A probability density function method for turbulent mixing and combustion on three-dimensional unstructured deforming meshes

S Subramaniam
Mechanical and Aerospace Engineering,
Rutgers University, Piscataway, New Jersey, USA

D C Haworth
Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, University Park, Pennsylvania, USA

Received 24 March 2000

Abstract: *A hybrid Lagrangian–Eulerian methodology is developed for numerical simulation of turbulent mixing and combustion in arbitrary three-dimensional time-dependent geometric configurations. The context is a probability density function (PDF) based approach intended for modelling in cylinder processes in reciprocating piston internal combustion (IC) engines. Issues addressed include mean estimation, particle tracking and particle number–density control on three-dimensional unstructured deforming meshes. The suitability of the methodology for statistically time-dependent three-dimensional turbulent flow with large density variations is demonstrated via simulations of turbulent freon vapour/air mixing on an unstructured deforming mesh representing an idealized IC engine* **[13]**. *Computed profiles of mean and r.m.s. freon mole fractions show good quantitative agreement with measurements. Moreover, inherent advantages of the Lagrangian–Eulerian PDF approach are demonstrated, compared to Eulerian finite volume solutions of an (approximately) equivalent set of moment equations. The new approach is, by design, compatible with existing computational fluid dynamics codes that are used for multi-dimensional modelling of in-cylinder thermal fluids processes. This work broadens the accessibility of PDF methods for practical turbulent combustion systems.*

Key words: *turbulent mixing, combustion, probability density function, modelling, unstructured deforming meshes*

1. Introduction

Improved physical models and numerical procedures for turbulent thermal fluids processes are essential in efforts to reduce the fuel consumption and pollutant emissions of practical non-homogeneous turbulent combustion systems. In the automotive industry, for example, direct injection (DI) engines are of interest for their high fuel-economy potential [**1**, **2**]. In DI engines, liquid fuel is injected directly into the combustion chamber, resulting in a highly non-homogeneous mixture of fuel, air and residual products of combustion at the time of ignition. Introduction of DI technology has been slowed by high emissions levels of unburned hydrocarbons, oxides of nitrogen and soot. Predictive three-dimensional transient simulations of flow, turbulence, mixing, fuel sprays and combustion are critical to DI combustion system design and optimization; however, the requisite physical models and numerical algorithms lie at the frontiers of current physical understanding and computability.

The motivation for this research has been to establish a computational methodology for highly stratified turbulent mixing and combustion processes in arbitrary three-dimensional time-dependent geometric configurations. The modelling approach is a composition probability density function (PDF) method. PDF methods are well suited to systems

with strong turbulence chemistry interaction and finite-rate chemistry [3]. Because of their large dimensionality, PDF transport equations are not amenable to grid-based numerical techniques; Lagrangian Monte Carlo (particle) methods have been developed as an alternative [4, 5]. Particle methods for the underlying hydrodynamic field (the filtered or averaged Navier–Stokes equations) also have been explored [6]. However, the latter remain immature for the complex flows of interest here. Therefore a hybrid Lagrangian–Eulerian approach is adopted where mean velocity, mean pressure and turbulence scales are computed using a 'conventional' grid-based computational fluid dynamics (CFD) solver.

Significant progress has been made in bringing PDF methods to bear on engineering flows. Anand *et al.* [7] extended a PDF method to incompressible statistically stationary two-dimensional elliptic flow in computations for a backward-facing step. Haworth [5] and Haworth and El Tahry [8, 9] developed PDF methodology for transient three-dimensional variable-density flows on unstructured deforming meshes, with applications to in-cylinder flow and mixing in a simple reciprocating piston engine. Correa *et al.* [10] reported PDF calculations for a statistically stationary two-dimensional methane–air flame. Several PDF2DV [11] modelling studies of statistically stationary two-dimensional reacting flows have appeared in the literature in recent years. In all cases cited here, a velocity composition PDF was used with hybrid Langrangian–Eulerian solution algorithms.

However, these studies and others have identified several remaining weaknesses both in physical modelling and in numerical methodology. Mean estimation on unstructured meshes is a particularly acute issue. Estimates of mean quantities obtained from particle values are needed to set the mixture density, fluid properties and modelled mean reaction rates; they also appear as coefficients in the particle evolution equations. In earlier internal combustion (IC) engine work [5, 8, 9], a computationally intensive spline-based algorithm was used. That approach is not well suited to arbitrary geometric configurations.

In this article methodology shortcomings that have limited the accessibility of PDF methods for practical turbulent combustion systems are addressed. The emphasis is on numerical aspects for unstructured deforming meshes:

1. A mean estimation algorithm [12] is extended to three-dimensional unstructured meshes; an analysis of numerical error is provided for canonical problems.
2. Other numerical issues including particle tracking and particle number–density control are addressed.
3. The methodology is demonstrated via simulations of turbulent freon/air mixing (density ratio 4.19) on an unstructured three-dimensional deforming mesh representing an idealized IC engine [13].

2. PDF Methods for Turbulent Reacting Flows

Salient aspects of PDF methods are summarized, with particular attention to the relationship between PDF equations and stochastic particle methods. For further information, the reader is referred to Pope [3].

2.1 Formulation

A low Mach number assumption is invoked so that spatial gradients in pressure do not affect the thermochemical equations. Then the mixture mass density ρ, specific heats c_p and c_v and chemical production rates $\boldsymbol{S}$ are functions only of composition, enthalpy and a reference pressure p_0 that is, at most, a function of time: $p_0 = p_0(t)$. Moreover, attention is limited to high Reynolds number flows; the turbulent transport of mass, momentum, chemical species and enthalpy dominates molecular transport processes.

The PDF considered is a one-point joint probability density function of the s scalar composition variables $\boldsymbol{\Phi}$ (N_S species mass fractions and enthalpy, say) that define the thermochemical state $f_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t)$. This Eulerian *composition PDF* is a density in the $s = N_S + 1$ dimensional composition space; it is, in addition, a function of up to three spatial coordinates $\boldsymbol{x}$ and of time t. The composition space can be large in the case of hydrocarbon–air systems. For example, twenty-nine chemical species were considered in a propane–air mechanism [14] used to construct a laminar flamelet library for IC engine combustion [15]; 30 composition variables (29 mass fractions and enthalpy) are carried in a composition PDF method based on this mechanism [16].

For variable density flows, it is advantageous to work with density weighted (Favre-averaged) quantities. The Favre PDF $\tilde{f}_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t)$, mass density function $\mathscr{F}_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t)$ and conventional PDF $f_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t)$ are related by $\mathscr{F}_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t) = \langle \rho(\boldsymbol{x}, t) \rangle \tilde{f}_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t) = \rho(\boldsymbol{\Psi}, p_0) f_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t)$, where $\langle \rho \rangle$ is the mean mixture density. Conventional ($\langle\ \rangle$) and density-weighted (~)

means of any function of the composition variables are readily computed from the PDF. Fluctuations about the conventional mean are designated by a single prime, while double primes are used for fluctuations about the Favre mean. Thus, for any $Q = Q(\boldsymbol{\Phi})$,

$$\int Q(\boldsymbol{\Psi}) f_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t)\, \mathrm{d}\boldsymbol{\Psi} = \langle Q(\boldsymbol{x}, t)\rangle$$

$$\int Q(\boldsymbol{\Psi}) \tilde{f}_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t)\, \mathrm{d}\boldsymbol{\Psi} = \tilde{Q}(\boldsymbol{x}, t)$$

$$Q(\boldsymbol{x}, t) = \langle Q(\boldsymbol{x}, t)\rangle + Q'(\boldsymbol{x}, t)$$

$$Q(\boldsymbol{x}, t) = \tilde{Q}(\boldsymbol{x}, t) + Q''(\boldsymbol{x}, t) \tag{1}$$

Starting from the Eulerian partial differential equations expressing conversion of mass, momentum, chemical species and enthalpy, a transport equation for $\tilde{f} \equiv \tilde{f}_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t)$ (or f or $\mathscr{F}$) can be derived:

$$\frac{\partial[\langle\rho\rangle\tilde{f}]}{\partial t} + \frac{\partial[\langle\rho\rangle\tilde{u}_i\tilde{f}]}{\partial x_i} + \frac{\partial[\langle\rho\rangle S_\alpha \tilde{f}]}{\partial \Psi_\alpha}$$
$$= -\frac{\partial}{\partial x_i}[\langle u_i'' | \boldsymbol{\Psi}\rangle\langle\rho\rangle\tilde{f}]$$
$$+ \frac{\partial}{\partial \Psi_\alpha}\left[\left\langle \rho^{-1}\frac{\partial J_i^\alpha}{\partial x_i}\middle| \boldsymbol{\Psi}\right\rangle \langle\rho\rangle\tilde{f}\right] \tag{2}$$

Here summation is implied over repeated indices i or α within a term ($\alpha = 1, \ldots, N_S + 1$). The notation $\langle u_i'' | \boldsymbol{\Psi}\rangle$ denotes the mean of the Favre velocity fluctuation, conditioned on the composition being $\boldsymbol{\Psi}$. Similarly, $\langle \rho^{-1}(\partial J_i^\alpha/\partial x_i) | \boldsymbol{\Psi}\rangle$ is the conditional mean of the divergence of the composition variable α molecular flux $\boldsymbol{J}^\alpha$. Transport in physical space by the Favre-mean velocity $\tilde{\boldsymbol{u}}$ and transport in composition space by chemical reaction $\boldsymbol{S}$ appear in closed form [left-hand side of equation (2)]. On the right-hand side are terms to be modelled. These represent transport in physical space by turbulence and transport in composition space by molecular diffusion. Equations (1) and (2) can be combined to derive moment transport equations; e.g. equations for the mean and variance of species mass fractions are readily derived (see below).

2.2 Lagrangian Monte Carlo method

The turbulent reacting flow is represented by a large number N of computational particles. Each particle is assigned a weight $w^{(i)}$ that is proportional to the mass of fluid that it represents. In the simplest case, $w^{(i)} = m/N$, where m is the total system mass. The ith particle is characterized by three position coordinates $\boldsymbol{x}^{(i)}(t)$ and by s scalar compositions $\boldsymbol{\Phi}^{(i)}(t)$ (again, species mass fractions and enthalpy).

Each computational particle can be interpreted as an independent realization of the flow or as a delta function discretization of the PDF. In a computational time step Δt, the position and compositions of each particle evolve according to

$$\boldsymbol{x}^{(i)}(t + \Delta t) = \boldsymbol{x}^{(i)}(t) + \tilde{\boldsymbol{u}}(\boldsymbol{x}^{(i)}(t), t)\,\Delta t + \Delta\boldsymbol{x}^{(i)}_{\text{turb}}$$
$$\boldsymbol{\Phi}^{(i)}(t + \Delta t) = \boldsymbol{\Phi}^{(i)}(t) + S^{(i)}(\boldsymbol{\Phi}^{(i)}(t), p_0(t))\,\Delta t + \Delta\boldsymbol{\Phi}^{(i)}_{\text{mix}} \tag{3}$$

Equations (3) account for convection by the mean flow ($\tilde{\boldsymbol{u}}$) and by turbulence ($\Delta\boldsymbol{x}^{(i)}_{\text{turb}}$): the latter is the 'turbulent diffusion'. The term $\Delta\boldsymbol{\Phi}^{(i)}_{\text{mix}}$ is the increment in composition due to molecular mixing.

