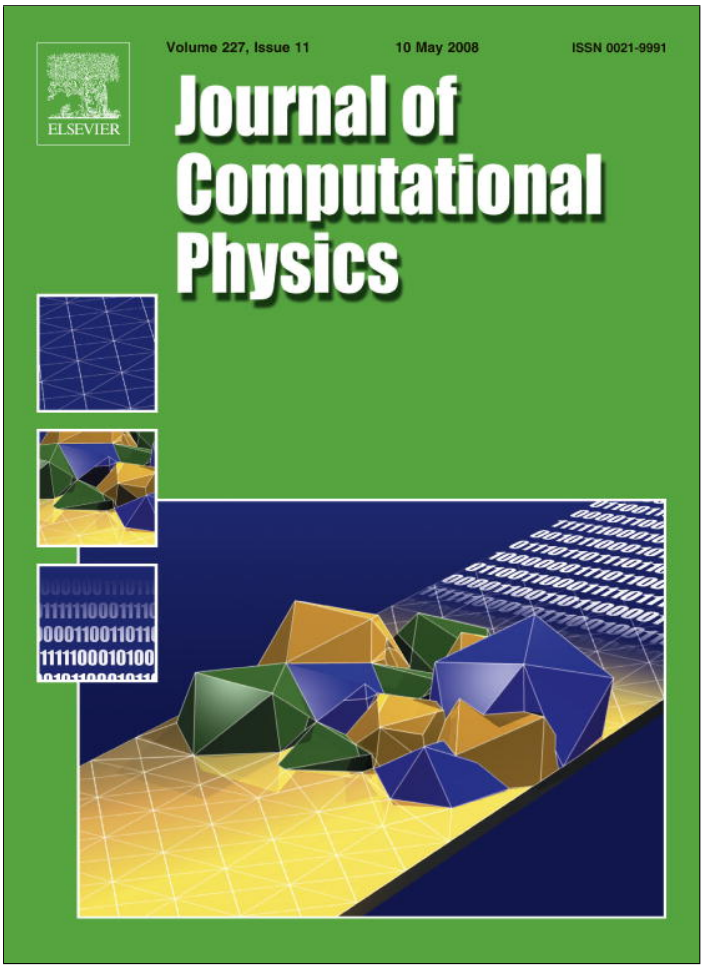

Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 5447–5491
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

# The parabolic edge reconstruction method (PERM) for Lagrangian particle advection

R. McDermott [a,*], S.B. Pope [b]

[a] Building and Fire Research Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8663, USA
[b] Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA

Received 5 September 2007; received in revised form 3 January 2008; accepted 30 January 2008
Available online 8 February 2008

**Abstract**

We describe a Lagrangian particle advection scheme which is intended for use in hybrid finite-volume (FV) large-eddy simulation/filtered density function (LES/FDF) methods for low-Mach flows, but which may also be applicable to unsteady probability density function (PDF) methods, direct numerical simulation (DNS) or any other situation where tracking fluid particles is of concern. A key ingredient of the scheme is a subgrid reconstruction of the filtered velocity field with desirable divergence properties, which is necessary for accurate evolution of the particle number density. We develop reconstructions for 2D and 3D Cartesian staggered non-uniform grids. The reconstructed velocity field is continuous and piecewise parabolic in the velocity-component direction. In the direction normal to the velocity component the reconstruction is piecewise linear. The divergence of the reconstructed field is bilinear in 2D (trilinear in 3D) within a given cell and consistent with the discrete divergence given by the staggered-grid velocities. Though the reconstructed divergence field may be discontinuous from cell to cell, the norm of the differences between the vertex values of the reconstructed divergence for neighboring cells is minimized. As a consequence, the divergence is everywhere zero for the constant-density case. A two-stage Runge–Kutta scheme is employed for advancement of the particle positions. To assess the performance of the scheme we utilize a set of non-trivial velocity test functions which are designed to mimic realistic flow fields. We show that an advection scheme based on the new velocity reconstruction method is effective at maintaining an accurate particle number density in the particle-tracking limit.


*PACS:* 65C20; 76F65

*Keywords:* Particle method; Velocity reconstruction; Velocity interpolation; Filtered density function methods; Large-eddy simulation; Direct numerical simulation; Turbulent reacting flow

## 1. Introduction

A recent trend in the modeling of turbulent reacting flows is to combine large-eddy simulation (LES) with the filtered density function (FDF) concept [9,24] into hybrid LES/FDF methods. In the present work, follow-

* Corresponding author. Tel.: +1 301 975 4310.
*E-mail address:* randall.mcdermott@nist.gov (R. McDermott).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.01.045

ing Sheikhi et al. [29] and Raman et al. [27,28], we consider the use of a joint composition filtered mass density function (FMDF) in which the filtered velocity is obtained from the LES solver and the subgrid velocity fluctuations are modeled. The FMDF is modeled by a Lagrangian particle method and is tied to the LES solution through the mean particle velocity and the mean particle mass density. These methods retain the principal advantage of the probability density function (PDF) approach [23] in that the chemical source term appears in closed form. They also leverage the maturity of low-Mach variable-density LES solvers (e.g. [7,13,22]) which achieve accuracy of the LES filtered velocity field together with stability of the numerical solution – a non-trivial matter.

A key issue in hybrid LES/FDF methods is the consistency between the filtered mass density used in the LES solver and the mean particle mass density derived from the ensemble of Lagrangian particles used to model the FMDF. Various strategies are employed to reconcile the redundant density fields (see e.g. [18,28,32]). At the numerical level, consistency between the corresponding Lagrangian and Eulerian density fields requires that the cloud-in-cell (CIC) mean particle mass density evaluated at the finite-volume (FV) cell center matches the FV LES filtered mass density (see e.g. [25] for a detailed discussion of CIC means). Given an initial particle position distribution which is consistent with the LES filtered mass density, the density fields will deviate due to: (1) errors in the numerical solution of the LES filtered mass density, (2) statistical errors in the initial particle position distribution, (3) bias in the CIC mean particle density at forward times, (4) inaccuracies in the particle advection scheme and (5) statistical noise from the model for the particle velocity fluctuations which account for subgrid-scale (SGS) turbulent transport and also typically account for mean molecular transport. In this paper, we address issues related to the accuracy of the particle advection scheme and thus take the particle velocity fluctuations to be zero. This is known as the "particle-tracking limit."

The inaccuracies of the advection scheme stem in part from errors in the estimation of the local filtered fluid velocity from the discrete LES data. As pointed out by Pope [23], the divergence field that the particles experience plays a key role in the evolution of the particle position density (which, as we will show, is proportional to the fluid filtered mass density). Jenny et al. [12] addressed this issue in a RANS (Reynolds-averaged Navier–Stokes) context for two-dimensional (2D) Cartesian grids by designing an interpolation scheme that is piecewise parabolic in the velocity-component direction and linear in the direction normal to the velocity component. The coefficients of the interpolation are specified so that the divergence varies bilinearly within the cell and approximates the finite-volume divergence at the cell vertices. Furthermore, the Jenny interpolation has the property that particles see a zero divergence everywhere in the constant-density limit.

Our goal here is to improve upon the scheme of Jenny et al. [12]. The main contributions of the present work are:

(1) a variant (with improved accuracy) of the 2D interpolation scheme of Jenny et al. [12] based on the piecewise parabolic method (PPM) of Colella and Woodward [5] and
(2) an extension of this scheme to 3D and non-uniform grids.

It should be mentioned that while the present work is motivated by hybrid LES/FDF methods, the scheme is applicable across a range of particle-tracking problems utilizing Cartesian grids.[1] For example, practically all FDF methods are applicable to unsteady PDF methods. Further, with the introduction of improved methods for treating particle diffusion [15], particle-tracking methods are also applicable to DNS. Other applications include multi-phase flows where the drag law requires the local fluid velocity, such as modeling smoke and water-droplet transport in fire simulations [16], tracking aerosols in environmental flows or cells in biological flows, and even describing fluid–structure interaction where structural materials are treated as material points [2,3] which can fragment and interact with the fluid.

The remainder of this paper is organized as follows: in Section 2 we present the relevant governing equations for LES and particle FDF methods and establish the correspondence between the Lagrangian particle position PDF and the Eulerian filtered mass density. Section 3 describes the new reconstruction method. In

[1] Note that a staggered grid is not a requirement for use of the present method. However, the details are described here in terms of a staggered arrangement. For collocated grids the reader should first apply the "Stage 2" correction described in [32] to obtain the final vertex velocities (see Section 3.2).

Section 3.1 we discuss the differences between PPM and PERM. In Section 3.2 we recast the Jenny interpolation in terms of "parabolic edge reconstructions" of the continuous velocity field. The principal advantage of this reformulation is that it allows the method to be readily extended to 3D. An overview of the 3D formulation is presented in Section 3.4 with details given in Appendix A. In Section 4 we describe two Runge–Kutta schemes for advancing the particle position ODE. Note that the velocity reconstruction and the time integration scheme together comprise the "Lagrangian particle advection scheme." In Section 5 we present results for three test cases: (1) a constant-density flow in 2D, (2) a variable-density flow in 2D and (3) a constant-density flow in 3D. Suggestions for future research are given in Section 6 and conclusions are presented in Section 7. Appendix B describes the extension of the scheme to non-uniform grids. An accuracy analysis is presented in Appendix C and in Appendix D we provide a convergence study and thus verification of our 2D code.

## 2. Governing equations

In this section we describe the fields, properties and evolution equations which are relevant to the present work for both the continuous and particles systems. We then discuss the requirements for correspondence between these two systems.

### 2.1. Continuous system

The LES and FDF *model* formulations are a closed set of deterministic equations subject to random initial conditions (i.c.s) and boundary conditions (b.c.s). For a given set of i.c.s and b.c.s, the solution of the transport equations uniquely determines one realization of the filtered field. This is the *continuous system*. Note that the continuous system is the model analog of the exact *fluid system* as discussed in Pope [25].

In LES the "large" and "small" scales are formally defined through a spatial filtering operation. The large scales are simulated explicitly and the effects of the small scales must be modeled. A filtered field is defined by convolution of an instantaneous field with a filter kernel $G(\mathbf{r};\Delta)$ of characteristic filter width $\Delta$. We take $\Delta$ to be constant and uniform and henceforth omit it from the kernel argument list. We take the filter kernel to be positive $(G(\mathbf{r}) \geqslant 0)$, symmetric $(G(\mathbf{r}) = G(-\mathbf{r}))$, and normalized $(\int G(\mathbf{r})\,\mathrm{d}\mathbf{r} = 1)$. For an arbitrary scalar $\phi(\mathbf{x},t)$, the filtered field is given by

$$\langle \phi(\mathbf{x},t)\rangle_\ell \equiv \int G(\mathbf{x}-\mathbf{x}')\phi(\mathbf{x}',t)\,\mathrm{d}\mathbf{x}'. \tag{1}$$

The Eulerian velocity field and fluid mass density are denoted $\mathbf{U}(\mathbf{x},t)$ and $\rho(\mathbf{x},t)$, respectively. In variable-density flows it is usual to work in terms of the Favre-filtered velocity field defined by

$$\langle U_j(\mathbf{x},t)\rangle_L \equiv \frac{\langle \rho(\mathbf{x},t)U_j(\mathbf{x},t)\rangle_\ell}{\langle \rho(\mathbf{x},t)\rangle_\ell}. \tag{2}$$

In hybrid LES/FDF methods $\langle \mathbf{U}(\mathbf{x},t)\rangle_L$ is obtained from the LES solver. Details pertaining to the LES formulation are beyond the scope of the present work. The interested reader is referred to [28].

In the present work we consider the transport of $n_\phi$ random scalar fields $\boldsymbol{\phi}(\mathbf{x},t)$, with $\boldsymbol{\psi} \equiv \{\psi_1,\psi_2,\ldots,\psi_{n_\phi}\}$ being sample space variables for each composition. A starting point for the development of the transport equation for the composition FDF is to consider the fine-grained joint-PDF of compositions, $f'(\boldsymbol{\psi};\mathbf{x},t)$, defined as a product of Dirac delta functions,

$$f'(\boldsymbol{\psi};\mathbf{x},t) \equiv \prod_{\alpha=1}^{n_\phi} \delta(\phi_\alpha[\mathbf{x},t]-\psi_\alpha) \equiv \delta(\boldsymbol{\phi}[\mathbf{x},t]-\boldsymbol{\psi}). \tag{3}$$

Following Jaberi et al. [11], we then define the joint composition filtered mass density function (FMDF) by

$$F_L(\boldsymbol{\psi};\mathbf{x},t) \equiv \int G(\mathbf{x}-\mathbf{x}')\rho(\mathbf{x}',t)f'(\boldsymbol{\psi};\mathbf{x}',t)\,\mathrm{d}\mathbf{x}'. \tag{4}$$

For an arbitrary random function, $Q(\mathbf{x},t)$, the conditionally filtered field is defined by

$$\langle Q|\boldsymbol{\psi}\rangle_L \equiv \frac{\int G(\mathbf{x}-\mathbf{x}')\rho(\mathbf{x}',t)Q(\mathbf{x}',t)\delta(\boldsymbol{\phi}[\mathbf{x}',t]-\boldsymbol{\psi})\,\mathrm{d}\mathbf{x}'}{F_L(\boldsymbol{\psi};\mathbf{x},t)}. \tag{5}$$

Utilizing (5) it can be shown that the FMDF evolves by

$$\frac{\partial F_L}{\partial t}+\frac{\partial}{\partial x_j}[F_L\langle U_j|\boldsymbol{\psi}\rangle_L]=-\frac{\partial}{\partial \psi_\alpha}\left[F_L\left\langle\frac{\mathrm{D}\phi_\alpha}{\mathrm{D}t}\middle|\boldsymbol{\psi}\right\rangle_L\right], \tag{6}$$

where $\mathrm{D}()/\mathrm{D}t\equiv\partial()/\partial t+\mathbf{U}\cdot\nabla()$ is the material derivative and summation is implied over repeated suffixes. The conditional mean on the right-hand side of (6) accounts for chemical reaction (which is closed) and molecular diffusion (which must be modeled).

### 2.2. Particle system

Due to the high dimensionality of $F_L$, the numerical solution of existing models for (6) via a finite difference method is intractable. Instead, it is usual to employ a Lagrangian particle method [23]. A general particle possesses the properties of mass, position, velocity, and composition, denoted $m^*,\mathbf{X}^*(t),\mathbf{U}^*(t)$ and $\boldsymbol{\phi}^*(t)$, respectively. Superscript indices are used to distinguish particles and the superscript asterisk denotes a general particle. We wish to represent the FMDF by an ensemble of such particles within the computational domain. These particles together with their properties and evolution equations for their properties comprise the *particle system*.

For the particle system it is convenient to work in terms of the mass density function, denoted $F^*_{\phi X}(\boldsymbol{\psi},\mathbf{x};t)$. Let $M$ represent the total mass of fluid in a closed or periodic domain so that $M$ is constant. This mass is equally distributed among $N$ particles such that each particle has mass $m=M/N$. The initial position of the $i$th particle, which is random, is $\mathbf{X}^{(i)}(t_0)$. In general we consider the fluid particle to follow a trajectory defined by a random velocity field (even if the filtered field is deterministic the fluctuations are not) and so the current particle position, $\mathbf{X}^{(i)}(t)$, is also random, as is the current particle composition, $\boldsymbol{\phi}^{(i)}(t)$. We first define the discrete mass density function to be

$$F^*_N(\boldsymbol{\psi},\mathbf{x};t)\equiv\frac{M}{N}\sum_{i=1}^{N}\delta(\boldsymbol{\phi}^{(i)}[t]-\boldsymbol{\psi})\delta(\mathbf{X}^{(i)}[t]-\mathbf{x}). \tag{7}$$

The expectation of $F^*_N$ is the mass density function (MDF),

$$\begin{aligned}F^*_{\phi X}(\boldsymbol{\psi},\mathbf{x};t)&\equiv\langle F^*_N(\boldsymbol{\psi},\mathbf{x};t)\rangle,\\&=M\langle\delta(\boldsymbol{\phi}^*[t]-\boldsymbol{\psi})\delta(\mathbf{X}^*[t]-\mathbf{x})\rangle,\\&=Mf^*_{\phi X}(\boldsymbol{\psi},\mathbf{x};t),\\&=Mf^*_X(\mathbf{x};t)f^*_{\phi|X}(\boldsymbol{\psi};t|\mathbf{x}),\end{aligned} \tag{8}$$

where $f^*_{\phi X}$ denotes the joint PDF of particle position and composition, $f^*_X$ is the marginal PDF of particle position, and $f^*_{\phi|X}$ is the PDF of particle composition conditional on particle position.

The particle position and composition, respectively, evolve by

$$\frac{\mathrm{d}X^*_j}{\mathrm{d}t}=U^*_j, \tag{9}$$

$$\frac{\mathrm{d}\phi^*_\alpha}{\mathrm{d}t}=A^*_\alpha, \tag{10}$$

where the composition drift coefficient $A^*_\alpha(t)$ accounts for the effects of chemical reaction (which is closed) and molecular mixing, neither of which are addressed in this paper. In hybrid LES/FDF methods, the particle velocity, $\mathbf{U}^*(t)$, is decomposed into a conditional mean, $\langle\mathbf{U}^*(t)|\mathbf{x}\rangle$, and a fluctuating component, $\mathbf{U}'^*(t)$. The mean is obtained by an interpolation of the LES filtered velocity (which is only available at discrete grid locations) to the particle position. The specifics of this interpolation scheme are a key focus of this paper. The fluc-

tuating velocity is modeled. For a discussion on the modeling considerations the reader is referred to [25]. Within the present work we consider only the "particle-tracking limit" in which $\mathbf{U}'^*(t) = 0$. With the particle position and composition evolving by (9) and (10), for a constant mass system the particle MDF evolves by

$$\frac{\partial F^*_{\phi X}}{\partial t} + \frac{\partial}{\partial x_j}\left[F^*_{\phi X}\langle U^*_j|\mathbf{x},\boldsymbol{\psi}\rangle\right] = -\frac{\partial}{\partial \psi_\alpha}\left[F^*_{\phi X}A^*_\alpha\right]. \tag{11}$$

### 2.3. Correspondence

An important point to appreciate about FDF methods is that the particle properties only indirectly represent statistical samples from (in the present case) the FMDF. The particle property evolution equations themselves imply a certain Fokker–Planck equation for the Lagrangian joint PDF of the particle properties, here represented by the evolution equation for the particle MDF (11). Conceptually, the FMDF, $F_L$, and the particle MDF, $F^*_{\phi X}$, are quite different entities in that the former is defined in terms of a physical-space filter and the latter is based on a statistical expectation. However, the particle system can be designed such that the evolution equations for the FMDF and particle MDF *correspond* (i.e., they have the same form), and hence with corresponding initial and boundary conditions their solutions are identical. It is in this way that the particles, which represent statistical samples from the Lagrangian joint MDF, also represent statistical samples from the FMDF. As we now show, the evolution of the marginal PDF of particle position, $f^*_X$, plays a key role in maintaining correspondence.

Note that the following left–right arrow symbol, $\Longleftrightarrow$, is to be read, "corresponds to." The fundamental correspondence discussed above is

$$F_L(\boldsymbol{\psi};\mathbf{x},t) \Longleftrightarrow F^*_{\phi X}(\boldsymbol{\psi},\mathbf{x};t). \tag{12}$$

Thus, the zeroth moment of the FMDF, which is the filtered density, corresponds to the zeroth moment of the MDF,

$$\langle\rho\rangle_\ell = \int F_L(\boldsymbol{\psi};\mathbf{x},t)\,\mathrm{d}\boldsymbol{\psi} \Longleftrightarrow \int F^*_{\phi X}(\boldsymbol{\psi},\mathbf{x};t)\,\mathrm{d}\boldsymbol{\psi} = M\langle\delta(\mathbf{X}^*[t]-\mathbf{x})\rangle = Mf^*_X(\mathbf{x};t). \tag{13}$$

Integrating (6) over composition space and manipulating the result we obtain the filtered continuity equation

$$\frac{\mathrm{D}_L \ln\langle\rho\rangle_\ell}{\mathrm{D}_L t} = -\nabla\cdot\langle\mathbf{U}\rangle_L, \tag{14}$$

where $\mathrm{D}_L()/\mathrm{D}_L t \equiv \partial()/\partial t + \langle\mathbf{U}\rangle_L\cdot\nabla()$. Similarly, when we integrate (11) over composition space we obtain

$$\frac{\mathrm{D}^* \ln f^*_X}{\mathrm{D}^* t} = -\nabla\cdot\langle\mathbf{U}^*|\mathbf{x}\rangle, \tag{15}$$

where $\mathrm{D}^*()/\mathrm{D}^* t \equiv \partial()/\partial t + \langle\mathbf{U}^*|\mathbf{x}\rangle\cdot\nabla()$.

By comparing (14) and (15), we can see that, with $\langle\mathbf{U}\rangle_L \Longleftrightarrow \langle\mathbf{U}^*|\mathbf{x}\rangle$ and with $f^*_X$ evolving by (15), an initially consistent position distribution (i.e., $Mf^*_X(\mathbf{x},t_0) = \langle\rho(\mathbf{x},t_0)\rangle_\ell$) remains proportional to the filtered mass density with the constant of proportionality being the inverse of the total system mass $M^{-1}$. Thus, as discussed by Pope [23] and Jenny et al. [12], the divergence field that the particles experience during their position evolution (i.e., during the integration of (9)) is of fundamental importance for maintaining the correspondence between the FMDF and the particle MDF.

## 3. Parabolic edge reconstruction

In the sections that follow we describe a variant of the 2D velocity reconstruction of Jenny et al. [12] intended for use with Cartesian staggered grids. With our particular problem motivated as discussed above, we switch to a more general terminology: We work in terms of the staggered-grid velocity data $[\overline{U}_{i+\frac{1}{2},j,k}, \overline{V}_{i,j+\frac{1}{2},k}, \overline{W}_{i,j,k+\frac{1}{2}}]^{\mathrm{T}}$ for all $i,j,k$, which are the fundamental quantities represented on the (3D) grid.

The suffix notation, which represents the physical-space storage location, is described more precisely below. All the parameters of the reconstruction are ultimately related to the staggered-grid velocities which are available from the flow solver only at discrete grid locations. The general problem we wish to solve is: given the discrete velocity data, provide "accurate" reconstructions of the continuous velocity field $\mathbf{U}(\mathbf{x}) = [U(\mathbf{x}), V(\mathbf{x}), W(\mathbf{x})]^{\mathrm{T}}$, to be used in an advection scheme for advancing the particle position. The reconstructed velocity-component fields are denoted $u(\mathbf{x}), v(\mathbf{x})$ and $w(\mathbf{x})$, respectively.

Note that the continuous fields introduced above are merely convenient constructs which we can use to relate the fundamental grid data to the reconstruction. Conceptually, these continuous fields represent the ideal reconstruction (the properties of which are discussed below). In practice (e.g. LES/FDF), there is no known $\mathbf{U}(\mathbf{x})$. The concept of the continuous field is still useful, however, because in tests we can generate a $U(\mathbf{x})$ field, for example, and extract all the $\overline{U}_{i+\frac{1}{2},j,k}$ grid data from it. We can then compare $u(\mathbf{x})$ with $U(\mathbf{x})$ to assess the accuracy of the reconstruction.

There are several aspects to consider regarding the "accuracy" of the *reconstruction*:

(i) Order of accuracy. With $h$ being the grid spacing in the $x$-direction and with $U(x)$ representing the continuous, smooth component field, when $u(x) = U(x) + \mathcal{O}(h^r)$ the reconstruction $u(x)$ is said to be $r$th-order accurate.
(ii) Continuity. Ideally, the reconstructed velocity and divergence fields would be continuous from cell to cell.
(iii) Cell integral constraint. With $S$ representing the surface of a cell and with $\mathbf{n}$ representing the surface normal, the cell integral constraint states that $\int_S \mathbf{u}\cdot\mathbf{n}\,\mathrm{d}S = \mathcal{V}(\nabla_h\cdot\overline{\mathbf{U}})$, where $\mathcal{V}$ is the cell volume and $\nabla_h\cdot()$ represents the discrete divergence operator on the staggered grid (defined below).
(iv) Limiting behavior for the incompressible case. We define the "incompressible case" as that in which the discrete divergence is zero, $\nabla_h\cdot\overline{\mathbf{U}} = 0$, for all cells. In this case we require that the divergence of the reconstructed field is identically zero, $\nabla\cdot\mathbf{u} = 0$, for all $\mathbf{x}$. This is a non-trivial aspect of the accuracy that precludes the use of simple interpolation schemes such as bi(tri)-linear interpolation or even high-order B-splines.

Items (i) and (ii) above are conventional notions of accuracy. The cell integral constraint, Item (iii), is also a standard requirement and is addressed by other reconstruction schemes (see e.g. the "Stage 2" correction of Zhang and Haworth [32]). Controlling the divergence properties, and in particular the divergence in the incompressible limit, was first addressed by Jenny et al. [12]. As we will see, controlling the divergence of the reconstructed field in the incompressible limit requires that we use "cross-component" information in the reconstruction for a given component (e.g. $\overline{V}$-component information for the $u$-component reconstruction). In addition to facilitating a 3D formulation, the method presented here improves upon the accuracy of the Jenny interpolation because the cross-component correction terms are of higher order (details can be found in Appendix C).

The form of the reconstruction developed below is inspired by the piecewise parabolic method (PPM) of Colella and Woodward [5]. The strategy is to cast the continuous velocity field in terms of parabolic edge reconstructions for cell edges parallel with the velocity-component direction. The reconstruction is determined on a cell-by-cell basis, with limited continuity between cells (as discussed below).

