Rep.* **3**, 72 (1985).
13. J. U. Brackbill, D. B. Kothe, and H. M. Ruppel, *Comput. Phys. Commun.* **48**, 25 (1988).
14. B. Larrouturou, *SIAM J. Sci. Stat. Comput.* **10**, 742 (1989).
15. G. Strang, *Introduction to Applied Mathematics* (Wellesley-Cambridge Press, Cambridge, MA, 1986).
16. D. Burgess, J. U. Brackbill, and D. Sulsky, *J. Comput. Phys.* **103**, 1 (1992).
17. P. J. Roache, *Computational Fluid Dynamics* (Hermosa, Albuquerque, NM, 1982).
18. C. W. Hirt, *J. Comput. Phys.* **2**, 339 (1968).
19. A. A. Amsden, P. J. O'Rourke, and T. D. Butler, LA-11560-MS, Alamos National Laboratory, 1989 (unpublished).