Stochastic processes are used to model turbulent diffusion and molecular mixing. Standard models suffice for present purposes. Turbulent diffusion is modelled as [3]

$$\Delta\boldsymbol{x}^{(i)}_{\text{turb}} = \left[\frac{\nabla\Gamma_{\text{T}}}{\langle\rho\rangle}\right]_{\boldsymbol{x}^{(i)}(t)}\Delta t + \left[\frac{2\Delta t\Gamma_{\text{T}}}{\langle\rho\rangle}\right]^{1/2}_{\boldsymbol{x}^{(i)}(t)}\boldsymbol{\eta} \tag{4}$$

Here $\Gamma_{\text{T}} = c_\mu\langle\rho\rangle\sigma_{\text{T}}^{-1}k^2/\varepsilon$ is the turbulent diffusivity and k, ε, c_μ and σ_{T} are, respectively, the turbulence kinetic energy, dissipation rate of turbulence kinetic energy, model coefficient ($c_\mu = 0.09$) in a standard two-equation k–ε turbulence model [17] and a turbulent Schmidt or Prandtl number ($\sigma_{\text{T}} = 0.7$). The quantity $\boldsymbol{\eta}$ represents a vector of standardized Gaussian random variables (zero mean, unit variance). Scalar mixing is modelled using 'interaction by exchange with the mean' (IEM) [18]:

$$\Delta\boldsymbol{\Phi}^{(i)}_{\text{mix}} = -\tfrac{1}{2}C_\phi(\boldsymbol{\Phi}^{(i)} - \tilde{\boldsymbol{\Phi}})\omega\,\Delta t \tag{5}$$

where $\omega = \varepsilon/k$ is a turbulence 'frequency' and C_ϕ ($= 2.0$) is a model constant.

Equations (3), (4) and (5) imply a transport equation for the corresponding Eulerian composition PDF $\tilde{g}_{\boldsymbol{\Phi}}(\boldsymbol{\Psi}; \boldsymbol{x}, t)$:

$$\frac{\partial[\langle\rho\rangle\tilde{g}]}{\partial t} + \frac{\partial[\langle\rho\rangle\tilde{u}_i\tilde{g}]}{\partial x_i} + \frac{\partial[\langle\rho\rangle S_\alpha\tilde{g}]}{\partial\Psi_\alpha}$$
$$= \frac{\partial}{\partial x_i}\left[\Gamma_{\text{T}}\frac{\partial\tilde{g}}{\partial x_i}\right] + \frac{1}{2}C_\phi\omega\frac{\partial}{\partial\Psi_\alpha}[\langle\rho\rangle\tilde{g}(\Psi_\alpha - \tilde{\Phi}_\alpha)] \tag{6}$$

Transport equations for the first two moments of $\tilde{g}$

are obtained by weighted integrals of equation (6) over $\boldsymbol{\Psi}$ space [equation (1)]:

$$\frac{\partial[\langle\rho\rangle\tilde{\Phi}_\alpha]}{\partial t}+\frac{\partial[\langle\rho\rangle\tilde{u}_i\tilde{\Phi}_\alpha]}{\partial x_i}=\frac{\partial}{\partial x_i}\left[\Gamma_\mathrm{T}\frac{\partial\tilde{\Phi}_\alpha}{\partial x_i}\right]+\langle\rho\rangle\tilde{S}_\alpha \tag{7}$$

$$\begin{aligned}&\frac{\partial[\langle\rho\rangle\widetilde{\Phi_\alpha''^2}]}{\partial t}+\frac{\partial[\langle\rho\rangle\tilde{u}_i\widetilde{\Phi_\alpha''^2}]}{\partial x_i}\\&=\frac{\partial}{\partial x_i}\left[\Gamma_\mathrm{T}\frac{\partial\widetilde{\Phi_\alpha''^2}}{\partial x_i}\right]+2\Gamma_\mathrm{T}\frac{\partial\tilde{\Phi}_\alpha}{\partial x_i}\frac{\partial\tilde{\Phi}_\alpha}{\partial x_i}\\&\quad+2\langle\rho\rangle(\widetilde{S_\alpha\Phi_\alpha}-\tilde{S}_\alpha\tilde{\Phi}_\alpha)-C_\Phi\langle\rho\rangle\omega\widetilde{\Phi_\alpha''^2}\end{aligned} \tag{8}$$

There is no summation over a repeated composition index α within a term in equation (8).

By comparing equations (2) and (6), it can be seen that the stochastic processes used to model turbulent transport and molecular mixing in the particle equations imply closure models for corresponding terms in the PDF transport equation. The correspondence among PDF, moment and particle equations can be used to advantage in developing physical models: the modelling can be carried out in whichever context is most natural. For example, at the particle level, IEM [equation (5)] causes particle compositions to approach the local mean composition on a time scale ω^{-1}. At the moment level, IEM results in no change in mean composition while the variance $\widetilde{\Phi_\alpha''^2}$ decays exponentially on the time scale ω^{-1}. At the PDF level, IEM drives the composition PDF towards a delta function at the local mean composition.

These equations also serve to illustrate trade-offs between particle-based and grid-based solution strategies. A compelling benefit of particle methods is that they circumvent the difficulty of solving a $(3+s)$-dimensional integral differential equation for the PDF: instead integration is carried out on a large number $[N(3+s)]$ of stochastic ordinary differential equations for the particle properties. On the other hand, several robust, accurate and efficient grid-based methods have been developed for solving the highly elliptic coupled PDFs that govern the three-dimensional time-dependent mean fields (mean velocity, mean pressure, mean composition and mean energy or enthalpy) for turbulent reacting flows in reciprocating-piston engines and other complex engineering devices [**15, 19, 20**]. While full particle-based solutions to the PDF equation are possible in principle, it is expected that a more expedient approach will be that of a hybrid Lagrangian–Eulerian method that can take advantage of the strengths of both particle-based and grid-based methods. Detailed analysis of consistency conditions, numerical errors and convergence rates for hybrid Lagrangian–Eulerian methods have been published recently by Muradoglu *et al.* [**21**] and by Xu and Pope [**22**].

Models for $\Delta x^{(i)}_\mathrm{turb}$ and $\Delta\boldsymbol{\Phi}^{(i)}_\mathrm{mix}$ generally involve mean quantities [formally, integrals over the PDF; equation (1)]. In the IEM model, for example, the mean composition appears in the particle equation [equation (5)]. The availability of robust, accurate and computationally efficient methods for extracting estimates of mean quantities from noisy particle information is crucial to the advancement of hybrid particle/grid-based solution methods.

3. Numerical Approach

3.1 Eulerian CFD solver

In the underlying grid-based CFD code [**15, 23, 24**], the principal equations solved express conservation of mass, momentum, chemical species and energy or enthalpy for a compressible multiple-species ideal gas mixture. These equations are solved on an unstructured deforming mesh of hexahedral volume elements including limited degeneracies (Fig. 1)

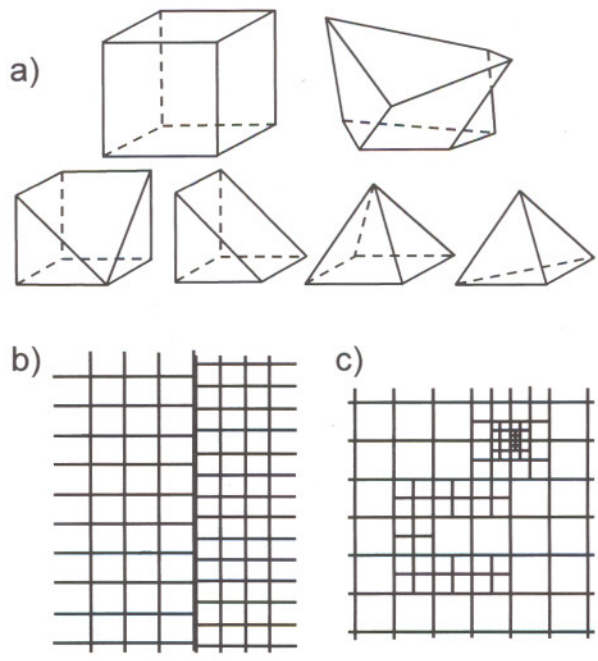


Fig. 1 *Element types and mesh topology. (a) Hexahedral element with examples of edge and face degeneracies. (b) A non-aligned interface: each element on one side of the interface communicates with an arbitrary number of elements on the other side* [**24**]. *(c) Local mesh refinement via cell splitting* [**25**].

using a finite volume discretization with co-located cell-centred variables. A pressure-based time-implicit algorithm is used. Accuracy is first order in time and up to second order in space. Governing equations are filtered (Reynolds-averaged) so that computed dependent variables represent ensemble-averaged quantities. Turbulence is treated using a standard two-equation ($k-\varepsilon$) model with wall functions [17]. To accommodate complex geometric configurations, non-aligned interfaces [24] and solution-adaptive local mesh refinement via cell splitting [25] have been implemented (Fig. 1). Second-order spatial accuracy is maintained across non-aligned interfaces.

Here the mean velocity [$\tilde{\boldsymbol{u}}(\boldsymbol{x}, t)$], mean pressure [$\langle p(\boldsymbol{x}, t)\rangle$] and turbulence scales [$k(\boldsymbol{x}, t)$, $\varepsilon(\boldsymbol{x}, t)$] are taken from the Eulerian grid-based computation; mean composition [mass fractions $\tilde{\boldsymbol{Y}}(\boldsymbol{x}, t)$ or mole fractions $\tilde{\boldsymbol{\chi}}(\boldsymbol{x}, t)$ and enthalpy $\tilde{h}$] is taken from the Lagrangian Monte Carlo solution. The particle compositions determine the mean mixture density [$\langle \rho(\boldsymbol{x}, t)\rangle$] and thermodynamic properties [$c_p(\boldsymbol{x}, t)$ and $c_v(\boldsymbol{x}, t)$] that are needed in the grid-based computation. Finite volume mean fields (vertex values) are interpolated to particle positions using trilinear basis functions.

3.2 Mean estimation

A three-stage mean estimation algorithm is employed. The method extends to three-dimensional unstructured meshes, an approach published by Dreeben and Pope [12] for one-dimensional problems.

To begin, the canonical problem of fitting noisy particle data is considered. Let $f(\boldsymbol{x})$ be a known function in physical space $\boldsymbol{x}$. This function is to be represented using an ensemble of N particles. The ith particle located at position $\boldsymbol{x}^{(i)}$ has an importance weight $w^{(i)}$ and a function value $\phi^{(i)}$. The particle data contain noise which is modelled as

$$\phi^{(i)} = f(\boldsymbol{x}^{(i)}) + \sigma\eta^{(i)} \tag{9}$$

where $\eta^{(i)}$ are independent, identically distributed (IID) standardized Gaussian random variables. Here σ is a measure of the noise in the particle data. The objectives are to construct an estimate $\hat{\phi}(\boldsymbol{x})$ for the function $f(\boldsymbol{x})$ from the particle data and to quantify the error in this estimate.

The computational domain is divided into a set of N_c convex hexahedral cells or elements; arbitrary edge and/or face degeneracies are permitted (Fig. 1). Each element is prescribed by an ordered list of eight nodes or vertices; there are a total of N_v vertices. The Cartesian coordinates of the vertices in the kth element are $\boldsymbol{x}_\nu^{[k]}$ ($\nu = 1, \ldots, 8$; $k = 1, \ldots, N_c$). There are $N^{[k]}$ particles in the kth element.