In the remainder of this section we first relate the present method to PPM for the 1D case. The full details of the 2D reconstruction are presented next in Section 3.2, followed by a summary of the properties of the scheme. Section 3.3 discusses reconstruction near boundaries. In Section 3.4, we discuss the extension of the method to 3D. Details of the 3D implementation are given in Appendix A.

### 3.1. 1D reconstruction: distinguishing PPM and PERM

In the notation of the gas dynamics community [30] the 1D reconstruction $u_j(x)$ for a cell centered at $x_j$ depends on the cell average velocity, $\bar{u}_j$, the first-order slope, $\Delta_j^{(1)}$ (associated with the first spatial derivative), and the second-order slope, $\Delta_j^{(2)}$ (associated with the second spatial derivative). Consider a 1D grid with

uniform spacing $h$. We define the local cell coordinate $q \equiv \frac{x-x_j}{h} + \frac{1}{2}$ for $x \in [x_j - \frac{h}{2}, x_j + \frac{h}{2}]$ so that $q \in [0,1]$. The PPM reconstruction [5,30] is then

$$u_j(q) = \bar{u}_j + \left(q - \frac{1}{2}\right)\Delta_j^{(1)} + \frac{1}{2}\left[\left(q - \frac{1}{2}\right)^2 - \frac{1}{12}\right]\Delta_j^{(2)}. \tag{16}$$

This reconstruction is designed to obey the cell-average constraint

$$\int_0^1 u_j(q)\,\mathrm{d}q = \bar{u}_j. \tag{17}$$

Note that time has been omitted from the argument list since all fields are located at the same temporal location during the reconstruction.

For the parabolic edge reconstruction developed here we adopt the general PPM formulation for edges that are parallel with the velocity component direction. However, we relax the cell-average constraint because these edges are not the storage locations for the primitive velocity data. The equivalent of the cell-average constraint for the staggered-grid, multi-dimensional case is an integral constraint for the face corresponding to the staggered velocity component; e.g. the "east" face for the $\overline{U}$ component (grids for the multi-dimensional cases are further discussed in Section 3.2, Appendix A, and in Appendix B). As we describe below, in the multi-dimensional case the vertex velocities for a given face are specified to satisfy the integral constraint. With the vertex velocities specified, the reconstruction along the parabolic edge is augmented from the PPM form as follows:

$$u_j(q) = \bar{u}_j + \left(q - \frac{1}{2}\right)\Delta_j^{(1)} + \frac{1}{2}\left[\left(q - \frac{1}{2}\right)^2 - \frac{1}{4}\right]\Delta_j^{(2)} \tag{18}$$

(comparing (16) and (18), note that the factor $\frac{1}{12}$ becomes $\frac{1}{4}$ in the third term on the RHS). A comparison between the reconstructions given by (16) and (18) is shown in Fig. 1. Notice that the edge vertex values,

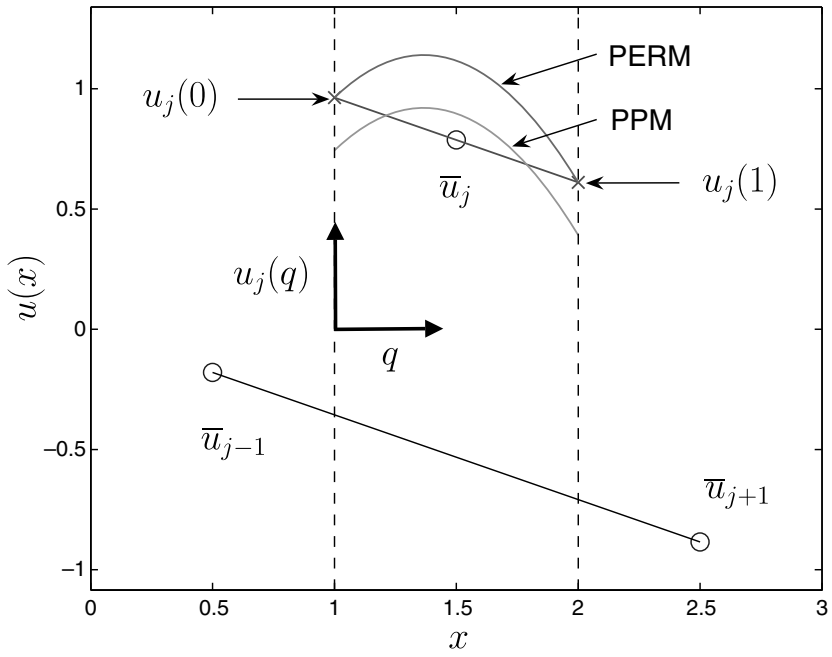


Fig. 1. Comparison between the piecewise parabolic method (PPM) and the parabolic edge reconstruction method (PERM) in 1D. In this example we show a 1D grid with three cells. In the center cell we have reconstructed the continuous velocity field in two ways: with PPM via (16) and with PERM via (18). We specify the cell averages $\bar{u}_j$ arbitrarily for this illustration. The first-order slope is specified using a central approximation to the first derivative (Fromm's method), $\Delta_j^{(1)} = \frac{1}{2}(\bar{u}_{j+1} - \bar{u}_{j-1})$, and the second-order slope is based on a central approximation to the second derivative, $\Delta_j^{(2)} = \bar{u}_{j+1} - 2\bar{u}_j + \bar{u}_{j-1}$. Notice that the PPM reconstruction obeys the cell-average constraint (17) and that the PERM reconstruction interpolates the vertex velocities $u_j(0)$ and $u_j(1)$. In this illustration, the vertex velocities result from the specification of the first-order slope. However, when using PERM in practice for the multi-dimensional case, the vertex velocities are specified and the first-order slope follows.

$u_j(q=0)$ and $u_j(q=1)$, are now independent of the second-order slopes, $\Delta_j^{(2)}$. Thus, $\bar{u}_j$ and $\Delta_j^{(1)}$ are uniquely determined as

$$\bar{u}_j = \frac{1}{2}[u_j(0) + u_j(1)] \tag{19}$$

and

$$\Delta_j^{(1)} = u_j(1) - u_j(0), \tag{20}$$

respectively.

To simplify the formulae, here and throughout the remainder of Section 3 we do not use suffix notation to distinguish velocity components. Hence, in the edge formula (18) the component "$u$" is representative of the "parabolic edge component," i.e., the velocity component for the direction parallel to the edge. The same holds for (19) and (20). Note that for the multi-dimensional case the suffix $j$ identifies a specific edge. For example, in 2D there are two edges, the "north" and "south" edges, which are parallel with the $x$-direction.

To summarize, in the 1D context the differences between PPM and PERM are as follows: With PPM, the cell average, $\bar{u}_j$, is specified; the first- and second-order slopes, $\Delta_j^{(1)}$ and $\Delta_j^{(2)}$, are selected based on accuracy and data variation (see e.g. [30]) considerations; and the form of the reconstruction (16) guarantees that the cell-average constraint (17) is satisfied. In PERM, the edge vertex velocities are specified. As is discussed later, the vertex velocities are responsible for the cell continuity properties. The form of the reconstruction (18) is based on PPM but is augmented such that the vertex values are independent of the second-order slopes. It follows that $\bar{u}_j$ and $\Delta_j^{(1)}$ are uniquely determined by (19) and (20). The second-order slopes are free parameters which we use to control the properties of the divergence of the reconstructed field for the multi-dimensional cases described below.

### 3.2. 2D reconstruction

We first define the grid velocities and divergence constraints upon which the reconstruction is based. We then present the continuous reconstructed field in terms of parabolic edge parameters and show how these parameters are obtained from the discrete grid velocities and divergence constraints.

To avoid confusion between directional indices and grid indices, we denote the directional components as $\mathbf{x} = [x, y]^{\mathrm{T}}$. On a non-uniform, rectangular grid, let $h_i$ and $g_j$ denote the width and height, respectively, of cell $(i, j)$. The center of the cell is located at $(x_i, y_j)$. Similar to the 1D case, we define the local cell coordinates

$$q_i \equiv \frac{x - x_i}{h_i} + \frac{1}{2} \quad \text{for } x \in \left[x_i - \frac{h_i}{2}, x_i + \frac{h_i}{2}\right] \tag{21}$$

and

$$r_j \equiv \frac{y - y_j}{g_j} + \frac{1}{2} \quad \text{for } y \in \left[y_j - \frac{g_j}{2}, y_j + \frac{g_j}{2}\right], \tag{22}$$

such that $q_i \in [0, 1]$ and $r_j \in [0, 1]$.

For simplicity in illustrating the concepts, in this section we consider only uniform, square grids. Hence, we have $g = h$ and we refer to the uniform grid spacing $h$ and local cell coordinates $q$ and $r$. Note that, with minor modifications to the interpolant weights given below, the current method may be applied to non-uniform, rectangular grids without loss of accuracy (extension to non-uniform grids is discussed in Appendix B).

The primitive staggered-grid velocities (also known as face velocities) are denoted by $\overline{\mathbf{U}} = [\overline{U}, \overline{V}]^{\mathrm{T}}$. The staggered storage location for the $\overline{U}$ component for cell $(i, j)$ is $(x_i + \frac{h}{2}, y_j)$. As a shorthand notation for the face velocities we write $\overline{U}(x_i + \frac{h}{2}, y_j)$ as $\overline{U}_{i+\frac{1}{2},j}$. Similarly, the staggered storage location for $\overline{V}_{i,j+\frac{1}{2}}$ is $(x_i, y_j + \frac{h}{2})$; see Fig. 2. In a finite-volume (FV) code, the face velocities represent the surface average of the normal velocity component for the corresponding face. Thus, the discrete divergence for cell $(i, j)$, denoted $(\nabla_h \cdot \overline{\mathbf{U}})_{i,j}$, or simply $\nabla_h \cdot \overline{\mathbf{U}}$ when the cell of interest is understood, is given by

$$(\nabla_h \cdot \overline{\mathbf{U}})_{i,j} \equiv \frac{1}{h}\left[\overline{U}_{i+\frac{1}{2},j} - \overline{U}_{i-\frac{1}{2},j} + \overline{V}_{i,j+\frac{1}{2}} - \overline{V}_{i,j-\frac{1}{2}}\right]. \tag{23}$$

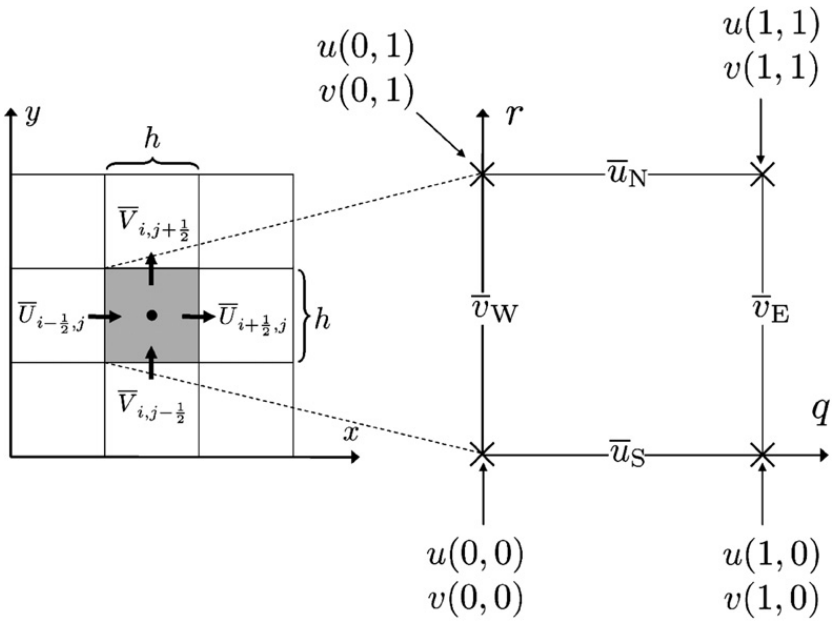


Fig. 2. (Left) Staggered grid in 2D showing the locations and indexing convention for the staggered-grid velocity data. The solid dot at the center of the shaded cell shows the location of the cell center $(x_i, y_j)$ and the solid arrows show the storage locations for the staggered velocities. The "east" face of cell $(i, j)$ is centered at $(x_i + h/2, y_j)$ and this is the storage location for $\overline{U}_{i+1/2,j}$. Similarly, the "west" face is centered at $(x_i - h/2, y_j)$ and stores $\overline{U}_{i-1/2,j}$. The "north" and "south" faces, centered at $(x_i, y_j + h/2)$ and $(x_i, y_j - h/2)$, respectively, store $\overline{V}_{i,j+1/2}$ and $\overline{V}_{i,j-1/2}$. (Right) Center cell in local cell coordinates $(q, r)$, where $q \equiv (x - x_i)/h + 1/2$ for $x \in [x_i - h/2, x_i + h/2]$ so that $q \in [0, 1]$ and $r \equiv (y - y_j)/h + 1/2$ for $y \in [y_j - h/2, y_j + h/2]$ so that $r \in [0, 1]$. The cell vertices are marked by the crosses and the values of the vertex velocity components for the given cell are indicated by $u(0, 0), v(0, 0)$, etc. The parabolic edge velocities are shown for each of the parabolic edges. For example, on the north edge we have $\bar{u}_N$. Although not shown to avoid clutter, each of these edges is also assigned a first- and second-order slope. All the edge parameters for the reconstruction are ultimately related to the staggered-grid velocity data.

We approximate the value of the divergence at cell vertices by a weighted restriction of the surrounding cell divergences. For later utility, the vertex divergence is scaled by the grid spacing and indexed by the physical-space cell vertex coordinate. Thus, for a given cell, the vector of interpolated vertex divergence values is

$$\boldsymbol{\Theta} = \left[\theta_{i-\frac{1}{2},j-\frac{1}{2}}\ \theta_{i-\frac{1}{2},j+\frac{1}{2}}\ \theta_{i+\frac{1}{2},j-\frac{1}{2}}\ \theta_{i+\frac{1}{2},j+\frac{1}{2}}\right]^{\mathrm{T}}, \tag{24}$$

where, for example,

$$\theta_{i-\frac{1}{2},j-\frac{1}{2}} = \frac{h}{4}[(\nabla_h \cdot \overline{\mathbf{U}})_{i,j} + (\nabla_h \cdot \overline{\mathbf{U}})_{i-1,j} + (\nabla_h \cdot \overline{\mathbf{U}})_{i,j-1} + (\nabla_h \cdot \overline{\mathbf{U}})_{i-1,j-1}]. \tag{25}$$

Within the $(i, j)$ cell, the 2D reconstruction of the continuous velocity is

$$\begin{aligned} u(q,r) = {} & (1-r)\left[\bar{u}_S + \left(q-\frac{1}{2}\right)\Delta^{(1)}_{u,S} + \frac{1}{2}\left\{\left(q-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{u,S}\right] \\ & + r\left[\bar{u}_N + \left(q-\frac{1}{2}\right)\Delta^{(1)}_{u,N} + \frac{1}{2}\left\{\left(q-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{u,N}\right] \end{aligned} \tag{26}$$

and

$$\begin{aligned} v(q,r) = {} & (1-q)\left[\bar{v}_W + \left(r-\frac{1}{2}\right)\Delta^{(1)}_{v,W} + \frac{1}{2}\left\{\left(r-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{v,W}\right] \\ & + q\left[\bar{v}_E + \left(r-\frac{1}{2}\right)\Delta^{(1)}_{v,E} + \frac{1}{2}\left\{\left(r-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{v,E}\right]. \end{aligned} \tag{27}$$

The local divergence of the reconstructed field, $h(\nabla \cdot \mathbf{u}) = \frac{\partial u}{\partial q} + \frac{\partial v}{\partial r}$, is

$$\begin{aligned} h(\nabla \cdot \mathbf{u}) &\equiv \phi(q,r) \\ &= (1-r)\left[\Delta^{(1)}_{u,\mathrm{S}} + \left(q-\frac{1}{2}\right)\Delta^{(2)}_{u,\mathrm{S}}\right] + r\left[\Delta^{(1)}_{u,\mathrm{N}} + \left(q-\frac{1}{2}\right)\Delta^{(2)}_{u,\mathrm{N}}\right] + (1-q)\left[\Delta^{(1)}_{v,\mathrm{W}} + \left(r-\frac{1}{2}\right)\Delta^{(2)}_{v,\mathrm{W}}\right] \\ &\quad + q\left[\Delta^{(1)}_{v,\mathrm{E}} + \left(r-\frac{1}{2}\right)\Delta^{(2)}_{v,\mathrm{E}}\right], \end{aligned} \tag{28}$$

where we have introduced the shorthand notation $\phi(q,r)$ with $q \in [0,1]$ and $r \in [0,1]$ to represent the continuous divergence within the cell (multiplied by $h$). It remains that we specify the parameters for each edge of the reconstruction: $\bar{u}_{\alpha,j}$, $\Delta^{(1)}_{\alpha,j}$ and $\Delta^{(2)}_{\alpha,j}$, for parabolic-edge component $\alpha$ on edge $j$. As previously mentioned, $\bar{u}_{\alpha,j}$ and $\Delta^{(1)}_{\alpha,j}$ are uniquely determined by the edge vertex velocities, which are specified below to satisfy cell continuity constraints. The second-order slope $\Delta^{(2)}_{\alpha,j}$ is then chosen to achieve accuracy and to control the cell divergence properties.

We now turn to the specification of the vertex velocities. For the faces corresponding to the staggered-grid velocity components we specify a linear velocity reconstruction that satisfies the face integral constraint. Consider the east face ($q = 1$) of the 2D cell centered at $(x_i, y_j)$. The linear velocity reconstruction has the form

$$u(1,r) = \overline{U}_{i+\frac{1}{2},j} + \left(r-\frac{1}{2}\right)\Delta^{(1)}_{u,\mathrm{E}}. \tag{29}$$

Note that the face integral constraint

$$\int_0^1 u(1,r)\,\mathrm{d}r = \overline{U}_{i+\frac{1}{2},j} \tag{30}$$

is satisfied for any specification of $\Delta^{(1)}_{u,\mathrm{E}}$. We choose to use a central difference approximation to the first-order slope, also known as the Fromm slope [8,30],

$$\Delta^{(1)}_{u,\mathrm{E}} = \frac{1}{2}\left[\overline{U}_{i+\frac{1}{2},j+1} - \overline{U}_{i+\frac{1}{2},j-1}\right]. \tag{31}$$

We make this choice because (a) the reconstruction (29) is then second-order accurate and (b) dispersion errors (i.e., wiggles) are permissible for the velocity field in an LES context.[2]

An alternate view of the Fromm face reconstruction, which is more easily extended to the 3D case, is to first generate temporary vertex values by a linear interpolation of the neighboring face values and then to add a constant shift to these temporary values to enforce the face integral constraint. The temporary vertex values, denoted by the hat symbol, for the east face are given by

$$\hat{u}_{i+\frac{1}{2},j-\frac{1}{2}} = \frac{1}{2}\left(\overline{U}_{i+\frac{1}{2},j} + \overline{U}_{i+\frac{1}{2},j-1}\right), \tag{32}$$

$$\hat{u}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\left(\overline{U}_{i+\frac{1}{2},j} + \overline{U}_{i+\frac{1}{2},j+1}\right) \tag{33}$$

and for the west face we have

$$\hat{u}_{i-\frac{1}{2},j-\frac{1}{2}} = \frac{1}{2}\left(\overline{U}_{i-\frac{1}{2},j} + \overline{U}_{i-\frac{1}{2},j-1}\right), \tag{34}$$

$$\hat{u}_{i-\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\left(\overline{U}_{i-\frac{1}{2},j} + \overline{U}_{i-\frac{1}{2},j+1}\right). \tag{35}$$

[2] Another way to state point (b) is that the subgrid velocity field is not subject to the same boundedness constraints as a scalar field such as mass fraction, which must remain in $[0,1]$ to be realizable. Note that any choice of the first-order slope will suffice to maintain continuity of the velocity-component field in the component direction. For example, Jenny et al. [12] choose to use a MINMOD slope limiter, which is advantageous for convergence in RANS simulations.

Let $\hat{\bar{u}}_E \equiv \frac{1}{2}(\hat{u}_{i+\frac{1}{2},j-\frac{1}{2}} + \hat{u}_{i+\frac{1}{2},j+\frac{1}{2}})$ and $\hat{\bar{u}}_W \equiv \frac{1}{2}(\hat{u}_{i-\frac{1}{2},j-\frac{1}{2}} + \hat{u}_{i-\frac{1}{2},j+\frac{1}{2}})$ denote the averages of the temporary vertex values on the east and west faces, respectively. The final vertex values for the $u$ component are then specified as[3]

$$u(1,0) = \hat{u}_{i+\frac{1}{2},j-\frac{1}{2}} + \overline{U}_{i+\frac{1}{2},j} - \hat{\bar{u}}_E, \tag{36}$$

$$u(1,1) = \hat{u}_{i+\frac{1}{2},j+\frac{1}{2}} + \overline{U}_{i+\frac{1}{2},j} - \hat{\bar{u}}_E, \tag{37}$$

$$u(0,0) = \hat{u}_{i-\frac{1}{2},j-\frac{1}{2}} + \overline{U}_{i-\frac{1}{2},j} - \hat{\bar{u}}_W, \tag{38}$$

$$u(0,1) = \hat{u}_{i-\frac{1}{2},j+\frac{1}{2}} + \overline{U}_{i-\frac{1}{2},j} - \hat{\bar{u}}_W. \tag{39}$$

Note that the linear face reconstruction implied by the vertex values (36) and (37) automatically satisfies the integral constraint (30). The analogous condition holds for the west-face integral constraint; i.e., $\int_0^1 u(0,r)\,dr = \overline{U}_{i-\frac{1}{2},j}$. Additionally, cell to cell continuity of the velocity-component field in the component direction is achieved since, for example $(u[1,0])_{i,j} = (u[0,0])_{i+1,j}$ and $(u[1,1])_{i,j} = (u[0,1])_{i+1,j}$. In other words, the $u$ vertex values on the east face of cell $(i,j)$ match the $u$ vertex values on the west face of cell $(i+1,j)$.

The $v$-component vertex values are specified analogously. The temporary vertex values on the north face are

$$\hat{v}_{i-\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\left(\overline{V}_{i,j+\frac{1}{2}} + \overline{V}_{i-1,j+\frac{1}{2}}\right), \tag{40}$$

$$\hat{v}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\left(\overline{V}_{i,j+\frac{1}{2}} + \overline{V}_{i+1,j+\frac{1}{2}}\right) \tag{41}$$

and for the south face we have

$$\hat{v}_{i-\frac{1}{2},j-\frac{1}{2}} = \frac{1}{2}\left(\overline{V}_{i,j-\frac{1}{2}} + \overline{V}_{i-1,j-\frac{1}{2}}\right), \tag{42}$$

$$\hat{v}_{i+\frac{1}{2},j-\frac{1}{2}} = \frac{1}{2}\left(\overline{V}_{i,j-\frac{1}{2}} + \overline{V}_{i+1,j-\frac{1}{2}}\right). \tag{43}$$

The final vertex values are then given by

$$v(0,1) = \hat{v}_{i-\frac{1}{2},j+\frac{1}{2}} + \overline{V}_{i,j+\frac{1}{2}} - \hat{\bar{v}}_N, \tag{44}$$

$$v(1,1) = \hat{v}_{i+\frac{1}{2},j+\frac{1}{2}} + \overline{V}_{i,j+\frac{1}{2}} - \hat{\bar{v}}_N, \tag{45}$$

$$v(0,0) = \hat{v}_{i-\frac{1}{2},j-\frac{1}{2}} + \overline{V}_{i,j-\frac{1}{2}} - \hat{\bar{v}}_S, \tag{46}$$

$$v(1,0) = \hat{v}_{i+\frac{1}{2},j-\frac{1}{2}} + \overline{V}_{i,j-\frac{1}{2}} - \hat{\bar{v}}_S, \tag{47}$$

where $\hat{\bar{v}}_N \equiv \frac{1}{2}(\hat{v}_{i-\frac{1}{2},j+\frac{1}{2}} + \hat{v}_{i+\frac{1}{2},j+\frac{1}{2}})$ and $\hat{\bar{v}}_S \equiv \frac{1}{2}(\hat{v}_{i-\frac{1}{2},j-\frac{1}{2}} + \hat{v}_{i+\frac{1}{2},j-\frac{1}{2}})$. Note that, due to the specification of the $v$-component vertex values (44)–(47), the integral constraints $\int_0^1 v(q,1)\,dq = \overline{V}_{i,j+\frac{1}{2}}$ and $\int_0^1 v(q,0)\,dq = \overline{V}_{i,j-\frac{1}{2}}$ are satisfied for the north and south faces, respectively, and also that the reconstruction achieves cell to cell continuity of the $v$-component field in the $y$-direction.

