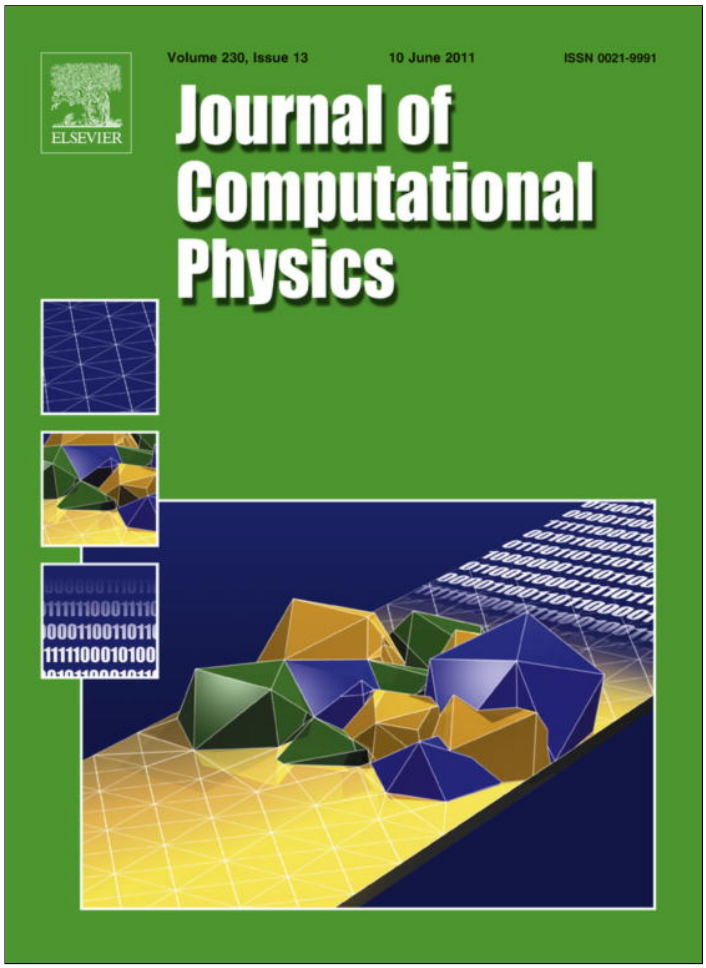

Journal of Computational Physics 230 (2011) 5328–5352

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Realizable high-order finite-volume schemes for quadrature-based moment methods

V. Vikas [a], Z.J. Wang [a,*], A. Passalacqua [b], R.O. Fox [b]

[a] Department of Aerospace Engineering, 2271 Howe Hall, Iowa State University, Ames, IA 50011, USA
[b] Department of Chemical and Biological Engineering, 2114 Sweeney Hall, Iowa State University, Ames, IA 50011, USA

ARTICLE INFO

Article history:
Received 12 May 2010
Received in revised form 5 February 2011
Accepted 21 March 2011
Available online 27 March 2011

Keywords:
Kinetic equation
Quadrature method of moments
Number density function
Realizablility
Finite-volume scheme
Unstructured grids

ABSTRACT

Dilute gas–particle flows can be described by a kinetic equation containing terms for spatial transport, gravity, fluid drag and particle–particle collisions. However, direct numerical solution of kinetic equations is often infeasible because of the large number of independent variables. An alternative is to reformulate the problem in terms of the moments of the velocity distribution. Recently, a quadrature-based moment method was derived for approximating solutions to kinetic equations. The success of the new method is based on a moment-inversion algorithm that is used to calculate non-negative weights and abscissas from the moments. The moment-inversion algorithm does not work if the moments are non-realizable, which might lead to negative weights. It has been recently shown [14] that realizability is guaranteed only with the 1st-order finite-volume scheme that has an inherent problem of excessive numerical diffusion. The use of high-order finite-volume schemes may lead to non-realizable moments. In the present work, realizability of the finite-volume schemes in both space and time is discussed for the 1st time. A generalized idea for developing realizable high-order finite-volume schemes for quadrature-based moment methods is presented. These finite-volume schemes give remarkable improvement in the solutions for a certain class of problems. It is also shown that the standard Runge–Kutta time-integration schemes do not guarantee realizability. However, realizability can be guaranteed if strong stability-preserving (SSP) Runge–Kutta schemes are used. Numerical results are presented on both Cartesian and triangular meshes.

1. Introduction

Kinetic equations occur in mesoscopic models for many physical phenomena, such as rarefied gases [7,10,11,24,35,47], plasmas [8,29,55], multiphase flows [14,44,48,54], optics [3,15,21,42,43], and quantum physics [22,25,26], to name just a few. In this work, we will use the kinetic equation describing dilute gas–particle flows as an example application. However, the proposed numerical schemes can easily be extended to treat a wide range of kinetic equations describing other applications.

Gas–particle flows occur in many engineering and natural systems such as fluidized-bed reactors, catalytic crackers, volcanic ash transport in the atmosphere, and helicopter brown-out. Currently, there exist several different approaches for simulating the kinetic equation describing the particle phase and its coupling to the gas phase. In general, all approaches use the same type of flow solver for the gas phase, but they differ in the way in which the kinetic equation is treated:

* Corresponding author. Tel.: +1 515 294 1614; fax: +1 515 294 3262.
E-mail addresses: vvikas@iastate.edu (V. Vikas), zjw@iastate.edu (Z.J. Wang), albertop@iastate.edu (A. Passalacqua), rofox@iastate.edu (R.O. Fox).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.03.038

(i) direct solver that discretizes the velocity phase space of the particle number density function [4,36],
(ii) Lagrangian solver that tracks all the particles individually [6],
(iii) hydrodynamic models with kinetic-theory moment closures [16],
(iv) quadrature method of moments (QMOM) that solves for moments of the particle number density function with quadrature-based closures [14,17,31,37].

In many applications, the direct solution of the kinetic equation is prohibitively expensive due to the high dimensionality of the space of independent variables, while Lagrangian solvers are computationally very expensive, since the number of particles to be tracked is very large. On the other hand, hydrodynamic models are developed assuming that the Knudsen number of the particle phase is very small, which is equivalent to assuming a Maxwellian (or nearly Maxwellian) equilibrium particle velocity distribution. This, however, is not correct in relatively dilute gas–particle flows, where the Knudsen number can be high, the collision frequency is small and phenomena like particle trajectory crossing can occur. In particular, Desjardins et al. [14] showed that the assumption that a gas–particle flow can be described accounting for only the mean momentum of the particle phase leads to incorrect predictions of all the velocity moments, including the particle number density, showing the need of using a multi-velocity method, in order to correctly capture the physics of the flow. Similar observations were made for moderately collisional gas–particle flows by Sakiz and Simonin [44].

QMOM for gas–particle flow [17–19] is based on the idea of tracking a set of velocity moments of arbitrarily high order, providing closures to the source terms and the moment spatial fluxes in the moment transport equations by means of a quadrature approximation of the number density function. The key step of the approach is an inversion algorithm that allows one to uniquely determine a set of weights and abscissas from the set of transported moments. The condition for the inversion algorithm to be applied is that the set of moments is realizable, meaning it actually corresponds to a velocity distribution. This condition is not generally satisfied by the traditional finite-volume methods used in computational fluid dynamics.

Desjardins et al. [14] recently showed that realizability is guaranteed only with the 1st-order finite-volume scheme. The use of any other high-order finite-volume scheme may lead to non-realizable moments, thereby resulting in negative weights. Weights are representative of particle density and hence cannot be negative. This limitation in turn leads to the use of a highly refined mesh for computation as a 1st-order finite-volume scheme produces highly diffused solutions on a coarse mesh. For this reason, improved finite-volume schemes are sought that could provide less-diffused solutions and simultaneously guarantee the realizability of moments.

In the present work, a generalized idea for developing improved finite-volume schemes for quadrature-based moment methods is presented. The realizability of the new improved finite-volume schemes is guaranteed under suitable realizability criteria. These new schemes give remarkable improvement in the solutions for the class of problems where the velocity abscissas are constant over a range of cells. The present work also shows that the standard Runge–Kutta time-integration schemes do not guarantee realizability. However, the realizability can be guaranteed if strong stability-preserving (SSP) Runge–Kutta schemes [23] are used.

The remainder of this paper is organized as follows. In Section 2, QMOM is reviewed. Section 2 also discusses finite-volume schemes and their realizability properties. Thereafter, in Section 3, new realizable high-order finite-volume schemes are presented. In Section 4, multi-stage time-integration is discussed and realizability properties of the standard RK2 and RK2SSP schemes are presented. Section 5 presents some numerical results including accuracy studies. Conclusions from the present study are summarized in Section 6. Finally, in Appendix A we present an extension of the realizable schemes to velocity-independent density functions.

2. Quadrature method of moments

2.1. Kinetic equation

Dilute gas–particle flows can be modeled by a kinetic equation [9,10,46] of the form:

$$\partial_t f + \mathbf{v}\cdot\partial_{\mathbf{x}} f + \partial_{\mathbf{v}}\cdot(f\mathbf{F}) = \mathbb{C}, \tag{1}$$

where $f(\mathbf{v},\mathbf{x},t)$ is the velocity-based number density function, $\mathbf{v}$ is the particle velocity, $\mathbf{F}$ is the force acting on an individual particle, and $\mathbb{C}$ is the collision term representing the rate of change in the number density function due to particle–particle collisions. The collision term can be described using the Bhatnagar–Gross–Krook (BGK) collision operator [5]:

$$\mathbb{C} = \frac{1}{\tau}(f_{\mathrm{eq}} - f), \tag{2}$$

where τ is the characteristic collision time, and f_{eq} is the Maxwellian equilibrium number density function given by

$$f_{\mathrm{eq}}(\mathbf{v}) = \frac{M^0}{\sqrt{(2\pi\sigma_{\mathrm{eq}})^3}}\exp\left(-\frac{|\mathbf{v}-\mathbf{U}_{\mathrm{p}}|^2}{2\sigma_{\mathrm{eq}}}\right), \tag{3}$$

in which $\mathbf{U}_p$ is the mean particle velocity, σ_{eq} is the equilibrium variance and $M^0 = \int f d\mathbf{v}$ is the particle number density. In gas–particle flows, the force term is given by the sum of the gravitational contribution ($\mathbf{F}_g$) and the drag term ($\mathbf{F}_d$) exerted from the fluid on the particles:

$$\mathbf{F} = \mathbf{F}_g + \mathbf{F}_d. \tag{4}$$

For dilute gas–particle flows, the drag force on a particle can be approximated by

$$\mathbf{F}_d = \frac{3 m_p \rho_g}{4 \rho_p d_p} C_d |\mathbf{U}_r| \mathbf{U}_r, \tag{5}$$

where $\mathbf{U}_r = \mathbf{U}_g - \mathbf{U}_p$ is the relative velocity between two phases, $\mathbf{U}_g$ is the gas velocity, $\mathbf{U}_p$ is the particle phase local mean velocity, ρ_g and ρ_p are gas and particle densities, respectively, and d_p is the particle diameter. The drag coefficient C_d is given by the Schiller–Naumann correlation [45]:

$$C_d = \frac{24}{Re_p}\left(1 + 0.15 Re_p^{0.687}\right), \tag{6}$$

in which $Re_p = \rho_g d_p |\mathbf{U}_g - \mathbf{U}_p| / \mu_g$, μ_g being the dynamic viscosity of gas phase.

2.2. Moment transport equations

In the quadrature-based moment method of Fox [17–19], a set of moments of the number density function f is transported and its evolution in space and time is tracked. Each element of the moment set is defined through integrals of the number density function. For the first few moments the defining integrals are:

$$M^0 = \int f d\mathbf{v}, M_i^1 = \int v_i f \, d\mathbf{v}, M_{ij}^2 = \int v_i v_j f \, d\mathbf{v}, M_{ijk}^3 = \int v_i v_j v_k f \, d\mathbf{v}. \tag{7}$$

In these equations, the superscript of M represents the order of corresponding moment. Moment transport equations are obtained by applying the definition of moments to (1). The transport equations for moments in (7) can be written as:

$$\begin{aligned}
&\frac{\partial M^0}{\partial t} + \frac{\partial M_i^1}{\partial x_i} = 0,\\
&\frac{\partial M_i^1}{\partial t} + \frac{\partial M_{ij}^2}{\partial x_j} = g_i M^0 + D_i^1,\\
&\frac{\partial M_{ij}^2}{\partial t} + \frac{\partial M_{ijk}^3}{\partial x_k} = g_i M_j^1 + g_j M_i^1 + C_{ij}^2 + D_{ij}^2,\\
&\frac{\partial M_{ijk}^3}{\partial t} + \frac{\partial M_{ijkl}^4}{\partial x_l} = g_i M_{jk}^2 + g_j M_{ik}^2 + g_k M_{ij}^2 + C_{ijk}^3 + D_{ijk}^3.
\end{aligned} \tag{8}$$

In (8), g_i, g_j, g_k are the components of gravity, D_i^1, D_{ij}^2, D_{ijk}^3 are due to the drag force and C_{ij}^2, C_{ijk}^3 are due to collisions.

2.3. Quadrature-based closures

Using the BGK model, the collision terms in (8) can be closed. However, the set of transport equations in (8) is still unclosed because of the spatial flux and drag terms. Each equation contains the spatial fluxes of the moments of order immediately higher. In quadrature-based moment methods, quadrature formulas are used to provide closures to these terms in the moment transport equations, by introducing a set of weights and abscissas. The number density function f is written in terms of the quadrature weights (n_α) and abscissas ($\mathbf{U}_\alpha$) using a Dirac delta representation:

$$f(\mathbf{v}) = \sum_{\alpha=1}^{\beta} n_\alpha \delta(\mathbf{v} - \mathbf{U}_\alpha). \tag{9}$$

The method based on (9) is called β-node quadrature method. The moments can be computed as a function of quadrature weights and abscissas by using the above definition of f in (7):

$$M^0 = \sum_{\alpha=1}^{\beta} n_\alpha, \quad M_i^1 = \sum_{\alpha=1}^{\beta} n_\alpha U_{i\alpha}, \quad M_{ij}^2 = \sum_{\alpha=1}^{\beta} n_\alpha U_{i\alpha} U_{j\alpha}, \quad M_{ijk}^3 = \sum_{\alpha=1}^{\beta} n_\alpha U_{i\alpha} U_{j\alpha} U_{k\alpha}. \tag{10}$$

The source terms due to drag and gravity are computed as:

$$\begin{aligned}
D_i^1 &= \sum_{\alpha=1}^{\beta} \frac{n_\alpha}{m_p} F_{i\alpha},\\
D_{ij}^2 &= \sum_{\alpha=1}^{\beta} \frac{n_\alpha}{m_p} \left(F_{i\alpha} U_{j\alpha} + F_{j\alpha} U_{i\alpha}\right),\\
D_{ijk}^3 &= \sum_{\alpha=1}^{\beta} \frac{n_\alpha}{m_p} \left(F_{i\alpha} U_{j\alpha} U_{k\alpha} + F_{j\alpha} U_{k\alpha} U_{i\alpha} + F_{k\alpha} U_{i\alpha} U_{j\alpha}\right).
\end{aligned} \tag{11}$$

The details of the computation of the force terms $F_{i\alpha}$, $F_{j\alpha}$ and $F_{k\alpha}$, can be found in [17].

