Journal of Computational Physics **173,** 187–207 (2001)
doi:10.1006/jcph.2001.6871, available online at http://www.idealibrary.com on IDEAL®

An Effective Particle Tracing Scheme on Structured/Unstructured Grids in Hybrid Finite Volume/PDF Monte Carlo Methods

Genong Li and Michael F. Modest

Mechanical Engineering Department, The Pennsylvania State University, University Park, Pennsylvania 16802
E-mail: mfm6@psu.edu

Received July 18, 2000; revised June 19, 2001

To date, PDF/Monte Carlo simulations, either in stand-alone codes or in hybrid finite volume/PDF Monte Carlo programs, appear to have been mostly carried out with cells of similar size. In many situations, such as capturing sharp gradients in a flow field, fine grids or unstructured solution-adaptive grids must be used in the finite volume code, resulting in a cell system with large variations in cell size. Such grids present a challenge for a combined PDF/Monte Carlo code. In this paper, a new particle tracing scheme is proposed, in which we introduce the concept of variable time steps. Using locally adaptive time steps in the integration of particle equations, the particles' parameters are updated more frequently in regions of strong gradients than in those of flat gradients, which greatly improves the time efficiency of particle tracing. To reduce statistical errors, a particle splitting and combination procedure is also used. The new scheme allows the hybrid finite volume/PDF Monte Carlo code to use any grid that is constructed in the finite volume code. This relaxation of restrictions on the grid makes it possible to couple PDF/Monte Carlo methods to all popular commercial CFD codes and, consequently, extend existing CFD codes' capability to simulate turbulent reactive flow in a more accurate way. The numerical performance of the new particle tracing scheme and the solution procedure are illustrated by considering a turbulent heat transfer problem in a parallel channel and a turbulent diffusion combustion problem in a cylindrical combustor. © 2001 Academic Press

Key Words: composition PDF method; Monte Carlo simulation; particle tracing; unstructured mesh; variance reduction; turbulent heat transfer; chemically reacting flow.

1. INTRODUCTION

Probability density function (PDF)/Monte Carlo methods have been increasingly used in turbulent reactive flow calculations [1–3], primarily because the PDF method is able to treat

0021-9991/01 $35.00

some of the nonlinear processes exactly. For example, in the velocity PDF method, the gradient diffusion assumption for turbulent transport is not necessary; in the composition PDF method, complete treatment of arbitrarily complex finite-rate chemical reactions is possible; in the velocity–composition PDF method, besides the aforementioned two processes, the interaction between turbulence and chemical reactions can be considered with ease. In a class of hybrid numerical methods considered in this paper, the flow field (including velocities, pressure, turbulent kinetic energy, and the dissipation rate of turbulent kinetic energy) is solved by finite volume techniques, while scalars such as mass fraction of species and temperature are solved by the composition PDF method. The hybrid methods combine the advantages of the finite difference method's efficiency and robustness in solving the flow field with the PDF method's exactness in dealing with chemical reactions. They are, therefore, increasingly used in turbulent reactive flow calculations.

The PDF transport equation can be derived from the conservation laws. However, the high degree of multidimensionality of this equation makes conventional finite difference or finite element methods inefficient for its solution. Therefore, a PDF/Monte Carlo approach has been developed over the past two decades, primarily by Pope and co-workers [1]. In this method, the PDF is represented by a large number of particles distributed in the computational domain. Each particle carries flow quantities with it, which evolve according to stochastic differential equations. Field means of the flow at any point are then calculated by taking weighted ensemble averages over values of nearby particles.

During the past decade, significant progress has been made in the development of the PDF method, and its application to various complicated flows has been quite successful. In some implementations, the method was coupled with finite volume codes [4]; in others it provided a complete solution on its own [5]. The method has been used from solving homogeneous shear flows [6] to handling inhomogeneous wall-bounded flows [7–9]. Along with the model development, the numerical methodology has been improved as well. Haworth and Pope [10] derived a high-order scheme for the solution of stochastic differential equations; Welton and Pope [11] introduced the "smoothed particle hydrodynamics" method for calculating mean fields; Delarue and Pope [12] included the calculation of the pressure field within the Monte Carlo algorithm; and recently Xu and Pope [13] and Muradoglu and Pope [14] identified and analyzed different numerical errors that occur in PDF/Monte Carlo methods.

Still, there are several important remaining weaknesses in the method's numerical methodology. The conventional particle tracing scheme of the PDF/Monte Carlo methods does not permit the use of cell systems with large variations in size in an economical way, even though this kind of cell system is essential in many situations, such as capture of sharp gradients in the flow fields. Conventional methods customarily use a constant time step for the integration of particle evolution equations, which tends to waste computer time, since, to capture sharp gradients, very thin cells need to be used in such regions. The time step would then be dictated by the small cells, leading to an extremely small value, which would have particles in large cells move only a short distance in a single time step compared to their cell size. Another complication of using cell systems with large cell-to-cell volume disparities is the need to control the number of particles over cells. If equal mass particles are used, small cells will contain too few particles and large cells will hold too many. In variable-density flows, particle populations over cells become unbalanced even at cell systems with equal cell volumes. This imbalance makes the particle tracing inefficient because tracing more particles than necessary in a cell wastes computer time, while tracing too few leads to large statistical errors (since the error scales as the inverse of square root of the number of particles in a cell). This imbalance problem is particularly severe in solving problems in cylindrical coordinates, in which cells near the axis tend to suffer from low particle counts because of their small volume while outside cells tend to hold too many particles. In many such flows, sharp gradients of the flow field occur near the axis, requiring cells to be smaller still, making things even worse. The imbalance of particle counts has been recognized as a problem in other types of Monte Carlo simulations as well [15, 16]. The need to control particle number density has been realized for a while [1], but this issue has been given little attention in the literature. To date, only a few attempts have been made [17, 18], using the idea of particle splitting and combination. While particle splitting and combination process are known to be very effective, possible side effects, such as degradation of the stochastic system, appear to have been overlooked.

In this paper, we address the shortcoming of the use of a constant time step. Instead of using a single constant time step in the integration of all particle equations for each particle, we define a global time step, which is divided locally into several adaptive substeps governed by the local flow field. The particle properties are updated at the end of each partial time step. The mean fields are obtained by taking weighted ensemble averages of local particle properties. To reduce statistical error and to further improve computational efficiency, a particle splitting and combining procedure has also been used. The possible side effects of splitting and combining on the stochastic system and the requirements of preserving mean quantities, such as mass, scalars, or higher moments of scalars, are pointed out.

In the following, we first review composition PDF/Monte Carlo methods and the conventional particle tracing scheme, followed by a detailed description of the new particle tracing scheme. Finally, two test problems, a turbulent heat transfer problem and a turbulent reacting flow problem, are solved to demonstrate the performance of the method in structured and unstructured grids, respectively.

2. COMPOSITION PDF METHODS AND THE CONVENTIONAL PARTICLE TRACING SCHEME

For variable-density flows, it is advantageous to work with density-weighted (Favre-averaged) quantities. Let $\underline{\Phi}$ represent physical-space composition variables (temperature or enthalpy and species concentration). The transport equation of the Favre composition PDF $\tilde{f}_{\underline{\Phi}}$ can be derived by using the transport equations for $\underline{\Phi}$ (i.e., transport equations for temperature or enthalpy and concentrations). The resulting PDF is generally solved by Monte Carlo methods. In such a PDF/Monte Carlo simulation, the PDF is represented by a large number of particles. Each particle can be interpreted as an independent realization of the flow or as a delta function discretization of the PDF. Every particle evolves according to a set of stochastic differential equations. Pope [19] has shown that there is a one-to-one correspondence between particle evolution equations, the modeled PDF equation, and the modeled Eulerian governing equations for the field means. In the composition PDF method, location, $\underline{x}$, and scalar quantities, $\underline{\Phi}$, of a stochastic particle evolve according to the following stochastic differential equations [18]:

$$\underline{x}^{\star}(t+\delta t)=\underline{x}^{\star}(t)+[\underline{\tilde{u}}+\nabla\Gamma_T/\langle\rho\rangle]\delta t+[2\delta t\Gamma_T/\langle\rho\rangle]^{1/2}\underline{\eta} \tag{1}$$

$$\underline{\Phi}^{\star}(t+\delta t)=\underline{\Phi}^{\star}(t)+\underline{S}^{\star}\delta t-\frac{1}{2}C_{\Phi}(\underline{\Phi}^{\star}-\underline{\tilde{\Phi}})\tau^{-1}\delta t. \tag{2}$$