Within each element, the function is approximated in a manner analogous to that used in classic finite elements methods (FEM) [26]. The global coordinates in the kth element are denoted by $\boldsymbol{x}^{[k]}$ and the corresponding element (local) coordinates by $\boldsymbol{\xi}$. Then,

$$\boldsymbol{x}^{[k]}(\boldsymbol{\xi}) = N_\nu(\boldsymbol{\xi})\boldsymbol{x}_\nu^{[k]} \qquad (\nu = 1, \ldots, 8) \tag{10}$$

where N_ν are the element shape functions and summation is implied over the repeated index ν. Similarly, the spatial variation of any dependent variable $u(\boldsymbol{x})$ in the interior of the kth element is determined by the eight vertex values $u_\nu^{[k]}$ and by the element interpolation functions b_ν:

$$u^{[k]}(\boldsymbol{\xi}) = b_\nu(\boldsymbol{\xi})u_\nu^{[k]} \qquad (\nu = 1, \ldots, 8) \tag{11}$$

If the interpolation functions are chosen to be the same as the shape functions, then the element is said to be *isoparametric*. Isoparametric elements satisfy basic convergence requirements including smoothness, continuity and completeness [26]. Here an isoparametric representation is adopted where the shape or interpolation functions are taken to be the hexahedral natural (trilinear) coordinates. Given the ith particle's global coordinates $\boldsymbol{x}^{(i)}$ and its parent cell [k], the transformation given by equation (10) can be inverted numerically to yield $\boldsymbol{\xi}^{(i):[k]}$, the particle's position in element coordinates.

The three stages in mean estimation are: (a) kernel estimation of weighted particle values, (b) calculation of knot (cell vertex) values and (c) construction of the continuous approximating function within elements. Details are provided in the Appendix. An important aspect of the present formulation is the use of linear interpolation functions. Linear functions are particularly appealing for the turbulent combustion application. In the construction of turbulence closure models for conserved passive scalar quantities, it is necessary to respect physical constraints that arise from the linearity of the governing equations [27]. For reacting flows, computed species mean mass fractions must satisfy $\sum_{\alpha=1}^{N_S} \tilde{Y}_\alpha \equiv 1$. Linear mean estimation functions ensure the boundedness of mean quantities and guarantee that physical constraints applied at the particle level carry through to the mean quantities that feed into the Eulerian grid-based calculation. A major shortcoming of earlier spline-based mean estimation algorithms [5, 28] was their violation of such constraints.

To analyse the numerical errors incurred in extracting estimates of mean quantities from noisy particle data, an error field $\varepsilon(x)$ is introduced as the difference between the estimator $\hat{\phi}(x)$ and the underlying function $f(x)$:

$$\varepsilon(x) \equiv \hat{\phi}(x) - f(x) \tag{12}$$

Three sources of error can be differentiated in particle-based methods (see the Appendix). For present purposes, the total error in the numerical method will be discussed and no attempt will be made to isolate each of the three components individually. Thus, following Haworth [5], a single integrated r.m.s. total error ε_P will be introduced:

$$\varepsilon_P \equiv \left\{ \frac{1}{P} \sum_{p=1}^{P} [\hat{\phi}(x_p) - f(x_p)]^2 \right\}^{1/2} \tag{13}$$

and the behaviour of ε_P will be monitored resulting from variations in mesh size [mesh spacing $h \equiv (V_{tot}/N_c)^{1/3}$, where V_{tot} is the total volume], in particle number density ($N_{pc} = N/N_c$) and in noise (σ).

3.3 Particle tracking on unstructured meshes

Lagrangian–Eulerian coupling demands that each particle be associated with an element or cell and that the association be maintained for arbitrary particle and mesh motion. Several procedures have been developed for tracking particles through three-dimensional unstructured deforming meshes. These include methods based on linear basis functions (from references [8], [9] and [29] and also S. B. Pope, Cornell University, Ithaca, New York, personal communication, 1992, and M. S. Fairfield, Los Alamos National Laboratory, Los Alamos, New Mexico, 1993), superposition of a coarse Cartesian background mesh and the *convex polyhedron* method outlined here. The latter has been found to be particularly robust for highly skewed unstructured three-dimensional meshes that often arise in engineering computations. Other advantages are enumerated below.

The convex polyhedron method is based on a tetrahedral element linear basis function approach suggested by Pope (personal communication, 1992). The computational domain is decomposed into arbitrary non-overlapping convex polyhedral regions, each region having N_{face} faces. It is natural to use the computational elements for this purpose. The essence of the method is illustrated in Fig. 2. A face-based data structure is employed. The nth face of volume element k carries a location x_{kn} (mean position of the vertices defining the face), an outward unit normal vector b_{kn} (cross-product of two face diagonals) and connectivity information l_{kn} (pointer to the element that shares face n with element k). Because these quantities are computed consistently for the two elements sharing the face, no ambiguity arises from non-planar face vertices. Each particle is uniquely associated with the single element k for which all particle-to-face heights h_{kn} ($n = 1, \ldots, N_{face}$) are positive.

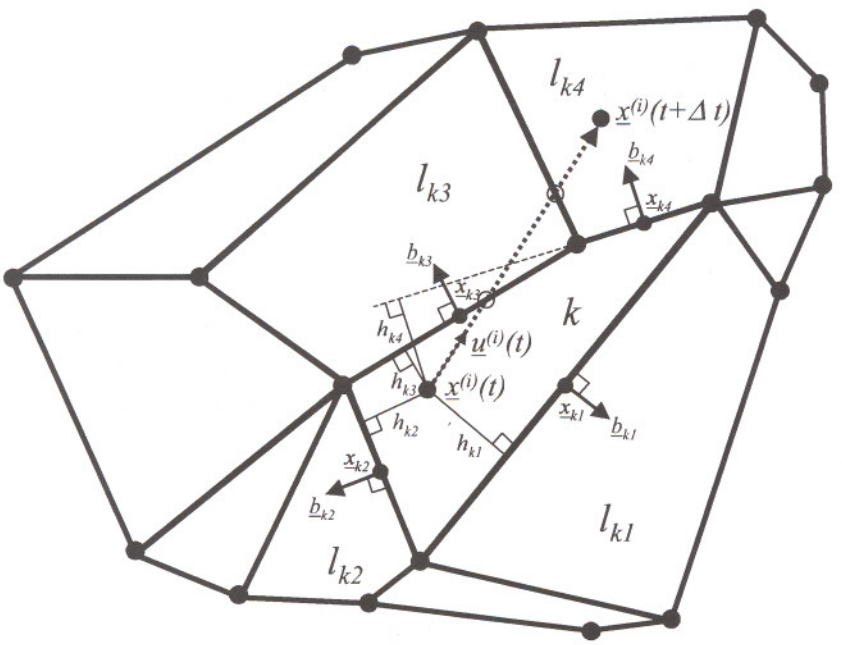


Fig. 2 *Particle tracking through an unstructured mesh. A two-dimensional example is shown, for clarity.*

At time t, particle i at location $\boldsymbol{x}^{(i)}(t)$ is associated with volume element k. It then undergoes a displacement by a velocity $\boldsymbol{u}^{(i)}(t)$ over a time step Δt. The time to intersect the nth face of element k is $t_{\text{int}kn} = h_{kn}/(\boldsymbol{u}^{(i)}(t) \cdot \boldsymbol{b}_{kn})$; the minimum time to intersect over the N_{face} faces is $t_{\min k} = \min_{n=1}^{N_{\text{face}}}\{\max\{0, t_{\text{int}kn}\}\}$. At the end of the time step, the particle remains inside element k if $t_{\min k} > \Delta t$. If $t_{\min k} < \Delta t$, the particle is moved to the face intersection point $\boldsymbol{x}_{\text{int}} = \boldsymbol{x}^{(i)}(t) + \boldsymbol{u}^{(i)}(t)t_{\min k}$, its element pointer is updated to l_{kn} and its time step is decremented by $t_{\min k}$. Tracking continues until all particles have zero time remaining.

Tracking from face to face has several advantages:

1. Degenerate (zero area) faces are accommodated: these simply are inaccessible to particles.
2. Deforming meshes are readily implemented via a two-step procedure. Firstly, the particle-to-element pointer is updated to account for vertex motion with particle positions fixed. Secondly, particles are tracked normally over the computational time step with fixed (new) vertex locations.
3. It is straightfoward to tabulate the fraction of time spent in each element when the 'particle Courant number' $|\boldsymbol{u}^{(i)}|\Delta t/\Delta x_k$ is greater than unity (Δx_k being the linear dimension of an element). Such information is needed in the construction of higher-order Lagrangian methods [**30**].
4. Particle boundary conditions are readily implemented since it is immediately known where and when a particle strikes a boundary of the computational domain.
5. Finally, non-aligned interfaces [**24**] and local mesh refinement [**25**] (Fig. 1) are accommodated with additional pointer structures.

This particle tracking scheme is particularly advantageous for tetrahedral meshes ($N_{\text{face}} = 4$): in that case the face heights h_{kn} are identical to the natural linear basis functions that would be used for mean estimation. A hexahedral mesh can be decomposed into tetrahedra, but the resulting data structure is cumbersome. The (small) computational overhead of computing trilinear basis functions for mean estimation is therefore accepted as an additional operation.

3.4 Particle number density control

Particle-based computations are ideally performed in the dense data limit. However, such large particle numbers are generally not practical for three-dimensional meshes of 10^5–10^6 volume elements. In any case, maintaining adequate control over the spatial distribution of particles is critical to the success of hybrid Lagrangian–Eulerian methods.

The number of particles per unit volume in physical space is initially prescribed to be uniform. To accommodate possible large element-to-element volume disparities, it is necessary to impose a minimum number of particles per computational element (typically between five and ten). Particle weights are initialized to the element's fluid mass divided by the number of particles in the element. This ensures that the particle and element mass and volume are initially consistent.

As the particles and mesh vertices move, particle number density control is exercised both to maintain consistency between particle and element mass (or volume) and to enforce a minimum number of particles per computational element. There is a relationship between the spatial distribution of particle mass or volume and the mean continuity equation [**3**]: if the interpolated mean velocity field used to advect particles satisfies the mean continuity equation, then the particle spatial distribution will remain consistent with the fluid mass and volume distribution.

Here rudimentary particle number density control is implemented [**9**]. Particles are shifted in physical space to maintain consistency between element and particle volumes. Particles are cloned in elements where the number of particles drops below a prescribed threshold. In cloning, a parent particle of weight $w^{(i)}$ is split into two particles of weight $w^{(i)}/2$ having the same properties $\boldsymbol{\Phi}^{(i)}$ as the parent. If the total number of particles N approaches a prescribed threshold $N_{\max}$, particles are selectively annihilated. Annihilation occurs only in elements having larger than nominal particle number density. Within an element, low mass (weight) particles are selected preferentially for annihilation; the mass associated with annihilated particles is distributed among remaining particles in the element.

4. Test Cases

Results are reported for two test cases. The first is a static function test where particles do not move relative to the mesh; this case serves to characterize the error associated with extracting estimates of mean quantities from noisy particle data. In the second problem, the suitability of the methodology is demonstrated for a configuration approaching that of a reciprocating piston IC engine.

4.1 Static function

The computational domain is a right circular cylinder of radius r_0 and length L; here $r_0 = 1$ and $L = 2$. The number of particles in each cell is proportional to the cell volume; within each cell, particles are distributed uniformly in local (trilinear) coordinates. For this simple geometry and mesh topology, the result is a nearly uniform particle number density in physical (global coordinate) space. Particle weights are all equal. The ith particle is assigned a function value $\phi^{(i)}$ according to equation (9). Following Haworth [5], the test function $f(\boldsymbol{x})$ is taken to be

$$f(\boldsymbol{x}) = \left(1 - \frac{r}{r_0}\right) \cos\left(\frac{5\pi r}{2r_0}\right) \sin\left(\frac{2\pi z}{L}\right) \tag{14}$$

The algorithm outlined in the Appendix is used to extract the estimator $\hat{\phi}(\boldsymbol{x})$ and the global r.m.s. error ε_P is computed using equation (13). The response of the error to parametric variations in mesh size $[h = (\pi r_0^2 L/N_c)^{1/3}]$, particle number density ($N_{pc} = N/N_c$) and noise (σ) is studied. In all cases, ε_P is computed using a uniform mesh of 51^3 test points.