In order to ensure that the moments remain realizable and that the discretized fluxes are hyperbolic, the spatial flux terms are closed according to their kinetic description [13,14,38,41]. Each moment involved in the spatial derivative is decomposed into two contributions, as shown in (12) for the zero-order moments:

$$M_i^1 = \int_{-\infty}^{0} v_i \left(\int f dv_j dv_k \right) dv_i + \int_{0}^{+\infty} v_i \left(\int f dv_j dv_k \right) dv_i. \tag{12}$$

In a 1-D case, the left integral will account for particles going from left to right and the right integral will account for particles going from right to left at the face. Using (9), (12) can be written as:

$$M_i^1 = \sum_{\alpha=1}^{\beta} n_\alpha \min(0, U_{i\alpha}) + \sum_{\alpha=1}^{\beta} n_\alpha \max(0, U_{i\alpha}). \tag{13}$$

For a 1-D case, the left integral/summation in (12)/(13) is evaluated using the values on left side of the face and right integral/summation is evaluated using the values on right side of the face.

In order to solve the abovementioned moment transport equations, boundary conditions are needed. These boundary conditions can be specified either in terms of moments or in terms of the weights and abscissas. The latter approach is more convenient and is followed in the present work. In this work, three types of boundary conditions are used: Dirichlet, periodic and wall-reflective. At a Dirichlet boundary, the weights and abscissas are specified. Periodic boundary conditions copy the weights and abscissas from the outgoing periodic boundary cell to the corresponding incoming periodic boundary cell. The boundary conditions at the walls are set so that a particle that collides with the wall is specularly reflected. This condition corresponds to changing the sign of the velocity component of the particle along the direction perpendicular to the wall. The implementation of this boundary condition in the quadrature-based algorithm is done by changing the sign of the abscissas in the appropriate direction [37]. If $i = 0$ indicates the position of the wall, perpendicular to the second direction of reference frame, and $i = 1$ indicates the neighboring computational cell, the boundary condition can be written as:

$$\begin{pmatrix} n_\alpha \\ U_\alpha \\ V_\alpha \\ W_\alpha \end{pmatrix}_{i=0} = \begin{pmatrix} n_\alpha / e_w \\ U_\alpha \\ -e_w V_\alpha \\ W_\alpha \end{pmatrix}_{i=1}, \tag{14}$$

where e_w is the particle–wall restitution coefficient. All the boundary conditions are applied using a ghost-cell approach.

2.4. Finite-volume method

The moment transport equations in (8) contain convection, drag and collision terms. The three terms are treated separately using an *operator-splitting* technique. The solution algorithm involving all the terms is given later. The collision and the force terms do not create non-realizability problems. Hence for all the analysis, these terms are dropped. For simplicity, a one-dimensional case with two quadrature nodes is discussed here. A general 3-D case is presented in a later section. For the 1-D case, the set of moment transport equations after dropping collision and drag terms can be written as:

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{H}(\mathbf{W})}{\partial x} = 0, \tag{15}$$

where

$$\mathbf{W} = [M^0 \; M^1 \; M^2 \; M^3]^T \quad \text{and} \quad \mathbf{H}(\mathbf{W}) = [M^1 \; M^2 \; M^3 \; M^4]^T. \tag{16}$$

For a 2-node quadrature there are two weights (n_1, n_2) and two abscissas (U_1, U_2). Let the set of weights and abscissas be denoted as $\mathbf{N} = [n_1 \; n_2 \; U_1 \; U_2]^T$. The first four moments can be written in terms of these weights and abscissas as:

$$\begin{aligned} M^0 &= n_1 (U_1)^0 + n_2 (U_2)^0, \\ M^1 &= n_1 (U_1)^1 + n_2 (U_2)^1, \\ M^2 &= n_1 (U_1)^2 + n_2 (U_2)^2, \\ M^3 &= n_1 (U_1)^3 + n_2 (U_2)^3. \end{aligned} \tag{17}$$

In the first two equations for M^0 and M^1, powers of U_1 and U_2 are redundant.

The conserved moments and moment fluxes in (16) can be written in terms of the number density function:

$$\mathbf{W} = \int \mathbf{K}(v) f(v) \, dv, \mathbf{H}(\mathbf{W}) = \int v \mathbf{K}(v) f(v) \, dv, \tag{18}$$

where

$$\mathbf{K}(v) = [1 \; v \; v^2 \; v^3]^T. \tag{19}$$

The moments in (15) can be advanced in time using a finite-volume scheme. If a single-stage explicit time-integration scheme is used, the updated set of moments can be written as:

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \frac{\Delta t}{\Delta x}\left[\mathbf{G}\left(\mathbf{W}_{i+1/2,l}^n, \mathbf{W}_{i+1/2,r}^n\right) - \mathbf{G}\left(\mathbf{W}_{i-1/2,l}^n, \mathbf{W}_{i-1/2,r}^n\right)\right], \tag{20}$$

where superscripts n and $n+1$ denote time levels, $\mathbf{G}$ is the numerical flux function evaluated at cell interfaces, and l and r denote the left and right states at the interfaces respectively. Henceforth, the variables with subscript i will denote the cell-averaged values and the ones with $(i+1/2)$ or $(i-1/2)$ as subscript will denote the reconstructed values at the interfaces. $\mathbf{G}$ is defined as:

$$\mathbf{G}(\mathbf{W}_l, \mathbf{W}_r) = \int v^+ \mathbf{K} f_l \, dv + \int v^- \mathbf{K} f_r \, dv, \tag{21}$$

where

$$v^+ = \frac{1}{2}(v + |v|) \quad \text{and} \quad v^- = \frac{1}{2}(v - |v|). \tag{22}$$

This corresponds to a splitting between particles going from left to right (first term) and particles going from right to left (second term). Inserting the expression for f in (21) yields:

$$\mathbf{G}(\mathbf{W}_l, \mathbf{W}_r) = \mathbf{H}^+(\mathbf{W}_l) + \mathbf{H}^-(\mathbf{W}_r), \tag{23}$$

with

$$\begin{aligned}
\mathbf{H}^+(\mathbf{W}_l) &= n_{1l}\max(U_{1l}, 0)\begin{pmatrix} 1 \\ U_{1l} \\ U_{1l}^2 \\ U_{1l}^3 \end{pmatrix} + n_{2l}\max(U_{2l}, 0)\begin{pmatrix} 1 \\ U_{2l} \\ U_{2l}^2 \\ U_{2l}^3 \end{pmatrix}, \\
\mathbf{H}^-(\mathbf{W}_r) &= n_{1r}\min(U_{1r}, 0)\begin{pmatrix} 1 \\ U_{1r} \\ U_{1r}^2 \\ U_{1r}^3 \end{pmatrix} + n_{2r}\min(U_{2r}, 0)\begin{pmatrix} 1 \\ U_{2r} \\ U_{2r}^2 \\ U_{2r}^3 \end{pmatrix}.
\end{aligned} \tag{24}$$

Before discussing the 1st-, 2nd- and 3rd-order finite-volume schemes, it is worth noticing that, in order to calculate the fluxes at the interface, variables need to be reconstructed at the faces of each cell. However, there is an ambiguity in the choice of variables to be chosen for reconstruction. Two choices are possible: (i) reconstructing the moments ($\mathbf{H}(\mathbf{W}) = [M^1\ M^2\ M^3\ M^4]^T$) and (ii) reconstructing the weights and abscissas ($\mathbf{N} = [n_1\ n_2\ U_1\ U_2]^T$). For the 1st-order finite-volume scheme both choices are equivalent. However, for second or any high-order finite-volume schemes, these two choices are in general different. The former approach often leads to non-realizable moments, especially in the problems involving discontinuous velocity fields. For this reason, the second approach is adopted in this work. In the sections below, any reconstruction essentially refers to reconstruction of the weights and abscissas. For example, $\mathbf{W}_{i+1/2,l}$ refers to moments at the left side of interface $i+1/2$ calculated using the reconstructed values of the weights and abscissas.

2.4.1. First-order finite-volume scheme

The 1st-order finite-volume scheme for solving moment transport equations uses a piecewise constant approximation and is described in [14]. The weights and abscissas are assumed to be constant over a cell:

$$\mathbf{N}_{i-1/2,r}^n = \mathbf{N}_i^n, \quad \mathbf{N}_{i+1/2,l}^n = \mathbf{N}_i^n. \tag{25}$$

2.4.2. Second-order finite-volume scheme

In the 2nd-order finite-volume scheme, a piecewise linear reconstruction for the weights and abscissas is used. The piecewise linear reconstruction is obtained using a minmod slope limiter [27]. For the ith cell, this can be written as:

$$\begin{aligned}
\mathbf{N}_{i-1/2,r}^n &= \mathbf{N}_i^n - \tfrac{\Delta x}{2}\partial\mathbf{N}_i, \\
\mathbf{N}_{i+1/2,l}^n &= \mathbf{N}_i^n + \tfrac{\Delta x}{2}\partial\mathbf{N}_i,
\end{aligned} \tag{26}$$

where

$$\partial\mathbf{N}_i = \text{minmod}\left(\frac{\mathbf{N}_i^n - \mathbf{N}_{i-1}^n}{\Delta x}, \frac{\mathbf{N}_{i+1}^n - \mathbf{N}_i^n}{\Delta x}\right). \tag{27}$$

The minmod function is defined as:

$$\text{minmod}(x, y) = \text{sign}(x)\left(\frac{1 + \text{sign}(xy)}{2}\right)\min(|x|, |y|). \tag{28}$$

2.4.3. Third-order finite-volume scheme

In the 3rd-order finite-volume scheme, a piecewise parabolic reconstruction for the weights and abscissas is used. For the ith cell, the reconstructed weights and abscissas can be written as:

$$\begin{aligned}\mathbf{N}^n_{i-1/2,r} &= \mathbf{N}^n_i - \left[\tfrac{1}{3}\left(\mathbf{N}^n_i - \mathbf{N}^n_{i-1}\right) + \tfrac{1}{6}\left(\mathbf{N}^n_{i+1} - \mathbf{N}^n_i\right)\right],\\ \mathbf{N}^n_{i+1/2,l} &= \mathbf{N}^n_i + \left[\tfrac{1}{6}\left(\mathbf{N}^n_i - \mathbf{N}^n_{i-1}\right) + \tfrac{1}{3}\left(\mathbf{N}^n_{i+1} - \mathbf{N}^n_i\right)\right].\end{aligned} \tag{29}$$

The 3rd-order MUSCL reconstruction without a limiter is the same as (29). However, in the case of a discontinuous solution a limiter is essential. A limited version of 3rd-order MUSCL can be found in [27].

2.5. Solution algorithm

The finite-volume schemes presented in the above section are used to solve for the spatial transport part of the moment transport equations. However, the complete moment transport equations have collision and force terms as well. As mentioned earlier, these two terms are treated using an *operator-splitting* technique. A detailed solution algorithm involving all the terms can be found in [14,18,37]. Here a brief overview of the steps involved in the solution procedure is presented, assuming a single-stage explicit time-integration:

1. Initialize weights and abscissas in the domain.
2. Apply boundary conditions to weights and abscissas.
3. Compute moments using weights and abscissas.
4. Compute time-step size Δt.
5. Reconstruct weights and abscissas at cell faces.
6. Compute spatial flux terms at cell faces.
7. Advance moments by Δt due to spatial flux terms using a finite-volume approach.
8. Compute weights and abscissas from moments using the moment-inversion algorithm.
9. Advance weights by Δt due to force terms (drag and gravity).
10. Compute moments using weights and abscissas.
11. Advance moments by Δt due to collision terms.
12. Compute weights and abscissas from moments using the moment-inversion algorithm.
13. Apply boundary conditions to weights and abscissas.
14. Repeat steps (4) through (l3) at each time step.

2.6. Non-realizability problem

At each time step, the weights and abscissas need to be recovered from the moments. The moment-inversion algorithm computes the set of weights and abscissas from the corresponding set of moments by solving a set of nonlinear equations. In the moment-inversion algorithm M^0, M^1, M^2, M^3 are known and n_1, n_2, U_1, U_2 are computed, by solving (17) in reverse direction using the product-difference (PD) algorithm [32,53,40]. However, the set of moments cannot be constituted by arbitrary values of each moment, but they have to conform to the definition of the *non-negative* number density function. The application of the PD algorithm to a set of realizable moments leads to a set of weights and abscissas that satisfy the properties of Gaussian quadrature. In particular the weights are always positive. During the development of the numerical schemes, which is the object of this work, a set of weights and abscissas will be said to represent realizable moments if weights are positive and abscissas lie in the interior of the support of f. Because of the non-linearity of the inversion problem, it is extremely difficult to determine in advance whether a given set of moments is realizable. However, Desjardins et al. [14] described how any finite-volume scheme that could guarantee non-negativity of the effective number density function will always keep the moments in realizable space.

The updated set of moments can be written as:

$$\mathbf{W}^{n+1}_i = \int \mathbf{K} h \, dv, \tag{30}$$

where

$$h = f^n_i - \lambda\left(v^+ f^n_{i+1/2,l} + v^- f^n_{i+1/2,r} - v^+ f^n_{i-1/2,l} - v^- f^n_{i-1/2,r}\right) = f^n_i - \lambda v^+ f^n_{i+1/2,l} - \lambda v^- f^n_{i+1/2,r} + \lambda v^+ f^n_{i-1/2,l} + \lambda v^- f^n_{i-1/2,r} \tag{31}$$

in which $\lambda = \Delta t/\Delta x$. In (31), h is the effective number density function and has different forms for different finite-volume schemes as the interface values are reconstructed in different ways. Desjardins et al. [14] stated that any finite-volume scheme that guarantees the non-negativity of h for all v, is realizable. Using this proposition, they derived the realizability criterion for the 1st-order finite-volume scheme. In the sections below, the realizability of the 1-D version of 1st-, 2nd- and high-order finite-volume schemes is discussed.

2.6.1. First-order finite-volume scheme

For the 1st-order finite-volume scheme (25) implies

$$f^n_{i-1/2,r} = f^n_i, f^n_{i+1/2,l} = f^n_i. \tag{32}$$

Using this in (31), the effective number density function for the 1st-order finite-volume scheme can be written as:

$$h = (1 - \lambda|v|)f^n_i + \lambda v^+ f^n_{i-1} - \lambda v^- f^n_{i+1}. \tag{33}$$

As the moments at time level n are assumed to be realizable, the non-negativity of the number density function at time level n is guaranteed, i.e., $f^n_i \geqslant 0$, $f^n_{i-1} \geqslant 0$ and $f^n_{i+1} \geqslant 0$ for all v. Also, $v^+ \geqslant 0$ and $v^- \leqslant 0$. Hence, the non-negativity of h will be guaranteed if $(1 - \lambda|v|) \geqslant 0$. When written in terms of the abscissas, this condition becomes:

$$\lambda \leqslant \frac{1}{\max\left(|U^n_{i1}|, |U^n_{i2}|\right)}. \tag{34}$$

This is the realizability criterion for the 1st-order finite-volume scheme [14].