Here the variables with asterisks refer to the location or scalar values of a Lagrangian particle, variables within an angle brace are conventional means, and variables with a tilde are Favre mean; $\tilde{\underline{u}}$ is the velocity vector, and $\Gamma_T = C_u\langle\rho\rangle\sigma_T^{-1}k^2/\varepsilon$ is the turbulent diffusivity, where k, ε, σ_T, C_u are turbulent kinetic energy, dissipation rate of turbulent kinetic energy, turbulent Schmidt or Prandtl number, and a model constant, respectively. The term $\underline{\eta}$ is a standardized Gaussian random vector (components with zero mean and unity variance) and $\tau = k/\varepsilon$ is the turbulent time scale. In Eq. (2), Dopazo's [20] scalar mixing model has been used, where C_Φ is another model constant. Finally, $\underline{S}^\star$ is a source term that results from chemical reactions.

The above set of particle equations implies the transport equation of a corresponding Eulerian composition PDF $\tilde{g}(\underline{\Phi_\alpha}; \underline{x}, t)$ [1],

$$\frac{\partial[\langle\rho\rangle\tilde{g}]}{\partial t} + \frac{\partial[\langle\rho\rangle\tilde{u}_i\tilde{g}]}{\partial x_i} + \frac{\partial[\langle\rho\rangle S_\alpha\tilde{g}]}{\partial\Psi_\alpha} = \frac{\partial}{\partial x_i}\left[\Gamma_T\frac{\partial\tilde{g}}{\partial x_i}\right] + \frac{1}{2}C_\Phi\omega\frac{\partial}{\partial\Psi_\alpha}[\langle\rho\rangle\tilde{g}(\Psi_\alpha - \tilde{\Phi}_\alpha)], \tag{3}$$

where $\tilde{g}$ is the modeled Favre composition PDF, Ψ is the sample space of the composition variable Φ, and i and α are summation indices in physical space and composition space, respectively. When compared to the exact transport equation, it is seen that the two terms on the right-hand side of the equation are contributions that result from conditional expectations of velocity fluctuation and molecular mixing, respectively.

Multiplying the PDF transport equation by the composition space variable Ψ_α, followed by integration over the entire $\underline{\Psi}$ space, one obtains the governing equation for the scalar mean,

$$\frac{\partial[\langle\rho\rangle\tilde{\Phi}_\alpha]}{\partial t} + \frac{\partial[\langle\rho\rangle\tilde{u}_i\tilde{\Phi}_\alpha]}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\Gamma_T\frac{\partial\tilde{\Phi}_\alpha}{\partial x_i}\right] + \langle\rho\rangle\tilde{S}_\alpha. \tag{4}$$

In PDF/Monte Carlo simulations, instead of solving the partial differential equation (4) directly, as is done in conventional moment methods, the particles are traced according to Eqs. (1) and (2). The means of the composition field are then deduced statistically from the properties of the particles.

In hybrid finite volume/PDF Monte Carlo methods, some Eulerian quantities, such as the velocity field, pressure, turbulent kinetic energy, and dissipation rate of turbulent kinetic energy, are computed by a finite volume code. This information is then transferred to the Monte Carlo code, which calculates the scalar fields and, in turn, supplies updated values for density to the finite volume code. An important feature of the hybrid methods is that some field variables may be solved independently in both finite volume and Monte Carlo codes. The issue of possible inconsistency has been addressed by Muradoglu *et al.* [14]. The present solution procedure is completely consistent since no duplicate fields are computed.

The Monte Carlo code involves integration of the set of particle equations and computation of field means from particle properties. The time step for integration is generally chosen to satisfy an extended Courant–Friedrichs–Lewy (CFL) condition [13]. For the two-dimensional case, the allowable maximum time step for particles in cell i is

$$\Delta t_i = \min\left\{\frac{k}{\varepsilon}, N_{\text{cfl}}\frac{\Delta x}{|\tilde{u}|}, N_{\text{cfl}}\frac{\Delta y}{|\tilde{v}|}\right\}_i \quad i = 1, \ldots, n, \tag{5}$$

where n is the total number of cells; Δx and Δy are the control volume sizes in structured grids or the maximum span of cell i in the x- and y-directions in unstructured grids; N_{cfl} is called the particle CFL number (usually taken to be less than 1); k/ε represents the turbulence time scale; and $\Delta x/|\tilde{u}|$ and $\Delta y/|\tilde{v}|$ represent the time scale of a particle passing through a cell. The time step used in the integration is then

$$\Delta t = \min\{\Delta t_i, i = 1, \ldots, n\}, \tag{6}$$

i.e., the minimum of the values that are determined by Eq. (5) over all n cells.

Another issue in the PDF/Monte Carlo methods is the estimation of means. Estimates of mean quantities from particle values are needed to evaluate Favre means, which also appear as coefficients in the particle evolution equations. Although a grid-free method, called the "smoothed particle hydrodynamics" method, has been proposed by Welton and Pope [11], its computational cost is prohibitive for two- and three-dimensional problems. To estimate field means, generally a cloud-in-cell method is used instead, in which a grid system sits on top of the computational domain and the means are approximated by averages of values of the particles in a nearby region. In a finite volume code, the computational domain is already divided into many control volumes and variables are stored at nodes, the center of the control volume. Generally, the same nodal system is also used in the Monte Carlo code. The field means at nodes are calculated from all particles in a region (dependence region) bounded by the surrounding nodes. The control volume, nodal system, the dependence region of any node and a particle's influence region in the computation of field means in both the structured and unstructured grids are shown in Fig. 1. The field means at nodal point V can be calculated from

$$\tilde{\Phi}_v = \frac{1}{m_v}\sum_{p\in C} b_{pv} m_p^{\star} \Phi_p^{\star}; \quad m_v = \sum_{p\in C} b_{pv} m_p^{\star}, \tag{7}$$

where C is a set of particles that fall into the dependence region of point V; $\Phi_p^{\star}$ are the scalar values of the pth particle; $m_p^{\star}$ is the normalized mass that the particle carries (in the conventional scheme, every particle carries the same amount of mass and $m_p^{\star} = 1.0$); and b_{pv} is the weight of the particle's contribution to point V and is determined by a bilinear basis function (see Appendix).

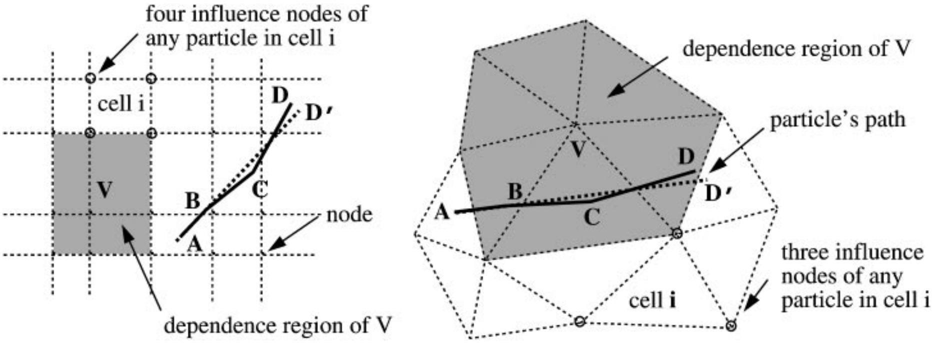
four influence nodes of any particle in cell i
cell i
V
D
D'
B
C
A
node
dependence region of V
dependence region of V
V
particle's path
D
A
B
C
D'
three influence nodes of any particle in cell i
cell i

FIG. 1. The nodal system used in the Hybrid FV/PDF method and the illustration of adaptive time step splitting.

