Modeling Particle-Laden Turbulent Flows with the Conditional Quadrature Method of Moments

by

Dennis Martin Dunn

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2015 by the
Graduate Supervisory Committee:

Kyle D. Squires, Chair
Ronald J. Calhoun
Kangping Chen
Lenore L. Dai
Marcus Herrmann

ARIZONA STATE UNIVERSITY

April 2015

ABSTRACT

Conventional fluid dynamics models such as the Navier-Stokes equations are derived for prediction of fluid motion at or near equilibrium, classic examples being the motion of fluids for which inter-molecular collisions are dominant. Flows at equilibrium permit simplifications such as the introduction of viscosity and also lead to solutions that are single-valued. However, many other regimes of interest include “fluids” far from equilibrium; for example, rarefied gases or particle-laden flows in which the dispersed phase can be comprised of granular solids, droplets, or bubbles. Particle motion in these flows is not typically dominated by collisions and may exhibit significant memory effects; therefore, is often poorly described using continuum, field-based (Eulerian) approaches. Non-equilibrium flows generally lack a straightforward counterpart to viscosity and their multi-valued solutions cannot be represented by most Eulerian methods. This strongly motivates different strategies to address current shortcomings and the novel approach adopted in this work is based on the Conditional Quadrature Method of Moments (CQMOM). In CQMOM, moment equations are derived from the Boltzmann equation using a quadrature approximation of the velocity probability density function (PDF). CQMOM circumvents the drawbacks of current methods and leads to multivariate and multidimensional solutions in an Eulerian frame of reference. In the present work, the discretized PDF is resolved using an adaptive two-point quadrature in three-dimensional velocity space. The method is applied to computation of a series of non-equilibrium flows, ranging from simple two-dimensional test cases to fully-turbulent three-dimensional wall-bounded particle-laden flows. The primary contribution of the present effort is on development, application, and assessment of CQMOM for predicting the key features of dilute particle-laden flows. Statistical descriptors such as mean concentration and mean velocity are in good agreement with previous results, for both collision-less and collisional flows at varying particle Stokes numbers. Turbulent statistics and measures of local accumulation agree less favorably with prior results and identify areas for improvement in the modeling strategy.

ACKNOWLEDGMENTS

I would like to acknowledge the guidance, patience, and support of my committee chair member and adviser, Dr. Kyle Squires, and the members of my committee. My gratitude also extends to Dr. Rodney Fox at the University of Iowa for his willingness to collaborate in implementing the methods which he greatly helped pioneer. I would also like to thank the engineering department for providing funding for my research and travels, and for giving me the opportunity to instruct courses at the university.

This dissertation would not have been possible without my family and role models encouraging me to lead the path of a scientist and engineer, and by giving me motivational support and the occasional nutritional home-cooked meal. Finally, I would like to thank my friends, colleagues, and lab mates for inspiration, mutual support, and insightful discussions, and the members of StackExchange community for their invaluable assistance in C++, MATLAB, and LaTeX source code related details.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

CHAPTER Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

NOMENCLATURE

$\mathbb{A}$	Threshold Ratio of Distances Between Abscissas, Equation (60)
A	Area [L^2]
$\boldsymbol{b}$	Conditional-Splitting Inversion Sequence $[d_1, ..., d_{\mathcal{X}}]$, Equation (86)
$\mathbb{B}$	Maximum Number of Multidimensional Nodes, Defining the Resolution of $\mathbb{N}$
B	Adaptive (1...β) Number of Nodes In Single Dimension
C_D	Drag Coefficient, Equation (5) $[-]$
c	Speed of Sound In a Fluid, $\propto \rho_f^{-0.5}$, Equation (14) [LT^{-1}]
C	Collision Term of the PDF Transport Equation (37)
$\mathbb{C}$	Collision Term of the Moment Transport Equation (122)
CFL	Numerical Stability Condition, Equation (109)
d	Spatial Dimension Index $(1, ..., D)$
D	Order of the Spatial Dimension, $1 \leq D \leq 3$
$\mathbb{D}$	Drag Source Term, Equation (49)
E	Scientific Exponential Notation, $XE^Y \equiv X \times 10^Y$
$\mathbb{E}$	Exponent Index Function for Moment Construction, Table 9
e	Natural Log Base, ≈ 2.71828
e_w	The Coefficient of Restitution, Equation (128)
$\mathbb{F}$	Second Order Tensor for Inelastic Collisions, Equation (120)
f	Probability Density Function (PDF), Equation (31)
$\boldsymbol{F}$	Force Per Unit Volume on Particles [MLT^{-2}]
$\boldsymbol{g}$	Gravitational Acceleration Vector [LT^{-2}]
$\boldsymbol{G}$	Net Moment Convection Function at a Single Cell Face, Equation (98)
$\boldsymbol{H}$	Moment Convection Function at a Single Cell Face, Equation (96)
h	Test Function for Gaussian Quadrature, Equation (93)

$\boldsymbol{I}$	Identity Matrix
$\mathbb{I}$	ID Matrix Indicating Occupied Nodes of $\mathbb{N}(t, \boldsymbol{x})$
K	Total Quantity of Moments in the Vector Set $\boldsymbol{M}$, Equation (56)
Kn	Knudsen Number, Equation (14)
L	Macroscopic Length Scale, Equation (12) [L]
L_p	Particle Diameter [L]
l	Multiplicity of d'th Abscissa, Equation (45)
L_c	Mean Free Path, Equation (14) [L]
L_V	Edge Length of a Finite Cell Volume [L]
m_p	Mass of a Single Particle, Equation (7) [M]
$M^{(\gamma)}_{l_1 l_2 l_3}$	3-D Velocity Moment $\int v_1^{l_1} v_2^{l_2} v_3^{l_3} f$, Equation (45)
$\boldsymbol{M}$	Optimal Moment Set of Length K
${M_0}^*$	Maximum Close-Packed Volume Fraction, Equation (18)
Ma	Mach Number, Equation (14)
$\mathcal{N}$	Maximum: Quantity, Sample Count, or Time-Step Index
n	Time Step Index
$\boldsymbol{N}$	Node Vector (Row of Matrix $\mathbb{N}$), Equation (88)
$\mathbb{N}$	$\equiv \{\hat{\boldsymbol{v}}_1, ..., \hat{\boldsymbol{v}}_D, \hat{\boldsymbol{\xi}}_1 ..., \boldsymbol{\xi}_Q, \boldsymbol{\phi}\} \equiv \{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\phi}\}$, Column Matrix of Abscissas and Single Weight, Equation (39) and (54)
$\mathbb{P}$	2-Point Abscissa Proximity Ratio (Vector), Equation (61)
P	CQMOM Flux Path Total Permutation Count, Table 7
p	CQMOM Flux Path, Table 7
Pr	Prandtl Number, Equation (16)
q	Quantity of Particles Per Unit Volume (Number Density) [L^{-3}]
Q	Length of $\boldsymbol{\xi}$: Dimension of the Internal State Space Excluding Particle Velocities
$\mathbb{R}$	Real Numbers State Space

r	Runge-Kutta Internal Iteration Index, $1, ..., R$, Table 8
R	Runge-Kutta Order, Table 8
Re	Reynolds Number, $= \rho \mathbb{V} L/\mu$, Equation (6)
RMS	Root-Mean-Square of Velocity Fluctuations, $\sqrt{\overline{v'^2}}$, Equation (70)
S_c	Collision Intensity, Equation (23)
s	Volume Fraction Ratio to Maximum Close-Packed Factor $M_0/{M_0}^*$, Equation (18)
s_0	Particle Radial Distribution Function, Equation (18)
St	Stokes Number, Equation (12)
T	Thermal Temperature [$^\circ$K]
t	Time [T]
$\boldsymbol{U}$	x-Velocity Abscissas Vector, Length $\mathbb{B}$, Equation (54) [LT^{-1}]
$\boldsymbol{u}$	Fluid Velocity [LT^{-1}]
$\mathbb{U}$	Characteristic Fluid Velocity Scale [LT^{-1}]
$\boldsymbol{V}$	y-Velocity Abscissas Vector, Length $\mathbb{B}$, Equation (54) [LT^{-1}]
$\hat{\boldsymbol{v}}$	Matrix of Velocity Abscissas, $\hat{\boldsymbol{v}}_{d,\alpha} \equiv \{\boldsymbol{U}^T, \boldsymbol{V}^T, \boldsymbol{W}^T\}$, With Row α and Column d for Node and Dimension Respectively, Equation (39) [LT^{-1}]
$\boldsymbol{v}$	$\equiv \{v_1, v_2, v_3\}$, Particle Velocity Vector of All D Dimensions [LT^{-1}]
$\mathbb{V}$	Characteristic Particle Velocity Scale, Equation (109) [LT^{-1}]
$\boldsymbol{W}$	z-Velocity Abscissas Vector, Length $\mathbb{B}$, Equation (54) [LT^{-1}]
$\mathbb{W}$	Vector of Weight Threshold Values, Equations (57) and (59)
Δx	Cell Width, Distance Between Both Faces, $x_{+\frac{1}{2}} - x_{-\frac{1}{2}}$, Equation (103)
$\mathcal{X}$	$\equiv D + Q$, Total Dimension of PDF Internal State Space, Table 11
x	1st Spatial Dimension, x_1 (Streamwise Coordinate) [L]
$\boldsymbol{x}$	Spatial Coordinate Vector, $\{x, y, z\}$ [L]
y	2nd Spatial Dimension, x_2 (Wall-normal Coordinate) [L]
z	3rd Spatial Dimension, x_3 (Spanwise Coordinate) [L]

Greek:

α	Quadrature Node Index, $1, ..., \beta$
β	Maximum Quadrature Node Count in Single Dimension
δ	Dirac Delta Function
ε	Relaxation Factor
η	Spatial Volume [L^3]
η_p	Single Particle Volume, Equation (34) [L^3]
$\mathbf{\Lambda}$	Vector of Optimal Equilibrium Moment Set, Equation (122)
γ	$\equiv l_1 + l_2 + l_3 + ...$, Moment Order, Sum of Abscissa Multiplicities/Exponents. Equation (45)
κ	Time Coefficient, Equation (127)
λ	$\equiv \frac{\Delta t}{\Delta x}$, Ratio of Time Step to Cell Spacing, Equation (110) [TL^{-1}]
μ	Dynamic Viscosity [ML^{-1}T^{-1}]
ν	Kinematic Viscosity [L^{2}T^{-1}]
ω	Elasticity $(1 + e_w)/2$, Equation (16)
Ω	State Space for Variable Determined by Subscript, Equation (30)
ϕ	Volume Fraction, “Weight” in PDF [$-$]
ψ	Total Particle Volume Contained Within a Finite Volume Element [L^3]
ρ	Density [ML^{-3}]
σ	Standard Deviation of Distribution Function (PDF)
Θ	Granular Temperature, Equation (65), [L^{2}T^{-2}]
τ	Response Time [T]
$\boldsymbol{\theta}$	Temperature Covariance Matrix, Equation (66)
Υ	Inter-Particle Collision Model Constant $0 < \Upsilon \leq 3/2$, Equation (16)
$\boldsymbol{\xi}$	Vector of Internal States in f Excluding Velocity, Equation (30)
$\boldsymbol{\zeta}$	$\equiv \{\boldsymbol{v}, \boldsymbol{\xi}\}$, Vector of Length $\mathcal{X}$ for all Internal States in f

Other:

$\bar{*}$	(Overbar) Time-Averaged Quantity
$\langle * \rangle$	(Angle Brackets) Cell-Averaged Quantity
$\dot{*}$	(Over Dot) Time Derivative
$\hat{*}$	(Hat) Abscissa; Discrete Value of a Continuous Variable
$\boldsymbol{*}$	(Bold) Vector or Matrix
$\lvert * \rvert$	Absolute Value or Vector Magnitude
$+=$	Addition Assignment Operator
$-=$	Subtraction Assignment Operator
$/=$	Division Assignment Operator
$:=$	Imperative Assignment Operator
$\binom{*}{*}$	Binomial Operator, Equation (114)
$\{*, *, ...\}$	Elements of a Vector
$*'$	(Prime) Unsteady Fluctuation, Equation (71)
$:$	Colon Operator Signifies Integer Range, $i : j \equiv \{i, i+1, ..., j-1, j\}$

Subscript/Superscript:

a	Spatial Coordinate Index for Finite Cell Discretization in Dimension d
c	Collision Attribute
eq	Related to the Maxwell-Boltzmann Equilibrium Distribution, Equation (122)
f	Fluid Phase Attribute
g	Grid/Mesh Attribute
ℓ	Left Side of a Cell Face, Equation (96)
m	Moment Index $0...K-1$, Table 9
p	Particle Phase (Drag) Attribute
r	Right Side of a Cell Face, Equation (97)
s	Slip Velocity, of Particle Relative to Carrier Fluid, Equation (4)

τ	Related to Friction (Velocity), Equation (1)
T	Vector or Matrix Transpose

I. INTRODUCTION

For many decades now, computational fluid dynamics (CFD) has been offering alternative insights to experiments that are financially expensive or technologically/physically prohibitive. Numerical models are capable of verifying and reproducing observations of complex physical phenomena and ultimately have advanced predictive capabilities in a wide range of science and engineering applications. However, the limitations of computing technology at the present necessitate the usage of simplifying assumptions to maintain practical computational costs. Furthermore, numerical discretization techniques are needed to convert the complexity of "continuous" physical systems into manageable mathematical constructs. Consequently, various techniques often lead to multiple solution methods and their applicability and versatility may be restricted to certain ranges of problems, but simultaneously optimized for that unique limited range. When multiple models are capable of describing the same physical system, such as a multiphase flow, they should (hopefully) arrive at the same results, but selecting the appropriate model depends on its accuracy vs. computational costs.

Essentially there are two ways to model a fluid: as a collection of Lagrangian particles or as an indefinitely divisible Eulerian continuum.[5] For example, the Eulerian formulation of Navier-Stokes equations have shown excellent predictive capabilities of hydrodynamic fluids at equilibrium (such as Newtonian fluids) where the concepts of viscosity and incompressibility are generally valid simplifying assumptions. Hydrodynamic fluid behavior is dominated by chaotic inter-molecular collisions, which need not be fully-resolved in order to make useful predictions. Alternately, Lagrangian discrete particle simulation (DPS) models provide fully-resolved solutions that track individual particles undergoing free-body motion. Lagrangian methods have played a major role in modeling sparse particle dispersions; however, DPS will remain computationally prohibitive for very dense particle populations encountered in many modern engineering applications for the foreseeable future.

Both of these models are optimal in their own regimes, but there still remains a gap in CFD capabilities for models that are accurate and efficient at moderately dense non-hydrodynamic "fluids". Such phases are *not* strictly dominated by inter-particle collisions, but are dense enough that a fully-resolved Lagrangian model is extremely computationally expensive. Therefore, the focus of the current research is to implement a novel Eulerian modeling strategy that is applicable for both dilute single-phase fluids and dispersed phases of multiphase flows. The main challenges are that non-hydrodynamic fluids are generally highly compressible, lack a straightforward counterpart to viscosity,[6] and experience prevalent self-diffusion.

"Particles" comprising a dilute phase may consist of either molecules or quasi-particle bodies such as granular solids, liquid droplets, or even gas bubbles. Therefore, the term "particle" throughout this work pertains to the smallest bodies/entities considered in the simulation. Both molecules within a single-phase fluid and small quasi-particle bodies exhibit similar behavior and share concepts in Kinetic Theory. One main difference is that during collisions, molecules are typically perfectly elastic (hard spheres) while macroscopic granular particles are inelastic/dissipative[7] and may undergo other macroscopic phenomenon such as aggregation and breakage.

Multiphase flows are commonly encountered and are composed of a mixture of two (or more) phases – typically a hydrodynamic fluid (gas or a liquid) carrier flow with a second dispersed granular phase of (solid, gas, or liquid) "particles." The dispersed phase exhibits a distinct surface interface that distinguishes the particle bodies from the carrier flow, e.g. dust particles in air, air bubbles in water, oil droplets in water, etc. By definition, the dispersed phase is comprised of macroscopic particles that are not materially connected, while the carrier fluid occupies the gaps in-between the particles. It is common to describe non-hydrodynamic fluids or dispersed phases as rarefied, dilute, and/or sparse. The ability to implement an *Eulerian* model which can realize the diverse *micro*scale interactions of molecules/particles while still offering predictive capabilities of *macro*scopic flow structures, is a highly attractive engineering goal for modeling single-phase and multiphase flows alike.

Table 1. Categories of Multiphase Flows[4]

Gas-liquid flows	Gas-droplet flows
	Separated flows
	Bubbly flows
Gas-solid flows	Gas-particle flows
	Pneumatic transport
	Fluidized beds
Liquid-solid flows	Slurry flows
	Hydrotransport
	Sediment transport
Liquid-liquid flows	Immiscible flows
Three-phase flows	Bubbles in a slurry flow
	Droplets and particles in a gaseous flow

As described by Table 1, many different two and three phase flows are prevalent in engineering applications. Cases involving rarefied single phase fluids or dispersed phases such as aerosols/droplets/sprays,[8–12] bubbles,[13,14] precipitates,[15,16] or dust/powders/soot[17–23] exhibit many real-world physical phenomena as listed in Table 2.

Table 2. Select Categories of Physical Phenomena

Collisions	Inter-particle and (in)elastic
Nucleation/cavitation	Particle production
Growth or evaporation[12]	Diameter change
Coalescence/Aggregation	Particle combining
Breakage[24]	Particle splitting
Body forces	Gravity, magnetic, etc.
Fluid drag force	one-way fluid coupling
High mass loading[25,26]	two-way fluid coupling

Predicting these desired fluid-particle properties is beneficial in a variety of industrialized processes, technologies, and scientific research fields, of which some are listed in Table 3. Very small scales are also applicable, such as modeling blood cells suspended in plasma or the industrial synthesis of nanoparticles.[27] The ability to employ accurate simulations of these physical phenomena may reduce engineering research and design costs as well as the overall design-time. Machinery developed solely to carry-out

Table 3. Example Applications and Technologies

Propulsion systems:	Thrusters[28] or rockets
Rarefied gas aerodynamics:	High altitude flight or low orbit
Astrophysics:	Interstellar clouds,[7] galaxies
Dust clouds:[21]	Volcanoes, Rotorcraft brownout
Combustion:[12]	Atomization or gasification
Pneumatic transport:	Evacuated tubes
Material processing:	Spray drying
Fluidized beds:[29]	Reactors or heat transfer
Pollution (exhaust) control:	Scrubbers, cleaners, filtration systems

these processes related to rarefied gases or particle-laden flows are typically large-scale where small efficiency improvements may induce large cost savings. Particle-laden (gas-solid) flows are of particular interest in the present work and will be focused upon almost exclusively. Therefore, the fluid-particle drag forces and inter-particle collision phenomena will be described in great detail. The family of novel Eulerian formulations known as Quadrature-Based Moment Methods (QBMM) are explored in the current work, and compared to the legacy Lagrangian formulation, DPS. Firstly, the Quadrature Method of Moments (QMOM)[8,30–32] will be introduced as the foundation for a more sophisticated, multi-dimensional and multivariate formulation known as Conditional Quadrature Method of Moments (CQMOM).

II. MODELING - BACKGROUND

Before describing the QBMM model, a set of constants, non-dimensional values, and concepts must be introduced. These constants describe single phase and/or multiphase flows, in Eulerian and/or Lagrangian frameworks, and can outline the limitations imposed by the different model assumptions.

A. Reynolds Number

The Reynolds number is the ratio of momentum to viscous effects that describes whether a fluid flow is laminar or turbulent. For boundary layers and channel flows, the Reynolds number of the carrier flow is typically defined using two different length scales: either the boundary layer thickness (L_δ) or the channel half height ($\frac{L_y}{2}$) respectively. Additionally, the velocities will often be non-dimensionalized using the friction velocity, $u_\tau = \sqrt{\nu \frac{\partial u_1}{\partial y}}$ at the wall.[18,19] This also suggests defining a new Reynolds number (Re_τ) using the friction velocity,

$$\mathrm{Re}_\tau = \frac{u_\tau L}{\nu_f} \tag{1}$$

It is common to assume that $u_\tau = 1$, $L = \frac{L_y}{2} = 1$, and then Re_τ is chosen and typically ranges from 150-700 in channels flows from previous work. These values play an important role in non-dimensionalizing the velocity profile, where the wall coordinate, y^+, is computed by one of two methods,

$$y^+ = \frac{y u_\tau}{\nu_f} = \frac{y \mathrm{Re}_\tau}{L} \tag{2}$$

and the velocity is normalized by the friction velocity too,

$$u^+ = \frac{u}{u_\tau} \tag{3}$$

Now the non-dimensional y^+ and u^+ values may be conveniently used directly in the numerical simulation. Notice that the convenient choice of $u_\tau = 1$ ensures that $u^+ = u$.

B. Drag Force

The dispersed particles (in both DPS and QBMM models) are coupled to the carrier flow through a particle-fluid drag that allows the particle acceleration to be computed. Currently the drag is only one-way coupled because the particles do not affect the fluid through mass loading. The derivation for the particle drag begins by assuming the particles are spherical bodies traveling through a viscous fluid, and is simplified with the following approximation,[4]

$$\boldsymbol{F} = \frac{1}{2}\rho_f A C_D |\boldsymbol{u} - \boldsymbol{v}|(\boldsymbol{u} - \boldsymbol{v}) \tag{4}$$

where ρ_f is the carrier fluid density, $A = \frac{1}{4}\pi {L_p}^2$ is the projected frontal area of a sphere (circular area) of particle diameter L_p, and $\boldsymbol{u} - \boldsymbol{v} \equiv \boldsymbol{v}_s$ is the relative slip velocity between the fluid and particle. The drag coefficient C_D is commonly modeled by,

$$C_D = \begin{cases} \frac{24}{\mathrm{Re}_p}\left(1 + 0.15\mathrm{Re}_p^{0.687}\right), & \mathrm{Re}_p < 1000 \\ 0.44, & \mathrm{Re}_p \geq 1000 \end{cases} \tag{5}$$

where the dispersed particle Reynolds number, Re_p, is defined by,

$$\mathrm{Re}_p \equiv \frac{\rho_f}{\mu_f}|\boldsymbol{v}_s|L_p = \frac{|\boldsymbol{v}_s|L_p}{\nu_f} \tag{6}$$

and $\nu_f \equiv \mu_f/\rho_f$ is the kinematic viscosity of the carrier fluid. Note that Re_p is a *scalar* computed from the relative velocity *vector* magnitude, therefore, only one C_D exists for each particle.

Next, the concept of the particle response time due to drag and inertia is introduced. If the single force acting the particle is a fluid drag force, then the acceleration ($\frac{dv}{dt}$) is determined from Equation (4) and Newton's second law of motion by[4]

$$m_p \frac{dv}{dt} = \frac{C_D}{2}\frac{\pi {L_p}^2}{4}\rho_f(\boldsymbol{v}_s)|\boldsymbol{v}_s| \tag{7}$$

where m_p is the mass of a single particle. Substituting Re_p and dividing Equation (7) by spherical particle mass $m_p = \frac{1}{6}\rho_p \pi {L_p}^3$ produces,

$$\frac{d\boldsymbol{v}}{dt} = \frac{18\mu_f}{\rho_p {L_p}^2}\frac{C_D \mathrm{Re}_p}{24}(\boldsymbol{u} - \boldsymbol{v}) \tag{8}$$

With the assumption of Stokes drag at low particle Reynolds numbers, the term $\frac{C_D \mathrm{Re}_p}{24} \approx 1$, and all the terms may be condensed into a single value τ_p, which is the *particle drag response time* related directly to the Stokes number, St. This value depends on the particle inertia and the influence that the fluid imposes on the particle through the Stokes drag force. The particle response time encompasses all variables from Equation (8),

$$\frac{d\boldsymbol{v}}{dt} = \frac{1}{\tau_p}(\boldsymbol{u} - \boldsymbol{v}) \tag{9}$$

Therefore,

$$\tau_p = \frac{4\rho_p L_p}{3\rho_f C_D |\boldsymbol{v}_s|} \approx \frac{\rho_p {L_p}^2}{18\rho_f \nu_f} \tag{10}$$

Note that the right-most term of Equation (10) is an approximation arising from the Stokes drag assumption (small Re$_p$). If τ_p is using the Stokes flow assumption, then it is a scalar value with no dependency on velocity, and by further assuming that $\boldsymbol{u}$ is constant, then Equation (9) becomes a first-order ODE and the solution to this ODE with a particle starting at rest $v(t_0) = 0$ is,[4]

$$\boldsymbol{v} = \boldsymbol{u}(1 - e^{-t/\tau_p}) \tag{11}$$

This ODE assumption is useful in QBMM for applying source terms in a PDE using operator splitting techniques. Additionally, τ_p is the time required for the particle to reach 63% ($\frac{e-1}{e}$) of the free stream velocity, $\boldsymbol{u}$. In other words, τ_p characterizes the lag time inherent in inertial particles based on the influence of drag from the carrier fluid.

C. Stokes Number

Following from the derivation of particle drag in the previous section, another crucial constant that requires introduction is the Stokes number, St. This non-dimensional value characterizes the relationship between the carrier and dispersed phases; more precisely it represents the ratio of dispersed particle and carrier fluid response times,

$$\text{St} \equiv \frac{\text{Particle response time}}{\text{Fluid response time}} = \frac{\tau_p}{\tau_f} = \frac{\tau_p \mathbb{U}}{L_f} \tag{12}$$

where $\tau_f \equiv \frac{L_f}{\mathbb{U}}$ is the characteristic fluid response time, $\mathbb{U}$ is a characteristic fluid velocity (such as the maximum fluid velocity), and L_f is a characteristic fluid length scale (such as a channel half-span or a vortex diameter). The Stokes number is critical because it describes the responsiveness of dispersed particles to follow the carrier fluid behavior, and it determines the regime for which the particle velocities may deviate from the fluid velocities (St ≥ 1). This typically occurs when the particle density is larger than the fluid density, $\rho_p \gg \rho_f$. Conversely, when the particle motion does indeed follow the carrier fluid very closely (St $\ll 1$) then the particle and fluid velocities are the same, $\boldsymbol{v} = \boldsymbol{u}$.

D. Knudsen Number

The non-dimensional Knudsen number (Kn) represents the importance of particle-particle collisions relative to free transport. It is therefore proportional to the particle collision time (Kn $\propto \tau_c$) and may be used to characterize fluid phases and dispersed phases alike. In other words, the “particles” in general may pertain to fluid molecules separated only by vacuum or to a dispersed phase of *granular* particles which could be contained within a carrier medium.

