Journal of Computational Physics 257 (2014) 352–373

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Implicit and explicit schemes for mass consistency preservation in hybrid particle/finite-volume algorithms for turbulent reactive flows

Pavel P. Popov, Stephen B. Pope

Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA

ARTICLE INFO

Article history:
Received 11 December 2012
Received in revised form 30 May 2013
Accepted 5 September 2013
Available online 13 September 2013

Keywords:
Large eddy simulation/probability density function
Turbulent reactive flows
Lagrangian Monte Carlo
Consistency conditions
Particle tracking
Interpolation
Corrective algorithms

ABSTRACT

This work addresses the issue of particle mass consistency in Large Eddy Simulation/Probability Density Function (LES/PDF) methods for turbulent reactive flows. Numerical schemes for the implicit and explicit enforcement of particle mass consistency (PMC) are introduced, and their performance is examined in a representative LES/PDF application, namely the Sandia–Sydney Bluff-Body flame HM1. A new combination of interpolation schemes for velocity and scalar fields is found to better satisfy PMC than multilinear and fourth-order Lagrangian interpolation. A second-order accurate time-stepping scheme for stochastic differential equations (SDE) is found to improve PMC relative to Euler time stepping, which is the first time that a second-order scheme is found to be beneficial, when compared to a first-order scheme, in an LES/PDF application. An explicit corrective velocity scheme for PMC enforcement is introduced, and its parameters optimized to enforce a specified PMC criterion with minimal corrective velocity magnitudes.

1. Introduction

In the field of turbulent reactive flow simulation, probability density function (PDF) methods have been shown to be effective in modeling turbulence–chemistry interactions [16,19,4], due to the fact that no modeling is required for the chemical source term, which is highly non-linear. Originally used as turbulence–chemistry interaction models in Reynolds-averaged Navier–Stokes (RANS) simulations of turbulent flow, PDF methods for turbulent reactive flow are now increasingly used in conjunction with Large Eddy Simulation (LES) turbulence models [14]. The resulting LES/PDF methods, which were first introduced in [17], have been highly effective in simulating laboratory-scale reactive flows [1,22,26,12].

In a typical LES/PDF simulation, the sample space of the PDF of chemical compositions has a high dimension, which makes standard, finite-difference-based methods for evolving the PDF prohibitively expensive. This necessitates the use of Monte Carlo methods for approximating the composition PDF. In this paper, we focus on the Lagrangian particle Monte Carlo approach for approximation of the composition PDF, in which an ensemble of particles, each with its own composition, is advected in physical space according to the LES Favre-averaged velocity and turbulent diffusivity, i.e., the diffusivity used to model the unresolved turbulent motions. Other Monte Carlo approximation approaches are available [15,23], each with its advantages and disadvantages over the Lagrangian particle approach.

Here, we focus on particle mass consistency (PMC), which is an important requirement in order for the Lagrangian particle ensemble to be a valid Monte Carlo approximation of the composition PDF. First identified in [11], PMC means that

E-mail address: ppopov@uci.edu (P.P. Popov).

0021-9991/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2013.09.005

the expected mass of particles in a given region should equal the mass of fluid in that region as implied by the resolved density of the finite-volume (FV) solver.

Previous work on the PMC problem has included the introduction of Cartesian velocity interpolation schemes designed to give accurate values for the velocity divergence [6,10], as well as the use of a corrective velocity in the advection of particles [11,27], which reduces discrepancies in the particle mass consistency condition that accumulate due to numerical errors. Here, we extend these results by adapting the Cartesian velocity interpolation scheme of [10] to cylindrical coordinates (which are more often used for the simulation of statistically axi-symmetric canonical laboratory flames), introducing a new scheme for scalar interpolation of the turbulent diffusivity, and testing the degree of satisfaction of the PMC condition in a turbulent reactive flow representative of typical LES/PDF applications. We also employ a corrective velocity scheme, which is conceptually similar to those of [11,27], but optimized to keep mass consistency errors down to an acceptable level while minimizing the magnitude of the corrective velocities. Additionally, we examine the influence of the particle tracking SDE time integration scheme on the satisfaction of PMC.

The rest of this paper is organized as follows: in Section 2, we introduce the governing equations for an LES/PDF Lagrangian Monte Carlo simulation, and define the PMC condition. Section 3 describes the turbulent reactive flow simulation which is used to test, in conditions representative of a typical LES/PDF simulation, different schemes for the preservation of PMC. In Section 4, we compare the performance (with respect to how well PMC is satisfied) of the Euler SDE time integration scheme with that of a second-order scheme introduced by Kloeden and Platen [7]. Section 5 introduces the corrective velocity scheme for reducing PMC errors once they have accumulated due to numerical error, and determines its optimal implementation, in terms of satisfying an appropriate PMC criterion with the least possible corrective velocity magnitude. Finally, Section 6 introduces the new interpolation schemes, and tests their performance in the PMC context. Conclusions are drawn in Section 7.

2. LES/PDF equations and the PMC problem

2.1. Governing equations and the PMC condition

We begin by defining the LES/PDF governing equations. We denote by $\bar{p}$ and $\bar{\rho}$ the LES resolved pressure and density, by $\tilde{u}_j$, $\tilde{\nu}$ and $\tilde{D}$ the Favre-averaged resolved velocity, molecular viscosity and molecular diffusivity, and by $\tilde{S}_{ij}$ the resolved strain rate.

For the test case considered here, the molecular viscosity and diffusivity are evaluated by the power law

$$\tilde{\nu} = \nu_0 \left(\frac{\tilde{T}}{300\ \mathrm{K}} \right)^{1.69}, \qquad \frac{\tilde{\nu}}{\tilde{D}} = \sigma, \qquad \sigma = 0.82,\ \nu_0 = 1.42 \times 10^{-5} \left[\frac{\mathrm{m}^2}{\mathrm{s}} \right], \tag{1}$$

where the exponent in the power law is based on a curve fit to a CHEMKIN laminar flame calculation [26], ν_0, σ are based on the properties of a stoichiometric fuel/air mixture at 300 K, and $\tilde{T}$ denotes the resolved temperature. The resolved density is defined later in this subsection.

A turbulent viscosity and a turbulent diffusivity, $\tilde{\nu}_T$ and $\tilde{D}_T$ respectively, are used to model the unresolved motions. The turbulent viscosity and diffusivity are evaluated by the Dynamic Smagorinsky procedure, with Δ denoting the filter size.

With these definitions, the equations of motion solved by the LES component of an LES/PDF algorithm take the form:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_j}{\partial x_j} = 0, \tag{2}$$

$$\frac{\partial (\bar{\rho} \tilde{u}_j)}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_i \tilde{u}_j)}{\partial x_i} = -\frac{\partial \bar{p}}{\partial x_j} + 2 \frac{\partial}{\partial x_i} \left(\bar{\rho} (\tilde{\nu} + \tilde{\nu}_T) \left(\tilde{S}_{ij} - \frac{1}{3} \tilde{S}_{kk} \delta_{ij} \right) \right). \tag{3}$$

The LES component also evaluates the scalar mixing frequency, Ω, defined as

$$\Omega = C_\phi \frac{\tilde{D}_T + 2\tilde{D}}{\Delta^2}, \tag{4}$$

which is used in the equations solved by the PDF component (defined below) to evaluate turbulent mixing by the IEM (Interaction by Exchange with the Mean) [25] mixing model, with the mixing constant set as $C_\phi = 2.0$. The IEM mixing model is one of the simplest turbulent mixing models, working on the assumption that the difference between a particle composition and the local (i.e., cell-averaged) mean decays exponentially in time – this results in the first term on the right-hand side of Eq. (11).

We note that, in an LES/PDF simulation, the LES component solves only for the continuity and momentum equations, unlike a standalone LES turbulent reactive flow simulation, which also needs to solve scalar evolution equations for the fields used to determine the chemical composition: in the LES/PDF context the evolution of the composition fields is performed by the PDF component, as described below.

In the present work material properties are a function of the composition vector, ϕ_α, which is either one-dimensional, consisting only of the mixture fraction, ξ, when a flamelet model is used, or it consists of a set of species' specific mole values, plus enthalpy. The particular flamelet model used here (referred to from here on simply as the "flamelet model") is the fast-chemistry flamelet model defined in [28], which is a function only of the mixture fraction and its subgrid variance, with the implicit assumption that the chemistry is much faster than the mixing processes.

The PDF component of an LES/PDF algorithm is a Monte Carlo approximation of the mass-weighted PDF of chemical compositions, $f(\psi;\mathbf{x},t)$, conditional on the resolved velocity field [2,20]. Here ψ_α denote points in the sample space of the composition vector ϕ_α. Using $\tilde{\phi}_\alpha(\mathbf{x},t)$ to denote the sample space mean of ϕ_α at $(\mathbf{x},t)$, the evolution equation for $f(\psi;\mathbf{x},t)$, is modeled to have the form [25]

$$\frac{\partial f}{\partial t}+\frac{\partial}{\partial x_i}(f\tilde{u}_i)=\frac{\partial}{\partial x_i}\left(\tilde{D}_T\frac{\partial f}{\partial x_i}\right)+\frac{\partial}{\partial \psi_\alpha}\big(f\Omega(\psi_\alpha-\tilde{\phi}_\alpha)\big)$$
$$-\frac{\partial}{\partial \psi_\alpha}\left(f\frac{1}{\bar{\rho}}\frac{\partial}{\partial x_i}\left(\bar{\rho}\tilde{D}\frac{\partial\tilde{\phi}_\alpha}{\partial x_i}\right)\right)+\frac{\partial}{\partial \psi_\alpha}\big(fS_\alpha(\psi)\big). \tag{5}$$

In the above equation, the left-hand side accounts for transport in physical space due to the velocity and diffusivity gradients, whereas the first three terms on the right-hand side of Eq. (5) represent respectively turbulent diffusion using the turbulent diffusivity hypothesis, turbulent mixing using the IEM mixing model, and molecular diffusion, and $S_\alpha(\boldsymbol{\psi})$ is the reaction source term ($S_\alpha(\boldsymbol{\psi})=0$ when a flamelet model is used). The advantage of modeling chemistry by using a mass-weighted composition PDF which evolves by Eq. (5) is that the source term, which is highly non-linear as a function of $\boldsymbol{\psi}$, requires no modeling. Additionally, in the DNS limit, when $\tilde{D}_T=0$, the results yielded by the present specification of the PDF are consistent with an exact solution of the Navier–Stokes and chemical transport equations [9].

With this definition of the mass-weighted PDF, the resolved specific volume, $\tilde{v}$, is defined by

$$\tilde{v}(\mathbf{x},t)=\int v(\boldsymbol{\psi})f(\boldsymbol{\psi};\mathbf{x},t)|\mathrm{d}\boldsymbol{\psi}|, \tag{6}$$

where $v(\boldsymbol{\psi})$ is the constitutive relation for specific volume as a function of the composition variables. Finally, the resolved density is defined as

$$\bar{\rho}=1/\tilde{v}. \tag{7}$$

An additional scalar which we solve for is the transported specific volume, $\hat{v}$, whose role in the coupling of the LES and PDF components is elaborated on in Section 2.2, and which is defined to evolve by the equation

$$\frac{\partial(\bar{\rho}\hat{v})}{\partial t}+\frac{\partial(\bar{\rho}\tilde{u}_j\hat{v})}{\partial x_j}=\frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{D}_T\frac{\partial\hat{v}}{\partial x_j}\right)+S_v+\omega_v, \tag{8}$$

where S_v and ω_v are source and relaxation terms [21]. In particular, introducing a relaxation time τ (which is in this study set to $4\Delta t$) the relaxation term is defined as

$$\omega_v=\bar{\rho}\frac{\tilde{v}-\hat{v}}{\tau}, \tag{9}$$

which insures that $\hat{v}$ tends to $\tilde{v}$ in the limit when τ goes to zero and the number of particles per cell goes to infinity.

In order to efficiently calculate the evolution of Eq. (5), without having to perform a discretization on the high-dimensional composition space, we perform a Monte Carlo approximation [16]. An ensemble of Lagrangian particles are initialized throughout the computational domain and evolve by a set of stochastic differential equations (SDEs). Throughout this work, we shall use the superscript * to denote particle quantities, as opposed to Eulerian fields (such as $\tilde{u}_j$ and $\bar{\rho}$); we shall also use it to denote Eulerian fields interpolated at a given particle's current location.

Thus, each particle has a mass m^*, determined at its initialization to correspond to the mass of fluid in the region taken up by the particle, a location X_j^* which evolves according to the LES velocity and turbulent diffusivity, and a composition vector, ϕ_α^*. The evolution equations for the particle position and composition vectors are as follows:

$$dX_j^*=\left[\tilde{u}_j+\frac{1}{\bar{\rho}}\frac{\partial(\bar{\rho}\tilde{D}_T)}{\partial x_j}\right]^*dt+\big[2\tilde{D}_T^*\big]^{1/2}dW_j^*, \tag{10}$$

$$d\phi_\alpha^*=-\Omega^*\big(\phi_\alpha^*-\tilde{\phi}_\alpha^*\big)\,dt+\left[\frac{1}{\bar{\rho}}\frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{D}\frac{\partial\tilde{\phi}_\alpha}{\partial x_j}\right)\right]^*dt+S_\alpha\big(\boldsymbol{\phi}^*\big)\,dt, \tag{11}$$

where the term dW_j^* in Eq. (10) denotes a Wiener increment, and the three terms on the right-hand side of Eq. (11) denote respectively turbulent mixing (in this case represented by the IEM model [24]), molecular diffusion, and chemical reaction.

Using the notation $\langle\cdot\rangle$ to denote expectation over all possible initial particle locations and all possible realizations of the Wiener process, we define the particle mass-weighted PDF, $f^*(\boldsymbol{\psi};\mathbf{x},t)$ by:

$$f^*(\boldsymbol{\psi};\mathbf{x},t)=\frac{\langle m^*\delta(\mathbf{X}^*(t)-\mathbf{x})\delta(\boldsymbol{\phi}^*(t)-\boldsymbol{\psi})\rangle}{\langle m^*\delta(\mathbf{X}^*(t)-\mathbf{x})\rangle}, \tag{12}$$

and, in the absence of numerical errors, Eqs. (10), (11) imply that f^* evolves by Eq. (5), so that, provided that $f = f^*$ initially and at the boundaries, the particle mass-weighted PDF f^* is equal to the mass-weighted PDF:

$$f(\boldsymbol{\psi};\mathbf{x},t) = f^*(\boldsymbol{\psi};\mathbf{x},t), \tag{13}$$

due to the fact that both PDF functions have the same evolution equation, initial and boundary conditions.