With the vertex velocities specified, the parabolic-edge parameters are given by analogy to (19) and (20). To be explicit, we have

$$\bar{u}_S = \frac{1}{2}[u(1,0) + u(0,0)], \tag{48}$$

$$\bar{u}_N = \frac{1}{2}[u(1,1) + u(0,1)], \tag{49}$$

$$\bar{v}_W = \frac{1}{2}[v(0,1) + v(0,0)], \tag{50}$$

$$\bar{v}_E = \frac{1}{2}[v(1,1) + v(1,0)] \tag{51}$$

[3] The constant shift, e.g. $\overline{U}_{i+\frac{1}{2},j} - \hat{\bar{u}}_E$ on the east face, is effectively the Cartesian staggered-grid equivalent of the "Stage 2" velocity correction implemented by Zhang and Haworth [32].

and

$$\Delta^{(1)}_{u,\mathrm{S}} = u(1,0) - u(0,0), \tag{52}$$

$$\Delta^{(1)}_{u,\mathrm{N}} = u(1,1) - u(0,1), \tag{53}$$

$$\Delta^{(1)}_{v,\mathrm{W}} = v(0,1) - v(0,0), \tag{54}$$

$$\Delta^{(1)}_{v,\mathrm{E}} = v(1,1) - v(1,0). \tag{55}$$

We now turn to the issue of specifying the second-order slopes. By evaluating (28) at the vertices, we find the vertex values of the reconstructed divergence are

$$\phi(0,0) = \Delta^{(1)}_{u,\mathrm{S}} + \Delta^{(1)}_{v,\mathrm{W}} - \frac{1}{2}\Delta^{(2)}_{u,\mathrm{S}} - \frac{1}{2}\Delta^{(2)}_{v,\mathrm{W}}, \tag{56}$$

$$\phi(0,1) = \Delta^{(1)}_{u,\mathrm{N}} + \Delta^{(1)}_{v,\mathrm{W}} - \frac{1}{2}\Delta^{(2)}_{u,\mathrm{N}} + \frac{1}{2}\Delta^{(2)}_{v,\mathrm{W}}, \tag{57}$$

$$\phi(1,0) = \Delta^{(1)}_{u,\mathrm{S}} + \Delta^{(1)}_{v,\mathrm{E}} + \frac{1}{2}\Delta^{(2)}_{u,\mathrm{S}} - \frac{1}{2}\Delta^{(2)}_{v,\mathrm{E}}, \tag{58}$$

$$\phi(1,1) = \Delta^{(1)}_{u,\mathrm{N}} + \Delta^{(1)}_{v,\mathrm{E}} + \frac{1}{2}\Delta^{(2)}_{u,\mathrm{N}} + \frac{1}{2}\Delta^{(2)}_{v,\mathrm{E}}. \tag{59}$$

We observe that (56)–(59) are *not* linearly independent, because

$$\frac{1}{4}[\phi(0,0) + \phi(0,1) + \phi(1,0) + \phi(1,1)] = \frac{1}{2}[\Delta^{(1)}_{u,\mathrm{S}} + \Delta^{(1)}_{u,\mathrm{N}} + \Delta^{(1)}_{v,\mathrm{W}} + \Delta^{(1)}_{v,\mathrm{E}}] = h(\nabla_h \cdot \overline{\mathbf{U}}), \tag{60}$$

independent of the second-order slopes.

We can write (56)–(59) in matrix notation as

$$\mathbf{\Phi} = A\mathbf{\Delta}^{(1)} + B\mathbf{\Delta}^{(2)}, \tag{61}$$

where

$$\mathbf{\Phi} = [\phi(0,0) \quad \phi(0,1) \quad \phi(1,0) \quad \phi(1,1)]^{\mathrm{T}}, \tag{62}$$

$$\mathbf{\Delta}^{(1)} = [\Delta^{(1)}_{u,\mathrm{S}} \quad \Delta^{(1)}_{u,\mathrm{N}} \quad \Delta^{(1)}_{v,\mathrm{W}} \quad \Delta^{(1)}_{v,\mathrm{E}}]^{\mathrm{T}}, \tag{63}$$

$$\mathbf{\Delta}^{(2)} = [\Delta^{(2)}_{u,\mathrm{S}} \quad \Delta^{(2)}_{u,\mathrm{N}} \quad \Delta^{(2)}_{v,\mathrm{W}} \quad \Delta^{(2)}_{v,\mathrm{E}}]^{\mathrm{T}}, \tag{64}$$

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{65}$$

and

$$B = \frac{1}{2}\begin{bmatrix} -1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \end{bmatrix}. \tag{66}$$

Note that $B$ has a rank deficiency of one.

Ideally, we would have

$$\mathbf{\Theta} = A\mathbf{\Delta}^{(1)} + B\mathbf{\Delta}^{(2)}. \tag{67}$$

In other words, we would have the vertex values $\mathbf{\Phi}$ of the reconstructed divergence equal to the interpolated vertex divergences $\mathbf{\Theta}$, which would imply cell to cell continuity of the reconstructed divergence field. However, because $B$ is rank deficient, given $\mathbf{\Theta}$ and $\mathbf{\Delta}^{(1)}$, in general a value of $\mathbf{\Delta}^{(2)}$ cannot be found to satisfy (67).

Alternatively, we seek to minimize (in some sense) the difference between the reconstructed divergence, which may be discontinuous from cell to cell, and the interpolated divergence,[4] which is continuous from cell to cell.

With this in mind, we propose the following specification of the second-order slopes. Let the second-order slopes be decomposed into

$$\boldsymbol{\Delta}^{(2)} = \hat{\boldsymbol{\Delta}} + \boldsymbol{\delta}. \tag{68}$$

The equation for the vertex values of the reconstruction may now be written as

$$\boldsymbol{\Phi} = A\boldsymbol{\Delta}^{(1)} + B\hat{\boldsymbol{\Delta}} + B\boldsymbol{\delta}. \tag{69}$$

We specify $\hat{\boldsymbol{\Delta}}$ based on an estimate of the second spatial derivative along a given edge. For example, on the south edge we have

$$h^2\left(\frac{\partial^2 u}{\partial x^2}\right)_{q=\frac{1}{2},r=0} \approx \hat{\Delta}_{u,\mathrm{S}} = \frac{1}{2}\left[\hat{u}_{i+\frac{3}{2},j-\frac{1}{2}} - \hat{u}_{i+\frac{1}{2},j-\frac{1}{2}} - \hat{u}_{i-\frac{1}{2},j-\frac{1}{2}} + \hat{u}_{i-\frac{3}{2},j-\frac{1}{2}}\right]. \tag{70}$$

Note that the $\hat{u}$ values are readily available in practice since they are also the "temporary vertex values," as in (32), for example. The correction, $\boldsymbol{\delta}$, is then specified by the least-squares/minimum-norm (LSMN) solution

$$\boldsymbol{\delta} = B^{+}(\boldsymbol{\Theta} - A\boldsymbol{\Delta}^{(1)} - B\hat{\boldsymbol{\Delta}}), \tag{71}$$

where the (Moore–Penrose) pseudo-inverse [6,17,20,21] of $B$ is

$$B^{+} = \frac{1}{4}\begin{bmatrix} -3 & -1 & 3 & 1 \\ -1 & -3 & 1 & 3 \\ -3 & 3 & -1 & 1 \\ -1 & 1 & -3 & 3 \end{bmatrix}. \tag{72}$$

There are an infinite number of least-squares solutions to (69); i.e., choices for $\boldsymbol{\delta}$ which minimize the Euclidean norm $\|\boldsymbol{\Phi} - \boldsymbol{\Theta}\|_2$. The minimum-norm solution (71) is the unique solution in the family of least-squares solutions that also minimizes $\|\boldsymbol{\delta}\|_2$.

Substituting (71) back into (69) we obtain

$$\boldsymbol{\Phi} = A\boldsymbol{\Delta}^{(1)} + B\hat{\boldsymbol{\Delta}} + BB^{+}(\boldsymbol{\Theta} - A\boldsymbol{\Delta}^{(1)} - B\hat{\boldsymbol{\Delta}}) = BB^{+}\boldsymbol{\Theta} + (I - BB^{+})A\boldsymbol{\Delta}^{(1)} + \underbrace{(B - \underbrace{BB^{+}B}_{B})}_{0}\hat{\boldsymbol{\Delta}}, \tag{73}$$

where $I$ is the identity matrix and, in the second step, the third term on the RHS is zero due to the properties of the Moore–Penrose pseudo-inverse. Hence, after subtracting $\boldsymbol{\Theta}$ from both sides, we find

$$\boldsymbol{\Phi} - \boldsymbol{\Theta} = (I - BB^{+})(A\boldsymbol{\Delta}^{(1)} - \boldsymbol{\Theta}). \tag{74}$$

Note that

$$(I - BB^{+})A = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{75}$$

and also

$$I - BB^{+} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{76}$$

[4] Here the "interpolated divergence" refers to a continuous (e.g. bilinear) interpolation of the cell-centered divergences. Note that the "weighted restriction" which results in $\boldsymbol{\Theta}$ is simply a bilinear interpolant evaluated at the vertex locations for a uniform grid.

Hence, by utilizing (60), we find that at the "south–west" vertex, for example, (74) yields

$$\phi(0,0) = \theta_{i-\frac{1}{2},j-\frac{1}{2}} + h(\nabla_h \cdot \overline{\mathbf{U}}) - \frac{1}{4}\left[\theta_{i-\frac{1}{2},j-\frac{1}{2}} + \theta_{i-\frac{1}{2},j+\frac{1}{2}} + \theta_{i+\frac{1}{2},j-\frac{1}{2}} + \theta_{i+\frac{1}{2},j+\frac{1}{2}}\right], \tag{77}$$

with similar expressions at the other vertices. Hence, due to linearity, the divergence of the reconstructed field is everywhere zero for the incompressible (divergence free) case.

In short, we specify the second-order slopes by (68). The properties of the resulting 2D reconstruction are:

1. For a given component, the reconstructed field in the component direction is continuous, piecewise parabolic and formally second-order accurate (further discussed below and in Appendix C.1).
2. In the component-normal direction the reconstructed field is piecewise linear and second-order accurate, but may be discontinuous from cell to cell.
3. In the component direction, the spatial derivative of that component is piecewise linear and second-order accurate.
4. In the component-normal direction, the spatial derivative is piecewise constant and first-order accurate.
5. The surface integral of the reconstructed field is consistent with the discrete cell divergence obtained from the staggered-grid velocity data, i.e., $\int_S \mathbf{u} \cdot \mathbf{n}\,\mathrm{d}S = \mathcal{V}(\nabla_h \cdot \overline{\mathbf{U}})$, where $S$ represents the cell surface with surface normal $\mathbf{n}$, and $\mathcal{V}$ is the cell volume.
6. The divergence of the reconstructed field is bilinear within a cell, but may be discontinuous from cell to cell.
7. Due to Items 5 and 6, the reconstructed divergence evaluated at the cell center matches the discrete cell divergence, i.e., $\phi(\frac{1}{2},\frac{1}{2}) = h(\nabla_h \cdot \overline{\mathbf{U}})$.
8. The Euclidean norm $\|\mathbf{\Phi} - \mathbf{\Theta}\|_2$ between the reconstructed and interpolated divergences at the cell vertices is minimized.
9. Due to Item 8, the divergence of the reconstructed field is everywhere zero in the constant-density case (in which $\mathbf{\Theta}$ is zero).

Regarding Item 1 above, the PERM velocity is formally second-order accurate due to the linear interpolation used in the component-normal direction (see Appendix C.1). However, as we show in Appendix D.1, for the test problems considered here (which are complex and general in terms of combustion modeling) the scheme exhibits third-order behavior for coarse grid resolution, which is more relevant in practice than the formal order for vanishing $h$.

For uniform, rectangular grids (i.e., $h = \alpha g$, where $\alpha$ is a constant) the scheme requires little modification. Only the definitions of the local cell coordinates are affected, as described by (21) and (22). For non-uniform grids, the scheme retains all the properties listed above but requires modification of the scaling for the interpolated vertex divergence, modification of the weights used to compute the temporary vertex values, and modification of the weights used in the formula for the specified second-order slopes. Details of these modifications are discussed in Appendix B.

### 3.3. Reconstruction near boundaries

Little modification of the method is required near boundaries to achieve the same order of accuracy. Here we consider simple boundaries that are aligned with a cell face. In 2D, consider an example where the south edge of the cell represents a wall boundary. Presumably, the values of the velocity components are known on the boundary. For example, the velocities will be zero if the no slip and impermeability conditions are to be satisfied. Clearly, the PERM parameters for the south edge are trivially zero. However, the $u$ component final vertex values for the north edge *are* affected by the boundary. For a given cell $(i,j)$, the final vertex values at the wall are $u(0,0) = u(1,0) = 0$. In order to satisfy the face average constraint on the east and west faces, the final vertex values for the north edge must be specified as

$$u(0,1) = u(0,0) + 2\overline{U}_{i-\frac{1}{2},j}, \tag{78}$$

$$u(1,1) = u(1,0) + 2\overline{U}_{i+\frac{1}{2},j}. \tag{79}$$

Note that if the velocity near the boundary is not sufficiently resolved, Eqs. (78) and (79) may lead to large discontinuities in the reconstruction. This is true of most reconstruction methods that specify the boundary value and then enforce the face average constraint; if continuity is achieved, then this comes at the cost of large dispersion errors in the reconstruction (see e.g. [14]). The remaining $u$ component north edge PERM parameters (edge velocity $\bar{u}_{\mathrm{N}}$, first-order slope $\mathbf{\Lambda}^{(1)}_{u,\mathrm{N}}$, and specified second-order slope $\hat{\mathbf{\Lambda}}_{u,\mathrm{N}}$) are found in the usual way described above.

For the $v$ component in this example, the situation is slightly different. The final vertex values are found as usual, and hence the edge velocities ($\bar{v}_{\mathrm{W}}$ and $\bar{v}_{\mathrm{E}}$) and first-order slopes ($\Delta^{(1)}_{v,\mathrm{W}}$ and $\Delta^{(1)}_{v,\mathrm{E}}$) are determined as shown above (see (54) and (55)). However, the specified second-order slopes must be obtained from a one-sided difference. To maintain accuracy, we require a second-order accurate representation of the second spatial derivative of $v$ in the wall-normal direction evaluated at the edge center. For example, on the west face we may use

$$\hat{\Delta}_{v,\mathrm{W}} = h^2\left(\frac{\partial^2 v}{\partial y^2}\right)_{q=0,r=\frac{1}{2}} = v(0,0) - 2\hat{v}_{i-\frac{1}{2},j+\frac{1}{2}} + \hat{v}_{i-\frac{1}{2},j+\frac{3}{2}} + \mathcal{O}(h^4). \tag{80}$$

With the edge parameters specified, the correction $\boldsymbol{\delta}$ is obtained from (71); the second-order slopes are obtained from (68); and the reconstruction within the cell is given by (26) and (27). Note that same principles may be applied to a corner cell.

### 3.4. 3D reconstruction

One of the strengths of the new formulation is the ease with which it extends to 3D. Details of the 3D reconstruction are presented in Appendix A. The key difference between the 2D and 3D cases is that in 3D we now have 12 parabolic edges and 8 vertices. Hence, in the 3D matrix equation similar to (69), $A$ and $B$ are now $8 \times 12$ matrices and $B$ has a column rank deficiency of 5. The LSMN solution analogous to (71) is used for the second-order slope correction and the resulting scheme inherits all the properties of the 2D scheme listed above.

## 4. Time integration

In this section we describe two simple methods for integrating (9) from time $t^n$ to time $t^{n+1} \equiv t^n + \Delta t$. The two-stage Runge–Kutta scheme (RK2) [10], also known as "Modified Euler," is given by

$$X^*_j(t^{(1)}) = X^*_j(t^n) + \Delta t\, u_j(\mathbf{X}^*[t^n], t^n), \tag{81}$$

$$X^*_j(t^{n+1}) = \frac{1}{2}X^*_j(t^n) + \frac{1}{2}[X^*_j(t^{(1)}) + \Delta t\, u_j(\mathbf{X}^*[t^{(1)}], t^{n+1})], \tag{82}$$

where $t^{(1)}$ represents an intermediate state in the integration and the suffix $j$ is a directional index. Note that RK2 utilizes the velocity reconstruction described in the previous sections to estimate the particle velocity, $\mathbf{U}^*(t) = \mathbf{u}(\mathbf{X}^*[t], t)$. Additionally, the RK2 scheme requires the staggered-grid velocity data to be advanced ahead of the particle positions so that the $\mathbf{u}(\mathbf{x}, t^{n+1})$ reconstruction can be evaluated. A practical matter to consider when applying RK2 to a particle method is that between the two stages of the scheme one incurs the cost of determining the cell location of a given particle. However, in considering the efficiency of the advection scheme (i.e., the amount of work for a specified error tolerance) this cost is well-justified.

For reference, we also show a three-stage Runge–Kutta scheme (RK3) [10] which we use as an accurate baseline for comparison in the test cases in Section 5,

$$X^*_j(t^{(1)}) = X^*_j(t^n) + \Delta t\, U_j(\mathbf{X}^*[t^n], t^n), \tag{83}$$

$$X^*_j(t^{(2)}) = \frac{3}{4}X^*_j(t^n) + \frac{1}{4}\left[X^*_j(t^{(1)}) + \Delta t\, U_j(\mathbf{X}^*[t^{(1)}], t^{n+1})\right], \tag{84}$$

$$X^*_j(t^{n+1}) = \frac{1}{3}X^*_j(t^n) + \frac{2}{3}\left[X^*_j(t^{(2)}) + \Delta t\, U_j\left(\mathbf{X}^*[t^{(2)}], t^n + \frac{\Delta t}{2}\right)\right]. \tag{85}$$

Note here that the particle velocity is taken from the continuous Eulerian field, $\mathbf{U}(\mathbf{x}, t)$. The exact fields used for the test cases are presented in the Section 5.

## 5. Test cases

In this section we present three test cases. The first uses a solenoidal velocity field and hence the ideal particle mass distribution is uniform. In the second test case we introduce a potential flow component with a finite divergence. In both cases the advection schemes are compared against an accurate baseline case. The second test case, which is more applicable to combustion simulations, illustrates the need for improved number density control algorithms in LES/FDF methods. In the third case we test the 3D implementation of the method. This case highlights that the RK2 time integration locally degenerates to zeroth-order accuracy at the cell discontinuities and hence the overall advection scheme is formally first-order accurate in time (this issue is further addressed in [26]). However, as shown below, with a reasonably small time step (meaning not too small for practical purposes) the temporal error is of the same order as the spatial error for the problem considered here.

### 5.1. Problem 1: Solenoidal, rotational flow

We consider the solenoidal, rotational flow defined by the velocity field [1]

$$U(x, y, t) = 1 - 2\cos(x - t)\sin(y - t), \tag{86}$$

$$V(x, y, t) = 1 + 2\sin(x - t)\cos(y - t), \tag{87}$$

on a square periodic domain of side $L = 2\pi$. The domain is uniformly divided into $N_x^2$ cells of side $h = L/N_x$, where $N_x$ is the number of cells in each direction. Within each cell the initial positions of $N_c$ particles are chosen at random from a uniform distribution using Latin hypercube sampling. The initial particle positions for a $4 \times 4$ grid with $N_c = 400$ particles per cell are shown in Fig. 3. The particles in the lower left corner cell are

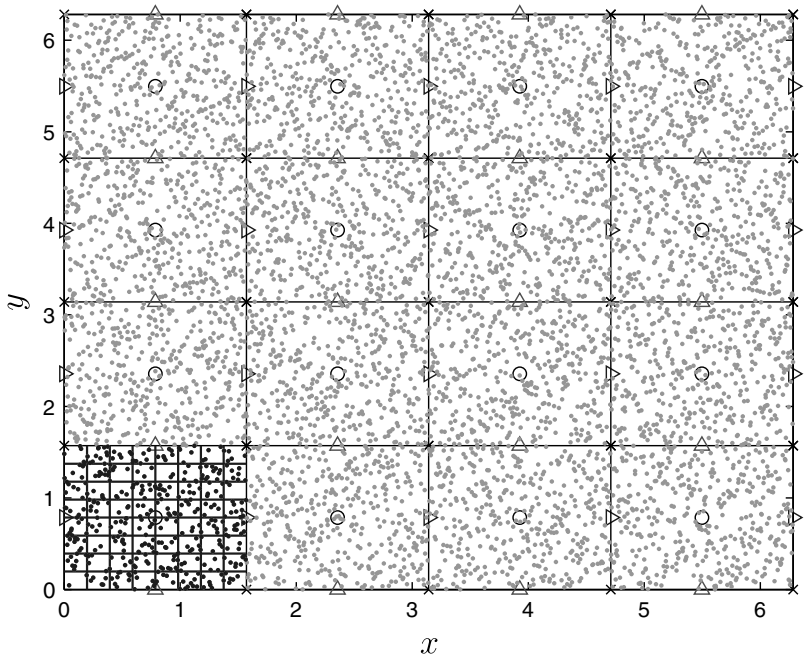


Fig. 3. Initial condition for a $4 \times 4$ grid with $N_c = 400$ particles per cell. The small dots represent the initial particle positions $\mathbf{X}^*(t_0)$. The periodic domain is a square of side $L = 2\pi$. Particle positions are chosen at random from a uniform distribution in each cell using Latin hypercube sampling. The darker-colored particles in the lower left cell are to be used as marker particles. The lower-left cell is subdivided into $8 \times 8$ fine cells which are used to construct the histogram of the subcell particle number density. The right-facing triangles represent the staggered storage locations for the $\overline{U}$-component data and the top-facing triangles represent the staggered storage locations for the $\overline{V}$-component data. The circles in the cell center indicate the storage location for the pressure (not used) in the staggered-grid arrangement and the crosses indicate cell vertices.

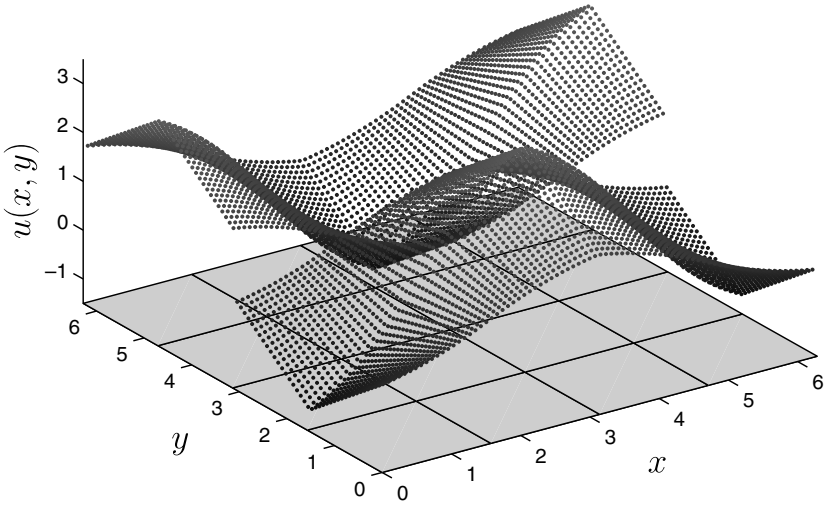


Fig. 4. The initial condition for the $u$ component of velocity interpolated to the particle positions for a $4 \times 4$ grid with 400 particles per cell. The particles are uniformly spaced here in order to more clearly visualize the reconstruction; other results presented in this paper do not use this initial particle position distribution. The particle shading is proportional to the velocity component value, with black representing the minimum and gray representing the maximum. Notice that the reconstruction is continuous and piecewise parabolic in the velocity-component direction, the $x$-direction in this case. The reconstruction is piecewise linear in the direction normal to the velocity component. The discontinuities reside at the cell boundaries.

colored dark gray to be used as marker particles throughout the simulation. The resulting PERM reconstruction for the initial $u$ component is shown in Fig. 4, where the particles are positioned with a uniform spacing to more easily visualize the reconstruction. The staggered-grid velocity components (e.g. $\overline{U}_{i+\frac{1}{2},j}$), which are needed for the subgrid velocity reconstruction described in Section 3.2, are obtained by sampling the exact solution, (86) and (87), at the staggered positions. Note that the staggered velocities are discretely divergence free. The $u$ field in Fig. 4 is piecewise parabolic in the $x$-direction (see Fig. 6) and piecewise linear in the $y$-direction (see Fig. 7). For comparison, in Fig. 5 the same velocity field, now sampled at the cell vertex locations, is interpolated to the particle positions using bicubic splines. The spline interpolation is continuous from cell to cell but does not satisfy the face integral constraints. Note that the uniform spacing pertains to Figs. 4 and 5 only and is not used in any of the simulations. Both bicubic spline interpolation (continuous, based on the vertex

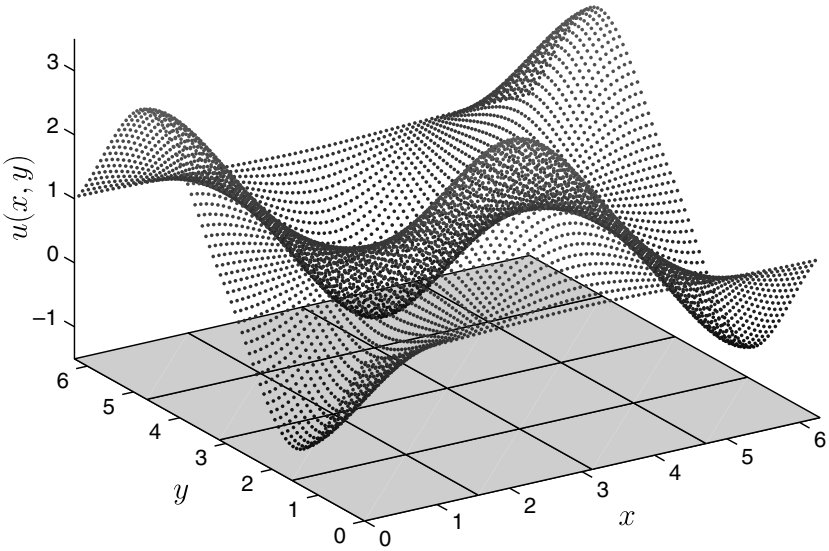


Fig. 5. The initial condition for the $u$ component of velocity interpolated from the cell vertices to the particle positions using bicubic splines for a $4 \times 4$ grid with 400 particles per cell. The particles are uniformly spaced here in order to more clearly visualize the reconstruction; other results presented in this paper do not use this initial particle position distribution. The particle shading is proportional to the velocity component value, with black representing the minimum and gray representing the maximum.