Visual comparisons between the underlying function $f(\boldsymbol{x})$ [equation (14)] and the estimator function $\hat{\phi}(\boldsymbol{x})$ [equation (30) in the Appendix] are provided in Fig. 3 (Algorithm IIA). There the three-dimensional function is evaluated at fixed values of y and z ($y = -0.0308$, $z = 0.97$). Since z/L is close to 0.5, the function values are small along this y = constant, z = constant cut: the maximum value is approximately 0.1

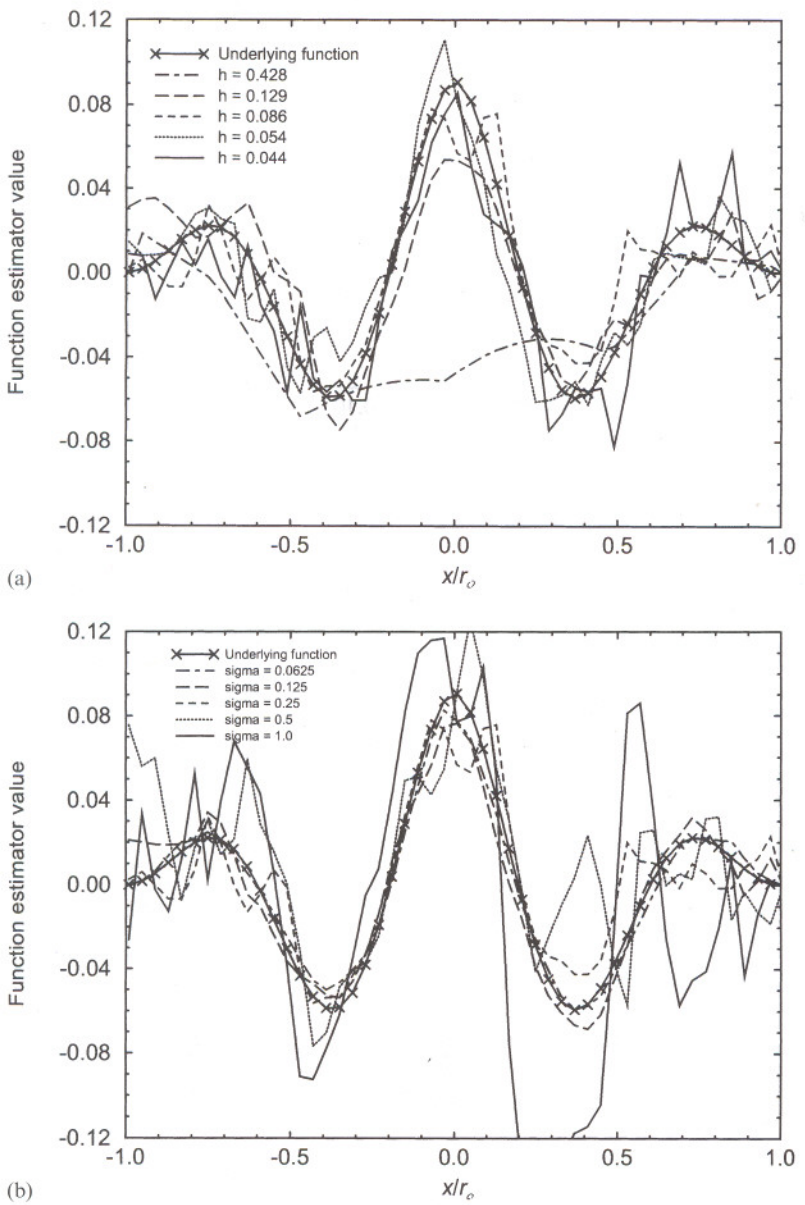


Fig. 3 *One-dimensional cut through a three-dimensional static test function $f(\boldsymbol{x})$ [equation (14)] and function estimators $\hat{\phi}(\boldsymbol{x})$ [equation (30)], Algorithm IIA, at $y = -0.0308$, $z = 0.97$. (a) Variation in h, for fixed $N_{pc} = 40$ and $\sigma = 0.25$. (b) Variation in σ, for fixed $h = 0.086$ ($N_c = 10\,000$) and $N_{pc} = 40$.*

at $x = 0$, compared to a global maximum value for equation (14) of unity. This choice serves to accentuate differences between $f(\boldsymbol{x})$ and $\hat{\phi}(\boldsymbol{x})$.

In Fig. 3a, mesh size (h) is varied with fixed N_{pc} and σ; this emphasizes the discretization error. In particular, the coarsest mesh ($N_c = 80$, $h = 0.428$) is inadequate to resolve the underlying function for any choice of N_{pc} and σ. For $\sigma = 0$, the estimators systematically clip the peaks in the underlying function (not shown). Figure 3b similarly shows the expected trend with increasing noise σ for fixed h and N_{pc}.

The error in estimating the underlying function is quantified in Figs 4 and 5. Figure 4 shows the global r.m.s. error ε_P as a function of mesh spacing h for fixed $N_{pc} = 40$ and zero noise ($\sigma = 0$). Algorithm IIA gives somewhat lower error than Algorithm IIC for this choice of bandwidth. For sufficiently small h, the error scales approximately as h^2.

Systematic variations in mesh size h, particle number density N_{pc} and function noise reveal the expected trends: error decreases with mesh refinement (decreasing h), error decreases with increasing

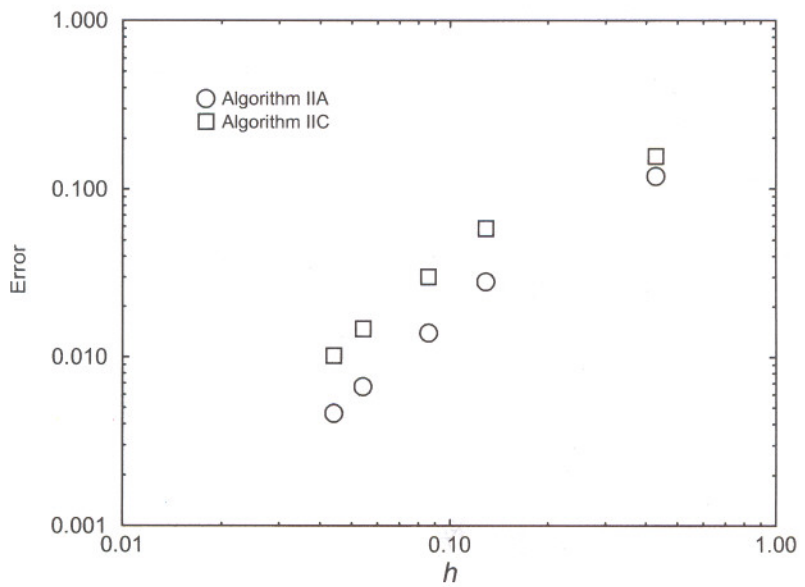


Fig. 4 *R.m.s. error ε_P [equation (13)] versus mesh size h for a three-dimensional static test function $f(\boldsymbol{x})$ [equation (14)]. Here $N_{pc} = 40$, $\sigma = 0$ and results for two different Stage II algorithms are shown. For Algorithm IIC, the 'bandwidth' R in units of the local cell size $V_c^{1/3}$ is equal to unity [equation (26)].*

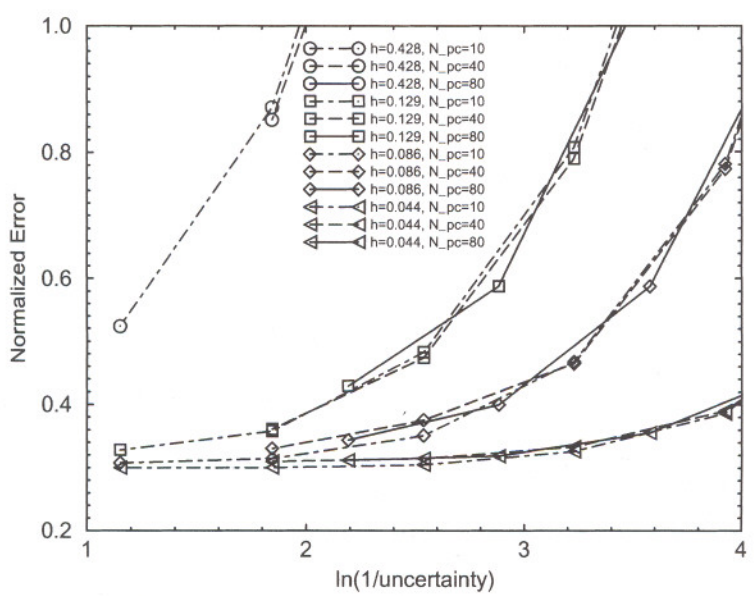


Fig. 5 *Normalized r.m.s. error ε^* [equation (16)] versus uncertainty γ [equation (15)] for a three-dimensional static test function $f(\boldsymbol{x})$ [equation (14)] with variations in mesh size h and in particle number density N_{pc} for Algorithm IIA.*

particle number density (increasing N_{pc}) and error decreases with decreasing noise (decreasing σ). Of particular interest in a Monte Carlo method is the behaviour of the random statistical error. Standard statistical arguments suggest that the error ε_P should depend not on σ and N independently, but rather on a measure of the *uncertainty* in the data γ, where [5, 28]

$$\gamma \equiv \sigma / N_{pc}^{1/2} \tag{15}$$

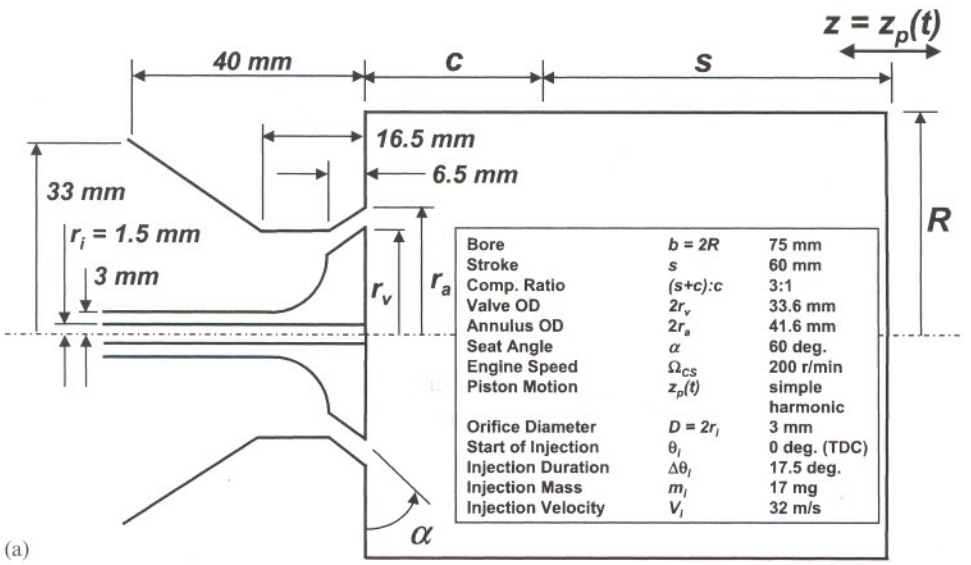
A normalized error ε^* is then defined as

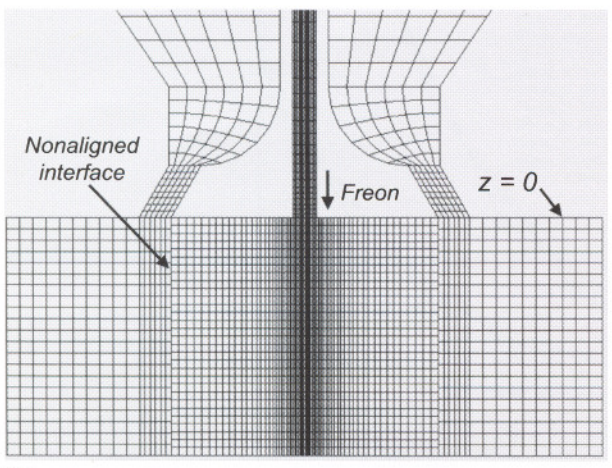
$$\varepsilon^* \equiv \varepsilon_P / \gamma \tag{16}$$

For large noise σ, ε^* should become independent of h and of N_{pc}. This behaviour is evident in Fig. 5, where large σ corresponds to the left-hand side of the figure [small $\ln(1/\gamma)$]. The limiting behaviour is achieved earlier on the finer meshes (smaller h). Figures 3 to 5 together serve to demonstrate the consistency and error scaling of the present three-dimensional mean estimation algorithm.