The mean free path, L_c, is the average distance a particle travels before colliding with another. To determine the frequency of collisions ($1/\tau_c$), consider a straightly moving particle which will collide with any other particle’s center that lies closer than the diameter of the particle (L_p). This creates a cylinder with radius L_p (the circular area is the collisional cross-section) along the motion axis and its projected cylindrical volume is $\pi {L_p}^2 v_c$.[4,7] The cylinder grows in length at the rate of $v_c = \sqrt{8\Theta/\pi}$, which is the mean relative velocity to the other particles (not to be confused with the speed of sound, c) and depends on the kinetic (thermal) energy or phase temperature, Θ. The collision frequency is then estimated by multiplying the projected cylindrical volume by the particle number density, q,

$$\tau_c = \frac{1}{\text{Collision frequency}} = \frac{1}{q\pi {L_p}^2 v_c} \tag{13}$$

For a granular fluid phase, Θ is appropriately termed the *granular temperature*,[1,7,33] from Kinetic Theory for Granular Flows (KTGF). An analogy can be made to the absolute temperature, T, for molecular fluids in kinetic theory: $\Theta \;\Leftrightarrow\; k_B T/m_p$, where k_B is the Boltzmann constant. When the ratio $k_B/m_p \;\rightarrow\; 1$ and becomes unitless[33] then the "temperature" is no longer in Kelvin, but has units of velocity squared, [L^{2}T^{-2}], and is favorable for describing granular phase velocity statistics. More usage of this value is described later in Section §*F. Temperature* and Section §*1. Inter-Particle Collisions*.

1. Fluid Phase

The molecular fluid phase Knudsen number, Kn$_f$, describes the ratio of the molecular mean free path to a characteristic length scale,[34] L. Note that the molecular mean free path in a gas is typically one or two orders of magnitude larger than the molecular diameter, with molecular velocities on the order of hundreds of meters per second. Liquids have a significantly smaller mean free path, usually on the order of the molecular diameter, which often makes them incompressible.[5] Therefore, an appropriate characteristic length scale (L) for Kn_f might be the length of a macroscopic body immersed in the flow field, which could even be the diameter of a dispersed particle[4] ($L = L_p$).

$$\mathrm{Kn}_f \equiv \frac{L_c}{L} \approx \frac{\mu_f}{\rho_f c L} = \frac{\mathrm{Ma}}{\mathrm{Re}} \tag{14}$$

As Kn $\rightarrow$ 0 the regime approaches the hydrodynamic limit, i.e. a "true fluid", that is appropriately modeled by a single-valued Eulerian model due to its dense particle population dominated by collisions, and the particle speed distribution is Maxwellian.[35] Conversely, Kn $\rightarrow \infty$ is the "free molecular" rarefied/dilute gas, collisionless, *Knudsen limit* that Lagrangian methods are well-suited for because the particle velocity distribution can become very sophisticated, i.e. difficult to describe with a continuous function or polynomial, and so a piecewise approach is appropriate. However, the transition region of finite Knudsen number, Kn $\approx O(1)$, between those limits is of particular interest in the current work. Four unique regimes are classified for the Knudsen number based on the characteristics of the Mach number in relation to the Reynolds number as listed in Table 4.

Table 4. Knudsen Number Regimes[4]

Regime Name	Knudsen range	Mach and Reynolds relation
Continuum	$\mathrm{Kn} < 10^{-3}$	$\mathrm{Ma} < 0.01\sqrt{\mathrm{Re}}$
Slip flow	$10^{-3} < \mathrm{Kn} < 0.25$	$0.01\sqrt{\mathrm{Re}} < \mathrm{Ma} < 0.1\sqrt{\mathrm{Re}}$
Transitional flow	$0.25 < \mathrm{Kn} < 10$	$0.1\sqrt{\mathrm{Re}} < \mathrm{Ma} < 3\mathrm{Re}$
Free molecular flow	$\mathrm{Kn} > 10$	$\mathrm{Ma} > 3\mathrm{Re}$

2. *Dispersed Phase*

Alternatively, a dispersed phase Knudsen number, Kn_p, relies on a larger length scale related to fluid structures or flow geometry. For low Mach numbers ($\mathrm{Ma} < 1$), the dispersed-phase Knudsen number is approximated as,[1]

$$\mathrm{Kn}_p \approx \frac{\tau_c \sqrt{\Theta}}{L_f} = \frac{\sqrt{\pi} L_p}{12 s_0 M_0 L_f} \tag{15}$$

where $\sqrt{\Theta}$ is taken to be the average particle velocity, and by rearranging this equation to solve for the collision time yields,[1,36]

$$\tau_c = \frac{\Upsilon L_p}{12 s_0 M_0} \sqrt{\frac{\pi}{\Theta}} \tag{16}$$

where $0 < \Upsilon \leq 3/2$ is a model constant that corresponds to $1/\mathrm{Pr}$ in the elastic limit ($\omega = 1$) known as the ES-BGK model.[1] For the regular BGK model, simply $\Upsilon = 1$ is adequate. Let M_0 be the particle volume fraction specifically for monodisperse spherical particles,

$$M_0 = \frac{\pi {L_p}^3 \mathcal{N}}{6\eta} \tag{17}$$

where $\mathcal{N}$ is the number of particles contained within a finite volume η. The volume fraction will not typically be able to reach unity and is limited ($0 \leq M_0 \leq {M_0}^*$) by the maximum packing factor ${M_0}^*$ for the given shape of particles.

Figure 1 plots the volume fraction of Equation (17) and leads to an important realization about the three coupled variables that determine the collision strength: volume

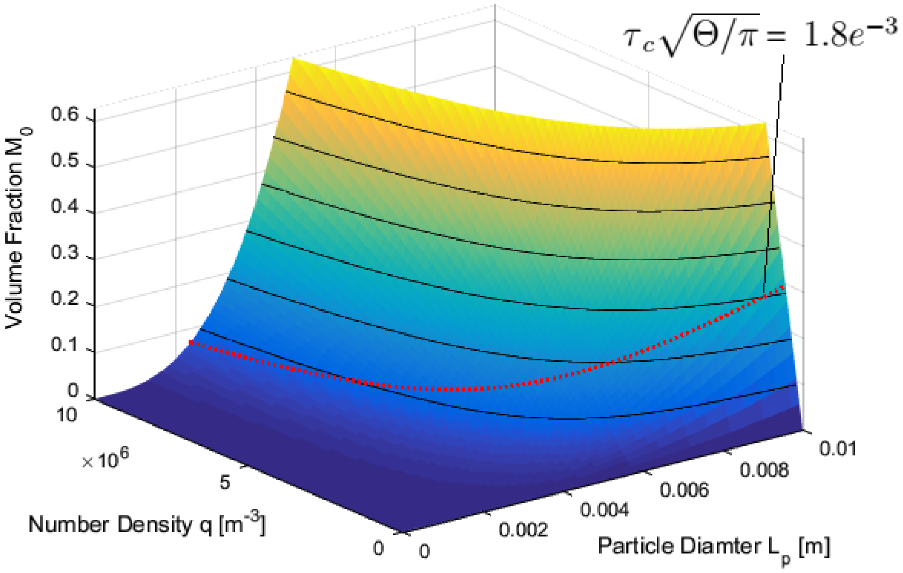


Figure 1. Volume Fraction (M_0) of Spherical Monodisperse Particles as a Function of Number Density (q) and Diameter (L_p)

fraction M_0, number density $q = \mathcal{N}/\eta$, and particle diameter L_p. Equation (17) provides conversion between all three variables, where any two of them can be used to compute the 3rd, and thus any two are sufficient to determine the collision effects. The red dotted curve in Figure 1 represents a constant quantity, $\frac{12\tau_c}{\Upsilon}\sqrt{\frac{\Theta}{\pi}}$, in Equation (16), and explains that by adjusting the particle diameter and volume fraction in conjunction, the collision frequency remains unchanged. In a DPS model, typically the particle diameter and particle number density are provided for closure; however, the volume fraction and particle number density provide an alternative description that may be more applicable to an Eulerian continuum approach.

Collision effects are exemplified at very high packing factors, $M_0 \to {M_0}^*$ and this is handled by the radial distribution model s_0 term[1,37]

$$s_0(s) = \frac{2 - s}{2(1 - s)^3} \tag{18}$$

where $s = M_0/{M_0}^*$ is a packing factor ratio and ${M_0}^* \approx 0.63$ is a good approximation for the maximum volume fraction of close-packed spheres.[1] A nearly equivalent radial distribution function is proposed as,[37–39]

$$s_0 = \frac{1}{1 - M_0} + \frac{3M_0}{2(1 - M_0)^2} + \frac{(M_0)^2}{2(1 - M_0)^3} \tag{19}$$

These collision models are only suited for particles of similar properties (monodisperse) and consequently, more complex (polydisperse) inter-particle collisions may require additional modeling considerations, especially between particles of varying mass and/or diameter. Equation (16) appears in the collision model of the current work and its implementation is described in §*1. Inter-Particle Collisions*.

3. Collision Intensity

It is important to realize that the collision frequency is better suited for numerical purposes because it approaches zero, $1/\tau_c \to 0$, for a collisionless state, which arises commonly. Additionally, high frequencies remain far from causing precision overflow issues. The same does not hold true for τ_c, which easily becomes infinite when no collisions are present, and runs the risk of division by zero or other unintended and ill-defined numerical side-effects. This motivates the usage of $1/\tau_c$ exclusively in the numerical approach.

However, a problem becomes apparent when attempting to perform averages on these collision times or frequencies. Let $\langle \tau_c \rangle \equiv \sum_{i=1}^{\mathcal{N}} \tau_{c,i}$ be a spatial average over $\mathcal{N}$ total cells, then it should be obvious that,

$$\langle \tau_c \rangle \neq \left\langle \frac{1}{\tau_c} \right\rangle^{-1} \tag{20}$$

In other words, the values are not linearly related and will produce different statistical results depending on whether time or frequencies are employed. One solution that ensures consistency between *both* approaches is to first take the natural logarithm (base e),

$$-\langle \log \tau_c \rangle = \left\langle \log \frac{1}{\tau_c} \right\rangle = \langle \tilde{S_c} \rangle \tag{21}$$

which guarantees the same linear result, $\langle \tilde{S_c} \rangle$, regardless of whether time or frequency values are used. This is a desired trait for performing statistical summations and the value $e^{\langle \tilde{S_c} \rangle}$ is bounded by the two non-negative extremes,

$$\left\langle \frac{1}{\tau_c} \right\rangle^{-1} \leq e^{\langle \tilde{S_c} \rangle} \leq \langle \tau_c \rangle \tag{22}$$

However, both time and frequency approaches still produce infinite values when $\tau_c \to \infty$ or $1/\tau_c \to 0$, because $-\log(\infty) = \log(0) = -\infty$. Therefore, an alternative approach is needed that can provide a meaningful statistical descriptor of the collision intensity without destroying the entire average when any local cell is in a collisionless state (or close to it, $\tau_c \gg 1$).

Assuming that $\tilde{S}_c$ indicates the magnitude or intensity of collisions in some manner, let the actual *collision intensity* be defined as

$$S_c(\tau_c) = \log\left(1 + \frac{1}{\tau_c}\right) \tag{23}$$

where the frequency appears directly and an additional 1 has been added to ensure that the collisionless state is consistent with zero intensity ($S_c = 0$), and that the intensity always remains positive, $S_c \geq 0$. This intensity is only slightly different from the simple logarithm value, $\tilde{S}_c$, with the greatest deviation occurring in the limit $\tau_c \to \infty$,

$$\lim_{\tau_c \to 0} S_c \to \tilde{S}_c \tag{24}$$

but guarantees a numerically well-defined value that is readily deployable into a numerical code. The curve of Equation (23) is plotted in Figure 2 and it is straightforward to approximate the statistical collision frequency using the spatially averaged collision intensity,

$$\frac{1}{\langle \tau_c \rangle} \approx e^{\langle S_c \rangle} - 1 \tag{25}$$

Whenever a spatially averaged collision frequency or time, $\langle \tau_c \rangle$, is mentioned in the current work's results, it has been computed using Equation (25). To summarize, the values of $\tilde{S}_c$ are more exact than S_c, but ill-defined at very low frequencies, such that

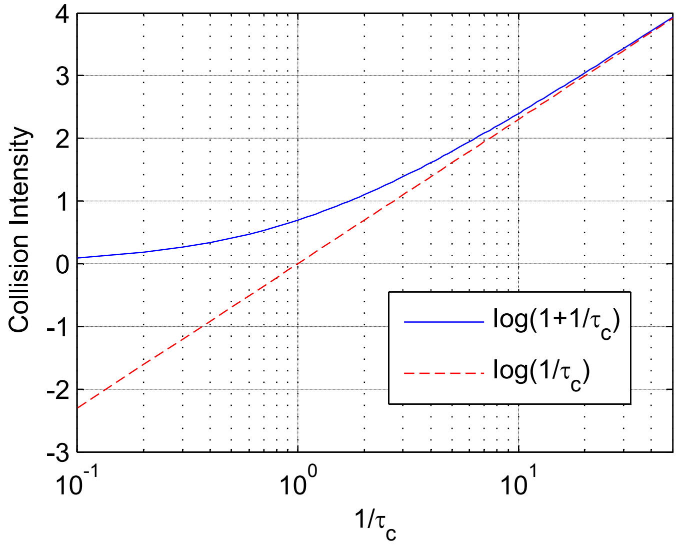


Figure 2. Collision Intensity Function, $S_c(\tau_c)$ Shown as Solid Line. Ensures Very Low Frequency Collisions Approach Zero Intensity Instead of $-\infty$

$\langle\tilde{S}_c\rangle \;\rightarrow\; -\infty$ destroys the average if even one cell in the ensemble experiences a collisionless state locally. The approximated collision intensity is not exact at low frequencies, $\langle S_c\rangle \approx \langle\tilde{S}_c\rangle$, but offers an alternative representation that is highly accurate at high collision frequencies and provides a useful value for comparisons between statistical analyses.

Lastly, as a side note, the function $\log(1+X)$ has a known numerical deficiency for very small X where $1+X$ is rounded to exactly 1. This is probably not a critical issue, but many libraries offer a $\mathrm{log1p}(X)$ function (or similar) to handle this by essentially assuming that

$$\mathrm{log1p}(X) \equiv \log(1+X) \quad \approx X\Big|_{X\approx 0} \tag{26}$$

for small X.

4. *Dispersed Flow*

The average distance between dispersed particles determines whether the flow is dilute or dense.[4] A dilute dispersed particle phase is dominated by drag and lift forces from the carrier phase, while a dense medium is dominated by inter-particle collisions. Comparing the particle drag response time with the collision time produces a ratio,

$$\frac{\tau_p}{\tau_c} \Rightarrow \frac{q\pi\rho_p {L_p}^4 v_c}{18\mu_f} \Rightarrow \frac{\langle\rho_p\rangle L_p v_c}{3\mu_f} \tag{27}$$

where the bulk density $\langle\rho_p\rangle = (\rho_p\pi\frac{{L_p}^3}{6})q$ is the product of the mass of a single spherical particle and the number density q. A dilute flow is characterized by a ratio less than unity in Equation (27).[4] The quantity $\frac{\tau_p}{\tau_c}$ is useful for plotting to determine which regions of the flow are dominated by drag or collisions. Note that τ_p is typically a constant, and so the field is simply a scaled value of $1/\tau_c$ (which does vary significantly locally).

E. Phase Properties

The previous background sections described many constants (all non-dimensional) that are used throughout this work and are summarized in Table 5. To summarize, two primary approaches – Lagrangian and Eulerian – are available for solving phases,[5,40]

Table 5. Phase Constants

Name	Abbr.	Description
Reynolds number	Re	Laminar vs. turbulent carrier flow
Stokes number	St	Fluid drag vs. particle inertia dominance
Knudsen number	Kn	Particle-particle collisions vs. free transport
Mach number	Ma	Convective vs. diffusive dominant transport
Prandtl number	Pr	Momentum vs. thermal diffusion rate

and both approaches offer predictive capabilities optimized for certain Knudsen number regimes. The Lagrangian approach best describes sparsely dispersed phases (Kn $\gg$ 1), and the N-S equations *only* describe hydrodynamic fluids (Kn < 0.1), but the no-slip condition is actually more strict (Kn < 0.001).[5] The Boltzmann equation offers solutions for *all* Knudsen numbers, and so too do the Quadrature-Based Moment Methods (QBMM). However, QBMM offers a computationally less expensive approach that bridges the modeling gap between the two Kn extremes, capable of modeling fluids with finite Knudsen number, Kn $\approx O(1)$. Figure 3 depicts the entire range of Knudsen numbers (similar to Table 4) and summarizes which approaches are most suitable per regime.

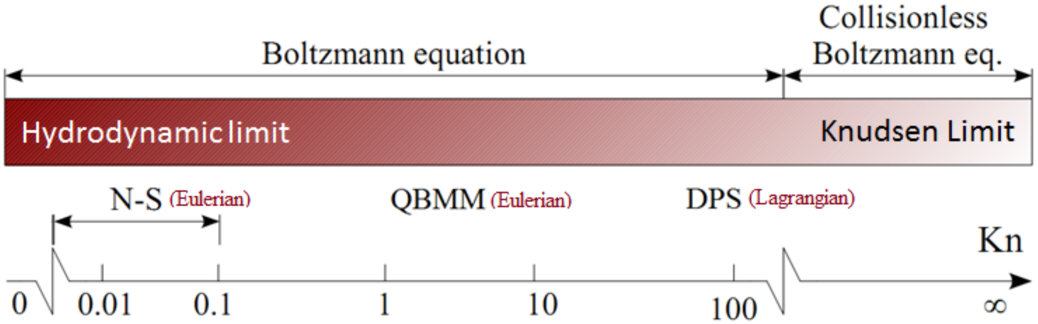


Figure 3. Methods Most Suitable For Kn Number Regimes (Adapted From 1, 2)

A hydrodynamic fluid phase is well-suited to an Eulerian model such as the Navier-Stokes (N-S) equation where the fluid molecules are closely spaced and the molecular motion is dominated by collisions (Kn $\ll$ 1). The basic assumption in these approaches is that the molecules collide so frequently that they share information rapidly and will always attempt to remain at maximum entropy (equilibrium). Therefore, all molecules share stochastic behavior within a small finite cell volume and can be described with a minimal set of cell-averaged variables. One example can be seen in the N-S equation where a single *bulk* velocity replaces all the thousands of molecular velocities contained within a finite cell volume. The bulk velocity is obtained by spatially averaging the ensemble of particle velocities contained within the cell. Another quantity, the time averaged *drift* velocity, can filter out the small random fluctuating molecular motion. Since the fluid is at the hydrodynamic limit (Kn $\ll$ 1), each molecule's drift velocity should be nearly identical and is therefore also nearly equal to the bulk velocity. More details can be found in Section §*F. Micro and macro scales*. Conversely, a more sparsely distributed phase approaching the Knudsen limit (Kn $\gg$ 1), behaves as a rarefied/granular gas where collisions are infrequent. If a carrier fluid is present then the dispersed particle motion becomes dominated by other influences such as the momentum exchange between the carrier and dispersed phases.

1. Lagrangian phase

Historically, the Lagrangian method has been widely employed because it offers a more straightforward/intuitive mathematical implementation. Point particles may be tracked by their free-body motion under forces such as gravity, fluid drag, and other body forces, $\boldsymbol{F}$. This numerical approach is known as Discrete Particle Simulation (DPS), and each particle individually retains its own state variables such as diameter, velocity, acceleration, etc., but most distinguishably, the exact position, $\boldsymbol{x}$, is stored and no grid/mesh is required (other than for an Eulerian carrier fluid). The high detail of this fully-resolved method allows it to be considered a Direct Numerical Simulation (DNS) for the particle phase,[3] and is applicable to *any* Knudsen number though not optimal at small Kn.

External forces exerted on a single particle can be summed using a free-body approach to determine the particle acceleration,

$$\frac{\partial \boldsymbol{v}_d}{\partial t} = \frac{\sum \boldsymbol{F}_d}{m_p} \tag{28}$$

Newton's second law of motion will determine the particle's acceleration ($\frac{\partial}{\partial t}\boldsymbol{v}_d$), and is then integrated to compute the change in particle velocity ($\Delta \boldsymbol{v}$) and integrated once more to determine displacement ($\Delta \boldsymbol{x}$) at every time step. For sparse particle distributions this is an optimal and highly accurate procedure because of its piecewise characteristics; however, under conditions of dense particle dispersions (many particles per cell) this procedure is computationally expensive and when including inter-particle collisions may become computationally prohibitive. Additionally, the states of the particles in the ensemble may become highly redundant and computationally wasteful. Furthermore, instantaneous binary collision models[33,38,41] become invalid at high densities and more complicated for inelastic deformable particles. The solution time step size must also become extremely small to resolve high inter-particle collision frequencies. Lastly, CPU load balancing in High Performance Computing (HPC) applications becomes an added challenge when modeling unsteady non-uniformly distributed particle concentrations (clustering)[26] with DPS.

2. *Eulerian phase*

Designing an Eulerian model capable of describing particles of a non-hydrodynamic phase is a desirable goal for its aforementioned applications in the fields of science and engineering. However, such a model requires considerable numerical erudition; for one, the fictive particles are treated as population densities instead of discrete point-particle entities. In an Eulerian model, the spatial domain is divided into finite volumes called *cells* or *elements* that comprise a spatially discretized mesh or grid. These cells are typically orders of magnitude larger than the fluid molecules or dispersed-phase particles such that the cell contains a large sample size for a bulk average. The Navier-Stokes equations are an example of a single-valued Eulerian model where a single instance of each state variable is stored for each cell. A single-valued or *monokinetic* field[42] specifically reserves only one velocity vector, $\boldsymbol{v} = \{v_1, v_2, v_3\}$, per cell. This reduces the overall amount of information because the fluid molecule properties are inherently spatially averaged over the cell volume. Consequently, the detailed properties of any single molecule/particle are lost, but this is typically a safe assumption because the information pertaining to a huge number of particles is superfluous and unnecessary for making useful models and/or predictions. Instead, some overarching/macroscopic descriptors effectively encompass the neglected information. These include cell-averaged quantities such as pressure, temperature, viscosity, thermal conductivity, etc.[43] that greatly reduce computational costs without sacrificing too much accuracy. Any number of particle properties (degrees of freedom) such as velocity,[42,44,45] diameter,[46] temperature, angular rotation, etc. may be included in the list of internal state variables pertaining to the population of particles.

A multiphase model that utilizes an Eulerian framework for both the carrier flow *and* dispersed phase is called an *Eulerian-Eulerian* or *two-fluid* model. This offers an immediate advantage because both phases can share the same grid and/or solution discretization. Furthermore, if the particle phase has similar momentum properties as the carrier fluid, $\mathrm{St} \ll 1$, then both fluids may share the same velocity field and a simplified "multi-species" approach need only track the diffusion of each species' volume fractions. However, for larger Stokes number flows ($\mathrm{St} > 1$) where the dispersed particle velocities may no longer match those of the fluid velocities, or in the case of a rarefied fluid ($\mathrm{Kn} \gg 1$), QBMM offers a solution to overcome the obstacles described in Section §*III. EULERIAN METHODS - BACKGROUND*.

In conclusion, the computational requirements of an Eulerian model are largely dependent upon the cell-count of the mesh and mostly independent of the particle population/concentrations. Eulerian models provide good CPU load-balancing in HPC applications even in the presence of *highly* non-uniform distributions of particle concentrations. However, extremely rarefied gases or very dilute particle phases will not benefit greatly when using a QBMM model where a Lagrangian model may be optimal. Eulerian models are optimized for describing *dense* populations that can benefit from exploiting statistical averages and the continuum limit,[4] i.e. "information overload" is avoided by retaining statistical descriptors only.

3. Hydrodynamic Constraints

It is important to ascertain the minimum scale at which the hydrodynamic assumption is valid for an Eulerian fluid.[34] In the hydrodynamic limit (Kn $\ll 1$), an infinitesimal volume, η, should contain approximately $\mathcal{N} \approx 10^4$ molecules for good statistical averaging, and a single mole (10^{23} molecules) of gas at standard conditions occupies about 22 liters of volume; therefore, the infinitesimal volume limit for a hydrodynamic gas is[4]

$$\eta \approx 22 \cdot 10^{-3} \frac{10^4}{10^{23}} \approx 10^{-21} [m^3] \tag{29}$$

which corresponds to a cube with a side length of approximately 0.1 microns. Keep in mind that for applications operating on similarly small length scales (such as micro-channels) a hydrodynamic formulation for the carrier-fluid itself may be inaccurate, especially for rarefied gases.

III. EULERIAN METHODS - BACKGROUND

An Eulerian approach uses a type of *Cellular Automata* system, exhibiting sites (cell-centers) connected by lines (mesh), several state variables defined at each cell-center, update rules based upon both local and neighboring cell information, and assumes the discretized medium is “continuous”. The continuum limit assumption neglects the fine details of noise and chaos at the smallest scales (such as the molecular level) and assumes that the cell averaged values are smooth and infinitely divisible at any point between the cell centers. More specifically, a finite-volume (FV) method numerically discretizes the cells into finite volumes and the cell-centers inherit spatially-averaged state values. This reduction in information can significantly lessen computational costs (and “information overload”), but imposes certain limitations upon the model. Theoretically, the Navier-Stokes equations could be substituted by a Lagrangian molecular dynamics (MD) model to achieve the same results, but the computational requirements are presently prohibitive for large scale geometries, especially for liquids.[5] Eulerian FV approaches offer statistically averaged information that provides adequate modeling capabilities in many hydrodynamic fluid applications, but requires extra consideration for non-hydrodynamic phases, specifically the discretization of velocity phase space.

A. Particle Trajectory Crossing

Considering the case of a multiphase flow with a low inertia particles (St $\ll 1$), the dispersed particles will rarely deviate from the fluid streamlines because their velocities,

$\boldsymbol{v}(t, \boldsymbol{x}) \equiv \{v_1, v_2, v_3\}$, obediently conform to the carrier fluid velocity, $\boldsymbol{u}(t, \boldsymbol{x}) \equiv \{u_1, u_2, u_3\}$, dominated by drag forces. Therefore, collisions also occur very rarely and the particle velocity field may be considered "identical" to the fluid velocity field ($\boldsymbol{v} \equiv \boldsymbol{u}$). Alternatively, in the case where the particle inertias are large (St > 1), the particles may begin to deviate from the fluid streamlines and a unique particle velocity field must be distinguished from the carrier fluid velocity field, $\boldsymbol{v} \not\equiv \boldsymbol{u}$. In other words, the particle phase is given a velocity state ($\boldsymbol{v}$) that differs from the carrier-fluid velocity ($\boldsymbol{u}$), and so additional governing equations are required for closure of $\boldsymbol{v}$.