3. NEW PARTICLE TRACING SCHEME

It is apparent that conventional particle tracing schemes, when a cell system with large cell-to-cell volume disparities is used, either lose computational efficiency if the time step is small or tend to smoothen out the flow field if the time step is large. We will now introduce a new scheme, in which the time step for each particle is divided adaptively into several substeps depending on the local flow field at the particle's location. And, to gain better control of the number of particles in each cell, a particle splitting and combination procedure is designed.

3.1. Adaptive Time Step Splitting and Mean Estimation Method

If a single time step is used for all particles, and particle properties are tallied only at the end of each time step, some particles may cross several small cells without any contribution to the cells they pass through. The main idea in the procedure of adaptive time step splitting is to ensure that a particle in the flow field does not change its physical position too much or too little in one move in comparison with the cell size in which it resides. The global time step for each individual particle (i.e., the time step between updates of the global velocity and scalar fields) is divided into several substeps. Thus, from time level k to level $k+1$, every particle experiences a series of moves, for example, $A \rightarrow B \rightarrow C \rightarrow D$ as illustrated in Fig. 1 instead of just a single move, $A \rightarrow D'$, in the conventional scheme. The final positions of D and D' might be slightly different because of the different frequencies to update the mean fields and the different Wiener terms in the particle equation. For a particle p in cell i after the $(s-1)$th submove at time level k, the substep for the sth move is computed as

$$\Delta t_p^{k,s} = \min\left\{\frac{k}{\varepsilon}, N_{\text{cfl}}\frac{\Delta x}{|\tilde{u}|}, N_{\text{cfl}}\frac{\Delta y}{|\tilde{v}|}\right\}_i. \tag{8}$$

Care is taken that the last substep brings all particles to the next global level $k+1$. After the partial time step is set, the set of SDEs for particle p needs to be integrated. Most PDF/particle methods to date have used a first-order-accurate scheme in time. Recently, a second-order-accurate predictor/corrector (P/C) scheme has been introduced by Pope and others [11, 14, 21] and is used here. The predictor step determines the particle's approximate location after the integration, with the explicit form

$$\hat{\underline{x}}^{\star k,s+1} = \underline{x}^{\star k,s} + [(\tilde{\underline{u}} + \nabla\Gamma_T/\langle\rho\rangle]_{\underline{x}^{\star k,s}}\Delta t_p^{k,s} + \left[2\Delta t_p^{k,s}\Gamma_T/\langle\rho\rangle\right]_{\underline{x}^{\star k,s}}^{1/2}\underline{\eta}; \tag{9}$$

superscripts k, s, and $k, s+1$ denote values at time $t^{k,s}$ and $t^{k,s+1}(=t^{k,s}+\Delta t_p^{k,s})$, respectively. All mean quantities appearing in the coefficients are evaluated at the particle's initial position $x^{\star k,s}$ as indicated by the subscript. In our calculations, all mean values are stored at the vertices of the rectangular (structured) or the triangular (unstructured) mesh. The field means at arbitrary particle position O are linearly interpolated from the vertex quantities of its host cell: if ξ denotes any mean quantity stored at vertices, the corresponding value at an arbitrary point is

$$\xi_O = \xi_{V_1}b_{OV_1} + \xi_{V_2}b_{OV_2} + \xi_{V_3}b_{OV_3} + \xi_{V_4}b_{OV_4}, \quad \text{structured grids} \tag{10}$$

$$\xi_O = \xi_{V_1}b_{OV_1} + \xi_{V_2}b_{OV_2} + \xi_{V_3}b_{OV_3}, \quad \text{unstructured grids,} \tag{11}$$

where ξ_{V_1}, ξ_{V_2}, ξ_{V_3}, and ξ_{V_4} are the mean quantities at the vertices, and the weight factors b are calculated from the linear basis function.

In the corrector step, all coefficients involving the field means are evaluated as the averages of their mean values at the initial and at the predicted location. The corrector step has the form

$$\begin{aligned}\underline{x}^{\star k,s+1} &= \underline{x}^{\star k,s} + \frac{1}{2}\left\{\left[\underline{\tilde{u}} + \frac{1}{2}\nabla\Gamma_T/\langle\rho\rangle\right]_{\underline{x}^{\star k,s}} + \left[\underline{\tilde{u}} + \frac{1}{2}\nabla\Gamma_T/\langle\rho\rangle\right]_{\underline{\hat{x}}^{\star k,s+1}}\right\}\Delta t_p^{k,s} \\ &\quad + \frac{1}{\sqrt{2}}\left\{\left[\Delta t_p^{k,s}\Gamma_T/\langle\rho\rangle\right]_{\underline{x}^{\star k,s}}^{1/2} + \left[\Delta t_p^{k,s}\Gamma_T/\langle\rho\rangle\right]_{\underline{\hat{x}}^{\star k,s+1}}^{1/2}\right\}\underline{\eta} \end{aligned} \tag{12}$$

$$\underline{\Phi}^{\star k,s+1} = \underline{\Phi}^{\star k,s} + \underline{S}^{\star k,s}\Delta t_p^{k,s} - \left(\underline{\Phi}^{\star k,s} - \frac{\underline{\tilde{\Phi}}_{x^{*k,s}} + \underline{\tilde{\Phi}}_{\hat{x}^{*k,s+1}}}{2}\right)\frac{C_\Phi \Delta t_p^{k,s}}{\tau_{x^{*k,s}} + \tau_{\hat{x}^{*k,s+1}}}. \tag{13}$$

Note that the same value of $\underline{\eta}$ is used for both predictor and corrector steps, but that different and independent values of $\underline{\eta}$ are used for each subtime step. It has been verified that this P/C scheme is equivalent to the Runge–Kutta form of Milischtein's general second-order weak scheme for stochastic systems [11].

Once the particle's new location is determined, its properties are updated. The field means are updated at the end of each global time step using, both, the particle's properties at the final position as well as those at intermediate positions. The contribution of each submove is weighted by its time fraction, so that Eq. (7) changes to

$$\underline{\tilde{\Phi}}_v^{k+1} = \frac{1}{m_v}\sum_{s=1}^{S}\sum_{p\in C} b_{pv}^{k,s} m_p^{\star k}\frac{\Delta t_p^{k,s}}{\Delta t}\underline{\Phi}_p^{\star k,s}; \quad m_v = \sum_{s=1}^{S}\sum_{p\in C} b_{pv}^{k,s} m_p^{\star k}\frac{\Delta t_p^{k,s}}{\Delta t}, \tag{14}$$

where S is the number of substeps for particle p between time levels k and $k+1$; C is a set of particles, which are found in the influence region of node V; $\underline{\Phi}_p^{\star k,s}$ are the scalar values of the pth particle at the sth substep. Thus, a particle traveling through space interacts with all cells it passes through rather than only the one in which it ends up at the end of the time step. In regions with strong gradients, particles update their properties more frequently, and particles always reflect the local flow field information. With this procedure, there is basically no upper limit in choosing a global time step for statistically stationary problems, and the magnitude of the global time step only affects the frequency of updates of the mean fields.

Strictly speaking, the mean estimation method of Eq. (14) is only valid for statistically stationary problems, since the particle's properties at some intermediate time points other than at $(t + \Delta t)$ are used as well. But for nonstationary problems, if the global time step is much less than the characteristic time scale of the flow field, this average method is still valid. If the global time step is not that small, the field means can be calculated by only tallying particles at the end of one global time step by using Eq. (7).

3.2. Particle Splitting and Combination Procedure

In the particle splitting and combination procedure, the number of particles in each cell is checked after every global time step. If the number in one cell is less than a prescribed minimum, a particle splitting process is initiated, and if the number of particles exceeds the prescribed maximum, a particle combination process is carried out. During this process, the particles' distribution and number may change before and after the particle splitting and combination procedure, but the two sets of particles should be equivalent in a statistical sense, and field means (including mass, scalars, and their higher-order moments) extracted from the particle field should be preserved.