2.6.2. Second- and high-order finite-volume schemes

For the 2nd-order and, in general, any high-order finite-volume scheme, the reconstructed values at the cell interfaces are different from the cell-averaged values. The equality of reconstructed values and cell-averaged values is a *special* property that holds only for the 1st-order finite-volume scheme. For 2nd and high-order finite-volume schemes, the effective number density function h can be obtained from (31) by putting in the reconstructed values. An interesting thing to notice in (31) is that on the right-hand side, out of the five terms, only three are non-negative. The second and fifth terms are always non-positive. As was stated in [14], realizability can only be guaranteed if the effective number density function is non-negative for all velocities. Clearly this does not hold, in general, for 2nd and high-order finite-volume schemes. Hence, realizability cannot be guaranteed for any finite-volume scheme other than 1st-order.

3. Improved realizable finite-volume schemes

3.1. Basic idea

As was stated earlier, the two non-positive terms on the right-hand side of (31) might lead to non-realizability problems. The non-negative terms present a physical view of the number density function. An important thing to notice is that, despite the presence of the two non-positive terms, the 1st-order finite-volume scheme is still realizable under the restriction of a realizability criterion. This is possible because in the 1st-order finite-volume scheme, the interface values are the same as the cell-averaged values $\left(f^n_i = f^n_{i+1/2,l} = f^n_{i-1/2,r}\right)$, thereby allowing a grouping of the first term with the two non-positive terms on right-hand side of (31). This idea of grouping the terms is essential to the realizability of the 1st-order finite-volume scheme and it also forms the basis of the development of new improved realizable finite-volume schemes.

For high-order finite-volume schemes, in general:

$$\mathbf{N}^n_i \neq \mathbf{N}^n_{i+1/2,l} \neq \mathbf{N}^n_{i-1/2,r}, \tag{35}$$

i.e.

$$n^n_{i,\alpha} \neq n^n_{i+1/2,\alpha,l} \neq n^n_{i-1/2,\alpha,r} \quad \text{and} \quad U^n_{i,\alpha} \neq U^n_{i+1/2,\alpha,l} \neq U^n_{i-1/2,\alpha,r}. \tag{36}$$

Consider a special reconstruction where

$$n^n_{i,\alpha} \neq n^n_{i+1/2,\alpha,l} \neq n^n_{i-1/2,\alpha,r} \quad \text{and} \quad U^n_{i,\alpha} = U^n_{i+1/2,\alpha,l} = U^n_{i-1/2,\alpha,r}. \tag{37}$$

For this special reconstruction, realizability can always be guaranteed with a suitable constraint on the time-step size. This is the subject of the following theorem.

Theorem 1. *Let $\beta, p \in \mathbb{N}$ and $\alpha \in \{1,2,\ldots,\beta\}$. Also let the cell-averaged and reconstructed values of the weights satisfy $n^n_{i,\alpha} > 0$ and $n^n_{i+1/2,\alpha,l}$, $n^n_{i-1/2,\alpha,r} \geqslant 0$ $\forall\alpha$. If a finite-volume scheme using a single-stage Euler explicit time-integration scheme is devised that uses a pth-order reconstruction for weights and 1st-order reconstruction for abscissas, the non-negativity of the effective number density function (31) in the ith cell can always be guaranteed under an explicit constraint on time-step size* $(\Delta t \in \mathbb{R}^+)$.

Proof. Using (31), the effective number density function, regardless of the finite-volume scheme used, can be written as:

$$h = f^n_i - \lambda v^+ f^n_{i+1/2,l} + \lambda v^- f^n_{i-1/2,r} + \xi^+. \tag{38}$$

In the above expression, the first non-negative term and the two non-positive terms have been represented explicitly. The other two terms are always non-negative and have been grouped under ξ^+. For a β-node quadrature, using (9), the expression for h becomes:

$$h = \sum_{\alpha=1}^{\beta} \left[n_{i,\alpha}^n \delta\left(v - U_{i,\alpha}^n\right) - \lambda v^+ n_{i+1/2,\alpha,l}^n \delta\left(v - U_{i+1/2,\alpha,l}^n\right) + \lambda v^- n_{i-1/2,\alpha,r}^n \delta\left(v - U_{i-1/2,\alpha,r}^n\right) \right] + \xi^+. \tag{39}$$

If a 1st-order reconstruction is used for the abscissas, then the interface values of the abscissas will be the same as the cell-averaged values:

$$U_{i,\alpha}^n = U_{i+1/2,\alpha,l}^n = U_{i-1/2,\alpha,r}^n. \tag{40}$$

Putting this in (39), the effective number density function becomes:

$$h = \sum_{\alpha=1}^{\beta} \left\{ n_{i,\alpha}^n - \lambda v^+ n_{i+1/2,\alpha,l}^n + \lambda v^- n_{i-1/2,\alpha,r}^n \right\} \delta\left(v - U_{i,\alpha}^n\right) + \xi^+. \tag{41}$$

For $\Delta t \in \mathbb{R}^+$ satisfying the condition:

$$\lambda = \min_{\alpha \in \{1,2,\ldots,\beta\}} \left(\frac{n_{i,\alpha}^n}{n_{i+1/2,\alpha,l}^n \max\left(U_{i,\alpha}^n, 0\right) - n_{i-1/2,\alpha,r}^n \min\left(U_{i,\alpha}^n, 0\right)} \right), \tag{42}$$

h is non-negative for all v. This concludes the proof. □

Clearly, Theorem 1 guarantees the realizability of the special reconstruction in (37), for all cases except for the one where $n_{i,\alpha}^n = 0$. However this case turns out to be trivial if a minmod limiter is used to limit the reconstructed values. A minmod limiter guarantees that whenever $n_{i,\alpha}^n = 0$, then $n_{i+1/2,\alpha,l}^n = n_{i-1/2,\alpha,r}^n = 0$, thereby automatically dropping the two non-positive terms in (39). For $p = 1$, (42) reduces to the same realizability criterion as in (34):

$$\lambda = \min_{\alpha \in \{1,2,\ldots,\beta\}} \left(\frac{1}{\left|U_{i,\alpha}^n\right|} \right). \tag{43}$$

This new reconstruction uses a high-order reconstruction for the weights, but a 1st-order reconstruction for the abscissas. To remove the ambiguity, it is worth clarifying that all the finite-volume schemes discussed earlier, in which the same order of reconstruction was used for the weights and abscissas will be termed *standard* finite-volume schemes. For example, a *standard* 2nd-order finite-volume scheme is the one discussed in Section 2.4.2, where a 2nd-order reconstruction is used for both the weights and abscissas. More generally, a *standard* pth-order finite-volume scheme uses a pth-order reconstruction for both the weights and abscissas. It is worth reiterating that the realizability of the *standard* pth-order finite-volume scheme is not guaranteed. Corresponding to the *standard* pth-order finite-volume scheme another scheme can be developed based on the special reconstruction discussed above. This new scheme will use pth-order reconstruction for the weights and 1st-order reconstruction for the abscissas. Realizability of this new scheme is guaranteed by the constraint, also known as realizability criterion, in (42). Because a 1st-order reconstruction is used for the abscissas, the new scheme is less accurate than the *standard* pth-order finite-volume scheme and henceforth will be termed a *quasi*-pth-order finite-volume scheme. The *standard* 1st-order finite-volume scheme and the *quasi*-1st-order finite-volume scheme are the same and will be simply referred to as the 1st-order finite-volume scheme.

Two important facts about the new schemes are: (i) they are better than 1st-order finite-volume schemes as far as accuracy is concerned and (ii) realizability is guaranteed under the realizability criterion that has an explicit form. This marks a significant improvement in solution methods for quadrature-based moment methods. Over the years, many high-order finite-volume schemes have been developed [1,2,12,28,49–52] for convection-dominated problems in the field of fluid dynamics. However, quadrature-based moment methods have not benefited from these high-order schemes because of the non-realizability limitation. However, with the new approach, all the already existing knowledge about high-order finite-volume schemes can now be utilized for solutions using quadrature-based moment methods.

In general, the new *quasi*-pth-order realizable finite-volume scheme will be less accurate compared to the *standard* pth-order finite-volume scheme but for the problems where the velocities are constant over a range of cells, the difference in accuracy will be negligible. This fact is further demonstrated in Section 5. Next the realizability criterion for different dimensions is presented, and we explain the way in which it is applied. In Appendix A we show how *quasi*-pth-order finite-volume schemes can also be written for the advection of *velocity-independent* density functions.

3.2. Realizability criterion for 1-D cases

The relation given in (42) is the realizability criterion for 1-D problems. It can also be written as:

$$\left\{ n_{i,\alpha}^n - \lambda \max(U_{i\alpha}, 0) n_{i+1/2,\alpha,l}^n + \lambda \min(U_{i\alpha}, 0) n_{i-1/2,\alpha,r}^n \right\} \geqslant 0 \quad \forall \alpha \in \{1, 2, \ldots, \beta\}. \tag{44}$$

This simple realizability criterion can be used for the calculation of λ and hence Δt for 1-D cases with β-node quadrature.

3.3. Realizability criterion for 2-D/3-D cases

In (44), $\left[\max(U_{i\alpha},0)n^n_{i+1/2,\alpha,l}-\min(U_{i\alpha},0)n^n_{i-1/2,\alpha,r}\right]$ is the total outgoing flux from the ith cell for the αth weight. Hence, (44) can also be written as:

$$\left\{n^n_{i,\alpha}-\lambda\sum\text{Outgoing\_Flux}^n_{i,\alpha}\right\}\geqslant 0\quad\forall\alpha\in\{1,2,\ldots,\beta\},\tag{45}$$

where the summation is over the faces of the ith cell. This form of the realizability criterion can be used for 2-D and 3-D cases. Although (45) was obtained using an analogy from (44), it can be derived directly from the expression for finite-volume scheme in 2-D or 3-D.

Consider a 3-D case. Let Ω_i and $\partial\Omega_i$ denote the ith cell and its boundary respectively. Also let $e\in\partial\Omega_i$ be a face of the ith cell, A_e be its area and $\hat{\mathbf{n}}_e=[n_{e,x}\ n_{e,y}\ n_{e,z}]$ be the outward unit normal at this face. The finite-volume expression for the single-stage Euler explicit time-integration, analogous to (20) can be written as:

$$\mathbf{W}^{n+1}_i=\mathbf{W}^n_i-\frac{\Delta t}{|\Omega_i|}\sum_{e\in\partial\Omega_i}\left\{\mathbf{G}\left(\mathbf{W}^n_{e,l},\mathbf{W}^n_{e,r},\hat{\mathbf{n}}_e\right)A_e\right\},\tag{46}$$

where $|\Omega_i|$ denotes the volume of the ith cell and $\mathbf{W}^n_{e,l}$ and $\mathbf{W}^n_{e,r}$ represent the reconstructed value of $\mathbf{W}$ on the left and right sides of face e. The outward normal vector $\hat{\mathbf{n}}_e$ defined above, points from the left side to the right side of face e. The form of $\mathbf{W}$ analogous to the one in (18) is:

$$\mathbf{W}=\int\mathbf{K}(u,v,w)f(u,v,w)\,du\,dv\,dw.\tag{47}$$

The numerical flux function can be written as:

$$\mathbf{G}\left(\mathbf{W}^n_{e,l},\mathbf{W}^n_{e,r},\hat{\mathbf{n}}_e\right)=\int v^+_n\mathbf{K}f_{e,l}\,du\,dv\,dw+\int v^-_n\mathbf{K}f_{e,r}\,du\,dv\,dw,\tag{48}$$

where

$$\begin{aligned}v^+_n&=\max(n_{e,x}u+n_{e,y}v+n_{e,z}v,0),\\ v^-_n&=\min(n_{e,x}u+n_{e,y}v+n_{e,z}v,0).\end{aligned}\tag{49}$$

Substituting (47) and (48) in (46), the effective number density can be written as:

$$h=f^n_i-\lambda\sum_{e\in\partial\Omega_i}\left(v^+_nf_{e,l}A_e+v^-_nf_{e,r}A_e\right).\tag{50}$$

In (50), $\lambda=\Delta t/|\Omega_i|$. For β-node quadrature, f has the same form as in (9). Using (9) and the special reconstruction $\mathbf{U}^n_{e,\alpha,l}=\mathbf{U}^n_{e,\alpha,r}=\mathbf{U}^n_{i,\alpha}$, h can be further written as:

$$h=\sum_{\alpha=1}^{\beta}\left\{n^n_{i,\alpha}-\lambda\sum_{e\in\partial\Omega_i}\left(v^+_nn^n_{e,\alpha,l}A_e+v^-_nn^n_{e,\alpha,r}A_e\right)\right\}\delta\left(\mathbf{v}-\mathbf{U}^n_{i,\alpha}\right)=\sum_{\alpha=1}^{\beta}\left\{n^n_{i,\alpha}-\lambda\sum_{e\in\partial\Omega_i}\left(v^+_nn^n_{e,\alpha,l}A_e\right)\right\}\delta\left(\mathbf{v}-\mathbf{U}^n_{i,\alpha}\right)+\xi^+,\tag{51}$$

where again all the non-negative terms except for the first one have been grouped under ξ^+. Non-negativity of h can be guaranteed if:

$$\left\{n^n_{i,\alpha}-\lambda\sum_{e\in\partial I}\left(v^+_nn^n_{e,\alpha,l}A_e\right)\right\}\geqslant 0\quad\forall\alpha\in\{1,2,\ldots,\beta\}.\tag{52}$$

This is exactly the same as (45). It is worth reiterating that the summation in (45) and (52) is for the outgoing fluxes only. Hence, for calculating these fluxes only the reconstructed weights on the interior sides (side towards the ith cell) of faces should be used, the ones on the opposite side should be set to zero, i.e., the flux coming in from neighboring cells should not be accounted for. Consider a simple 2-D Cartesian case shown in Fig. 1. For the sake of simplicity, subscript α and superscript n will be dropped and only 1-node quadrature will be demonstrated. The cell in the center, Ω_0, has four neighbors – Ω_1, Ω_2, Ω_3, Ω_4 – and the reconstructed values of the weights on inner sides of the corresponding faces are n_{01}, n_{02}, n_{03}, n_{04}, respectively. The cell-averaged weight for $cell_0$ is n_0 and the corresponding X-direction and Y-direction abscissas are U_0 and V_0, respectively. The realizability condition for Ω_0 can be written as:

$$\{n_0-\lambda[n_{01}A_1\max(U_0,0)+n_{02}A_1\max(V_0,0)-n_{03}A_1\min(U_0,0)-n_{04}A_4\min(V_0,0)]\}\geqslant 0,\tag{53}$$

where A_1, A_2, A_3, A_4 are the areas of the four faces. In (53), $\lambda=\Delta t/|\Omega_0|$.