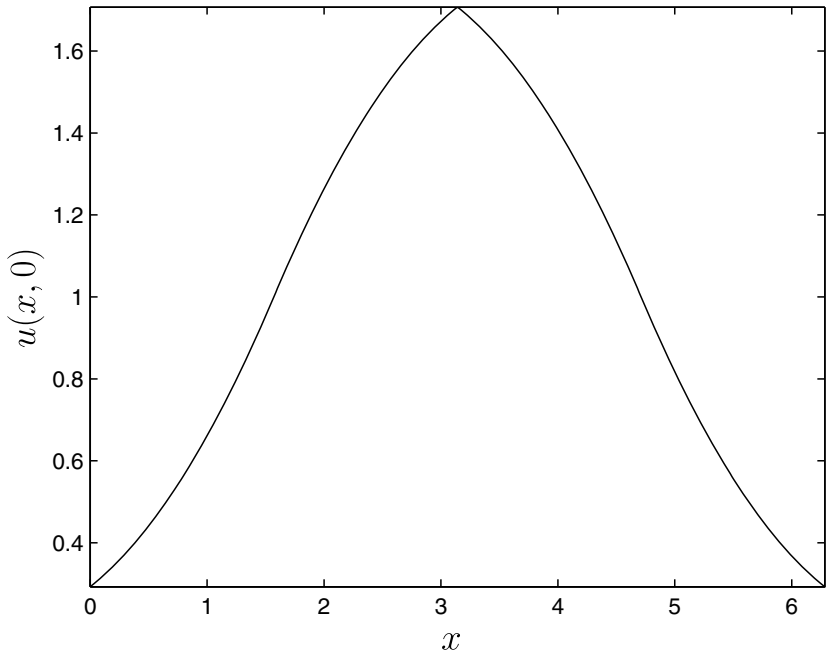


Fig. 6. A 1D slice taken from the reconstruction shown in Fig. 4 for the $u$ component in the $x$-direction at $y = 0$. This plot shows that the $u$ component reconstruction is continuous and piecewise parabolic in the $x$-direction. Note that the domain of length $L = 2\pi$ is periodic in $x$ and that the apparent $C^1$ continuity (i.e., smoothness) observed at $x = L/4$ and $x = 3L/4$ is an artifact of the particular test function (see Eq. (86)) used in this example. In general, all spatial first derivatives may be discontinuous from cell to cell.

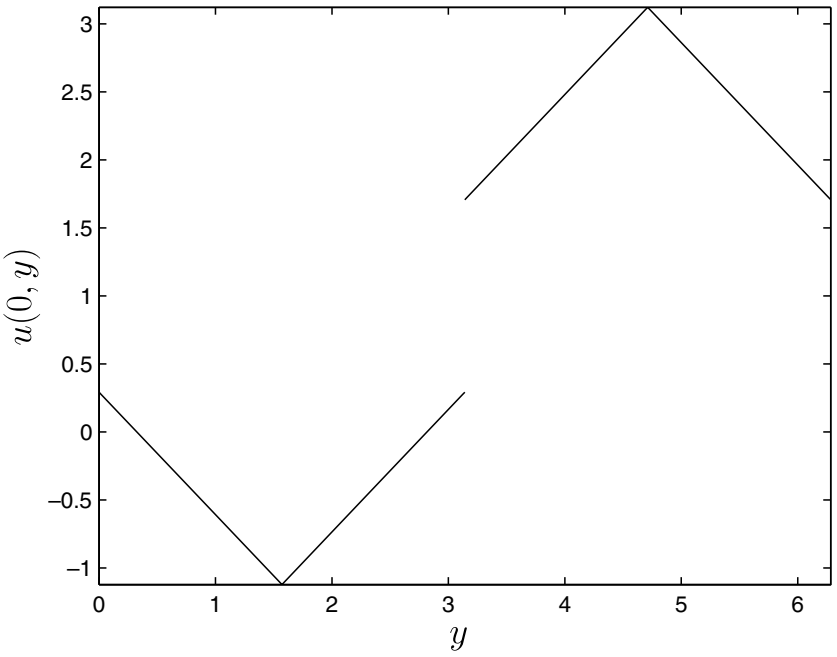


Fig. 7. A 1D slice taken from the reconstruction shown in Fig. 4 for the $u$ component in the $y$-direction at $x = 0$. This plot clearly shows that the $u$ component reconstruction is piecewise linear in the $y$-direction. Note that the domain of length $L = 2\pi$ is periodic in $y$ and that the continuity observed at $y = L/4$ and $y = 3L/4$ is an artifact of the particular test function (see Eq. (86)) used in this example.

velocities, similar to [31]) and bilinear reconstruction (continuous in the component direction but discontinuous in the component-normal direction, based on the staggered-grid velocities, similar to [32]) are compared qualitatively to PERM below.

For a given grid resolution, all simulations start with identical particle positions (i.e., we specify the same seed for the pseudo-random number generator in each case). Eq. (9) is integrated forward in time using the

schemes described in Section 4. To observe the qualitative differences in the various schemes we integrate each run to a time of $t = 2L/U_c$, which is equivalent to two flow-through times ($U_c = 1$ is the mean convective velocity), and visually examine the resulting particle position distributions. Given that the analytical solution is divergence free, ideally the particle position distribution remains uniform both at the grid level and at the subgrid level. The time step is specified by $\Delta t = \mathrm{CFL}\frac{h}{\max(|\mathbf{U}|)}$, where CFL is the Courant–Friedrichs–Lewy number. Figs. 8–10 show the results for the PERM reconstruction, the bilinear reconstruction, and the bicubic spline interpolation, respectively, for the RK2 scheme using CFL = 1/4 on a $4 \times 4$ grid with $N_c = 400$ particles per cell. The bilinear reconstruction is obtained by simply setting the second-order slopes to zero in (26) and (27). Notice that the PERM results show uniform position distributions at both the grid and subgrid levels, whereas both the bilinear and bicubic results contain noticeable voids and clustering of particles at the subgrid level due to the inaccuracy in the divergence field. From this qualitative comparison we may conclude that higher-order continuous interpolation methods do not necessarily improve the accuracy of the subgrid particle distribution if (1) the divergence field is inaccurate and (2) the face integral constraint is not satisfied. The bilinear scheme, which is discontinuous in the component-normal direction but *does* satisfy the face integral constraint, performs just as well for this case as the more computationally expensive continuous bicubic scheme. Hence, in the quantitative comparisons presented below we consider only the bilinear reconstruction as a practical alternative to PERM.

Here the choice of a $4 \times 4$ grid has significance. In practical LES, as opposed to RANS, we expect the solution to contain significant energy in modes up to the grid Nyquist limit. Therefore, the most relevant test case uses as few cells in 1D as possible while still retaining the character of the underlying solution. In the present case the choice of $N_x = 2$ does not allow distinction between the two reconstruction methods, since, due to periodicity, all $\Delta^{(2)}$ for the PERM method are zero. Using $N_x = 4$ allows the solution to be reasonably well approximated by a quadratic function.

We now present measures for quantifying the grid-scale and subgrid-scale distributions. Let $N_e = N_x^2$ denote the number of cells in the domain and let $\bar{\rho}_k$ denote the FV mass density for the $k$th cell, which in the constant-density case we take to be unity for each cell. Let $m^{(i)}$ denote the mass of the $i$th particle. In the present case all particles are of equal mass, $m^{(i)} = M/N$, where $M = L^2\bar{\rho}$ is the total mass in the domain

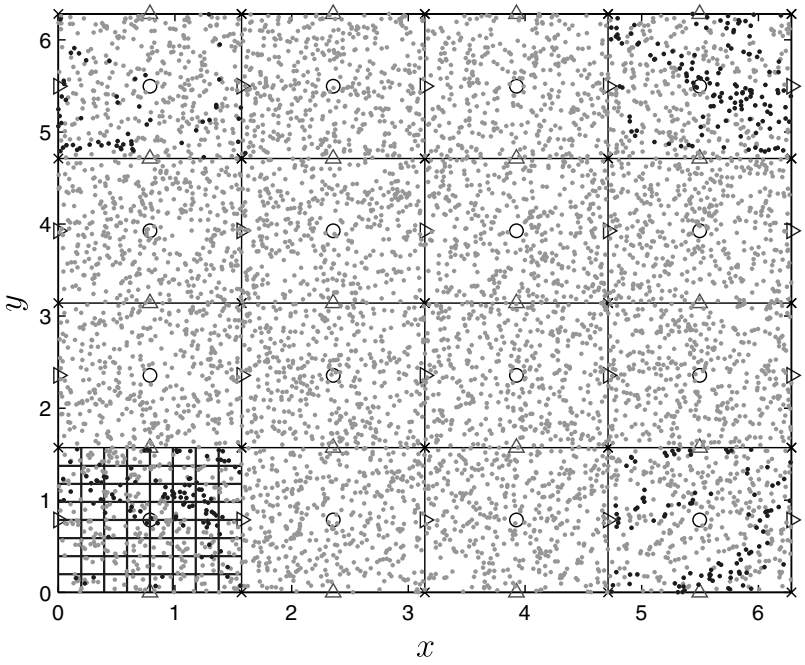


Fig. 8. Final particle positions (represented by the small dots) after two flow-through times (i.e., $t = 2L/U_c$, where $U_c = 1$ is the mean convective velocity and $L = 2\pi$ is the domain length scale) using the PERM reconstruction and the RK2 time integration scheme (81) and (82) with CFL = 0.25. This initial condition for this case is shown in Fig. 3. As can be seen, the uniformity of the subcell particle number density is reasonably well maintained.

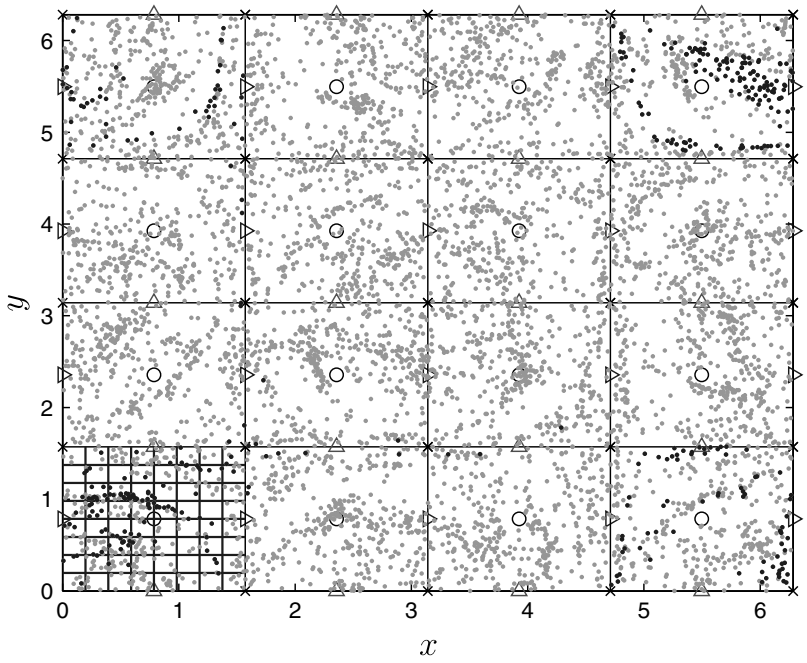


Fig. 9. Final particle positions (represented by the small dots) after two flow-through times (i.e., $t = 2L/U_c$, where $U_c = 1$ is the mean convective velocity and $L = 2\pi$ is the domain length scale) using the bilinear reconstruction (second-order slopes set to zero) and the RK2 time integration scheme (81) and (82) with CFL = 0.25. This initial condition for this case is shown in Fig. 3. Notice the voids and clusters which form at the subcell level due to the particles experiencing a non-zero divergence.

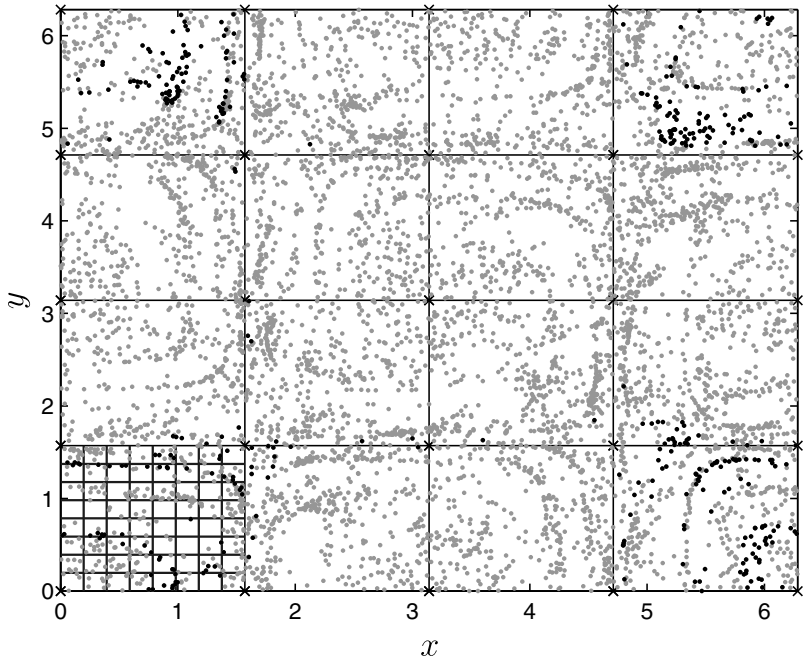


Fig. 10. Final particle positions (represented by the small dots) after two flow-through times (i.e., $t = 2L/U_c$, where $U_c = 1$ is the mean convective velocity and $L = 2\pi$ is the domain length scale) using bicubic spline interpolation of the vertex velocities and the RK2 time integration scheme (81) and (82) with CFL = 0.25. This initial condition for this case is shown in Fig. 3. Voids and clusters form at the subcell level due to the non-zero divergence experienced by the particles even for this higher-order interpolation scheme.

and $N = N_c N_e$ is the total number of particles. The mean particle mass density for the $k$th cell, denoted $\hat{\rho}_k$, is given by

$$\hat{\rho}_k = \frac{\sum_{i \in k} m^{(i)}}{h^2}, \tag{88}$$

where the summation is over all particles in the $k$th cell. Following Zhang and Haworth [32], we define the global measure of the deviation between the mean particle mass density and the FV mass density by

$$\Delta\rho_{\mathrm{rms}} \equiv \left[\frac{\sum_{k=1}^{N_{\mathrm{e}}}(\Delta\rho_k)^2}{N_{\mathrm{e}}}\right]^{1/2}, \tag{89}$$

where

$$\Delta\rho_k \equiv \frac{\hat{\rho}_k - \bar{\rho}_k}{\bar{\rho}_k}. \tag{90}$$

The value $\Delta\rho_{\mathrm{rms}}$ measures the mean particle mass density deviation from the FV mass density (which is exactly known in this case) at the grid level.

We monitor the subgrid particle position distribution for the lower-left corner cell throughout a given simulation. For this problem a single cell is representative of all the subgrid distributions. Within the cell we construct an $N_{\mathrm{f}} \times N_{\mathrm{f}}$ uniform square grid so that the FV cell is partitioned into $N_{\mathrm{f}}^2$ "fine cells" (for all simulations presented here we specify $N_{\mathrm{f}} = 8$). We then construct the histogram of $n$, the number of particles in a fine cell. The narrower the histogram, the more uniform the number density. We define the "subgrid particle distribution parameter" $\zeta$ as the standard deviation of $n$ scaled by the initial average number of particles per fine cell,

$$\zeta \equiv \frac{\mathrm{std}(n)}{n_{\mathrm{ave}}}, \tag{91}$$

where in this case we have $n_{\mathrm{ave}} = N_{\mathrm{c}}/N_{\mathrm{f}}^2$. Note that $\zeta$ is made up of two components. First, there is the deterministic component due to non-uniformities in the *expected* subgrid particle number density. Second, there is a statistical component due to sampling from this distribution. If we suppose that the first contribution is zero, because the expected number density is uniform (as in the constant-density case), we have $\langle n\rangle = n_{\mathrm{ave}} \sim N_{\mathrm{c}}$. With the particle positions being random, the variance of $n$ scales with the initial average number of particles per cell, $\mathrm{var}(n) \sim N_{\mathrm{c}}$, and thus $\mathrm{std}(n) \sim N_{\mathrm{c}}^{1/2}$. We then have $\zeta \sim N_{\mathrm{c}}^{-1/2}$. The measure (91) is therefore chosen such that $\zeta \to 0$ in the limit $N_{\mathrm{c}} \to \infty$. Ideally $n_{\mathrm{ave}}$ is large enough that the random contribution is small. Then $\zeta$ measures the non-uniformity in the expected subgrid particle number distribution, which is what we are trying to control with our reconstructed velocity field.

The quantitative results for Problem 1 are presented in Table 1. For a given case we list the reconstruction method (Recon), the time integration method (Time int), the CFL, the maximum value of $\Delta\rho_{\mathrm{rms}}$ observed over the course of the run, and the maximum value of $\zeta$ observed for the lower left cell over the course of the run. All simulations are run to a total time of $t = 2L/U_{\mathrm{c}}$, where $U_{\mathrm{c}} = 1$ is the mean convective velocity, allowing two flow-through times.

Table 1
Quantitative results for Problem 1

| Case | Figure(s) | Recon | Time int | CFL | max($\Delta\rho_{\mathrm{rms}}$) | max($\zeta$) |
|---|---|---|---|---|---|---|
| Baseline | 3 and 11 | Exact | RK3 | 1/16 | 0.062 | 0.485 |
| 1a | 3 and 8 | PERM | RK2 | 1/4 | 0.071 | 0.461 |
| 1b | 3 | PERM | RK2 | 1/8 | 0.052 | 0.464 |
| 1c | 3 and 9 | Bilinear | RK2 | 1/4 | 0.17 | 0.816 |
| 1d | 3 | Bilinear | RK2 | 1/8 | 0.21 | 0.830 |

The 2D periodic domain is a square of side $L = 2\pi$. We use a $4 \times 4$ uniform grid. There are initially $N_{\mathrm{c}} = 400$ particles per cell with positions chosen at random from a uniform distribution using Latin hypercube sampling within each cell (see Fig. 3). Eq. (9) is integrated to a total time of $t = 2L/U_{\mathrm{c}}$, where $U_{\mathrm{c}} = 1$, two flow-through times. In the table below, the first column gives the case number, the second column lists the relevant figures for each case, the third column (Recon) gives the reconstruction method, the fourth column (Time int) gives the time integration method, the fifth column gives the CFL, the sixth column shows our measure of the grid-scale mass density deviation (ideally zero), and the last column gives a measure of the subgrid particle position distribution (ideally zero). The lower left cell is subdivided into $N_{\mathrm{f}} \times N_{\mathrm{f}}$ fine cells (with $N_{\mathrm{f}} = 8$) and $n$ represents the histogram of the number of particles in a fine cell. The subgrid particle distribution parameter is defined by $\zeta = \mathrm{std}(n)/(N_{\mathrm{c}}/N_{\mathrm{f}}^2)$. In the last two columns we report the maximum values of $\Delta\rho_{\mathrm{rms}}$ and $\zeta$ observed over the run time.

The "baseline" case is taken as the benchmark for comparison. It essentially represents the best that can be expected from *any* advection scheme. The reconstruction is "exact," meaning that the particle velocity is taken from (86) and (87) evaluated at the current particle position. The time integration scheme is given by (83)–(85), which is highly accurate, and is used in conjunction with a small time step (CFL = 1/16). Hence, the grid-scale and subgrid-scale deviations shown in the last two columns of Table 1 (ideally these values would be zero) for the baseline case are mainly due to statistical errors (i.e., the use of a finite number of particles and imperfections in the initial particle position distribution).

Proceeding down Table 1, Cases 1a and 1b compare the PERM scheme at two different CFL numbers. We note that the results for both cases are in reasonable agreement with the baseline, but we observe only minor quantitative improvement in the grid-scale distribution and no significant change in the subgrid-scale distribution as the time step is decreased. This indicates that the temporal error is no longer significant for this scheme on this problem. The next set of cases, 1c and 1d, use the bilinear scheme at the same CFL numbers. Again, little quantitative difference is seen between the two bilinear cases. However, we can observe that the PERM cases are in better quantitative agreement with the baseline than the bilinear cases. For $\max(\Delta\rho_{\mathrm{rms}})$, the PERM results are approximately equal to the baseline result, where as the bilinear results are a factor of two to three times larger. Similarly, for $\max(\zeta)$ the PERM results differ from the baseline results by only 5% and the bilinear results differ by almost 70%. These differences can also be observed qualitatively (compare Figs. 8 and 9 with Fig. 11).

For further quantitative comparison, the time histories for the grid-scale number density variation, $\Delta\rho_{\mathrm{rms}}$, and the subgrid particle distribution parameter, $\zeta$, taken from Cases 1a and 1c, are plotted in Figs. 12 and 13. Within the statistical variation, the PERM results are in excellent agreement with the time histories taken from the exact reconstruction. The bilinear results, however, exhibit large, oscillating variations away from the baseline.

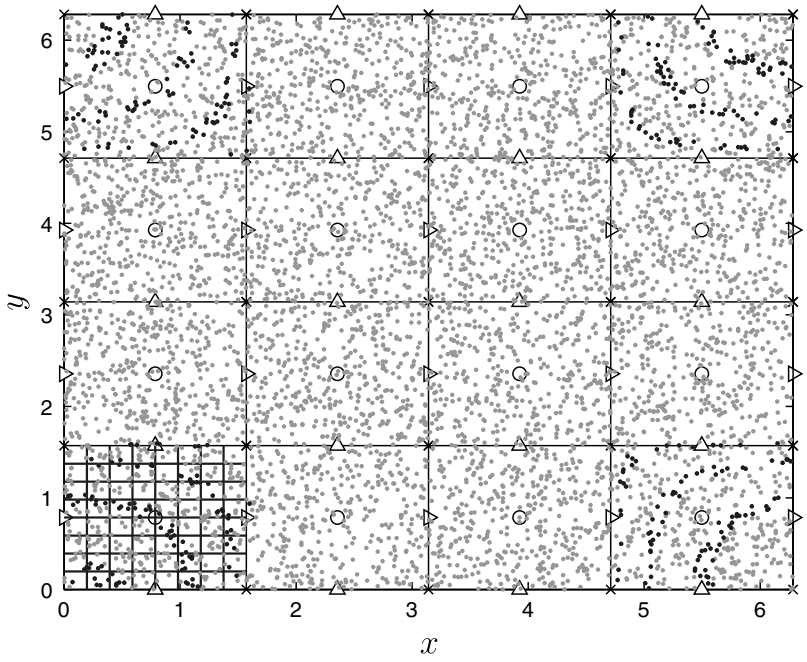


Fig. 11. Final particle positions (represented by the small dots) after two flow-through times (i.e., $t = 2L/U_c$, where $U_c = 1$ is the mean convective velocity and $L = 2\pi$ is the domain length scale) using an exact reconstruction (i.e., the particle velocities are taken directly from (86) and (87) evaluated at the particle position) and the RK3 time integration scheme (83)–(85) with CFL = 0.0625. This initial condition for this case is shown in Fig. 3. This result represents essentially the best that can be expected from any reconstruction method. Variability in the uniformity of the particle number density at the grid or subcell levels is due mainly to statistical variability in the initial distribution and bias in evaluating the cell means. By visually comparing this result with the PERM result in Fig. 8, we find that, while the exact particle positions may be in error (compare the positions of the marker particles), the subcell particle number densities are in good agreement. This observation is confirmed quantitatively in Figs. 12 and 13 below.

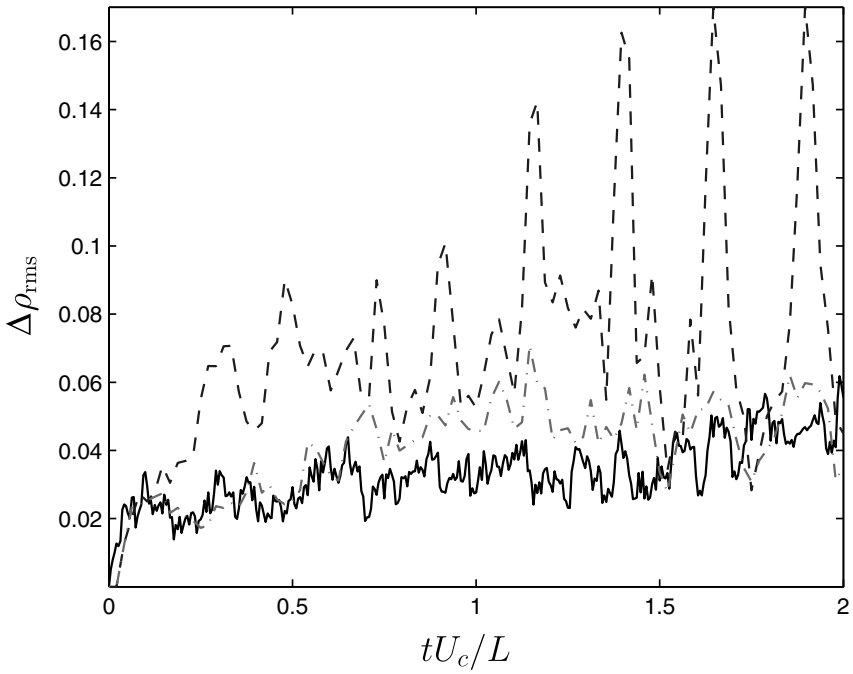


Fig. 12. Time history of the grid-scale rms density variation $\Delta\rho_{\mathrm{rms}}$ given by (89) for the cases shown in Figs. 8–11. The abscissa represents the non-dimensional flow-through time $tU_{\mathrm{c}}/L$, where $U_{\mathrm{c}} = 1$ is the mean convective velocity and $L = 2\pi$ is the length scale of the domain. The solid line represents the baseline accurate solution (exact reconstruction with RK3 time integration and CFL = 0.0625), the dash-dotted (gray) line represents the PERM solution (RK2 CFL = 0.25), and the dashed line represents the bilinear solution (RK2 CFL = 0.25). The PERM result is in good quantitative agreement with the baseline solution, while the bilinear result shows significant oscillations (up to a factor three greater) away from the baseline.