The simple splitting strategy is this: if the number of particles in the cell is N less than the prescribed minimum, the first N high-weight particles are split; each of the N newly created particles carries the same scalar properties as its parent, but only half the mass; both are placed at the same location as their parent, but evolve independently thereafter because of the random paths generated by the Wiener process. This approach greatly reduces statistical error and improves computational efficiency. During this process, the weight of a particle may be reduced significantly after several splitting events and its contribution to the problem may become minimal. To avoid wasteful tracing of such small particles, a lower limit of mass is set below which particle splitting is not applied. It is clear that the above procedure preserves the mean quantities. However, the splitting process can change the evolution of the stochastic system afterward, since splitting produces several identical particles in a cell and a high degree of data dependency may lead to biased results. This would be the case, for example, if the particle-interaction mixing model [1] were used in the particle evolution process. If every particle evolves independent of others, this data dependency will not cause any statistical degradation. In our current numerical implementation, Dopazo's mixing model is used, in which particles only interact with the mean fields and, therefore, splitting does not introduce any side effects.

If the number of particles in a cell is larger than a prescribed maximum, a combination process is employed. Subramaniam and Haworth proposed a simple scheme [18], in which low mass particles is annihilated and the mass associated with these particles is distributed evenly among remaining particles in that cell. Although the method preserves the mass, it preserves neither mean scalars nor their higher-order moments. Actually, there is a simpler scheme that can preserve mean quantities. If two particles ($\underline{x}_1^\star$, $m_1^\star$, $\underline{\Phi}_1^\star$ and $\underline{x}_2^\star$, $m_2^\star$, $\underline{\Phi}_2^\star$) in a cell are combined into one, the new particle is put at the mass-weighted average location $(m_1^\star \underline{x}_1^\star + m_2^\star \underline{x}_2^\star)/(m_1^\star + m_2^\star)$ and takes the properties of particle 1 with probability $m_1^\star/(m_1^\star + m_2^\star)$ and those of particle 2 with probability $m_2^\star/(m_1^\star + m_2^\star)$. This method is used in this study. In our actual simulations, all particles in each cell are sorted in descending order of the mass. If the number of particles in a cell is larger than the prescribed maximum by N particles, the low-ranked $2N$ particles are combined into N pairs to obtain the desired number.

The splitting and combining processes are carried out at the end of each global time step, followed by renumbering of all particles in the flow field, before the time marching process continues.

3.3. Particle Tracking in Triangular Meshes

To extract field means from particle properties by either Eq. (7) or Eq. (14), it is first necessary to identify the host cell where each particle is located. It is a trivial task to determine the host cell in rectangular coordinates, but the search becomes complicated when the computational cell is no longer rectangular in the physical domain. A number of search strategies are possible [22]. The use of successive neighbor searches turns out to be the most suitable method, considering the fact that particles most likely stay in the same cell or simply move to one of their neighboring cells after a single time step, which is

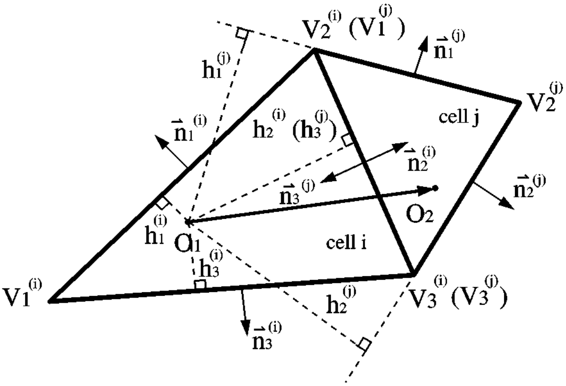

FIG. 2. Illustration of particle tracing in triangular cells.

especially true in the calculation here because of the use of the adaptive time step splitting technique.

Assume a particle resides in cell i with position O_1 originally and moves to O_2 after one time step. To identify its new host cell, three quantities are calculated,

$$\frac{\overrightarrow{O_1O_2} \cdot \vec{n}_1^{(i)}}{h_1^{(i)}}, \quad \frac{\overrightarrow{O_1O_2} \cdot \vec{n}_2^{(i)}}{h_2^{(i)}}, \quad \frac{\overrightarrow{O_1O_2} \cdot \vec{n}_3^{(i)}}{h_3^{(i)}},$$

where $\vec{n}_1^{(i)}$, $\vec{n}_2^{(i)}$, and $\vec{n}_3^{(i)}$ are the three outward normals of the three surfaces making up cell i; $h_1^{(i)}$, $h_2^{(i)}$, and $h_3^{(i)}$ are the shortest distances from point O_1 to these surfaces (see Fig. 2), and the superscript (i) is used to denote the quantities' association with cell i. If all three quantities are less than unity, the particle is still in cell i; if their maximum is greater than unity, this implies that the particle leaves the cell through the surface on which the maximum is computed. For example, in the case shown in Fig. 2, the second of the above quantities is the largest and the particle leaves the cell through surface 2. The connectivity information of the mesh gives the cell number next to that surface, in this case, j. The cell pointer for the particle is updated to that neighboring cell, and the above three quantities are recalculated with respect to cell j. This process is repeated until the new host cell is found, i.e., when all three values are less then unity.

We call this strategy an element-to-element search. In Subramaniam and Haworth's paper [18], a face-to-face search strategy is used, in which the time fraction spent in each cell is calculated. Such information is needed if a cell-center-based PDF code is used (the field means are stored at the centroid of the cells) and one wants to distribute a particle's contribution to all the cells it passes through according to its residence time in each cell. In the vertex-based scheme with time step splitting procedure that is used in our current method (the field means are stored at vertices of the cells), the element-to-element search scheme is more efficient.

3.4. Numerical Performance Prediction

The amount of computational work can be estimated for both the conventional scheme and the new particle tracing scheme. In the conventional scheme, the amount of computational work (W_{conv}) is proportional to the number of total particles used in the simulation (N), the number of steps to get to the final result (I_1), and the computational work per particle per step (w_p):

$$W_{\text{conv}} \simeq N I_1 w_p; \quad I_1 = \frac{t}{\Delta t_1}. \tag{15}$$

Here t is the total computational time period, which usually requires several times ($M = 2$ to 5, say) the residence time (t_s), i.e., the time that an average particle needs to travel through the entire computational domain. The residence time may be estimated as $t_s = l/\tilde{u}_{av}$, where l is the length of the computational domain and $\tilde{u}_{av}$ is the average streamwise velocity. Therefore, the estimate of computational work is

$$W_{\text{conv}} \simeq N I_1 w_p = M N \frac{t_s}{\Delta t_1} w_p. \tag{16}$$

With the new particle tracing scheme, some particles may make several moves in one global time step, and the particle splitting and combining procedure requires an extra amount of cpu time (w_a) per particle per global step. If L is used to denote the average number of moves for a particle to move in one global step, the amount of computational work can be approximated as

$$W_{\text{new}} \simeq N I_2 (L w_p + w_a) = M N \frac{t_s}{\Delta t_2} (L w_p + w_a). \tag{17}$$

To evaluate the efficiency of the new scheme, we consider the ratio of the two and obtain

$$\frac{W_{\text{new}}}{W_{\text{conv}}} = \left(L + \frac{w_a}{w_p}\right) \frac{\Delta t_1}{\Delta t_2}. \tag{18}$$

In the new particle tracing scheme, L is a function of Δt_2. It increases as Δt_2 increases, but its rate of increase is always slower. As a result, the work ratio is always less than unity for nonequally spaced grids.

In PDF/Monte Carlo methods, the numerical results are subject to three kinds of numerical errors: statistical error, discretization error, and bias [21]. Statistical error is unique to Monte Carlo simulations and is often dominant. The statistical error (Σ) of a field mean is measured by its standard deviation, which is proportional to the reciprocal of the square root of the number of particles in a cell, $\Sigma \propto 1/\sqrt{N_c}$. In the conventional particle tracing scheme the largest statistical error occurs in the smallest cell in which the number of particle is minimum, $N_{c,1} \simeq V_{\text{min}}/V_{\text{total}} N$; in the new scheme, the particle splitting and combination procedure makes the number of particles in each cell roughly equal, or $N_{c,2} \simeq N/n$, where n is the total number of cells in the computational domain. $N_{c,2}$ is usually much larger than $N_{c,1}$ except in the extreme case of using equal size cells, in which both are the same. As a result, statistical error by using the new scheme is smaller than that by using the old scheme if the same number of particles are used.