All the analysis to this point has been done for a number density function that depends on the velocity. However, many population balance models involve density functions that do not depend on velocity. For such cases, each moment can be advected independently using a 1st-order finite-volume scheme without violating realizability of the moment set. On the other hand, the use of any high-order finite-volume scheme may lead to unphysical or non-realizable moment sets

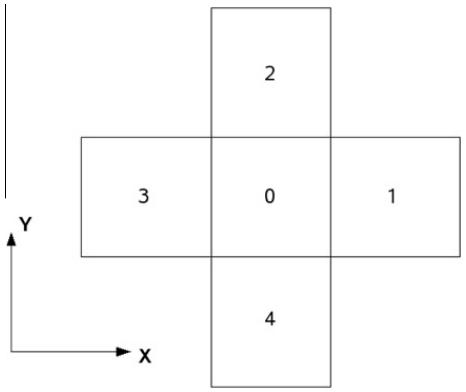

Fig. 1. Cells with faces aligned along Cartesian axes.

[33,34,39,56]. As demonstrated in Appendix A, this problem can be avoided if QMOM is used with the *quasi-p*th-order finite-volume schemes described above.

Before moving onto the next section, here is a brief discussion on the Courant–Friedrichs–Lewy (CFL) criterion vs. the realizability criterion. The realizability criterion is analogous to a CFL criterion. However, their purposes are different. A CFL criterion guarantees the stability of the solution while the realizability criterion guarantees the physical nature of the solution. An explicit form of the CFL criterion exists only for 1-D cases, which is equivalent to the realizability criterion. For 2-D and 3-D problems, there is no explicit form for the CFL criterion. However, there exists an explicit form for the realizability criterion even for 2-D and 3-D cases as shown above. It also turns out from numerical experiments that the realizability criterion is a more strict constraint on Δt as compared to the CFL criterion. Also the realizability criterion corresponding to a pth-order ($p > 1$) finite-volume scheme is in general stricter than the one for the 1st-order finite-volume scheme.

4. Multi-stage time-integration

For simplicity, all the previous sections employed single-stage Euler explicit time-integration. However, single-stage Euler explicit time-integration is only 1st-order accurate in time. To improve time-accuracy, multi-stage explicit time-integration schemes are used in practice. Nevertheless, not all multi-stage time-integration schemes guarantee realizability. Here, two different time-integration schemes are presented: RK2 and RK2SSP [23]. The former is the standard 2nd-order two-stage Runge–Kutta scheme, while the latter is the 2nd-order two-stage strong-stability-preserving (SSP) Runge–Kutta scheme.

4.1. RK2 scheme

The standard 2nd-order two-stage Runge–Kutta scheme can be written as:

$$\mathbf{W}_i^* = \mathbf{W}_i^n - \frac{\Delta t}{2\Delta x}\left[\mathbf{G}\left(\mathbf{W}_{i+1/2,l}^n, \mathbf{W}_{i+1/2,r}^n\right) - \mathbf{G}\left(\mathbf{W}_{i-1/2,l}^n, \mathbf{W}_{i-1/2,r}^n\right)\right], \tag{54}$$

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \frac{\Delta t}{\Delta x}\left[\mathbf{G}\left(\mathbf{W}_{i+1/2,l}^*, \mathbf{W}_{i+1/2,r}^*\right) - \mathbf{G}\left(\mathbf{W}_{i-1/2,l}^*, \mathbf{W}_{i-1/2,r}^*\right)\right]. \tag{55}$$

4.1.1. First stage

The 1st stage is the same as the single-stage Euler explicit time-integration scheme, the only difference being the factor of (1/2) in $\frac{\Delta t}{2\Delta x}$. If $\lambda = \frac{\Delta t}{2\Delta x}$, realizability can be guaranteed for 1st-order and *quasi-p*th-order finite-volume schemes subject to condition (42).

4.1.2. Second stage

The set of moments for the ith cell after the 2nd stage can be written as:

$$\mathbf{W}_i^{n+1} = \int \mathbf{K} h^{n+1} dv, \tag{56}$$

where

$$\begin{aligned} h_i^{n+1} &= f_i^n - \lambda\left(v^+ f_{i+1/2,l}^* + v^- f_{i+1/2,r}^* - v^+ f_{i-1/2,l}^* - v^- f_{i-1/2,r}^*\right) = f_i^n - \lambda v^+ f_{i+1/2,l}^* - \lambda v^- f_{i+1/2,r}^* + \lambda v^+ f_{i-1/2,l}^* + \lambda v^- f_{i-1/2,r}^* \\ &= f_i^n - \lambda v^+ f_{i+1/2,l}^* + \lambda v^- f_{i-1/2,r}^* + \xi^+, \end{aligned} \tag{57}$$

in which $\lambda = \Delta t/\Delta x$. For the 1st-order and *quasi-p*th-order finite-volume schemes:

$$U^*_{i+1/2,\alpha,l} = U^*_{i-1/2,\alpha,r}, \tag{58}$$

but in general

$$U^n_{i,\alpha} \neq U^*_{i+1/2,\alpha,l} \quad \text{and} \quad U^n_{i,\alpha} \neq U^*_{i-1/2,\alpha,r}. \tag{59}$$

Hence, grouping of terms is not possible and realizability cannot be guaranteed for the 2nd stage.

The RK2 scheme does not guarantee realizability of the moment set.

4.2. RK2SSP scheme

The 2nd-order two-stage strong stability-preserving Runge–Kutta scheme can be written as:

$$\mathbf{W}^*_i = \mathbf{W}^n_i - \frac{\Delta t}{\Delta x}\left[\mathbf{G}\left(\mathbf{W}^n_{i+1/2,l}, \mathbf{W}^n_{i+1/2,r}\right) - \mathbf{G}\left(\mathbf{W}^n_{i-1/2,l}, \mathbf{W}^n_{i-1/2,r}\right)\right], \tag{60}$$

$$\mathbf{W}^{n+1}_i = \frac{1}{2}\left[\mathbf{W}^n_i + \mathbf{W}^*_i - \frac{\Delta t}{\Delta x}\left\{\mathbf{G}\left(\mathbf{W}^*_{i+1/2,l}, \mathbf{W}^*_{i+1/2,r}\right) - \mathbf{G}\left(\mathbf{W}^*_{i-1/2,l}, \mathbf{W}^*_{i-1/2,r}\right)\right\}\right]. \tag{61}$$

4.2.1. First stage

The 1st stage is exactly the same as the single-stage Euler explicit time-integration scheme and realizability can be guaranteed for 1st-order and *quasi-p*th-order finite-volume schemes subject to condition (42).

4.2.2. Second stage

The set of moments for the *i*th cell after the 2nd stage can be written as:

$$\mathbf{W}^{n+1}_i = \int \mathbf{K} h^{n+1}_i \, dv, \tag{62}$$

where

Table 1
L_1 error and order of accuracy of schemes using 1-node quadrature.

Grid size	L_1 error	Order
1st-order scheme		
25	0.504533	–
50	0.347427	0.54
100	0.207609	0.74
200	0.114032	0.86
Standard 2nd-order scheme		
25	0.133869	–
50	0.055688	1.27
100	0.016926	1.72
200	0.004903	1.79
Quasi-2nd-order scheme		
25	0.133869	–
50	0.055688	1.27
100	0.016926	1.72
200	0.004903	1.79
Quasi-3rd-order scheme (without limiter)		
25	0.010411	–
50	0.001318	2.98
100	0.000165	2.99
200	0.000024	2.77
Quasi-3rd-order scheme		
25	0.019774	–
50	0.004261	2.21
100	0.000885	2.27
200	0.000180	2.30

$$h_i^{n+1} = \frac{1}{2}\left[f_i^n + f_i^* - \lambda\left(v^+ f^*_{i+1/2,l} + v^- f^*_{i+1/2,r} - v^+ f^*_{i-1/2,l} - v^- f^*_{i-1/2,r}\right)\right]$$
$$= \frac{1}{2}\left[f_i^n + f_i^* - \lambda v^+ f^*_{i+1/2,l} - \lambda v^- f^*_{i+1/2,r} + \lambda v^+ f^*_{i-1/2,l} + \lambda v^- f^*_{i-1/2,r}\right] = \frac{1}{2}\left[f_i^n + f_i^* - \lambda v^+ f^*_{i+1/2,l} + \lambda v^- f^*_{i-1/2,r}\right] + \xi^+$$
$$= \frac{1}{2}\left[f_i^* - \lambda v^+ f^*_{i+1/2,l} + \lambda v^- f^*_{i-1/2,r}\right] + \psi^+, \tag{63}$$

in which $\lambda = \Delta t/\Delta x$ and $\psi^+ = \xi^+ + \frac{1}{2} f_i^n$. For the 1st-order and *quasi-p*th-order finite-volume schemes, grouping of the first three terms is possible because

$$U^*_{i,\alpha} = U^*_{i+1/2,\alpha,l} = U^*_{i-1/2,\alpha,r}. \tag{64}$$

After grouping:

$$h_i^{n+1} = \sum_{\alpha=1}^{\beta}\left\{n^*_{i,\alpha} - \lambda v^+ n^*_{i+1/2,\alpha,l} + \lambda v^- n^*_{i-1/2,\alpha,r}\right\}\delta\left(v - U^*_{i,\alpha}\right) + \psi^+. \tag{65}$$

Table 2
L_1 error and order of accuracy of schemes using 2-node quadrature.

Grid size	L_1 error	Order
1st-order scheme		
25	0.210375	–
50	0.148604	0.50
100	0.091184	0.70
200	0.051804	0.82
Standard 2nd-order scheme		
25	0.071737	–
50	0.028923	1.31
100	0.011792	1.29
200	0.004584	1.36
Quasi-2nd-order scheme		
25	0.071198	–
50	0.028921	1.30
100	0.011793	1.29
200	0.004584	1.36
Quasi-3rd-order scheme		
25	0.022921	–
50	0.009345	1.29
100	0.002946	1.67
200	0.000990	1.57

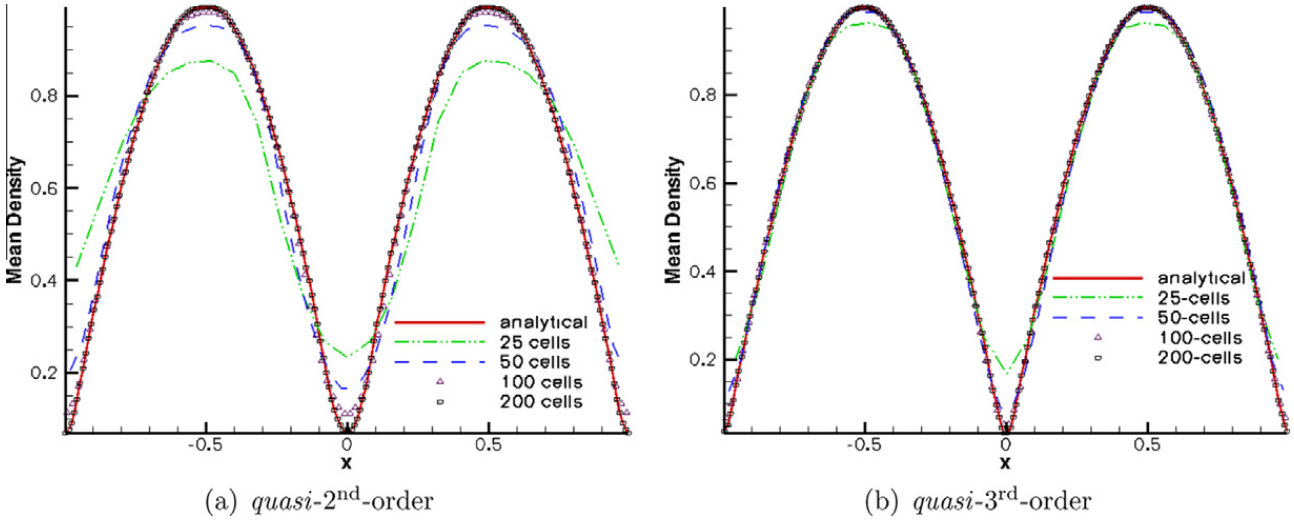

Fig. 2. Grid convergence study for different schemes.

This is exactly the same form as in (42) with the superscript ^{n} replaced by *. Hence, the realizability condition can be guaranteed using (42), by replacing superscript ^{n} with *.

The RK2SSP scheme, combined with *quasi-p*th-order finite-volume schemes, guarantees realizability of the moment set.

4.2.3. Alternate form

Another way to write RK2SPP is:

$$\begin{aligned}
\mathbf{W}_i^{(0)} &= \mathbf{W}_i^n, \\
\mathbf{W}_i^{(1)} &= \mathbf{W}_i^{(0)} - \tfrac{\Delta t}{\Delta x}\left[\mathbf{G}\left(\mathbf{W}_{i+1/2,l}^{(0)}, \mathbf{W}_{i+1/2,r}^{(0)}\right) - \mathbf{G}\left(\mathbf{W}_{i-1/2,l}^{(0)}, \mathbf{W}_{i-1/2,r}^{(0)}\right)\right], \\
\mathbf{W}_i^{(2)} &= \mathbf{W}_i^{(1)} - \tfrac{\Delta t}{\Delta x}\left[\mathbf{G}\left(\mathbf{W}_{i+1/2,l}^{(1)}, \mathbf{W}_{i+1/2,r}^{(1)}\right) - \mathbf{G}\left(\mathbf{W}_{i-1/2,l}^{(1)}, \mathbf{W}_{i-1/2,r}^{(1)}\right)\right], \\
\mathbf{W}_i^{n+1} &= \tfrac{1}{2}\left[\mathbf{W}_i^{(0)} + \mathbf{W}_i^{(2)}\right].
\end{aligned} \tag{66}$$

This form is more amenable to using an operator-splitting technique for the collision and drag terms.