An additional quantity which we introduce at this point is the particle mass density, q, which is defined as the expected mass density of particles at $\mathbf{x}$,

$$q(\mathbf{x},t) \equiv \big\langle m^*\delta\big(\mathbf{X}^*(t) - \mathbf{x}\big)\big\rangle. \tag{14}$$

The main focus of this paper is in the preservation of the consistency,

$$q = \bar{\rho}, \tag{15}$$

between particle mass density and LES resolved density: we shall refer to Eq. (15) as the PMC condition. As pointed out by [11,27], Eq. (15) is one of the essential conditions for the correspondence between the PDFs f and f^* which we use in our Monte Carlo approximation. In the absence of numerical errors the consistency condition of Eq. (15) is an identity because Eq. (2) can be simply rearranged to yield

$$\frac{\tilde{D}\ln(\bar{\rho})}{\tilde{D}t} = -\frac{\partial\tilde{u}_j}{\partial x_j}, \tag{16}$$

where $\frac{\tilde{D}}{\tilde{D}t}$ denotes a convective derivative with velocity $\tilde{u}_j$, whereas Eq. (10) implies that q evolves by

$$\frac{D^*\ln(q)}{Dt^*} = -\frac{\partial\tilde{u}_j^*}{\partial x_j}, \tag{17}$$

where $\frac{D^*}{Dt^*}$ denotes a convective derivative with velocity $\tilde{u}_j^*$. Therefore, in the absence of numerical errors, the evolution equations, Eqs. (16), (17) imply $q = \bar{\rho}$ for all time provided that $q = \bar{\rho}$ at $t = 0$ and at inflow boundaries.

2.2. Numerical implementation

The LES equations are solved on a finite volume (FV) grid, with x, r, θ denoting respectively the axial, radial and azimuthal coordinates [13]. The domain is also divided into a set of PDF cells, each of which contains approximately n_{pc} particles, where n_{pc} is a specified parameter. The PDF cells are identical to the LES cells away from the centerline; close to the centerline, one PDF cell consists of several LES cells, concatenated in the azimuthal direction [26].

The ensemble of particles evolving by Eqs. (10), (11) is used to determine local cell values as mass-weighted sums. More specifically, let $B(\mathbf{x})$ be a given cell basis function: in the present work, $B(\mathbf{x})$ are continuous tent functions, piecewise linear in each of the x, r, θ coordinates, which form a partition of unity. For a particular cell, we have:

$$B(\mathbf{x}) = B_x(x)B_r(r)B_\theta(\theta), \tag{18}$$

where, denoting by x_0 the axial location of the cell's midpoint, and denoting by x_1, x_{-1} the axial locations of its axial neighbors, $B_x(x)$ is defined as

$$B_x(x) = \max\left(0, \min\left(\frac{x - x_1}{x_0 - x_1}, \frac{x - x_{-1}}{x_0 - x_{-1}}\right)\right), \tag{19}$$

and $B_r(r), B_\theta(\theta)$ are similarly defined.

For a specific particle property y, the Favre mean of y at the location of a given LES/PDF cell is approximated as

$$\tilde{y} \approx \frac{\sum_{cell} m^* y^* B(\mathbf{X}^*(t))}{\sum_{cell} m^* B(\mathbf{X}^*(t))}, \tag{20}$$

where the summation on the right-hand side is over all particles which are currently in the support of $B(\mathbf{x})$, which consists of the cell on which $B(\mathbf{x})$ is centered, and parts of its immediate neighbors in each of the x, r, θ directions.

This methodology for cell value estimation is used to communicate information from the PDF to the LES component of the LES/PDF algorithm. For example, a straightforward method for evaluating the resolved density, $\bar{\rho}$, used in Eqs. (2), (3), is to just take the inverse of the Favre-averaged specific volume, $\tilde{v}$. In the present work, $\bar{\rho}$ is estimated by another method, called the transported specific volume approach, which reduces the amount of statistical error (see [21] for more details). In that approach, we use the additional transported scalar $\hat{v}$, which is solved for by the LES code (see Eq. (8)), and use the inverse of that specific volume as an approximation of the resolved density:

$$\hat{\rho} \equiv \frac{1}{\hat{v}} \approx \bar{\rho}. \tag{21}$$

As previously pointed out, Eqs. (8), (9) imply that, for the present choice of $\tau = 4\Delta t$, $\hat{v}$ tends to $\tilde{v}$ in the limit as Δt goes to zero and the number of particles per cell goes to infinity, and hence using Eq. (21) to obtain the resolved density yields a valid numerical approximation to the system of equations defined in the previous subsection.

In the present work, q is approximated on LES/PDF cells by the formula

$$q \approx \frac{\sum_{cell} m^* B(\mathbf{X}^*)}{\int_{cell} B(\mathbf{x})|\mathrm{d}\mathbf{x}|}, \tag{22}$$

where $\int_{cell}$ is an integral over the support of $B(\mathbf{x})$.

As previously noted, in the absence of numerical error $q = \bar{\rho}$ exactly. However, in a practical LES/PDF simulation, differences between q and $\bar{\rho}$ will develop, and need to be kept at a low level so that the combination of the LES (Eqs. (2), (3)) and PDF (Eqs. (10), (11)) components of the code yields a valid approximation to the mass-weighted composition PDF.

Here, we introduce a PMC error variable ϵ^c, defined by

$$\epsilon^c = q/\bar{\rho} - 1, \tag{23}$$

to quantify the difference between q and $\bar{\rho}$, and we rank the performance of different PMC preservation schemes by the approximate L^1 and L^∞ measures of ϵ^c, defined by

$$\|\epsilon^c\|_1 = \frac{\sum_{i,m} |\epsilon_i^c(t_m)| V_i \Delta t}{\sum_{i,m} V_i \Delta t} \tag{24}$$

and

$$\|\epsilon^c\|_\infty = \max_{i,m} |\epsilon_i^c(t_m)| \tag{25}$$

where i is the index of a given LES cell, t_m denotes the midpoint of a given time step in the second half of the simulation, and $\epsilon_i^c(t_m)$ and V_i denote respectively the value of ϵ^c for a given cell at time t_m, and the volume of that cell.

Our use of the L^∞ and L^1 norms of the error variable, ϵ^c, stems from the fact that they are respectively the strictest and the least strict measures of error: for any function f on a domain of volume 1 (which applies to the present case, since in Eq. (24) we divide by the domain's volume), it is identically true that $\|f\|_1 \leqslant \|f\|_p \leqslant \|f\|_\infty$ for any $1 < p < \infty$. While we could use this fact to limit our attention to the L^∞ norm only, that norm has little physical significance and for certain applications may prove to be overly restrictive. As the results in Sections 4 through 6 indicate, however, this is not the case in the present simulation.

3. Description of the bluff-body jet simulation

Here, we describe the turbulent reactive flow simulation used as a test case for different PMC preservation schemes.

We simulate the Sandia–Sydney Bluff-Body Flame HM1, a detailed description of which can be found in Masri and Bilger [8]. The flame consists of a jet of diameter 3.6 mm inside a bluff body of diameter 50 mm, which is itself located inside a square wind tunnel whose sides are 150 mm. The jet is a 1 : 1 by volume mixture of CH_4 and H_2 at a temperature of 300 K, and leaves the outlet at a bulk velocity of 118 m/s – the Reynolds number, based on the jet velocity and diameter, is 29,900. The coflow consists of air, also at 300 K, with a bulk velocity of 40 m/s. The computational domain is $x \in [0, 20R_B]$, $r \in [0, 3.39R_B]$, where R_B is the bluff-body radius. There is a wall boundary at $r = 3.39R_B$, which yields the same hydraulic diameter as the wind tunnel used in the experiment. The grid size is $96 \times 64 \times 48$, which is not sufficient for an accurate simulation of the bluff body, but allows us to test the PMC preservation properties of a number of different numerical schemes in a flow representative of LES/PDF applications.

In the present simulation, we use the GRI 1.2 chemical mechanism with chemical composition being advanced either by flamelet modeling, for which the composition vector ϕ_α^* consists only of the mixture fraction ξ, or via integration in the full composition space, in which case ϕ_α^* is 33-dimensional – we shall use the abbreviation FC from now on to refer to the full composition space simulations. In FC simulations, *in situ* adaptive tabulation (ISAT) [18] is used for speed up of the integration of the chemical source term. The number of particles per cell is either $n_{pc} = 30$ or $n_{pc} = 50$, and the time step is $\Delta t = 8.48$ µs. The reason for the flamelet model in addition to FC, which is typically used in LES/PDF simulations, is that the former has considerably lower computational cost which makes it suitable for parameter studies and simulations with a larger number of particles. The flamelet model simulations provide an accurate approximation, where PMC error is concerned, to the FC simulations that we are mainly interested in, due to the fact that the density variation over the entire composition space is comparable between the two – this is supported by the results in Section 4, which show almost identical PMC errors in a comparison between flamelet and FC solutions with the same number of grid cells and particles.

The simulations are run for 12 flow-through times based on the coflow velocity (which amounts to 20,000 time steps), by the latter half of which period the flow has reached a statistically stationary state: the results presented are based on statistics collected during the second half of the simulation interval.

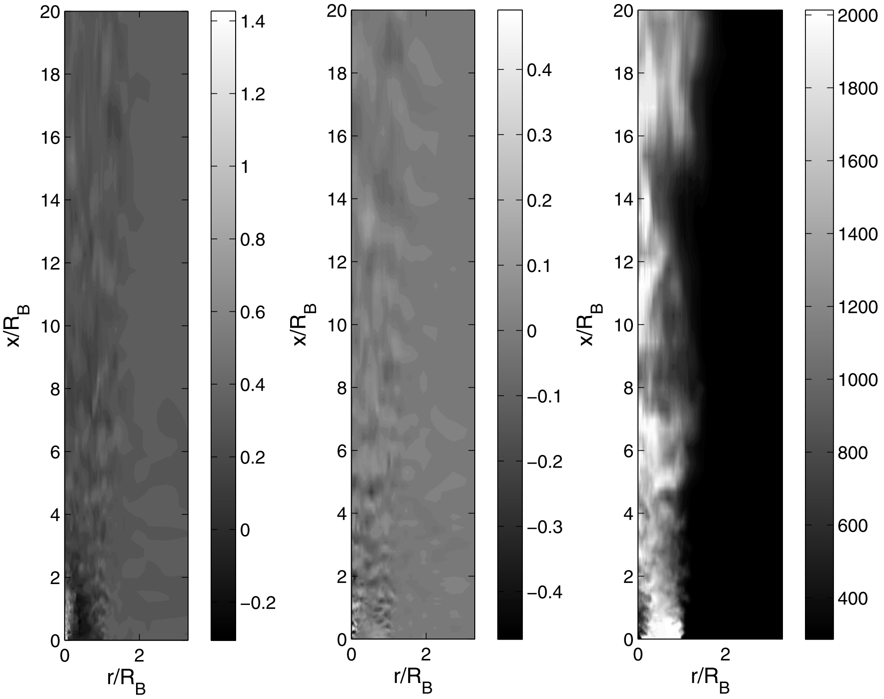

Fig. 1. Instantaneous contour plots at the end of an FC simulation of the HM1 test case. Left: axial velocity, normalized by the jet bulk velocity. Middle: radial velocity, normalized by the jet bulk velocity. Right: temperature (K).

Figs. 1–4 show contour plots of the Favre means of the velocity and temperature fields, as well as plots of the instantaneous fields at the end of the simulation. As can be seen, the flow is highly non-trivial, with a recirculation region extending up to a distance $2R_B$ downstream from the bluff body, and considerable fluctuations about the mean. This makes it a good test case to study the problem PMC, which depends considerably on the evolution of particles in physical space.

In the subsequent sections, we present results for this test case with different implicit and explicit schemes of preservation of PMC.

4. SDE time integration as an implicit scheme for PMC preservation

First, we examine the effect on PMC of the numerical scheme used to integrate the position advection SDE, Eq. (10). The standard Euler SDE time integration scheme is most commonly used [3,5] in particle/FV algorithms for turbulent reactive flows; those studies which use an SDE integration scheme with a higher order of accuracy [26] have detected little improvement in the overall solution, for test cases which are representative of typical LES/PDF turbulent flows. Here, we compare the performance of the Euler scheme with that of the weakly second-order accurate derivative-free SDE integration scheme introduced by Kloeden and Platen [7], from here on referred to as the KP scheme, for the sake of brevity.

Table 1 gives a PMC error comparison between simulations using Euler and the KP time integration schemes. The simulation is as described in Section 3 – both the flamelet and FC calculations are used. The value of n_{pc} used is $n_{pc} = 30$, and the interpolation scheme for the velocity and scalar fields is standard multilinear interpolation; no explicit PMC correction algorithms are used. Also given is the computational cost, in overall simulation wall clock time per particle per step – the total number of particles is approximated as n_{pc} times the number of grid cells.

As can be seen on the table, in both the flamelet and FC simulations, the use of the KP scheme instead of Euler reduces both the L^1 and L^∞ errors by approximately 30%, for a modest increase in computational cost, especially for the FC simulations. Thus we see that, in contrast to previous experience (in which PMC errors were not examined) with particle/FV methods, it is advantageous to use a higher-order SDE integration scheme. It can also be seen from Table 1 that, with respect to PMC errors, there is little difference between the flamelet and FC simulations.