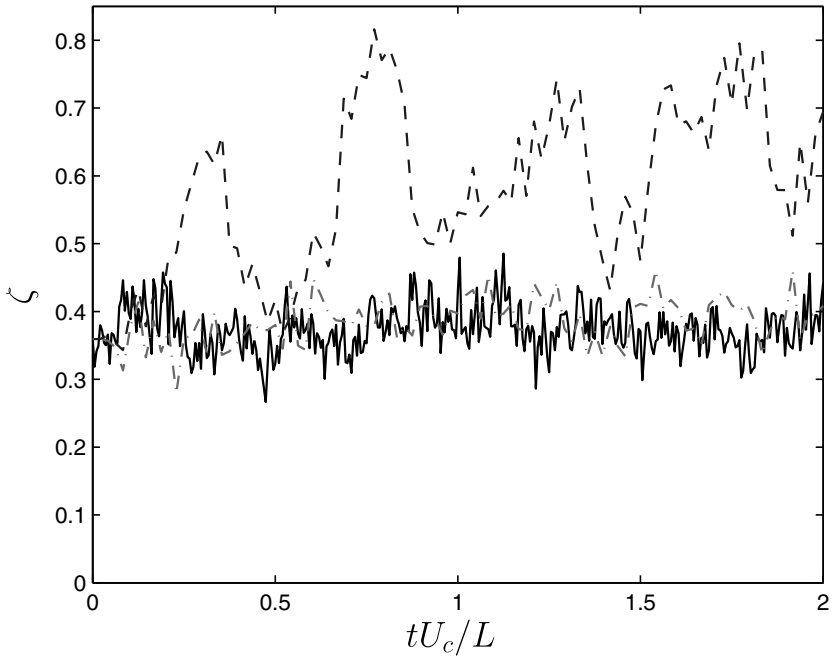


Fig. 13. Time history of the subgrid particle distribution parameter, $\zeta$. There are initially $N_{\mathrm{c}} = 400$ particles per cell for each run. The initial condition is shown in Fig. 3. The lower-left cell of the $4 \times 4$ grid is subdivided into $N_{\mathrm{f}}^2$ fine cells (with $N_{\mathrm{f}} = 8$) as shown in Figs. 8–11 and $n$ represents the number of particles per fine cell. The subgrid distribution parameter $\zeta$ is then given by $\zeta = \mathrm{std}(n)/(N_{\mathrm{c}}/N_{\mathrm{f}}^2)$. The abscissa shows the non-dimensional flow-through time $tU_{\mathrm{c}}/L$, where $U_{\mathrm{c}} = 1$ is the mean convective velocity and $L = 2\pi$ is the length scale of the domain. The solid line represents the baseline accurate solution (exact reconstruction with RK3 time integration and CFL = 0.0625), the dash-dotted (gray) line represents the PERM solution (RK2 CFL = 0.25), and the dashed line represents the bilinear solution (RK2 CFL = 0.25). The PERM result is in good quantitative agreement with the baseline solution, while the bilinear result shows significant oscillations away from the baseline (up to 70% variation).

### 5.2. Problem 2: rotational flow with finite divergence

Since we are interested in combustion modeling, a more relevant test problem includes a non-trivial finite divergence field. Using the Helmholtz decomposition [19], any velocity field can be decomposed into a rotational, solenoidal component and an irrotational (potential) component that accounts for the divergence. Here we add a potential flow component to the rotational flow used in the previous section. The exact velocity field is specified to be

$$U(x,y,t) = 1 - 2\cos(x-t)\sin(y-t) + \sin(x)\cos(2t), \tag{92}$$

$$V(x,y,t) = 1 + 2\sin(x-t)\cos(y-t) + \sin(y)\cos(2t). \tag{93}$$

The divergence of this flow field is

$$\nabla \cdot \mathbf{U} = [\cos(x) + \cos(y)]\cos(2t). \tag{94}$$

The potential part is periodic in space and time and causes a "pulsing" of the divergence field: fluid particles are initially compressed toward the center of the periodic domain; the flow then cycles to compress the fluid particles toward the corners of the domain. The frequency is set to give two compression cycles per flow-through time.

Except for the velocity field, the problem set-up is essentially identical to Problem 1. The initial particle positions are chosen at random from a uniform distribution in each cell using Latin hypercube sampling. The initial grid-level particle number density is exactly uniform with $N_c = 400$ particles per cell. Eq. (9) is integrated forward in time to $t = 2L/U_c$, two flow-through times (in this case, four compression cycles), where $U_c = 1$ is the mean convective velocity. The main difference between this problem and the previous one is that here we do not have an analytic expression for the mass density. Instead we rely on the "exact RK3" scheme to provide an accurate numerical solution for comparison. The velocity reconstruction is taken exactly from (92) and (93) and (9) is integrated accurately in time using the RK3 scheme given by (83)–(85). Note that we can add the following evolution equation to the particle system to obtain a highly accurate numerical solution for the mass density at the particle position,

$$\frac{\mathrm{d}\ln\rho^*}{\mathrm{d}t} = -\nabla \cdot \langle \mathbf{U}^* | \mathbf{x} \rangle, \tag{95}$$

where $\rho^*(t_0) \equiv \rho(\mathbf{X}^*[t_0], t_0)$ and we obtain $\nabla \cdot \langle \mathbf{U}^*(t) | \mathbf{x} \rangle = \nabla \cdot (\mathbf{U}[\mathbf{x}, t])$ from (94).

The PERM and bilinear reconstructions require the staggered-grid velocity components, which are sampled from the analytic solution, (92) and (93), at the staggered positions; the discrete cell divergence is then obtained from (23).

The complexity of the flow pattern can be seen in Fig. 14 where we show the baseline accurate result for a $4 \times 4$ grid with initially $N_c = 400$ particles per cell (uniformly distributed, see Fig. 3). Note that the grid for the baseline case is shown simply for reference; the grid spacing $h$ only affects the time step in the baseline calculation (through the CFL number). The results for the PERM and bilinear cases, which are integrated using the RK2 scheme, are shown in Figs. 15 and 16, respectively. The baseline case uses a CFL of 1/16 and the RK2 cases use a CFL of 1/4. The PERM and bilinear results are in reasonable agreement with the baseline case. However, the PERM case does a better job of capturing the fine structure of the flow pattern.

Quantifying the accuracy of the subgrid particle number density is difficult in this case because the large void volumes with few particles introduce large statistical errors. Thus, for this problem the simple measure used in the previous section, the subgrid particle distribution parameter $\zeta$, is not a good indicator of the accuracy of the advection scheme. In practical simulations particle-number-density-control algorithms attempt to keep the particle number density uniform by "splitting" and "clustering" of the particles [32]. The mean particle mass density varies through adjustments in the particle mass weightings. It is important to note that these algorithms have no effect in the particle-tracking limit since they rely on the random velocity fluctuation for an individual particle to keep the number density uniform. If this velocity is zero then even if particles are split their trajectories remain identical. In LES it is important to consider the properties of the formulation in the "DNS limit." That is, the limit in which statistical fluctuations approach zero. As this problem illustrates,

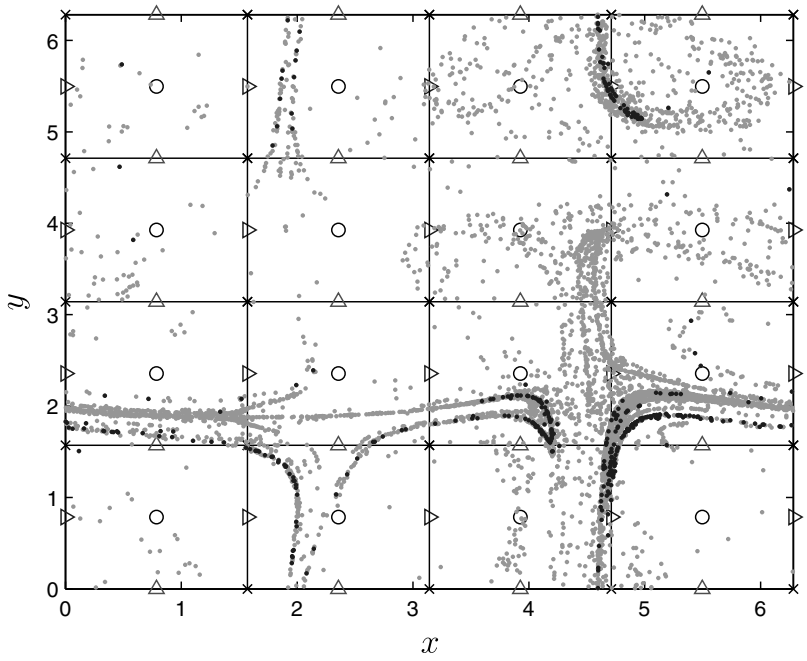


Fig. 14. Baseline accurate solution for Problem 2 after two flow-through times, $t = 2L/U_c$, where $U_c = 1$ is the mean convective velocity and $L = 2\pi$ is the length scale of the domain. The initial particle positions are as shown in Fig. 3. The flow field is defined by (92) and (93). The reconstruction is exact; i.e., the particle velocity is taken from the exact flow field evaluated at the particle position. The positions are advanced using the RK3 time integration scheme with CFL = 0.0625. Particle splitting and clustering algorithms are ineffective in the particle-tracking limit and the resulting void volumes suggest the need the for alternate number-density-control algorithms which do not rely on random walks of the particle positions. The fine structures that emerge are a result of chaotic stretching and folding interacting with a pulsating divergence field. These structures are used as a baseline for qualitative comparison of the PERM and bilinear reconstruction methods shown in Figs. 15 and 16 below.

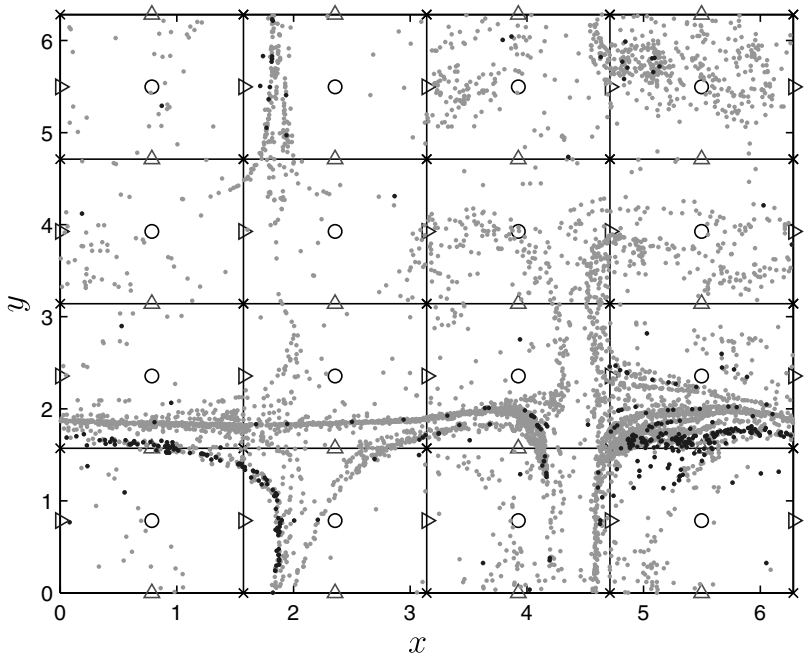


Fig. 15. PERM solution for Problem 2 after two flow-through times, $t = 2L/U_c$, where $U_c = 1$ is the mean convective velocity and $L = 2\pi$ is the length scale of the domain. The initial particle positions are as shown in Fig. 3. The flow field is defined by (92) and (93). The positions are advanced using the RK2 time integration scheme with CFL = 0.25. Though the marker particles clearly show that the position errors are significant over this time scale (compared with Fig. 14), the qualitative structure of the solution is maintained by the PERM advection scheme.

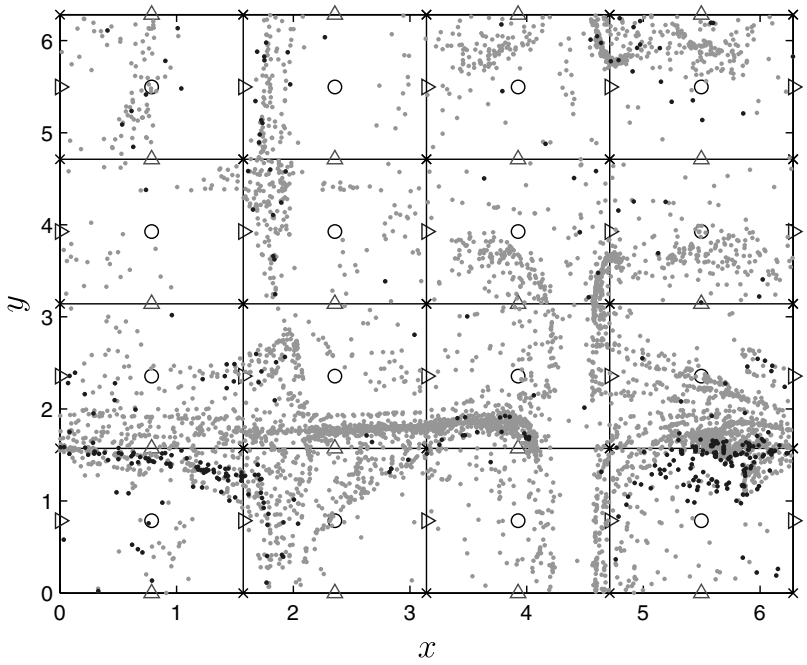


Fig. 16. Bilinear solution for Problem 2 after two flow-through times, $t = 2L/U_c$, where $U_c = 1$ is the mean convective velocity and $L = 2\pi$ is the length scale of the domain. The initial particle positions are as shown in Fig. 3. The flow field is defined by (92) and (93). The positions are advanced using the RK2 time integration scheme with CFL = 0.25. By comparing this result with Fig. 14, we can see that much of the detailed structure of the accurate solution is lost using the bilinear reconstruction.

improvements in number-density-control algorithms are required that do not rely on statistical fluctuations of the particle position.

While the statistics of the subgrid particle number density are hard to quantify, we can quantify the accuracy of the reconstruction for velocity fields with a non-trivial divergence. To this end, a convergence study based on the test problem from this section is presented in Appendix D. The study confirms that the bilinear scheme is first-order accurate for the divergence and that PERM is second-order accurate for the divergence.

### 5.3. Problem 3: solenoidal, rotational flow in 3D

In this section we present a test case and results for the 3D implementation. A simple 3D extension of the solenoidal toy problem from Section 5.1 is the following:

$$U(x,y,z,t) = 1 - 2\cos(x-t)\sin(y-t)\sin(z-t), \tag{96}$$

$$V(x,y,z,t) = 1 + 4\sin(x-t)\cos(y-t)\sin(z-t), \tag{97}$$

$$W(x,y,z,t) = 1 - 2\sin(x-t)\sin(y-t)\cos(z-t). \tag{98}$$

Note that the divergence of this field is zero for all time. We utilize the exact solution (96)–(98) in the same manner as before: the staggered-grid velocity data are sampled at the staggered locations and these data are used as the basis for the subgrid velocity reconstruction.

We consider a periodic, cubic domain of side $L = 2\pi$ with $N_x = 4$ cells in each direction. To maintain a volumetric particle number density similar to the 2D case, and hence a similar level of statistical error, we specify initially $N_c = 8000$ particles per cell ($N_c = 400 = 20 \times 20$ for the 2D case and $N_c = 8000 = 20 \times 20 \times 20$ for the 3D case). The particle positions are advanced in time using the RK2 scheme (81) and (82) for a period equal to four flow through times. That is, the simulations are run to time $t = 4L/U_c$, where $U_c = 1$ is the mean convective velocity.

We first run a baseline case using an "exact" reconstruction. As before, in the exact reconstruction the particle velocities are taken from the specified velocity field (96)–(98) evaluated at the current particle positions. In contrast to the 2D problem, here the RK2 scheme is used for the time integration, again with a CFL of 1/16.

Table 2
Quantitative results for Problem 3

| Case | Figures | Recon | Time int | CFL | max($\Delta\rho_{\rm rms}$) | max($\zeta$) |
|---|---|---|---|---|---|---|
| Baseline | 18 and 20 | Exact | RK2 | 1/16 | 0.014 | 0.280 |
| 3a | 18 and 20 | PERM | RK2 | 1/4 | 0.12 | 0.476 |
| 3b | 18 and 20 | PERM | RK2 | 1/16 | 0.023 | 0.300 |
| 3c | 18 and 20 | Trilinear | RK2 | 1/4 | 0.15 | 0.580 |
| 3d | 18 and 20 | Trilinear | RK2 | 1/16 | 0.24 | 0.552 |

The 3D periodic domain is a cube of side $L = 2\pi$. We use a $4 \times 4 \times 4$ uniform grid. There are initially $N_c = 8000$ particles per cell with positions chosen independently at random from a uniform distribution (no Latin hypercube sampling in this case). Eq. (9) is integrated to a total time of $t = 4L/U_c$, four flow-through times, where $U_c = 1$ is the mean convective velocity. In the table below, the first column gives the case number, the second column lists the relevant figures for each case, the third column (Recon) gives the reconstruction method, the fourth column (Time int) gives the time integration method, the fifth column gives the CFL, the sixth column shows our measure of the grid-scale mass density deviation, and the last column gives a measure of the subgrid particle position distribution. Cell $\{i = j = k = 1\}$ is subdivided into $N_f^3$ fine cells (with $N_f = 8$) and $n$ represents the histogram of the number of particles in a fine cell. The subgrid particle distribution parameter is defined as $\zeta = \mathrm{std}(n)/(N_c/N_f^3)$. In the last two columns we report the maximum values of $\Delta\rho_{\rm rms}$ and $\zeta$ observed over the run time (see the time histories in Figs. 18 and 20).

The change to the RK2 scheme for the baseline accurate solution is made simply to ease the implementation in the 3D code and, as can be seen below, does not adversely affect the results for the constant-density case considered here. In Table 2 we report the baseline measures of the grid-level and subgrid-level particle density variations, max($\Delta\rho_{\rm rms}$) and max($\zeta$), respectively. The grid-level deviation, $\Delta\rho_{\rm rms}$, is taken from (89). To obtain the subcell particle number density we subdivide cell $\{i = j = k = 1\}$ into $N_f^3$ uniform fine cells (with $N_f = 8$), $n$ represents the number of particles in a fine cell, $n_{\rm ave} = N_c/N_f^3$, and the subgrid particle distribution parameter is obtained from (91).

Table 2 also reports the grid-level and subgrid-level particle densities for the PERM and trilinear cases at two different CFL numbers, CFL = 1/4 and CFL = 1/16, using the RK2 scheme. As can be seen, the temporal error is significant for both the PERM and trilinear cases for a CFL of 1/4. With a smaller time step, CFL = 1/16, the PERM results converge to the baseline accurate case, while the trilinear results do

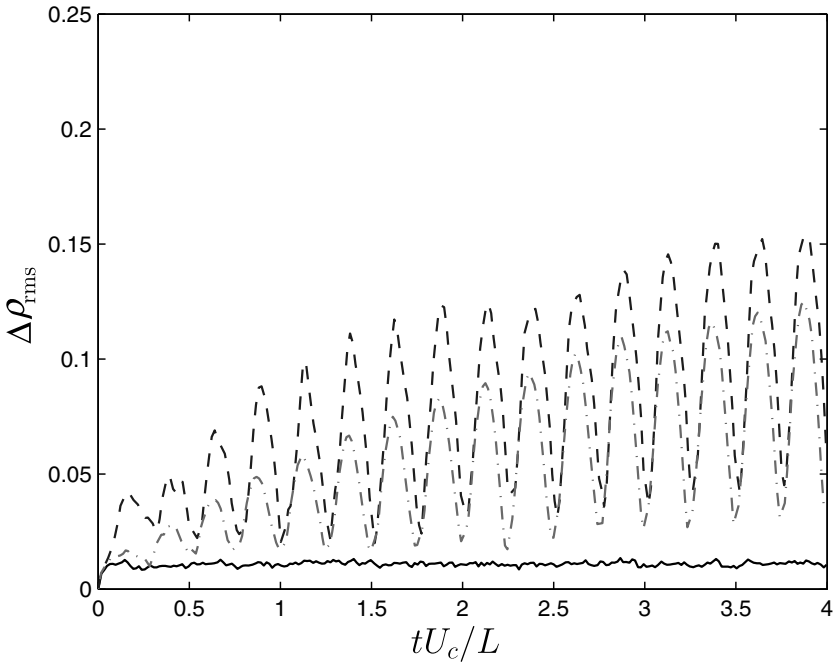


Fig. 17. Time history of the grid-level mass density variation, $\Delta\rho_{\rm rms}$, given by (89), for the 3D baseline case and cases 3a and 3c from Table 2. The abscissa shows the non-dimensional flow-through time $tU_c/L$, where $U_c = 1$ is the mean convective velocity scale and $L = 2\pi$ is the domain length scale. The solid line represents the baseline accurate results using an "exact" reconstruction (particle velocities taken from the analytical solution) and the RK2 time integration scheme with CFL = 0.0625. The dash-dotted (gray) line represents the PERM results using the RK2 scheme with CFL = 0.25 and the dashed line represents the trilinear results using the RK2 scheme with CFL = 0.25.

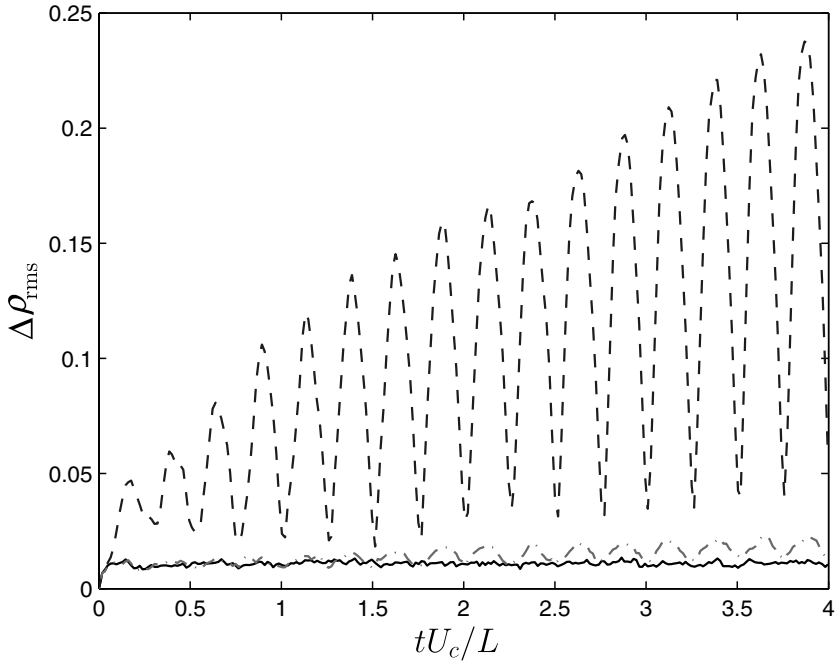


Fig. 18. Time history of the grid-level mass density variation, $\Delta\rho_{\mathrm{rms}}$, given by (89), for the 3D baseline case and cases 3b and 3d from Table 2. The abscissa shows the non-dimensional flow-through time $tU_{\mathrm{c}}/L$, where $U_{\mathrm{c}} = 1$ is the mean convective velocity scale and $L = 2\pi$ is the domain length scale. The solid line represents the baseline accurate results using an "exact" reconstruction (particle velocities taken from the analytical solution) and the RK2 time integration scheme with CFL = 0.0625. The dash-dotted (gray) line represents the PERM results using the RK2 scheme with CFL = 0.0625 and the dashed line represents the trilinear results using the RK2 scheme with CFL = 0.0625.