4.3. Calculation of time-step size

The global time-step size (Δt) should satisfy both the CFL and realizability criteria to guarantee stability and physical nature preservation. Usually, the largest value of CFL $\sim$ O(1) that gives a stable solution is used as a CFL criterion for all 1-D/2-D/3-D cases. The definition of the CFL varies, but a general form is:

$$\mathrm{CFL} = \Delta t\left(\frac{\text{cell velocity magnitude}}{\text{cell length scale}}\right). \tag{67}$$

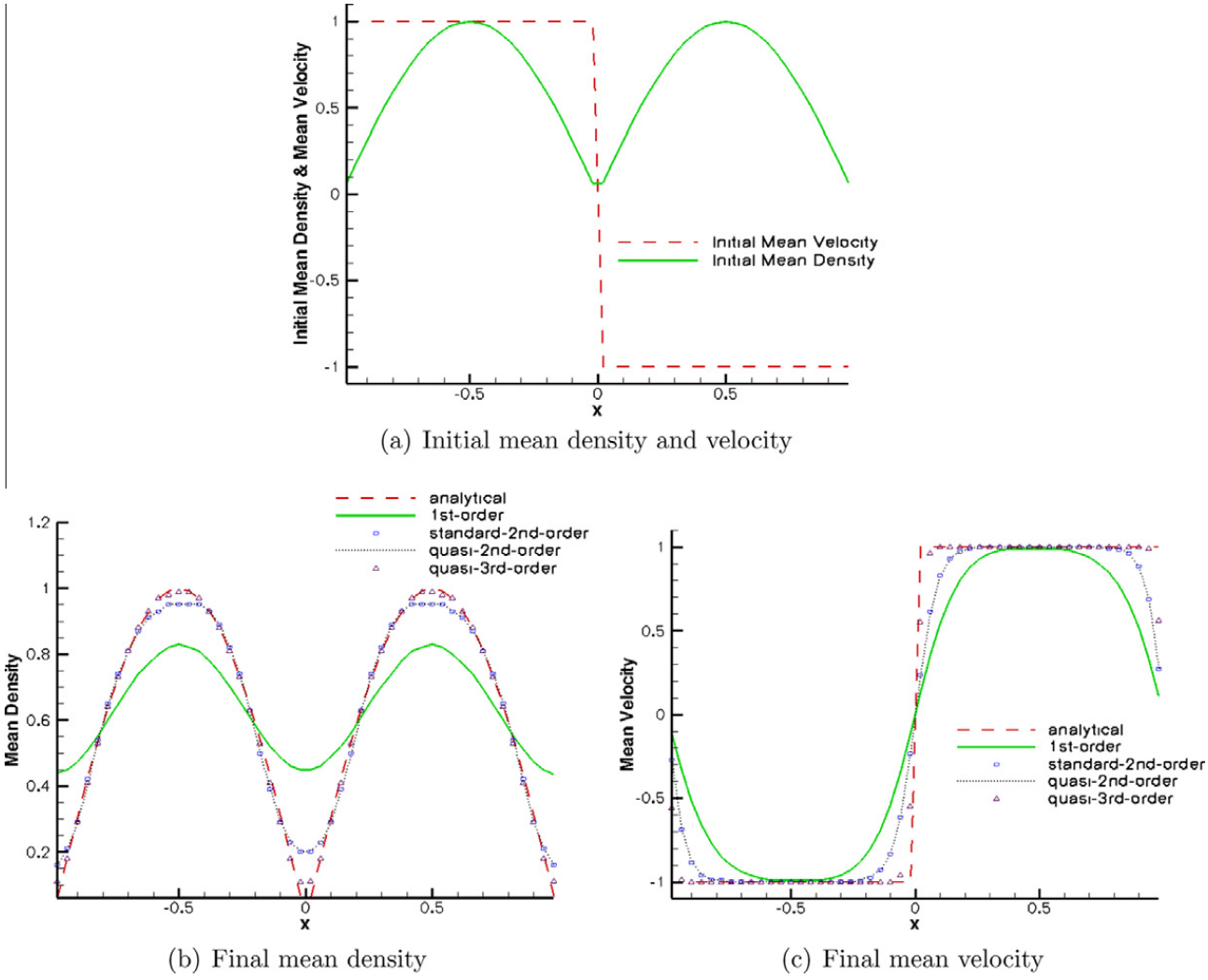

Fig. 3. Comparison of schemes for 1-D case with only convection terms.

Let $\Delta t_{\mathrm{CFL},i}$ and $\Delta t_{\mathrm{realizable},i}$ be the time-step sizes in the ith cell satisfying the CFL and realizability criteria, respectively. An obvious way to calculate global time-step size is:

$$\Delta t = \min_i \left(\Delta t_{\mathrm{CFL},i}, \Delta t_{\mathrm{realizable},i}\right). \tag{68}$$

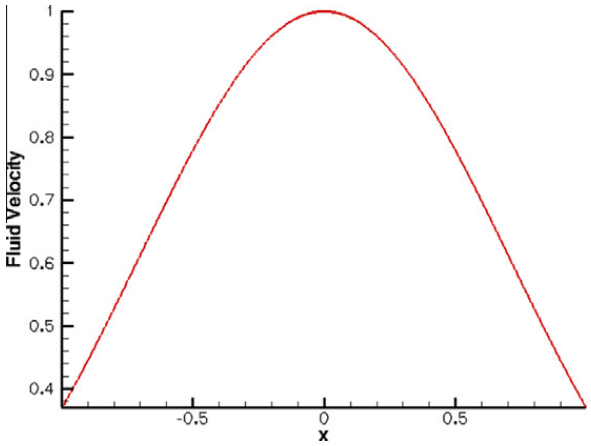

Fig. 4. Fluid velocity for 1-D case.

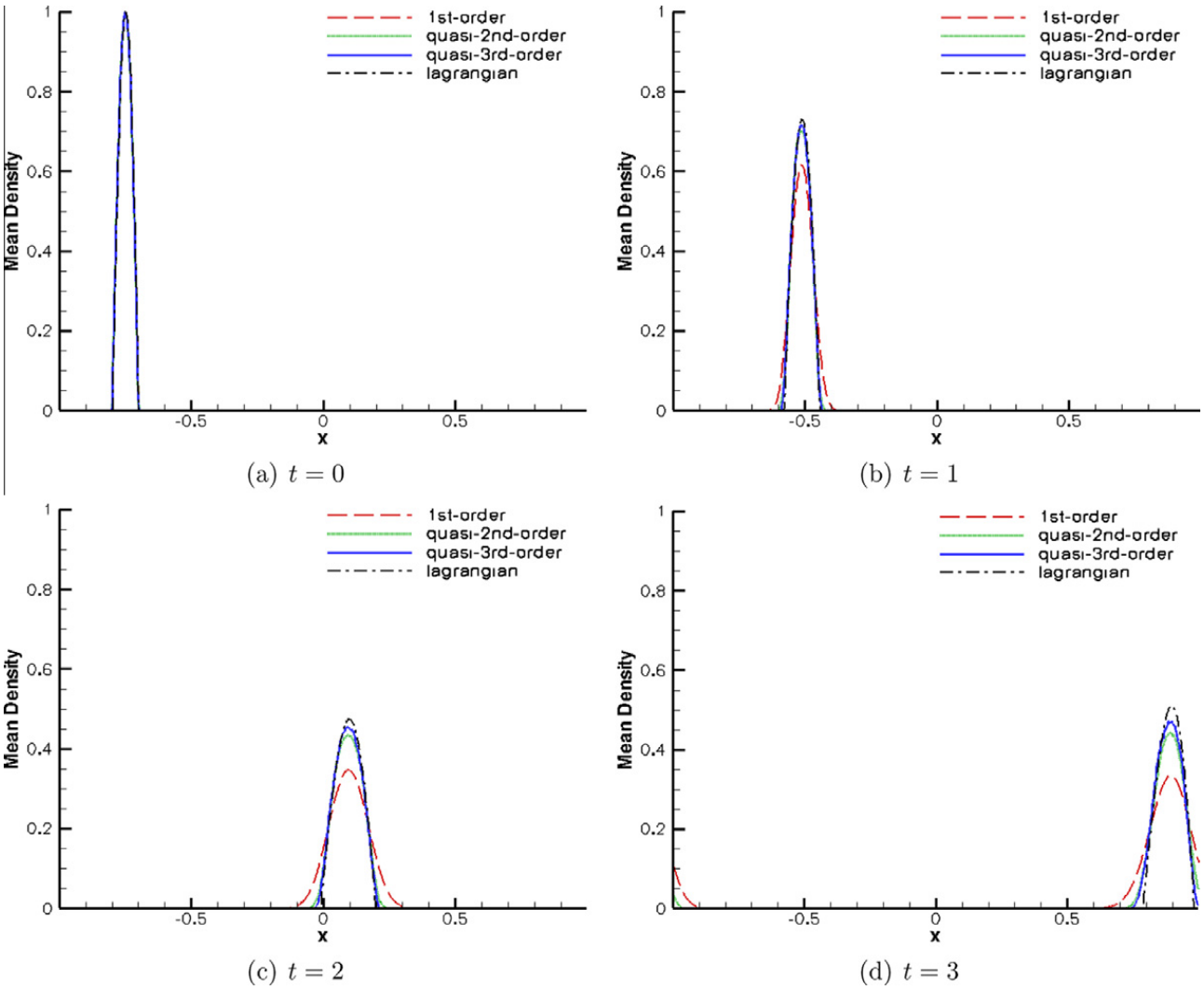

Fig. 5. Comparison of schemes using mean density for 1-D case with convection and drag terms.

However, in the present paper a slightly different approach is used for the calculation of Δt.

As stated earlier, in general $\Delta t_{\text{realizable},i} < \Delta t_{\text{CFL},i}$. This implies that most of the time $\Delta t = \min_i(\Delta t_{\text{realizable},i})$. To have a better time accuracy, a multi-stage time-integration is used. For a multi-stage time-integration, Δt calculated for the 1st-stage is used in all the later stages. But $\min_i(\Delta t_{\text{realizable},i})$ may be different for different stages of time integration and in that case Δt calculated for the 1st-stage will not satisfy the realizability criterion for a later stage. In order to get rid of this problem, the following approach is used. Suppose, the problem under consideration uses a *quasi-p*th-order finite-volume scheme. A value of the CFL $\sim O(1)$ is pre-specified and Δt is calculated as $\Delta t = \min_i(\Delta t_{\text{CFL},i})$. During each stage of time integration, this Δt is used as the global time-step size, and the realizability condition corresponding to *quasi-p*th-order finite-volume scheme is checked in each cell. For the cells in which the check succeeds, a *quasi-p*th-order reconstruction is used for the weights. And for the cells in which the check fails, a 1st-order reconstruction is used for the weights. Thereafter, in the failed cells, the realizability criterion corresponding to 1st-order finite-volume scheme is checked. This check succeeds on almost all occasions if the CFL is not very large, thereby satisfying the realizability criterion in each cell. However, if the last check fails, the whole process is re-initiated using a smaller CFL.

5. Numerical results

In this section several results are presented for 1-D and 2-D cases. For all the cases a 2nd-order RK2SSP scheme is used for time-integration and Δt is calculated using CFL = 0.5 unless otherwise stated. Periodic boundary conditions are used for all 1-D cases while for the 2-D cases a combination of wall, Dirichlet and periodic boundary conditions is used. The domain for the 1-D cases is defined by $x \in [-1, 1]$. The simulations consider either the spatial flux terms alone or in combination with fluid drag terms. Collisions are not included. The drag force is calculated using (5) and the abscissas are updated using simple kinematic relations. More details can be found in [17–19]. The drag terms are dependent on the Stokes number defined by

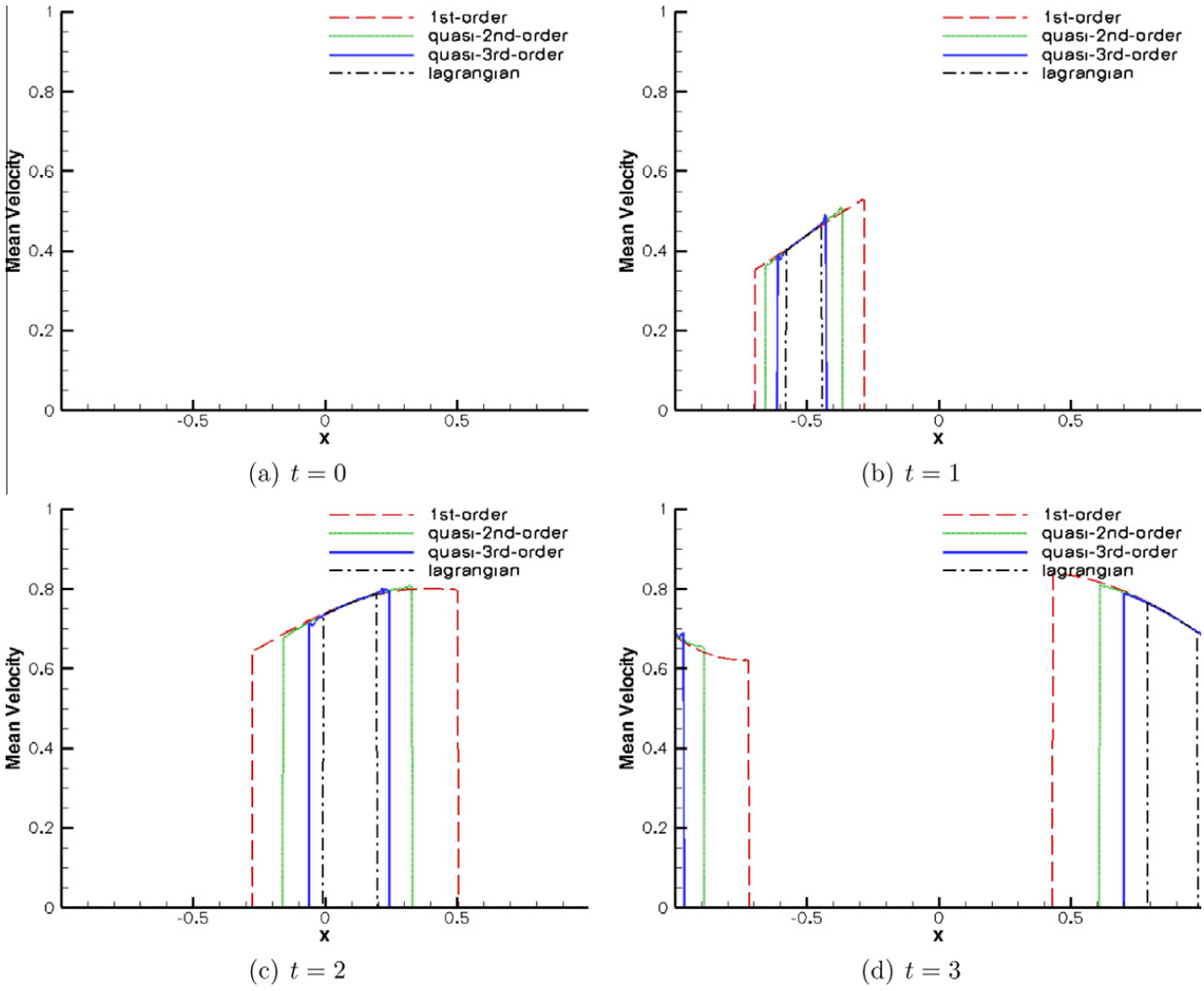

Fig. 6. Comparison of schemes using mean velocity for 1-D case with convection and drag terms.

$$\mathrm{St} = \frac{1}{18}\frac{\rho_p d_p^2}{\mu_g}. \tag{69}$$

For the two cases presented here that include the drag terms, St = 1 is used.

For the 1-D cases, the numerical results using the *quasi*-2nd-order and *quasi*-3rd-order schemes are presented. For the *quasi*-2nd-order scheme, a linear reconstruction with the minmod limiter as described in Section 2.4.2 is used while for the *quasi*-3rd-order scheme a MUSCL technique [27,49] is used. For the 2-D cases, results using the *quasi*-2nd-order scheme are presented. A 2nd-order least-squares reconstruction [1,50] is used for the weights using neighboring cell-averaged values. Moreover, a limiter [1,20,50] is applied to the least-squares reconstruction to avoid spurious oscillations. The results for the *standard* 2nd-order scheme are presented only for the simplest 1-D cases with constant abscissas because of the non-realizability problem. In all the cases, the mean density is computed using $\sum n_\alpha$ and the mean velocity is computed with $\sum(n_\alpha u_\alpha)/\sum n_\alpha$.