Fig. 5 gives a detailed look at the PMC error variable, ϵ^c, throughout the domain, for the FC simulation with Euler time stepping – the results for the other simulations are similar, the main difference being in the magnitude of ϵ^c. As can be seen on Fig. 5, while there is considerable noise in the instantaneous values of ϵ^c, there is also a deterministic component in the ϵ^c field, which can be seen on the time-averaged contour plot. At both the jet and coflow shear layers, we see a region with positive $\langle \epsilon^c \rangle_T$, i.e. a greater mass density of particles than there should be, on the bluff-body side of the shear layers, and a region with negative $\langle \epsilon^c \rangle_T$ at the inlet sides of the shear layers.

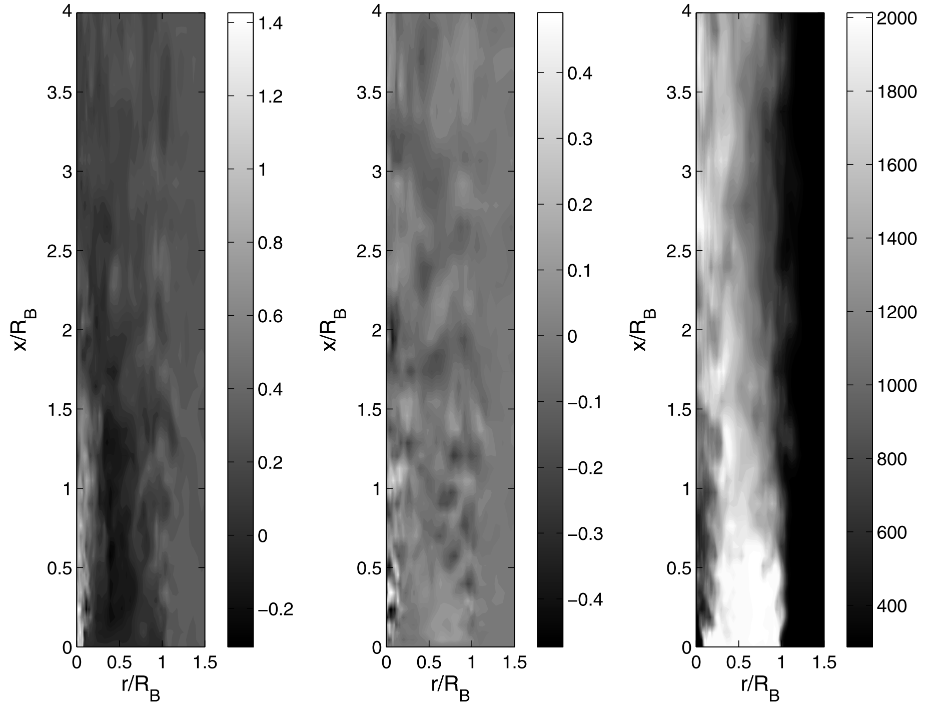

Fig. 2. Instantaneous contour plots (enlarged view of the recirculation region) at the end of an FC simulation of the HM1 test case. Left: axial velocity, normalized by the jet bulk velocity. Middle: radial velocity, normalized by the jet bulk velocity. Right: temperature (K).

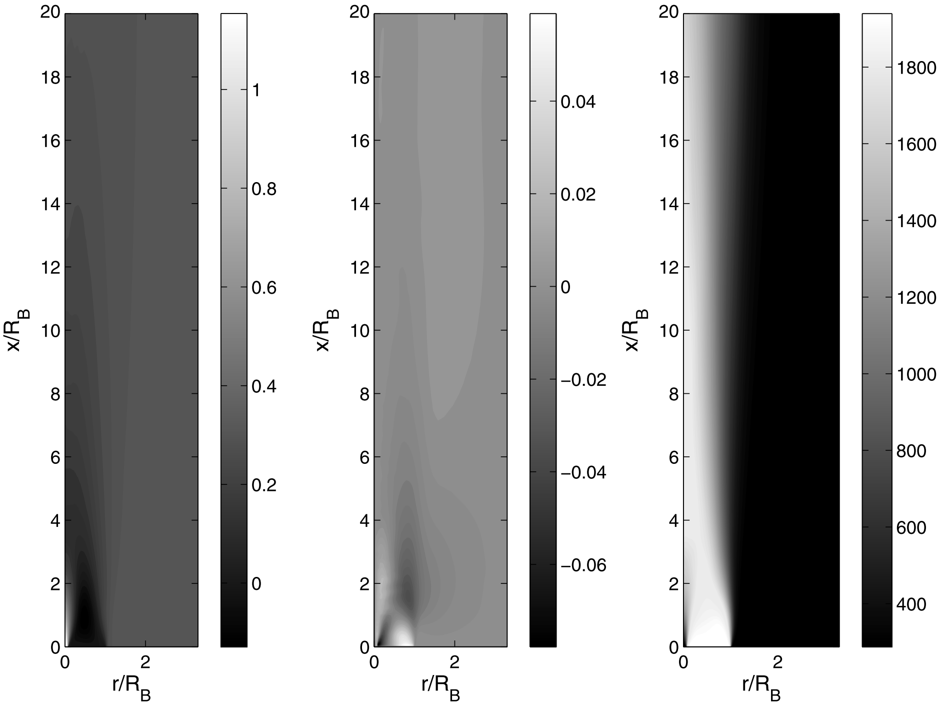

Fig. 3. Contour plots of time-averaged fields in the FC simulation of the HM1 test case. Left: axial velocity, normalized by the jet bulk velocity. Middle: radial velocity, normalized by the jet bulk velocity. Right: temperature (K).

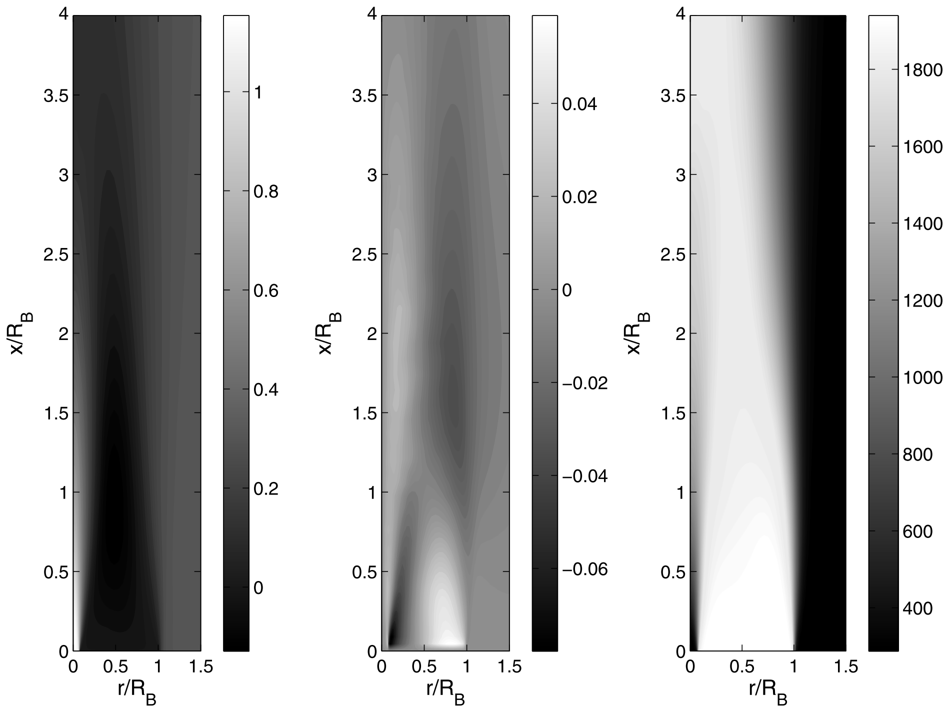

Fig. 4. Contour plots (enlarged view of the recirculation region) of time-averaged fields in the FC simulation of the HM1 test case. Left: axial velocity, normalized by the jet bulk velocity. Middle: radial velocity, normalized by the jet bulk velocity. Right: temperature (K).

Table 1
L^1 and L^∞ measures, $\|\epsilon^c\|_1$ and $\|\epsilon^c\|_\infty$, of the instantaneous PMC error, L^1 and L^∞ measures, $\|\langle\epsilon^c\rangle_T\|_1$ and $\|\langle\epsilon^c\rangle_T\|_\infty$, of the time-averaged error $\langle\epsilon^c\rangle_T$, and computational cost in wall clock time per particle per step per core, for simulations with multilinear interpolation and no explicit mass correction.

Chemistry/SDE scheme	Flamelet/Euler	Flamelet/KP	FC/Euler	FC/KP
$\|\epsilon^c\|_1$	1.54×10^{-1}	1.09×10^{-1}	1.67×10^{-1}	1.16×10^{-1}
$\|\epsilon^c\|_\infty$	5.78×10^{-1}	3.97×10^{-1}	5.82×10^{-1}	3.95×10^{-1}
$\|\langle\epsilon^c\rangle_T\|_1$	1.88×10^{-2}	1.05×10^{-2}	2.32×10^{-2}	1.03×10^{-2}
$\|\langle\epsilon^c\rangle_T\|_\infty$	2.12×10^{-1}	1.22×10^{-1}	2.14×10^{-1}	1.20×10^{-1}
cost/ptcl	10.2 μs	13.3 μs	81.7 μs	84.2 μs

Table 1 also provides L^1 and L^∞ measures of these time-averaged values, denoted by $\langle\epsilon^c\rangle_T$, for the four simulations.

As we can see, the KP scheme yields approximately a 45% reduction in the time-averaged, deterministic components of PMC error. Also, we note again that, with respect to PMC errors, there is little difference between flamelet and FC simulations.

The mechanism for accumulation of deterministic PMC errors is replicated in a simple 2D numerical test case, described in Appendix A. In that test case, the only source of error is due to the time-stepping scheme, which allows us to isolate the PMC errors due to time stepping, and evaluate the performance of different SDE integration schemes, apart from the other components of an LES/PDF solution. The 2D test flow described in the appendix is a model for the bluff-body shear layer: it contains an interface between regions of high and low axial velocity; this interface coincides with a region of increased turbulent diffusivity, similarly to the flow features seen on Fig. 6. The results are shown on Figs. 7 and 8; as can be seen on these figures, there is a buildup of deterministic mass consistency errors, with positive $\langle\epsilon^c\rangle_T$ in the region of low axial velocity, and negative $\langle\epsilon^c\rangle_T$ in the region of high axial velocity. This distribution is similar to the distribution of deterministic errors seen on Fig. 5, which suggests that the latter are in large part caused by time-stepping error. This explains the reduction of deterministic PMC error seen on Table 1, as it can be seen on Figs. 7 and 8 that the KP scheme greatly reduces the deterministic PMC errors due to time stepping.

From the above results we conclude that, despite the modest increase in computational cost, from a PMC standpoint it is advantageous to use an SDE time integration scheme with a higher order of accuracy than the Euler scheme. Note that while it is intuitive that using a higher-order integration algorithm will improve computational accuracy, in the LES/PDF context it is generally accepted that the Euler scheme is sufficiently accurate [26], due to the small time steps which are necessitated by the chemical source terms, and hence it is the Euler scheme that is used in most LES/PDF studies [3,5]. The importance of the results in this section lies in the finding that the PMC error, which has previously not been examined as a function of the time-stepping scheme, is reduced significantly by the use of the second-order KP scheme. This is the

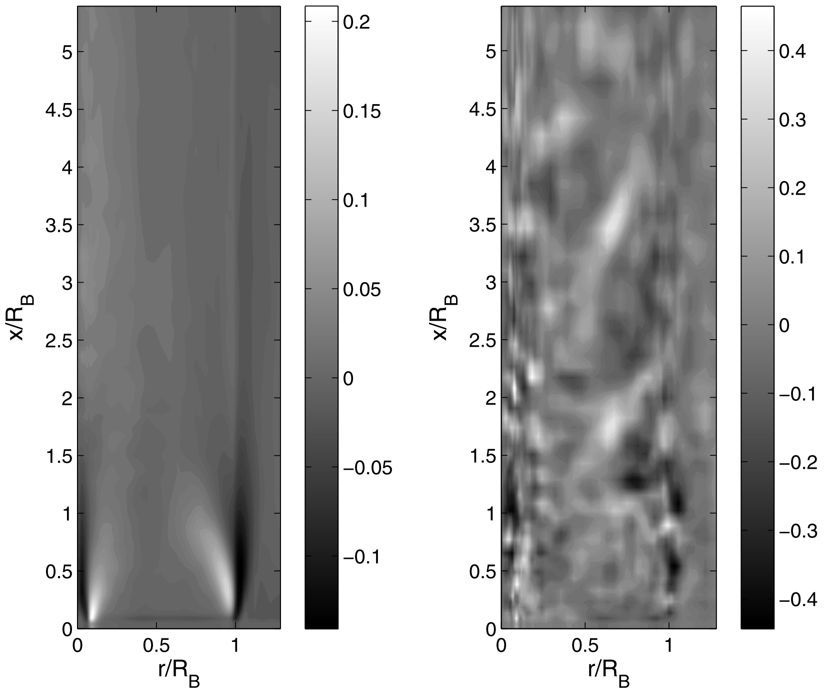

Fig. 5. Contour plots of the PMC error, ϵ^c for an FC simulation with multilinear interpolation, $n_{pc} = 30$ and no explicit PMC correction. Left: contour plot of ϵ^c averaged in time over the second half of the simulation. Right: contour plot of ϵ^c at the last simulation time step.