4. TEST PROBLEMS

Two test problems are used to demonstrate the performance of the new scheme. The first is a turbulent nonreacting heat transfer problem with incompressible flow and, therefore, the scalar solver and flow solver are decoupled. This problem is also used to validate the PDF code since the scalar field can also be solved exactly by traditional finite volume codes. The second problem is a turbulent reacting flow problem where a cylindrical combustor is simulated, in which the two solvers are closely coupled.

4.1. Turbulent Heat Transfer Problem

A simple two-dimensional test problem has been chosen, namely a hot jet flowing through a parallel duct, in which no chemical reaction is considered. Although the strengths of the PDF/Monte Carlo methods are not obvious in this simple problem, it serves well to test the validity and efficiency of the proposed particle tracing scheme.

The geometry and boundary conditions of the problem are shown in Fig. 3. The channel is 5 m long and 1 m high. A stream of hot fluid is injected near the center at high velocity, while cold fluid at low velocity is injected at top and bottom. Therefore, in the central region near the inlet, the temperature decreases rapidly from a maximum in the hot center stream to a lower level in the cold side streams. This sharp gradient in the temperature profile requires fine grids and the use of nonequal size grids makes this a good test case for the new particle tracing scheme.

To obtain the flow field quantities, FLUENT [24] is used as the finite volume code, solving the Navier–Stokes equations in conjunction with the k-ε turbulence model. For this statistically stationary and incompressible flow problem, the energy equation is decoupled from the governing equations for momentum, which allows the energy equation to be solved independently after the velocity field has been obtained first. The energy equation can be written as

$$\frac{\partial[\rho\langle T\rangle]}{\partial t}+\frac{\partial[\rho\langle u_i\rangle\langle T\rangle]}{\partial x_i}=\frac{\partial}{\partial x_i}\left[\Gamma_T\frac{\partial\langle T\rangle}{\partial x_i}\right]. \tag{19}$$

Comparing with Eq. (4), and noting that conventional means are equal to Favre means for incompressible flows, it is apparent that the PDF/Monte Carlo approach can simulate this problem with particle equations (1) and (2), if Φ is taken as temperature. Temperature field can be obtained by the finite volume code as well, which has been done to document the accuracy of our PDF/Monte Carlo scheme.

To resolve the sharp velocity and temperature gradients near the inlet, non-equally-spaced nodes are used in both coordinate directions. Closely spaced gridlines are laid out near the inlet in the streamwise direction and around the center line in the cross-stream direction. A grid system of 20 × 30 was found to give satisfactory results by FLUENT, although

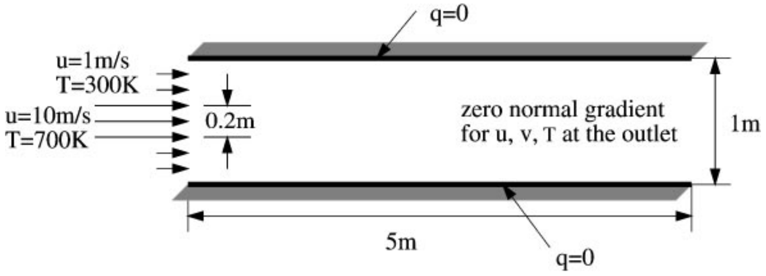

FIG. 3. Geometry of the channel and its boundary conditions.

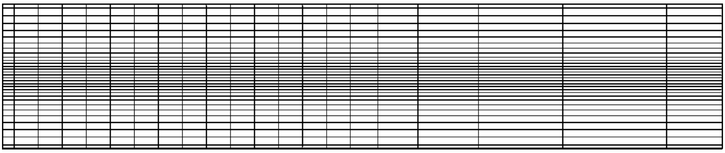

FIG. 4. Grid system used in the FV and the PDF/Monte Carlo codes.

80 × 120 gave grid-independent solutions. Considering that the two results are close and because of the expense of coupling a PDF/Monte Carlo code with so many cells, in our calculation, a 20 × 30 grid system for this problem is used as shown in Fig. 4. In the present grid system the volume of the smallest cell is about 3×10^{-3} m^2, and the largest one is 0.04 m^2, with a ratio of maximum to minimum cell volume of about 13. The time step calculated to meet the CFL condition ($\Delta t \langle u \rangle / \Delta x \leq 0.3$) varies greatly from cell to cell, the allowable time step being only 0.005 s for particles in the center cells near the inlet, but as high as 0.2 s for some cells near the outlet.

Both the conventional and the new particle tracing scheme are used to find the temperature field. The number of particles used in a single simulation is 12,000, and 15 similar simulations are run for a total of 180,000 particles to obtain standard deviations of the statistical mean values. Different sequences of random numbers are used in different simulations to make sure the results are independent of each other. At the beginning of each simulation, 12,000 particles are distributed throughout the computational domain. After the start of the simulation, each particle evolves independently. When a particle hits the wall, it is simply reflected without a change in temperature because of the insulated wall condition. During each time step, a number of particles will leave the computational domain and, at the same time, many new ones are introduced at the inlet to maintain global mass conservation. In this incompressible test problem, the mass density level remains uniform in the calculation, although the particle number density may change as a result of the splitting and combination procedure.

In the conventional scheme, the time step used for integration of the particle equations is dictated by the smallest cell, resulting in a constant time step of 0.005 s for all particles. In the new scheme, a much larger global time step of 0.2 s is used. A statistically stationary temperature profile is obtained after approximately $t = 5$ s, which corresponds to 2.8 times the residence time for both simulations. The temperature profile and its standard deviation at the centerline and at one downstream cross section are shown in Fig. 5. The results from FLUENT are also shown in the figure. The results from both schemes give satisfactory agreement with those of FLUENT considering the relatively small number of particles used in the simulations. The greatest discrepancy between the particle solutions and the FLUENT solution occurs near the walls, which is caused by the different near wall treatments in both approaches.

It takes about 63 μs cpu time on an SGI-O200 (270 MHz, single R10000 processor) to trace each particle for one move. The particle splitting and combining procedure requires 35 μs when averaged to one particle for one global time step. In the new scheme, it takes about 7.1 substeps for an average particle to move one global time step ($\Delta t_2 = 0.2$ s). Thus, for this test problem, and using Eq. (18), the computational work for the new scheme should be about 19% of the old one. In an actual comparison, we found that one entire simulation with the conventional particle tracing scheme required a total of 202 min, while,

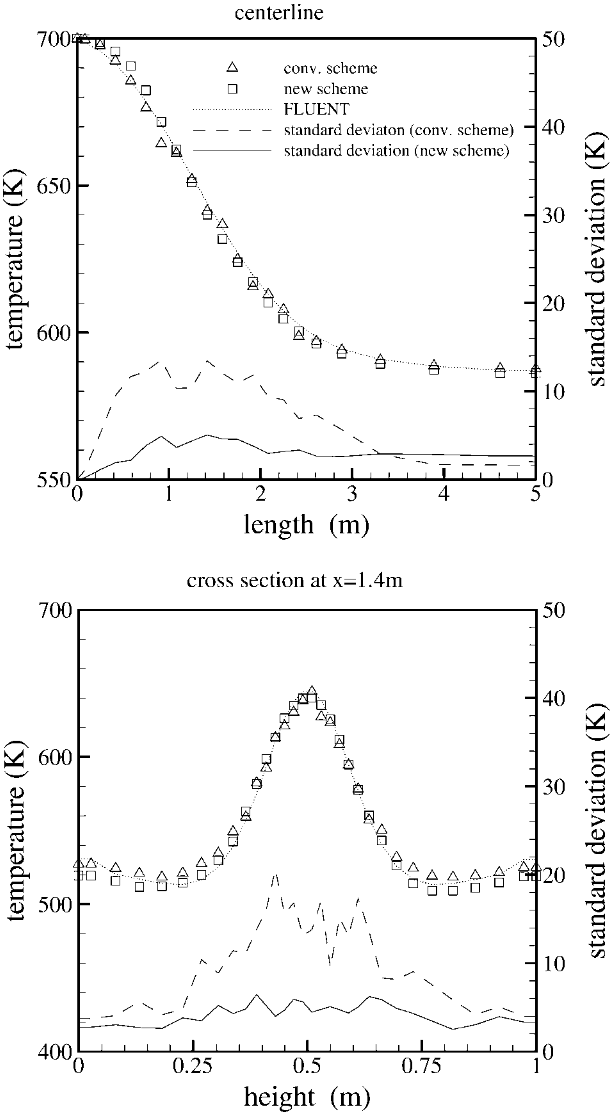

FIG. 5. Computed temperature profile and its standard deviation.

with the new particle tracing scheme, only 38 min were needed, which agrees well with the predictions from Eqs. (16) and (17) (189 min for old scheme and 36 min for the new, respectively).