4.2 Idealized IC engine

The next test case is intended to demonstrate the suitability of the methodology for a three-dimensional transient engine-like configuration with a large density variation. A three-dimensional mesh is used to model the axisymmetric geometry of Fig. 6 [13]. In the experiment, freon 12 (dichlorodifluoromethane) vapour was injected into air through a





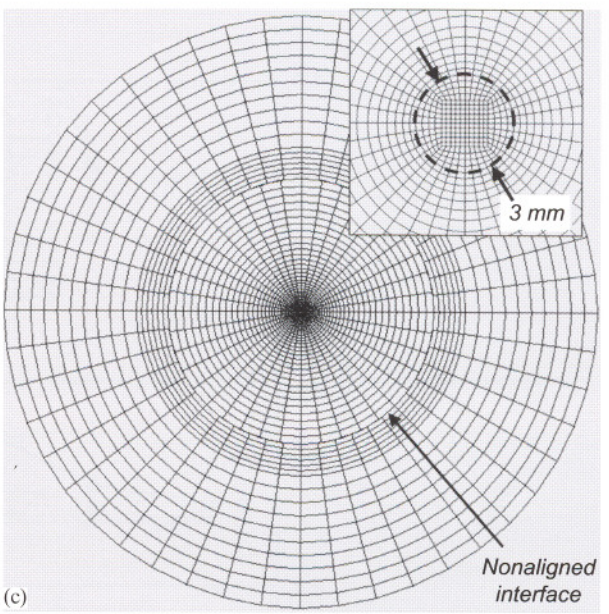


Fig. 6 *An idealized reciprocating piston IC engine with 'fuel' (freon) injection* [13]. *(a) Geometric configuration. (b) Computational mesh on a cutting plane containing the geometric axis of symmetry. (c) Computational mesh on a cutting plane normal to the geometric axis of symmetry, $z < 0$.*

3 mm diameter hole bored along the centre-line of the valve stem; this was done to study fundamental aspects of fuel injection processes in direct injection engines. Rayleigh scattering was used to extract ensemble-averaged profiles of freon mole fraction mean (first moment) and variance (second central moment). The freon–air molecular weight ratio is 4.19. Other relevant parameters are summarized in Fig. 6.

Mesh size, particle number density and the computational time step are selected to be representative of practical values for engineering calculations. The computational mesh contains 78 008 volume elements (84 543 vertices), corresponding to a nominal in-cylinder mesh size of about 1 mm^3. The number of computational particles is maintained between 1 300 000 and 1 500 000 for a nominal particle number density of 16.20 per element. The computational time step ranges from one-eighth of a crank angle (CA) degree (1.042×10^{-4} s) during 'fuel' injection to one-half of a degree (4.167×10^{-4} s) for the remainder of the computation. The unstructured character of the mesh and large variation in element volumes are apparent in Fig. 6. The in-cylinder mesh deforms in a simple 'accordian' motion as the piston moves. A non-aligned interface is included to isolate the mesh topology in the vicinity of the freon jet from the remainder of the domain. Local mesh refinement is not used.

The composition PDF of three species mass fractions ($s = N_S = 3$) is computed. Species 1 corresponds to 'residual' air initially in the cylinder ($z < 0$), species 2 to 'fresh' air initially upstream of the cylinder ($z > 0$) and species 3 to freon. Computations begin at piston top dead centre (TDC) with quiescent mean flow, uniform ambient pressure and temperature, and delta function mass fraction PDFs. The mass flowrate of freon is prescribed as a function of time at the inlet orifice. Isothermal walls are assumed (300 K). Particle positions and weights are initialized as described in Section 3.4 and evolve in time according to equations (3) to (5) with $S = 0$. Mean quantities are extracted from particle values using Algorithm IIA (see the Appendix). The combination of a low particle number density and the simplest (and least accurate) mean estimation schemes are selected to challenge the robustness of the proposed computational methodology.

The fluid mixture density and specific heats are prescribed assuming an ideal gas mixture. With χ_α denoting the mole fraction of species α and W_α its molecular weight, the following relations pertain:

$$\chi_\alpha = \frac{Y_\alpha W}{W_\alpha}$$

$$Y_\alpha = \frac{\chi_\alpha W_\alpha}{W}$$

$$W = \sum_{\alpha=1}^{N_S} W_\alpha \chi_\alpha = \left(\sum_{\alpha=1}^{N_S} \frac{Y_\alpha}{W_\alpha} \right)^{-1}$$

$$\rho = pW(RT)^{-1} \tag{17}$$

Here R is the universal gas constant. Since $\chi_{\text{freon}} = 1 - \chi_{\text{air}}$ here and the effect of temperature fluctuations on mixture density is ignored, the mean mixture density $\langle \rho \rangle$ can be written as a function of the freon mole fraction mean as

$$\langle \rho \rangle = \rho_{\text{air}} \left[1 + \langle \chi_{\text{freon}} \rangle \left(\frac{W_{\text{freon}}}{W_{\text{air}}} - 1 \right) \right] \tag{18}$$

where $\rho_{\text{air}} = \rho W_{\text{air}} (RT)^{-1}$.

It is significant that density fluctuations resulting from composition fluctuations are accounted for properly in the Lagrangian PDF formulation. Each particle's mixture density is computed using equation (17) and the mean mixture density is extracted using the mean estimation algorithm described earlier. Conventional ($\langle\ \rangle$) or density-weighted ($\tilde{\cdot}$) means, and either mass or mole fractions, can be extracted without further assumption. By contrast, density fluctuations cannot be accommodated in an equivalent manner in a moment formulation, even in the absence of chemical reaction. With $S \equiv 0$, equations (7) and (8), together with a suitable prescription for $\langle \rho \rangle$ as a function of $\tilde{Y}_\alpha$ and $\widetilde{Y_\alpha''^2}$, provide a closed set of moment equations for the Favre mean and variance of the species mass fractions. Using only two moments, it is not possible in general to specify $\langle \rho \rangle$ in a manner that is consistent with that implied by the full mass fraction PDF. The usual approximation would be to ignore fluctuations in molecular weight. In that case,

$$\langle \rho \rangle_{\text{FV}} = p\bar{W}(RT)^{-1}, \qquad \text{with } \bar{W} = \left(\sum_{\alpha=1}^{N_S} \frac{\tilde{Y}_\alpha}{W_\alpha} \right)^{-1} \tag{19}$$

Here the hybrid Lagrangian–Eulerian PDF solution is compared to a 'conventional' Eulerian finite volume solution of an *almost* equivalent set of moment equations [equations (7) and (8) with equation (19)].

Figure 7 shows computed instantaneous fields of

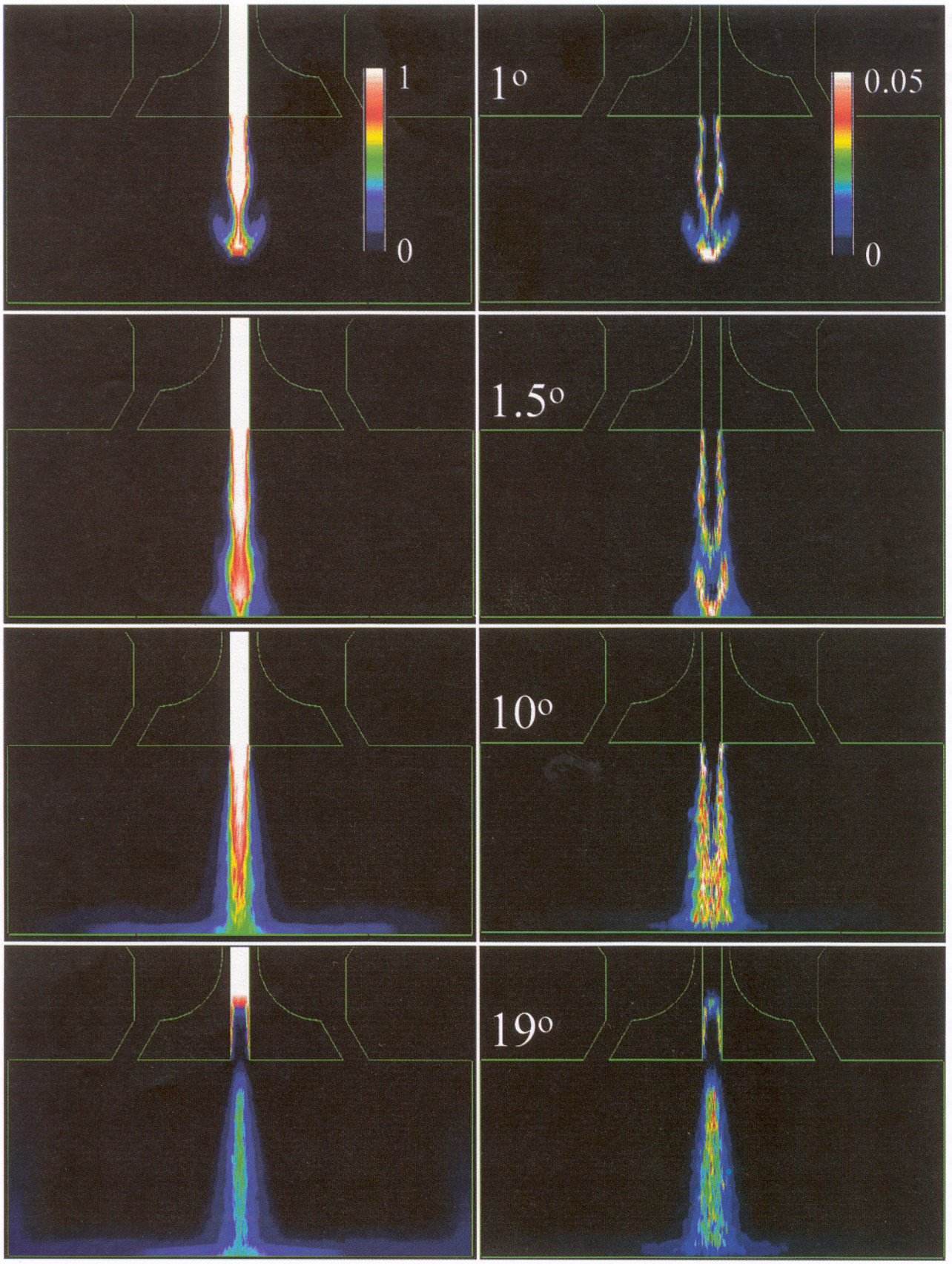


Fig. 7 *Computed contours of freon mole fraction mean (left) and variance (right) during the injection event on a cutting plane containing the geometric axis of symmetry.*

freon (species 3) mole fraction mean and variance on a cutting plane through the geometric axis of symmetry during the injection event. [Injection begins at piston TDC (0°) and ends at 17.5° after TDC; see Fig. 6.] These computed fields should be axisymmetric in the limit of sufficiently high particle number density; departures from axisymmetry indicate statistical error in the Monte Carlo method. A richness of structure can be seen in the impulsively started jet for about 4° CA after the start of injection. This is followed by a quasi-steady phase that lasts until close to the end of injection. Results for species 1 (residual air) in Fig. 8 provide an indication of the character of the mixing field through a full engine cycle.