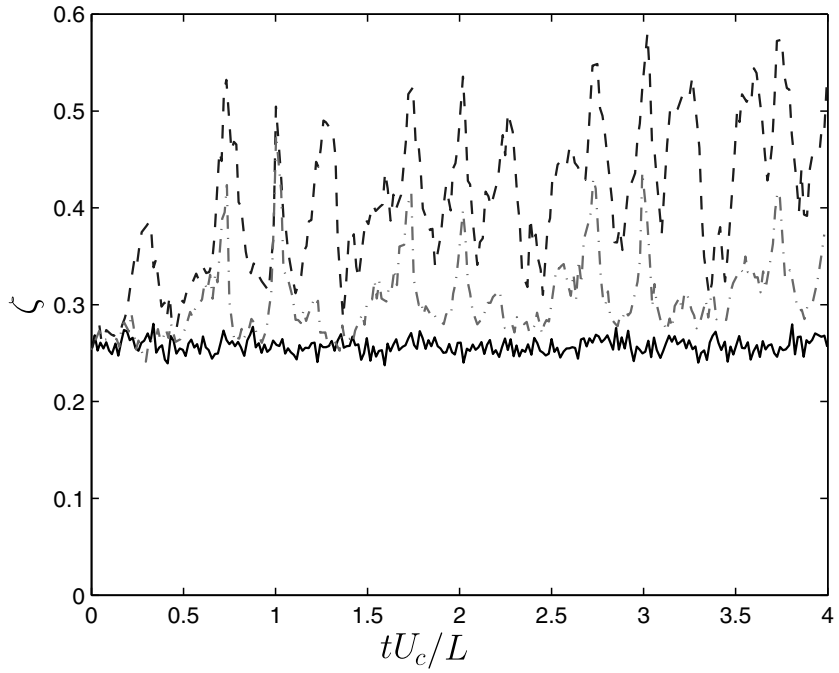


Fig. 19. Time history of the subgrid particle distribution parameter, $\zeta$, for the 3D baseline case and cases 3a and 3c from Table 2. There are initially $N_{\mathrm{c}} = 8000$ particles per cell. We obtain $n$ by subdividing cell $\{i = j = k = 1\}$ into $N_{\mathrm{f}}^3$ fine cells (with $N_{\mathrm{f}} = 8$) and constructing the histogram of the number of particles $n$ in each fine cell. We then obtain the subgrid distribution parameter by $\zeta = \mathrm{std}(n)/(N_{\mathrm{c}}/N_{\mathrm{f}}^3)$. The abscissa shows the non-dimensional flow-through time $tU_{\mathrm{c}}/L$, where $U_{\mathrm{c}} = 1$ is the mean convective velocity scale and $L = 2\pi$ is the domain length scale. The solid line represents the baseline accurate results using an "exact" reconstruction (particle velocities taken from the analytical solution) and the RK2 time integration scheme with CFL = 0.0625. The dash-dotted (gray) line represents the PERM results using the RK2 scheme with CFL = 0.25 and the dashed line represents the trilinear results using the RK2 scheme with CFL = 0.25.

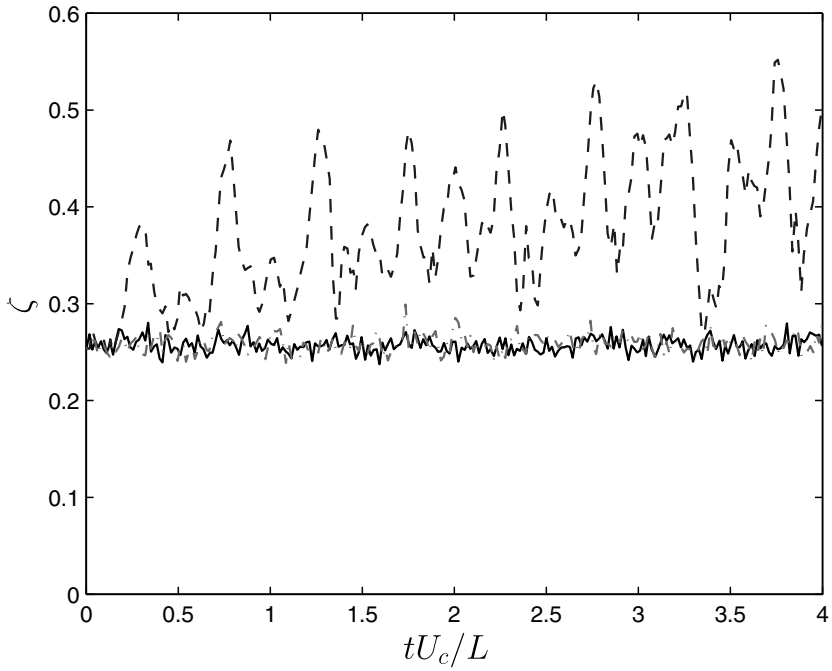


Fig. 20. Time history of the subgrid particle distribution parameter, $\zeta$, for the 3D baseline case and cases 3b and 3d from Table 2. There are initially $N_c = 8000$ particles per cell. We obtain $n$ by subdividing cell $\{i = j = k = 1\}$ into $N_f^3$ fine cells (with $N_f = 8$) and constructing the histogram of the number of particles $n$ in each fine cell. We then obtain the subgrid distribution parameter by $\zeta = \mathrm{std}(n)/(N_c/N_f^3)$. The abscissa shows the non-dimensional flow-through time $tU_c/L$, where $U_c = 1$ is the mean convective velocity scale and $L = 2\pi$ is the domain length scale. The solid line represents the baseline accurate results using an "exact" reconstruction (particle velocities taken from the analytical solution) and the RK2 time integration scheme with CFL = 0.0625. The dash-dotted (gray) line represents the PERM results using the RK2 scheme with CFL = 0.0625 and the dashed line represents the trilinear results using the RK2 scheme with CFL = 0.0625.

not improve. In fact, for this problem over the range of time steps studied, the grid-level variation for the trilinear case worsens as the time step is decreased.

Time histories of the grid- and subgrid-level densities for the cases listed in Table 2 are shown in Figs. 17 and 19 for CFL = 1/4 and in Figs. 18 and 20 for CFL = 1/16. For large time steps, it appears that the trilinear and PERM results are approaching similar levels of error. With a sufficiently small time step the PERM results converge to the baseline accurate results at both the grid and subgrid levels. These results suggest that improvements to the time integration scheme may be beneficial. It should be noted that, while the two-stage RK scheme is a significant improvement over explicit Euler, the overall advection scheme is still formally first-order due to the discontinuities at the cell boundaries. Improvements to this scheme to achieve formal second-order accuracy will necessarily require more sophisticated (potentially more costly) particle-tracking methods (i.e., determination of when and where a particle leaves one cell and enters another). This issue is addressed in a supplemental paper [26]. It is to be noted that the inaccuracies of the time integration need not negate the improved spatial accuracy of the reconstruction introduced in this paper. The results of this section, which – given the coarseness of the underlying grid – are well representative of practical LES cases, show that sufficient accuracy may be achieved with a reasonable time step. Improvements to the overall advection scheme should seek to reduce the overall cost to achieve comparable agreement with the baseline solution. This may occur through reducing the cell-to-cell discontinuities or improving the accuracy of the time integration (see [26]).

## 6. Future work

Higher-order PERM-like reconstructions may be needed in conjunction with high-order finite difference schemes for the velocity field. The present reconstruction strategy targets second-order finite-volume methods. Note that this second-order method may still be useful for DNS especially when coupled with adaptive mesh refinement (see [4]). However, in finite difference methods it is common to employ schemes with spectral-like

accuracy for the velocity field on uniform or stretched grids and thus higher-order interpolation schemes for particle tracking are also needed. PERM is extremely effective at tracking particles for constant-density flows because here the divergence field is identically zero. As we note in Section 5.1, moderately high-order schemes such as cubic splines which do not control the divergence lead to voids and clustering of particles at the subgrid level. As the degree of density variation increases PERM yields the best estimate of the subgrid divergence field that is possible for a second-order scheme. However, if the divergence field varies as a polynomial of degree three or higher, then higher-order interpolation schemes may be required.

The results of the 3D test case clearly point out that improved accuracy in the advection scheme can be achieved through improved time integration. The current RK2 scheme is second-order accurate within a cell, but reverts to zeroth-order accuracy at a cell discontinuity. The remedy to this situation may require more costly particle-tracking algorithms (i.e., determination of when and where a particle crosses a cell boundary) and leads to the question of whether greater efficiency (i.e., lower computational cost for a specified error tolerance) can be better achieved by: (1) the present method on a refined grid, (2) improved time integration, or (3) reduction in (or elimination of) the spatial discontinuities, which can only come through a higher-order reconstruction. An improved time integration scheme which supplements the present work is developed in [26].

The natural extension of the present method to higher order is to use cubic reconstructions for the edges parallel with the component direction and quadratic reconstructions in the cross-component directions. Such a method has the potential to eliminate, or at least significantly reduce, the cell-to-cell discontinuities in the cross-directional velocities and divergence fields. Of course, as usual in numerical methods, accuracy and (one-time) cost are (inversely) coupled. We say "one-time cost" because often for a specified error tolerance the overall simulation is more efficient when using a more accurate scheme. As a case in point, the study by Yeung and Pope [31] shows that in the DNS context cubic-spline interpolation is much more efficient than trilinear interpolation. In the LES/FDF context the situation is not as clear. Many aspects of practical simulations are inaccurate, and with the particle position suffering a random walk there may be limited benefit from utilizing a highly-accurate particle-tracking scheme. Guaranteeing proper particle density at the grid level (addressed by [32]) *and* subgrid level (open research area) is most likely the key issue.

## 7. Conclusions

Within this work we have introduced the parabolic edge reconstruction method (PERM), a variant of the 2D reconstruction method of Jenny et al. [12] based on the piecewise parabolic method (PPM) of Colella and Woodward [5]. In addition to improving the properties of the 2D reconstruction, we have shown how the new scheme is easily extended to 3D (Appendix A) and non-uniform grids (Appendix B). The new formulation obeys the following general principle: For a given velocity component, the second-order slopes for the parabolic edges aligned with that component are determined primarily by their own fields (as opposed to the velocity divergence, which involves other fields) and a minimum correction is made to the second-order slopes along the parabolic edges so that the reconstructed divergences at cell vertices are as close a possible to the average values of the divergences of the incident cells.

As discussed in the introduction, the divergence that the particles experience plays a critical role in maintaining correspondence between the PDF of particle position and the Eulerian filtered mass density in the particle-tracking limit (no velocity fluctuations). It is desirable that hybrid LES/FDF formulations treat particle-tracking accurately because the model then behaves appropriately in the DNS limit (discussed in more detail in [15]). Additionally, accurate particle-tracking schemes may be useful for aerosol transport across a range of problems. The philosophy underlying the current scheme is that the most cost-effective way to control the velocity divergence at the particle position is to use a velocity reconstruction that is parabolic in the velocity-component direction and linear in the component-normal direction. In addition to the added accuracy achieved for the given component in one direction, through a minimum correction we are able to control the divergence of the reconstructed velocity field and this proves to be useful in maintaining a uniform particle number density in the constant-mass-density limit. The key advantages that PERM displays over other commonly used particle-tracking schemes (such as spectral methods and B-splines) are its compact support and its control over the subgrid divergence. We highlight these advantages in Section 5.1 where it is made clear that

accurate control over the subgrid divergence leads to a more accurate subgrid particle number density and that this can be achieved with a relatively low-order method which satisfies the face integral constraint.

The properties of the new method are reiterated below:

1. For a given component, the reconstructed field in the component direction is continuous, piecewise parabolic, and formally second-order accurate (though third-order behavior is observed at coarse resolution for the test problems presented here).
2. In the component-normal direction the reconstructed field is piecewise linear and second-order accurate.
3. In the component direction, the spatial derivative of that component is piecewise linear and second-order accurate.
4. In the component-normal direction, the spatial derivative is piecewise constant and first-order accurate.
5. The surface integral of the reconstructed field is consistent with the discrete cell divergence obtained from the staggered-grid velocity data.
6. The divergence of the reconstructed field is piecewise bilinear in 2D (trilinear in 3D), but may be discontinuous from cell to cell.
7. Due to Items 5 and 6, the reconstructed divergence evaluated at the cell center matches the discrete cell divergence.
8. At cell vertices the reconstructed divergence is as close as possible (given the quadratic form of the reconstruction) to the average divergence of the surrounding cells.
9. Due to Item 8, the divergence of the reconstructed field is everywhere zero in the constant-density case.

Given the apparent complexity in deriving the method, it is worth mentioning that the implementation is straightforward and efficient. Even though the edge parameters are concisely written in matrix notation above and in the appendices, there is no need to perform a matrix inversion or solve a linear system during a simulation. The components of the pseudo-inverse $B^{+}$ are simply stored upon initialization of the program. Further, note that the $A$ and $B$ matrices are sparse and this fact should be exploited for efficient matrix–vector multiplications. Lastly, the edge parameters need to be computed only once per time step for the two-stage Runge–Kutta scheme used here.

In Section 5 we present results for: (1) a 2D solenoidal, rotational flow; (2) a 2D rotational flow with non-zero divergence; and (3) a 3D solenoidal, rotational flow. From the first problem we can draw the conclusion that the PERM method is effective at maintaining a uniform particle number density for constant-density flows in 2D. In this case the time step used is reasonable for practical calculations (CFL $= 0.25$) and the sinusoidal solution on the coarse underlying grid is non-trivial and representative of a (2D) LES field. The second problem adds the effect of a pulsating divergence field and therefore is more relevant to turbulent reacting flows. The resulting flow field, which has a complex structure resulting from the stretching and folding of chaotic advection, leads to large particle void volumes, even in the baseline accurate case. Since particle splitting and clustering algorithms are ineffective in the particle-tracking limit, and since uniform particle number density is necessary to minimize the statistical error encountered when computing cell means, we can conclude that improved algorithms for particle number density control are needed that do not rely on a random walk of the particles. The third and final problem mimics the first problem but in 3D. Perhaps surprisingly, we find that temporal errors are controlling for this problem for a CFL of 1/4. This is understandable, however, because even though the two-stage RK scheme is a dramatic improvement over explicit Euler (though tests are not shown here, this is indeed the case), the scheme is still formally first-order in the presence of discontinuities at cell boundaries. We find that a CFL of 1/16 is sufficient to reduce the temporal error to approximately the same level as the spatial errors for the problem considered. The issue of improved time integration is further discussed in a supplemental paper [26]. Note that in all the test cases presented here a large number of particles is used in order to minimize statistical error.

## Acknowledgments

This research was performed while the first author held a National Research Council Research Associateship Award at the National Institute of Standards and Technology. The second author was supported by the

National Science Foundation through Grant CBET-0426787. Additionally, the authors would like to thank Pavel Popov for helpful analyses and discussions.

## Appendix A. 3D reconstruction

In this appendix we provide details on the 3D formulation for PERM. The 3D reconstruction follows all the basic principles outlined in Section 3.2 for the 2D scheme. The only significant differences result from the shapes of the matrices $A, B$ and $B^+$ and the rank deficiency of the matrix $B$, since each cubic cell has 12 edges and 8 vertices. The 3D reconstruction is presented below in (102)–(104) and the divergence of the reconstructed field is given by (105)–(108). For brevity, not all the final vertex velocities are explicitly shown. Instead, we provide an example for obtaining the final vertex velocity components on the north face. All the remaining components are specified analogously and the parabolic edge velocities and first-order slopes are obtained using (19) and (20). The second-order slopes are obtained from the least-squares/minimum-norm solution (121). Details for specifying the elements of the matrices $A$ and $B$ (for our choice of ordering, which is explained below) are provided via Algorithms 1 and 2. The pseudo-inverse $B^+$ is obtained from the singular value decomposition of $B$.

We denote the directional components as $\mathbf{x} = [x, y, z]^{\mathrm{T}}$. The center of cell $(i, j, k)$ is located at $(x_i, y_j, z_k)$ and the local cell coordinates are defined as

$$q_i \equiv \frac{x - x_i}{h_i} + \frac{1}{2} \quad \text{for } x \in \left[x_i - \frac{h_i}{2}, x_i + \frac{h_i}{2}\right], \quad q_i \in [0, 1], \tag{99}$$

$$r_j \equiv \frac{y - y_j}{g_j} + \frac{1}{2} \quad \text{for } y \in \left[y_j - \frac{g_j}{2}, y_j + \frac{g_j}{2}\right], \quad r_j \in [0, 1], \tag{100}$$

$$s_k \equiv \frac{z - z_k}{f_k} + \frac{1}{2} \quad \text{for } z \in \left[z_k - \frac{f_k}{2}, z_k + \frac{f_k}{2}\right], \quad s_k \in [0, 1], \tag{101}$$

where $h_i, g_j$ and $f_k$ are the local cell width, height, and depth, respectively. In the remainder of this appendix we consider uniform, cubic grids. Hence, we denote the grid spacing as $h = h_i = g_j = f_k$ and we drop the subscripts on $q, r$ and $s$. The extension of the method to non-uniform grids is discussed in Appendix B.

The parabolic edge reconstruction is

$$\begin{aligned}
u(q,r,s) = (1-s)\Bigg\{&(1-r)\left[\bar{u}_{\mathrm{BS}} + \left(q-\frac{1}{2}\right)\Delta^{(1)}_{u,\mathrm{BS}} + \frac{1}{2}\left\{\left(q-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{u,\mathrm{BS}}\right] \\
&+ r\left[\bar{u}_{\mathrm{BN}} + \left(q-\frac{1}{2}\right)\Delta^{(1)}_{u,\mathrm{BN}} + \frac{1}{2}\left\{\left(q-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{u,\mathrm{BN}}\right]\Bigg\} \\
+ s\Bigg\{&(1-r)\left[\bar{u}_{\mathrm{TS}} + \left(q-\frac{1}{2}\right)\Delta^{(1)}_{u,\mathrm{TS}} + \frac{1}{2}\left\{\left(q-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{u,\mathrm{TS}}\right] \\
&+ r\left[\bar{u}_{\mathrm{TN}} + \left(q-\frac{1}{2}\right)\Delta^{(1)}_{u,\mathrm{TN}} + \frac{1}{2}\left\{\left(q-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{u,\mathrm{TN}}\right]\Bigg\},
\end{aligned} \tag{102}$$

$$\begin{aligned}
v(q,r,s) = (1-s)\Bigg\{&(1-q)\left[\bar{v}_{\mathrm{BW}} + \left(r-\frac{1}{2}\right)\Delta^{(1)}_{v,\mathrm{BW}} + \frac{1}{2}\left\{\left(r-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{v,\mathrm{BW}}\right] \\
&+ q\left[\bar{v}_{\mathrm{BE}} + \left(r-\frac{1}{2}\right)\Delta^{(1)}_{v,\mathrm{BE}} + \frac{1}{2}\left\{\left(r-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{v,\mathrm{BE}}\right]\Bigg\} \\
+ s\Bigg\{&(1-q)\left[\bar{v}_{\mathrm{TW}} + \left(r-\frac{1}{2}\right)\Delta^{(1)}_{v,\mathrm{TW}} + \frac{1}{2}\left\{\left(r-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{v,\mathrm{TW}}\right] \\
&+ q\left[\bar{v}_{\mathrm{TE}} + \left(r-\frac{1}{2}\right)\Delta^{(1)}_{v,\mathrm{TE}} + \frac{1}{2}\left\{\left(r-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{v,\mathrm{TE}}\right]\Bigg\},
\end{aligned} \tag{103}$$

$$w(q,r,s) = (1-q)\left\{(1-r)\left[\bar{w}_{\mathrm{SW}} + \left(s-\frac{1}{2}\right)\Delta^{(1)}_{w,\mathrm{SW}} + \frac{1}{2}\left\{\left(s-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{w,\mathrm{SW}}\right]\right.$$
$$\left. + r\left[\bar{w}_{\mathrm{NW}} + \left(s-\frac{1}{2}\right)\Delta^{(1)}_{w,\mathrm{NW}} + \frac{1}{2}\left\{\left(s-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{w,\mathrm{NW}}\right]\right\}$$
$$+ q\left\{(1-r)\left[\bar{w}_{\mathrm{SE}} + \left(s-\frac{1}{2}\right)\Delta^{(1)}_{w,\mathrm{SE}} + \frac{1}{2}\left\{\left(s-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{w,\mathrm{SE}}\right]\right.$$
$$\left. + r\left[\bar{w}_{\mathrm{NE}} + \left(s-\frac{1}{2}\right)\Delta^{(1)}_{w,\mathrm{NE}} + \frac{1}{2}\left\{\left(s-\frac{1}{2}\right)^2 - \frac{1}{4}\right\}\Delta^{(2)}_{w,\mathrm{NE}}\right]\right\}. \tag{104}$$

The divergence of this field is

$$h(\nabla\cdot\mathbf{u}) \equiv \phi(q,r,s) = \frac{\partial u}{\partial q} + \frac{\partial v}{\partial r} + \frac{\partial w}{\partial s}, \tag{105}$$

where

$$\frac{\partial u}{\partial q} = (1-s)\left\{(1-r)\left[\Delta^{(1)}_{u,\mathrm{BS}} + \left(q-\frac{1}{2}\right)\Delta^{(2)}_{u,\mathrm{BS}}\right] + r\left[\Delta^{(1)}_{u,\mathrm{BN}} + \left(q-\frac{1}{2}\right)\Delta^{(2)}_{u,\mathrm{BN}}\right]\right\}$$
$$+ s\left\{(1-r)\left[\Delta^{(1)}_{u,\mathrm{TS}} + \left(q-\frac{1}{2}\right)\Delta^{(2)}_{u,\mathrm{TS}}\right] + r\left[\Delta^{(1)}_{u,\mathrm{TN}} + \left(q-\frac{1}{2}\right)\Delta^{(2)}_{u,\mathrm{TN}}\right]\right\}, \tag{106}$$

$$\frac{\partial v}{\partial r} = (1-s)\left\{(1-q)\left[\Delta^{(1)}_{v,\mathrm{BW}} + \left(r-\frac{1}{2}\right)\Delta^{(2)}_{v,\mathrm{BW}}\right] + q\left[\Delta^{(1)}_{v,\mathrm{BE}} + \left(r-\frac{1}{2}\right)\Delta^{(2)}_{v,\mathrm{BE}}\right]\right\}$$
$$+ s\left\{(1-q)\left[\Delta^{(1)}_{v,\mathrm{TW}} + \left(r-\frac{1}{2}\right)\Delta^{(2)}_{v,\mathrm{TW}}\right] + q\left[\Delta^{(1)}_{v,\mathrm{TE}} + \left(r-\frac{1}{2}\right)\Delta^{(2)}_{v,\mathrm{TE}}\right]\right\}, \tag{107}$$

$$\frac{\partial w}{\partial s} = (1-q)\left\{(1-r)\left[\Delta^{(1)}_{w,\mathrm{SW}} + \left(s-\frac{1}{2}\right)\Delta^{(2)}_{w,\mathrm{SW}}\right] + r\left[\Delta^{(1)}_{w,\mathrm{NW}} + \left(s-\frac{1}{2}\right)\Delta^{(2)}_{w,\mathrm{NW}}\right]\right\}$$
$$+ q\left\{(1-r)\left[\Delta^{(1)}_{w,\mathrm{SE}} + \left(s-\frac{1}{2}\right)\Delta^{(2)}_{w,\mathrm{SE}}\right] + r\left[\Delta^{(1)}_{w,\mathrm{NE}} + \left(s-\frac{1}{2}\right)\Delta^{(2)}_{w,\mathrm{NE}}\right]\right\}. \tag{108}$$

The specification of the parabolic edge parameters follows all the basic principles detailed for the 2D case. We first specify the vertex values. For the 3D case we do not go into all the details, which are tedious. Instead, as an example, we show how to obtain the $v$-component vertex values for the "north" face (see Fig. 21). The $v$-component values for the south face, the $u$-component vertex values for the east and west faces, and the $w$-component vertex values for the top and bottom faces follow analogously. With the vertex values specified, the parabolic edge velocities and first-order slopes follow by analogy with (19) and (20). The second-order slopes are specified analogously to the method discussed in Section 3.2. Algorithms for specifying the elements of the matrices are provided below.

Let us start by defining temporary vertex values for the north face,

$$\hat{v}_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = \frac{1}{4}\left(\overline{V}_{i,j+\frac{1}{2},k} + \overline{V}_{i-1,j+\frac{1}{2},k} + \overline{V}_{i-1,j+\frac{1}{2},k-1} + \overline{V}_{i,j+\frac{1}{2},k-1}\right), \tag{109}$$

$$\hat{v}_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = \frac{1}{4}\left(\overline{V}_{i,j+\frac{1}{2},k} + \overline{V}_{i,j+\frac{1}{2},k-1} + \overline{V}_{i+1,j+\frac{1}{2},k-1} + \overline{V}_{i+1,j+\frac{1}{2},k}\right), \tag{110}$$

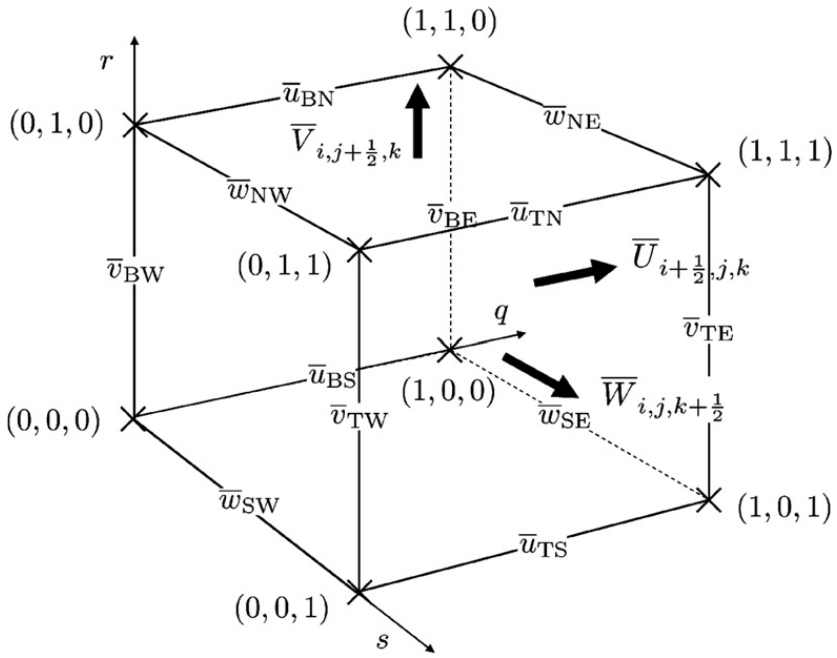


Fig. 21. A 3D cell showing the storage locations for the "top", "north", and "east" staggered-grid velocity data $\overline{W}_{i,j,k+\frac{1}{2}}, \overline{V}_{i,j+\frac{1}{2},k}, \overline{U}_{i+\frac{1}{2},j,k}$, respectively. We also show the labeling convention of the cell vertices in terms of the local cell coordinates $(q, r, s)$, defined by (100) and (101). The parabolic edge velocity components are shown in order to identify each of the 12 edges. Considering the $u$-component parabolic edges, for example, there are four: the bottom-south (BS), the bottom-north (BN), the top-north (TN), and the top-south (TS), with edge velocities $\bar{u}_{\mathrm{BS}}, \bar{u}_{\mathrm{BN}}, \bar{u}_{\mathrm{TN}}$, and $\bar{u}_{\mathrm{TS}}$, respectively. The parabolic edges for the other components follow a similar naming convention. What matters in the formulation is not how the edges are named but rather how they are ordered. The ordering used in the matrix equations, (121) for example, is adopted based on the pattern which is apparent in (102)–(104). The ordering of the vertices is based on a logical nesting of **for** loops as shown in Algorithms 1 and 2.