Figure 4 depicts the onset of particle trajectory crossing (PTC), that is, when particles do *not* collide, but pass in close proximity to one another and cross paths. A finite cell volume may be orders of magnitude larger in size than the distance between the particles, and any attempt to isolate these unique particle velocities through mesh refinement is impractical, i.e. it is not feasible to assume that each cell would contain no more than one unique particle at any time. Consequently, the velocities of *each*

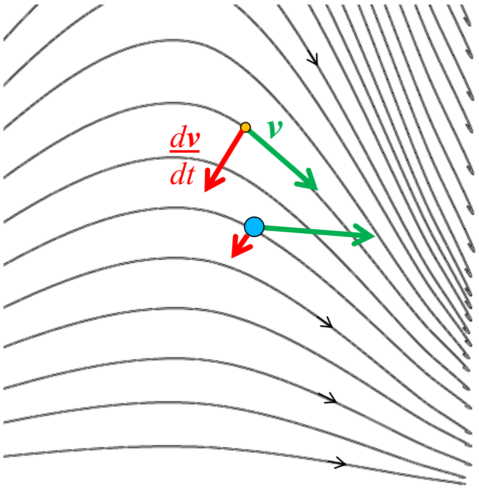
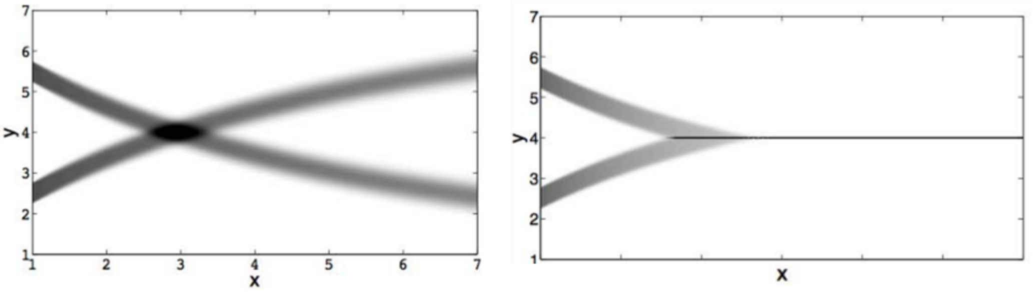


Figure 4. Fluid Streamlines and Particle Trajectory Crossing

particle must somehow be retained if PTC is to be captured properly, and the fundamental assumption of a single-valued Eulerian model must be reconsidered. A single-valued (monokinetic) velocity field is insufficient at describing the multivalued nature of interpenetrating particles that may exist within a single cell. A probability density function (PDF) is one method of defining a diverse ensemble of particle velocities (or any state for that matter). If the variance (σ^2) of the velocity statistics (which measures the diversity of particle motion in a given cell volume) is zero, $\sigma^2 = 0$, then all the particles move at the same velocity and a single-valued assumption is valid. However, diverse particles within close proximity to one another ($\sigma^2 > 0$) will require a multivalued velocity state space to retain the important diversity.

A hydrodynamic model benefits from a monokinetic cell-averaged velocity field (even with a non-zero velocity variance) because the particles are collision-dominated and these collisions restrict the movement of particles; conversely, sparse dispersions of particles are not nearly as spatially confined by collisions and may exhibit substantial particle displacements appearing as self-diffusion at the macroscale. Additionally, the macroscopic values such as: pressure, temperature, and especially viscosity, lose their significance in the presence of strong PTC, i.e. a non-hydrodynamic phase, Kn $> O(1)$.

Figure 5 shows a domain in which two right-bound incoming collisionless (Kn $\gg 1$) particle jets are aimed to cross paths with the goal being to create a large velocity variance ($\sigma^2 > 0$) at their intersection. Figure 5(a) shows the inadequacies associated with a single-valued Eulerian model where the two very different y-velocities become averaged to exactly zero at the intersection, resulting in a delta shock solution; see also



(a) DPS

(b) Eulerian 1-point

Figure 5. Particle Volume Fractions for a Very Sparse Flow (Collisionless, Kn → ∞) with (a) Hydrodynamic Model Showing Unrealizable PTC, and (b) DPS Model Showing Realizable PTC[3]

Kah *et al.*[12] This might be expected for a hydrodynamic fluid, but *not* for a sparse fluid! Figure 5(b) on the other hand, employs a collisionless Lagrangian DPS model that easily predicts the correct physical solution at the intersection of the two sparse jets. An adequate Eulerian model *must* retain multiple instances of the velocity vector within each cell (known as a multi-valued model) similar to how the DPS method stores velocities for *each* individual particle, and the derivation of such a method is described in the next sections.

B. Probability Density Function

To effectively model the transport of any quantity as a continuous state, the accommodating theoretical framework must be carefully selected. A fluid model must be capable of retaining an adequate range of information necessary to describe the details of a possibly diverse and complex particle population describing the medium. A probability density function (PDF), f, of a particle ensemble offers the ability to represent *all* of the diversity for *any* particle attribute(s). f can be an *exact* description of the discrete particle ensemble by employing piecewise Delta functions[22] equivalent to a Lagrangian/DPS framework, but more importantly, for very large populations the continuum limit assumption allows the PDF to be described by a smooth polynomial or some general continuous function. The PDF may have any number of desired variables, i.e., degrees of freedom, $f(t, \boldsymbol{x}; \boldsymbol{v}, \boldsymbol{\xi})$, where t is time, $\boldsymbol{x}$ is the position vector, $\boldsymbol{v}$ is the particle velocity vector, and the vector $\boldsymbol{\xi}$ contains any additional optional values, of length Q, necessary to encompass desired internal states such as particle diameter (L_p), pressure, temperature (T), angular velocity, etc. Numerically, time and position ($t, \boldsymbol{x}$) are discretized by a time-step and mesh respectively, therefore, they are considered *external* parameters of the function f, while the other state vectors ($\boldsymbol{v}, \boldsymbol{\xi}$) remain independent coordinates of the PDF's *internal* state space. For clarity, a semicolon is used to distinguish the external from the internal coordinates in the function parameter list, and the external coordinates are implied when omitted. Note that the discretized grid cell volumes must be small enough to resolve mesoscale gradients, but large enough to contain an adequate ensemble of particles for good statistical sampling. Then the continuum limit assumption ensures that any infinitesimal point in-between the cell centers has a well-defined value also.[5]

The state spaces ($\boldsymbol{\Omega}$) are the domains which encompass the respective state vector coordinates and are denoted by a matching subscript,

$$\boldsymbol{x} \in \boldsymbol{\Omega_x} \equiv \mathbb{R}^D \qquad \boldsymbol{v} \in \boldsymbol{\Omega_v} \equiv \mathbb{R}^D \qquad \boldsymbol{\xi} \in \boldsymbol{\Omega_\xi} \equiv \mathbb{R}^{+Q} \tag{30}$$

where D is the total spatial dimension being modeled. Note that for velocity $\boldsymbol{v}$ and spatial coordinate $\boldsymbol{x}$, the domain includes all real numbers $\mathbb{R}^D$, but for vector $\boldsymbol{\xi}$, individual domains may only extend to positive real values $\mathbb{R}^+$ being appropriate for particle diameters (L_p), Kelvin temperatures (T), absolute pressures, etc.

To summarize, the function f offers a detailed description of any number of desired particle attributes. The parameters to the function act as input coordinates of the state-space, and the corresponding probability of finding particles matching all the specified input coordinates is returned by $f(...)$. Therefore, the function f represents a density of *every* possible coordinate set lying in the domain Ω and must be integrated over any/all coordinates to obtain the desired cumulative/total statistic. The units chosen for f are commonly: particle population number, volume,[47] or mass (all per unit volume of *all* state spaces). In the present work the particle volume is chosen because when integrated properly it lends itself to units of "volume fraction" (ϕ), which keeps the values non-dimensional and always ranging from 0-1. In conclusion, the actual units of $f(t, \boldsymbol{x}; \boldsymbol{v}, \boldsymbol{\xi})$ are "total particle volume (ψ) per unit spatial volume $d\boldsymbol{x}$ per unit volume $d\boldsymbol{v}$ per unit volume $d\boldsymbol{\xi}$ at time t". Note that the term "volume" is not limited solely to spatial volumes because it also applies to internal coordinate volumes of the respective internal state-spaces $\boldsymbol{\Omega}$.

1. PDF Integration

More explanation is required to describe the different ways in which the PDF may be integrated because it has a diverse set of internal coordinates. By convention, the entire state-vector containing all the sub-vectors is defined as $\{t, \boldsymbol{x}; \boldsymbol{\zeta}\}$. These correspond to the time coordinate t, spatial coordinate $\boldsymbol{x}$, and an *internal* state-vector $\boldsymbol{\zeta}$ which includes the previously described vectors $\boldsymbol{\zeta} \equiv \{\boldsymbol{v}, \boldsymbol{\xi}\}$. Together, these vectors comprise the complete and equivalent coordinates for the density function f of a particle population,

$$f(t, \boldsymbol{x}; \boldsymbol{\zeta}) \equiv f(t, \boldsymbol{x}; \boldsymbol{v}, \boldsymbol{\xi}) \tag{31}$$

It is also appropriate to say the units of f are defined as the "volume of particles per unit volume $\boldsymbol{dx}$ per unit volume $\boldsymbol{d\zeta}$". To request the particle volume fraction from f, one simply needs to integrate f with respect to the internal coordinates $\boldsymbol{v}$ and $\boldsymbol{\xi}$; yielding the particle volume fraction at event $(t, \boldsymbol{x})$ contained between the integrated limits [$\boldsymbol{v}$, $\boldsymbol{v}+d\boldsymbol{v}$], and [$\boldsymbol{\xi}$, $\boldsymbol{\xi} + d\boldsymbol{\xi}$],

$$\phi(t, \boldsymbol{x}) = \int_{\boldsymbol{\xi}}^{\boldsymbol{\xi}+d\boldsymbol{\xi}} \int_{\boldsymbol{v}}^{\boldsymbol{v}+d\boldsymbol{v}} f(t, \boldsymbol{x}; \boldsymbol{v}, \boldsymbol{\xi}) d\boldsymbol{v} d\boldsymbol{\xi} \tag{32}$$

where $\phi(t, \boldsymbol{x})$ is the volume fraction, obtained by integrating over the internal state space $(\boldsymbol{v}, \boldsymbol{\xi})$, but not over the external state space $(t, \boldsymbol{x})$. In this case $\phi(t, \boldsymbol{x})$ is a volume fraction (or particle volume per unit spatial volume) at location $\boldsymbol{x}$ at time t (but other units are also appropriate/possible).

Subsequently, further integration of $\phi(t, \boldsymbol{x})$ over coordinate $d\boldsymbol{x}$ can be carried out to obtain the actual volume of particles, ψ, contained within a spatial volume between limits

$[\boldsymbol{x}, \boldsymbol{x} + d\boldsymbol{x}]$.

$$\psi(t)\Big|_{\boldsymbol{x}} = \int_{\Omega_{\boldsymbol{x}}} \phi(t, \boldsymbol{x}) d\boldsymbol{x} \equiv \int_{\Omega_{\boldsymbol{x}}} \int_{\Omega_{\boldsymbol{v}}} \int_{\Omega_{\boldsymbol{\xi}}} f(t, \boldsymbol{x}; \boldsymbol{v}, \boldsymbol{\xi}) d\boldsymbol{v} d\boldsymbol{\xi} d\boldsymbol{x} \tag{33}$$

Given that a finite cell volume is defined by the edge lengths ($d\boldsymbol{x} \equiv dx_1 \cdot dx_2 \cdot dx_3$). The units of $\psi(t)|_{\boldsymbol{x}}$ are described as "volume of all particles contained between locations $\boldsymbol{x}$ and $\boldsymbol{x} + d\boldsymbol{x}$ at time t". The particle number density may be approximated by dividing the volume fraction ϕ by the volume of a single particle (η_p),

$$q(t)\Big|_{\boldsymbol{x}} = \int_{\Omega_{\boldsymbol{x}}} \frac{\phi(t, \boldsymbol{x})}{\eta_p(t, \boldsymbol{x})} d\boldsymbol{x} \approx \frac{\psi(t)}{\eta_p(t)}\Big|_{\boldsymbol{x}} \tag{34}$$

where the approximation on the right only holds in the case that the particles are monodisperse, i.e. particle volume η_p is constant within $d\boldsymbol{x}$.

It is possible to bound the limits of integration to capture the probabilities of particles that lie within certain intervals of desired traits. For clarity, a 3-D example is shown below which obtains the volume fraction of particles with any velocity and any diameter, by choosing the maximum extent of the limits of integration for each corresponding space's "volume": $d\boldsymbol{v} \equiv dv_1 ... dv_D$ or $d\boldsymbol{\xi} \equiv d\xi_1 ... d\xi_Q$. In this example $D = 3$ and $Q = 1$ (i.e. particle diameter as ξ_1),

$$\phi(t, \boldsymbol{x}) = \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty f(t, \boldsymbol{x}; \boldsymbol{v}, \boldsymbol{\xi}) dv_1 dv_2 dv_3 d\xi_1 \tag{35}$$

while t and $\boldsymbol{x}$ remain as external coordinates to the function $\phi(t, \boldsymbol{x})$. The limits of integration can be restricted independently as required.

2. PDF bins

The continuous nature of the analytical PDF represents only a theoretical *smoothed* representation of a particle distribution, but in reality the particles are discrete such that the actual physical PDF will not be perfectly continuous. Experimentally, a PDF is obtained by measuring the unique particle properties and is limited by the measurement resolution. A polydisperse particle phase in which the particle diameters (ξ_1) vary may be plotted as a histogram with a finite number of bins as shown in Figure 6. This typical distribution, $f(\xi_1)$, for particle diameters may be obtained by sieving with screens of varying fineness.[48] Thus the width of the bins is determined by the experimental sieving equipment. Similarly, other experimental measurements (velocity, temperature, pressure, etc.) will exhibit similar resolution restrictions on the corresponding PDF's. Probability distributions exist for all of the internal state space dimensions and experimental

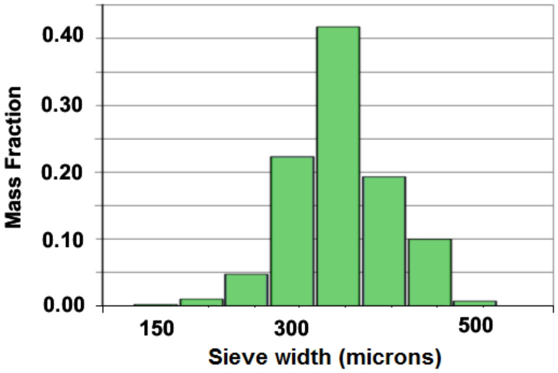


Figure 6. Example of Experimentally Obtained PDF of Particle Diameter $f(\xi_1)$

measurements can produce corresponding histograms.Typically these distributions are normal (Gaussian), log-normal, or Rosin-Rammler.[20]

C. Boltzmann Equation

In the present work, only the particle velocities are being tracked, such that $\boldsymbol{\xi} = \{\}$ is empty, and the particle diameters are assumed equal (monodisperse particle phase); therefore, $f(t, \boldsymbol{x}; \boldsymbol{v})$ describes the desired degrees of freedom (the input coordinates) sufficient for the current purposes.

$$\boldsymbol{\zeta} \equiv \boldsymbol{v} = \{v_1, v_2, v_3\} \tag{36}$$

Ludwig Boltzmann presented a PDF transport equation in 1872 which is valid for a function, f, with any degree of freedom. This fundamental kinetic governing equation is known as the Boltzmann Equation,[7,41] and the specialized version for $f(t, \boldsymbol{x}; \boldsymbol{v})$ is given as,

$$\frac{\partial f}{\partial t} + \boldsymbol{v}\frac{\partial f}{\partial \boldsymbol{x}} + \frac{\partial}{\partial \boldsymbol{v}} \cdot (\dot{\boldsymbol{v}} f) = C \tag{37}$$

The PDF is the central component within the Boltzmann equation, and it provides the foundation upon which many Eulerian CFD transport models operate. Even the DPS formulation is a method of solving the Boltzmann equation! Furthermore, the Navier-Stokes equations can be derived from the Boltzmann equation by assuming a single-valued velocity field. However, a useful Eulerian model for large Knudsen or Stokes numbers must make *no* assumption that restricts the continuous phase to a single-valued solution. A multi-valued Eulerian solution is the motivation of this work and a highly desirable capability. If the coordinates of f are restricted to velocity and particle size, then the equation is known as the Williams equation.[11,12,49,50] If instead f is normalized to represent particle population, the equation is termed a population balance equation (PBE).[24,47,51] Solving the complete continuous Boltzmann equation is an extremely difficult mathematical procedure and certain assumptions about f will make the kinetic equation manageable to solve numerically. One method is to employ a quadrature decomposition that discretizes the distribution f into a set of node points.

D. Discretized Nodes

Suppose that a set of attributes are desired to represent all internal degrees of freedom for a single particle such that a vector, $\{\hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{\xi}, ...\}$, sufficiently describes the particle within a cell at time and position, $t, \boldsymbol{x}$, (which are external parameters). Let the corresponding PDF have the same input coordinates and an associated output weight such that $\phi = f(\hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{\xi}_1, ...)$. One method of discretizing f is to store these arbitrary inputs with their corresponding output weight, thus forming a *node*,

$$\{\hat{\boldsymbol{\zeta}}; \phi\} \equiv \{\hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{\xi}, ...; \phi\} \tag{38}$$

Now if f is a very diverse PDF, then one should consider storing multiple instances of these nodes to increase the *resolution* of the internal state space and encompass a better representation of the distribution. Let these node vectors be grouped to form a row-matrix representation,

$$\begin{array}{cc} & \begin{array}{cccccc} \multicolumn{4}{c}{\xrightarrow{\text{Internal variables (1...D, 1...Q)}}} & & \\ \boldsymbol{U} & \boldsymbol{V} & \boldsymbol{W} & \boldsymbol{\xi}_1 & & \text{Weight} \end{array} \\ \text{Node, } \alpha \Bigg\downarrow \begin{array}{c} 1 \\ 2 \\ \vdots \\ \mathbb{B} \end{array} & \begin{pmatrix} \hat{v}_{1,1} & \hat{v}_{2,1} & \hat{v}_{3,1} & \hat{\xi}_{1,1} & \cdots & \phi_1 \\ \hat{v}_{1,2} & \hat{v}_{2,2} & \hat{v}_{3,2} & \hat{\xi}_{1,2} & \cdots & \phi_2 \\ & \vdots & & & & \vdots \\ \hat{v}_{1,\mathbb{B}} & \hat{v}_{2,\mathbb{B}} & \hat{v}_{3,\mathbb{B}} & \hat{\xi}_{1,3} & \cdots & \phi_{\mathbb{B}} \end{pmatrix} \end{array} = \mathbb{N} \tag{39}$$

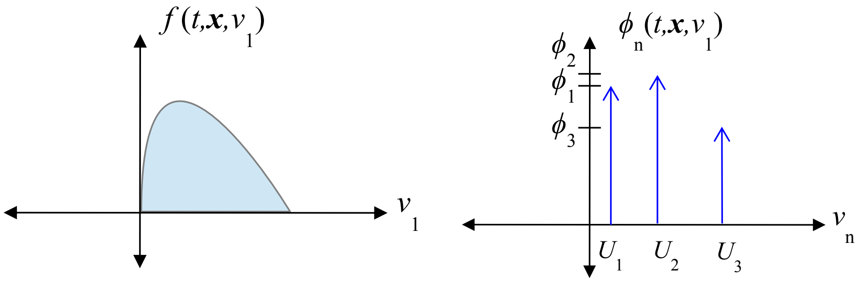
where each row, $\alpha = 1, ..., \mathbb{B}$, describes a single "node" (instance of the multi-valued representation), and each column vector, $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\xi}_1, ...\}$, describes a dimension of the internal state space from: $1, ..., \mathcal{X}$. The total degrees of freedom, $\mathcal{X} \equiv D + Q$, represents the total dimension of the internal state space of f. The final column $(\mathcal{X} + 1)$ is reserved to store the weights, $\boldsymbol{\phi}$, which are the probability densities associated with finding a particle with the attributes in that same row, $\phi_\alpha = f(\hat{v}_{1,\alpha}, \hat{v}_{2,\alpha}, \hat{v}_{3,\alpha}, \hat{\xi}_{1,\alpha}, ...)$. The discrete instances of the coordinates (denoted by the hat) are called *abscissas* and multiple instances of the nodes are stored where $\mathbb{B}$ is the customizable number of allotted nodes. Let this multi-valued node representation of f be termed matrix $\mathbb{N}(t, \boldsymbol{x})$ which exists for every coordinate in the mesh. Consequently, $\mathbb{B}$ defines the *resolution* of the discretized PDF, and a total of $\mathbb{B} \cdot (\mathcal{X} + 1)$ variables will be stored in $\mathbb{N}$.

In essence, the discretized values in $\mathbb{N}$ represent Dirac delta functions that can be used to construct f in a piecewise manner. Unlike a direct solver (DPS), a quadrature-based discretization of a phase space into only $\mathbb{B}$ total nodes may be very sparse,[52] $\mathbb{B} \ll \mathcal{N}$, compared to the actual diversity present within the cell's population containing $\mathcal{N}$ particles, but nonetheless is still very capable of predicting flows. In fact, the nodes are most effective when they are representative of a large number of particles. Retaining at least $\mathbb{B} = 2$ nodes allows particles to cross paths (PTC), which is why flows with large Stokes number require a multi-valued ($\mathbb{B} > 1$) approach. Increasing the number of nodes, $\mathbb{B} > 2$, will allow for more diversity within any cell, because it effectively allows for more bins of a histogram or a more refined representation of the particle population. In the present work, a two node system ($\beta = 2$) is considered for *each* dimension ($\mathbb{B} = \beta^D$), known as a 2-point quadrature approximation, and its implementation is described throughout the remainder of the paper.

It should also be mentioned that the choice of Dirac delta functions as the quadrature basis is made for simplicity, and a more complex Multi-Gaussian (MG)[53–55] quadrature method is also possible in Extended QMOM (EQMOM). In an MG approach, the nodes are represented by Gaussian distribution functions in the velocity phase space.

E. Quadrature Approximation

In its continuous form, the PDF is extremely difficult to solve in a Partial Differential Equation (PDE) such as Equation (37), but as explained in the previous section, nodes in matrix $\mathbb{N}$ provide a discretized representation of f which is a starting point for a numerical approach. Employing β-point quadrature approximations to represent the polynomial f, effectively reduces the PDF to a set of abscissas and weights as depicted in Figure 7. Recall that the choice of this *basis* function is entirely for simplicity, and is by no means limited only to Dirac Delta functions. Essentially, a quadrature process exists (see Section §*E. Moment Inversion*) that *elects* representative node points to describe the diverse population.



(a) Continuous (b) Quadrature approximation

Figure 7. The x-velocity (v_1) 1-D PDF Distribution Represented (a) Continuously and By (b) $\beta = 3$ Approximation

One process of representing a continuous function f into β number of discrete points, is known as a β-point Gaussian Quadrature. This method automatically situates the abscissas and weights to best retain the area under the curve and the general shape of the curve. As $\beta \rightarrow \mathcal{N}$, the quadrature approximation fully resolves the state-space for $\mathcal{N}$ particles in the ensemble. However, for very large $\mathcal{N}$, a continuum limit assumption can be made to treat the PDF as a continuous function. This is advantageous because if the continuous form of f is a polynomial of order $2\beta - 1$ or smaller, then the β-point Gaussian quadrature approximation of the polynomial is *exact* and saves computational costs. Otherwise the quadrature representation attempts to *approximate* the shape of higher order polynomials, f, to the best of its abilities. The abscissas correspond to the zeros of a Hermite polynomial of order β, and in general these are the zeros of an orthogonal polynomial where the unknown weighting function is the PDF.[56]

Let $\hat{\boldsymbol{v}}$ be the abscissa matrix (see Equation 39), and let each dimension of velocity be represented by a column vectors: $\boldsymbol{U}, \boldsymbol{V}$, and $\boldsymbol{W}$ such that,

$$\{\hat{v}_{1,\alpha}, \hat{v}_{2,\alpha}, \hat{v}_{3,\alpha}\} \equiv \{U_\alpha, V_\alpha, W_\alpha\} \tag{40}$$

with associated weight ϕ_α. Figure 7 (b) shows an example of the x_1-dimension of the velocity space, where $\beta = 3$ instances of U_α comprise the discretized PDF. Not depicted are the x_2 and x_3 dimensional abscissas V and W which could also have up to $\beta = 3$ abscissas each.

F. Micro and macro scales

The renowned hydrodynamic fluid assumption operates on the concept that the microscale motion can be disregarded without greatly sacrificing the ability to predict the macroscale behavior. For a molecular fluid, the microscale velocity, $\boldsymbol{v}$, is that pertaining to stochastic Brownian molecular motion or temperature, and appears directly in the Maxwell-Boltzmann distribution. Fluctuations at the microscale are small enough (compared to the macroscopic velocity) that microscale transport (self-diffusion[57,58]) caused by the Brownian motion can usually be neglected in the hydrodynamic limit (Kn $\ll$ 1), especially for a liquid. In fact, for isotropic systems, the average motion is zero because there is no preferred direction for the particles to move, but the mean square displacement does increase very slowly over time, resulting in some self-diffusion[7] many orders of magnitude smaller than the fluid length scales of interest. Essentially, collision-dominated particles remain locally confined even when exhibiting very large and random instantaneous velocities. One way to distinguish the prominence of self diffusion is to consider a *drift* velocity $\bar{\boldsymbol{v}}$ of the i'th particle as the time average velocity over a large enough time period ($\Delta t = t_1 - t_0 \approx L_V/\mathbb{U}$) such that the microscale fluctuations in v after numerous collisions are filtered out,

$$\bar{\boldsymbol{v}}_i = \frac{1}{\Delta t}\int_{t_0}^{t_1} \boldsymbol{v}_i(t)dt = \frac{\boldsymbol{x}_i(t_1) - \boldsymbol{x}_i(t_0)}{\Delta t} \tag{41}$$

Alternately, the spatial displacement of the i'th particle, $\Delta\boldsymbol{x}_i$, over the same time period, Δt, can also determine the drift velocity.