Quantitative comparisons between computations and measurement are given in Figs 9 and 10. Figure 9 contains profiles of the mean and r.m.s. freon mole fraction along the centre-line at 10° after the start of injection. Computed results are compared to the free-jet measurements of Arcoumanis *et al.* [**13**]. For the PDF computations, five curves are shown: one corresponding to the geometric centre-line and the other four to profiles extracted along lines parallel to the centre-line a short radial distance (0.25 mm) away. These multiple curves serve to illustrate visually the degree of statistical error in the particle-based method at low particle number density; ideally, the off-axis curves should be coincident for this axisymmetric configuration. In spite of the statistical error, both the decay in the mean and the growth in the r.m.s. with downstream position are captured reasonably well by the Lagrangian–Eulerian PDF method. By contrast, the finite volume moment solution yields similar results for the mean but underpredicts the r.m.s. by about a factor of two.

Similar behaviour is evident in radial profiles at fixed axial positions (Fig. 10). Here the several curves for the PDF calculation correspond to radial rays in different azimuthal directions from the centre-line. Close to the jet exit (Fig. 10a), the PDF computation appears to smear the steep mean gradient at the edge of the jet somewhat and broaden the r.m.s. profile, but the peak r.m.s. value is captured quite well. By the second measurement station (Fig. 10b) the PDF-computed peak r.m.s. mole fraction is slightly lower than the measured value. These results suggest that the combination of simple physical models for turbulent diffusion and molecular mixing, standard k–ε turbulence and a fairly coarse mesh together yield results that are somewhat overdiffusive. The moment solution mean profile is similar to that from the PDF calculation, while the r.m.s. values again are lower by a factor of two.

The factor of two difference in r.m.s. between the Lagrangian PDF solution and the finite volume moment solution is significant. This difference results in part from the different effective equations of state in the two methods, as discussed above; however, this effect is probably small. The difference is most likely dominated by numerical dissipation. Since the mean mole fraction profiles are similar between the PDF and moment solutions, the production of variance [the second term on the right-hand side of equation (8)] should be comparable for both methods. The *physical* dissipation of variance [the last term on the right-hand side of equation (8)] is augmented by *numerical* dissipation in the solution methodology. For the Lagrangian particle method, numerical dissipation is inherently low.

In the case of a chemically reacting jet, the mean reaction rate would be (approximately) proportional to the fuel mass fraction variance. Thus, in order to make the overall heat release rate consistent with measurements, a moment-based approach would need to compensate for underestimation of the variance through a larger coefficient of proportionality, or other spurious model changes.

The most significant conclusion to be drawn from the results presented in Figs 7 to 10 is that hybrid Lagrangian–Eulerian PDF-based modelling is feasible and can yield quantitatively accurate results for three-dimensional time-dependent flows with large density variation using relatively low particle number densities.

5. Summary and Discussion

Mean estimation and other numerical aspects of hybrid Lagrangian–Eulerian methods for unstructured deforming meshes have been addressed. Test cases have been selected to provide an assessment of numerical error, quantitative comparison with experimental measurements and a demonstration of the methodology in a configuration approaching that of a realistic reciprocating piston IC engine. Advantages of the hybrid Lagrangian–Eulerian solution methodology include the ability to account properly for density fluctuations and inherently low numerical dissipation. These advantages become even more compelling in the case of chemically reacting flows.

A principal concern in Lagrangian Monte Carlo methods compared to conventional grid-based

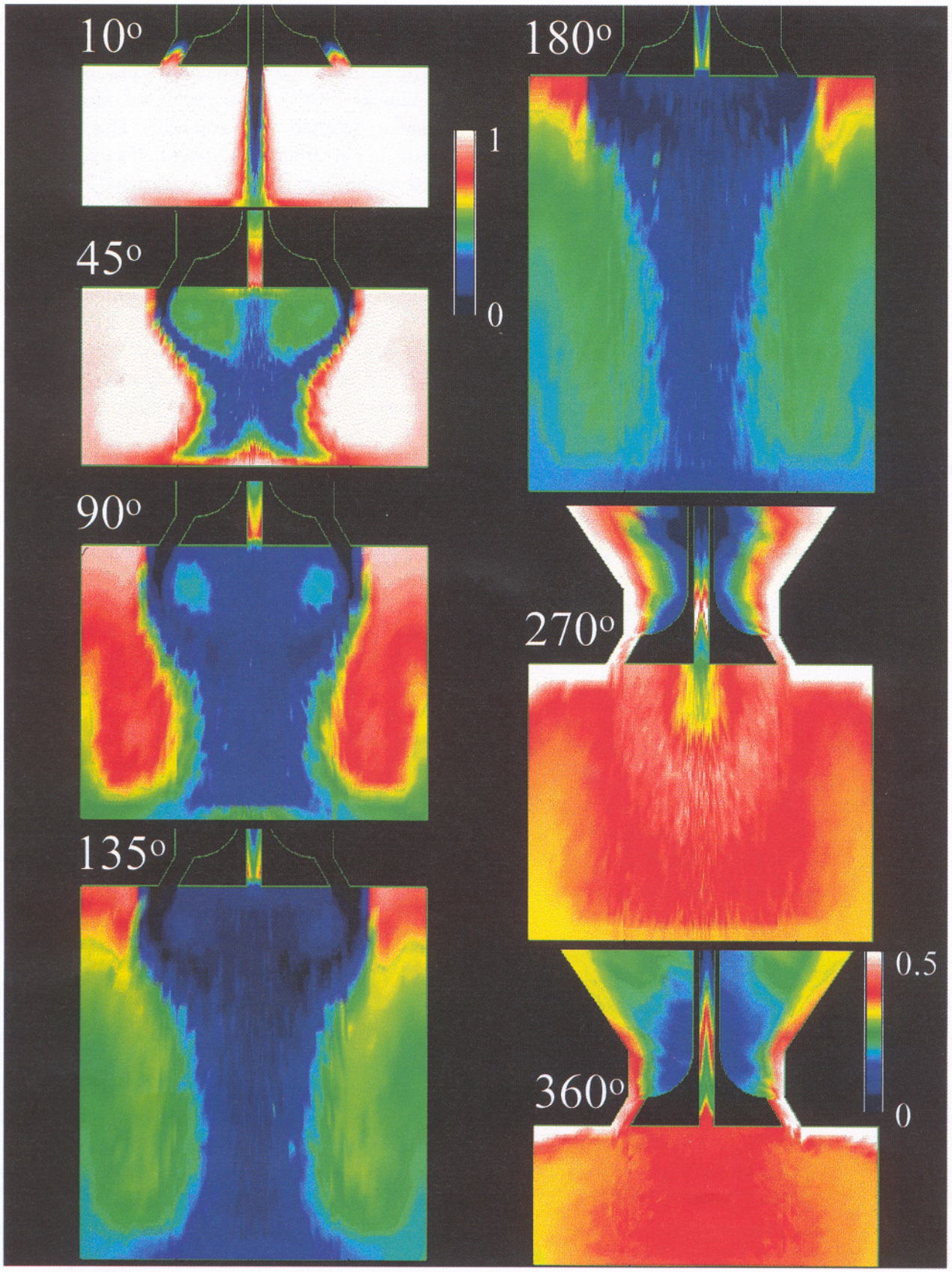


Fig. 8 *Computed contours of 'residual' air (species 1) mean mole fraction through a full engine cycle on a cutting plane containing the geometric axis of symmetry.*

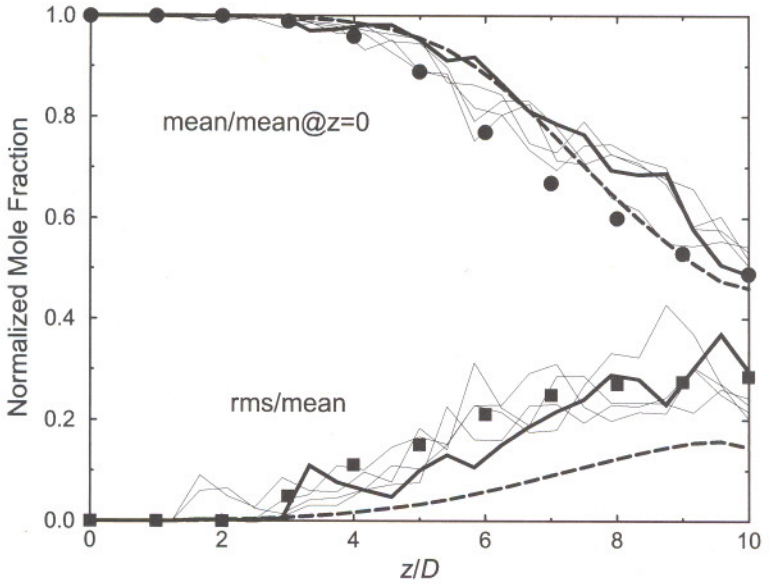


Fig. 9 *Computed and measured axial profiles of the mean and r.m.s. freon mole fraction along the axis of symmetry at 10° CA after piston TDC. Symbols are the free-jet experimental measurements of Arcoumanis et al.* [13]. *Solid lines are results from the hybrid Lagrangian–Eulerian PDF method; bold lines correspond to $r = 0$ and lighter lines to $r = 0.25$ mm. Dashed lines are results from an Eulerian finite volume solution of the moment equations.*

approaches is statistical error. The present results suggest that usable results can be obtained using practical particle number densities for three-dimensional time-dependent flow configurations. A second issue is computational cost. The hybrid Lagrangian–Eulerian solution for the freon jet required approximately three times the CPU time of the moment solution. It should be noted that this represents a fairly crude initial implementation of the particle algorithms, with little specific attention to computational efficiency. The particle-based models involve inherently simple and repetitive operations with weak particle-to-particle coupling; they are thus particularly well suited to a scalable parallel implementation. It is anticipated that the computational overhead can be reduced significantly.

This work broadens the accessibility of PDF methods for practical turbulent combustion systems. While some specific choices have been made to facilitate implementation in the (predominantly) hexahedral-based finite volume code used in the present study, the methodology is compatible with most Eulerian grid-based CFD solution methods [19, 20]. In fact, it is most natural for node-based tetrahedral element data structures.

The methodology presented is equally applicable to composition PDFs, to velocity–composition PDFs and to velocity composition dissipation PDFs [31]. Moreover, this provides the framework necessary for implementing particle-based subgrid scale models of turbulence, mixing and combustion in large-eddy simulation [32, 33]. The present study represents one component of a broader effort to develop consistent particle-based methods and models for liquid fuel sprays, species mixing, flamelet and non-flamelet modes of combustion and subgrid scale models in large-eddy simulation of turbulent reacting flows. Fundamental aspects of Lagrangian particle methods for representing two-phase flows have been explored by Subramaniam [34, 35].

Several interesting issues have arisen in the course of modelling the simplified engine configuration. These include: the detailed structure of the impulsively started jet and its implications for fuel injection processes in engines, the ongoing search for improved models of turbulence and turbulent mixing and implications of the potentially high freon residual fraction that may accumulate over multiple engine cycles. These issues are beyond the scope of the present methodology article and their discussion is deferred to subsequent reports.

Acknowledgement

This work has been supported by the General Motors Research and Development Center, who made available generous amounts of computer time for the simulations reported herein.