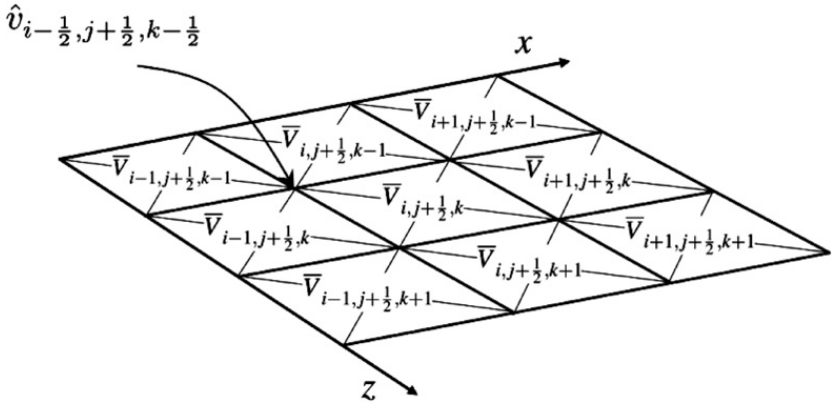


Fig. 22. North face vertex connectivity. The temporary vertex velocities are obtained by averages of the surrounding staggered-grid velocity data. For example, as shown, $\hat{v}_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} = \frac{1}{4}[\overline{V}_{i,j+\frac{1}{2},k} + \overline{V}_{i-1,j+\frac{1}{2},k} + \overline{V}_{i-1,j+\frac{1}{2},k-1} + \overline{V}_{i,j+\frac{1}{2},k-1}]$.

$$\hat{v}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{4}(\overline{V}_{i,j+\frac{1}{2},k} + \overline{V}_{i+1,j+\frac{1}{2},k} + \overline{V}_{i+1,j+\frac{1}{2},k+1} + \overline{V}_{i,j+\frac{1}{2},k+1}), \tag{111}$$

$$\hat{v}_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{4}(\overline{V}_{i,j+\frac{1}{2},k} + \overline{V}_{i,j+\frac{1}{2},k+1} + \overline{V}_{i-1,j+\frac{1}{2},k+1} + \overline{V}_{i-1,j+\frac{1}{2},k}). \tag{112}$$

These are averages of the staggered-grid velocities from the faces surrounding the vertex (see Fig. 22). In practice, the temporary vertex velocities should be stored at the grid level (they are coincident for all cells), as they are also used in the estimate for the second-order slopes. Let $\hat{\bar{v}}_{\mathrm{N}}$ denote the average of the temporary vertex velocities on the north face. The final vertex values are specified as

$$v(0,1,0) = \hat{v}_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + \overline{V}_{i,j+\frac{1}{2},k} - \hat{\bar{v}}_{\mathrm{N}}, \tag{113}$$

$$v(1,1,0) = \hat{v}_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + \overline{V}_{i,j+\frac{1}{2},k} - \hat{\bar{v}}_{\mathrm{N}}, \tag{114}$$

$$v(1,1,1) = \hat{v}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} + \overline{V}_{i,j+\frac{1}{2},k} - \hat{\bar{v}}_{\mathrm{N}}, \tag{115}$$

$$v(0,1,1) = \hat{v}_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} + \overline{V}_{i,j+\frac{1}{2},k} - \hat{\bar{v}}_{\mathrm{N}}. \tag{116}$$

The remaining vertex velocities are specified similarly.

By analogy with (61), the starting point for specifying the second-order slopes is to look at the matrix equation for vertex values of the reconstructed divergences,

$$\boldsymbol{\Phi} = A\boldsymbol{\Delta}^{(1)} + B\boldsymbol{\Delta}^{(2)}, \tag{117}$$

where now $\boldsymbol{\Phi}$ is an eight-vector which we choose to order based on a logical nesting of **for** loops as shown in Algorithms 1 and 2 below,

$$\boldsymbol{\Phi} = [\phi(0,0,0),\ \ \phi(0,0,1),\ \ \phi(0,1,0),\ \ \phi(0,1,1),\ \ \phi(1,0,0),\ \ \phi(1,0,1),\ \ \phi(1,1,0),\ \ \phi(1,1,1)]^{\mathrm{T}}. \tag{118}$$

The first- and second-order slopes are 12-vectors which we order in accord with the pattern suggested by (106)–(108),

$$\boldsymbol{\Delta}^{(1)} = [\Delta^{(1)}_{u,\mathrm{BS}}, \Delta^{(1)}_{u,\mathrm{BN}}, \Delta^{(1)}_{u,\mathrm{TS}}, \Delta^{(1)}_{u,\mathrm{TN}}, \Delta^{(1)}_{v,\mathrm{BW}}, \Delta^{(1)}_{v,\mathrm{BE}}, \Delta^{(1)}_{v,\mathrm{TW}}, \Delta^{(1)}_{v,\mathrm{TE}}, \Delta^{(1)}_{w,\mathrm{SW}}, \Delta^{(1)}_{w,\mathrm{NW}}, \Delta^{(1)}_{w,\mathrm{SE}}, \Delta^{(1)}_{w,\mathrm{NE}}]^{\mathrm{T}}, \tag{119}$$

$$\boldsymbol{\Delta}^{(2)} = [\Delta^{(2)}_{u,\mathrm{BS}}, \Delta^{(2)}_{u,\mathrm{BN}}, \Delta^{(2)}_{u,\mathrm{TS}}, \Delta^{(2)}_{u,\mathrm{TN}}, \Delta^{(2)}_{v,\mathrm{BW}}, \Delta^{(2)}_{v,\mathrm{BE}}, \Delta^{(2)}_{v,\mathrm{TW}}, \Delta^{(1)}_{v,\mathrm{TE}}, \Delta^{(2)}_{w,\mathrm{SW}}, \Delta^{(2)}_{w,\mathrm{NW}}, \Delta^{(2)}_{w,\mathrm{SE}}, \Delta^{(2)}_{w,\mathrm{NE}}]^{\mathrm{T}}. \tag{120}$$

The elements of the sparse $8 \times 12$ matrices $A$ and $B$ are thus given by Algorithms 1 and 2 below.

As in the 2D version, to specify the second-order slopes we first decompose $\boldsymbol{\Delta}^{(2)}$ into $\hat{\boldsymbol{\Delta}} + \boldsymbol{\delta}$. The values of $\hat{\boldsymbol{\Delta}}$ are specified by an estimate of the second spatial derivative for the edge component along the edge direction analogous to (70). The correction, $\boldsymbol{\delta}$, is then obtained from the LSMN solution

$$\boldsymbol{\delta} = B^{+}(\boldsymbol{\Theta} - A\boldsymbol{\Delta}^{(1)} - B\hat{\boldsymbol{\Delta}}), \tag{121}$$

where

$$\boldsymbol{\Theta} = [\theta_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}},\ \theta_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}},\ \theta_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}},\ \theta_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, \theta_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}},\ \theta_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}},\ \theta_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}},\ \theta_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}]^{\mathrm{T}}. \tag{122}$$

Note that $B^{+}$ (the Moore–Penrose pseudo-inverse of $B$) is easily obtained in practice using the Matlab function `pinv` (see e.g. [6] for further discussion on $B^{+}$).

Substituting the second-order slopes $\boldsymbol{\Delta}^{(2)} = \hat{\boldsymbol{\Delta}} + \boldsymbol{\delta}$ back into (117) and subtracting $\boldsymbol{\Theta}$ from both sides we obtain

$$\boldsymbol{\Phi} - \boldsymbol{\Theta} = (I - BB^{+})(A\boldsymbol{\Delta}^{(1)} - \boldsymbol{\Theta}), \tag{123}$$

**Algorithm 1**

Elements of $A$. Note that the column entries are taken from the coefficients as presented in (106)–(108). Also, the row ordering from the nested **for** loops leads to the ordering of the eight-vector $\mathbf{\Phi}$ given in (117).

```
A=zeros(8,l2); % Initialize A as an 8 × l2 matrix of all zeros.
irow=0; % Row index.
 for q=0:l
  for r=0:l
   for s=0:l
    irow=irow+l;
    A(irow,l)=(l-s)*(l-r);
    A(irow,2)=(l-s)*r;
    A(irow,3)=s*(l-r);
    A(irow,4)=s*r;
    A(irow,5)=(l-s)*(l-q);
    A(irow,6)=(l-s)*q;
    A(irow,7)=s*(l-q);
    A(irow,8)=s*q;
    A(irow,9)=(l-q)*(l-r);
    A(irow,l0)=(l-q)*r;
    A(irow,ll)=q*(l-r);
    A(irow,l2)=q*r;
   end for
  end for
end for
```

**Algorithm 2**

Elements of $B$. Note that the column entries are taken from the coefficients as presented in (106)–(108). Also, the row ordering from the nested **for** loops leads to the ordering of the eight-vector $\mathbf{\Phi}$ given in (117).

```
B=zeros (8,l2); % Initialize B as an 8 × l2 matrix of all zeros.
irow=0; % Row index.
 for q=0:l
  for r=0:l
   for s=0:l
    irow=irow+l;
    B(irow,l)=(l-s)*(l-r)*(q-0.5);
    B(irow,2)=(l-s)*r*(q-0.5);
    B(irow,3)=s*(l-r)*(q-0.5);
    B(irow,4)=s*r*(q-0.5);
    B(irow,5)=(l-s)*(l-q)*(r-0.5);
    B(irow,6)=(l-s)*q*(r-0.5);
    B(irow,7)=s*(l-q)*(r-0.5);
    B(irow,8)=s*q*(r-0.5);
    B(irow,9)=(l-q)*(l-r)*(s-0.5);
    B(irow,l0)=(l-q)*r*(s-0.5);
    B(irow,ll)=q*(l-r)*(s-0.5);
    B(irow,l2)=q*r*(s-0.5);
   end for
  end for
end for
```

which is similar to the 2D result (74). Here we have $(I - BB^+)A = \frac{1}{4}\mathbf{ones}(8,12)$ and $I - BB^+ = \frac{1}{8}\mathbf{ones}(8,8)$, where $\mathbf{ones}(m,n)$ is an $m \times n$ matrix of all ones. Therefore, at the "bottom-south–west" vertex ($q = r = s = 0$), for example, we have

$$\phi(0,0,0) = \theta_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + h(\nabla_h \cdot \overline{\mathbf{U}}) - \frac{1}{8}\mathbf{sum}(\mathbf{\Theta}), \tag{124}$$

where $\mathbf{sum}(\mathbf{\Theta})$ is the summation of the elements of $\mathbf{\Theta}$ from (122). Thus, the 3D reconstruction also reduces to zero divergence everywhere for the incompressible case. In fact, the 3D reconstruction possesses all the same properties as the 2D case, which are listed in Section 3.2.

## Appendix B. Extension to non-uniform grids

In this appendix we first consider the reconstruction for cell $(i,j)$ on a 2D non-uniform grid. We then summarize the procedure for non-uniform grids in 3D.

In 2D, the grid spacings in the $x$-direction are $[\alpha h_i, h_i, \beta h_i]$ for cells in the $[i-1, i, i+1]$ columns, respectively. Similarly, in the $y$-direction we have grid spacings $[\kappa g_j, g_j, \gamma g_j]$ for cells in rows $[j-1, j, j+1]$. A sketch of the grid is shown in Fig. 23. Note that we do not require smoothness restrictions on the relationships between $\alpha$ and $\beta$ or $\kappa$ and $\gamma$. We assume only that these parameters are constant, positive, and $\mathcal{O}(1)$.

A key difference between the uniform and non-uniform reconstructions is the scaling of the interpolated divergence at the cell vertices. We must first compute the scaled discrete divergence, denoted $\Psi$, for all the cells in the neighborhood of $(i,j)$,

$$\Psi^{(i,j)}_{\ell,m} = h_i\left(\frac{\overline{U}_{\ell+\frac{1}{2},m} - \overline{U}_{\ell-\frac{1}{2},m}}{h_\ell}\right) + g_j\left(\frac{\overline{V}_{\ell,m+\frac{1}{2}} - \overline{V}_{\ell,m-\frac{1}{2}}}{g_m}\right), \tag{125}$$

for $\ell = \{i-1, i, i+1\}$ and $m = \{j-1, j, j+1\}$. The subscript $\ell + \frac{1}{2}$, for example, is used to identify a staggered velocity component on the "east" face of a cell centered at $x_\ell$ (which is of width $h_\ell$). Note that the nine elements of the array $\Psi^{(i,j)}$ are unique to cell $(i,j)$.

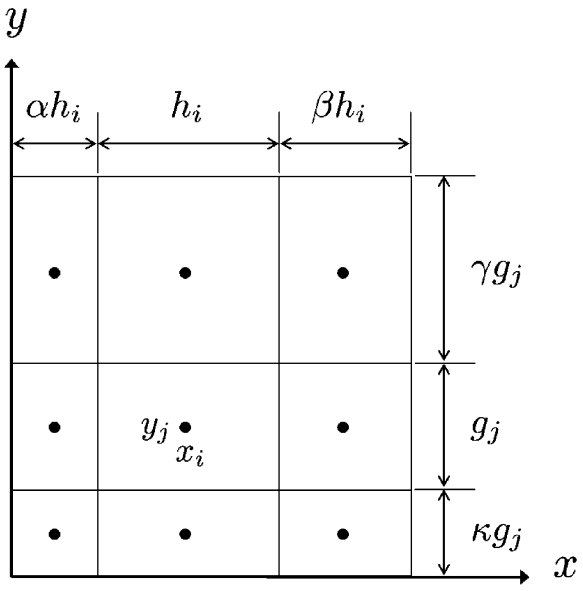


Fig. 23. Non-uniform grid in 2D. For cell $(i,j)$ centered at $(x_i, y_j)$ the cell width and height are $h_i$ and $g_j$, respectively. As shown, the neighboring cell widths and heights vary by the $\mathcal{O}(1)$ factors $\alpha, \beta, \kappa$, and $\gamma$.

The interpolated vertex divergence values $\boldsymbol{\Theta}$ are determined from bilinear interpolation of the elements of $\Psi^{(i,j)}$. For example, at the south–west vertex we have

$$\theta_{i-\frac{1}{2},j-\frac{1}{2}} = \Psi^{(i,j)}_{i-1,j-1}\left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\kappa}\right) + \Psi^{(i,j)}_{i,j-1}\left(\frac{\alpha}{1+\alpha}\right)\left(\frac{1}{1+\kappa}\right) + \Psi^{(i,j)}_{i-1,j}\left(\frac{1}{1+\alpha}\right)\left(\frac{\kappa}{1+\kappa}\right) + \Psi^{(i,j)}_{i,j}\left(\frac{\alpha}{1+\alpha}\right)\left(\frac{\kappa}{1+\kappa}\right). \tag{126}$$

Temporary vertex velocities are obtained using the "inverse lever rule." For example, the north–east temporary vertex value for the $u$-component is computed by

$$\hat{u}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{\overline{U}_{i+\frac{1}{2},j+1} + \gamma\overline{U}_{i+\frac{1}{2},j}}{1+\gamma}. \tag{127}$$

The final vertex values are then obtained in the same manner as in the uniform-grid case (see Eqs. (36), (37), (39), (45)–(47)). Similarly, the parabolic edge velocities and first-order slopes do not require modification for non-uniform grids (see Eqs. (48)–(55)).

We obtain the specified second-order slopes $\hat{\boldsymbol{\Delta}}$ from an estimate of the second spatial derivative in the edge direction (i.e., the direction parallel with the edge) evaluated at the edge center. As an example, consider an edge centered at $x_i$; the specified second-order slope is given by (the $y$ locations of the temporary vertex velocities are suppressed to avoid clutter)

$$\hat{\Delta}_i = \frac{K_\alpha\hat{u}_{i+(\frac{1}{2}+\beta)} - K_\beta\hat{u}_{i-(\frac{1}{2}+\alpha)} - \left([1+\beta]K_\alpha + \alpha K_\beta\right)\hat{u}_{i+\frac{1}{2}} + \left(\beta K_\alpha + [1+\alpha]K_\beta\right)\hat{u}_{i-\frac{1}{2}}}{K_\alpha\left(\frac{[1/2+\beta]^2}{2} - \frac{1}{8}\right) - K_\beta\left(\frac{[1/2+\alpha]^2}{2} - \frac{1}{8}\right)}, \tag{128}$$

where the factors $K_\alpha$ and $K_\beta$ are defined by

$$K_\alpha \equiv \frac{\left[\frac{1}{2}+\alpha\right]}{24} - \frac{\left[\frac{1}{2}+\alpha\right]^3}{6} \tag{129}$$

and

$$K_\beta \equiv \frac{\left[\frac{1}{2}+\beta\right]^3}{6} - \frac{\left[\frac{1}{2}+\beta\right]}{24}. \tag{130}$$

Note that (128) is second-order accurate, that is,

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{x=x_i} = \frac{\hat{\Delta}_i}{h_i^2} + \mathcal{O}(h_i^2) \tag{131}$$

and reduces to (70) for a uniform grid, $\alpha = \beta = 1$. A one-sided difference similar to (128) can be derived for use near boundaries.

With all the parameters of the reconstruction established as described above in this appendix, the second-order slope correction $\boldsymbol{\delta}$ is obtained by (71) and the second-order slopes are given by $\boldsymbol{\Delta}^{(2)} = \hat{\boldsymbol{\Delta}} + \boldsymbol{\delta}$ without loss of accuracy. The reconstructed velocity for cell $(i,j)$ is then given by (26) and (27), with $q = q_i$ and $r = r_j$ as defined by (21) and (22), respectively. The resulting reconstruction on non-uniform grids retains all the properties summarized in the conclusions (Section 7).

The extension to 3D, non-uniform grids may be approached similarly:

1. The respective $x$-direction, $y$-direction, and $z$-direction terms of the scaled discrete divergence $\Psi^{(i,j,k)}$ for cell $(i,j,k)$ are scaled by the local cell width, height, and depth $[h_i, g_j, f_k]$ analogous to (125). Note that the divergence array has 27 elements per cell: $\Psi^{(i,j,k)}_{\ell,m,n}$, for $\ell = \{i-1, i, i+1\}, m = \{j-1, j, j+1\}$, and $n = \{k-1, k, k+1\}$.
2. The interpolated vertex divergence $\boldsymbol{\Theta}$ is obtained from trilinear interpolation of the elements of $\Psi^{(i,j,k)}$.

3. The temporary vertex values are obtained from bilinear interpolation of the staggered-grid velocity data.
4. The final vertex values, the parabolic edge velocities, and the first-order slopes are obtained in the same manner as the uniform-grid case (see Appendix A).
5. The specified second-order slopes $\hat{\boldsymbol{\Delta}}$ are obtained by (128).
6. The second-order slope correction $\boldsymbol{\delta}$ is obtained in the same manner as the uniform-grid case (see Eq. (121)) and the second-order slopes are given by $\boldsymbol{\Delta}^{(2)} = \hat{\boldsymbol{\Delta}} + \boldsymbol{\delta}$.
7. The 3D reconstruction is given by (102)–(104), as in the uniform case, with $q = q_i, r = r_j$, and $s = s_k$ as defined by (99)–(101), respectively.

## Appendix C. Formal accuracy of the parabolic edge reconstruction method

In this appendix we derive the formal accuracy of the parabolic edge reconstruction method (PERM). It is sufficient to consider the $u$ component for the 2D reconstruction. For the purpose of assessing accuracy, we imagine that there exists a smooth, continuous, Eulerian field $U(\mathbf{x}, t)$ at some time $t$ (time is henceforth omitted from the argument list). This field represents the particle conditional mean field $\langle U^*|\mathbf{x}\rangle$, for example. As depicted in Fig. 24, we imagine a cell centered at $(x_i \equiv ih - \frac{h}{2}, y_j \equiv jg - \frac{g}{2})$ where $h$ and $g$ are the uniform grid spacings in the $x$- and $y$-directions, respectively. For simplicity, we only consider $h = g$, but, as we will see, distinguishing these parameters makes the results easier to interpret. To exploit symmetry, we work in terms of the local cell coordinates $\xi \equiv x - x_i$ and $\eta \equiv y - y_j$, which give the position relative to the cell center (note that these differ from the local coordinates used earlier in the paper). The staggered-grid velocity data are stored on the "east" and "west" faces of the cell. Here we denote these data by $\overline{U}_{i+\frac{1}{2},j}$ and $\overline{U}_{i-\frac{1}{2},j}$ for all $i, j$, respectively.

The PERM reconstruction for the $u$ component in terms of the new local cell coordinates is

$$u(\xi,\eta) = \left(\frac{1}{2} - \frac{\eta}{g}\right)\left[\bar{u}_{\mathrm{S}} + \frac{\xi}{h}\Delta^{(1)}_{u,\mathrm{S}} + \frac{1}{2}\left(\frac{\xi^2}{h^2} - \frac{1}{4}\right)\Delta^{(2)}_{u,\mathrm{S}}\right] + \left(\frac{1}{2} + \frac{\eta}{g}\right)\left[\bar{u}_{\mathrm{N}} + \frac{\xi}{h}\Delta^{(1)}_{u,\mathrm{N}} + \frac{1}{2}\left(\frac{\xi^2}{h^2} - \frac{1}{4}\right)\Delta^{(2)}_{u,\mathrm{N}}\right], \tag{132}$$

where $\bar{u}_j, \Delta^{(1)}_{u,j}$ and $\Delta^{(2)}_{u,j}$ are the parabolic edge velocity, the first-order slope and second-order slope, respectively, for edge $j$. Ultimately, each of these parameters derives from the staggered-grid velocity data.

The Taylor series expansion about the local cell origin (i.e., the Maclaurin series) for the smooth velocity field is given by

$$U(\xi,\eta) = U + \xi U_x + \eta U_y + \frac{\xi^2}{2}U_{xx} + \xi\eta U_{xy} + \frac{\eta^2}{2}U_{yy} + \frac{\xi^3}{6}U_{xxx} + \frac{\xi^2\eta}{2}U_{xxy} + \frac{\xi\eta^2}{2}U_{xyy} + \frac{\eta^3}{6}U_{yyy} + \cdots \tag{133}$$

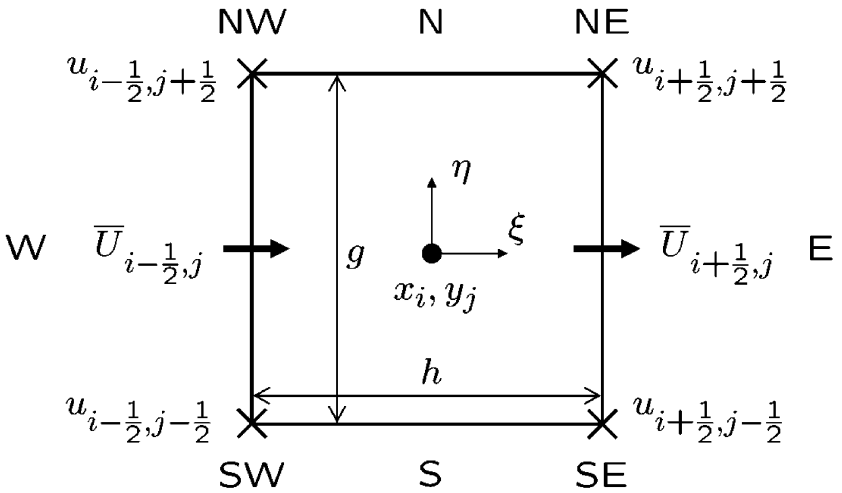


Fig. 24. A 2D cell showing the local cell-centered coordinates $\xi \equiv x - x_i$ and $\eta \equiv y - y_j$ used for the accuracy analysis. The staggered-grid velocity data, e.g. $\overline{U}_{i+\frac{1}{2},j}$ are stored on the cell faces. Ultimately, these data form all the parameters of the reconstruction: the parabolic edge velocities and the first-order and second-order slopes. Also shown are the vertex locations marked by the ×'s. The grid spacing is $h$ in the $x$-direction and $g$ in the $y$-direction.

where suffix notation is used to denote partial derivatives. Note that, unless otherwise indicated, $U$ and its derivatives are evaluated at the cell center. Hence, $U_x$ is a compact notation for $\left(\frac{\partial U}{\partial x}\right)_{x_i,y_j}$. In some circumstances below, where the derivative is evaluated away from the cell center, we also use a bracketed superscript to denote a partial derivative, e.g. $U^{(yy)}_{i+\frac{1}{2},j} = \left(\frac{\partial^2 U}{\partial y^2}\right)_{x_i+\frac{h}{2},y_j}$.