From Fig. 5 it is also apparent that the standard deviation of the temperature field obtained by the new scheme is much smaller and more uniform over the computational domain. In the standard method, the average number of particles in a cell ranges from 8 in the smallest cell to 100 in the largest cell. However, in the new scheme, the average number of particles in each cell is held around 20, resulting in much better statistics. As a consequence, if a tolerance of statistical error is to be met, one expects to use considerably less particles in the simulation using the new scheme than that using the conventional one. To demonstrate this, the standard deviation of the temperature at one point (1.0 m, 0.5 m), a rather typical position along the centerline, for different numbers of particles, is shown in Fig. 6. For

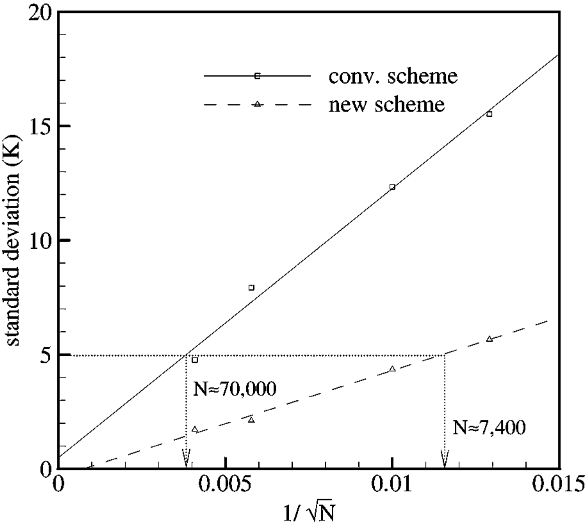

FIG. 6. Standard deviation of temperature at point (1.0 m, 0.5 m) vs the total number of particles. The lines are the linear fits to the data.

example, if a standard deviation below 5K is desired, this requires 70,000 particles in a simulation with the old particle tracing method, while the new method needs only 7400 particles.

One of the strengths of PDF methods is that, once the PDF equation for the scalars is solved, it includes complete information for these scalars, allowing the direct calculation of turbulent moments and other statistics. To get similar moments or statistics from traditional finite volume methods, one has to model and solve many coupled partial differential equations. In this problem, for example, the fluctuations of the temperature field ($\sqrt{\langle T'^2 \rangle}$) can be calculated conveniently, as shown in Fig. 7.

4.2. Combustion Test Problem

Numerical simulation of an axisymmetric combustor is used to further demonstrate the superior performance of the new particle scheme. The geometry of the combustor is shown in Fig. 8. A small nozzle in the center of the combustor introduces methane at 80 m/s. Ambient air enters the combustor coaxially at 0.5 m/s. The overall equivalence ratio is approximately 0.76 (about 28% excess air). The high-speed methane jet initially expands with little interference from the outer wall, and entrains and mixes with the low-speed air.

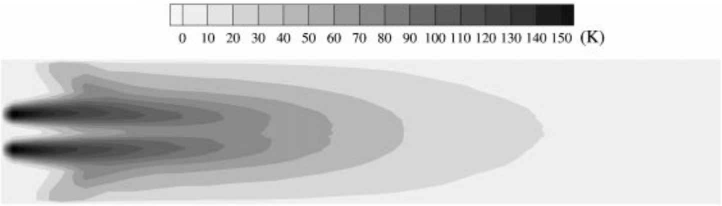

FIG. 7. Temperature fluctuation, $\sqrt{\langle T'^2 \rangle}$, for the turbulent mixing test problem.

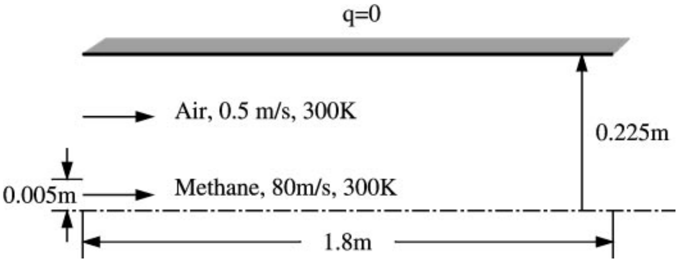

FIG. 8. Geometry of the cylindrical combustor.

The Reynolds number based on the methane jet diameter is approximately 28,000. The two solvers are closely coupled in this case since the PDF solver needs flow field information from the flow solver, and the flow solver, on the other hand, needs the density field computed by the PDF solver. Consequently, the two codes have to run interactively throughout the iterative process. One key issue in the hybrid scheme is the passing of information between the two codes. The PDF/Monte Carlo technique is a statistical method and, hence, always has fluctuations associated with its solutions. If the level of these fluctuations is too high, it may cause divergence or other computational difficulties when information is communicated to the finite volume solver. The application of our adaptive time step splitting and particle splitting and combination keeps statistical fluctuations low and no divergence problem was observed in the finite volume code after the updated density field was fed back from the PDF solver, even for low particle number densities, say 10 particles per cell in average.

To capture the strong gradients in such flows, a locally clustered mesh or even an adaptive mesh has to be used in the finite volume solver. Unstructured grids not only provide greater flexibility in discretizing complex domains but also allow straightforward implementation of adaptive meshing techniques where mesh points may be added or deleted locally [23]. Many finite volume codes, such as FLUENT [24], allow the use of such unstructured meshes. Using this capability of the finite volume solver, an unstructured mesh is employed for the solution of this problem.

Two sets of unstructured meshes, one with 1247 nodes and 2317 cells and the other with 4753 nodes and 9103 cells, were used to demonstrate grid-independence of the finite volume solution. The coarser mesh was employed for the solution of this problem in the hybrid finite volume/PDF MC calculation, which is shown in Fig. 9. Since the diameter of the inner fuel nozzle is very small, many cells have to cluster near the fuel inlet to adequately capture the flow field. As a result, the minimum cell area is only $7.3 \times 10^{-7}\,\mathrm{m}^2$ while the maximum cell area is $1.4 \times 10^{-3}\,\mathrm{m}^2$. In cylindrical coordinates, a cell in the x–r plane actually represents an annulus in three-dimensional space. For this problem, the total volume of the computational domain is about $4.55 \times 10^{-2}\ \mathrm{m}^3$, and the minimum and

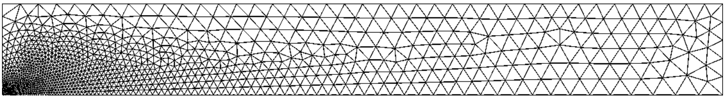

FIG. 9. The unstructured mesh used in the hybrid FV/PDF Monte Carlo method.

maximum cell volumes are 4.0×10^{-10} m^3 and 2.0×10^{-4} m^3, respectively, or

$$\frac{V_{\mathrm{min}}}{V_{\mathrm{total}}} = 8.8 \times 10^{-9}; \quad \frac{V_{\mathrm{max}}}{V_{\mathrm{total}}} = 4.4 \times 10^{-3}; \quad \frac{V_{\mathrm{max}}}{V_{\mathrm{min}}} = 5 \times 10^5. \tag{20}$$

In the traditional PDF/MC method, particles of equal mass are used throughout. If 1,000,000,000 particles were used in the simulation, the largest cell alone would hold 4,400,000 particles while the smallest cell would hold only about 9 particles. Obviously, the traditional method is extremely inefficient in tracing particles in a problem with such unbalanced control volumes.

While solving this kind of problem by traditional methods may still be possible, although very inefficient, with a super computer, it becomes completely impossible when the required time step is considered. A particle entering with the fuel stream takes about 0.0225 s to travel through the entire computational domain, while a particle from the air stream will take about 3.6 s. In conventional methods, a constant time step for all the particles is used, which is constrained by the smallest cell region where the velocity is the largest. If such a time step is used, particles in the large outside cells would move only a tiny distance in each single time step and the simulation would essentially take forever.