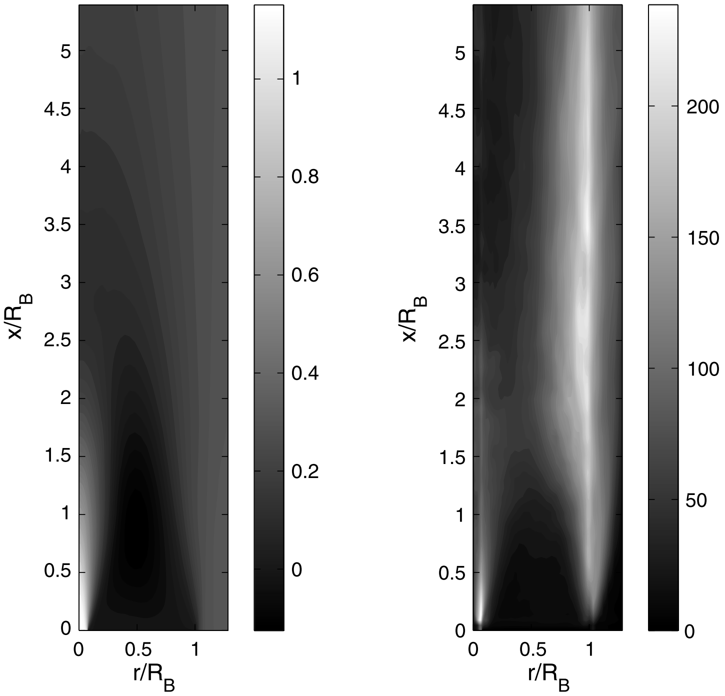

Fig. 6. For the test case described in Appendix A. Left: contour plot of $\langle \tilde{u}_x \rangle_t / U_J$, the time-averaged axial velocity normalized by the jet bulk velocity. Right: contour plot of $\frac{\langle \tilde{D}_T \rangle_t}{\Delta^2/\tau_J}$, the time-averaged turbulent diffusivity normalized by the LES filter size, Δ, and the jet characteristic time scale $\tau_J = 2R_B/U_J$.

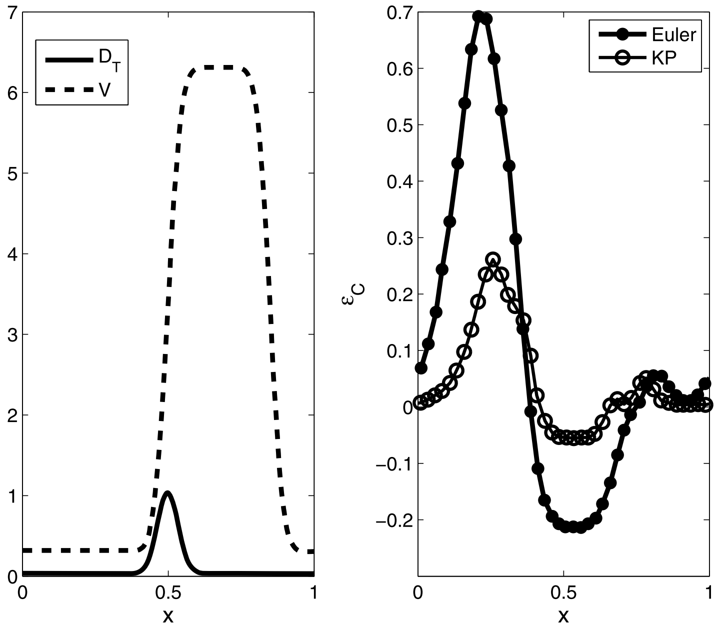

Fig. 7. Results from the 2D PMC test case described in Appendix A. Left: plots of $V(x)$ and $D_T(x)$. Right: plots of time-averaged PMC error, $\langle\epsilon^c\rangle_T$ for the region $0.3 < y < 0.45$, and for $\Delta t = 0.0125$.

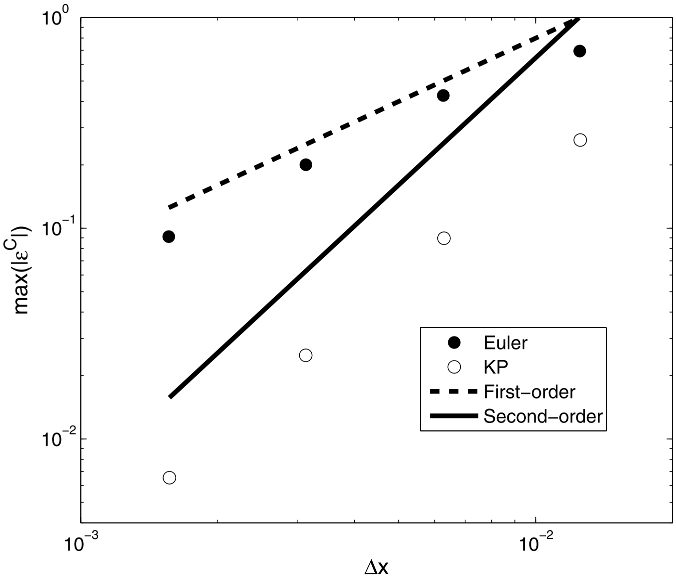

Fig. 8. Results from the 2D PMC test case described in Appendix A. Maximal value of ϵ^C plotted against the time step.

first time that a second-order accurate time-stepping scheme has been shown to have superior performance to Euler in an LES/PDF context: in particular, a comparison was made between the first- and second-order accurate time-stepping schemes in an LES/PDF context by Wang and Pope [26] and in that study no advantage was found in using the higher-order scheme. In the following sections, unless otherwise specified, it is implicit that the KP scheme is used.

5. Explicit PMC preservation schemes: Velocity correction

In this section, we describe and test a scheme for the reduction of PMC errors once they have accumulated due to numerical error.

5.1. Determining the desired maximum level of PMC error

Before we describe the explicit PMC correction schemes used here, let us consider the desired level, ϵ_0^c, of PMC error which we wish to enforce. Due to the stochastic nature of LES/PDF codes, it is undesirable to completely reduce ϵ^c to zero – for a given value of n_{pc} (which is typically in the range of 20 to 50 for most practical simulations) the error ϵ^c will contain a component proportional to $n_{pc}^{-1/2}$, which is due to the finite sample size in the approximation of q and does not in itself imply a discrepancy between q and $\bar{\rho}$. This is true even with perfect integration of the SDE in Eq. (10), which occurs for example when we have constant velocity and no turbulent diffusivity, so that the interpolation and SDE integration schemes contain no numerical errors; in this idealized case, $q = \bar{\rho}$ is satisfied exactly, and ϵ^c contains only sampling error which should be ignored.

Therefore, we shall set ϵ_0^c to be equal to the maximal rms ϵ^c error for a case with perfect integration of Eq. (10). For simplicity, let us consider a uniform Cartesian grid whose grid cells have volume 1, assume that the velocity is constant throughout the domain, that there is no diffusivity and that the basis functions $B(\mathbf{x})$ used for evaluation of cell means are the indicator functions of a given cell. Also, let us assume that the density $\bar{\rho}$ is constant, and that all particles have the same mass, so that $q/\bar{\rho}$ for a given cell can be evaluated as the number of particles in that cell, divided by n_{pc}.

At the beginning of a simulation, exactly n_{pc} particles are initialized with their position having uniform probability in each cell, so that $q/\bar{\rho} = 1$, hence $\epsilon_0^c = 0$. However, as the simulation progresses and the particle distribution shifts with respect to the FV grid, each FV grid cell will cover a region of particles which were initialized in 8 separate grid cells (i.e. 2^3 grid cells for the 3D case considered). Let us consider a given cell, C, and denote by V_1 through V_8 the volumes of the intersections between that cell and the Lagrangian mappings for the present time of the 8 cells, C_1 through C_8, whose particles currently reside in C. Due to the uniform initialization of particles and the fact that we have chosen all cells to have volume 1, we have that the probability that a particle initialized in C_i currently lies in C is equal to V_i, with $\sum_{i=1}^{8} V_i = 1$. Therefore, if we denote by N the number of particles currently in C, we get that N the sum of 8 binomial random variables with parameters n_{pc} and V_i, respectively. Hence, since $\sum_{i=1}^{8} V_i = 1$, the expectation of N is $E(N) = n_{pc}$, and the variance of N is

$$\operatorname{Var}(N) = n_{pc} \sum_{i=1}^{8} V_i (1 - V_i). \tag{26}$$

Given the constraint $\sum_{i=1}^{8} V_i = 1$, for a fixed n_{pc} the right-hand side of Eq. (26) attains its maximum for $V_1 = V_2 = \cdots = V_8 = 1/8$ (from the inequality between arithmetic and quadratic means), which gives us that the maximal value for $\operatorname{Var}(N)$ is $\operatorname{Var}(N) = 7/8n_{pc}$, and so the maximal possible standard deviation for $q/\bar{\rho} = N/n_{pc}$ for this simplified case is

$$\max\left(\operatorname{std}\left(\frac{q}{\bar{\rho}}\right)\right) = \left(\frac{7}{8n_{pc}}\right)^{1/2}. \tag{27}$$

Based on this, we set

$$\epsilon_0^c = \left(\frac{7}{8n_{pc}}\right)^{1/2} \tag{28}$$

as the maximal allowable consistency error, i.e. we require that $\|\epsilon^c\|_\infty < \epsilon_0^c$. For a criterion on $\|\epsilon^c\|_1$, we define ϵ_1^c to be the expectation of $|N/n_{pc} - 1|$:

$$\epsilon_1^c = E(N/n_{pc} - 1), \tag{29}$$

which is the L^1 counterpart of the variance-based (i.e., using an L^2 norm) criterion of Eq. (28), and we require that $\|\epsilon^c\|_1 < \epsilon_1^c$. For $n_{pc} = 30, 50$, ϵ_1^c has the values $0.136, 0.105$ respectively (obtained numerically). As we shall see below $\|\epsilon^c\|_\infty < \epsilon_0^c$ is a more restrictive criterion than $\|\epsilon^c\|_1 < \epsilon_1^c$.

5.2. Corrective schemes for the reduction of ϵ^c

As seen in the previous section, even though using the KP time integration scheme reduces the magnitude of the PMC errors, those errors are still considerably higher than the desired level ϵ_0^c introduced in Eq. (28) of the previous subsection – for $n_{pc} = 30$ we have $\epsilon_0^c = 0.1708$, whereas the smallest value of $\|\epsilon^c\|_\infty$ obtained in the previous section is $\|\epsilon^c\|_\infty = 0.397$. This necessitates the use of a correction algorithm to reduce PMC errors down to a desired level. Such correction algorithms typically take the form of velocity correction algorithms [27,11]. In this work, we also use a velocity correction algorithm similar to that of [27].

In particular, we introduce a corrective velocity, u_j^c, as a discrete FV field, and we set

$$\tilde{u}_j^* = \tilde{u}_j + u_j^c. \tag{30}$$

Then, making the assumption that $\|u_j^c\| \ll \|\tilde{u}_j\|$ and that interpolation errors are small, we get that $\frac{\tilde{D}}{\tilde{D}t} \approx \frac{D^*}{Dt^*}$, and so Eqs. (16), (17) yield that

$$\frac{\tilde{D}\ln(q/\bar{\rho})}{\tilde{D}t} \approx \frac{D^*\ln(q)}{Dt^*} - \frac{\tilde{D}\ln(\bar{\rho})}{\tilde{D}t} = -\frac{\partial u_j^c}{\partial x_j}. \tag{31}$$

Then, assuming that $\epsilon^c \ll 1$ (the results to follow validate this assumption when a corrective scheme is used), so that $\frac{q}{\bar{\rho}} \approx 1$, we get that

$$\frac{\tilde{D}\ln(q/\bar{\rho})}{\tilde{D}t} = \frac{\bar{\rho}}{q}\frac{\tilde{D}(q/\bar{\rho})}{\tilde{D}t} \approx \frac{\tilde{D}\epsilon^c}{\tilde{D}t}, \tag{32}$$

and hence

$$\frac{\tilde{D}\epsilon^c}{\tilde{D}t} \approx -\frac{\partial u_j^c}{\partial x_j}, \tag{33}$$

which means that setting $\frac{\partial u_j^c}{\partial x_j}$ to be proportional to ϵ^c will enforce a decay of ϵ^c toward zero. More specifically, we introduce a velocity potential, ζ, so that

$$u_j^c = \frac{\partial \zeta}{\partial x_j}, \tag{34}$$

and we require ζ to satisfy the Poisson equation

$$\frac{\partial u_j^c}{\partial x_j} = \frac{\partial^2 \zeta}{\partial x_i \partial x_i} = \frac{F(\epsilon^c)}{\tau^c}, \tag{35}$$

where on the right-hand side of the above equation: $F(\cdot)$ is an operator acting on the ϵ^c field, the simplest example being $F(\epsilon^c) = \epsilon^c$; and where τ^c is a control parameter with units of time, which manifests as the time scale of the decay of ϵ^c toward zero.

A similar definition of the corrective velocity u_j^c has been introduced previously, by Zhang and Haworth [27], in which the authors use $F(\epsilon^c) = \epsilon^c$, and vary the value of τ^c. The main contribution of this work in the area of explicit mass correction algorithms is in testing alternative definitions of $F(\epsilon^c)$, and establishing which are optimal for the purpose of maintaining the consistency error below the desired level ϵ_0^c, while keeping the magnitude of the corrective velocity to a minimum.

We use three different definitions of $F(\epsilon^c)$. The first, denoted $F_1(\epsilon^c)$ is the straightforward choice

$$F_1(\epsilon^c) = \epsilon^c. \tag{36}$$

In the second choice, spatial smoothing is applied to the field ϵ^c: the algorithm used is the implicit 3-point smoothing algorithm, described in Viswanathan et al. [25], in which the amount of spatial smoothing performed on fields is controlled by a parameter α, which is defined so that the amount of the resultant implicit smoothing is equivalent to explicit smoothing over α cells in each direction (in particular $\alpha = 1$ implies that no smoothing is performed). Using the operator $S^\alpha[\cdot]$ to denote the smoothing procedure, we define $F_2^\alpha(\epsilon^c)$ as

$$F_2^\alpha(\epsilon^c) = S^\alpha[\epsilon^c]. \tag{37}$$

This choice of $F_2^\alpha(\epsilon^c)$ is motivated by the reasoning that, by ignoring the highest wavenumber components of ϵ^c, the resulting correction algorithm will yield, for the same values of the control parameter τ^c, lower magnitudes of the corrective velocity u_j^c, thus allowing us to use smaller values of the decay parameter τ^c.