5.1. Spatial accuracy study

It was stated earlier that although in general the new *quasi* schemes have lower spatial accuracy compared to *standard* schemes, for problems where the velocities are constant over a range of cells, both schemes have almost the same order of spatial accuracy. Here, the order of spatial accuracy of the *standard* and the new *quasi* schemes is discussed for cases where

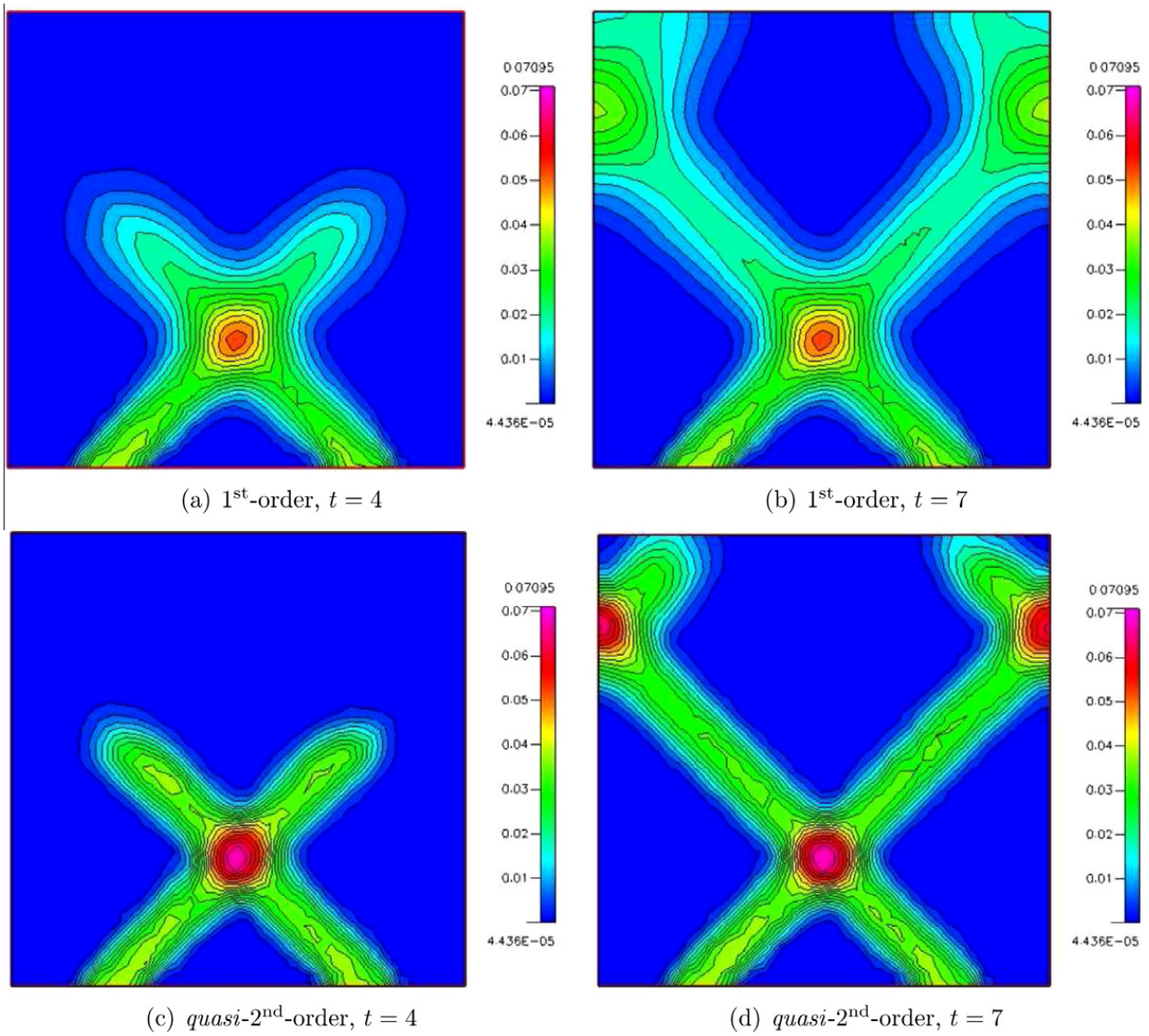

Fig. 7. Mean particle density for crossing particle jets for different schemes at two times.

the abscissas are constant over a range of cells. Results in this section are based on simple 1-D cases for which analytical solutions exist. L_1 errors are calculated by comparing the numerical solution with the analytical solution for mean density. L_1 errors and order of convergence are presented for four schemes: 1st-order, *standard* 2nd-order, *quasi*-2nd-order, *quasi*-3rd-order. Tables 1 and 2 show the results for 1-node and 2-node quadrature, respectively. For both cases $\Delta t = 0.001$ is used. For 1-node quadrature, the initial weight (n) and abscissa (U) are given as:

$$n = 1.0 + \sin(\pi x), \quad U = 1. \tag{70}$$

For 2-node quadrature, initial weights (n_1, n_2) and abscissas (U_1, U_2) are given as:

$$\begin{aligned} &n_1 = |\sin(\pi x)|, \quad n_2 = 0, \quad U_1 = 1, \quad U_2 = 0 \quad \text{for } x \in [-1, 0), \\ &n_1 = 0, \quad n_2 = |\sin(\pi x)|, \quad U_1 = 0, \quad U_2 = -1 \quad \text{for } x \in [0, 1]. \end{aligned} \tag{71}$$

For 1-node quadrature, errors are calculated at $t = 4$, while for 2-node quadrature errors are calculated at $t = 1$. It is observed that the formal order of accuracy can be obtained for 1-node quadrature, but as the number of quadrature nodes is increased, the order of accuracy for all schemes decreases. Although the reason for this loss of order of accuracy has not been studied extensively, it can be attributed to the combined effects of an increase in the number of equations and use of the moment-inversion algorithm for ill-conditioned points. At many points the values of the weights are very small and the non-linear equations solved using the moment-inversion algorithm are often ill-conditioned for these points in the case of multiple quadrature nodes. It can also be observed that the *quasi*-2nd-order scheme always has approximately the same order of convergence as the *standard* 2nd-order scheme, and the *quasi*-3rd-order scheme is better compared to both.

5.2. Grid convergence study

Grid convergence studies for the *quasi*-2nd-order and *quasi*-3rd-order schemes in 1-D are presented in Fig. 2. For both the schemes, the mean density obtained using different grid resolutions is compared with the analytical solution. Four different uniform grids have been considered with the number of cells equal to 25, 50, 100, 200. The comparisons have been done for a 2-node quadrature case with the same initializations as in (71). Fig. 2(a) and (b) shows grid convergence for the *quasi*-2nd-order and *quasi*-3rd-order schemes, respectively. As the number of grid cells is increased, the solutions using both schemes converge towards the analytical solution.

5.3. Comparison of schemes for 1-D case with only convection terms

For this case, 2-node quadrature is used with the weights being sinusoidal functions and the abscissas being square functions. Fifty grid points are used and the results are shown in Fig. 3. The initial ($t = 0$) conditions are same as in (71) and are shown in Fig. 3(a). Fig. 3(b) and (c) shows the final conditions for the mean density and mean velocity, respectively ($t = 1$). The weight distribution is symmetric about $x = 0$, with the left wave moving towards the right (negative abscissa) and the right wave moving towards the left (positive abscissa). The final time has been chosen such that the waves coalesce at $x = 0$ and then separate again. Four different schemes have been compared: 1st-order, *standard* 2nd-order, *quasi*-2nd-order,

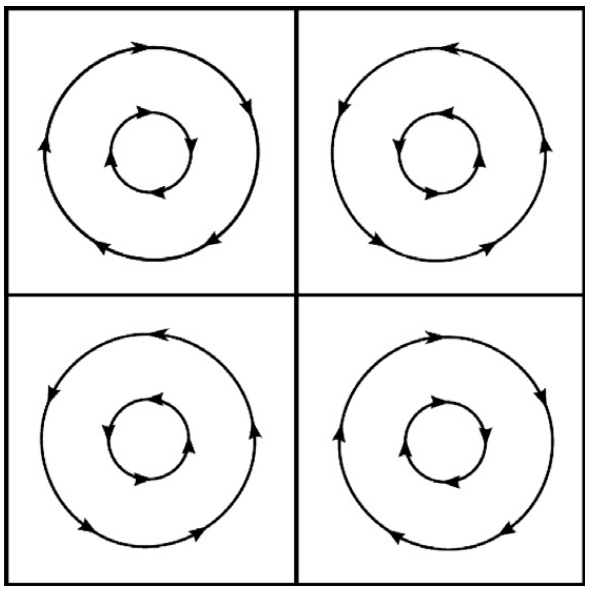

Fig. 8. Fluid velocity for 2-D Taylor–Green flow.

quasi-3rd-order. The *standard* 2nd-order and *quasi*-2nd-order results are on top of each other. The *quasi*-3rd-order scheme shows an improvement over the *quasi*-2nd-order scheme.

5.4. Comparison of schemes for 1-D case with both convection and drag terms

For this case, 4-node quadrature is used. Initial conditions ($t = 0$) for $x \in [-0.8, -0.7]$ are given as:

$$\begin{aligned} n_1 &= n_2 = n_3 = n_4 = \sin(10\pi(x + 0.8)), \\ U_1 &= U_2 = U_3 = U_4 = 0. \end{aligned} \tag{72}$$

Everywhere else both the weights and abscissas are zero. The particle flow is driven by fluid drag. The fluid velocity is given by

$$U_g = e^{-x^2}, \tag{73}$$

and is shown in Fig. 4. The QMOM results using three schemes – 1st-order, *quasi*-2nd-order, *quasi*-3rd-order – are compared with the Lagrangian results. The results for the mean density are presented at four different times in Fig. 5. The *quasi*-3rd-order and *quasi*-2nd-order results are closer to the Lagrangian results as compared to the 1st-order ones. Fig. 6 shows

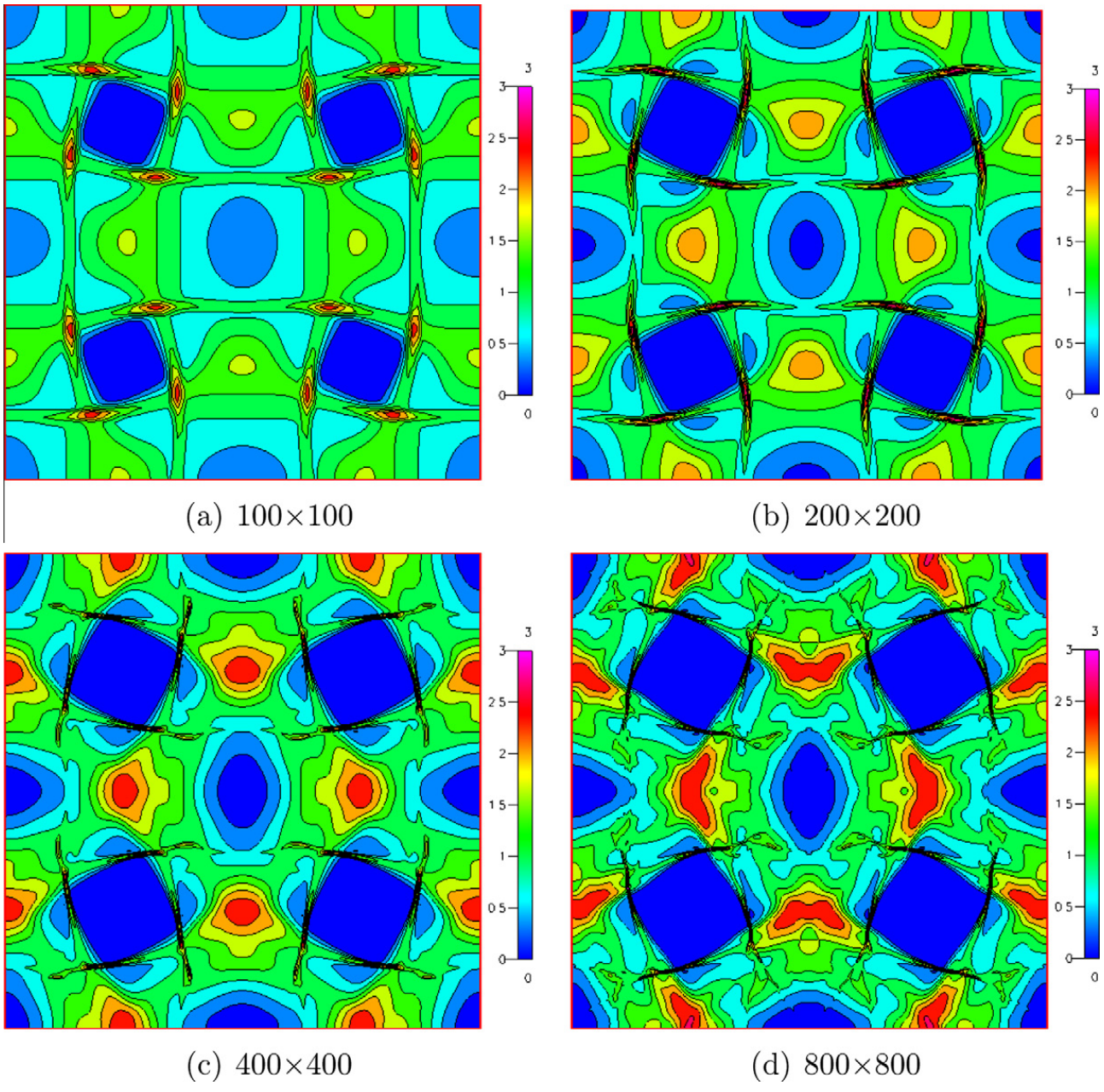

Fig. 9. Grid resolution study of mean particle density in 2-D Taylor–Green flow at $t = 4$ with 1st-order scheme.

comparisons for the mean velocity at the same times as in Fig. 5. It is evident that the mean velocity is not constant but varies over a range of cells at each time. Clearly in such scenarios as well, the new realizable schemes give better solutions compared to the 1st-order scheme.