Our new particle tracing scheme with adaptive time step splitting and with the particle splitting and combination technique can handle this extreme case without any problems. The number of particles in the computational field is about 46,500 in the simulation, and the particle splitting and combination procedure ensures that the number of particles in each cell remains in the range between 10 and 50. The scalar quantities in this problem include five mass fractions of different species (CH_4, O_2, CO_2, H_2O, and N_2) and the temperature, T. To reduce statistical error, the temperature and species fields in the PDF solver are obtained by averaging their solutions over the previous five time steps. In the entire simulation, a global time step of 0.004 s is used and 1000 global time steps are taken to reach the statistically stationary solution, taking about 50 min on an SGI-O200 (270 MHz, single R10000 processor) to complete the whole simulation.

The conventional way to define residual error in the finite volume method is meaningless in the hybrid FV/PDF MC simulation because the statistical error is generally larger than the truncation error. Therefore, the numerical error for any variable ϕ is defined here as

$$\mathrm{err} = \sqrt{\frac{1}{N_v} \sum_{v=1}^{N_v} \frac{[\langle \phi_v(t + \triangle t)\rangle - \langle \phi_v(t)\rangle]^2}{\langle \phi_v(t)\rangle^2}}. \tag{21}$$

This error never converges to zero, but rather to a value representative of the statistical fluctuation of the solution when statistically stationary state is reached. This level mainly depends on the number of particles in the simulation and on the number of time steps over which the solution is averaged. If temperature is used to monitor the numerical error, a value on the order of 10^{-2} has been reached in the present calculations.

During every global time step, some particles will pass several cells through a series of submoves while others might just take one move depending on their location in the computational domain. During each time step, a number of particles leave the computational domain and, at the same time, a similar number of new ones are introduced at the inlet to maintain global mass conservation. The new particles are released such that the mass flow rate is constant for the fuel and air streams and their ratio is equal to the given equivalence ratio of the fuel. Although PDF/Monte Carlo methods can treat chemical reactions exactly

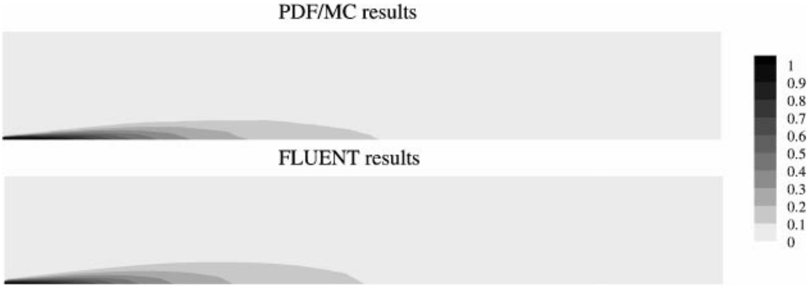

FIG. 10. Distribution of mass fractions of CH_4 for the combustion test problem.

if correct relations for chemical kinetics are used, here, for the purpose of comparison with results obtained from FLUENT, an infinitely fast chemical reaction rate is used for every particle, implying that the global chemical reaction rate is controlled by the mixing process, which is similar to the eddy-dissipation model in FLUENT. In the simulation, although the particle number density may change as a result of the splitting and combination procedure, the particle's mass density level is always proportional to the flow field density. The contours for the mass fraction of CH_4, O_2, CO_2 and temperature are shown in Figs. 10–13 (the mass fraction of H_2O has the same profile as that of CO_2, when scaled properly, and is not shown). The results from FLUENT with the eddy-dissipation combustion model are also shown for comparison, showing that the PDF/MC method and FLUENT give very similar scalar fields. The slight discrepancies are due to several sources. First, FLUENT's results depend on the specific combustion model chosen in the simulation and are always approximate. Although the infinite reaction rate assumption has been used in the PDF/Monte Carlo calculation to create a similar situation as that in FLUENT, the two combustion models are not exactly the same. Second, the simple mixing model used in our PDF/Monte Carlo method also leads to some numerical errors. Finally, PDF/MC results always have fluctuations, and the number of particles used in our simulation is quite low for this complex problem and a much smoother solutions can be expected if more particles were employed. While further improvements can be made in these areas, the main focus in this study is on the development of a solution procedure.

As in the previous test problem, the fluctuation of the scalar fields can be computed within the PDF solver conveniently. Contours for equal fluctuations

$$\sqrt{\widetilde{y_{CH_4}^{\prime 2}}},\quad \sqrt{\widetilde{y_{O_2}^{\prime 2}}},\quad \text{and}\quad \sqrt{\widetilde{T^{\prime 2}}}$$

are shown in Figs. 14–16.

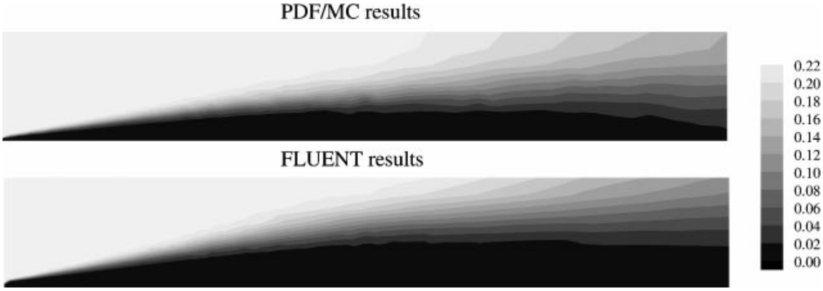

FIG. 11. Distribution of mass fractions of O_2 for the combustion test problem.

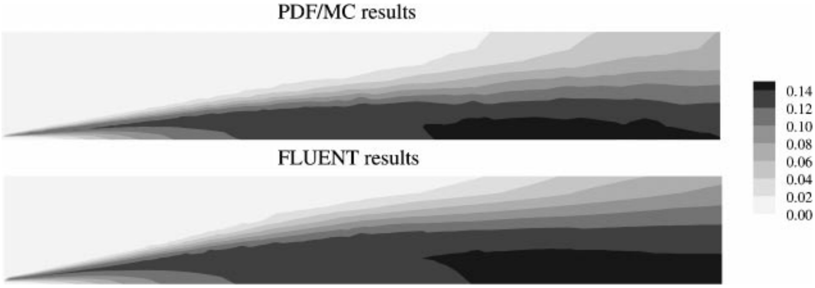

FIG. 12. Distribution of mass fractions of CO_2 for the combustion test problem.

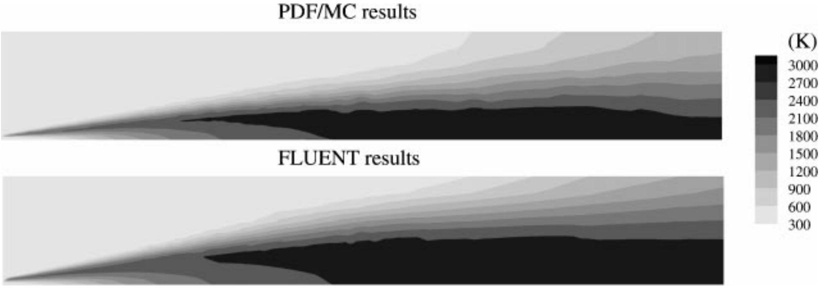

FIG. 13. Distribution of temperatures for the combustion test problem.

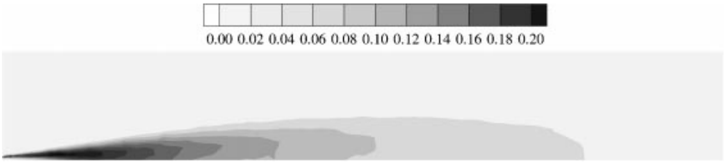

FIG. 14. Distribution of methane concentration fluctuations, $\sqrt{\widetilde{y_{CH_4}'^2}}$, for the combustion test problem.