Finally, we also consider a definition, $F_3(\epsilon^c)$ which aims to reduce the amount of corrective velocity by only correcting PMC errors when they become significant compared to ϵ_0^c. In particular, $F_3(\epsilon^c)$ is defined by

$$F_3(\epsilon^c) = \begin{cases} 0, & \text{for } |\epsilon^c| < \epsilon_0^c/2, \\ \epsilon^c \sin^2(\frac{\pi(|\epsilon^c| - \epsilon_0^c)}{\epsilon_0^c}), & \text{for } \epsilon_0^c/2 \leqslant |\epsilon^c| < \epsilon_0^c, \\ \epsilon^c, & \text{for } |\epsilon^c| \geqslant \epsilon_0^c, \end{cases} \tag{38}$$

that is, between $|\epsilon^c| = \epsilon_0^c/2$ and $|\epsilon^c| = \epsilon_0^c$, F_3 varies smoothly between $F_3 = 0$ (for $|\epsilon^c| \leqslant \epsilon_0^c/2$) and $F_3 = \epsilon_c$ (for $|\epsilon^c| \geqslant \epsilon_0^c$).

The overall performance of the three corrective strategies is shown on Table 2. The first three rows show results for a simulation with $n_{pc} = 30$, the last three for a simulation with $n_{pc} = 50$. The table compares L^1 and L^∞ measures of the corrective velocity, respectively denoted as $\|\epsilon^U\|_1$ and $\|\epsilon^U\|_\infty$ (where ϵ^U is a dimensionless velocity error normalized by the maximal flow velocity), with the PMC errors produced by each scheme. The control parameter τ^c is set to have the largest

Table 2
Comparison between the three corrective schemes, for flamelet simulations with $n_{pc} = 30$ and $n_{pc} = 50$.

Corrective scheme	$\tau^c/\Delta t$	n_{pc}	$\|\epsilon^c\|_1$	$\|\epsilon^c\|_\infty$	$\|\epsilon^U\|_1$	$\|\epsilon^U\|_\infty$
F_1	8	30	4.4×10^{-2}	1.53×10^{-1}	9.1×10^{-2}	1.58×10^{-1}
F_3	8	30	5.1×10^{-2}	1.60×10^{-1}	8.1×10^{-2}	1.49×10^{-1}
$F_2^{\alpha=3.0}$	4	30	4.6×10^{-2}	1.52×10^{-1}	4.5×10^{-2}	9.7×10^{-2}
F_1	10	50	3.5×10^{-2}	1.29×10^{-1}	7.4×10^{-2}	1.22×10^{-1}
F_3	10	50	4.4×10^{-2}	1.31×10^{-1}	6.7×10^{-2}	1.19×10^{-1}
$F_2^{\alpha=3.0}$	6	50	3.7×10^{-2}	1.31×10^{-1}	3.8×10^{-2}	7.9×10^{-2}

Table 3
L^1 consistency and corrective velocity errors for $n_{pc} = 30$ – a parameter study for α and τ^c.

τ^c	$\|\epsilon^c\|_1\|/\|\epsilon^U\|_1$ $\alpha = 1.0$		$\|\epsilon^c\|_1\|\|\epsilon^U\|_1$ $\alpha = 2.0$		$\|\epsilon^c\|_1\|\|\epsilon^U\|_1$ $\alpha = 3.0$		$\|\epsilon^c\|_1\|\|\epsilon^U\|_1$ $\alpha = 4.0$	
2.0	0.027	0.246	0.034	0.120	0.038	0.070	0.041	0.060
4.0	0.033	0.139	0.040	0.065	0.046	0.047	0.049	0.039
6.0	0.039	0.102	0.049	0.056	0.051	0.036	0.056	0.035
8.0	0.044	0.091	0.057	0.048	0.059	0.032	0.065	0.027

possible value while still maintaining $\|\epsilon^c\|_\infty < \epsilon_0^c$ and $\|\epsilon^c\|_1 < \epsilon_1^c$, where $\epsilon_0^c = 0.171$, $\epsilon_1^c = 0.136$ and $\epsilon_0^c = 0.132$, $\epsilon_1^c = 0.105$, respectively for $n_{pc} = 30$ and $n_{pc} = 50$. Due to the large number of simulations required to explore the parameter space for τ^c and α, all of the results presented in this section are obtained from simulations with flamelet chemistry modeling.

As we can see on the table, the criteria $\|\epsilon^c\|_1 < \epsilon_1^c$, $\|\epsilon^c\|_\infty < \epsilon_0^c$ can be enforced by all three schemes; also note that, as mentioned above $\|\epsilon^c\|_\infty < \epsilon_0^c$ is the more restrictive criterion. Contrary to expectations, the F_3 corrective scheme does not yield considerable improvement over F_1: the L^1 measure of the PMC error increases by approximately 20%, while the L^1 measure of the corrective velocity decreases by approximately 10%, and the difference in the L^∞ error measures is even less.

In contrast, the F_2 scheme with $\alpha = 3.0$, and a smaller value for Δt gives considerable improvements over F_1 – the consistency errors are similar, whereas the corrective velocity errors are reduced by approximately 50% in the L^1 sense and 35% in the L^∞ sense. The choice of $\alpha = 3.0$ was arrived at by varying α from 1.0 to 4.0, and τ^c from its F_1 value down to $\tau^c = 2\Delta t$. The L^1 results of this parameter study are shown on Table 3, for $n_{pc} = 30$.

As we can see $\alpha = 3.0$, $\tau^c = 4\Delta t$ yields the least corrective velocity for the same (or less) consistency error as in the $F_1(\cdot)$ case with $\tau^c = 8\Delta t$, which is the greatest value of τ^c such that the criterion of Eq. (28) is satisfied. The results are similar for $n_{pc} = 50$ – the optimal combination is $\alpha = 3.0$, $\tau^c = 6\Delta t$. Henceforth, we shall use F_2 with these parameters as our explicit PMC correction algorithm.

Note that in a different LES/PDF simulation the value of the control parameter τ^c required to maintain $\|\epsilon^c\|_1 < \epsilon_1^c$, $\|\epsilon^c\|_\infty < \epsilon_0^c$ may be different; as the test case in Appendix A shows for example, the PMC errors are a function of the turbulent diffusivity magnitude. Therefore, when applying the $F_2^{\alpha=3.0}$ corrective scheme to a new flow, the user is encouraged to monitor the levels of $\|\epsilon^c\|_1$, $\|\epsilon^c\|_\infty$ in the initial stages of the simulation, and adjust τ^c accordingly – reduce it if the PMC errors are above the desired level, increase it if they are below the desired level.

6. Velocity and scalar interpolation schemes as implicit methods for preserving PMC

In this section, we introduce new interpolation schemes for the velocity and scalar fields in Eq. (10), which are designed to reduce PMC errors, both in terms of reducing ϵ^c for simulations without explicit PMC correction, and in terms of reducing the corrective velocity imposed by the correction algorithms discussed in the previous section. We then compare the performance of these new schemes with that of standard multilinear and fourth-order Lagrangian interpolation.

6.1. Velocity interpolation: the polar parabolic edge reconstruction method

The Polar Parabolic Edge Reconstruction Method (PPERM) is an adaptation for cylindrical grids of a previous velocity interpolation scheme, called the Parabolic Edge Reconstruction Method (PERM), introduced by McDermott and Pope [10]. As noted by the authors of that paper, the combination of the evolution equations for $\bar{\rho}$ and q, respectively Eq. (16), (17), implies that, with $\bar{\rho} = q$ at the beginning of the simulation (which is always the case, provided the particles are initialized correctly), the magnitude of the differences between $\bar{\rho}$ and q is proportional to the magnitude of the differences between $\tilde{u}_j$ and $\tilde{u}_j^*$, and between $\frac{\partial \tilde{u}_j}{\partial x_j}$ and $\frac{\partial \tilde{u}_j^*}{\partial x_j}$. Therefore, a velocity interpolation scheme must yield accurate values for the velocity $\tilde{u}_j^*$ and its divergence, $\frac{\partial \tilde{u}_j^*}{\partial x_j}$, in order to maintain PMC.

To this end, PERM, which is an extension of multilinear interpolation with additional polynomial terms, is a second-order accurate (with respect to the grid spacing) velocity interpolation scheme, with second-order accurate values for the divergence for the interpolated velocity. This is an advantage over multilinear interpolation, whose divergence of the velocity is

Table 4
Comparison of the performance of different interpolation schemes for an FC simulation with $n_{pc} = 30$, without PMC correction.

Simulation	FC, $n_{pc} = 30$, multilinear	FC, $n_{pc} = 30$, PPERM/MLG	FC, $n_{pc} = 30$, 4th order
$\|\epsilon^c\|_1$	1.16×10^{-1}	7.92×10^{-2}	9.67×10^{-2}
$\|\epsilon^c\|_\infty$	3.95×10^{-1}	2.54×10^{-1}	3.17×10^{-1}
µs/ptcl · step	84.2	91.1	102.3

Table 5
Comparison of the performance of different interpolation schemes for a flamelet simulation with $n_{pc} = 50$, without PMC correction.

Simulation	Flamelet, $n_{pc} = 50$, multilinear	Flamelet, $n_{pc} = 50$, PPERM/MLG	Flamelet, $n_{pc} = 50$, 4th order
$\|\epsilon^c\|_1$	1.03×10^{-1}	6.77×10^{-2}	8.91×10^{-2}
$\|\epsilon^c\|_\infty$	2.73×10^{-1}	1.88×10^{-1}	2.27×10^{-1}
µs/ptcl · step	13.3	19.5	29.2

only first-order accurate. An additional advantage of PERM over multilinear interpolation is that for an FV discrete velocity field $\tilde{u}_j$ which is discretely divergence-free (i.e., the total velocity flux through the faces of each FV cell is 0), the PERM interpolated velocity is divergence-free at all points in the domain.

The new interpolation scheme, PPERM, is an adaptation of PERM to cylindrical coordinate grids which retains these properties. A detailed description of PPERM, the process for its evaluation, and numerical tests of its convergence properties are given in Appendix B.

6.2. Diffusivity and density interpolation: the multilinear gradients method

The new Multilinear Gradients Method (MLG) introduced here is a third-order accurate interpolation scheme for scalar fields – in this work, it is applied to $\bar{\rho}\tilde{D}_T$ and $\bar{\rho}$ in Eq. (10). The MLG-interpolated fields are piecewise polynomial in x, r, and θ. The 20 polynomial terms used are of up to fourth degree, and comprise the minimal set necessary so that the gradient of the MLG reconstructed field has the same functional form as when the gradient is itself interpolated by a multilinear velocity interpolation scheme. This is done so that the value of $\frac{\partial(\bar{\rho}\tilde{D}_T)}{\partial x_j}$ used in Eq. (10) is second-order accurate with respect to the grid spacing.

Note that second-order accuracy of $\frac{\partial(\bar{\rho}\tilde{D}_T)}{\partial x_j}$ can also be achieved by performing multilinear interpolation of $\frac{\partial(\bar{\rho}\tilde{D}_T)}{\partial x_j}$ as a vector field, separately from the interpolation of the scalars $\bar{\rho}$, $\bar{\rho}\tilde{D}_T$. The advantages of using MLG is that the additional computational work performed in obtaining second-order accurate values for the gradient is also utilized in obtaining third-order accurate scalar fields, and that the interpolated value for $\frac{\partial(\bar{\rho}\tilde{D}_T)}{\partial x_j}$ is in fact the gradient of the interpolation of $\bar{\rho}\tilde{D}_T$. A detailed description of the functional form of MLG, the process for its evaluation, and numerical tests of its convergence properties are given in Appendix C.

6.3. PMC performance of PPERM/MLG in an LES/PDF context

In this subsection, we compare the PMC performance of the combination of PPERM and MLG interpolation schemes with that of standard multilinear and fourth-order Lagrangian interpolation. First we consider simulations without explicit PMC correction. Tables 4 and 5 present results from two types of simulations, respectively an FC simulation with $n_{pc} = 30$ and a flamelet model simulation with $n_{pc} = 50$. As mentioned previously, the SDE integration scheme used is the second-order KP scheme.

As we can see on both tables, using fourth-order accurate Lagrangian interpolation provides an advantage over standard multilinear interpolation, and PPERM/MLG provides an even greater advantage – L^1 consistency errors are decreased by approximately 30%, and L^∞ consistency errors are decreased by approximately 35%, relative to the simulation with multilinear interpolation. Furthermore, the added computational cost of using PPERM/MLG is lower than that of using fourth-order Lagrangian interpolation, and lower than 10% of the overall computational cost for a simulation with full chemistry modeling using the FC mechanism. For flamelet calculations, the added computational cost of using the higher-order interpolation schemes is considerable, but the FC test case is more representative of a typical LES/PDF simulation.

Note that, even with fourth-order Lagrangian and PPERM/MLG interpolation, we still need to perform explicit PMC correction in order to meet the target values for ϵ_0^c – 0.171 and 0.132 for $n_{pc} = 30$ and $n_{pc} = 50$, respectively. For this reason, we perform another set of simulations with the optimal explicit corrective scheme from Section 5: $F(\epsilon^c) = F_2^{\alpha=3.0}(\epsilon^c)$, with $\tau^c = 4\Delta t$ and $\tau^c = 6\Delta t$ for $n_{pc} = 30$ and $n_{pc} = 50$, respectively – since this choice of parameters yielded $\|\epsilon^c\|_\infty < \epsilon_0^c$ in Section 5, where multilinear interpolation was used, it is to be expected that for PPERM/MLG and fourth-order Lagrangian interpolation the resulting consistency errors will be even lower.

Tables 6 and 7 present the simulation results, again for FC simulations with $n_{pc} = 30$ and flamelet simulations with $n_{pc} = 50$. Time histories of the L^1 and L^∞ norms of the corrective velocities are also shown on Fig. 9, on which we can

Table 6
Comparison of the performance of different interpolation schemes for an FC simulation with $n_{pc} = 30$, with PMC correction.

Simulation	FC, $n_{pc} = 30$, multilinear	FC, $n_{pc} = 30$, PPERM/MLG	FC, $n_{pc} = 30$, 4th order
$\Vert\epsilon^c\Vert_1$	4.8×10^{-2}	4.0×10^{-2}	4.3×10^{-2}
$\Vert\epsilon^c\Vert_\infty$	1.55×10^{-1}	1.20×10^{-1}	1.33×10^{-1}
$\Vert\epsilon^U\Vert_1$	4.7×10^{-2}	2.6×10^{-2}	3.6×10^{-2}
$\Vert\epsilon^U\Vert_\infty$	9.6×10^{-2}	4.2×10^{-2}	6.5×10^{-2}

Table 7
Comparison of the performance of different interpolation schemes for a flamelet simulation with $n_{pc} = 50$, with PMC correction.