As an example, consider that a molecular fluid in hydrostatic equilibrium at *rest* would exhibit $\bar{\boldsymbol{v}}_i \approx \mathbf{0}$ even if $|\boldsymbol{v}_i| \gg 0$ at high temperature, T. Consider another variable, the *bulk* velocity, $\langle\boldsymbol{v}\rangle$, as the spatially averaged value over a small volume, η, that encompasses a large ensemble of $\mathcal{N}$ particle velocities,

$$\langle\boldsymbol{v}\rangle = \frac{1}{\eta}\int \boldsymbol{v}_i(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{\mathcal{N}}\sum_{i=1}^{\mathcal{N}} \boldsymbol{v}_i \tag{42}$$

This bulk velocity would also be nearly zero, $\langle\boldsymbol{v}\rangle \approx 0$, for the same hydrostatic example fluid. Notice that $\bar{\boldsymbol{v}}_i$ is a temporal average of a *single* particle and $\langle\boldsymbol{v}\rangle$ is a spatial average of an instantaneous particle *ensemble*, yet both are nearly zero. This leads to an interesting conclusion for the collision-dominated hydrodynamic fluid: if the particles are indeed dominated by collisions, not only must all particles in the ensemble have nearly the same drift velocity, but their drift velocities must also nearly equal the instantaneous bulk velocity,

$$\langle \boldsymbol{v} \rangle \approx \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \bar{\boldsymbol{v}}_i \approx \bar{\boldsymbol{v}}_i \tag{43}$$

because $\mathcal{N}$ is a large sample size of stochastic particles. It then follows that the PDF of particle drift velocities (residing within volume η) is approximately a Delta function, $f(\bar{\boldsymbol{v}}) = \delta(\bar{\boldsymbol{v}} - \langle \boldsymbol{v} \rangle)$, yet the instantaneous $f(\boldsymbol{v})$ could be a Maxwell-Boltzmann distribution for a molecular fluid. Therefore, a hydrodynamic fluid benefits by not directly resolving the microscale velocities, and instead encompasses the microscale behavior into simplified scalar quantities such as fluid viscosity (μ), temperature (T), and pressure.

All of these assumptions are still valid when the fluid undergoes bulk macroscopic currents, i.e. $\langle \boldsymbol{v} \rangle \neq \boldsymbol{0}$, and the first row of Figure 8 depicts all the distributions and variables so far described in the hydrodynamic limit (Kn $\ll 1$). In order to safely neglect self-diffusion, the Brownian motion must be very small,[4] i.e. the drift velocities must be nearly the same as the bulk velocities per Equation (43). Conversely, when collisions are negligible or infrequent, the drift velocity and fully resolved velocities become synonymous, $\boldsymbol{v}_i \equiv \bar{\boldsymbol{v}}_i$, when there are no collisions (or other external forces) to alter the particle trajectories. If PTC is indeed dominant, then the particles are less spatially constrained such that $|\bar{\boldsymbol{v}}_i - \langle \boldsymbol{v} \rangle| \gg 0$. In other words, $\bar{\boldsymbol{v}}$ from Equation (41) may vary widely for individual particles contained within a small volume η. As a consequence, the instantaneous $f(\boldsymbol{v})$ may be any arbitrary non Maxwell-Boltzmann distribution, and the drift velocities, $f(\bar{\boldsymbol{v}})$, are no longer confined to a Delta function either. When self-diffusion becomes prevalent, $f(\boldsymbol{v})$ is dictated by the terms on the left hand side of

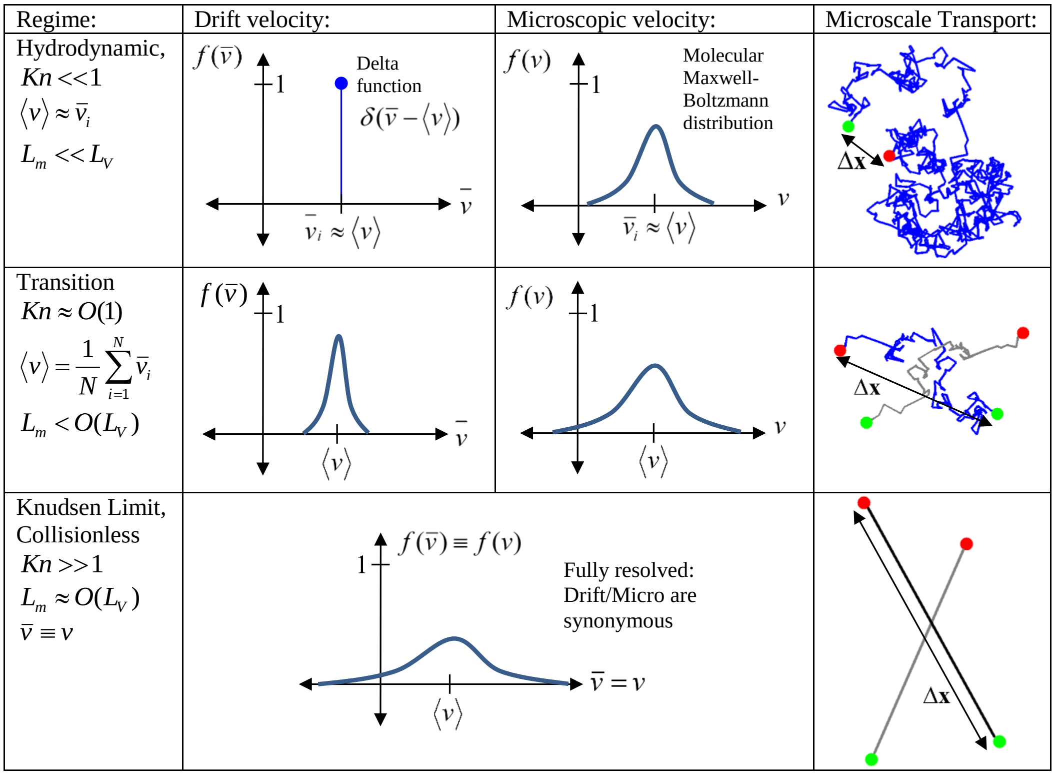


Figure 8. The Molecular (and Granular) Microscale and Macroscale Relationships Within Various Knudsen Regimes

Equation (37), therefore the real challenge is in determining the macroscopic behavior of these sparse particle concentrations characterized by Kn $> O(1)$.

In conclusion, as $Kn \rightarrow \infty$, it is insufficient to simply describe the macroscopic behavior with the bulk velocity, $\boldsymbol{v} \neq \langle \boldsymbol{v} \rangle$, and the concept of hydrodynamic viscosity breaks down. In fact, $\boldsymbol{v}$ must be represented using a PDF of the velocity state-space and a collision model must replace the viscosity assumption. However, the concepts of temperature and pressure retain much of the same meaning in the granular sense.[1] QBMM attempts to resolve the microscale motion with a multi-valued velocity field and offers a collision model that can seamlessly operate in any Knudsen regime.

G. Viscosity Limitations

The viscosity of a hydrodynamic fluid is related to the momentum transfer caused by collisions between molecules. The viscosity of a gas is proportional to the mean relative molecular speed, density, and the mean free path as, $\mu_f \approx |v_c|\rho_f L_c$, respectively. However, as the density increases, the mean free path decreases proportionally, therefore viscosity is actually independent of density and pressure.[58] Furthermore, it is known that the molecular *speeds* in a fluid at the hydrodynamic limit (Kn $\ll 1$) and at thermal equilibrium exhibit a Maxwell-Boltzmann distribution.[7] The *velocity*-space distribution corresponding to the Maxwell-Boltzmann distribution is the *Gaussian distribution*, $f_{eq}(\boldsymbol{v})$, which arises from the stochastic behavior of the particles.[33,43] For a given variance (σ^2) in a PDF, entropy is maximized by a Gaussian distribution centered about the mean velocity, and a system of collision-dominated particles left unperturbed will naturally return to that velocity distribution over time. Maximum entropy is attained more quickly when Kn $\rightarrow 0$ because the increased frequency of collisions provide a rapid mechanism for the diffusion of information,

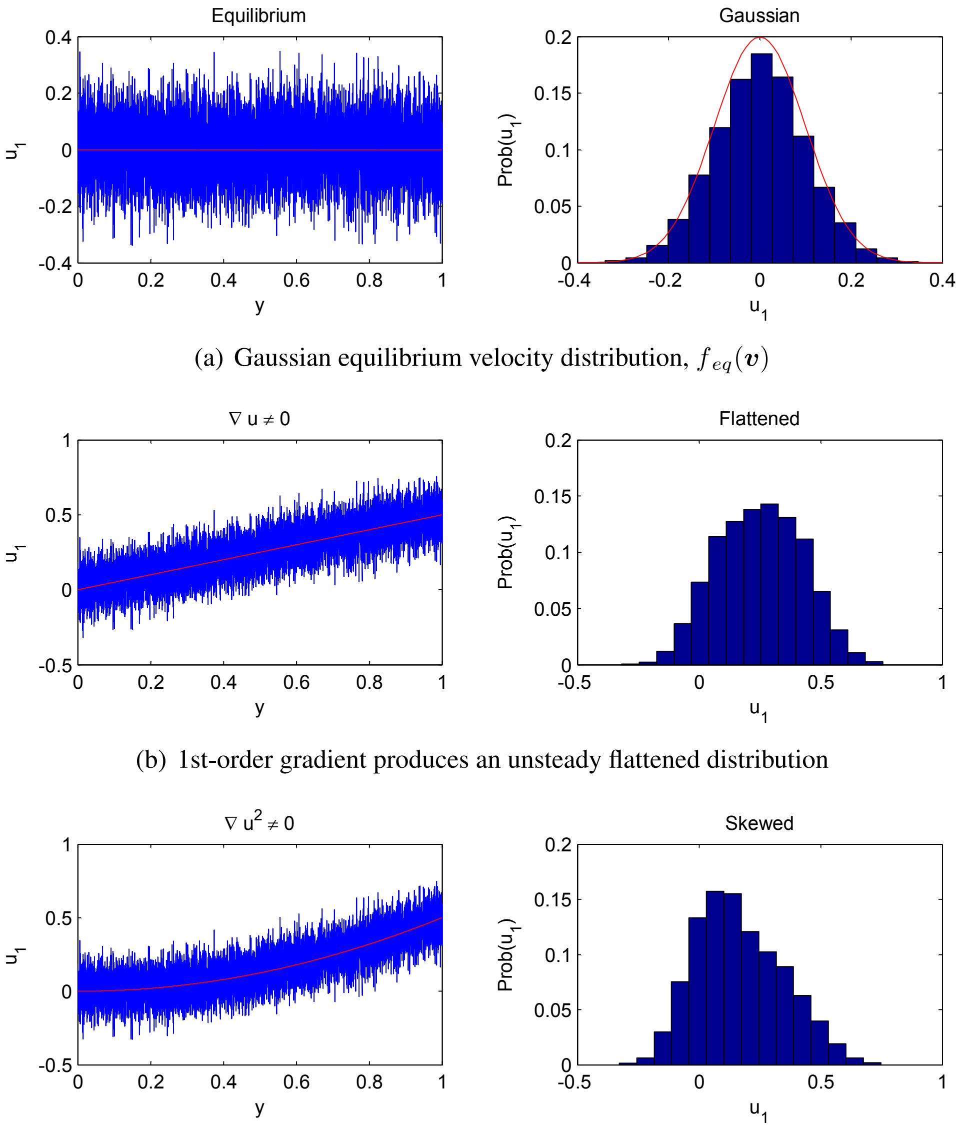
$$\lim_{\text{Kn}\rightarrow 0} f(\boldsymbol{v}) \rightarrow f_{eq}(\boldsymbol{v}) \tag{44}$$

As a result, the Gaussian distribution is a steady-state solution to the Boltzmann equation,[52] and any deviation from equilibrium, $f(\boldsymbol{v}) \neq f_{eq}(\boldsymbol{v})$, will introduce unsteady effects that depend on the strength of the collision term. Note that Kn $= 0$ is a theoretical limit in which the molecules would always be in contact (colliding) like a tightly packed solid where information is shared instantaneously; a physical impossibility.

At the opposite extreme, two unique scenarios will produce a Knudsen limit "fluid" (Kn $\rightarrow \infty$). Either the particles are extremely sparse *or* they do not exhibit variance ($\sigma^2 = 0$); both cases are collisionless ($\tau_c \rightarrow \infty$). However, an extremely important realization is that the velocity PDF distribution will *always* seek to return to the steady-state Gaussian distribution of maximum entropy, $f_{eq}(\boldsymbol{v})$, regardless of the Knudsen number! The limiting factor is the collision frequency that allows information between particles to be shared and diffused, and this frequency dictates the duration required to reach the steady-state Gaussian distribution. The collision time, τ_c, represents the average time between collisions and is therefore proportional to the Knudsen number ($\tau_c \propto$ Kn) and inversely proportional to the collision frequency ($1/\tau_c$). In other words, as Kn increases, the distribution, f, can remain further from equilibrium for longer durations because the mechanism for returning to equilibrium (collisions) becomes a less dominant factor. Accordingly, a collision model which is valid for any Kn may be derived by assuming that the particles will *always* attempt to achieve the maximum entropy distribution, $f_{eq}(\boldsymbol{v})$, albeit more slowly when τ_c is larger. As will be shown later in Equation (118), the collision model does indeed depend inversely upon τ_c.

The inter-molecular collisions are directly responsible for the viscosity in hydrodynamic fluids and to demonstrate this, consider the most basic isotropic case. For isotropic collisions, the steady-state Gaussian distribution should be observable at any scale that encompasses an adequately large ensemble of molecules. Consider a finite volume with non-dimensional width $\Delta x_2 = 1$ in the y direction, and the random molecular velocity u_1 with standard deviation $\sigma = 0.1$ in the perpendicular x direction. Figure 9(a) shows the random distribution of molecular velocities along the cell's entire spatial dimension from $y = 0$ to $y = 1$ at hydrostatic equilibrium. The right side of Figure 9(a) shows the corresponding histogram distribution of molecular velocities of the *entire* cell and the red curve is an exact Gaussian distribution. Note that zooming-out further to encompass a larger domain would yield the same self-similar distribution.

Now consider the next case of a 2-D Couette flow where a gradient $\frac{\partial u_1}{\partial y} \neq 0$ is present. One would expect that the molecular velocities will quickly conform to their neighbors due to localized momentum transfer, i.e. the Gaussian distribution may be more pronounced at the smallest scales, but the velocity gradient will be very apparent at larger scales when zooming-out. In other words, the distribution, $f(u_1)$, across the entire cell from $y = 0$ to $y = 1$ becomes *flattened* as shown in Figure 9(b), and even though it may still be symmetrical it does *not* provide the maximum entropy that only a Gaussian distribution can. Next, consider also a Poiseuille flow in which the fluid experiences a macroscale second-order gradient $\nabla^2 u = \frac{\partial^2 u_1}{\partial y^2} \neq 0$. In this case, the cell's velocity distribution becomes *skewed* as shown in Figure 9(c) and $f(u_1)$ is no longer self-similar because it depends on the spatial scale. Both flows are examples of non-equilibrium distributions, $f(\boldsymbol{u}) \neq f_{eq}(\boldsymbol{u})$, that will cause an unsteady response if left unperturbed as the molecules attempt to return to maximum entropy. In conclusion, the tendency of molecules to seek the maximum entropy distribution (Gaussian) is manifested as viscosity and this phenomenon is the source of the viscous shear force. A perturbation method has



(a) Gaussian equilibrium velocity distribution, $f_{eq}(\boldsymbol{v})$

(b) 1st-order gradient produces an unsteady flattened distribution

(c) 2nd-order gradient produces an unsteady skewed distribution

Figure 9. Velocity Distributions for Various Macroscale Gradients

been developed as a collision model based purely upon these statistical mechanics and may also be extended to both sparser flows, Kn $> O(1)$, and granular particles alike with reasonable effectiveness. More details about this collision source term are described in Section §*1. Inter-Particle Collisions*.

IV. MULTI-VALUED EULERIAN MODEL

As previously introduced for a multiphase flow, if the fluid-particle relationship is of a low Stokes number ($\mathrm{St} \ll 1$), the dispersed particle trajectories (and thus velocities) will closely adhere to the carrier flow streamlines. Conversely, if the relationship is of a higher Stokes number ($\mathrm{St} > 1$), the larger inertia of the particles will allow them to stray and diverge from the path of the fluid streamlines. This can lead to PTC which cannot be represented by a single-valued, monokinetic, Eulerian model in which only one (cell-averaged) instance of each velocity variable is present ($\beta = 1$ node).[1] The Quadrature-Based Moment Methods (QBMM) overcome this issue by allowing multiple nodes of particles to be present at each discretized location in space, i.e. a multi-valued field. The Lattice Boltzmann Method (LBM) is similar to QBMM, but LBM uses fixed abscissas while varying only the weights and has shown success in discretizing the velocity state space.[6,27,43,44,52,59,60] QBMM provides additional flexibility because it allows the abscissas to shift their values as needed and their quantity is adaptable depending on the desired PDF resolution. Another general tool is Monte Carlo simulation,[6] but this randomized "brute force" approach can be expensive, and DPS may even fit into this category. Lastly, Smoothed Particle Hydrodynamics (SPH) is another contender for dispersed phases that is quite useful for modeling free surfaces.[27]

A. Moments of a PDF

The motivation behind the QBMM family is to develop a system of transport governing equations[61] using Equation (37) for the continuous PDF, f. However, a few simplifying assumptions must first be made to discretize f into a set of nodes which are appropriate for numerical modeling. Information about the PDF's exact curve is representable through a Gaussian quadrature approximation, but "moments" of f also retain useful information about the PDF. These moments are constructed by weighting/skewing the integration of f as shown in Equation (45) by using multiples of the internal state variables as factors.

Let D=3 and Q=0 describe a PDF with exclusively three particle velocity components, $\boldsymbol{v}$, such that the PDF is fully described as $f(t, \boldsymbol{x}; \boldsymbol{v})$. The velocity moment $M^{(\gamma)}_{l_1,l_2,l_3}$ will be defined as,

$$M^{(\gamma)}_{l_1,l_2,l_3}(t, \boldsymbol{x}) = \int v_1^{l_1} v_2^{l_2} v_3^{l_3} \; f(t, \boldsymbol{x}; \boldsymbol{v}) d\boldsymbol{v} = \int h_{l_1,l_2,l_3} \; f(t, \boldsymbol{x}; \boldsymbol{v}) d\boldsymbol{v} \qquad (45)$$

where the subscript integers $l_1, l_2, l_3 \geq 0$ (of which there are $\mathcal{X} = D + Q$ subscripts in total) represent the arbitrary multiplicities of the dth abscissa which define the test function $h_{l_1,l_2,l_3} = v_1^{l_1} v_2^{l_2} v_3^{l_3} = \prod_{d=1}^{\mathcal{X}} v_d^{l_d}$ for construction of the moment set; see Equation (93). The sum of the multiplicities, $\gamma = l_1 + l_2 + l_3 + ... = \sum_{d=1}^{\mathcal{X}} l_d$, is simply the *order* of the moment (written in the superscript). Limiting the choices of l_1, l_2, l_3 strictly to integers produces a Gaussian Quadrature[52] which is the highest accuracy representation available, as opposed to choosing non-integers.[47,62] Note that a *simple* moment is defined as having *only one* nonzero l_1, l_2, or l_3, otherwise it is known as a *mixed* moment when at least two indices are non-zero.

B. Moment and Distribution Properties

Each moment describes a specific feature of the original detailed PDF, $f(t,\boldsymbol{x};\boldsymbol{v},...)$ exactly. $M^{(0)}$ describes the “area under the curve”, $M^{(1)}$ is the *mean*, $M^{(2)}$ describes the energy (thus also the variance), $M^{(3)}$ pertains to the *skew*,[63] and higher orders represent *Kurtosis*. The higher orders do not have standard names and may not always represent physically intuitive values, but may provide useful information describing details about the shape of the PDF. Theoretically, a sufficiently large set of moments should be able to describe any arbitrarily sophisticated PDF. Therefore, a moment set, vector $\boldsymbol{M}(t,\boldsymbol{x})$, offers an alternative representation of the information contained in $f(t,\boldsymbol{x};...)$.

Additionally, the *variance* and granular temperature ($\sigma^2 \equiv \Theta$) are identical, so both are inherently positive and describe how *spread-out* the distribution f is. The Dirac Delta function uniquely has *zero* variance ($\Theta = 0$) where all the particles conform to a single value. Equation (46)[1] describes up to the 2nd-order *simple* velocity moments of $f(t,\boldsymbol{x};\boldsymbol{v})$,

$$\begin{aligned}
M^{(0)}_{000} &\equiv M^{(0)} \equiv \int_{\Omega_{\boldsymbol{v}}} f(\boldsymbol{v})d\boldsymbol{v} \quad\;\; = \langle\phi\rangle \\
M^{(1)}_{l_1l_2l_3} &\equiv M^{(1)}_d \equiv \int_{\Omega_{\boldsymbol{v}}} v_d f(\boldsymbol{v})d\boldsymbol{v} = \langle\phi\rangle\langle v_d\rangle \\
M^{(2)}_{l_1l_2l_3} &\equiv M^{(2)}_d \equiv \int_{\Omega_{\boldsymbol{v}}} v_d^2 f(\boldsymbol{v})d\boldsymbol{v} = \langle\phi\rangle\left(\Theta_d + \langle v_d\rangle^2\right)
\end{aligned} \tag{46}$$

where $\langle\phi\rangle \approx \sum_{\alpha=1}^{\beta}\phi_\alpha$ is the sum of all discrete node weights and $\langle v_d\rangle \equiv M^{(1)}_d/M^{(0)}_d$ is the cell-average velocity in dimension d. The simplified notation with a single subscript d uniquely identifies a *simple* moment of order γ in dimension d where *only* the v_d

component of $\boldsymbol{v}$ appears as a product on $f(\boldsymbol{v})$ over the entire velocity state space, $\Omega_{\boldsymbol{v}}$. Granular temperature vector $\boldsymbol{\Theta}$, described in Equation (65), corresponds to the D total components of the diagonals of the velocity covariance tensor, θ, described in Equation (66).

1. Limitations

An Eulerian representation of particles has some inherent limitations as compared to the "pinpoint" accuracy of a Lagrangian method. Since the Eulerian particles are contained within the finite cells of a mesh/grid, the local f inherently describes the ensemble of $\mathcal{N}$ particles contained in that cell. Consequently, even if the computational power were available to describe *each* particle with a its own node ($\beta = \mathcal{N}$), the exact particle positions are still unknown. In other words, the velocity distribution $f(t, \boldsymbol{x}; \boldsymbol{v})$ could be known exactly for all $\boldsymbol{v}$, but the particle positions themselves are not precisely defined like in a DPS model, and are only as resolved as the cell center location, being contained somewhere within the bounds of the local cell dimensions: $[\boldsymbol{x}, \boldsymbol{x} + d\boldsymbol{x}]$.

Furthermore, the state of any represented Eulerian particle has the ability to influence neighboring cells through convection/flux transport terms of the governing equation. In fact, particles may exist partially among multiple neighboring cells simultaneously while transitioning between them in a probabilistic continuum limit description. The continuum approach mandates that the particle probability gradually shifts from one cell to another during Eulerian transport (depending on the CFL number), and is best at modeling high concentrations of particles.

C. Moment Transport

The main purpose and motivation behind utilizing moments is for their ability to be transported through space and evolved over time with a set of hyperbolic PDE's. More specifically, each moment has a unique transport equation (derived from the Boltzmann Equation), effectively creating the mass, momentum, energy, and higher order conservation equations needed to evolve the PDF (f) as an Eulerian continuum. The moment transportation equations are obtained by multiplying both sides of the Boltzmann Equation (37) by the abscissas $\hat{\boldsymbol{v}}$ raised to the l_1, l_2, l_3 exponents, and then integrating both sides[44] as demonstrated in Equation (45). This yields an integro-differential equation,

$$\frac{\partial}{\partial t} M^{(\gamma)}_{l_1,l_2,l_3} + \frac{\partial}{\partial x_1} M^{(\gamma+1)}_{l_1+1,l_2,l_3} + \frac{\partial}{\partial x_2} M^{(\gamma+1)}_{l_1,l_2+1,l_3} + \frac{\partial}{\partial x_3} M^{(\gamma+1)}_{l_1,l_2,l_3+1} = \\ l_1 \dot{v}_1 M^{(\gamma-1)}_{l_1-1,l_2,l_3} + l_2 \dot{v}_2 M^{(\gamma-1)}_{l_1,l_2-1,l_3} + l_3 \dot{v}_3 M^{(\gamma-1)}_{l_1,l_2,l_3-1} + \mathbb{C}^{(\gamma)}_{l_1,l_2,l_3} \quad (47)$$

where the dependance on $(t, \boldsymbol{x})$ is always implied and suppressed on $M^{(\gamma)}_{l_1,l_2,l_3}(t, \boldsymbol{x})$. Any subscripts (multiplicities) resulting in -1 are assumed zero by convention. Notice that the convection terms have all acquired an additional multiple of v_d which increases their moment order by one (γ+1). On the right-hand side, moments of order γ-1 result from the state-space acceleration[42] ($\dot{\boldsymbol{v}} = \frac{d\boldsymbol{v}}{dt}$) terms which have lost one order of v_d and gained an l_1,l_2, or l_3 multiple due to the derivative taken with respect to $\boldsymbol{v}$. For simplicity, Equation (47) can be generalized as

$$\underset{\text{Unsteady}}{\frac{\partial}{\partial t}M^{(\gamma)}} + \underset{\text{Convective}}{\nabla_{\boldsymbol{x}} \cdot M^{(\gamma+1)}} = \underset{\text{Acceleration}}{\boldsymbol{l}\dot{\boldsymbol{v}} \cdot M^{(\gamma-1)}} + \underset{\text{Source}}{\mathbb{C}^{(\gamma)}} \tag{48}$$

where the vector $\boldsymbol{l} \equiv \{l_1, l_2, l_3\}$ contains the permutation of integer exponents, $\dot{\boldsymbol{v}} \equiv \{\dot{v}_1, \dot{v}_2, \dot{v}_3\}$ is the acceleration vector, and the operation $\boldsymbol{l}\dot{\boldsymbol{v}} \equiv \{l_1\dot{v}_1, l_2\dot{v}_2, l_3\dot{v}_3\}$ is the element-wise scalar multiplication of the two vectors. Let the $M^{(\gamma-1)}$ drag/gravity/body acceleration term be written as $\mathbb{D}$, and let $\mathbb{C}$ represents the source term due to collisions. Both of their implementations are described in detail in Section §*E. Source Terms*. Additional moment sink/source terms may be added here as well, but are beyond the scope of this work. Note that the source terms may use a numerical operator-splitting technique[64] so they need not be added verbatim into the PDE as Equation (48) suggests.

Given the general formula by Equation (47), any number of moment transport equations may be constructed. Notice the dilemma however; regardless of the number of equations constructed, closure can never be achieved because each γ'th equation always contains $\gamma + 1$ moments. This is where the previously described quadrature approximation will provide closure[50] while simultaneously applying the convective/flux term, $\nabla_{\boldsymbol{x}} \cdot M^{(\gamma+1)}$, constructed using the set of abscissas and weights.