5.5. Comparison of schemes for 2-D case with only convection terms

Here a dilute impinging-jet problem in 2-D is presented. The domain consists of a square (7×7) box with two openings on the bottom wall through which particle jets enter. As time progresses, the jets cross each other, strike the wall and then rebound. These simulations are done using 4-node quadrature (n_1, U_1, V_1), (n_2, U_2, V_2), (n_3, U_3, V_3), (n_4, U_4, V_4). Initial $(t = 0)$ conditions are given as:

$$\begin{aligned} &n_1 = n_2 = n_3 = n_4 = 0.0001, \\ &U_1 = 0.001, \quad U_2 = -0.001, \quad U_3 = 0.001, \quad U_4 = -0.001, \\ &V_1 = 0.001, \quad V_2 = 0.001, \quad V_3 = -0.001, \quad V_4 = -0.001. \end{aligned} \tag{74}$$

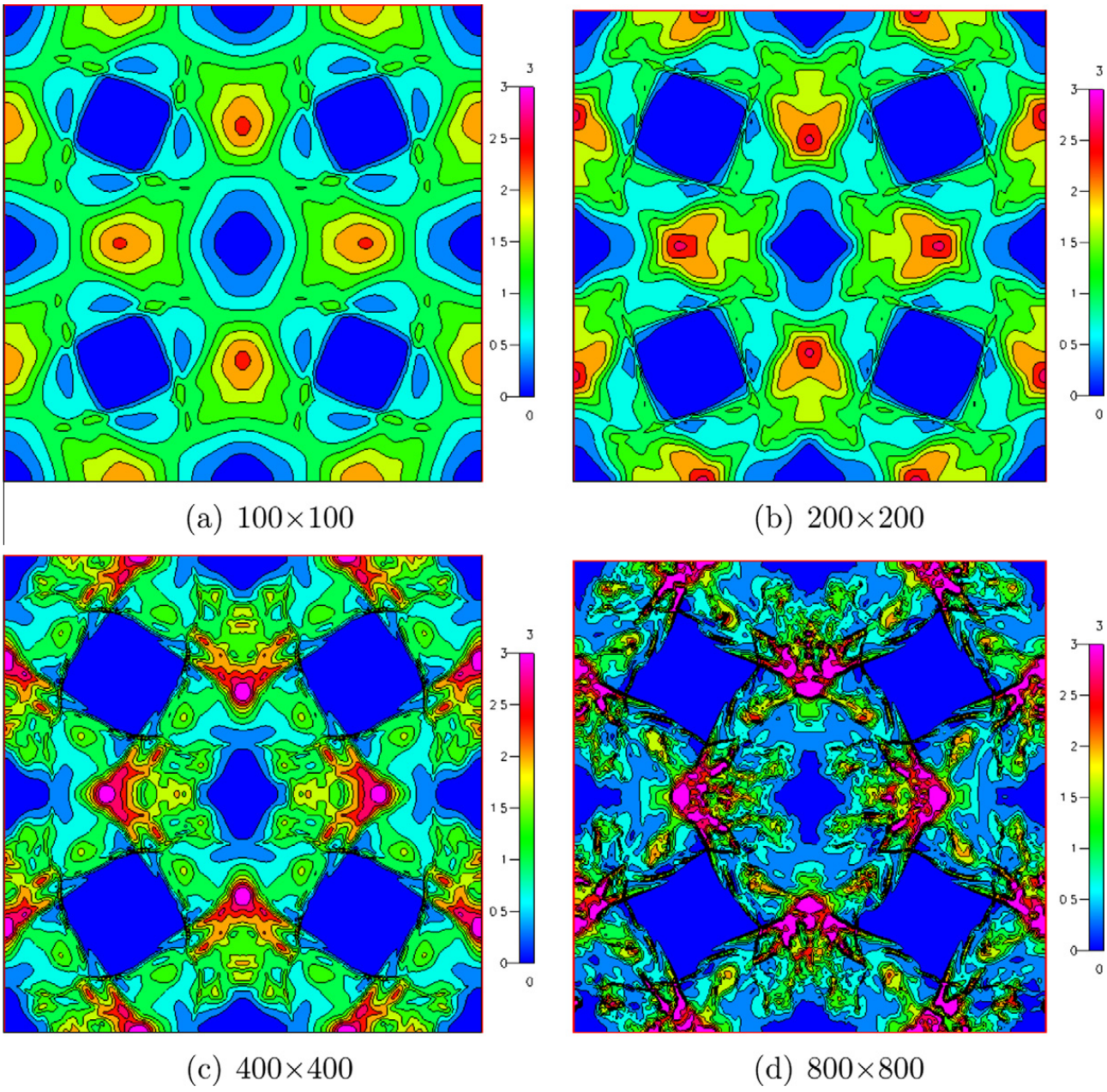

Fig. 10. Grid resolution study of mean particle density in 2-D Taylor–Green flow at $t = 4$ with *quasi*-2nd-order scheme.

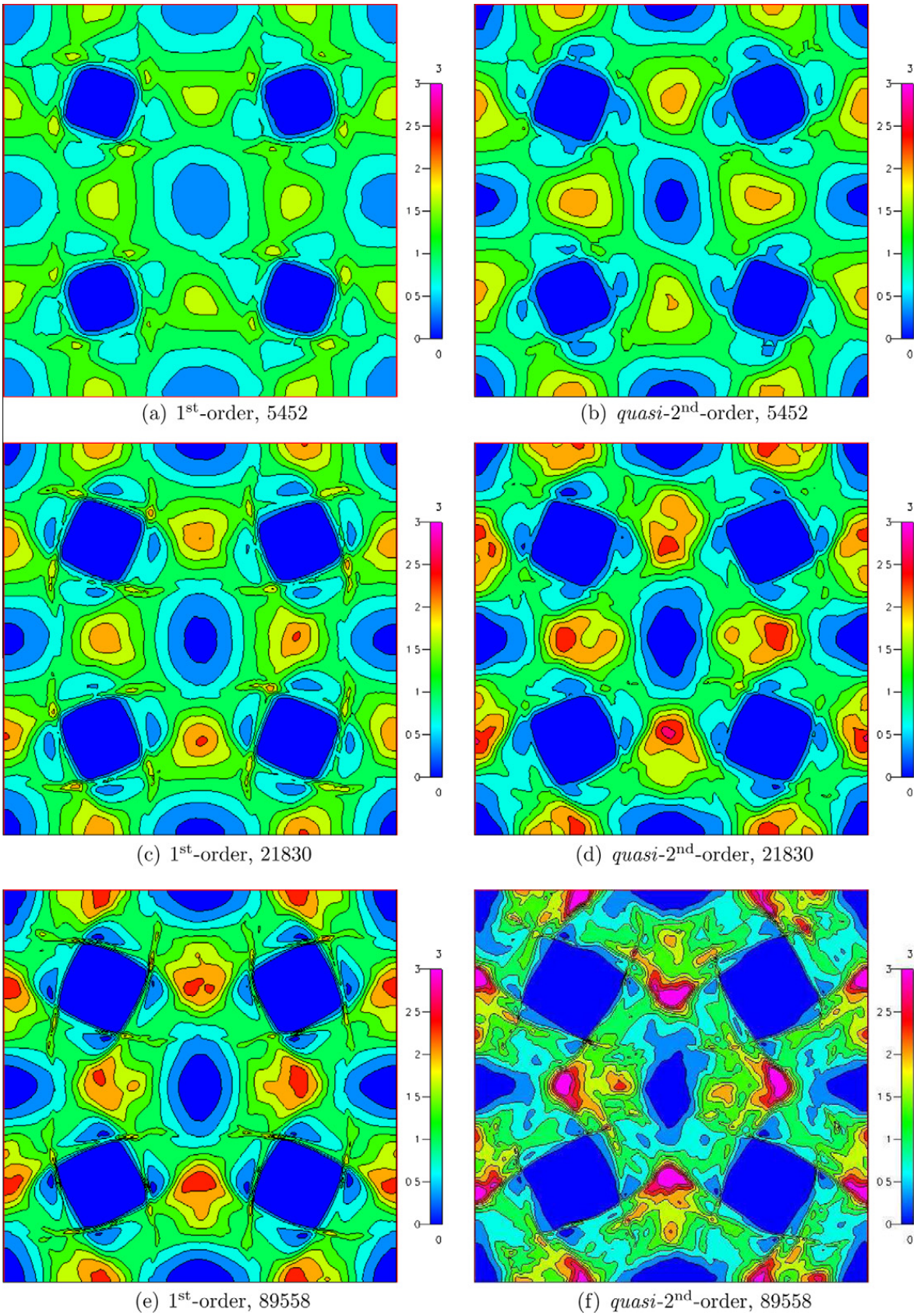

Fig. 11. Mean particle density in 2-D Taylor–Green flow at $t = 4$ on triangular mesh with different schemes and cell numbers.

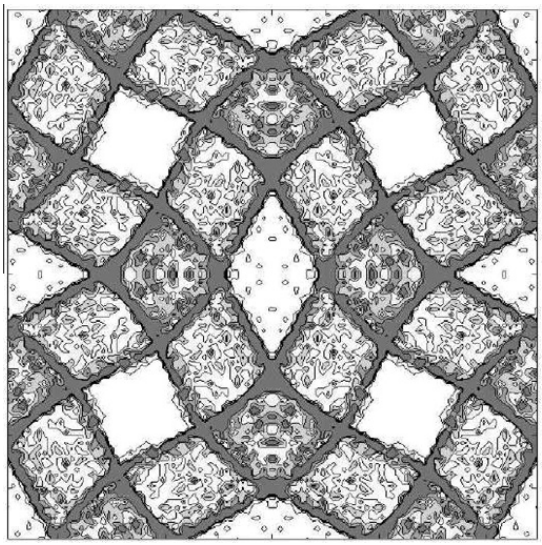

Fig. 12. Particle number density in 2-D Taylor–Green flow obtained by Lostec et al. [30] using a Lagrangian simulation. Density is proportional to darkness.

The values of the weights and abscissas at the left inlet jet (Dirichlet) are:

$$\begin{aligned} &n_1 = n_2 = n_3 = n_4 = 0.01, \\ &U_1 = 1.001, \quad U_2 = 0.999, \quad U_3 = 1.001, \quad U_4 = 0.999, \\ &V_1 = 1.001, \quad V_2 = 1.001, \quad V_3 = 0.999, \quad V_4 = 0.999. \end{aligned} \tag{75}$$

And the values at the right inlet jet (Dirichlet) are:

$$\begin{aligned} &n_1 = n_2 = n_3 = n_4 = 0.01, \\ &U_1 = -1.001, \quad U_2 = -0.999, \quad U_3 = -1.001, \quad U_4 = -0.999, \\ &V_1 = 1.001, \quad V_2 = 1.001, \quad V_3 = 0.999, \quad V_4 = 0.999. \end{aligned} \tag{76}$$

For elastic collisions with the walls, $e_w = 1$. Results are presented for the 1st-order and *quasi*-2nd-order schemes in Fig. 7. The computational grid contains 2562 triangular elements. Fig. 7(a) and (c) show the mean density using the 1st-order and *quasi*-2nd scheme, respectively, at $t = 4$ after the jets cross. Fig. 7(b) and (d) shows the mean density at $t = 7$ after the jets bounce off the walls. The solution obtained using the 1st-order scheme is more diffused. The improvement in the solution using the *quasi*-2nd-order scheme is clearly evident.

5.6. Comparison of schemes for 2-D case with both convection and drag terms

In this case, the evolution of particles in a Taylor–Green flow is presented. The domain consists of a square (1×1) box. All the boundaries are periodic. The fluid velocity in the Taylor–Green flow is given by

$$\begin{aligned} U_{gx} &= \sin(2\pi x)\cos(2\pi y), \\ U_{gy} &= -\sin(2\pi y)\cos(2\pi x), \end{aligned} \tag{77}$$

and is shown in Fig. 8. Figs. 9 and 10 present results for different schemes on a structured mesh, and Fig. 11 presents results for unstructured meshes. Results are presented for the 1st-order and *quasi*-2nd-order schemes at $t = 4$. For the structured mesh, four different grid resolutions are used: 100×100, 200×200, 400×400, 800×800. Lostec et al. [30] presented results for the same problem using both a Lagrangian simulation and 1st-order QMOM. Here, the Lagrangian result obtained in [30] is presented in Fig. 12. Note that the particle number density shown in Fig. 12 is proportional to the mean particle density in our results. The results presented here clearly show that the *quasi*-2nd-order scheme fares much better compared to the 1st-order scheme in resolving the various features obtained in the Lagrangian simulation in [30]. The unstructured mesh consists of triangular cells. Three different grid resolutions are used with 5452, 21,830 and 89,558 triangular cells. The unstructured mesh results also confirm the improvement in the solutions when the *quasi*-2nd-order scheme is used.

6. Conclusions

Hitherto, the use of finite-volume schemes for quadrature-based moment methods was limited to the 1st-order scheme in order to guarantee realizability. Over the years, an extensive research effort has been spent on developing high-order finite-volume schemes in the field of computational fluid dynamics. However, the issue of non-realizability has often acted

as a barrier, making these high-order finite-volume schemes inaccessible to models based on quadrature-based moment methods. Because of the non-realizability problem with *standard* high-order finite-volume schemes and the high numerical dissipation inherent in the 1st-order finite-volume scheme, there is a need to develop schemes that can provide less-diffused results and guarantee realizability simultaneously.

In the present work, realizability of finite-volume schemes in both space and time has been discussed for the first time. A generalized idea has been proposed to develop high-order realizable finite-volume schemes. According to the new idea, a *quasi-p*th-order realizable finite-volume scheme can be constructed using a *p*th-order reconstruction for the weights and a 1st-order reconstruction for the abscissas along with a realizability criterion. These new schemes give remarkably improved solutions for a class of problems where the velocity field is constant or slowly varying over a range of cells. This marks a significant step as it makes all the high-order finite-volume schemes, developed over the years for fluid flows, automatically accessible to quadrature-based moment methods. It has also been shown that the standard Runge–Kutta time-integration schemes do not guarantee realizability. Instead, strong-stability-preserving Runge–Kutta schemes must be used. Numerical simulations have been carried out using both Cartesian and triangular meshes, and clearly demonstrate the increased accuracy and robustness of the proposed realizable high-order schemes.

Acknowledgments

The study was funded by NSF Grant CISE-0830214. The views and conclusions herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF or the US Government.