Simulation	Flamelet, $n_{pc} = 50$, multilinear	Flamelet, $n_{pc} = 50$, PPERM/MLG	Flamelet, $n_{pc} = 50$, 4th order
$\Vert\epsilon^c\Vert_1$	3.7×10^{-2}	2.9×10^{-1}	3.2×10^{-2}
$\Vert\epsilon^c\Vert_\infty$	1.31×10^{-1}	1.06×10^{-1}	1.09×10^{-1}
$\Vert\epsilon^U\Vert_1$	3.8×10^{-2}	2.2×10^{-2}	2.9×10^{-2}
$\Vert\epsilon^U\Vert_\infty$	7.9×10^{-2}	3.8×10^{-2}	6.0×10^{-2}

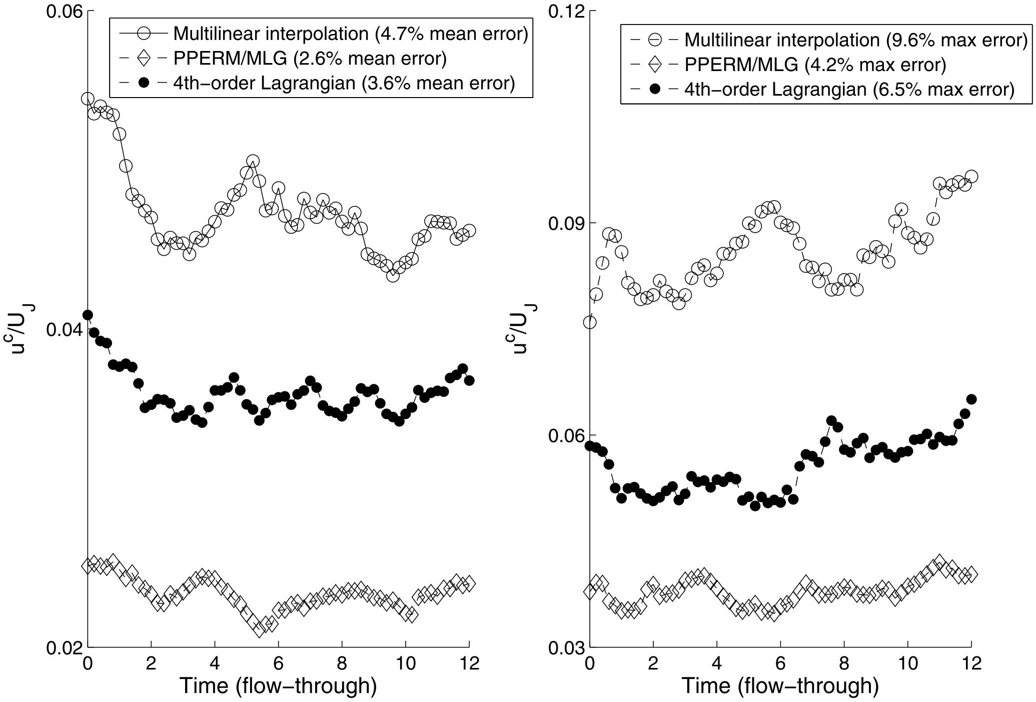

Fig. 9. Time histories of the corrective velocity for an FC PPERM/MLG simulation with $n_pc = 30$, and an $F_3^{\alpha=3.0}(\epsilon^c)$ correction scheme with $\tau^c = 4\Delta t$. Left: L^1 measures of the corrective velocity. Right: L^∞ measures of the corrective velocity.

see that while there is some fluctuation of the corrective velocity magnitudes over the simulation time span, the overall magnitude of the fields does not change considerably (the fluctuation is on the order of 20% of the mean value). Finally, Fig. 10 displays instantaneous contour plots of the consistency error and axial component of the corrective velocity at the end of the FC, PPERM/MLG simulation. We can see on this figure that the PMC error field has length scales much smaller than those of the corrective velocity, which is to be expected due to the smoothing in the PMC corrective scheme.

As we can see, with the addition of PMC correction, the PPERM/MLG simulations again give a reduction in the consistency errors, though not as great as in Tables 4 and 5 – for example, PPERM/MLG interpolation yields a reduction of approximately 20% in $\Vert\epsilon^c\Vert_\infty$ over multilinear interpolation, compared with the 35% reduction obtained previously. However, we also see a considerable decrease in corrective velocities when PPERM/MLG and fourth-order Lagrangian interpolation are used – based on the L^1 measures of corrective velocity, PPERM/MLG reduces the corrective velocities by approximately 45%, and fourth-order Lagrangian interpolation reduces them by approximately 25%.

Based on these tests, we conclude that, from a PMC standpoint, the PPERM/MLG interpolation schemes have an advantage over both multilinear and fourth-order Lagrangian interpolation. Moreover, we note that the overall performance of the PPERM/MLG schemes with KP SDE integration, and PMC correction enforced by the optimal scheme from Section 5 is quite

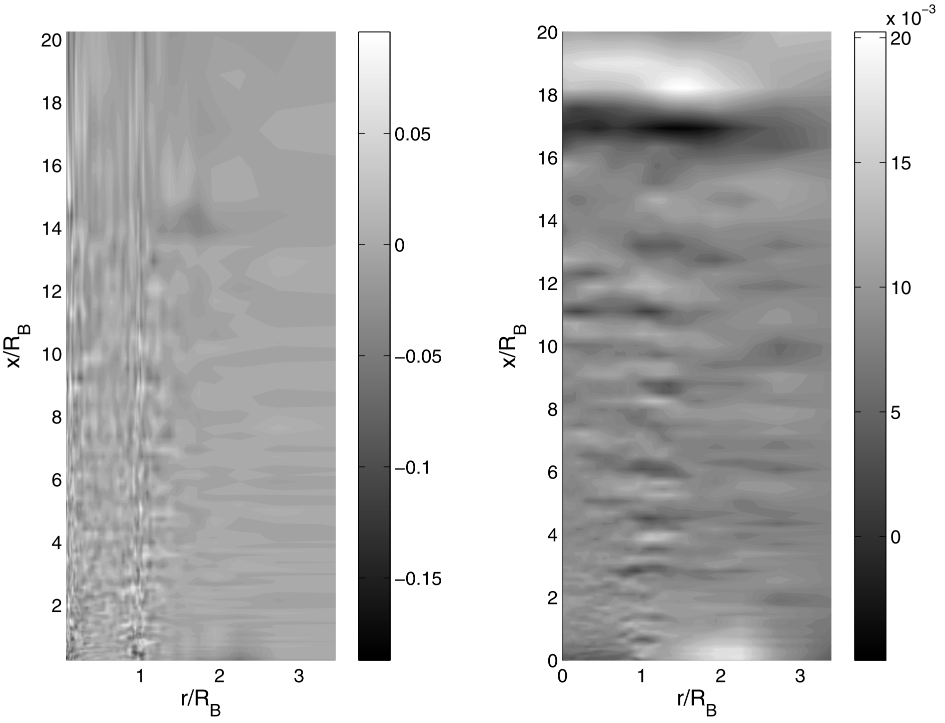

Fig. 10. Contour plots, at the azimuthal location $\theta = 0$, of consistency error and corrective velocities for an FC PPERM/MLG simulation with $n_{pc} = 30$, and an $F_3^{\alpha=3.0}(\epsilon^c)$ correction scheme with $\tau^c = 4\Delta t$. Left: consistency error at the end of the simulation. Right: axial component of the normalized corrective velocity at the end of the simulation.

satisfactory – the desired PMC criteria are achieved with corrective velocities of approximately 2.5% relative magnitude in the L^1 sense, and approximately 4% relative magnitude in the L^∞ sense.

7. Conclusions

This paper addresses the issue of particle mass consistency (PMC) in hybrid LES/PDF methods for turbulent reactive flows. The particle mass consistency error is defined as the departure from one of the ratio between particle mass density and LES resolved density. A set of new implicit and explicit PMC preservation and correction schemes is introduced. A second-order SDE integration scheme, in itself an implicit PMC preservation scheme when compared to Euler SDE integration, is shown to decrease the instantaneous PMC error of the simulation by 30% and the time-averaged, deterministic PMC error by 45%. Explicit PMC correction is optimized by employing a smoothed version of the consistency error field; compared to a standard explicit PMC scheme, the variant using the smoothed consistency error field decreases the corrective velocity necessary for attaining the desired accuracy criterion by 35% in the L^∞ sense and by 50% in the L^1 sense. The implicit schemes also include new velocity and scalar interpolation algorithms designed for this issue, which yield an overall reduction of up to 45% in the corrective velocity necessary for maintaining PMC, relative to standard multilinear interpolation, and which moreover perform better than the standard 4th order Lagrangian interpolation scheme. Using the optimal explicit and implicit schemes, the overall performance of the simulation is shown to be quite satisfactory, meeting the desired consistency criterion while using a corrective velocity of relatively low magnitude.

Acknowledgements

This work was supported in part by NSF Award No. CBET-1033246 and in part by Grant Number FA9550-09-1-0611 funded by the National Center for Hypersonic Combined Cycle Propulsion, sponsored by the AFOSR and NASA ARMD.

Appendix A. A model for the development of mass consistency errors in the shear layer

In this section of the appendix, we describe a simple 2D analytic test case which reproduces the deterministic (time-averaged) PMC errors observed in Section 4. Particles of equal mass are initialized uniformly on the domain $x \in [0, 1] \times y \in [0, 1]$, with a density of 10^6 particles per unit area, and evolve by the SDE

$$dX^* = \frac{\partial D_T}{\partial x}\left(X^*\right)dt + D_T\left(X^*\right)dW_1^*, \tag{39}$$

$$dY^* = V\left(X^*\right)dt + D_T\left(X^*\right)dW_2^*, \tag{40}$$

for

$$V(x) = 3\left(0.1 + erf\left((x-0.5)/0.05\right) - erf\left((x-0.85)/0.05\right)\right) \tag{41}$$

and

$$D_T(x) = 0.02 + exp\left(-\left((x-0.5)/0.05\right)^2\right) \tag{42}$$

which, from Eq. (17) implies that $\frac{D^*q}{Dt^*} = 0$, hence in the absence of numerical errors the particle distribution should remain uniform. The velocity and diffusivity fields defined in Eqs. (41), (42), which are plotted on Fig. 7, are qualitatively similar to the velocity and drift fields in the bluff-body flame: there is a shear layer at $x = 0.5$, in which region the diffusivity is considerably higher than everywhere else in the domain. Another shear layer is located at $x = 0.85$, for the sake of having the fields be approximately periodic in x (with a negligible discontinuity at $x = 1$), but we shall see that this shear layer does not influence the particle distribution as much, because it is located in a region of low diffusivity.

All particles are advected by either the Euler or KP SDE time integration schemes, with a time step of $\Delta t = 0.0125 \times p$, for $400/p$ time steps, where $p = 1, 1/2, 1/4, 1/8$ is a time step refinement parameter. A periodic boundary condition in the x-direction is enforced. In the y-direction, we impose inflow boundary conditions by deleting, after each time step, all particles in the region $y \in [0, 0.3] \cup [0.7, 1.0]$, and initializing new particles in that region, with a uniform density of 10^6 particles per unit area. Thus, the inlets are the lines $y = 0.3$ and $y = 0.7$.

The results for $p = 1$ are plotted on the right-hand side of Fig. 7. The figure shows values of ϵ^c, averaged over time and over the y-coordinate, from $y = 0.3$ to $y = 0.45$, plotted against x. We see a region of increased particle density (positive ϵ^c) at the low-velocity part of the shear layer, $0.0 < x < 0.4$, and a region of decreased particle density at the high-velocity part of the shear layer, $0.4 < 0 < 0.7$. Qualitatively, this agrees well with the deterministic PMC error fields observed in Section 4. Fig. 8 plots values of $\max(\epsilon^c)$ against the time step. As can be expected, we see that $\max(\epsilon^c)$ converges in a second-order fashion for the KP scheme, and in a first-order fashion for the Euler scheme.

This test case demonstrates that considerable deterministic PMC errors can develop even when the only source of numerical error is the error due to the SDE integration scheme. It also illustrates the advantage of using a higher-order SDE integration scheme: even for the largest time steps, the PMC error is reduced by a factor of 3 when the second-order KP scheme is used instead of the Euler scheme.

Appendix B. The PPERM velocity interpolation scheme

In this section of the appendix, we describe in detail the functional form of the PPERM velocity interpolation scheme, and the algorithm for its evaluation. Consider a given cell $x \in (x_0, x_1)$, $r \in (r_0, r_1)$, $\theta \in (\theta_0, \theta_1)$ in cylindrical coordinates, and denote $\Delta x = x_1 - x_0$, $\Delta r = r_1 - r_0$, $\Delta\theta = \theta_1 - \theta_0$.

Here, we use the subscript index 1 to denote a positive cell face (e.g., $\{(x, r, \theta) \mid x = x_1\}$) and the subscript index 0 to denote a negative cell face (e.g., $\{(x, r, \theta) \mid x = x_0\}$). We shall also use subscript index pairs to denote cell edges (e.g., for a radial edge, the subscript index 01 denotes $\{(x, r, \theta) \mid x = x_0,\ \theta = \theta_1\}$), and subscript index triples to denote cell vertices (e.g., for a radial edge, the subscript index 101 denotes $\{x = x_1,\ r = r_0,\ \theta = \theta_1\}$): in addition to the 8 cell vertices, we shall also use the subscripts $i\frac{1}{2}k$, for $i, k \in \{0, 1\}$ to denote the midpoints of the four radial cell edges. This is illustrated on Fig. 13.