Since the zero-order moment, $M^{(0)}$, represents the “mass” (area under the f curve) a transport equation for $M^{(0)}$ acts as a *continuity equation*. Any first order moment $M^{(1)} =$

$\int v_d f(t,\boldsymbol{x};\boldsymbol{v})d\boldsymbol{v}$ (of which there are D permutations of d possible) is representative of mean particle momentum, while the second order moments $M^{(2)} = \int v_{d_1} v_{d_2} f(t,\boldsymbol{x};\boldsymbol{v})d\boldsymbol{v}$ (of which there are $2 \cdot D = 6$ permutations of d_1, d_2 possible) are representative of kinetic energy, and so forth. The set of 3-D moment transport equations (up to 5th order) are listed in Equation (49). The number listed to the right indicates the number of permutations of $\{l_1, l_2, l_3\}$, that sum to γ, however, not all permutations are needed for the optimal moment set. A detailed description of the optimal moment selection can be found in Section §*C. Optimal Moment Set*. The first three moment equations represent mass, momentum, and energy conservation[65] respectively,

$$\begin{aligned}
&\tfrac{\partial}{\partial t} M^{(0)}_{0,0,0} + \tfrac{\partial}{\partial \boldsymbol{x}} M^{(1)} = 0 && \text{(Mass Continuity: 1)}\\
&\tfrac{\partial}{\partial t} M^{(1)}_{l_1,l_2,l_3} + \tfrac{\partial}{\partial \boldsymbol{x}} M^{(2)} = \mathbb{D}^{(1)}_{l_1,l_2,l_3} && \text{(Momentum: 3)}\\
&\tfrac{\partial}{\partial t} M^{(2)}_{l_1,l_2,l_3} + \tfrac{\partial}{\partial \boldsymbol{x}} M^{(3)} = \mathbb{D}^{(2)}_{l_1,l_2,l_3} + \mathbb{C}^{(2)}_{l_1,l_2,l_3} && \text{(Energy: 6)}\\
&\tfrac{\partial}{\partial t} M^{(3)}_{l_1,l_2,l_3} + \tfrac{\partial}{\partial \boldsymbol{x}} M^{(4)} = \mathbb{D}^{(3)}_{l_1,l_2,l_3} + \mathbb{C}^{(3)}_{l_1,l_2,l_3} && \text{(3rd-Order: 10)}\\
&\tfrac{\partial}{\partial t} M^{(4)}_{l_1,l_2,l_3} + \tfrac{\partial}{\partial \boldsymbol{x}} M^{(5)} = \mathbb{D}^{(4)}_{l_1,l_2,l_3} + \mathbb{C}^{(4)}_{l_1,l_2,l_3} && \text{(4th-Order: 15)}\\
&\tfrac{\partial}{\partial t} M^{(5)}_{l_1,l_2,l_3} + \tfrac{\partial}{\partial \boldsymbol{x}} M^{(6)} = \mathbb{D}^{(5)}_{l_1,l_2,l_3} + \mathbb{C}^{(5)}_{l_1,l_2,l_3} && \text{(5th-Order: 21)}\\
&\vdots
\end{aligned} \tag{49}$$

The unsteady term, $\frac{\partial}{\partial t} M^{(\gamma)}_{l_1,l_2,l_3}$, is solved at each time step while the remaining terms are either known or closed by the quadrature assumption. Recall that $\mathbb{D}^{(\gamma)}_{l_1,l_2,l_3}$ is the drag/gravity/body force term, and $\mathbb{C}^{(\gamma)}_{l_1,l_2,l_3}$ is the collision term which are both described in section §*E. Source Terms*. In conclusion, the moment-space ($\boldsymbol{M}$) provides a working

PDE for temporal and spatial transport of the moment quantities, and a collision model. The next section provides a numerical discretization that unifies the moments and PDF.

D. Quadrature-Based Moment Methods

The quadrature approximation[8] simplifies $f(\boldsymbol{v},\boldsymbol{\xi})$ into a sum of weighted Dirac delta functions, δ,

$$f(\boldsymbol{v},\boldsymbol{\xi}) \approx \sum_{\alpha=1}^{\beta} \phi_\alpha \delta[\hat{\boldsymbol{v}}_{\boldsymbol{\alpha}} - \boldsymbol{v}]\delta[\hat{\boldsymbol{\xi}}_{\boldsymbol{\alpha}} - \boldsymbol{\xi}] \tag{50}$$

where $\hat{\boldsymbol{v}}_\alpha \equiv \{U_\alpha, V_\alpha, W_\alpha\}$ is the set of velocity abscissas pertaining to node index α containing all $D = 3$ dimensions of the velocity, and ϕ_α is the scalar weight associated with each set of abscissas $\hat{\boldsymbol{v}}_\alpha$. Likewise, vector $\hat{\boldsymbol{\xi}}_\alpha$ are the abscissas of any additional internal coordinates. Note that all of these values are also typically dependent upon the external coordinates $(t,\boldsymbol{x})$. Methods that use this approximation are called quadrature-based moment methods (QBMM). Numerically, it is convenient to store the equivalent information in a node matrix $\mathbb{N}$ as described in Equation (39); in fact, Equation (50) provides the approximate mapping $\mathbb{N} \rightarrow f(\boldsymbol{v},\boldsymbol{\xi})$.

The multivariate case has been presented, and it is straightforward to add more variables to $\boldsymbol{\xi}$ if needed. Additionally, a larger total number of nodes, β, will increase the accuracy of the represented diversity of the integral of f. The total required number of transport equations required for closure depends upon which internal variables are present in the phase space. Recall that the moment set $\boldsymbol{M}$ is determined by integrating the function f over the vectors $\boldsymbol{v}$ and $\boldsymbol{\xi}$ as shown in Equation (45). Conveniently, integrating the Dirac delta functions simply returns the abscissas, $\hat{\boldsymbol{v}}_\alpha$ and $\hat{\boldsymbol{\xi}}_{\boldsymbol{\alpha}}$, when integrated over the entire domain. Also recall that $\boldsymbol{\zeta} \equiv \{\boldsymbol{v}, \boldsymbol{\xi}\}$, thus the general quadrature approximation for any internal vector $\boldsymbol{\zeta}$ is,

$$M^{(\gamma)}_{l_1,l_2,l_3,...} = \int_{\Omega_{\boldsymbol{\zeta}}} f(\boldsymbol{\zeta})\; h_{l_1,l_2,l_3,...}(\boldsymbol{\zeta})\; d\boldsymbol{\zeta} \approx \sum_{\alpha=1}^{\beta} \phi_\alpha \prod_{d=1}^{\mathcal{X}} \hat{\boldsymbol{\zeta}}_d^{l_d} \tag{51}$$

where $h_{l_1,l_2,l_3,...}(\hat{\boldsymbol{\zeta}}) = U^{l_1}V^{l_2}W^{l_3}...$ is the test function, defined also in Equation (93), which returns the product of all the components in $\hat{\boldsymbol{\zeta}}$ raised to their integer powers $l_1, l_2, l_3, ...$ respectively. See Equation (45) for the specific case with only velocity abscissas. Note that after integration, the state variables receive *hats* to indicate they are no longer continuous but are instead discrete abscissas defining the coordinates of the Dirac Delta functions.

In the current work, vector $\boldsymbol{\xi}$ is empty and consequently $\boldsymbol{\zeta} = \boldsymbol{v}$. Various 3-D ordered moments may be constructed by applying varying integer exponents l_1, l_2, l_3 to the individual components of $\hat{\boldsymbol{v}}_\alpha \equiv \{U_\alpha, V_\alpha, W_\alpha\}$, thus the exact Equation (45) is approximated by

$$M^{(\gamma)}_{l_1,l_2,l_3} \approx \sum_{\alpha=1}^{\beta} \phi_\alpha U_\alpha^{l_1} V_\alpha^{l_2} W_\alpha^{l_3} \qquad \gamma = 0, 1, 2, ... \tag{52}$$

The proper choice for values of the exponents are dependent upon which QMOM model is used. For CQMOM, Section $\S$*C. Optimal Moment Set* will describe how to determine which $\{l_1, l_2, l_3\}$ permutations are required to use the optimal system of moment transport equations effectively. Numerically, Equation (52) defines a mapping from nodes to moments which can be written as $\mathbb{N} \rightarrow \boldsymbol{M}$. Any number of ordered moments may be generated from this mapping, defining the process known as *moment construction* in Equation (52). If these moments are to be useful later, one must aim to retain an adequate quantity of moments as to provide a sufficient description of the desired features of f.

Below are some examples of this numerical mapping being used to *construct* typical moments from 2-point quadratures,

$$\begin{array}{ll} (a)\; M^{(0)}_{0,0,0} = \phi_1 + \phi_2 & (b)\; M^{(2)}_{0,2,0} = \phi_1 V_1^2 + \phi_2 V_2^2 \\ (c)\; M^{(1)}_{1,0,0} = \phi_1 U_1 + \phi_2 U_2 & (d)\; M^{(6)}_{3,1,2} = \phi_1 U_1^3 V_1 W_1^2 + \phi_2 U_2^3 V_2 W_2^2 \end{array} \tag{53}$$

where (a) $\{0, 0, 0\}$ represents the total "mass" (volume fraction), (b) $\{0, 2, 0\}$ is the particle y-direction kinetic energy, (c) $\{1, 0, 0\}$ is the particle x-momentum, and (d) $\{2, 1, 2\}$ is a 6th-order moment with no intuitive physical meaning, but nonetheless may retain required information to adequately represent f as a quadrature.

The detailed makeup of matrix $\mathbb{N}$ is shown in Equation (39) by concatenating D abscissa columns together with one weight column, $\mathbb{N} \equiv \{\hat{\boldsymbol{v}}, \boldsymbol{\phi}^T\} \equiv \{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\phi}\}$. This local matrix set of nodes $\mathbb{N}(t, \boldsymbol{x})$ exists for each grid point in the mesh at a distinct time, but the $(t, \boldsymbol{x})$ notation is suppressed for clarity. Expanded specifically for the $D = 3$

and $Q = 0$ case, the weights and abscissas are stored as,

$$\mathbb{N} \equiv \begin{bmatrix} \boldsymbol{N}_1 \\ \boldsymbol{N}_2 \\ \vdots \\ \boldsymbol{N}_{\mathbb{B}} \end{bmatrix} \equiv \begin{bmatrix} U_1 & V_1 & W_1 & \phi_1 \\ U_2 & V_2 & W_2 & \phi_2 \\ & & \vdots & \\ U_{\mathbb{B}} & V_{\mathbb{B}} & W_{\mathbb{B}} & \phi_{\mathbb{B}} \end{bmatrix} \equiv \begin{bmatrix} \boldsymbol{U} & \boldsymbol{V} & \boldsymbol{W} & \boldsymbol{\phi} \end{bmatrix} \equiv \begin{bmatrix} \hat{\boldsymbol{v}}_1 & \hat{\boldsymbol{v}}_2 & \hat{\boldsymbol{v}}_3 & \boldsymbol{\phi} \end{bmatrix} \quad (54)$$

Considering $\mathbb{N}$ as a row matrix, each row ($\alpha = 1, ..., \mathbb{B}$) contains a single "node", $\boldsymbol{N}_\alpha \equiv \{U_\alpha, V_\alpha, W_\alpha, \phi_\alpha\}$, which is a complete set of abscissas and weights, and α is the node/row index. Furthermore, let $\hat{\boldsymbol{v}}_\alpha$ represent the α'th instance (row α of $\mathbb{N}$) of a velocity vector,

$$\hat{\boldsymbol{v}}_\alpha \equiv \{\hat{v}_{1,\alpha}, \hat{v}_{2,\alpha}, \hat{v}_{3,\alpha}\} \quad (55)$$

The maximum quantity of moments, K, will match the number of abscissas and weights being tracked, such that

$$K = (D + Q + 1) \cdot \mathbb{B} \quad (56)$$

ensuring adequate closure in the mapping $\mathbb{N} \rightarrow \boldsymbol{M}$ and its inverse $\boldsymbol{M} \rightarrow \mathbb{N}$, described in the next section. Note also that the computational requirements grow drastically with each additional variable or node corresponding to a total of K transport equations.

E. Moment Inversion

The previous section defined how to use a node matrix to construct a moment vector $\mathbb{N} \rightarrow \boldsymbol{M}$, but it is necessary to also define the inverse transformation which will map the moments back to their respective set of nodes, $\boldsymbol{M} \rightarrow \mathbb{N}$, in a stable and efficient manner. This process known as "moment inversion" is mathematically sophisticated, highly non-linear, and sensitive to instabilities; thus numerous researchers have given the problem great consideration over many decades and improved its stability. Still, the inversion process can become computationally expensive for very large sets of moments/nodes. In general, the number of variables in $\boldsymbol{M}$ and $\mathbb{N}$ must match for proper closure.

Consider the $\beta = 2$ node quadrature; the first four (2β) moments are extremely useful and capable of approximately describing the distribution f by its total weight, mean, variance, and skew. For example, a 1-D, 2-point, x-dimension inversion may be performed in which the four unknowns (two abscissas and two weights) are solved: $\{M^{(0)}_{000}, M^{(1)}_{100}, M^{(2)}_{200}, M^{(3)}_{300}\} \rightarrow \{\phi_1, \phi_2, U_1, U_2\}$. Retaining a larger moment set (including the higher order moments) allows for the description of f with more nodes, if so desired, but the added information from each additional higher order moment becomes less significant while the computational costs become exponentially larger. Described in the next section, are two different noteworthy implementations of the moment inversion method: the PD Algorithm and the Wheeler Algorithm.

1. PD Algorithm

The Product-Difference (PD) Algorithm[66] is an older procedure of inverting a one-dimensional moment set, and applicable to QMOM[8,16,67,68] for solving a population balance equation (PBE). It finds the eigenvalues of a real symmetric tridiagonal matrix, but a severe limitation is that it can only be applied to monovariate quadratures[69] and fails under certain conditions.[70] With some modifications and assumptions, it is also applicable to bivariate quadratures,[42] but cannot typically be used for higher dimension (2D and 3D) internal spaces.[71]

2. Wheeler Algorithm

The 1-D Wheeler algorithm[62] (MATLAB source code published by Yuan *et al.*[44]) is employed in the present work within CQMOM, and is also only capable of inverting a set of 1-D moments. CQMOM however, takes advantage of the Wheeler algorithm by applying a nested sequence of 1-D inversions in each dimension, allowing 2-D and 3-D simulations to be performed. The basic idea involves solving the eigenvalues of a real symmetric tridiagonal matrix to determine the best fit of abscissas and weights that can describe a 1-D subset of the optimal moment set $\boldsymbol{M}$. Efficient formulations for this common type of eigenvalue problem are readily available.[72] The Wheeler algorithm also offers improved stability when dealing with very small weights $\phi \approx 0$ which are very sensitive to division by zero, and maintaining realizable non-zero weights at all times.

Firstly, the algorithm checks *a priori* whether the total "mass", $M^{(0)}$, is large enough to attempt any $\beta > 1$ inversion. This is the first threshold value, $\mathbb{W}_1 \geq 0$, that is user-defined and problem-dependent,

$$M^{(0)} \geq \mathbb{W}_1 \tag{57}$$

If this equality is not satisfied, then only $\beta = 1$ node is allotted to represent the moment set and that single node is given the average velocity $\langle \hat{\boldsymbol{v}}_d \rangle$ and weight $\langle \phi \rangle$,

$$\begin{aligned} \langle \hat{\boldsymbol{v}}_d \rangle &:= M_d^{(1)} / M^{(0)} \\ \langle \phi \rangle &:= M^{(0)} \end{aligned} \tag{58}$$

of the applicable dimension d. This helps reduce the amount of algebra that is performed on very small values which may suffer from numerical round-off error and prevents ill-conditioned situations. If the minimum weight $M^{(0)} \geq \mathbb{W}_1$ *is* satisfied, then the moment-inversion is first performed assuming a maximum total of $\beta = B$ nodes. However, before accepting the results of the first inversion, the weights and abscissas go through a quality control test. If the resulting nodes are deemed unsuitable by the test, then the total node count is reduced by 1, such that $\beta = B - 1$, and the inversion process is repeated to find a new set of weights and abscissas until $\beta = 1$ is reached.

The second quality control check is performed on the relative weights of each node to ensure that they are similar in order of magnitude. The fraction resulting from the smallest weight divided by the largest weight yields the weight ratio,

$$\frac{\min(\phi_1, ..., \phi_B)}{\max(\phi_1, ..., \phi_B)} \geq \mathbb{W}_B \tag{59}$$

Note that each B-point quadrature inversion may have a its own unique ratio, $\mathbb{W}_2, ..., \mathbb{W}_\beta$, where $B = 2, ..., \beta$. If the relation is not satisfied, at least one of the nodes in the set is considered "negligible" because its weight is significantly smaller than the largest weight of another node. Neglecting abscissas helps ensure that f is represented efficiently and reserves room for a possibly more significant weight that may arise in the future.

Next, a test is performed after the inversion to ensure that the abscissa coordinates are not too close together or "redundant". The first test is only applicable to $\beta \geq 3$, and uses the ratio of the smallest and largest distances between any two abscissas,

$$\frac{\min(\Delta\hat{v}_{d,1}, ..., \Delta\hat{v}_{d,B})}{\max(\Delta\hat{v}_{d,1}, ..., \Delta\hat{v}_{d,B})} \geq \mathbb{A}_B \tag{60}$$

where $\mathbb{A}$ is the 3rd user-defined threshold value, and $\Delta\hat{v}_{d,\alpha} = |\hat{v}_{d,\alpha_1} - \hat{v}_{d,\alpha_2}|$ represents all possible combinations of the difference between node α_1 and α_2 where $\alpha_1 \neq \alpha_2$ and $\alpha_2 = 1, ..., B$. Note that this check should be performed among similar abscissas only, i.e. do not compare velocity abscissas to particle diameters, temperatures, etc., which may operate at entirely different scales with different units. Each B-point quadrature inversion may have a its own unique ratio $\mathbb{A}_3, ..., \mathbb{A}_\beta$, where $B = 3, ..., \beta$ to offer more flexibility.

Lastly, and specifically for a 2-point inversion, another "redundancy" check ensures that the two abscissas remain distinct enough to avoid ill-conditioned matrices that may lead to nearly singular matrix inversions. If the two abscissas are too close together then they should be reduced into a single-node representation. This final threshold known as the *abscissa proximity ratio*, $\mathbb{P}$, is determined by the ratio of the larger abscissa divided by the smaller one,

$$1 \leq \max\left(\frac{\hat{v}_{d,1}}{\hat{v}_{d,2}}, \frac{\hat{v}_{d,2}}{\hat{v}_{d,1}}\right) \leq \mathbb{P} \tag{61}$$

If the condition in Equation (61) is satisfied, then a 2-point quadrature is ill-conditioned and a very simple 1-point quadrature should be attempted instead. Note that abscissas of opposite signs will always succeed regardless of how similar their magnitudes are. The proximity ratio ($\mathbb{P} > 1$) must be chosen depending on the problem. For the 2-D Taylor-Green flow a value of $\mathbb{P} = 1.05$ was sufficient to prevent asymmetries. See Section §*A. Multivariate Conditional Inversion* for more details about multidimensional inversions where $\mathbb{P}$ is a vector.

Simplifying the quadrature representation to fewer nodes will reduce the spread of a PDF and impact the statistics. In particular, forcing a 1-point assumption instead of a 2-point quadrature representation will reduce the RMS slightly. To determine how large the effect is, consider a 2-point representation with equal weights of $\phi_1 = \phi_2$, which yields the largest spread possible for two arbitrary abscissas $\hat{v}_{d,1}$ and $\hat{v}_{d,2}$. Then assume that the two velocities are at their maximum distinctness, $\hat{v}_{d,2} = \mathbb{P}\hat{v}_{d,1}$ according to the proximity ratio assumption, which will generate the largest possible error. The lower order moments, $M^{(0)}$ and $M^{(1)}$, of these two different representations ($\beta = 1$ vs. $\beta = 2$) are identical, and only the higher order moments $M^{(2)}$, $M^{(3)}$, etc. will be affected. Substituting those node values into the equation for the second order moment of a 2-point quadrature produces,

$$M^{(2)}_{\beta=2} = \phi_1 \hat{v}^2_{d,1} + \phi_2 \hat{v}^2_{d,2} = \phi_1 \hat{v}^2_{d,1}(1 + \mathbb{P}^2) \tag{62}$$

By then substituting a single average velocity, $\frac{\hat{v}_{d,1}+\hat{v}_{d,2}}{2}$, and total weight for the equivalent 1-point approximation and calculating $M^{(2)}$ again,

$$M^{(2)}_{\beta=1} = (\phi_1 + \phi_2)\left(\frac{\hat{v}_{d,1} + \hat{v}_{d,2}}{2}\right)^2 = \frac{\phi_1}{2}\hat{v}^2_{d,1}(1 + \mathbb{P})^2 \tag{63}$$

the ratio of Equation (62) and Equation (63) can be taken. This reveals that the 2-point representation (numerator) will always yield a larger $M^{(2)}$ than the 1-point representation (denominator),

$$\frac{M^{(2)}_{\beta=2}}{M^{(2)}_{\beta=1}} = \frac{2(1 + \mathbb{P}^2)}{(1 + \mathbb{P})^2} \geq 1 \tag{64}$$

As $\mathbb{P} \rightarrow 1$, the error vanishes and the two representations become equivalent (although the two abscissas are perfectly redundant and singular). However, even for fairly large $\mathbb{P}$, the relative error remains small, thus making it a very useful assumption for drastically improving the stability of the CQMOM inversion algorithm.

Through this “process of elimination” any redundant, ill-conditioned, and/or negligible nodes will flag the entire set of weights and abscissas as “unsuitable” and then attempt to perform a lower order $1 \leq \beta \leq B$ inversion. Once the threshold criteria of $\mathbb{W}_\alpha$ is met in Equations (57) and (59), and $\mathbb{A}$ of Equation (60), or no further reduction is possible, i.e. having reached $\beta = 1$ (single node), then the inversion is complete. Note that in the case of $\beta = 1$ being reached, the "inversion" is simplified by Equation (58) needing only the first two moments.

To summarize, these four user-defined threshold values exist to allow the problem-dependent accuracy and error to be balanced:

Table 6. Thresholds For Wheeler Inversion

$\mathbb{W}_1$	Minimum weight required to attempt any $\beta > 1$ moment inversion
$\mathbb{W}_{2...\beta}$	Minimum weight ratio required to keep additional α'th node
$\mathbb{A}$	Minimum ratio of distances between abscissas (valid for $\beta \geq 3$ nodes)
$\mathbb{P}$	Minimum ratio between two abscissas (valid for $\beta = 2$ nodes only)

Using small threshold values makes the process more likely to reward additional nodes, but if the moments contain errors caused by numerical diffusion, then the nodes can be detrimentally exhausted to represent that error. Therefore, it is advantageous to maintain threshold values that are small enough to retain important flow features (such as $\mathbb{W}_1$ being $1E^{-3}$ times smaller than the maximum initial weight), but large enough that spurious errors are neglected. Values of $\mathbb{W}_2 \approx 1E^{-2}$ and $1.01 \leq \mathbb{P} \leq 1.3$ have shown great success in the present work.

3. Algorithm

To summarize the 1-D inversion process, Figure 10 depicts a flowchart showing the general process of reducing the order of the α-point inversion until a suitable set of nodes is obtained, as described previously. Both these preemptive and corrective strategies help minimize the likelihood of ill-conditioned inversions and resulting abscissas. The CFL number may also be used as another consideration to determine whether an abscissa is erroneously large.

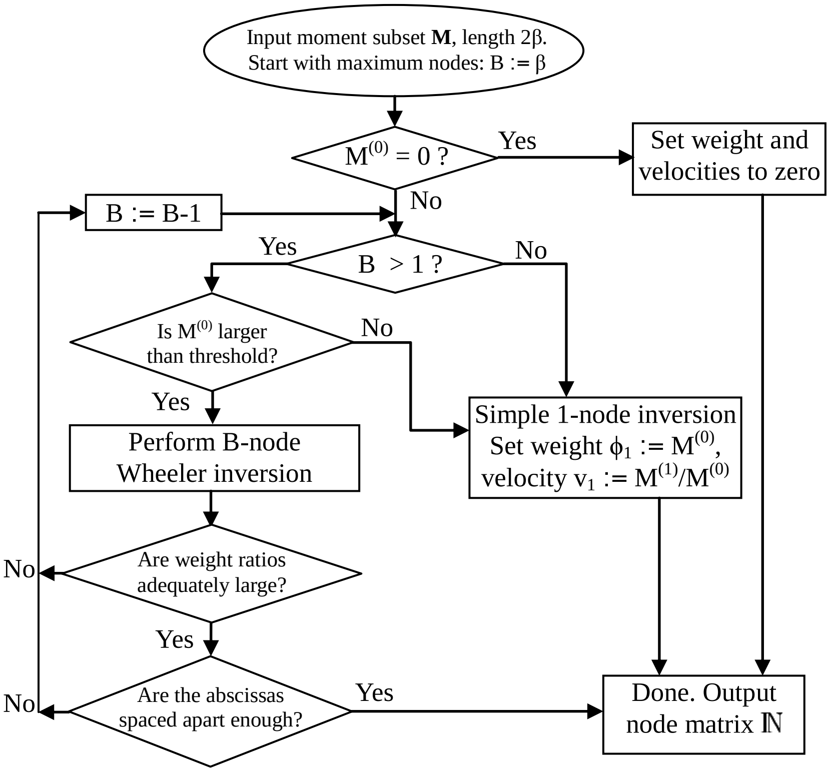

Input moment subset M, length 2β.
Start with maximum nodes: B := β
M(0) = 0 ?
Yes
Set weight and velocities to zero
No
B := B-1
Yes
B > 1 ?
No
Is M(0) larger than threshold?
No
Yes
Perform B-node Wheeler inversion
Simple 1-node inversion Set weight ϕ1 := M(0), velocity v1 := M(1)/M(0)
No
Are weight ratios adequately large?
Yes
No
Are the abscissas spaced apart enough?
Yes
Done. Output node matrix ℕ

Figure 10. 1-D Adaptive β-Point Inversion Algorithm Flowchart

F. Temperature

When all the particles in an ensemble possess the same state value then their PDF has zero variance ($\sigma^2 = 0$); therefore, f is a single Dirac Delta function and is fully represented by a 1-point quadrature. As the diversity of a distribution increases, so too does the variance, and the PDF "spreads-out" ($\sigma^2 > 0$). At some point, a quadrature representation will require two nodes (or more) to describe the PDF sufficiently. If the PDF represents velocity space, then a non-zero variance may cause particles to experience collisions if the particle concentration is dense enough. By convention, a granular temperature has units of velocity squared so it is synonymous with the variance, $\Theta \equiv \sigma^2$.