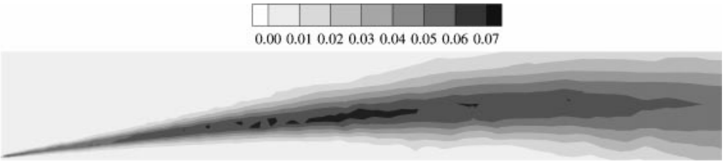

FIG. 15. Distribution of oxygen concentration fluctuations, $\sqrt{\widetilde{y_{O_2}'^2}}$, for the combustion test problem.

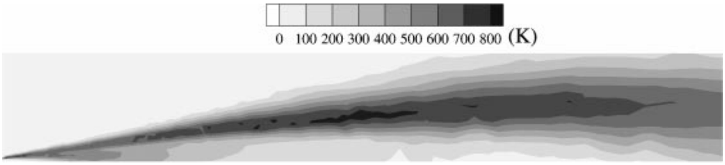

FIG. 16. Distribution of temperature fluctuations, $\sqrt{\widetilde{T'^2}}$, for the combustion test problem.

5. CONCLUSIONS

To capture sharp gradients in a flow field, cell systems with large variations in size have to be used, which makes conventional particle tracing schemes in PDF/Monte Carlo methods very inefficient, resulting in large statistical errors, or even making them incapable of handling this kind of problem. A new particle tracing scheme has been proposed, which saves computer time and reduces statistical error. A locally adaptive time step has been developed for the integration of particle equations, which ensures that each particle changes its location neither too little nor too much in a single move, by comparing the move to the cell size in which the particle resides. The properties of the particles are updated at appropriate frequency during their evolution in physical space. To reduce the statistical error of mean quantities, a particle splitting and combination procedure was also used. With this procedure, the number of particles in each cell is maintained at a prescribed level. As a result, statistical errors of the mean fields are much less than in the conventional scheme, leading to more accurate results with the same number of particles. Numerical experiments were performed on two test problems. The first simple test problem with mildly varying cell sizes showed a reduction in cpu time by a factor of 5, with a concurrent decrease in statistical error by almost one half. The second problem, with extreme cell size variations, common in combustion applications, could not be solved using conventional methods, while a solution was found with ease using our new method.

APPENDIX: LINEAR BASIS FUNCTION *b*

If point *o* is in cell *i* (see Fig. 17), then the linear basis function *b* is calculated in the following way.

- In rectangular meshes,

$$b_{o1} = fg; \quad b_{o2} = (1 - f)g \tag{A.1}$$

$$b_{o3} = (1 - f)(1 - g); \quad b_{o4} = f(1 - g), \tag{A.2}$$

where

$$f = \frac{x_2 - x_o}{x_2 - x_1}, \qquad g = \frac{y_2 - y_o}{y_2 - y_1}.$$

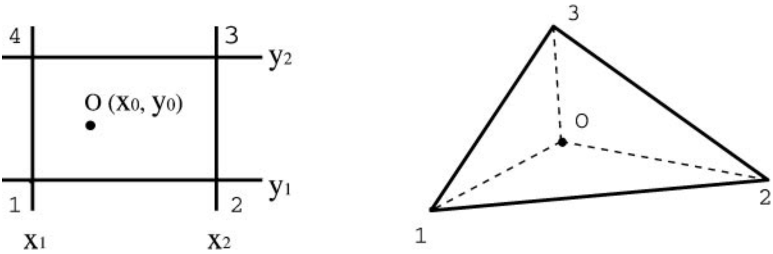

FIG. 17. Sketch of grid.

- In triangular meshes,

$$b_{o1} = \frac{\Delta_{O23}}{\Delta_{123}}, \quad b_{O2} = \frac{\Delta_{O13}}{\Delta_{123}}, \quad b_{O3} = \frac{\Delta_{O12}}{\Delta_{123}}, \qquad [A.3]$$

where Δ_{123} and so forth, are the area of triangle 123 and its subtriangles as shown in Fig. 17.

ACKNOWLEDGMENT

This work was funded by the NSF/Sandia National Laboratories life cycle engineering program through Grant CTS-9732223. The authors thank Dr. L. R. Collins and Dr. D. C. Haworth for many helpful discussions.

REFERENCES

1. S. B. Pope, PDF methods for turbulent reactive flows, *Prog. Energy Combust. Sci.* **11**, 119 (1985).
2. W. Kollmann, The PDF approach to turbulent flow, *Theor. Compt. Fluid Dyn.* **1**, 249 (1990).
3. C. Dopazo, Recent developments in PDF methods, in *Turbulent Reacting Flow*, edited by P. A. Libby and F. A. William (Academic Press, San Diego, 1994), pp. 375–474.
4. M. S. Anand, S. B. Pope, and H. C. Mongia, *A PDF Method for Turbulent Recirculating Flows, Turbulent Reactive Flows*, edited by R. Borghi and S. N. B. Murthy, Lecture Notes in Engineering (Springer-Verlag, New York, 1989), Vol. 40, pp. 672–693.
5. S. B. Pope and Y. L. Chen, The velocity-dissipation probability density function method for turbulent flows, *Phys. Fluids A* **2**(8), 1437 (1990).
6. D. C. Haworth and S. B. Pope, A general Langevin model for turbulent flows, *Phys. Fluids* **29**(2), 387 (1986).
7. S. Mazumder and F. M. Modest, Boundary treatment and an efficient pressure algorithm for internal turbulent flows using the PDF method, *Int. J. Numer. Meth. Fluids* **24**, 215 (1997).
8. T. Dreeben and S. B. Pope, PDF/Monte Carlo simulation of near-wall turbulent flows, *J. Fluid Mech.* **357**, 141 (1998).
9. J. P. Minier and J. Pozorski, Wall-boundary conditions in PDF methods and application to a turbulent channel flow, *Phys. Fluids* **11**(9), 2632 (1999).
10. D. C. Haworth and S. B. Pope, A second-order Monte Carlo method for the solution of the Ito stochastic differential equation, *Stochastic Anal. Appl.* **4**(2), 151 (1986).
11. W. C. Welton and S. B. Pope, Model calculations of compressible turbulent flows using smoothed particle hydrodynamics, *J. Comput. Phys.* **134**, 150 (1997).
12. B. J. Delarue and S. B. Pope, Application of PDF methods to compressible turbulent flows, *Phys. Fluids* **9**(9), 2704 (1997).
13. J. Xu and S. B. Pope, Assessment of numerical accuracy of PDF/Monte Carlo methods for turbulent reacting flows, *J. Comput. Phys.* **152**, 192 (1999).
14. M. Muradoglu, P. Jenny, and S. B. Pope, A consistent hybrid finite-volume/particle method for the PDF equations of turbulent reactive flows, *J. Comput. Phys.* **154**, 342 (1999).
15. K. C. Kannenberg and I. D. Boyd, Strategies for efficient particle resolution in the direct simulation Monte Carlo method, *J. Comput. Phys.* **157**, 727 (2000).
16. G. Lapenta, Dynamic and selective control of the number of particles in kinetic plasma simulations, *J. Comput. Phys.* **115**, 213 (1994).
17. D. C. Haworth and S. H. El Tahry, PDF Approach for multidimensional turbulent flow calculations in reciprocating engines, *AIAA J.* **29**, 208 (1991).
18. S. Subramaniam and D. C. Haworth, A PDF method for turbulent mixing and combustion on three-dimensional unstructured deforming meshes, submitted for publication.
19. S. B. Pope, On the relationship between stochastic Lagrangian models of turbulence and second-moment closures, *Phys. Fluids* **6**(2), 973 (1994).

20. C. Dopazo, Probability density function approach for a turbulent axisymmetric heated jet. Centerline evolution, *Phys. Fluids* **18**(4), 397 (1975).

21. S. B. Pope, Particle method for turbulent flows: Integration of stochastic model equations, *J. Comput. Phys.* **117**, 332 (1995).

22. R. Lohner, A vectorized particle tracer for unstructured grids, *J. Comput. Phys.* **91**, 22 (1990).

23. D. J. Mavriplis, Unstructured grid techniques, *Annu. Rev. Fluid Mech.* **29**, 473 (1997).

24. FLUENT, Computational fluid dynamics software, *Version 5* (1998).