Introducing the local coordinates

$$a_1 = \frac{x - x_0}{x_1 - x_0}, \qquad b_1 = \frac{r - r_0}{r_1 - r_0}, \qquad c_1 = \frac{\theta - \theta_0}{\theta_1 - \theta_0},$$
$$a_0 = 1 - a_1, \qquad b_0 = 1 - b_1, \qquad c_0 = 1 - c_1, \tag{43}$$

and the coefficients $U_{jk}^{[l]}$, $V_{ik}^{[m]}$, $W_{ij}^{[n]}$, for $i, j, k \in \{0, 1\}$, $l, n \in \{0, 1, 2\}$, and $m \in \{0, 1, 2, 3\}$ (where the superscripts correspond to the degree of the respective polynomial term), we define the PPERM-interpolated velocity $I(\mathbf{u})(x, r, \theta)$ in cylindrical coordinates as

$$I(\mathbf{u})_x = \sum_{j,k=0}^{1} b_j c_k \left(U_{jk}^{[0]} + (a_1 - 0.5)U_{jk}^{[1]} + \left((a_1 - 0.5)^2 - 0.25\right)U_{jk}^{[2]}\right), \tag{44}$$

$$rI(\mathbf{u})_r = \sum_{i,k=0}^{1} a_i c_k \left(V_{jk}^{[0]} + (b_1 - 0.5)V_{jk}^{[1]} + \left((b_1 - 0.5)^2 - 0.25\right)V_{jk}^{[2]} + b_1(b_1 - 0.5)(b_1 - 1)V_{jk}^{[3]}\right), \tag{45}$$

$$I(\mathbf{u})_\theta = \sum_{i,j=0}^{1} a_i b_j \big(W_{ij}^{[0]} + (c_1 - 0.5)W_{ij}^{[1]} + \big((c_1 - 0.5)^2 - 0.25\big)W_{ij}^{[2]}\big). \tag{46}$$

Note that instead of interpolating for u_r, we interpolate for ru_r – this is done so that the divergence formula, which has the cylindrical coordinate form

$$r(\nabla\cdot\mathbf{u}) = r\frac{\partial u_x}{\partial x} + \frac{\partial r u_r}{\partial r} + \frac{\partial u_\theta}{\partial\theta}, \tag{47}$$

yields the following result for the interpolated divergence:

$$\begin{aligned} r\big(\nabla\cdot I(\mathbf{u})\big) = {} & r\sum_{j,k=0}^{1} b_j c_k \big(U_{jk}^{[1]} + 2(a_1 - 0.5)U_{jk}^{[2]}\big)/\Delta x \\ & + \sum_{i,k=0}^{1} a_i c_k \big(V_{jk}^{[1]} + 2(b_1 - 0.5)V_{jk}^{[2]} + \big(3(b_1)^2 - 3b_1 + 0.5\big)V_{jk}^{[3]}\big)/\Delta r \\ & + \sum_{i,j=0}^{1} a_i b_j \big(W_{ij}^{[1]} + (2c_1 - 0.5)W_{ij}^{[2]}\big)/\Delta\theta. \end{aligned} \tag{48}$$

Also note that we are using an additional 3rd degree polynomial in the interpolation of ru_r. We do this so that we can get $r(\nabla\cdot I(\mathbf{u}))$ to be quadratic in b and linear in a, c – this enables us to obtain 3rd order accurate values of $r(\nabla\cdot I(\mathbf{u}))$ near the centerline (i.e., in the limit as $r\downarrow 0$), which is needed for 2nd order accurate values of $r(\nabla\cdot I(\mathbf{u}))$ near the centerline.

Next, we describe the algorithm for the evaluation of the coefficients $U_{jk}^{[l]}, V_{ik}^{[m]}, W_{ij}^{[n]}$.

1. **Interpolation of FV velocity and divergence onto cell corners and radial edge midpoints:** Using standard linear interpolation, evaluate second-order accurate approximations of u_x, u_r, u_θ at the corners of the present cell and its neighbors: denote these as $u_{x,ijk}, u_{r,ijk}, u_{\theta,ijk}$ respectively, for $i, j, k \in \{-1, 0, 1, 2\}$. Also, evaluate second-order accurate approximations of $(\nabla\cdot\mathbf{u})$ at the 8 cell corners and 4 radial edge midpoints of the present cell: denote these as d_{ijk}, for $i, k \in \{0, 1\}$, $j \in \{0, \frac{1}{2}, 1\}$.
2. **Get provisional values for the 0th and 1st order PPERM coefficients:** Set

   $$U_{jk}^{[1]} = u_{x,1jk} - u_{x,0jk}, \qquad U_{jk}^{[1]} = (u_{x,1jk} + u_{x,0jk})/2, \tag{49}$$

   $$V_{ik}^{[1]} = r_1 u_{r,i1k} - r_0 u_{r,i0k}, \qquad V_{ik}^{[1]} = (r_1 u_{r,i1k} + r_0 u_{r,i0k})/2, \tag{50}$$

   $$W_{ij}^{[1]} = u_{\theta,ij1} - u_{\theta,ij0}, \qquad W_{jk}^{[1]} = (u_{\theta,ij1} + u_{\theta,ij0})/2. \tag{51}$$

   Since the second and third degree PPERM terms vanish at the cell corners, Eqs. (7)–(9) ensure that at this point in the interpolation process the corner values of u_x, u_r, u_θ are exactly equal to the second-order accurate approximations from Step 1.
3. **Flux correction for the 0th and 1st order PPERM coefficients:** For the 6 cell faces, subtract the face averages of the interpolated velocity, implied by the coefficients obtained in Step 2, from the FV face-average velocities. Denote these differences respectively $\Delta u_{x,1}$ and $\Delta u_{x,0}$ for the positive and negative x-faces, $\Delta u_{r,1}$ and $\Delta u_{r,0}$ for the positive and negative r-faces, and $\Delta u_{\theta,1}$ and $\Delta u_{\theta,0}$ for the positive and negative θ-faces. Using $\tilde{u}_{x,i}, \tilde{u}_{r,i}, \tilde{u}_{\theta,i}$, for $\imath \in \{0, 1\}$ to denote the FV face average, this calculation takes the form

   $$\lambda_1 = \frac{(1/2)2r_0 + (1/6)(r_1 - r_0)}{r_0 + r_1}, \qquad \lambda_0 = 1 - \lambda_1, \tag{52}$$

   $$\Delta u_{x,i} = \tilde{u}_{x,i} - \sum_{j,k=0}^{1} \lambda_j \big(U_{jk}^{[0]} + (i - 0.5)U_{jk}^{[1]}\big)/2, \tag{53}$$

   $$\Delta u_{r,j} = \tilde{u}_{r,j} - \sum_{i,k=0}^{1} \big(V_{ik}^{[0]} + (j - 0.5)V_{ik}^{[1]}\big)/(4r_j), \tag{54}$$

   $$\Delta u_{\theta,k} = \tilde{u}_{\theta,k} - \sum_{i,j=0}^{1} \big(W_{ij}^{[0]} + (k - 0.5)W_{ij}^{[1]}\big)/4. \tag{55}$$

   Note that this calculation can be performed at this stage, as the velocity flux through the cell faces depends only on the 0th and 1st order PPERM coefficients. Next, perform the corrections:

$$U_{jk}^{[0]} \leftarrow U_{jk}^{[0]} + (\Delta u_{x,1} + \Delta u_{x,0})/2, \qquad U_{jk}^{[1]} = U_{jk}^{[1]} + (\Delta u_{x,1} - \Delta u_{x,0}), \tag{56}$$

$$V_{ik}^{[0]} \leftarrow V_{ik}^{[0]} + (r_1 \Delta u_{r,1} + r_0 \Delta u_{r,0})/2, \qquad V_{ik}^{[1]} = V_{ik}^{[1]} + (r_1 \Delta u_{r,1} - r_0 \Delta u_{r,0}), \tag{57}$$

$$W_{ij}^{[0]} \leftarrow W_{ij}^{[0]} + (\Delta u_{\theta,1} + \Delta u_{\theta,0})/2, \qquad W_{ij}^{[1]} = W_{ij}^{[1]} + (\Delta u_{\theta,1} - \Delta u_{\theta,0}). \tag{58}$$

This ensures that the face averages of the interpolated velocity are identically equal to the FV face-average velocities, and hence that the total flux of interpolated velocity through the cell is equal to the FV cell flux.

4. **First estimate for the 2nd and 3rd order PPERM coefficients:** From the cell corner values of velocity obtained in Step 1, calculate 4-point approximations of the second derivatives of u_x, ru_r, u_θ along the cell edges parallel to the respective velocity component, and set $U_{jk}^{[2]}, V_{ik}^{[2]}, W_{ij}^{[2]}$ so that the edge-averages of second derivatives of the interpolated velocity match the approximated FV second derivatives. For a uniform grid, this calculation reduces to

$$U_{jk}^{[2]} = \frac{1}{4}(u_{x,2jk} - u_{x,1jk} - u_{x,0jk} + u_{x,-1jk}), \tag{59}$$

$$V_{ik}^{[2]} = \frac{1}{4}\big((r_1 + \Delta r)u_{r,i2k} - r_1 u_{r,i1k} - r_0 u_{r,i0k} + (r_0 - \Delta r)u_{r,i-1k}\big), \tag{60}$$

$$W_{ij}^{[2]} = \frac{1}{4}(u_{\theta,ij2} - u_{\theta,ij1} - u_{\theta,ij0} + u_{\theta,ij-1}). \tag{61}$$

Next, calculate 4-point approximations of the third derivatives of ru_r along the radial cell edges, and set $V_{ik}^{[3]}$ so that the edge-averages of third derivatives of the interpolated velocity match the approximated FV second derivatives. For a uniform grid, this calculation reduces to

$$V_{ik}^{[3]} = \frac{1}{6}\big((r_1 + \Delta r)u_{r,i2k} - 3r_1 u_{r,i1k} + 3r_0 u_{r,i0k} - (r_0 - \Delta r)u_{r,i-1k}\big). \tag{62}$$

5. **Divergence correction for the 2nd and 3rd order PPERM coefficients:** Subtract from the second-order approximations of the divergence, d_{ijk}, calculated in Step 1, the divergence of the interpolated velocity, with the present PPERM coefficients, denoted by $[\nabla \cdot I(\mathbf{u})]_{ijk}$, at the respective locations. Denote the resulting differences as δd_{ijk}, for $i, j \in \{0, 1\}$, $j \in \{0, \frac{1}{2}, 1\}$. Let A be the 12×16 matrix (calculated from Eqs. (44)–(46)) which relates a change in the 16-component vector $(U_{jk}^{[2]}, V_{ik}^{[2]}, V_{ik}^{[3]}, W_{ij}^{[2]})$ to a change in the 12-component vector $[\nabla \cdot I(\mathbf{u})]_{ijk}$:

$$\delta\big[\nabla \cdot I(\mathbf{u})\big]_{ijk} = A\big(\delta U_{jk}^{[2]}, \delta V_{ik}^{[2]}, \delta V_{ik}^{[3]}, \delta W_{ij}^{[2]}\big). \tag{63}$$

Correct the 2nd and 3rd order PPERM coefficients by setting

$$U_{jk}^{[2]} \leftarrow U_{jk}^{[2]} + \delta U_{jk}^{[2]}, \tag{64}$$

$$V_{ik}^{[2]} \leftarrow V_{ik}^{[2]} + \delta V_{ik}^{[2]}, \qquad V_{ik}^{[3]} = V_{ik}^{[3]} + \delta V_{ik}^{[3]}, \tag{65}$$

$$W_{ij}^{[2]} \leftarrow W_{ij}^{[2]} + \delta W_{ij}^{[2]}, \tag{66}$$

where the 16-component vector $(\delta U_{jk}^{[2]}, \delta V_{ik}^{[2]}, \delta V_{ik}^{[3]}, \delta W_{ij}^{[2]})$ is the least squares-minimal norm (LSMN) solution of the linear system

$$A\big(\delta U_{jk}^{[2]}, \delta V_{ik}^{[2]}, \delta V_{ik}^{[3]}, \delta W_{ij}^{[2]}\big) = \delta d_{ijk}. \tag{67}$$

This concludes the PPERM evaluation procedure. We note that the 12×16 matrix A has rank 11, and its kernel corresponds to the constraint

$$\sum_{i,k=0}^{1} \frac{1}{r_0}\delta\big[\nabla \cdot I(\mathbf{u})\big]_{i0k} + \frac{4}{r_0 + \Delta r/2}\delta\big[\nabla \cdot I(\mathbf{u})\big]_{i\frac{1}{2}k} + \frac{1}{r_1}\delta\big[\nabla \cdot I(\mathbf{u})\big]_{i1k} = 0, \tag{68}$$

or equivalently, to the fact that altering the 2nd and 3rd degree PPERM coefficients does not change the total velocity flux through the cell.

For a discretely divergence-free FV velocity field, the correction procedure of Step 3 ensures that the total flux through the cell is zero. Due to the fact that $r(\nabla \cdot I(\mathbf{u}))$ is linear in a, c and quadratic in b by design (from Eqs. (44)–(47)), this implies that

$$\sum_{i,k=0}^{1} \frac{1}{r_0}\delta\big[\nabla \cdot I(\mathbf{u})\big]_{i0k} + \frac{4}{r_0 + \Delta r/2}\delta\big[\nabla \cdot I(\mathbf{u})\big]_{i\frac{1}{2}k} + \frac{1}{r_1}\delta\big[\nabla \cdot I(\mathbf{u})\big]_{i1k} = 0, \tag{69}$$

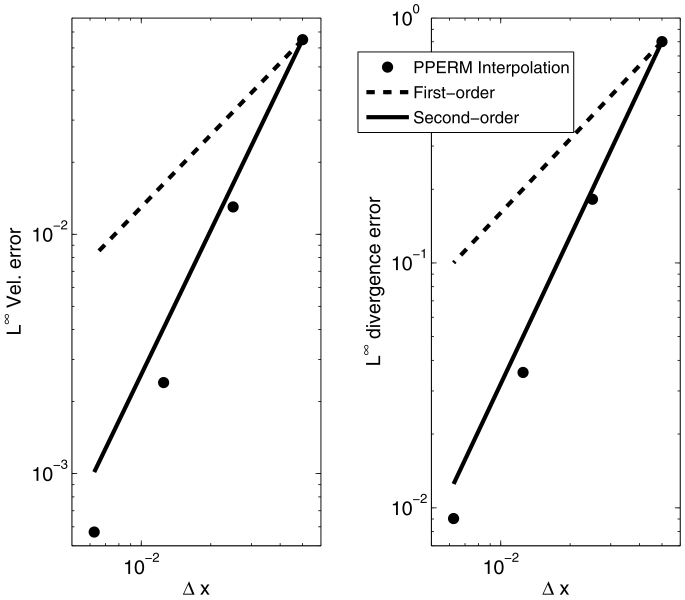

Fig. 11. Convergence plots for the PPERM interpolation scheme when applied to the analytic test flow described at the end of Appendix B. Left: L^{∞} errors for the interpolated velocity. Right: L^{∞} errors for the divergence of the interpolated velocity.

and so for this case the LSMN solution of Eq. (67) is an exact solution. Therefore, for a discretely divergence-free FV velocity field, the divergence of the interpolated velocity is zero at the cell corners and the midpoints of the 4 radial cell edges. Again, since $r(\nabla \cdot I(\mathbf{u}))$ is linear in a, c and quadratic in b, $r(\nabla \cdot I(\mathbf{u})) = 0$ at the above-mentioned locations implies that $r(\nabla \cdot I(\mathbf{u})) = 0$ everywhere in the cell, which establishes the divergence-free property of the PPERM-interpolated velocity, when applied to a discretely divergence-free FV velocity field.