The scalar granular temperature, Θ_{eq}, can be used as an approximation for an isotropic collision state, by using the average of all D temperature components: $\Theta_{eq} = \frac{1}{D}\sum_1^D \Theta_d$.[33,36,44,45] The granular temperature in dimension d may also be obtained from the first three *simple* moments: mass, momentum, and energy,

$$\sigma_d^2 = \Theta_d = \frac{M_d^{(2)}}{M^{(0)}} - \left(\frac{M_d^{(1)}}{M^{(0)}}\right)^2 \tag{65}$$

as previously described by Equation (46) for $M_{l_1 l_2 l_3}^{(2)}$ and $M_{l_1 l_2 l_3}^{(1)}$. As $M_d^{(2)} \rightarrow (M_d^{(1)})^2/M^{(0)}$ the distribution approaches Dirac Delta distribution, adhering to the limit of $M_d^{(2)} = (M_d^{(1)})^2/M^{(0)}$ exactly, while all other forms where $\Theta_d > 0$ will result in $M_d^{(2)} > (M_d^{(1)})^2/M^{(0)}$. This is a very important constraint for ensuring realizability and that the moment set retains a physical meaning. It is also the cause of numerical instabilities that are discussed in Section §*3. Moment Corruption*.

1. Velocity Covariance

A temperature also exists for every combination of dimensions (*mixed* moments) and can be fully represented by the $D \times D$ *velocity covariance matrix*, $\boldsymbol{\theta}$, shown for $D = 3$ below,[52]

$$\boldsymbol{\theta} = \begin{bmatrix} M_{200}^{(2)}/M^{(0)} - \langle v_1 \rangle^2 & M_{110}^{(2)}/M^{(0)} - \langle v_1 \rangle \langle v_2 \rangle & M_{101}^{(2)}/M^{(0)} - \langle v_1 \rangle \langle v_3 \rangle \\ M_{110}^{(2)}/M^{(0)} - \langle v_1 \rangle \langle v_2 \rangle & M_{020}^{(2)}/M^{(0)} - \langle v_2 \rangle^2 & M_{011}^{(2)}/M^{(0)} - \langle v_2 \rangle \langle v_3 \rangle \\ M_{101}^{(2)}/M^{(0)} - \langle v_1 \rangle \langle v_3 \rangle & M_{011}^{(2)}/M^{(0)} - \langle v_2 \rangle \langle v_3 \rangle & M_{002}^{(2)}/M^{(0)} - \langle v_3 \rangle^2 \end{bmatrix} \tag{66}$$

where $\{\langle v_1\rangle, \langle v_2\rangle, \langle v_3\rangle\}$ are the cell-averaged particle velocities. The cell-averaged node vectors can be determined directly from the set of optimal moments, $\boldsymbol{M}$, by,

$$\begin{bmatrix} \langle v_1\rangle \\ \langle v_2\rangle \\ \langle v_3\rangle \\ \langle \phi\rangle \end{bmatrix}^T = \begin{bmatrix} M^{(1)}_{1,0,0}/M^{(0)} \\ M^{(1)}_{0,1,0}/M^{(0)} \\ M^{(1)}_{0,0,1}/M^{(0)} \\ M^{(0)} \end{bmatrix}^T = \begin{bmatrix} M^{(1)}_{1}/M^{(0)} \\ M^{(1)}_{2}/M^{(0)} \\ M^{(1)}_{3}/M^{(0)} \\ M^{(0)} \end{bmatrix}^T \tag{67}$$

Additionally, the diagonals of the matrix, $\boldsymbol{\theta}$, are the previously defined temperature vector, $\Theta_d \equiv \theta_{d,d}$. Some collision models work on a simplifying assumption that only requires the diagonals for scalar equilibrium granular temperature (Θ_{eq}), and is obtained by averaging the diagonals of the covariance matrix, $\boldsymbol{\theta}$.

$$\Theta_{eq} = \frac{1}{D}\sum_{d}^{D}\theta_{d,d} = \frac{tr(\boldsymbol{\theta})}{D} \overset{D=3}{\Rightarrow} \frac{(\theta_{1,1}+\theta_{2,2}+\theta_{3,3})}{3} \tag{68}$$

which is expanded specifically in 3-D to,[36, 39]

$$\Theta_{eq} = \frac{1}{3}\left\{\left(\frac{M^{(2)}_{200}}{M^{(0)}} - \langle v_1\rangle^2\right) + \left(\frac{M^{(2)}_{020}}{M^{(0)}} - \langle v_2\rangle^2\right) + \left(\frac{M^{(2)}_{002}}{M^{(0)}} - \langle v_3\rangle^2\right)\right\} \tag{69}$$

If the individual components $\theta_{d,d}$ vary greatly, then this simplified equilibrium granular temperature (Θ_{eq}) may not be a good assumption for some models. Note that $\theta_{d,d} \geq 0$, so each of the D diagonal terms of the matrix that appear in Equation (69) must be non-negative.

2. *Root-mean-square*

Recall that the “root-mean-square” (RMS) is defined as,

$$\text{RMS} = \sqrt{\overline{v'^2}} \tag{70}$$

where,

$$v'(t) = v(t) - \overline{v} \tag{71}$$

is the instantaneous velocity fluctuation at a given sample time. The time-averaged velocity is $\overline{v} = \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} v(t_n)$, where $\mathcal{N}$ is the number of time samples and n is the time index. The RMS depends on the square of the velocity fluctuation, $(v')^2$, to prevent cancellation between positive and negative velocities during averaging. Consider temporarily the “mean-square” (RMS^2) to suppress the square root in the following derivation,

$$\text{RMS}^2 = \overline{v'^2} = \overline{(v - \overline{v})^2} \tag{72}$$

which has units of velocity squared, a mass-less kinetic energy. A simpler form of the RMS^2 can be computed to aid in post-processing. Squaring both sides of Equation (71) and then time-averaging yields,

$$\begin{aligned} v^2 &= (v' + \overline{v})^2 \\ v^2 &= v'^2 + 2v'\overline{v} + \overline{v}^2 \\ \overline{v^2} &= \overline{v'^2} + \overline{v}^2 \end{aligned} \tag{73}$$

because $\overline{2v'\overline{v}} \Rightarrow 2\overline{v}\overline{v'} \Rightarrow 0$ and $\overline{\overline{v}^2} \Rightarrow \overline{v}^2$. This new alternate form for the RMS may be more suitable for numerical post-processing,

$$\text{RMS} = \sqrt{\overline{v'^2}} = \sqrt{\overline{(v - \overline{v})^2}} \equiv \sqrt{\overline{v^2} - \overline{v}^2} \tag{74}$$

since it requires tracking only the velocity ($\overline{v}$) and velocity squared ($\overline{v^2}$) during time-sampling. This implementation of Equation (74) for a DPS population or a DNS fluid field is straightforward, however, the CQMOM particle field requires some additional consideration. The Eulerian particle field contributions to the RMS are twofold: one is the typical cell-averaged velocity deviation, and the 2nd is the contribution due to positive variance in the PDF. Moments are representative of cell-averaged quantities that represent a multitude of fictive particles, and the cell-averaged velocity is explicitly defined as,

$$\overline{\langle v_d \rangle} = \overline{\left(\frac{M_d^{(1)}}{M^{(0)}} \right)} = \overline{\left(\frac{\sum_{\alpha=1}^{\mathbb{B}} \phi_\alpha \hat{v}_{d,\alpha}}{\sum_{\alpha=1}^{\mathbb{B}} \phi_\alpha} \right)} \tag{75}$$

and the cell-averaged square of the velocity is,

$$\overline{\langle v_d^2 \rangle} = \overline{\left(\frac{M_d^{(2)}}{M^{(0)}} \right)} = \overline{\left(\frac{\sum_{\alpha=1}^{\mathbb{B}} \phi_\alpha \hat{v}_{d,\alpha}^2}{\sum_{\alpha=1}^{\mathbb{B}} \phi_\alpha} \right)} \tag{76}$$

Notice that $\overline{\langle v_d \rangle}^2 \neq \overline{\langle v_d^2 \rangle}$ and depend uniquely upon the first and second order moments respectively (both also require the zero order moment). In fact, it is guaranteed that $\overline{\langle v_d \rangle}^2 \leq \overline{\langle v_d^2 \rangle}$ because during PTC the cell-average velocity may shrink toward zero when oppositely signed velocities interfere destructively, i.e. cancel-out. Conversely, $\langle v_d^2 \rangle = M_d^{(2)}/M^{(0)}$ represents the total kinetic energy divided by the total mass and ensures that all quadrature nodes within a cell are added constructively (no canceling occurs). Furthermore, the difference between the two yields the cell-localized PDF variance, $\langle v_d^2 \rangle - \langle v_d \rangle^2 = \sigma_d^2 \geq 0$ such that the RMS – sans the "root-mean" – receives a contribution simply from the instantaneous velocity diversity present in the spatial average (no time-averaging required).

To conclude, the computation of the RMS$_d^2$ for DNS (where $\langle v_d \rangle$ is known directly), DPS, or in a moment-based method are written explicitly as,

$$\text{RMS}_d^2 = \overline{\langle v_d^2 \rangle} - \overline{\langle v_d \rangle}^2 = \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \overset{(n)}{\langle v_d^2 \rangle} - \left(\frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \overset{(n)}{\langle v_d \rangle} \right)^2 =$$
$$\frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \frac{\overset{(n)(2)}{M_d}}{\overset{(n)(0)}{M}} - \left(\frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \frac{\overset{(n)(1)}{M_d}}{\overset{(n)(0)}{M}} \right)^2 \quad (77)$$

where $\mathcal{N}$ is the total sample count at the different time step samples n. Notice that $\overline{\langle v_d^2 \rangle} - \overline{\langle v_d \rangle}^2$ appears *similar* to a time-averaged version of the granular temperature $\boldsymbol{\Theta}$ defined in Equation (65).

3. Moment Corruption

Given a realizable moment set, $\boldsymbol{M}$, the Wheeler inversion algorithm will yield realizable nodes, i.e. positive weights. However, since the flux computations directly alter the moments, each individual moment in the set is susceptible to different magnitudes of truncation errors when applying higher order discretization schemes for the convective term (see results in Section §*1. Moment Misalignment*). When a moment set slowly becomes misaligned, unrealizable, or unphysical, this is known as *moment corruption*, and sometimes small errors in the moment set can lead to large errors in the inverted nodes. Corruption threatens the stability of the solver and so it is important to check that the moment set is valid before attempting an inversion and that the resulting nodes obtained from the inversion are also feasible. In the worst cases, a very large unphysical velocity abscissa may be produced during inversion, which may compromise the simulation as the CFL condition may be violated and its treatment is described in Section §*7. CFL Condition*.

Recall from Equation (65) that the standard deviation σ_d^2 in dimension d is related to the granular temperature, $\Theta_d = \sigma_d^2$, and is an extremely important value which *must* equal zero for a quadrature exactly represented by a single Dirac Delta function. Therefore, this gives a constraint for the first three moments,

$$M_d^{(2)} \geq \left(M_d^{(1)}\right)^2 / M^{(0)} \tag{78}$$

If the equality of Equation (78) is violated then the moment set is unphysical. If the error is small it may simply be due to round-off error, but in a more severe case it can be caused by larger errors in the flux scheme. In the event that $\Theta_d < 0$ occurs, a 2-point inversion may produce unphysical abscissas or possibly negative weights, however, the Wheeler algorithm checks for this violation before attempting it. Problems arise in the collision term ($\mathbb{C}$) which relies directly on the square root of the variance and would yield an imaginary standard deviation if ($\sigma_d^2 < 0$). This type of error is detectable and easy to correct by simply assuming $\sigma_d^2 = 0$ in the case of small round-off errors.

Furthermore, if the temperature is positive, but extremely small, $0 < \Theta_d \ll 1$, this poses an ill-conditioned system for the moment inversion process, specifically when also accompanied by a non-zero skew.[50] In fact, the smaller $\Theta_d \rightarrow 0$ is, the more ill-conditioned the inversion becomes. This has been termed the "frontier" of the moment set by Chalons *et al.*,[50] and occurs whenever two abscissas approach the same value, $\hat{v}_{d,1} \rightarrow \hat{v}_{d,2}$, for $\beta \geq 2$. Yuan *et al.*[44] employed a numerical `cutoff` value in their algorithm described in §*2. Wheeler Algorithm*; however, an alternative detection method, the "Abscissa Proximity Ratio" (also described in the same section) also helps prevent singularity instability conditions.

Test cases have shown that the inversion process can yield abscissas larger than what should be physically possible. When the skew is exactly zero such that the PDF is a symmetric Gaussian then a 2-point inversion is well-behaved regardless of temperature since equal weight (half the total $M^{(0)}$) is given to both abscissas. However, when the PDF is *nearly* a Dirac Delta ($\Theta_d \approx 0$) and a non-zero skew is also imposed, the inverted PDF exhibits a very long tail where one abscissa is located very near the mean ($M^{(1)}/M^{(0)}$) and the other abscissa is a very large unphysical distance from the mean with a small weight. Figure 11 shows sample 1-D inversions involving the equal and opposite abscissas with slightly skewed weights given by Equation (79).

$$\begin{bmatrix} U_1 & \phi_1 \\ U_2 & \phi_2 \end{bmatrix} = \begin{bmatrix} -0.4 & 0.05 \\ 0.4 & 0.045 \end{bmatrix} \tag{79}$$

$$\{M^{(0)}, M^{(1)}, M^{(2)}, M^{(3)}\} = \{0.095, -0.002, 0.0152, -0.00032\} \tag{80}$$

By varying only the $M^{(2)}$ moment and displaying the 2-point inversion results in

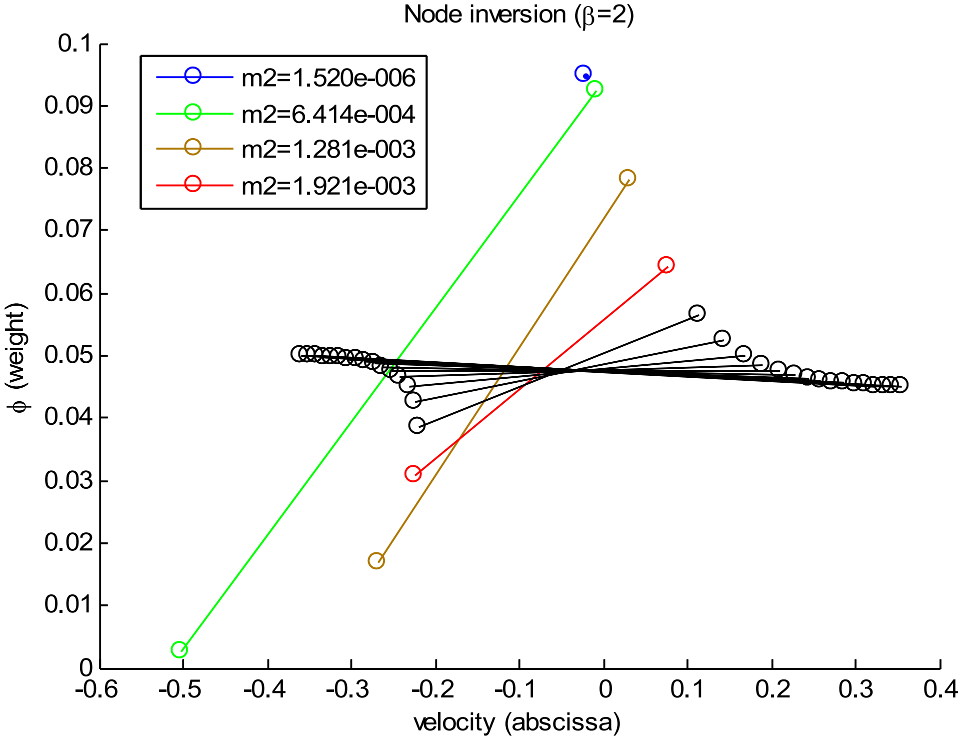


Figure 11. Near Zero Temperature Induced Inversion Instabilities. $M^{(2)}$ Values Shown for the Lowest Four Samples Only.

Figure 11, the instability located near zero temperature is clearly illustrated. By fixing the other moments, the total weight ($M^{(0)} = \phi_1 + \phi_2$), mean momentum ($M^{(1)} = \phi_1 U_1 + \phi_2 U_2$), and 3rd order moment ($M^{(3)} = \phi_1 U_1^3 + \phi_2 U_2^3$) are preserved. For large temperatures (and small skew), the weights are fairly evenly distributed where an inflection point is visible occurring as the skew transitions from negative to positive. Zero-skew is represented by a horizontal line where both weights are equal ($\phi_1 = \phi_2$). The instability occurs in the limit $M^{(2)} \rightarrow (M^{(1)})^2/M^{(0)}$, which is the value required for zero temperature. Figure 11 shows that approaching this value results in one abscissa located near $U_1 = 0$, while the second abscissa is located near $U_2 = -0.5$, larger in magnitude than either of the two original abscissas present in Equation (79). In other words, a very small temperature may result in a very large abscissa, and this is detrimental even if the associated weight, $\phi_2 \approx 0.003$, is small.

Specifically for multidimensional CQMOM (see Section §*V. CQMOM*), an erroneously large velocity abscissa may violate the CFL condition, but more importantly may adversely impact all subsequent nested conditional inversions on the same "branch" (see Figure 13) for other dimensions. When the temperature reaches exactly zero, the 2-point inversion simply reduces to a single node as the PDF is a Dirac Delta function. The best course of action when encountering extremely small (or negative) temperatures is to perform only a 1-point inversion, which preserves $M^{(0)}$ and $M^{(1)}$, while redefining $M^{(2)}$ and $M^{(3)}$ accordingly, i.e. during moment reconstruction the higher order moments ($M^{(2)}$ and $M^{(3)}$) are automatically repaired to represent zero temperature and zero skew. This method is effective at increasing the stability of the simulation, but at the cost of reducing the range of PDF's that can be accommodated. Note that the Wheeler inversion weight ratio threshold ($\mathbb{W}_1$) could prevent this erroneous abscissa from forming, but again, it might then prevent legitimate abscissas from forming too. A threshold weight ratio of $\mathbb{W}_1 \approx 5E^{-2}$ would suffice in this case (Figure 11), which produces very stable simulation results, but is sometimes too restrictive in turbulent flows. A reliable (though not versatile) method of detecting erroneous velocity abscissas is to check that they remain well under the CFL limit.

4. Moment Correction

Some methods exist to correct moment sets, but they are limited in applicability to PDF's which are known to be log-normal distributions. Wright[10] demonstrates how to repair a moment set to adhere to a log-normal distribution that best describes an equilibrium distribution. It is capable of using the lower-order moments, which are more critical of maintaining physical meaning, to correct the higher order moments which are more easily corrupted and less vital to maintaining the physical solution. McGraw[1] also published a scheme to repair a moment set into a realizable quadrature, however, neither of these methods are applicable to the general situations where abscissas may be negative. Other arbitrary distributions such as bimodal functions[63] occurring in PTC cannot be readily corrected in that manner, but ensuring that $M^{(0)} \geq 0$ and the equality in Equation (78) are both met, are two quality-control tests applicable to *any* moment set. Section §*3. Moment Corruption* describes in significant detail how moment corruption may be detected and resolved, especially for the 2-point case.

Lastly, a collision source term ($\mathbb{C}$) should have the natural effect of restoring a moment set to a Gaussian equilibrium distribution while preserving the "mass" and mean velocity. Therefore, a simulation with a collision model should exhibit improved stability.

A carrier fluid drag source term ($\mathbb{D}$) may generate instabilities in a dispersed phase by encouraging the formation of small granular temperatures in the velocity-space PDF. It does so by attempting to align each of the diverse particle node velocity vectors to match that of the fluid field, effectively reducing the temperature toward zero; however, this also reduces instabilities by guiding erroneous abscissas toward proper values (near the fluid velocity value).

G. DQMOM

Direct QMOM (DQMOM) reduces computational requirements because it avoids the need for the moments entirely and directly tracks the abscissas and weights,[13,68] thus removing the moment-inversion algorithm. Consequently, this simplification prevents PTC from being realized, but this method is effective and efficient for low Stokes number flows while still allowing multiple classes of particles to be tracked. DQMOM is usually only capable of modeling monovariate cases, with some exceptions.[71] Typically the particle diameter[73] ($\xi_1 = L_p$) is assigned to the single variable. Mathematically DQMOM is equivalent to QMOM,[74] though limited in applicability. For example, a monovariate model (where $\boldsymbol{\xi} \equiv \xi_1$ is scalar) requires a very minimal number of equations for closure. In fact, only the 2β lowest order moment equations are required, where β is the total number of classes being tracked.

DQMOM has further limitations because it can only be used to transport passive variables in $\boldsymbol{\xi}$ such as particle diameter, temperature, etc., not the particle velocity $\boldsymbol{v}$, which is non-passive, and requires another unique governing equation entirely. A separate kinetic theory can be used to describe the velocity transport[47,48] of each class individually. So although multiple classes of particles may exist where each has its own velocity, the individual velocity fields all act independently, preventing PTC from occurring within a single class. Adding another $\beta \cdot D$ variables at each grid point accommodates the particle velocity vector for each particle class. In total, $\beta \cdot (D+2)$ variables are required at each grid point to store β classes in a D-dimensional velocity space (the additional two are from the weight and single non-velocity abscissa of each class).

1. Derivation

The moments do not appear in the final implementation of DQMOM, but they are required for the derivation. Since only one internal variable exists for this DQMOM example ($Q = 1$), it suffices to use a single subscript to define the moment $M_{l_1}^{(\gamma)}$ (where $\gamma \equiv l_1$ is implied) computed over all particle classes (of which there are β total classes). The general Equation (52) can be applied by assuming $l_2 = 0$ and $l_3 = 0$. Specifically, selecting the first 2β integer values starting at 0 for l_1 produces the greatest accuracy (Gaussian quadrature). For example, the first four moments (up to third-order) for a $\beta = 2$-point quadrature, would yield

$$\begin{aligned} M_0^{(0)} &= \phi_1 + \phi_2 \quad ; \quad M_1^{(1)} = \phi_1\xi_1 + \phi_2\xi_2 \\ M_2^{(2)} &= \phi_1\xi_1^2 + \phi_2\xi_2^2 \quad ; \quad M_3^{(3)} = \phi_1\xi_1^3 + \phi_2\xi_2^3 \end{aligned} \tag{81}$$

Note that this setup is independent of the number of spatial dimensions in the model, but for a 2-D example these first four moments will suffice. These moments are also applicable in the original QMOM. As discussed, the velocity vector remains an external variable to simplify the moment setup, i.e. only one subscript is required to describe a moment, which is the abscissa for the one internal variable ξ, e.g. particle diameter. The particle velocity vectors $\boldsymbol{v}$ are left as externalized variables which are conditioned by the diameter, i.e. drag force depends on particle size. Therefore, the particle velocity field will be guided by the fluid velocity vectors $\boldsymbol{u}$ through the drag force and any other forcing functions such as stress, collisions, etc. The particle volume fraction in a cell (ϕ) and the diameter of the particles (ξ), both will influence the particle velocity evolution. In return, the particle velocity will convect the particle diameter and weight correspondingly.

2. *Granular kinetic theory*

One method of handling an externalized particle velocity field is to employ the granular kinetic theory transport equation. For class α and spatial dimension d (use Einstein Summation for dimension k), the general equation is

$$\rho_\alpha \eta_{p,\alpha} \left[\frac{\partial}{\partial t}(q_\alpha \hat{v}_{d,\alpha}) + \frac{\partial}{\partial x_k}(q_\alpha \hat{v}_{d,\alpha} \hat{v}_{k,\alpha}) \right] = q_\alpha F_{d,\alpha} + q_\alpha \rho_\alpha \eta_{p,\alpha} g_d \tag{82}$$

where $\boldsymbol{F}_\alpha$ is the fluid-particle interaction force vector on a single particle and $\eta_{p,\alpha}$ is the volume of a single particle in class α. This transport equation will predict the particle velocity field evolution. Using the relation $\phi_\alpha \approx q_\alpha \eta_{p,\alpha}$, Equation (82) may be rewritten as

$$\frac{\partial}{\partial t}(\phi \hat{v}_{d,\alpha}) + \nabla_x \cdot (\phi \hat{v}_{d,\alpha} \boldsymbol{v}_\alpha) = \frac{q_\alpha F_{d,\alpha}}{\rho_\alpha} + \phi g_d \tag{83}$$

Closure equations for the drag force can be found in Section §*B. Drag Force*.

3. Transport Equations

Recall that DQMOM reduces CPU usage by tracking the abscissas and weights (ξ and ϕ) directly and without the need for the moments explicitly. In the specific case where there is no production, breakage, aggregation, or collisions, the RHS becomes 0, and the conservation equations reduce to

$$\frac{\partial \phi_\alpha}{\partial t} + \nabla_x \cdot (\phi_\alpha \boldsymbol{v}_\alpha) = 0 \tag{84}$$

$$\frac{\partial \phi_\alpha \xi_\alpha}{\partial t} + \nabla_x \cdot (\phi_\alpha \xi_\alpha \boldsymbol{v}_\alpha) = 0 \tag{85}$$

These transport equations are representations of those in the original QMOM approach, where Equation (84) corresponds to the zero-order moment and Equation (85) to the first order moment, but at no point is a moment-inversion necessary, and so the moments may be disregarded entirely.

V. CQMOM

Conditional QMOM (CQMOM) is an extremely powerful model because it can theoretically accommodate any number of nodes ($\beta > 1$) for a *multivariate* internal space ($\mathcal{X} > 1$). It accomplishes this task by selectively performing a sequence of nested 1-D moment inversions with a "dimensional-splitting" formulation, henceforth termed "conditional-splitting". Each unique 1-D inversion demands a subset of 2β moments, and the union of all the subsets is the complete *optimal* moment set, $\boldsymbol{M}$. Additionally, the optimal moment set is efficient because the 1-D inversions overlap by sharing many of the same core moments.