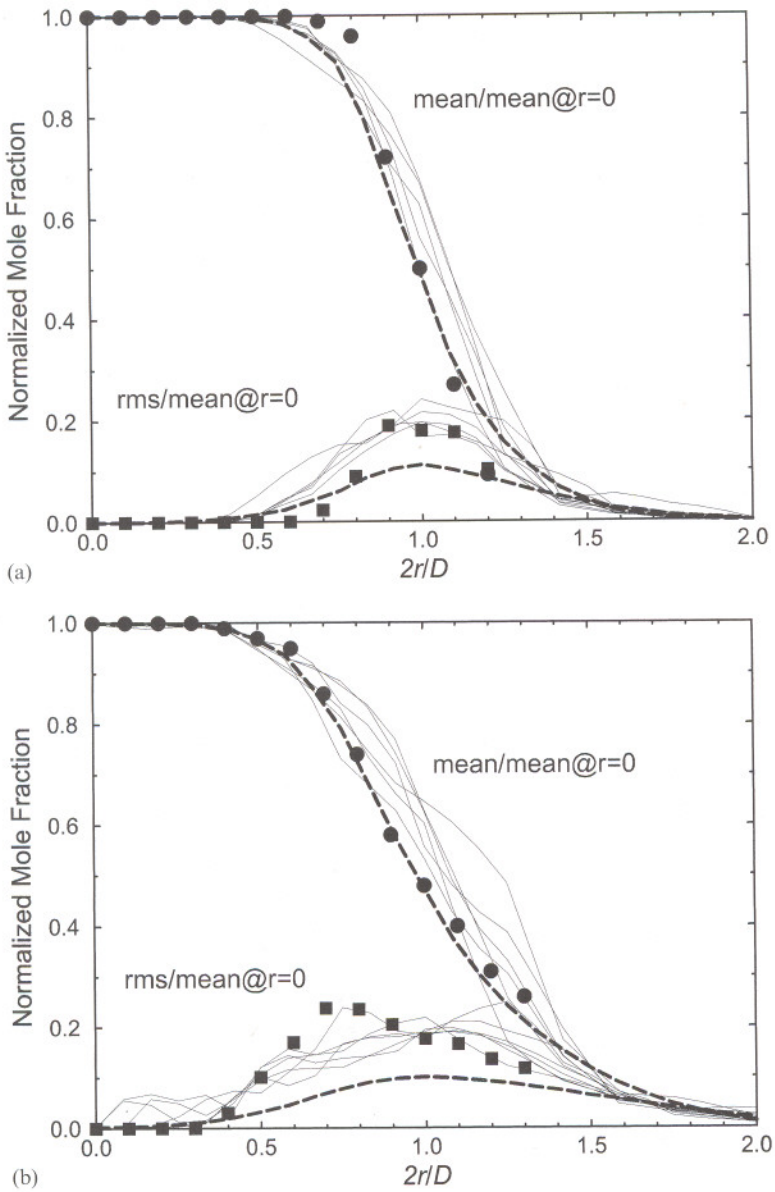


Fig. 10 *Computed and measured radial profiles of the mean and r.m.s. freon mole fraction at 10° CA after piston TDC. Symbols are free-jet experimental measurements of Arcoumanis et al.* [13]. *Solid lines are results from the hybrid Lagrangian–Eulerian PDF method; each line corresponds to a different azimuthal ray from the centre-line. Dashed lines are results from an Eulerian finite volume solution of the moment equations. (a) x/D = 1 (z = −3 mm), (b) x/D = 2 (z = −6 mm).*

References

1 **Krieger, R. B., Siewert, R. M., Pinson, J. A., Gallopoulos, N. E., Hilden, D. L., Monroe, D. R., Rask, R. B., Solomon, A. S. P.** and **Zima, P.** Diesel engines: one option to power future personal transportation vehicles. SAE Paper 972683, 1997.

2 **Lai, M. C., Zhao, F. Q.** and **Harrington, D. L.** A review of mixture preparation and combustion control strategies for spark ignited direct injection gasoline engines. SAE Paper 970627, 1997.

3 **Pope, S. B.** Pdf methods for turbulent reactive flows. *Prog. Energy Combust. Sci.*, 1985, **11**, 119–192.

4 **Pope, S. B.** A Monte Carlo method for pdf equations of turbulent reactive flow. *Combust. Sci. Technol.*, 1981, **25**, 159–174.

5 **Haworth, D. C.** Developments in applying a pdf/Monte Carlo approach to engine cylinder flows. Technical Report GMR 5892, General Motors Research Laboratories, Warren, Michigan, 1987.

6 **Brackbill, J. U.** and **Monaghan, J. J.** (Eds) Proceedings of the Workshop on *Particle Methods in Fluid Dynamics and Plasma Physics*. In *Comput. Phys. Commun.*, 1988, **48**.

7 **Anand, M. S., Pope, S. B.** and **Mongia, H. C.** A pdf method for turbulent recirculating flows. In *Turbulent Reactive Flows: Lecture Notes in Engineering*, 1989, Vol. 40, pp. 672–693.

8 **Haworth, D. C.** and **El Tahry, S. H.** Application of a pdf method to in-cylinder flows in reciprocating engines. In Proceedings of the Seventh Symposium on *Turbulent Shear Flows*, Stanford University, Stanford, California, 1989, pp. 13.1.1–13.1.6.

9 **Haworth, D. C.** and **El Tahry, S. H.** Probability density function approach for multidimensional turbulent flow calculations in reciprocating engines. *Am. Inst. Aeronaut. Astronaut. J.*, 1991, **29**, 208–218.

10 **Correa, S. M., Gulati, A.** and **Pope, S. B.** Raman

measurements and joint pdf modeling of a nonpremixed bluff body stabilized methane flame. In Twenty-Fifth Symposium (International) on *Combustion*, 1994, pp. 1167–1173 (The Combustion Institute, Pittsburgh, Pennsylvania).

11 **Pope, S. B.** PDF2DV: a Fortran code for solving modeled joint-pdf equations in two dimensions. Cornell University, Ithaca, New York, 1994 (unpublished).

12 **Dreeben, T. D.** and **Pope, S. B.** Nonparametric estimation of mean fields with application to particle methods for turbulent flows. Technical Report FDA 92 13, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, New York, 1992.

13 **Arcoumanis, C., Green, H. G.** and **Whitelaw, J. H.** The application of laser Rayleigh scattering to a reciprocating model engine. SAE Paper 840376, 1984.

14 **Blint, R. J.** Stretch in premixed laminar flames under IC engine conditions. *Combust. Sci. Technol.*, 1991, **75**, 115–127.

15 **Khalighi, B., El Tahry, S. H., Haworth, D. C.** and **Huebler, M. S.** Computation and measurement of flow and combustion in a four valve engine with intake variations. SAE Paper 950287, 1995.

16 **Blint, R. J.** and **Haworth, D. C.** Modeling oxidation of exhaust gases diluted by air under exhaust manifold conditions. In *Proceedings of the Combustion Institute*, 2000, Vol. 28 (to appear).

17 **Launder, B. E.** and **Spalding, D. B.** *Mathematical Models of Turbulence*, 1972 (Academic Press, New York).

18 **Dopazo, C.** Probability density function approach for a turbulent axisymmetric heated jet: centerline evolution. *Phys. Fluids*, 1975, **18**, 397–404.

19 **Amsden, A. A.** KIVA-3V: a block-structured KIVA program for engines with vertical or canted valves. Los Alamos National Laboratory Report LA-13313-MS, 1997.

20 **O'Rourke, P. J.** and **Sahota, M. S.** CHAD: a parallel, 3D implicit, unstructured grid, multi-material hydrodynamics code for all flow speeds. In Proceedings of NECDC (Nuclear Explosives Code Developer's Conference), Las Vegas, Nevada, 26–30 October 1998.

21 **Muradoglu, M., Jenny, P., Pope, S. B.** and **Caughey, D. A.** A consistent hybrid finite-volume/particle method for the pdf equations of turbulent reactive flow. *J. Comput. Phys.*, 1999, **154**, 342–371.

22 **Xu, J.** and **Pope, S. B.** Numerical studies of pdf/Monte Carlo methods for turbulent reactive flows. *J. Comput. Phys.*, 1999, **152**, 192–230.

23 **Haworth, D. C., El Tahry, S. H., Huebler, M. S.** and **Chang, S.** Multidimensional port and cylinder flow calculations for two and four valve per cylinder engines: influence of intake configuration on flow structure. SAE Paper 900257, 1990.

24 **Haworth, D. C., Maguire, J. M., Matthes, W. R., Rhein, R.** and **El Tahry, S. H.** Dynamic fluid flow analysis of oil pumps. SAE Paper 960422, 1996.

25 **Chang, S.** and **Haworth, D. C.** Adaptive grid refinement using cell level and global imbalances. *Int. J. Numer. Meth. Fluids*, 1997, **24**, 375–392.

26 **Hughes, T. J. R.** *The Finite Element Method*, 1987 (Prentice-Hall, New York).

27 **Pope, S. B.** Consistent modeling of scalars in turbulent flows. *Phys. Fluids*, 1983, **26**, 404–408.

28 **Pope, S. B.** and **Gadh, R.** Fitting noisy data using cross validated cubic smoothing splines. *Commun. Statist. Simulation*, 1988, **17**, 349–376.

29 **Lohner, R.** and **Ambrosiano, J.** A vectorized particle tracer for unstructured grids. *J. Comput. Phys.*, 1990, **91**, 22–31; *J. Comput. Phys.*, 1990, **154**, 342–371.

30 **Haworth, D. C.** and **Pope, S. B.** A second order Monte Carlo method for the solution of the Ito stochastic differential equation. *Stochastic Analysis and Applications*, 1986, **4**, 151–186.

31 **Pope, S. B.** and **Chen, Y. L.** The velocity–dissipation probability density function model for turbulent flows. *Phys. Fluids*, 1990, **2**, 1437–1449.

32 **Colucci, P. J., Jaberi, F. A., Givi, P.** and **Pope, S. B.** Filtered density function for large eddy simulation of turbulent reacting flows. *Phys. Fluids*, 1998, **10**, 499–515.

33 **Haworth, D. C.** and **Jansen, K.** Large eddy simulation on unstructured deforming meshes: towards reciprocating IC engines. *Computers and Fluids*, 2000, **29**, 493–524.

34 **Subramaniam, S.** Statistical representation and modeling of sprays: I—point process representation. *Phys. Fluids*, 2000 (to appear).

35 **Subramaniam, S.** Statistical representation and modeling of sprays: II—the droplet distribution function model. *Phys. Fluids*, 2000 (to appear).

36 **Priestley, M. B.** and **Chao, M. T.** Non-parametric function fitting. *J. R. Statist. Soc.*, 1972, **34**, 385–392.

37 **Cleveland, W. S., Devlin, S. J.** and **Grosse, E.** Regression by local fitting: methods, properties, and computational algorithms. *J. Econometrics*, 1988, **37**, 87–114.

Appendix

A three-stage mean estimation algorithm is presented, and numerical errors incurred in extracting estimates of mean quantities from noisy particle data are discussed.