### C.1. Accuracy of the PERM velocity components

It can be shown that the PERM reconstruction for the $u$ component in terms of the cell-centered smooth velocity is

$$u(\xi,\eta) = U + \xi U_x + \eta U_y + \frac{\xi^2}{2}U_{xx} + \xi\eta U_{xy} + \frac{g^2}{24}U_{yy} + \frac{h^2\xi}{24}U_{xxx} + \frac{\xi^2\eta}{2}U_{xxy} + \frac{\xi g^2}{24}U_{xyy} + \frac{5g^2\eta}{24}U_{yyy} + \mathcal{O}(\xi^\alpha\eta^\beta h^\gamma g^\iota), \tag{134}$$

where $\alpha,\beta,\gamma$, and $\iota$ are nonnegative integers which sum to four. Since we have $h = g$, $\xi = \mathcal{O}(h)$ and $\eta = \mathcal{O}(g)$, the last line in (134) can be written as $\mathcal{O}(h^4)$. By comparing (133) and (134) term by term, we see that the error in the reconstructed velocity is $\mathcal{O}(h^3) + \mathcal{O}(g^2)$.

Note that while the $U_{xxx}$ term is not exact, $\xi$ is at most $\frac{h}{2}$, in which case $\frac{\xi^3}{6} = \frac{h^2\xi}{24}$ and this term is exact on the east and west faces of the cell. Further, as a consequence, this term is continuous from cell to cell in the velocity-component direction. Of course, the $U_{xxx}$ term is also exact at the cell center, since $\xi = 0$ there.

Taking $h$ to be a constant multiple of $g$, upon refinement of the grid we observe $\mathcal{O}(h^2)$ convergence in general due to the $U_{yy}$ error term. Hence, the method is formally second-order accurate. Notice, however, that if the $U_{yy}$ error term is zero (which occurs if the particle position is such that $|\eta| = g/\sqrt{12}$ or if the $U$ field is linear in $y$) then the scheme is third-order accurate. These observations are confirmed by convergence studies in Appendix D below.

It is also worth noting that the correction $\boldsymbol{\delta}$ to the second-order slopes is $\mathcal{O}(h^4)$. This is an improvement over the interpolation scheme developed by Jenny et al. [12], as we briefly describe below in Section C.3 of this appendix.

### C.2. PERM derivative accuracy

The Maclaurin series for $U_x$ is

$$U_x(\xi,\eta) = U_x + \xi U_{xx} + \eta U_{xy} + \frac{\xi^2}{2}U_{xxx} + \xi\eta U_{xxy} + \frac{\eta^2}{2}U_{xyy} + \cdots \tag{135}$$

Differentiating (134) with respect to $\xi$ we obtain

$$u_x(\xi,\eta) = U_x + \xi U_{xx} + \eta U_{xy} + \frac{h^2}{24}U_{xxx} + \xi\eta U_{xxy} + \frac{g^2}{24}U_{xyy} + \mathcal{O}(h^3). \tag{136}$$

Hence, by comparing (135) and (136), we see that the representation of the derivative in the velocity-component direction is second-order accurate.

The Maclaurin series for $U_y$ is

$$U_y(\xi,\eta) = U_y + \xi U_{xy} + \eta U_{yy} + \frac{\xi^2}{2}U_{xxy} + \xi\eta U_{xyy} + \frac{\eta^2}{2}U_{yyy} + \cdots \tag{137}$$

Differentiating (134) with respect to $\eta$ and noting $\frac{\partial u}{\partial y} = \frac{\partial u}{\partial \eta}$ we obtain

$$u_y(\xi,\eta) = U_y + \xi U_{xy} + \mathcal{O}(h^2). \tag{138}$$

Note that (138) does not contain the $\eta U_{yy}$ term from the exact expansion (137). With $\eta = \mathcal{O}(g)$, the derivative in the direction normal to the velocity component is represented to first-order accuracy, as expected due to the linear reconstruction.

### C.3. Accuracy using LSMN second-order slopes

When cast in the PERM framework, the interpolation scheme of Jenny et al. [12] is recovered by using the following least-squares/minimum-norm (LSMN) solution for the second-order slopes,

$$\boldsymbol{\Delta}^{(2)} = B^{+}(\boldsymbol{\Theta} - A\boldsymbol{\Delta}^{(1)}). \tag{139}$$

Using (139) together with the parabolic edge velocities and first-order slopes[5] presented in the main body of this paper, the reconstruction (132) can be rewritten in terms of the smooth velocity and its derivatives at the cell center as follows:

$$u(\xi,\eta) = U + \xi U_x + rU_y + \frac{\xi^2}{2}U_{xx} + \xi\eta U_{xy} + \frac{g^2}{24}U_{yy} + \frac{h^2 q}{24}U_{xxx} + \left(\frac{h^2\eta}{16} + \frac{\xi^2\eta}{4}\right)U_{xxy} + \frac{\xi g^2}{24}U_{xyy} + \frac{5g^2\eta}{24}U_{yyy} + \left(\frac{\xi^2\eta}{4} - \frac{hgr}{16}\right)V_{xyy} + \mathcal{O}(h^4). \tag{140}$$

By comparing (134) and (140), we notice that the leading order differences are found in the $U_{xxy}$ and $V_{xyy}$ terms. For PERM these terms are exact (compare (134) with (133)). When using the LSMN second-order slopes, however, both these terms are in error. Of particular concern is fact that the $u$-component reconstruction depends upon derivatives of the $v$-component field. The method for determining the PERM second-order slopes–specifying the second-order slopes $\hat{\boldsymbol{\Delta}}$ and then using the LSMN solution for the correction $\boldsymbol{\delta}$–forces the $v$-component dependence on $u$ to be $\mathcal{O}(h^4)$. In this sense PERM offers an improvement to the interpolation scheme of Jenny et al. As we show below in Appendix D, however, the degree of improvement is problem-dependent: either the $U_{xxy}$ term or the $V_{xyy}$ term in (140) must be of significance in order to see a difference between the PERM and LSMN errors.

It is also useful to examine the expansion of the velocity-component derivatives obtained using the LSMN second-order slopes. Differentiating (140) we obtain

$$u_x(\xi,\eta) = U_x + \xi U_{xx} + \eta U_{xy} + \frac{h^2}{24}U_{xxx} + \frac{\xi\eta}{2}(U_{xxy} + V_{xyy}) + \frac{g^2}{24}U_{xyy} + \mathcal{O}(h^3). \tag{141}$$

Notice that again the $v$-component cross derivative $V_{xyy}$ shows up in the sixth term of the expansion causing an additional $\mathcal{O}(h^2)$ error for the $u$-component derivative.

Similarly, the expansion for the $v$-component $y$ derivative is

$$v_y(\xi,\eta) = V_y + \xi V_{xy} + \eta V_{yy} + \frac{h^2}{24}V_{xxy} + \frac{\xi\eta}{2}(U_{xxy} + V_{xyy}) + \frac{g^2}{24}V_{yyy} + \mathcal{O}(h^3). \tag{142}$$

Notice that by adding (141) and (142) we fortuitously eliminate the $U_{xxy}$ and $V_{xyy}$ cross-derivative errors for the divergence: $\xi\eta(U_{xxy} + V_{xyy})$ is an exact term in the expansion for the divergence of the smooth velocity field. Thus, as we confirm below in Appendix D, to leading order the divergence error obtained using the LSMN second-order slopes is equivalent to that obtained using the PERM second-order slopes. This is true even though the velocity components and the component derivatives obtained from the two methods are not equivalent.

## Appendix D. Verification of the rate of convergence

In this appendix we present convergence studies to confirm the order of accuracy suggested by the analysis of Appendix C for both the velocity component and divergence fields on a uniform grid. As discussed previously, the convergence of the velocity-component error may be affected by cross-component derivatives. Thus, below in Appendix D.1 we present two convergence studies for the velocity: the first utilizes a sinusoidal field which is representative of well-resolved combustion problems and the second utilizes poly-

[5] Note that exclusion of the MINMOD limiter, used by Jenny et al. [12] in determining the first-order slopes for their RANS application, eliminates a potentially $\mathcal{O}(h)$ error (depending on the degree of limitation) that would otherwise contaminate the reconstructed velocity field. Central differencing without a limiter is preferred in low-Mach LES.

nomial functions specifically designed to target certain terms in the Taylor expansion. This latter case highlights the potential advantages that the PERM reconstruction has over the Jenny interpolation scheme. In Appendix D.2 we present a convergence study for the divergence field utilizing the polynomial fields from the second velocity case. This test is designed to confirm that the correction does not degrade the second-order accuracy of the reconstructed divergence field for PERM and also confirms that the cross-component derivative errors present in the LSMN reconstruction (Jenny's scheme) cancel for the divergence. Hence, to leading order the LSMN and PERM divergence errors are equivalent.

As in Section 5, we consider a square, periodic domain of side $L = 2\pi$. We examine the error for particles located in a square cell of side $h = L/N_x$ centered at $(L/2, L/2)$. The cell domain, denoted $\Omega$, is partitioned into an $m \times m$ uniform grid with subcell spacing $h/m$. We locate one particle in the center of each subcell. There are thus $N_c = m^2$ particles in the cell. For the cases presented in this section we set $m = 20$ (i.e., $N_c = 400$).

### D.1. Velocity component error

For the sinusoidal case the exact velocity field is specified by (92) and (93) with $t = 0$ for $U(x, y)$ and $V(x, y)$, respectively. This field is rotational and exhibits a non-zero divergence. The staggered-grid velocity data are sampled from the exact field at the staggered locations. The discrete cell divergence is then taken from (23).

The measure of the error for the velocity component, denoted $\varepsilon_u$, is taken as the average difference within the cell between the reconstruction and the exact velocity component. Using a finite number of particles, the error is computed as

$$\varepsilon_u \equiv \frac{1}{h^2}\int_\Omega |u(\mathbf{x}) - U(\mathbf{x})|\,\mathrm{d}\mathbf{x} \approx \frac{1}{N_c}\sum_{i=1}^{N_c} |u(\mathbf{X}^{(i)}) - U(\mathbf{X}^{(i)})|. \tag{143}$$

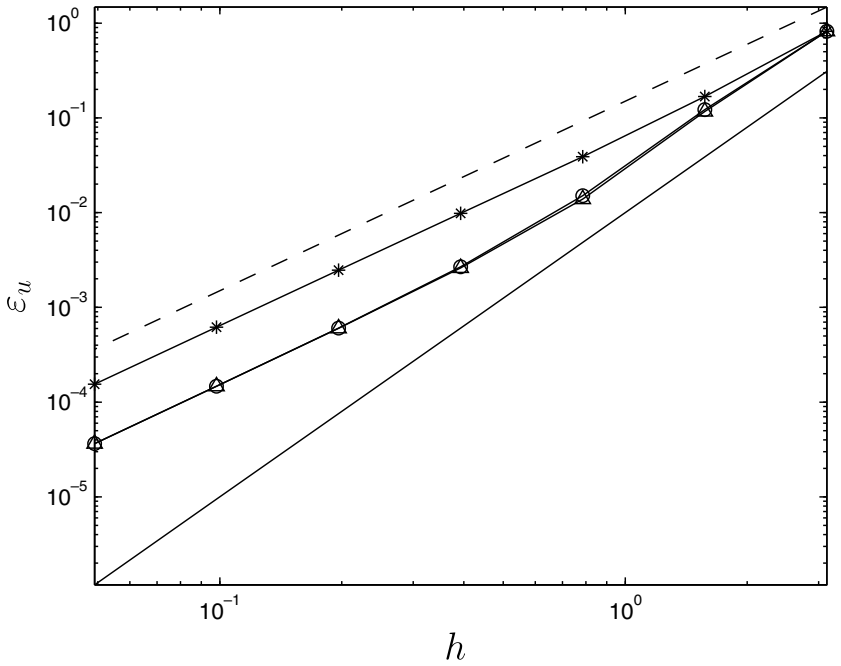


Fig. 25. Convergence for the $u$ component of velocity–sinusoidal case. The exact velocity field is given by (92) and (93) with $t = 0$. The exact divergence of this field is given by (94) with $t = 0$. The staggered-grid velocity data are obtained by sampling the exact field at the staggered locations. The error $\varepsilon_u$ in the reconstructed velocity component (see Eq. (143)) is plotted against the grid spacing $h = L/N_x$ where $L = 2\pi, N_x = 2^n$, and $n = \{1, 2, \ldots, 7\}$. The dashed line indicates $\mathcal{O}(h^2)$ convergence and the solid line represents $\mathcal{O}(h^3)$ convergence. The bilinear case (stars) shows second-order convergence. Both the LSMN (circles) and the PERM (triangles) reconstructions are formally second-order accurate and show little difference for this test problem.

In Fig. 25 we plot the error $\varepsilon_u$ for a range of grid spacings ($h = L/N_x$ with $N_x = 2^n$ and $n = \{1, 2, \ldots, 7\}$) for the bilinear (stars), LSMN (circles), and PERM (triangles) reconstructions. In this case we maintain equal grid spacings in each direction. The PERM reconstruction for the $u$ component is obtained from (26) using corrected second-order slopes. The LSMN reconstruction (equivalent to the Jenny scheme without slope limiting) is obtained by using (139) for the second-order slopes. The bilinear reconstruction is obtained by simply setting the second-order slopes to zero. As expected, the bilinear case achieves second-order convergence. The LSMN and PERM cases show similar convergence behavior. Both are formally second-order accurate, but exhibit third-order convergence at coarse grid resolutions for this problem which is complex and generally representative of combustion modeling problems.

The second case is designed to exercise specific terms in the Taylor expansion due to cross-component derivatives. To this end, we specify the following smooth velocity field,

$$U(x, y) = x^3 + y, \tag{144}$$

$$V(x, y) = xy^3. \tag{145}$$

The divergence of this field is

$$\nabla \cdot \mathbf{U} = 3x^2 + 3xy^2. \tag{146}$$

The staggered-grid velocity data are sampled from (144) and (145) and the discrete cell divergence is then taken from (23).

In Fig. 26 again we plot the error $\varepsilon_u$ for a range of grid spacings ($h = L/N_x$ with $N_x = 2^n$ and $n = \{1, 2, \ldots, 7\}$) for the bilinear (stars), LSMN (circles), and PERM (triangles) reconstructions. The bilinear scheme is $\mathcal{O}(h^2)$. Both LSMN and PERM show third-order convergence since the $U_{yy}$ error term is absent for this problem. However, for the LSMN case the $V_{xyy}$ term contaminates the error and so the PERM case converges more rapidly at the coarse grid resolutions. In such instances, PERM shows improved accuracy over

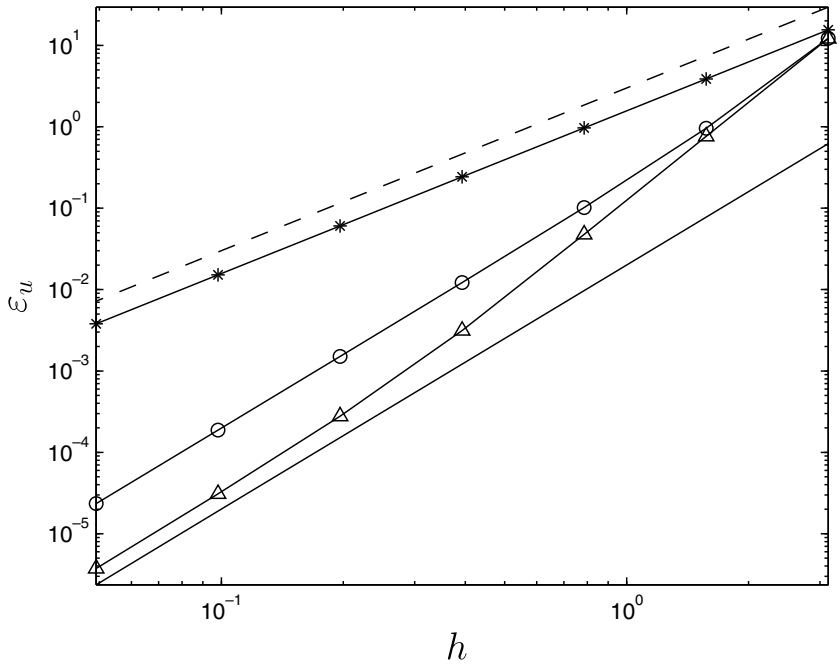


Fig. 26. Convergence for the $u$ component of velocity–polynomial case. The exact velocity field is given by (144) and (145). The exact divergence of this field is given by (146). The staggered-grid velocity data are obtained by sampling the exact field at the staggered locations. The error $\varepsilon_u$ in the reconstructed velocity component (see Eq. (143)) is plotted against the grid spacing $h = L/N_x$ where $L = 2\pi, N_x = 2^n$, and $n = \{1, 2, \ldots, 7\}$. The dashed line indicates $\mathcal{O}(h^2)$ convergence and the solid line represents $\mathcal{O}(h^3)$ convergence. The bilinear case (stars) shows second-order convergence. Ultimately, both the LSMN (circles) and PERM (triangles) cases attain third-order convergence. However, due to the presence of cross-component derivative errors in the LSMN reconstruction the PERM error converges more rapidly at the coarse grid resolutions for this test problem.

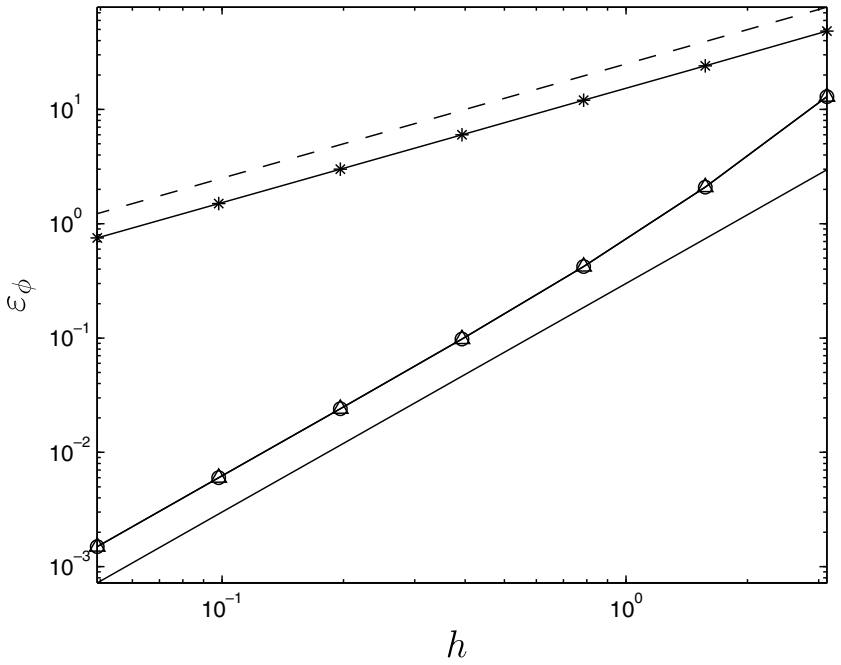


Fig. 27. Convergence for the divergence field. The exact velocity is given by the polynomial fields (144) and (145). The exact divergence of this field is given by (146). The staggered-grid velocity data are obtained by sampling the exact field at the staggered locations. The error $\varepsilon_\phi$ in the reconstructed divergence (see Eq. (147)) is plotted against the grid spacing $h = L/N_x$ where $L = 2\pi, N_x = 2^n$, and $n = \{1, 2, \ldots, 7\}$. The dashed line indicates $\mathcal{O}(h)$ convergence and the solid line represents $\mathcal{O}(h^2)$ convergence. The bilinear case (stars) shows first-order convergence. The LSMN (circles) and PERM (triangles) reconstructions are second-order accurate. To leading order, the errors are identical due to a cancelation of cross-component derivative errors in the LSMN reconstruction. This is behavior is true in general; it is not problem-dependent.

the Jenny scheme for the velocity components and the individual component derivatives, though, as we show in the next section, the divergence error is not improved.

### D.2. Divergence error

In this section we examine the error in the divergence of the reconstructed field. Here, the exact velocity field is specified by (144) and (145). The discrete divergence for each cell is then given by (23). By analogy with (143) we define and compute the measure of the divergence error by

$$\varepsilon_\phi \equiv \frac{1}{h^2}\int_\Omega |\nabla\cdot\mathbf{u} - \nabla\cdot\mathbf{U}|\,\mathrm{d}\mathbf{x} \approx \frac{1}{N_c}\sum_{i=1}^{N_c}\left|\frac{\phi(\mathbf{X}^{(i)})}{h} - (\nabla\cdot\mathbf{U})_{\mathbf{x}=\mathbf{X}^{(i)}}\right|, \tag{147}$$

where $\phi(\mathbf{x})$ is given by (28) and the divergence of the exact field is given by (146).

The results of the convergence study are shown in Fig. 27 for the bilinear (stars), LSMN (circles), and PERM (triangles) reconstructions. The bilinear case is first-order accurate. Both the LSMN and PERM cases show second-order convergence. As discussed, to leading order the divergence errors are identical even though the individual velocity component and derivative errors deviate (with PERM being more accurate) due to a cancelation of the errors in the cross-component derivative terms of the Taylor expansion (refer to the discussion at the end of Appendix C.3).

## References

[1] A.S. Almgren, J.B. Bell, W.Y. Crutchfield, Approximate Projection Methods: Part I. Inviscid Analysis, Lawrence Berkeley National Laboratories, Berkeley, CA, year unknown.
[2] B. Banerjee, Material point method simulations of fragmenting cylinders, in: 17th ASCE Engineering Mechanics Conference, University of Delaware, Newark, DE, 2004.

[3] S.G. Bardenhagen, E.M. Kober, The generalized interpolation material point method, Comp. Model. Eng. Sci. 5 (2004) 477–496.
[4] J. Bell, AMR for Low Mach Number Reacting Flow, Lawrence Berkeley National Laboratory Paper LBNL-54351, 2004.
[5] P. Colella, P. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comp. Phys. 54 (1984) 174–201.
[6] J.W. Demmel, Applied Numerical Linear Algebra, SIAM, 1997.
[7] O. Desjardins, G. Blanquart, G. Balarac, H. Pitsch, High order conservative finite difference scheme for variable density low mach number turbulent flows, J. Comp. Phys. (submitted for publication).
[8] J.E. Fromm, A method for reducing dispersion in convective differencing schemes, J. Comp. Phys. 3 (1968) 176–189.
[9] F. Gao, E.E. O'Brien, A large-eddy simulation scheme for turbulent reacting flows, Phys. Fluids A 5 (1993) 1282.
[10] S. Gottlieb, C.W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.
[11] F.A. Jaberi, P.J. 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.
[12] P. Jenny, S.B. Pope, M. Muradoglu, D.A. Caughey, A hybrid algorithm for the joint PDF equation of turbulent reactive flows, J. Comp. Phys. 166 (2001) 218–252.
[13] A. Kempf, Large-eddy simulation of non-premixed turbulent flames, Ph.D. thesis, Technischen Universität Darmstadt, 2004.
[14] R. McDermott, Toward one-dimensional turbulence subgrid closure for large-eddy simulation, Ph.D. thesis, The University of Utah, 2005.
[15] R. McDermott, S.B. Pope, A particle formulation for treating differential diffusion in filtered density function methods, J. Comp. Phys. 226 (1) (2007) 947–993.
[16] K. McGrattan, S. Hostikka, J. Floyd, H. Baum, R. Rehm, Fire Dynamics Simulator (Version 5) Technical Reference Guide, NIST Special Pub. 1018-5, <http://fire.nist.gov/fds/>, 2007.
[17] E.H. Moore, On the reciprocal of the general algebraic matrix, Bull. Am. Math. Soc. 26 (1920) 394–395.
[18] 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. Comp. Phys. 172 (2001) 841–878.
[19] R.L. Panton, Incompressible Flows, second ed., John Wiley and Sons, 1996.
[20] R. Penrose, A generalized inverse for matrices, in: Proceedings of the Cambridge Philosophical Society, vol. 51, 1955, pp. 406–413.
[21] R. Penrose, On best approximation solution of linear matrix equations, in: Proceedings of the Cambridge Philosophical Society, vol. 52, 1956, pp. 17–19.
[22] C.D. Pierce, Progress-variable approach for large-eddy simulation of turbulent combustion, Ph.D. thesis, Stanford University, 2001.
[23] S.B. Pope, PDF methods for turbulent reactive flows, Prog. Energy Combust. Sci. 11 (1985) 119–192.
[24] S.B. Pope, Computations of turbulent combustion: progress and challenges, in: Proceedings of Comb. Inst., vol. 30, 1990, pp. 591–612.
[25] S.B. Pope, Turbulent Flows, Cambridge, 2000.
[26] P.P. Popov, R. McDermott, S.B. Pope, An accurate time advancement algorithm for particle tracking, J. Comp. Phys. (submitted for publication).
[27] V. Raman and H. Pitsch, A consistent LES/filtered-density function formulation for the simulation of turbulent flames with detailed chemistry. in: Proceedings of Comb. Inst., vol. 31, 2007, pp. 1711–1719.
[28] V. Raman, H. Pitsch, R.O. Fox, Hybrid large-eddy simulation/Lagrangian filtered-density function approach for simulating turbulent combustion, Combust. Flame 143 (2005) 56–78.
[29] M.R.H. Sheikhi, T.G. Drozda, P. Givi, F.A. Jaberi, S.B. Pope, Large-eddy simulation of a turbulent nonpremixed piloted methane jet flame (Sandia Flame D), in: Proceedings of Comb. Inst., vol. 30, 2005, pp. 549–556.
[30] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, second ed., Springer, 1999.
[31] P.K. Yeung, S.B. Pope, An algorithm for tracking fluid particles in numerical simulations of homogeneous turbulence, J. Comp. Phys. 79 (1988) 373–416.
[32] Y.Z. Zhang, D.C. Haworth, A general mass consistency algorithm for hybrid particle/finite-volume PDF methods, J. Comp. Phys. 194 (2004) 156–193.