A. Multivariate Conditional Inversion

A multivariate moment inversion $\boldsymbol{M} \to f(\boldsymbol{v}, \boldsymbol{\xi})$ is possible through a set of multiple (nested) 1-D inversions, i.e. a "conditional-splitting" technique. The splitting is defined by the conditional inversion sequence vector $\boldsymbol{b} = [d_1, d_2, d_3, ..., d_{\mathcal{X}}]$, that describes which order to perform the full multivariate inversion in. Note that the *square* bracket notation indicates a sequence vector. The current work uses the trivariate, 3-D case ($D = 3$ and $Q = 0$), where six different permutations of the conditional inversion sequence exist,

$$[1,2,3]\,;[1,3,2]\,;[2,1,3]\,;[2,3,1]\,;[3,1,2]\,;[3,2,1] \tag{86}$$

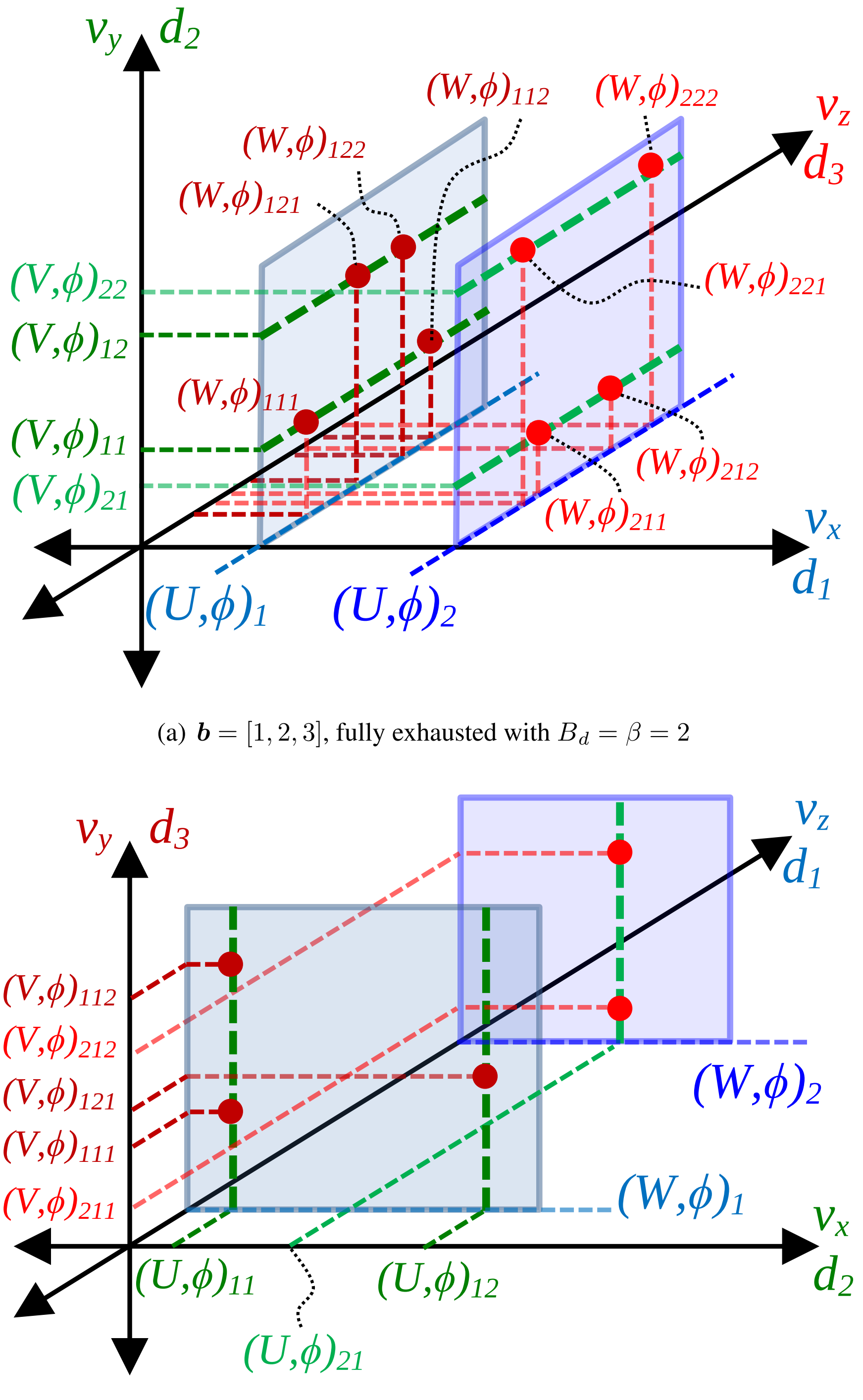
Choosing a proper multidimensional conditional sequence is *not* a trivial task, but the simplest example performs a velocity moment inversion in the x,y,z dimensions consecutively (in that specific order), denoted as $\boldsymbol{b} = [1,2,3] \equiv [x,y,z]$. With $\mathcal{X}$

elements, the vector $\boldsymbol{b}$ has $\mathcal{X}!$ possible permutations. Therefore, the jump from 2-D to 3-D adds significantly more computations, and $Q > 0$ even more so. Note that a 2-D simulation is much simpler, since only two permutations exist: $[1, 2]\,; [2, 1]$. Performing a *multidimensional* (3-D) moment inversion consists of computing three 1-D inversions[45] in the d_1 then d_2 then d_3 direction, denoted by $\boldsymbol{M} \xrightarrow{[d_1,d_2,d_3]} \overset{[d_1]}{\mathbb{N}}$. The centered superscript on $\overset{[d_1]}{\mathbb{N}}$ indicates the *first* conditional dimension in the sequence, $\boldsymbol{b} = [d_1, d_2, d_3]$, and its importance is described in a later section.

To begin the multidimensional inversion, first the d_1 dimension is inverted to obtain a quadrature representation of PDF $f(v_{d_1})$, i.e. nodes. Second, the d_2 velocities are computed on the condition that d_1 nodes are known, $f(v_{d_2}|v_{d_1})$, noting that this must be repeated for all B_1 nodes that exist in $f(v_{d_1})$. Lastly, the d_3 moment inversion is computed on the conditioned nodes of d_1 *and* d_2, yielding $f(v_{d_3}|v_{d_1}, v_{d_2})$, again being repeated for all B_1 and B_2 nodes in $f(v_{d_2}|v_{d_1})$. This implies that an inversion performed on an optimal moment set, $\boldsymbol{M}$, will produce different nodes depending on the conditional sequence, $\boldsymbol{b}$. The first depth dimension (d_1) will only have up to B_1 nodes, while the 2nd depth dimension (d_2) may have up to B_1B_2, and the 3rd depth (d_3) up to $B_1B_2B_3$.

Consider also that the number of nodes in each dimension d can range between $1 \leq B_d \leq \beta$, implying that B_d is adaptive and may grow/shrink as necessary to accommodate the moment set in each dimension individually. Thus, $B_d = 1$ indicates that the PDF has zero variance ($\Theta_d = 0$) in dimension d, thus only a single instance of abscissa and weight are required for complete description of the particle phase velocity in that dimension. This adaptive approach will improve the efficiency of the solver by only considering additional nodes $B_d > 1$ when necessary, but being restricted up to a total of $B_d = \beta$ when completely exhausting the node count in that dimension. The total number of 1-D inversions is flexible between the range $D, ..., \mathbb{B}$, at a minimum when all D instances of $B_d = 1$ and at a maximum when all $B_d = \beta$ respectively. Thus, a full multivariate inversion may produce up to a maximum of $\mathbb{B} = \beta^{\mathcal{X}}$ nodes. Herein lies some of the variability in the computational requirements. Non-diverse PDF's with zero variance require only a single node in each dimension, d, (a Dirac Delta function) whereas regions with large variance in multiple dimensions may exhaust all $\mathbb{B}$ nodes. Figure 12(a) demonstrates an example of a fully exhausted 2-point quadrature in 3-D, totaling $\mathbb{B} = 8$ nodes computed following conditional sequence $\boldsymbol{b} = [1, 2, 3] \equiv [x, y, z]$. For both nodes computed in the x-dimension, $\{(U, \phi)_1, (U, \phi)_2\}$, two more nodes result in the y-dimension for both, $\{(V, \phi)_{11}, (V, \phi)_{12}, (V, \phi)_{21}, (V, \phi)_{22}\}$, then two more nodes in the z-dimension for *each* of those $\{(W, \phi)_{111}, (W, \phi)_{112}, (W, \phi)_{121}, (W, \phi)_{122}, (W, \phi)_{211}, (W, \phi)_{212}, (W, \phi)_{221}, (W, \phi)_{222}\}$.

Alternatively, Figure 12(b) demonstrates a *partially* exhausted quadrature representation which was computed following conditional sequence $\boldsymbol{b} = [3, 1, 2] = [z, x, y]$. Note that the first z-inversion resulted in two nodes, $\{W_1, \phi_1, W_2, \phi_2\}$, thus two more x-inversions were performed in the 2nd dimension, one for each conditional z-node. The first x-inversion produced two more nodes, $\{U_{11}, \phi_{11}, U_{12}, \phi_{12}\}$, but the second x-inversion only produced one node, $\{U_{21}, \phi_{21}\}$. Lastly, the y-inversion is performed on all three previously computed conditional nodes to solve $f(v_2|v_3, v_1)$. The first



(a) $\boldsymbol{b} = [1, 2, 3]$, fully exhausted with $B_d = \beta = 2$

(b) $\boldsymbol{b} = [3, 1, 2]$, partially exhausted with various $1 \leq B_d \leq 2$

Figure 12. Example Visualization of a 3-D Velocity Space for Discretized PDFs. Nested Moment Inversion Produces Nodes for Two Conditional Sequences, *b*

inversion produced two nodes, $\{V_{111}, \phi_{111}, V_{112}, \phi_{112}\}$, the second produced one node, $\{V_{121}, \phi_{121}\}$, and the third produced two nodes $\{V_{211}, \phi_{211}, V_{212}, \phi_{212}\}$. Altogether, five nodes have been allocated by this adaptive process, which is less than reserved maximum $\mathbb{B} = 8$.

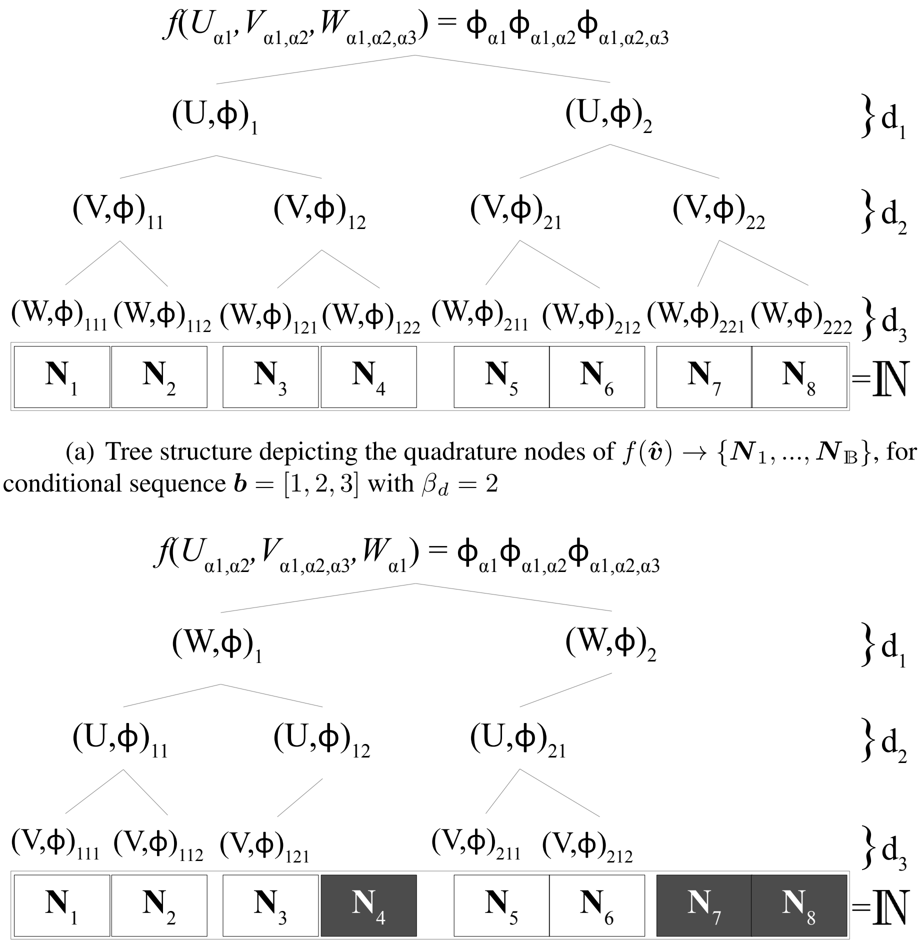
The discretized 3-D PDF depicted in Figure 12(a), specifically for conditional sequence $\boldsymbol{b} = [1, 2, 3]$, is represented mathematically by a summation of Dirac delta functions[22] that covers all conditional nodes,[44]

$$\overset{[1]}{f}(\boldsymbol{v}) \approx \sum_{\alpha_1}^{\beta_1} \sum_{\alpha_2}^{\beta_2} \sum_{\alpha_3}^{\beta_3} \phi_{\alpha_1} \phi_{\alpha_1,\alpha_2} \phi_{\alpha_1,\alpha_2,\alpha_3} \delta(v_1 - U_{\alpha_1}) \delta(v_2 - V_{\alpha_1,\alpha_2}) \delta(v_3 - W_{\alpha_1,\alpha_2,\alpha_3}) \quad (87)$$

where d_1=1 corresponds to the *first* dimension that the inversion was performed in, i.e. represented by α_1 and β_1. It is further demonstrated by Equation (87) that *not* all $\mathbb{B}$ nodes are required (unless $B_d = \beta$ for all d) and any unused nodes can simply retain a weight of zero with any arbitrary abscissa. At most, $\mathbb{B}$ non-zero values are obtainable from $f(\boldsymbol{v})$ and only specifically when $\boldsymbol{v} = \hat{\boldsymbol{v}}$, where $\hat{\boldsymbol{v}} = \{U_{\alpha_1}, V_{\alpha_1,\alpha_2}, W_{\alpha_1,\alpha_2,\alpha_3}\}$ is a vector of abscissas such that $f(\hat{\boldsymbol{v}}) = \phi_{\alpha_1} \phi_{\alpha_1,\alpha_2} \phi_{\alpha_1,\alpha_2,\alpha_3}$ returns the non-zero weight. Therefore, these 3-D abscissas and corresponding weights can be mapped, $f(\hat{\boldsymbol{v}}) \rightarrow \mathbb{N}$, similar to Equation (50) by a discretized node matrix shown in Equation (88), where $\boldsymbol{N}_\alpha$ is a single row of $\mathbb{N}$ from Equation (54), and the *linearized* node index, $\alpha = \beta^2(\alpha_1 - 1) + \beta(\alpha_2 - 1) + (\alpha_3 - 1)$, ensures that each node receives a unique index ranging from $1, ..., \mathbb{B}$ and stems from a tree structure framework.

$$\boldsymbol{N}_\alpha = \begin{bmatrix} U_\alpha \\ \\ V_\alpha \\ \\ W_\alpha \\ \\ \phi_\alpha \end{bmatrix}^T = \begin{bmatrix} U_{\alpha_1} \\ \\ V_{\alpha_1,\alpha_2} \\ \\ W_{\alpha_1,\alpha_2,\alpha_3} \\ \\ \phi_{\alpha_1}\phi_{\alpha_1,\alpha_2}\phi_{\alpha_1,\alpha_2,\alpha_3} \end{bmatrix}^T \text{for } \alpha = 1, ..., \mathbb{B} \tag{88}$$

In fact, the conditional-splitting method is best viewed as tree structure which is depicted in Figure 13(a), where each row corresponds to the dimensions d_1, d_2, and d_3. Each complete branch extending from the root to the bottom represents a complete 3-D node, and is a visual representation of Equation (88). For any other conditional sequence $[d_1, d_2, d_3]$, the U, V, and W will swap rows accordingly to match. Furthermore, for cases with a larger number of nodes ($\beta > 2$), additional branches would be necessary at each juncture. The benefit of this tree architecture is that the available branches are only used when necessary, i.e. adaptive quadrature. If one node suffices ($B_d = 1$) to explain the particle phase within a dimension d, then the computations/inversions of unused branches are ignored entirely. This same logic applies to all levels of the architecture tree, thus it is possible to have as few as one complete node (which still needs a minimum of $D = 3$, 1-D inversions), and up to $\mathbb{B} = 8$ conditional nodes when all available nodes are exhausted (up to $B_1 = B_2 = B_3 = \beta = 2$). Another tree structure is shown in Figure 13(b) corresponding to the visual example in Figure 12(b), which has fewer nodes ($B_d = 1$) for some inversions. Therefore, some of the nodes are unused and given zero weight for those rows in $\mathbb{N}$.



(a) Tree structure depicting the quadrature nodes of $f(\hat{\boldsymbol{v}}) \rightarrow \{\boldsymbol{N}_1, ..., \boldsymbol{N}_{\mathbb{B}}\}$, for conditional sequence $\boldsymbol{b} = [1, 2, 3]$ with $\beta_d = 2$

(b) Tree structure depicting the quadrature nodes of $f(\hat{\boldsymbol{v}}) \rightarrow \{\boldsymbol{N}_1, ..., \boldsymbol{N}_{\mathbb{B}}\}$, for conditional sequence $\boldsymbol{b} = [3, 1, 2]$

Figure 13. Example 3-D PDF Discretization With Conditional Nodes for Two Different Conditional-Splitting Inversion Sequences, *b*

Lastly, one more topic must be addressed regarding the abscissa proximity ratios, $\mathbb{P}_d$, which are relevant when a $\mathbb{B}_d = 2$-point inversion is performed in dimension d. The values obtained from earlier inversions (those with smaller depths in the nesting) are shared among all the successively nested (higher depth) inversions. If the lower depth inversions are ill-conditioned then negative effects will cascade to the larger depth inversions and increase the likelihood of ill-conditioned inversions. As such, the $\mathbb{P}_d$

values should be most stringent at the lower depths and relaxed at higher depths; in fact, the final depth need not be restricted at all ($\mathbb{P}_{\mathcal{X}}$=1). For example, the 3-D turbulent channel flow was able to avoid ill-conditioned situations by using $\{\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3\} \approx \{1.1, 1.05, 1.0\}$, but with a CFL check could go even lower than $\{1.01, 1.01, 1.0\}$.

To summarize, CQMOM offers a multivariate inversion that maps moments into nodes, $\boldsymbol{M} \xrightarrow{[d_1,d_2,d_3]} \overset{[d_1]}{\mathbb{N}}$. Figures 12 and 13 show visual and tree structure representations of a CQMOM 3-D inversion for two different conditional sequences, $[1, 2, 3]$ and $[3, 1, 2]$), where Equation (87) represents the 3-D nodes mathematically and Equation (88) describes the various discretized representations of the important information resulting from the multidimensional inversion process. The 3-D node set, $\overset{[d_1]}{\mathbb{N}}$ can now be used to perform moment convection in the d_1 direction of the transport equations, as is described in the next section.

B. Spatial flux scheme

This section is devoted to describing the multidimensional flux scheme which is a sophisticated portion of the CQMOM algorithm and is coupled with the results of the inversion process. Node matrix $\mathbb{N}(t, \boldsymbol{x})$ is used directly in the flux term to reconstruct the moments for their transport between cells.

1. Flux Splitting

The quadrature representation of f obtained through conditional moment inversion in the previous section must provide closure to the convective term of Equation (48), appearing as $\nabla \boldsymbol{x} \cdot M^{(\gamma+1)}$, and is computed by constructing the higher order moment,

$M^{(\gamma+1)}$ with Equation (52) while *simultaneously* applying the flux term in dimension d_1, written as, $\overset{[d_1]}{\mathbb{N}} \xrightarrow{x_{d_1}} \boldsymbol{M}$. $M^{(\gamma+1)}$ is effectively constructed by increasing the corresponding abscissa multiplicity by one (l_{d_1}+= 1).[42] To calculate the *entire* multidimensional convection however, a flux-splitting method is chosen because it provides 1-D convection at fractional steps that cover all D dimensions. Additionally, a *full* multidimensional inversion provides $\mathbb{N}$ to close the moment transport equation,[44] before convecting the moments in a single dimension, d_1. Fortunately, $D = 3$ is the maximum dimension for a physical velocity-space that applies to the flux-splitting scheme, i.e. $Q > 0$ plays no role in the flux-splitting.

Recall that a full multidimensional inversion, $\boldsymbol{M} \xrightarrow{[d_1,d_2,\ldots,d_{\mathcal{X}}]} \overset{[d_1]}{\mathbb{N}}$, produces a $\mathcal{X}$-dimensional node set, $\overset{[d_1]}{\mathbb{N}}$, which is conditioned *first* by dimension d_1 (indicated by the superscript). The first condition dimension is particularly important because it is the least susceptible to numerical error and corruption that occurs from successive *nested* inversions that rely on higher order mixed moments. Therefore, it is sometimes beneficial to use the first dimension's velocity for the convection term but this convention can be limiting and is not mandatory. As is described further in section §*3. Convection Term*, $M^{(\gamma+1)}$ gains a multiplicity of one for the velocity in dimension d_1. Consequently, only a maximum of β unique velocities ($\hat{v}_{d_1,\alpha_1}$) are present for convecting the moment information, regardless of the total number of nodes ($\mathbb{B}$) occupied in the multidimensional PDF. Therefore, it may be useful to perform the inversion in the convection dimension *last* in the sequence to obtain up to $\mathbb{B}$ unique velocities with which to convect the moments during construction of $M^{(\gamma+1)}$.

The time iteration scheme uses the closed flux term construction to update the new moment set, $\overset{[d_1]}{\boldsymbol{M}}$, which has just undergone flux in the single dimension, d_1, matching the superscript. Therefore, a complete notation for an "inversion-flux pair" operation is of the form,

$$\boldsymbol{M} \xrightarrow{[d_1, d_2, d_3, \ldots]} \overset{[d_1]}{\mathbb{N}} \xrightarrow{x_{d_1}} \overset{[d_1]}{\boldsymbol{M}} \tag{89}$$

This full inversion and single dimension flux reconstruction must be repeated consecutively for all D dimensions to complete the multidimensional flux-splitting that fully evolves the moment set from time step n to $n+1$,

$$\overset{(n)}{\boldsymbol{M}} \xrightarrow{[1,\ldots]} \overset{[1]}{\mathbb{N}} \xrightarrow{x_1} \overset{[1]}{\boldsymbol{M}} \xrightarrow{[2,\ldots]} \overset{[2]}{\mathbb{N}} \xrightarrow{x_2} \overset{[2]}{\boldsymbol{M}} \xrightarrow{[3,\ldots]} \overset{[3]}{\mathbb{N}} \xrightarrow{x_3} \overset{[3]}{\boldsymbol{M}} \rightarrow \overset{(n+1)}{\boldsymbol{M}} \tag{90}$$

After the x_1-inversion is completed, the moments are convected only in the x_1 dimension of the flux-splitting scheme. The subsequent dimensions (x_2 and x_3) are convected in successive inversions. Note that the first dimension of the conditional inversion sequence, $\boldsymbol{b} = [d_1, \ldots]$, must match the flux dimension, x_{d_1}, that immediately follows. Together, the conditional-splitting provides a multivariate inversion and the flux-splitting provides multidimensional convection, creating a complete *path*, from $\overset{(n)}{\boldsymbol{M}} \rightarrow \overset{(n+1)}{\boldsymbol{M}}$ as shown in Equation (90). It should also be noted that the *split* nodes $\overset{[d]}{\mathbb{N}}$ and moment vectors $\overset{[d]}{\boldsymbol{M}}$ are merely intermediate temporary placeholders, denoted by a tilde in Equation (91),

$$\overset{(n)}{\boldsymbol{M}} \xrightarrow{[1,\ldots]} \widetilde{\mathbb{N}} \xrightarrow{x_1} \widetilde{\boldsymbol{M}} \xrightarrow{[2,\ldots]} \widetilde{\mathbb{N}} \xrightarrow{x_2} \widetilde{\boldsymbol{M}} \xrightarrow{[3,\ldots]} \widetilde{\mathbb{N}} \xrightarrow{x_3} \widetilde{\boldsymbol{M}} \rightarrow \overset{(n+1)}{\boldsymbol{M}} \tag{91}$$

Consequently, the final resulting moment set $\overset{(n+1)}{\boldsymbol{M}}$ is a single completed time-step, containing values from a multidimensional flux-splitting path.

It should be apparent that many different sequence choices are valid for the complete path. Different sequences will produce different results, i.e. any single path is dependent upon the conditional sequences $\boldsymbol{b} = [d_1, d_2, d_3, ...]$, so a standard unifying method must be employed. To ensure equality *and* a unique solution, *all* possible paths (p in P) must be computed then *averaged* together. In 3-D, there are $D! = 6$ conditional permutations of $\boldsymbol{b}$ from Equation (86), and there are $P = (D-1)!^D = 8$ unique paths for a flux-splitting sequence $[x_1, x_2, x_3] \equiv [x, y, z]$.