Stage I: kernel estimation

A kernel estimate corresponding to the jth vertex is formed by summing over the particles in each element that shares the vertex; the set of elements sharing the jth vertex is denoted by C_j. (Figure 2 shows that each interior vertex of a *structured* three-dimensional hexahedral element mesh is shared by exactly eight elements.) Kernel estimates of particle weight $\bar{w}_{(j)}$, weighted particle position $\bar{\boldsymbol{X}}_{(j)}$ and weighted particle property $\bar{\phi}_{(j)}$ are given by

$$\bar{w}_{(j)} = \sum_{k \in C_j} \sum_{i=1}^{N^{[k]}} w^{(i)} b_\alpha(\boldsymbol{\xi}^{(i):[k]}) \delta_{\alpha\nu:[k]}$$

$$\bar{\boldsymbol{X}}_{(j)} = \sum_{k \in C_j} \sum_{i=1}^{N^{[k]}} w^{(i)} \boldsymbol{x}^{(i)} b_\alpha(\boldsymbol{\xi}^{(i):[k]}) \delta_{\alpha\nu:[k]} / \bar{w}_{(j)}$$

$$\bar{\phi}_{(j)} = \sum_{k \in C_j} \sum_{i=1}^{N^{[k]}} w^{(i)} \phi^{(i)} b_\alpha(\boldsymbol{\xi}^{(i):[k]}) \delta_{\alpha\nu:[k]} / \bar{w}_{(j)}$$

$$(\alpha = 1, \ldots, 8;\ \nu = 1, \ldots, 8;\ j \equiv \nu : [k]) \qquad (20)$$

Here $\delta_{\alpha\nu}$ is Kronecker's delta, and the notation $j \equiv \nu : [k]$ is used to establish the equivalence between the global ordering of vertices and the ordering within the kth element. It is important to note that the kernel estimates for weight and weighted particle property are values at the weighted particle position $\bar{X}_{(j)}$ and not at the vertex location $x_\nu^{[k]}$.

Stage II: vertex values

The next step is to establish 'knot values' at element vertices.

IIA: simple approximation

The simplest approach is to ignore the distinction between weighted particle position and vertex location; i.e. the vertex value is set to the corresponding kernel estimate,

$$\phi_\nu^{[k]} = \bar{\phi}_{(j)} \qquad (j \equiv \nu : [k]) \tag{21}$$

IIB: least-squares minimization

At the next level of sophistication, vertex values are determined from kernel estimates by a least-squares error minimization. The error to be minimized, E_j, is the difference between the interpolated function evaluated at the mean weighted particle position and the kernel estimate associated with the vertex.

$$E_j = \sum_{k \in C_j} \sum_{l=1}^{D_k} [\phi_\nu^{[k]} b_\nu(\xi(\bar{X}_l)) - \bar{\phi}_{(l)}]^2 \tag{22}$$

Here D_k is the number of kernel estimates that lie in the interior of the kth element. Least-squares minimization requires that $\partial E_j / \partial \phi_\mu^{[k]} = 0$ $(j \equiv \mu : [k])$, which results in

$$\sum_{k \in C_j} \sum_{l=1}^{D_k} [\delta_{\nu\mu} b_\nu(\xi(\bar{X}_l))][\phi_\nu^{[k]} b_\nu(\xi(\bar{X}_l)) - \bar{\phi}_{(l)}] = 0$$

or

$$\sum_{k \in C_j} \sum_{l=1}^{D_k} b_\mu(\xi(\bar{X}_l)) b_\nu(\xi(\bar{X}_l)) \phi_\nu^{[k]} = \sum_{k \in C_j} \sum_{l=1}^{D_k} b_\mu(\xi(\bar{X}_l)) \bar{\phi}_{(l)} \tag{23}$$

Solving for $\phi_\nu^{[k]}$ requires solution of a square linear system of size $(N_v)^2$ having a banded matrix structure. With the assumption of diagonal dominance, the vertex values can be solved for explicitly:

$$\phi_\mu^{[k]} = \frac{\sum_{k \in C_j} \sum_{l=1}^{D_k} b_\mu(\xi(\bar{X}_l)) \bar{\phi}_{(l)}}{\sum_{k \in C_j} \sum_{l=1}^{D_k} b_\mu(\xi(\bar{X}_l))} \tag{24}$$

While equation (24) does not correspond to an exact least-squares minimization, it does account for the influence of kernel estimates at neighbouring vertices, and thus presumably is a better approximation than equation (21).

IIC: non-parametric local regression

Neither IIA nor IIB provides independent control over the size of the neighbourhood that influences a vertex value, beyond that implicit in the element structure. Moreover, those two algorithms are parametric; i.e. they assume a functional form within each element that is given by the interpolation functions. By contrast, algorithm IIC is non-parametric. It is based only on certain smoothness criteria of the function to be estimated [**36**]. This algorithm is a direct extension of the local regression ideas developed by Cleveland *et al.* [**37**] and Dreeben and Pope [**12**].

A local polynomial estimate $\hat{\phi}_j(\boldsymbol{x})$ for the function $f(\boldsymbol{x})$ is formed using the kernel estimate data points in a neighbourhood of the jth vertex. The neighbourhood is taken to be a sphere of radius R centred at $\boldsymbol{x}_{(j)}$. The vertex value at the jth vertex is then taken to be the local estimate evaluated at the vertex location. Here linear estimates are used. Extension to quadratic or higher-order polynomials is straightforward but computationally more expensive. It can be shown that, for a polynomial of order p, there are $[1 + (3^p - 3)/2 + 3]$ coefficients to be solved for via a matrix equation.

For this algorithm it is convenient to think of the vertices as a global set rather than as ordered octets associated with the N_c elements. Given the kernel estimates associated with the vertices, the set of kernel estimate data points in the neighbourhood of the jth vertex $D_{(j)}$ is defined as $D_{(j)} = \{l \,|\, \rho_{(l)(j)} \equiv |\bar{X}_{(l)} - x_{(j)}| < R\}$, where R is a user-defined radius of influence. Here R is specified as a multiple of a local characteristic cell size $V_c^{1/3}$. In the neighbourhood of the jth vertex, the local linear estimate for f (for the purposes of determining the jth vertex value) is given by

$$\hat{\phi}_{(j)}(\boldsymbol{x}) = a + b_n(x_n - x_{n(j)}) \qquad (n = 1, 2, 3) \tag{25}$$

The coefficients a and b_n are determined by minimizing E, the weighted error between the local linear

estimate for f and the kernel estimate data points that lie in the neighbourhood of the jth vertex. The weighted error E can be written as

$$E = \sum_{i \in D_{(j)}} [a + b_n(\bar{X}_{n(i)} - x_{n(j)}) - \bar{\phi}_{(i)}]^2 W\left(\frac{\rho_{(i)(j)}}{R}\right) \bar{w}_{(i)} \tag{26}$$

where $W(u)$ ($u \in [0, 1]$) is a weight function; here a simple top-hat function has been used. Upon imposing the minimization requirements ($\partial E/\partial a = 0$ and $\partial E/\partial b_n = 0$), a matrix equation for the coefficients a and b_n ($n = 1, 2, 3$) is obtained:

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix} \begin{bmatrix} a \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} \bar{\phi}_{(i)} \\ \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{1(i)} - x_{1(j)}) \bar{\phi}_{(i)} \\ \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{2(i)} - x_{2(j)}) \bar{\phi}_{(i)} \\ \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{3(i)} - x_{3(j)}) \bar{\phi}_{(i)} \end{bmatrix} \tag{27}$$

where the elements of the symmetric matrix M_{ij} are given by

$$\begin{aligned}
M_{11} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} \\
M_{12} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{1(i)} - x_{1(j)}) \\
M_{13} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{2(i)} - x_{2(j)}) \\
M_{14} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{3(i)} - x_{3(j)}) \\
M_{22} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{1(i)} - x_{1(j)})^2 \\
M_{23} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{2(i)} - x_{2(j)})(\bar{X}_{1(i)} - x_{1(j)}) \\
M_{24} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{3(i)} - x_{3(j)})(\bar{X}_{1(i)} - x_{1(j)}) \\
M_{33} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{2(i)} - x_{2(j)})^2 \\
M_{34} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{3(i)} - x_{3(j)})(\bar{X}_{2(i)} - x_{2(j)}) \\
M_{44} &= \sum_{i \in D_{(j)}} \hat{W}_{(i)(j)} (\bar{X}_{3(i)} - x_{3(j)})^2
\end{aligned} \tag{28}$$

The weight function $\hat{W}_{(i)(j)}$ is given by

$$\hat{W}_{(i)(j)} = W\left(\frac{\rho_{(i)(j)}}{R}\right) \bar{w}_{(i)} \tag{29}$$

This 4×4 system is solved locally for each vertex.

Stage III: element interiors

Finally, the continuous variation of the approximating function $\hat{\phi}$ (the estimate for f) in the kth element is prescribed in terms of the vertex values formed in Stage II:

$$\hat{\phi}^{[k]}(\xi) = b_\nu(\xi) \phi_\nu^{[k]} \qquad (\nu = 1, \ldots, 8) \tag{30}$$

Numerical errors in mean estimation

For an estimator of the form given in equation (30), three sources of error can be differentiated [**12, 22, 28**]:

$$\varepsilon(x) = d(x) + \beta(x) + r(x) \tag{31}$$

Here $d(x)$ is the *deterministic error* or *discretization error*, $\beta(x)$ is the *bias* and $r(x)$ is the *random statistical error*.

The discretization error results from the interpolation functions $N_\nu = b_\nu$ used to construct a continuous field from the finite set of vertex values. If there were no noise in the particle data ($\sigma = 0$) and the particle number density were infinite ($N^{[k]} \to \infty$) then Stage II would return precisely the value of the underlying function at each node: i.e. $\phi_\nu^{[k]} = f(x_\nu^{[k]})$. The corresponding estimator is denoted as $\hat{\phi}_0(x)$:

$$\hat{\phi}_0^{[k]}(\xi) = b_\nu(\xi) f(x_\nu^{[k]}) \qquad (\nu = 1, \ldots, 8) \tag{32}$$

The deterministic error is defined in terms of $\hat{\phi}_0(x)$ and $f(x)$ as

$$d(x) \equiv \hat{\phi}_0(x) - f(x) \tag{33}$$

This error is equal to zero at vertices. As the particle number density increases, the condition of *dense data* is reached, where the deterministic error becomes independent of the number of particles.

The remaining error $\hat{\phi}(x) - f(x) - d(x)$ is then decomposed into systematic and random components. The remaining systematic component is the

bias $\beta(x)$. To understand bias, the probabilistic mean or expected value of the estimator $\langle\hat{\phi}(x)\rangle$ is needed [equation (2)]. Bias is the difference between this value and the function $\hat{\phi}_0(x)$:

$$\beta(x) \equiv \langle\hat{\phi}(x)\rangle - \hat{\phi}_0(x) \tag{34}$$

For an unbiased method, $\langle\hat{\phi}(x)\rangle$ is equal to $\hat{\phi}_0(x)$ and *not* to the underlying function $f(x)$. The latter would occur only in the exceptional case where the spatial variation in $f(x)$ is exactly compatible with the interpolation functions $N_\nu(\xi) = b_\nu(\xi)$. Clearly, unbiased estimators ($\beta(x) \equiv 0$) are desirable. In practice, bias in particle methods arises from feedback of estimated mean quantities into the particle evolution equations.

Finally, the random statistical error $r(x)$ is all that remains:

$$r(x) \equiv \hat{\phi}(x) - \langle\hat{\phi}(x)\rangle \tag{35}$$

By definition, $r(x)$ has zero mean ($\langle r(x)\rangle = 0$). Statistical error arises from the finite sample size and noise in the particle data. The magnitude of the statistical error scales as $N^{-1/2}$.

Integrated mean squared error quantities can be defined corresponding to each of the three error fields:

$$\varepsilon_a^2 \equiv \frac{1}{V}\int_V [d(x)]^2\,\mathrm{d}x$$

$$\varepsilon_\beta^2 \equiv \frac{1}{V}\int_V [\beta(x)]^2\,\mathrm{d}x$$

$$\varepsilon_r^2 \equiv \frac{1}{V}\int_V [r(x)]^2\,\mathrm{d}x \tag{36}$$

A one-dimensional study by Dreeben and Pope [**12**] demonstrated that the choice of the grid spacing (for a finite number of particles) affects the balance of these contributions to the total error. In numerical simulations, these mean squared errors are approximated by numerical counterparts computed on a test grid of P points distributed uniformly through the computational domain.