Appendix A. Realizable advection of velocity-independent density functions

For simplicity, only the one-dimensional case is discussed here. Discussion on two- and three-dimensional cases can be based by analogy on Section 3.3. If the density function does not include the velocity, v is not an independent variable. Denote the density function by $f \equiv f(b,x,t)$, where b is some scalar measure. For example, if b denotes the particle radius, then f is the density function for the particle size distribution. The population balance equation[1] with only the advection term can be written as:

$$\frac{\partial f}{\partial t} + \frac{\partial v f}{\partial x} = 0, \tag{A.1}$$

where $v \equiv v(x,t)$. The pth-order moment of f can be defined as:

$$M^p = \int b^p f \, db. \tag{A.2}$$

Applying the definition of moments to (A.1), the moment transport equations can be written as:

$$\frac{\partial M^p}{\partial t} + \frac{\partial v M^p}{\partial x} = 0 \quad \forall p \in \{0,1,2,\ldots\}. \tag{A.3}$$

Define the set of conserved moments as $\mathbf{W} = [M^0 \; M^1 \; M^2 \; \cdots]^{\mathrm{T}}$ and the set of moment fluxes as $\mathbf{H}(v,\mathbf{W}) = v\mathbf{W} = v[M^0 \; M^1 \; M^2 \; \cdots]^{\mathrm{T}}$. Then the set of moment transport equations can be written as:

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{H}(v,\mathbf{W})}{\partial x} = 0. \tag{A.4}$$

The conserved moments and moment fluxes can be written in terms of the density function:

$$\mathbf{W} = \int \mathbf{K} f \, db, \mathbf{H} = v \int \mathbf{K} f \, db, \tag{A.5}$$

where

$$\mathbf{K} = [1 \; b \; b^2 \; \cdots]^{\mathrm{T}}. \tag{A.6}$$

The moments can be advanced in time using a finite-volume scheme. If a single-stage Euler explicit time-integration scheme is used, the updated set of moments can be written as:

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \lambda \left[\mathbf{G}\left(v_{i+1/2,l}^n, \mathbf{W}_{i+1/2,l}^n, v_{i+1/2,r}^n, \mathbf{W}_{i+1/2,r}^n \right) - \mathbf{G}\left(v_{i-1/2,l}^n, \mathbf{W}_{i-1/2,l}^n, v_{i-1/2,r}^n, \mathbf{W}_{i-1/2,r}^n \right) \right], \tag{A.7}$$

[1] The transport equation for a velocity-independent density function is usually referred to as a population balance equation. In the aerosol literature, it is referred to as the general dynamic equation.

where $\lambda = \Delta t/\Delta x$. $\mathbf{G}$ is the numerical flux function defined as:

$$\mathbf{G}(v_l^n, \mathbf{W}_l^n, v_r^n, \mathbf{W}_r^n) = v_l^+ \int \mathbf{K} f_l \, db + v_r^- \int \mathbf{K} f_r \, db, \tag{A.8}$$

where

$$v_l^+ = \frac{1}{2}(v_l + |v_l|) \quad \text{and} \quad v_r^- = \frac{1}{2}(v_r - |v_r|). \tag{A.9}$$

The updated set of moments can be written as:

$$\mathbf{W}^{n+1} = \int \mathbf{K} h \, db, \tag{A.10}$$

where

$$\begin{aligned} h &= f_i^n - \lambda\left(v_{i+1/2,l}^{n+} f_{i+1/2,l}^n + v_{i+1/2,r}^{n-} f_{i+1/2,r}^n - v_{i-1/2,l}^{n+} f_{i-1/2,l}^n - v_{i-1/2,r}^{n-} f_{i-1/2,r}^n\right) \\ &= f_i^n - \lambda v_{i+1/2,l}^{n+} f_{i+1/2,l}^n - \lambda v_{i+1/2,r}^{n-} f_{i+1/2,r}^n + \lambda v_{i-1/2,l}^{n+} f_{i-1/2,l}^n + \lambda v_{i-1/2,r}^{n-} f_{i-1/2,r}^n. \end{aligned} \tag{A.11}$$

Assuming that the moments at time level n correspond to a non-negative density function, three terms on the right-hand side of (A.11) – first, third, fourth – are non-negative and two terms – second, fifth – are non-positive. For the moments at time level $n+1$ to correspond to a non-negative density function, h should be non-negative for all b. The non-negativity of h can only be guaranteed for the 1st-order finite-volume scheme under a constraint on λ given by

$$\left(1 - \lambda\left|v_i^n\right|\right) \geqslant 0. \tag{A.12}$$

For 2nd-order and, in general, any high-order finite-volume schemes, non-negativity of h cannot be guaranteed. However, if QMOM is used along with the *special* reconstruction discussed in Section 3, non-negativity of h can be guaranteed as shown below. For β-node QMOM, the density function can be written as:

$$f = \sum_{\alpha=1}^{\beta} n_\alpha \delta(b - B_\alpha), \tag{A.13}$$

where n_α are weights and B_α are abscissas.

Theorem 2. *Let $\beta, p \in \mathbb{N}$ and $\alpha \in \{1,2,\ldots,\beta\}$. Also let the cell-averaged and reconstructed values of the weights satisfy $n_{i,\alpha}^n > 0$ and $n_{i+1/2,\alpha,l}^n, n_{i-1/2,\alpha,r}^n \geqslant 0$ $\forall \alpha$. If a finite-volume scheme using a single-stage Euler explicit time-integration scheme is devised that uses a pth-order reconstruction for the weights and 1st-order reconstruction for the abscissas, the non-negativity of the effective density function (A.11) in the ith cell can always be guaranteed under an explicit constraint on time-step size* ($\Delta t \in \mathbb{R}^+$).

Proof. Using (A.11), the effective density function, regardless of the finite-volume scheme used, can be written as:

$$h = f_i^n - \lambda v_{i+1/2,l}^{n+} f_{i+1/2,l}^n + \lambda v_{i-1/2,r}^{n-} f_{i-1/2,r}^n + \xi^+. \tag{A.14}$$

In the above expression, the third and fourth non-negative terms have been grouped under ξ^+. For a β-node quadrature, using (A.13), the expression for h becomes:

$$h = \sum_{\alpha=1}^{\beta} \left[n_{i,\alpha}^n \delta\left(b - B_{i,\alpha}^n\right) - \lambda v_{i+1/2,l}^{n+} n_{i+1/2,\alpha,l}^n \delta\left(b - B_{i+1/2,\alpha,l}^n\right) + \lambda v_{i-1/2,r}^{n-} n_{i-1/2,\alpha,r}^n \delta\left(b - B_{i-1/2,\alpha,r}^n\right)\right] + \xi^+. \tag{A.15}$$

If a 1st-order reconstruction is used for the abscissas, then the interface values of the abscissas will be the same as the cell-averaged values:

$$B_{i,\alpha}^n = B_{i+1/2,\alpha,l}^n = B_{i-1/2,\alpha,r}^n. \tag{A.16}$$

Putting this in (A.15), the effective density function becomes:

$$h = \sum_{\alpha=1}^{\beta} \left\{n_{i,\alpha}^n - \lambda v_{i+1/2,l}^{n+} n_{i+1/2,\alpha,l}^n + \lambda v_{i-1/2,r}^{n-} n_{i-1/2,\alpha,r}^n\right\} \delta\left(b - B_{i,\alpha}^n\right) + \xi^+. \tag{A.17}$$

For $\Delta t \in \mathbb{R}^+$ satisfying the condition:

$$\lambda = \min_{\alpha \in \{1,2,\ldots\beta\}} \left(\frac{n_{i,\alpha}^n}{n_{i+1/2,\alpha,l}^n v_{i+1/2,l}^{n+} - n_{i-1/2,\alpha,r}^n v_{i-1/2,r}^{n-}}\right), \tag{A.18}$$

h is non-negative for all b. This concludes the proof. □

The extension to realizable high-order time integration is analogous to the methods presented in Section 4.

References

[1] T.J. Barth, D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA 89-0366, 1989.
[2] T.J. Barth, P.O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA 90-0013, 1990.
[3] J.-D. Benamou, Big ray tracing: multivalued travel time field computation using viscosity solutions of the Eikonal equation, Journal of Computational Physics 128 (1996) 463–474.
[4] A.E. Beylich, Solving the kinetic equation for all Knudsen numbers, Physics of Fluids 12 (2000) 444–465.
[5] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Physical Reviews 94 (1954) 511–525.
[6] G.A. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Clarendon Press, Oxford, 1994.
[7] F. Bouchut, S. Jin, X.T. Li, Numerical approximations of pressureless gas and isothermal gas dynamics, SIAM Journal of Numerical Analysis 41 (2003) 135–158.
[8] J.A. Carrillo, A. Majorana, F. Vecil, A semi-Lagrangian deterministic solver for the semiconductor Boltzmann–Poisson system, Communications in Computational Physics 2 (2007) 1027–1054.
[9] C. Cercignani, The Boltzmann Equation and Its Applications, Springer, New York, 1988.
[10] S. Chapman, T.G. Cowling, The Mathematical Theory of Nonuniform Gases, Cambridge University Press, Cambridge, 1970.
[11] C.K. Chu, Kinetic-theoretic description of the formation of a shock wave, Physics of Fluids 8 (1965) 12–22.
[12] P. Collela, The piecewise parabolic method for gas-dynamical simulations, Journal of Computational Physics 54 (1984) 174–201.
[13] S. Deshpande, A second order accurate, kinetic theory based, method for inviscid compressible flows, Technical Report NASA Langley 2613, 1986.
[14] O. Desjardins, R.O. Fox, P. Villedieu, A quadrature-based moment method for dilute fluid–particle flows, Journal of Computational Physics 227 (2008) 2514–2539.
[15] B. Engquist, O. Runborg, Multiphase computations in geometrical optics, Journal of Computational and Applied Mathematics 74 (1996) 175–192.
[16] H. Enwald, E. Peirano, A.E. Almstedt, Eulerian two-phase flow theory applied to fluidization, International Journal of Multiphase Flow 22 (1996) 21–66.
[17] R.O. Fox, A quadrature-based third-order moment method for dilute gas–particle flows, Journal of Computational Physics 227 (2008) 6313–6350.
[18] R.O. Fox, Higher-order quadrature-based moment methods for kinetic equations, Journal of Computational Physics 228 (2009) 7771–7791.
[19] R.O. Fox, Optimal moment sets for multivariate direct quadrature method of moments, Industrial and Engineering Chemistry Research 48 (2009) 9686–9696.
[20] S.K. Godunov, Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics, Math. Sbornik 47 (1959) 271–306.
[21] L. Gosse, Using K-branch entropy solutions for multivalued geometric optics computations, Journal of Computational Physics 180 (2002) 155–182.
[22] L. Gosse, S. Jin, X.T. Li, On two moment systems for computing multiphase semiclassical limits of the Schrödinger equation, Mathematical Models and Methods in Applied Science 13 (2003) 1689–1723.
[23] S. Gottlieb, C.W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Review 43 (2001) 89–112.
[24] H. Grad, On the kinetic theory of rarefied gases, Communications on Pure and Applied Mathematics 2 (1949) 331–407.
[25] S. Jin, X.T. Li, Multi-phase computations of the semiclassical limit of the Schrödinger equation and related problems: Whitham vs Wigner, Physica D 182 (2003) 46–85.
[26] S. Jin, H. Liu, S. Osher, R. Tsai, Computing multi-valued physical observables for the semiclassical limit of the Schrödinger equation, Journal of Computational Physics 205 (2005) 222–241.
[27] D.D. Knight, Elements of Numerical Methods for Compressible Flows, Cambridge University Press, Cambridge, 2006.
[28] R. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge, 2002.
[29] X.T. Li, J.G. Wöhlbier, S. Jin, J.H. Booske, Eulerian method for computing multivalued solutions of the Euler–Poisson equations and applications to wave breaking in klystrons, Physical Review E 70 (2004) 016502.
[30] N.L. Lostec, R.O. Fox, O. Simonin, P. Villedieu, Numerical description of dilute particle-laden flows by a quadrature-based moment method, in: Proceedings of the Summer Program 2008, Center for Turbulence Research, Stanford University, 2008, pp. 209–221.
[31] D.L. Marchisio, R.O. Fox, Solution of population balance equations using the direct quadrature method of moments, Journal of Aerosol Science 36 (2005) 43–73.
[32] R. McGraw, Description of aerosol dynamics by the quadrature method of moments, Aerosol Science and Technology 27 (1997) 255–265.
[33] R. McGraw, Correcting moment sequences for errors associated with advective transport, 2006. <http://www.ecd.bnl.gov/pubs/momentcorrection_mcgraw2006.pdf>.
[34] R. McGraw, Numerical advection of correlated tracers: Preserving particle size/composition moment sequences during transport of aerosol mixtures, Journal of Physics: Conference Series 78 (2007) 012045.
[35] I. Nicodin, R. Gatignol, Unsteady half-space evaporation and condensation problems on the basis of the discrete kinetic theory, Physics of Fluids 18 (2006) 127105.
[36] Y. Ogata, H.-N. Im, T. Yabe, Numerical method for Boltzmann equation with Soroban-grid CIP method, Communications in Computational Physics 2 (2007) 760–782.
[37] A. Passalacqua, R.O. Fox, R. Garg, S. Subramaniam, A fully coupled quadrature-based moment method for dilute to moderately dilute fluid–particle flows, Chemical Engineering Science 65 (2010) 2267–2283.
[38] B. Perthame, Boltzmann type schemes for compressible Euler equations in one and two space dimensions, SIAM Journal of Numerical Analysis 29 (1990) 1–19.
[39] M.J. Prather, Numerical advection by conservation of second-order moments, Journal of Geophysical Research 91 (1986) 6671–6681.
[40] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran 77: The Art of Scientific Computing, Cambridge University Press, Cambridge, 1992.
[41] D.I. Pullin, Direct simulation methods for compressible inviscid ideal gas-flows, Journal of Computational Physics 34 (1980) 53–66.
[42] O. Runborg, Some new results in multiphase geometrical optics, Mathematical Modelling and Numerical Analysis 34 (2000) 1203–1231.
[43] O. Runborg, Mathematical models and numerical methods for high frequency waves, Communications in Computational Physics 2 (2007) 827–880.
[44] M. Sakiz, O. Simonin, Numerical experiments and modelling of non-equilibrium effects in dilute granular flows, in: Proceedings of the 21st International Symposium on Rarefied Gas Dynamics, Cépaduès-Éditions, Toulouse, France, 1998.
[45] L. Schiller, A. Naumann, A drag coefficient correlation, V.D.I. Zeitung 77 (1935) 318–320.
[46] H. Struchtrup, Macroscopic Transport Equations for Rarefied Gas Flows, Springer, New York, 2005.
[47] M. Torrilhon, H. Struchtrup, Regularized 13-moment equations: shock structure calculations and comparison to Burnett models, Journal of Fluid Mechanics 513 (2004) 171–198.
[48] M.W. Vance, K.D. Squires, O. Simonin, Properties of the particle velocity field in gas–solid turbulent channel flow, Physics of Fluids 18 (2006) 063302.
[49] B. van Leer, Towards the ultimate conservative difference schemes V. A second order sequel to Godunov's method, Journal of Computational Physics 135 (1997) 229–248.
[50] Z.J. Wang, A quadtree-based adaptive Cartesian/quad grid flow solver for Navier–Stokes equations, Computers and Fluids 27 (1998) 529–549.

[51] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, Journal of Computational Physics 178 (2002) 210–251.
[52] Z.J. Wang, High order methods for Euler and Navier–Stokes equations on unstructured grids, Journal of Progress in Aerospace Sciences 43 (2007) 1–47.
[53] J.C. Wheeler, Modified moments and Gaussian quadratures, Rocky Mountain Journal of Mathematics 4 (1974) 287–296.
[54] F.A. Williams, Spray combustion and atomization, Physics of Fluids 1 (1958) 541–545.
[55] J.G. Wöhlbier, S. Jin, S. Sengele, Eulerian calculations of wave breaking and multivalued solutions in a traveling wave tube, Physics of Plasmas 12 (2005) 023106.
[56] D.L. Wright, Numerical advection of moments of the particle size distribution in Eulerian models, Journal of Aerosol Science 38 (2007) 352–369.