A numerical test of the convergence properties of PPERM is performed on the cylindrical domain $x \in [0,1] \times r \in [0,1] \times \theta \in [0,2\pi]$ discretized on an $N \times N \times 2N$ cylindrical grid, so that $\Delta x = 1/N$. An analytic velocity field, with non-zero divergence, is specified in Cartesian coordinates:

$$\begin{aligned}
u_x(x,y,z) &= \sin\bigl(2\pi(x-0.57)\bigr)\cos\bigl(2\pi(y+0.23)\bigr),\\
u_y(x,y,z) &= -\cos\bigl(2\pi(x-0.57)\bigr)\sin\bigl(2\pi(y+0.23)\bigr),\\
u_z(x,y,z) &= -\cos\bigl(2\pi(z+x+0.57)\bigr)\sin\bigl(2\pi(y+z+0.23)\bigr).
\end{aligned} \tag{70}$$

Fig. 11 is a log–log plot of the L^{∞} velocity and divergence errors, plotted against Δx. As we can see on the figure, both the velocity and divergence errors fall closely to a line of slope 2, which verified the second-order spatial convergence of the PPERM velocity and divergence.

To verify the divergence-free property for discretely divergence-free FV fields, we perform a separate test simulation, in which we make the modification $u_z = 0$ in Eq. (70), which yields a divergence-free velocity field. Then, for the $20 \times 20 \times 40$ cylindrical grid, the maximal divergence in the domain has magnitude 1.37×10^{-13}, which can be attributed to roundoff error.

Appendix C. The MLG scalar interpolation scheme

In this appendix, we present the functional form of the MLG scalar interpolation scheme. Similarly to Appendix B, consider a given cell $x \in [x_0, x_1]$, $r \in [r_0, r_1]$, $\theta \in [\theta_0, \theta_1]$ in cylindrical coordinates, and denote $\Delta x = x_1 - x_0$, $\Delta r = r_1 - r_0$, $\Delta\theta = \theta_1 - \theta_0$. Define

$$x_C = \frac{x_0 + x_1}{2}, \qquad r_C = \frac{(r_0)^2 + (r_1)^2}{r_0 + r_1}, \qquad \theta_C = \frac{\theta_0 + \theta_1}{2}, \tag{71}$$

and set the local coordinates a, b, c so that

$$a = x - x_C, \qquad b = r - r_C, \qquad c = \theta - \theta_C. \tag{72}$$

Then, the MLG scalar interpolation of a scalar φ has the functional form of a truncated Taylor series in (x, r, θ) about the point (x_C, r_C, θ_C):

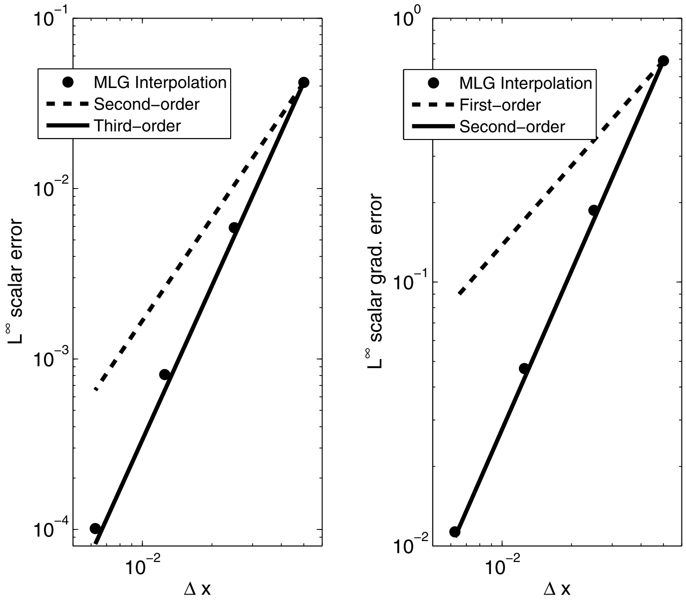

Fig. 12. Convergence plots for the MLG interpolation scheme when applied to the analytic test flow described at the end of Appendix C. Left: L^∞ errors for the interpolated scalar. Right: L^∞ errors for the gradient of the interpolated scalar.

$$\begin{aligned} I(\varphi) = {} & \Phi + abc(\Phi_{abc} + a\Phi_{a^2bc} + b\Phi_{ab^2c} + c\Phi_{abc^2}) \\ & + a\big(\Phi_a + b\Phi_{ab} + a\Phi_{a^2} + ab\Phi_{a^2b} + b^2\Phi_{ab^2}\big) \\ & + b\big(\Phi_b + c\Phi_{bc} + b\Phi_{b^2} + bc\Phi_{b^2c} + c^2\Phi_{bc^2}\big) \\ & + c\big(\Phi_c + a\Phi_{ac} + c\Phi_{c^2} + ac\Phi_{ac^2} + a^2\Phi_{a^2c}\big), \end{aligned} \tag{73}$$

where the quantities Φ are scalar coefficients. Using cell-average FV values of φ from the current cell and its closest neighbors, we set Φ_0 to equal the cell average of φ in the current cell, which ensures that the cell average of the interpolated $I(\varphi)$ equals the FV cell average, and we determine all other Φ coefficients by numerically approximating the respective derivative of φ at (x_C, r_C, θ_C).

The truncated Taylor series of Eq. (73) contains all first- and second-order terms of the full Taylor series expansion, and hence the interpolated $I(\varphi)$ is third-order accurate with respect to the grid spacing. Moreover,

$$\begin{aligned} \frac{\partial(I(\varphi))}{\partial a} = {} & bc(\Phi_{abc} + 2a\Phi_{a^2bc} + b\Phi_{ab^2c} + c\Phi_{abc^2}) + \Phi_a + b\Phi_{ab} + a\Phi_{a^2} \\ & + b^2\Phi_{ab^2} + 2ab\Phi_{a^2b} + c\Phi_{ac} + c^2\Phi_{ac^2} + 2ac\Phi_{a^2c}, \end{aligned} \tag{74}$$

which contains all first-order terms in the Taylor series expansion of $\frac{\partial(I(\varphi))}{\partial a}$, and so $\frac{\partial(I(\varphi))}{\partial a}$, and analogously $\frac{\partial(I(\varphi))}{\partial b}$ and $\frac{\partial(I(\varphi))}{\partial c}$, are second-order accurate with respect to the grid spacing.

Similarly to Appendix B, a numerical test of the convergence properties of MLG is performed on the cylindrical domain $x \in [0,1] \times r \in [0,1] \times \theta \in [0,2\pi]$ discretized on an $N \times N \times 2N$ cylindrical grid, so that $\Delta x = 1/N$. The analytic scalar field is specified in Cartesian coordinates:

$$\varphi(x,y,z) = 0.5 + 0.35\sin\big(2\pi(x+y+z+t)\big)\cos\big(2\pi(x-y+z-t)\big). \tag{75}$$

Fig. 12 is a log–log plot of the L^∞ scalar and scalar gradient errors, plotted against Δx. As we can see on the figure, the scalar errors fall closely to a line of slope 3, whereas the scalar gradient errors fall closely to a line of slope 2, which verifies MLG's convergence properties.

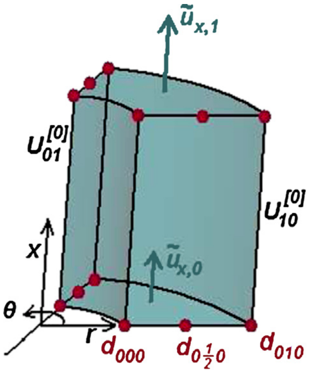

Fig. 13. Illustration of the PPERM indexing notation. Single-subscript quantities, color-coded in teal, denote face-averaged information such as the face-average velocities $\tilde{u}_{x,0}$ and $\tilde{u}_{x,1}$ shown on the schematic. Double-subscript quantities, color-coded in black, denote PPERM edge coefficients, such as the 0th order axial velocity coefficients $U_{10}^{[0]}$ and $U_{01}^{[0]}$ shown on the schematic. Triple-subscript quantities, color-coded in crimson, denote pointwise information at the 8 corners of the cell plus the 4 midpoints of the radial edges. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

References

[1] P. Colucci, F. Jaberi, P. Givi, S.B. Pope, Filtered density function for large eddy simulation of turbulent reacting flows, Phys. Fluids 10 (1998) 499–515.

[2] R.O. Fox, Computational Models for Turbulent Reactive Flows, Cambridge University Press, 2003.

[3] Y. Ge, M.J. Cleary, A.Y. Klimenko, Sparse-Lagrangian FDF simulations of Sandia Flame E with density coupling, Proc. Combust. Inst. 33 (2011).

[4] D.C. Haworth, Progress in probability density function methods for turbulent reacting flows, Prog. Energy Combust. Sci. 36 (2) (2010) 168–259.

[5] F. Jaberi, P. Colucci, S. James, P. Givi, S.B. Pope, Filtered mass density function for large eddy simulation of turbulent reacting flows, J. Fluid Mech. 401 (1999) 85–121.

[6] P. Jenny, S.B. Pope, M. Muradoglu, D.A. Caughey, A hybrid algorithm for the joint PDF equation of turbulent reactive flows, J. Comput. Phys. 166 (2001) 218–252.

[7] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer-Verlag, Berlin, 1992.

[8] A.R. Masri, R.W. Bilger, Turbulent diffusion flames of hydrocarbon fuels stabilized on a bluff body, Proc. Combust. Inst. 20 (1985).

[9] R. McDermott, S.B. Pope, A particle formulation for treating differential diffusion in filtered density function methods, J. Comput. Phys. 226 (2007) 947–993.

[10] R. McDermott, S.B. Pope, The parabolic edge reconstruction method (PERM) for Lagrangian particle advection, J. Comput. Phys. 227 (2008) 5447–5491.

[11] M. Muradoglu, S.B. Pope, D.A. Caughey, The hybrid method for the PDF equations of turbulent reactive flows: Consistency conditions and correction algorithms, J. Comput. Phys. 172 (2001) 841–878.

[12] R. Mustata, L. Valiño, C. Jiménez, W.P. Jones, S. Bondi, A probability density function Eulerian Monte Carlo field method for large eddy simulations: Applications to a turbulent piloted methane/air diffusion flame (Sandia D), Combust. Flame 145 (2006) 88–104.

[13] C.D. Pierce, P. Moin, Progress-variable approach for large-eddy simulation of turbulent combustion, J. Fluid Mech. 504 (2004) 73–97.

[14] H. Pitsch, Large-eddy simulation of turbulent combustion, Annu. Rev. Fluid Mech. 38 (2006) 453–482.

[15] S.B. Pope, A Monte Carlo method for the PDF equations of turbulent reactive flow, Combust. Sci. Technol. 25 (1981) 159–174.

[16] S.B. Pope, PDF methods for turbulent reactive flows, Prog. Energy Combust. Sci. 11 (1985) 119–192.

[17] S.B. Pope, Computations of turbulent combustion: Progress and challenges, Proc. Combust. Inst. 23 (1990) 591–612.

[18] S.B. Pope, Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation, Combust. Theory Model. 1 (1997) 41–63.

[19] S.B. Pope, Turbulent Flows, Cambridge University Press, Cambridge, 2000.

[20] S.B. Pope, Self-conditioned fields for large-eddy simulations of turbulent flows, J. Fluid Mech. 652 (2010) 139–169.

[21] P.P. Popov, H. Wang, S.B. Pope, Specific volume coupling and convergence properties in hybrid particle/finite volume algorithms for turbulent reactive flows, J. Comput. Phys. (2013), submitted for publication.

[22] V. Raman, H. Pitsch, A consistent LES/filtered density function formulation for the simulation of turbulent flames with detailed chemistry, Proc. Combust. Inst. 31 (2007) 1711–1719.

[23] L. Valiño, A field Monte Carlo formulation for calculating the probability density function of a single scalar in a turbulent flow, Flow Turbul. Combust. 60 (1998) 157–172.

[24] J. Villermaux, J.C. Devillon, Reprèsentation de la redistribution des domaines de sègrègation dans un fluide par un modèle d'interaction phènomènologique, in: 2nd Int. Symp. Chem. React. Engng., Amsterdam, 1972.

[25] S. Viswanathan, H. Wang, S.B. Pope, Numerical implementation of mixing and molecular transport in LES/PDF studies of turbulent reacting flows, J. Comput. Phys. 230 (2011) 6916–6957.

[26] H. Wang, S.B. Pope, Large eddy simulation/probability density function modeling of a turbulent $CH_4/H_2/N_2$ jet flame, Proc. Combust. Inst. 33 (2011) 1319–1330.

[27] Y.Z. Zhang, D.C. Haworth, A general particle mass consistency algorithm for hybrid particle/finite volume PDF methods, J. Comput. Phys. 194 (2004) 156–193.

[28] A.R. Masri, S.B. Pope, PDF calculations of piloted turbulent nonpremixed flames of methane, Combust. Flame 81 (1990) 13–29.