Table 7. Conditional sequences $\boldsymbol{b} = [d_1, d_2, d_3]$ for Flux-Splitting Path $[x_1, x_2, x_3]$

$d\downarrow$	p=1	2	3	4	5	6	7	P=(D-1)!^{D}
x_1	$[1,2,3]$	$[1,2,3]$	$[1,2,3]$	$[1,2,3]$	$[1,3,2]$	$[1,3,2]$	$[1,3,2]$	$[1,3,2]$
x_2	$[2,1,3]$	$[2,1,3]$	$[2,3,1]$	$[2,3,1]$	$[2,1,3]$	$[2,1,3]$	$[2,3,1]$	$[2,3,1]$
x_3	$[3,1,2]$	$[3,2,1]$	$[3,1,2]$	$[3,2,1]$	$[3,1,2]$	$[3,2,1]$	$[3,1,2]$	$[3,2,1]$

Each column of Table 7 represents a unique conditional *path*, p. For simplicity, the sequence of the dimensions $[x_1, x_2, x_3] \equiv [x, y, z]$ (rows of Table 7) will remain fixed; consequently, this may give preferential treatment to certain dimensions of the flux computations that may produce asymmetries in symmetrical flows, which is a known side-effect of numerical flux-splitting methods.[64] A proposed fix is described later in Section §2. *Dimension Stepping*. Note that the flux-splitting process is independent of β and Q, and depends on the PDF's total *spatial* dimension, D. Equation (92) shows the list of paths from Table (7) that will complete just one time step evolution of the moments. All $P = 8$ paths must be computed and the 8 resulting sets, $\overset{[3]}{\boldsymbol{M}}$, must be averaged together to obtain $\overset{(n+1)}{\boldsymbol{M}}$.

$$\overset{(n)}{\boldsymbol{M}} \left\{ \begin{matrix}
\xrightarrow{[1,2,3]} \overset{[1]}{\mathbb{N}} \xrightarrow{x_1} \overset{[1]}{\boldsymbol{M}} \xrightarrow{[2,1,3]} \overset{[2]}{\mathbb{N}} \xrightarrow{x_2} \overset{[2]}{\boldsymbol{M}} \xrightarrow{[3,1,2]} \overset{[3]}{\mathbb{N}} \xrightarrow{x_3} \overset{[3]}{\boldsymbol{M}}_{p=1} \\
\xrightarrow{[1,2,3]} \overset{[1]}{\mathbb{N}} \xrightarrow{x_1} \overset{[1]}{\boldsymbol{M}} \xrightarrow{[2,1,3]} \overset{[2]}{\mathbb{N}} \xrightarrow{x_2} \overset{[2]}{\boldsymbol{M}} \xrightarrow{[3,2,1]} \overset{[3]}{\mathbb{N}} \xrightarrow{x_3} \overset{[3]}{\boldsymbol{M}}_{p=2} \\
\xrightarrow{[1,2,3]} \overset{[1]}{\mathbb{N}} \xrightarrow{x_1} \overset{[1]}{\boldsymbol{M}} \xrightarrow{[2,3,1]} \overset{[2]}{\mathbb{N}} \xrightarrow{x_2} \overset{[2]}{\boldsymbol{M}} \xrightarrow{[3,1,2]} \overset{[3]}{\mathbb{N}} \xrightarrow{x_3} \overset{[3]}{\boldsymbol{M}}_{p=3} \\
\xrightarrow{[1,2,3]} \overset{[1]}{\mathbb{N}} \xrightarrow{x_1} \overset{[1]}{\boldsymbol{M}} \xrightarrow{[2,3,1]} \overset{[2]}{\mathbb{N}} \xrightarrow{x_2} \overset{[2]}{\boldsymbol{M}} \xrightarrow{[3,2,1]} \overset{[3]}{\mathbb{N}} \xrightarrow{x_3} \overset{[3]}{\boldsymbol{M}}_{p=4} \\
\xrightarrow{[1,3,2]} \overset{[1]}{\mathbb{N}} \xrightarrow{x_1} \overset{[1]}{\boldsymbol{M}} \xrightarrow{[2,1,3]} \overset{[2]}{\mathbb{N}} \xrightarrow{x_2} \overset{[2]}{\boldsymbol{M}} \xrightarrow{[3,1,2]} \overset{[3]}{\mathbb{N}} \xrightarrow{x_3} \overset{[3]}{\boldsymbol{M}}_{p=5} \\
\vdots \\
\xrightarrow{[1,*,*]} \overset{[1]}{\mathbb{N}} \xrightarrow{x_1} \overset{[1]}{\boldsymbol{M}} \xrightarrow{[2,*,*]} \overset{[2]}{\mathbb{N}} \xrightarrow{x_2} \overset{[2]}{\boldsymbol{M}} \xrightarrow{[3,*,*]} \overset{[3]}{\mathbb{N}} \xrightarrow{x_3} \overset{[3]}{\boldsymbol{M}}_{p} \\
\vdots \\
\xrightarrow{[1,3,2]} \overset{[1]}{\mathbb{N}} \xrightarrow{x_1} \overset{[1]}{\boldsymbol{M}} \xrightarrow{[2,3,1]} \overset{[2]}{\mathbb{N}} \xrightarrow{x_2} \overset{[2]}{\boldsymbol{M}} \xrightarrow{[3,2,1]} \overset{[3]}{\mathbb{N}} \xrightarrow{x_3} \overset{[3]}{\boldsymbol{M}}_{p=P}
\end{matrix} \right\} \to \overset{(n+1)}{\boldsymbol{M}} \tag{92}$$

Equation (92) may first appear to suggest that a total of $D(D-1)!^D$ inversion-flux pairs are required, but notice that some of the inversions are identical. For example, in dimension x_1, the first four paths ($p = 1, ..., 4$) all begin with the same $\boldsymbol{b} = [1, 2, 3]$ inversion sequence first, while the last four paths ($p = 5, ..., 8$) use inversion sequence $\boldsymbol{b} = [1, 3, 2]$. In fact, every x_d row in Table 7 only has $(D - 1)!$ unique inversion sequences: $[d, *, *]$ where all permutations will generate all possible paths. Therefore, it is useful to conceptualize the paths as a tree structure rather than a table. Furthermore, it is beneficial to store the results of operation: $\boldsymbol{M} \xrightarrow{[1,2,3]} \overset{[1]}{\mathbb{N}} \xrightarrow{x_1} \overset{[1]}{\boldsymbol{M}}$, and reuse them for all paths that share that same first inversion sequence, $[1, 2, 3]$. This optimization reduces the total number of inversion-flux pairs from $D(D-1)!^D$ to only $\sum_{d=1}^{D} (D-1)!^d$.

2. *Dimension Stepping*

Per Table 7 and Equation (92), the dimension sequence of the flux-splitting is always $[x_1, x_2, x_3]$. This order gives preferential treatment to the dimensions and if kept the same for *every* time step may introduce asymmetries into the solution. In general, the sequence should be able to change as $[x_{d_1}, x_{d_2}, x_{d_3}]$, where d_1, d_2, and d_3 are any permutation of the sequence $[1, 2, 3]$. One method to achieve perfect symmetry is to compute every possible sequence and average them together; this however would be extremely costly and so a common alternative is proposed next.

To improve the physical solution symmetry, the sequence must be incremented or "stepped" after each time step iteration so that no sequence is favored over another. Start by selecting an initial sequence $[x_{d_1}, x_{d_2}, x_{d_3}] \equiv [x_1, x_2, x_3] \equiv [x, y, z]$ for the first iteration, then increment each dimension by one and modulate any value which exceeds D by subtracting D from it, i.e. for the second iteration, $[x_2, x_3, x_1] \equiv [y, z, x]$, and so forth for any additional iterations. Additionally, if a multi-step scheme such as Runge-Kutta is employed, the stepping can occur within the R iterations. For example, a 3rd-order Runge-Kutta ($R = 3$) method in 3-D space could step as proceeds, where n is the time step index starting at 0, and $r = 0...(R - 1)$ is the iteration index for an R'th-order Runge-Kutta scheme (or any similar scheme which has multiple iterations per time step). Table 8 can be modified for any order time iteration scheme. This dimension stepping of the flux-splitting sequence can significantly improve symmetry in flow

Table 8. Conditional Dimension Stepping Sequence $R = 3$, $D = 3$

n	r	Sequence
0	0	$[x, y, z]$
	1	$[y, z, x]$
	2	$[z, x, y]$
1	0	$[y, z, x]$
	1	$[z, x, y]$
	2	$[x, y, z]$
2	0	$[z, x, y]$
	1	$[x, y, z]$
	2	$[y, z, x]$
	$\vdots$	

solutions, but does not remedy the asymmetries entirely, as can be confirmed with a 2-D Taylor-Green flow simulation.

3. Convection Term

This method was described by Desjardins *et al.*[42] in 1-D, and the 3-D extension is described below. Let h be the test function which provides the product of exponentiated velocities such that,

$$h(m, \boldsymbol{v}) = \prod_{d=1}^{D} v_d^{\mathbb{E}(m,d)} = v_1^{l_1} v_2^{l_2} v_3^{l_3} \tag{93}$$

where the function $\mathbb{E}(m, d)$ returns the single exponent for the corresponding abscissa, based on the m'th moment index and d'th dimension of $\boldsymbol{v}$, as described by Table 9. Let $\boldsymbol{h}$ be the vector containing all $m = 0, ..., K - 1$ of these combinations of exponents on the velocity abscissa vector, $\boldsymbol{v}$,

$$\boldsymbol{h}(\hat{\boldsymbol{v}}) = \{h(0, \hat{\boldsymbol{v}}), h(1, \hat{\boldsymbol{v}}), ..., h(K-1, \hat{\boldsymbol{v}})\} = \{\hat{v}_1^0 \hat{v}_2^0 \hat{v}_3^0, \hat{v}_1^1 \hat{v}_2^0 \hat{v}_3^0, ...\} \tag{94}$$

Then it follows that the summation of these velocity products multiplied by the weight, ϕ_α, for all nodes, constructs the entire moment set, $\mathbb{N} \rightarrow \boldsymbol{M}$, just as Equation (52) does (quadrature approximation),

$$\boldsymbol{M} = \int_{\Omega_{\boldsymbol{v}}} \boldsymbol{h}(\boldsymbol{v}) f(\boldsymbol{v}) d\boldsymbol{v} \approx \sum_{\alpha=1}^{\mathbb{B}} \phi_\alpha \boldsymbol{h}(\hat{\boldsymbol{v}}_\alpha) \tag{95}$$

where $\boldsymbol{M}(t, \boldsymbol{x})$ contains the K optimal moments from Table 9. At any single face, there exists the positive velocity (right-bound), $\boldsymbol{H}^+$, and negative velocity (left-bound), $\boldsymbol{H}^-$, convective moments,[36,42]

$$\boldsymbol{H}^+(d, \mathbb{N}) = \sum_{\alpha=1}^{\mathbb{B}} \max(\hat{v}_{d,\alpha,\ell}, 0)\ \phi_{\alpha,\ell}\ \boldsymbol{h}(\hat{\boldsymbol{v}}_{\alpha,\ell}) \tag{96}$$

$$\boldsymbol{H}^-(d, \mathbb{N}) = \sum_{\alpha=1}^{\mathbb{B}} \min(\hat{v}_{d,\alpha,r}, 0)\ \phi_{\alpha,r}\ \boldsymbol{h}(\hat{\boldsymbol{v}}_{\alpha,r}) \tag{97}$$

where the subscripts ℓ and r denote the left or right side of the face respectively. The summations include all α nodes from 1...$\mathbb{B}$, and in fact, Equations (96) and (97) are moment construction formulas similar to Equation (52), but with one extra multiple of $\hat{v}_{d,\alpha}$, where d is the dimension that the convection is being considered in. In fact, $\boldsymbol{H}$ can be thought of as a vector of moments, $M^{(\gamma+1)}$, one element for each optimal transport equation in Equation (49). Note that the positive velocity flux is determined by the reconstructed node at the left face, ℓ, and the negative velocity flux is determined by the reconstructed node at the right face; in correspondence with the upwind scheme's shock treatment[64] and appropriate slope limiter.[45] At no point can $\boldsymbol{H}_m^+$ and $\boldsymbol{H}_m^-$ both be non-zero, only one of the fluxes can exist for each index m. The net flux on a single face, $\boldsymbol{G}$, partly describes the transformation, $\overset{[d]}{\mathbb{N}} \xrightarrow{x_d} \boldsymbol{M}$, and is calculated as the sum of Equations (96) and (97),

$$\boldsymbol{G}(d, \mathbb{N}_\ell, \mathbb{N}_r) = \boldsymbol{H}^+(d, \mathbb{N}_\ell) + \boldsymbol{H}^-(d, \mathbb{N}_r) \tag{98}$$

This convection function, $\boldsymbol{G}$, represents a vector of the computed convective terms, $\frac{\partial}{\partial \boldsymbol{x}} M^{(\gamma+1)}$, that appears in all transport Equations (49), and must be computed at each face. Furthermore, each face is shared by two neighboring cells so those two cells also share an instance of $\boldsymbol{G}$ because any flux leaving one cell must enter the neighboring cell, and vice versa. This “equal and opposite” consideration is useful for code optimization to prevent redundant calculations, where only the sign (+/-) of $\boldsymbol{G}$ needs to be opposite between the two neighboring cells. For a Cartesian mesh (and mandatory for a flux-splitting method) it is advantageous to perform the flux computations in one d dimension for *all* the cells before switching to the next dimension. This optimization works well when cycling through the entire domain consecutively as it only requires that two $\boldsymbol{G}$ vectors be stored in memory simultaneously.

Rewritten for completeness, the numerical convective scheme becomes,

$$\boldsymbol{G}(d, \mathbb{N}_\ell, \mathbb{N}_r) = \int \hat{v}^+_{d,\ell} \boldsymbol{h}(\hat{\boldsymbol{v}}_\ell) f(\hat{\boldsymbol{v}}_\ell) d\hat{\boldsymbol{v}} + \int \hat{v}^-_{d,r} \boldsymbol{h}(\hat{\boldsymbol{v}}_r) f(\hat{\boldsymbol{v}}_r) d\hat{\boldsymbol{v}}$$
$$\approx \sum_{\alpha=1}^{\mathbb{B}} \hat{v}^+_{d,\alpha,\ell} \boldsymbol{h}(\hat{\boldsymbol{v}}_{\alpha,\ell}) \phi_{\alpha,\ell} + \sum_{\alpha=1}^{\mathbb{B}} \hat{v}^-_{d,\alpha,r} \boldsymbol{h}(\hat{\boldsymbol{v}}_{\alpha,r}) \phi_{\alpha,r} \quad (99)$$

where the convection velocities v^+ and v^- will increase the multiplicity of the constructed moment by one. Depending which direction the velocity is going determines which neighboring node is to be used; therefore, *at most* only one of the two is nonzero, i.e. $v^+ v^- = 0$.

$$v^+ = \max(v, 0) \qquad v^- = \min(v, 0) \quad (100)$$

The min and max functions ensure that the right-bound (positive velocity) and left-bound (negative velocity) fluxes are being applied properly, i.e. that $v^+ \geq 0$ and that $v^- \leq 0$. This is better handled with an "if-statement" condition in a source code context; however, it is still advantageous to sum the negative and positive components separately as shown (compute $\boldsymbol{H}^-$ and $\boldsymbol{H}^+$ separately before adding them together) otherwise numerical round-off error is far more likely to create asymmetries in a symmetrical flow if both positive and negative quantities are added immediately into one sum.

4. Slope limiter

An upwind slope limiter is introduced[45,51] during face reconstruction (only applicable at 2nd-order accurate and higher spatial flux schemes) to prevent the moments from becoming negative and to ensure that they remain Total Variation Diminishing

(TVD) which prevents spurious oscillations and preserves monotonicity.[64]

$$\texttt{minmod}(X, Y) = \text{sign}(X) \left(\frac{1 + \text{sign}(XY)}{2} \right) \min(|X|, |Y|) \tag{101}$$

The `minmod` limiter attempts to use the slope ($|X|$ or $|Y|$) which is smaller in steepness, or uses zero slope (reducing to a first-order assumption) if they have opposite signs. In the flux context, the two slopes are neighboring both sides of a cell center. To actually determine the proper slope about the cell-center ($\mathbb{N}_a$), it is implemented as,

$$\frac{\partial}{\partial x}\mathbb{N}_a = \texttt{minmod}\left(\frac{\mathbb{N}_a - \mathbb{N}_{a-1}}{x_a - x_{a-1}}, \frac{\mathbb{N}_{a+1} - \mathbb{N}_a}{x_{a+1} - x_a} \right) \tag{102}$$

where a is the cell-center index along the desired spatial dimension, d, and $x_a - x_{a-1}$ is the distance between two neighboring cell-centers, not to be confused with $\Delta x_a = x_{a+\frac{1}{2}} - x_{a-\frac{1}{2}}$ which is the cell width or distance between both faces. Distinguishing the cell dimensions in this manner allows for non-uniform (rectilinear) grid distributions to be modeled. The slope $\frac{\partial}{\partial x}\mathbb{N}_a$ exists about the cell center of each a'th cell.

Special consideration is required whenever unused nodes exist in $\mathbb{N}$ (such that fewer than $\mathbb{B}$ nodes are occupied) because unoccupied nodes (those with zero weight, $\phi_\alpha = 0$) should be ignored entirely. When a higher order flux scheme attempts to operate between two neighboring cells having differing inversion sequences, the unoccupied nodes in $\mathbb{N}$ must *not* be used to compute adjoining slopes related to the flux at cell faces. To account for this, a matrix $\mathbb{I}(t, \boldsymbol{x})$ may store an ID (such as an integer value) describing which nodes are occupied in $\mathbb{N}(t, \boldsymbol{x})$, and if two neighboring cells differ, then a simple zero-order flux scheme should be applied, $\frac{\partial}{\partial x}\mathbb{N}_a = 0$. For more information, see the numerical implementation of variable “`ndx`” in Yuan *et al.*[44]

Figure 14 depicts an example of the `minmod` slope limiter and the reconstruction at the left (ℓ) and right (r) face sides. The cell centers lie at the locations $(x_a, \mathbb{N}_a)$, which is halfway ($\frac{\Delta x_a}{2}$) between the two cell faces, shown as vertical dotted lines. As defined, `minmod` always chooses the slope, $\frac{\partial}{\partial x}\mathbb{N}_a$, that is *lesser in steepness* or chooses a zero slope at local maximums or minimums. Once the slopes, $\frac{\partial}{\partial x_d}\mathbb{N}_a$ are computed in

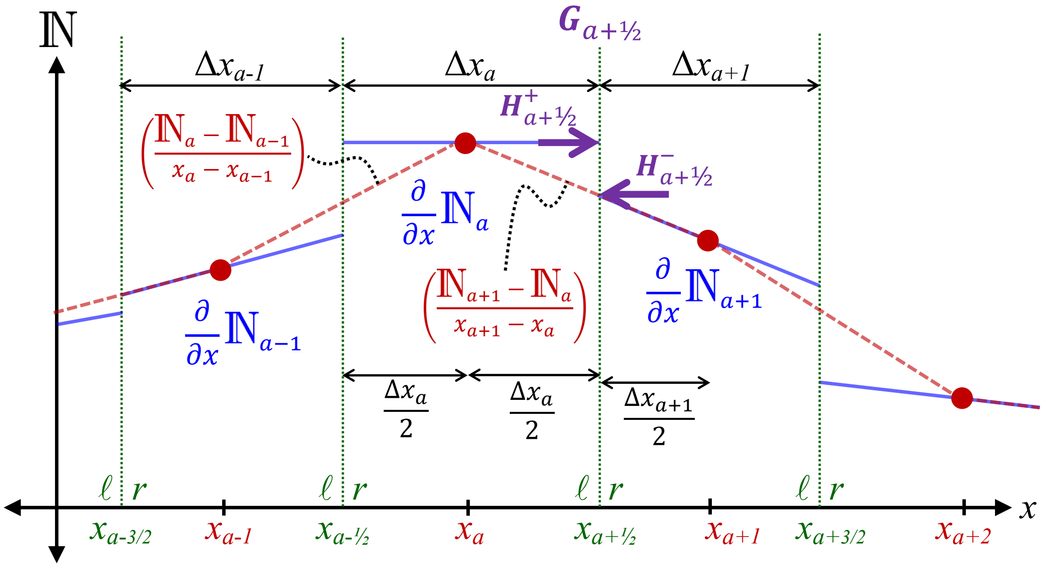

$\mathbb{N}$
$G_{a+½}$
Δx_{a-1}
Δx_a
Δx_{a+1}
$H^+_{a+½}$
$\left(\frac{\mathbb{N}_a - \mathbb{N}_{a-1}}{x_a - x_{a-1}}\right)$
$H^-_{a+½}$
$\frac{\partial}{\partial x}\mathbb{N}_a$
$\left(\frac{\mathbb{N}_{a+1} - \mathbb{N}_a}{x_{a+1} - x_a}\right)$
$\frac{\partial}{\partial x}\mathbb{N}_{a+1}$
$\frac{\partial}{\partial x}\mathbb{N}_{a-1}$
$\frac{\Delta x_a}{2}$
$\frac{\Delta x_a}{2}$
$\frac{\Delta x_{a+1}}{2}$
ℓ r
ℓ r
ℓ r
ℓ r
x
$x_{a-3/2}$
x_{a-1}
$x_{a-½}$
x_a
$x_{a+½}$
x_{a+1}
$x_{a+3/2}$
x_{a+2}

Figure 14. Visual Representation of `minmod` Slope Limiter

dimension d about the a’th cell centers (for *all* abscissas and weights in $\mathbb{N}$ individually), they are used to construct the values at the left and right faces per Equation (103),

$$\begin{aligned}\mathbb{N}_{a-\frac{1}{2},r} &= \mathbb{N}_a - \tfrac{\Delta x_a}{2}\tfrac{\partial}{\partial x_d}\mathbb{N}_a \\ \mathbb{N}_{a+\frac{1}{2},\ell} &= \mathbb{N}_a + \tfrac{\Delta x_a}{2}\tfrac{\partial}{\partial x_d}\mathbb{N}_a\end{aligned} \tag{103}$$

Note that each cell can determine the reconstruction at the right side of the left face, $x_{a-\frac{1}{2},r}$, and the left side of the right face, $x_{a+\frac{1}{2},\ell}$, using the cell half-width $\frac{\Delta x_a}{2}$ which is the distance from the cell center to either face. These reconstructions at the faces are used in function $\boldsymbol{H}(d,\mathbb{N})$ of Equations (96) and (97). Whether the positive or negative $\boldsymbol{H}$ is chosen depends on the velocity at that *single* face (which requires the reconstruction from two different cell centers), then the net flux across that *single* face appears in the convective term, $\boldsymbol{G}(d,\mathbb{N}_{a+\frac{1}{2},\ell},\mathbb{N}_{a+\frac{1}{2},r})$ of Equation (98).

Another prominent scheme is the superbee slope limiter[64] which is capable of significantly reducing the numerical diffusion at sharp discontinuities where minmod may severely limit the allowable shock slopes.

$$\texttt{superbee}(X,Y) = \texttt{maxmod}\Bigg(\texttt{minmod}(X,\varepsilon Y),\texttt{minmod}(\varepsilon X,Y)\Bigg) \tag{104}$$

The limiter remains TVD in general,[64] but the coefficient, $1 \leq \varepsilon \leq 2$, may be chosen less than 2 to relax the method, where $\varepsilon = 1$ recovers the minmod limiter. In the current work, $\varepsilon = 1.5$ was chosen.

5. *Non-Uniform Grid*

The previous formulations are compatible with a non-uniform grid. In the present work, many test case simulations involving a rectilinear grid were skewed using a hyperbolic tangent curve applied in the y-direction only, and achieved using,

$$y_{a-\frac{1}{2}} = L_y \left(1 - \frac{\tanh\left(\varepsilon_g \left[1 - \frac{a}{\mathcal{N}}\right]\right)}{\tanh(\varepsilon_g)} \right) \tag{105}$$

where $\mathcal{N}$ is the number of cells that span the channel half-width, L_y, and ε_g is the skewing coefficient, where $\varepsilon_g \rightarrow 0$ reduces back to a uniform grid distribution and a larger value of ε_g produces smaller cell widths near the wall ($a = 0$). Note that by employing a factor of 2 on a, such that $\left[1 - \frac{2a}{\mathcal{N}}\right]$ in Equation (105), produces a full channel profile with full width $2L_y$ spanning $\mathcal{N}$ cells.

6. *Volume-averaging error*

An error is inherent in a finite-volume (FV) method for moments simply from performing a cell volume average.[44] Consider a 1-D first-order flux scheme such that a $\beta = 1$-point quadrature (single weight and abscissa) uses 1-D linear piecewise interpolation, $\phi(x) \equiv \phi + x\nabla\phi$ and $v(x) \equiv v + x\nabla v$, where ϕ and v indicate the cell-center values and the 1-D slopes ($\frac{\partial\phi}{\partial x} = \nabla\phi$, $\frac{\partial v}{\partial x} = \nabla v$) define a line as a function of scalar position x. Then the moments of a linearly interpolated cell quantity are given as,

$$M_{\Delta x}^{(\gamma)} = \frac{1}{\Delta x} \int_{-\Delta x/2}^{\Delta x/2} \phi(x) v(x)^{\gamma} dx \tag{106}$$

Solving that equation for the first four moments yields,

$$\begin{aligned}
M^{(0)}_{\Delta x} &= \phi \\
M^{(1)}_{\Delta x} &= \phi v + \frac{1}{12}\nabla\phi\nabla v(\Delta x)^2 \\
M^{(2)}_{\Delta x} &= \phi v^2 + \frac{1}{12}\nabla v(\phi\nabla v + 2\nabla\phi v)(\Delta x)^2 \\
M^{(3)}_{\Delta x} &= \phi v^3 + \frac{1}{4}v\nabla v(\phi\nabla v + \nabla\phi v)(\Delta x)^2 + \frac{1}{80}\nabla\phi\nabla v^3(\Delta x)^4
\end{aligned} \tag{107}$$

Notice that the first term in each line is the correct value, but all the extra terms (that depend on Δx) are errors caused by nonzero interpolation slopes, $\nabla\phi \neq 0$ or $\nabla v \neq 0$. Although a zero-order flux scheme eliminates the averaging errors, it is too numerically diffusive (1st-order accurate); in fact, a 1st-order (2nd-order accurate) scheme is recommended.[53] Reducing the slopes is also not practical, but reducing the size of Δx will help reduce the errors. Consider also that these errors will impact the corresponding PDF representation. For example, the variance computed from Equation (65) of these cell-averaged moments of Equation (107) is,

$$\sigma^2_{\Delta x} = \frac{(\nabla v\Delta x)^2}{12}\left(1 - \left(\frac{\nabla\phi\Delta x}{\sqrt{12}\phi}\right)^2\right) \tag{108}$$

Therefore, the variance is positive, resulting entirely from both errors in $M^{(1)}_{\Delta x}$ and $M^{(2)}_{\Delta x}$ being slightly different in value. A single node (Dirac delta function) was given in the original node configuration that constructed the moment set, and should have a zero variance. In other words, the very act of representing the nodes with a non-constant piecewise interpolation (linear or otherwise), misaligns the moments in the cells even before any flux/transportation occurs. Moment alignment is discussed further in the results Section §*1. Moment Misalignment*.

7. CFL Condition

The Courant-Friedrichs-Lewy (CFL) condition ensures that during a single time step the solution does not change too rapidly. This is critical when computing the transport of states through finite cells, especially where the numerical stability of hyperbolic differential equations is concerned.[64] If the CFL number is too large, the values might overshoot and lead to spurious oscillations and instabilities, which is detrimental. A TVD method is extremely important for maintaining non-negative state quantities, e.g. "mass", $M^{(0)}$ and energy, $M^{(2)}$, etc. The 1-D definition of the CFL number for any cell in the mesh is

$$1 \geq \text{CFL} \geq \frac{\mathbb{V}\Delta t}{\Delta x} \tag{109}$$

where $\mathbb{V}$ is the characteristic velocity scale for the particle phase (or fluid phase, depending which aspect is under consideration) within a finite cell of width Δx, and a time step size of Δt (which does not necessarily need to remain constant throughout the simulation). Given satisfactory initial conditions and only drag forces with no other external driving forces, the dispersed particle velocity scale will always be less than or equal to the carrier fluid velocity scale, $\mathbb{V} \leq \mathbb{U}$. Generally, the larger of the two velocities will be the limiting factor in the CFL condition. Let λ represent the ratio of the time step size and cell width,

$$\lambda \equiv \frac{\Delta t}{\Delta x} \leq \frac{\text{CFL}}{\mathbb{V}} \tag{110}$$

If Δt is constant and the mesh is static (such that the cell spacings Δx do not change over time) then it is computationally advantageous and efficient to compute $\lambda_a = \frac{\Delta t}{\Delta x_a}$ (at cell index a) for the entire mesh during initialization because λ_a appears directly in the convective term of most PDE transport equations and the *local* CFL condition is readily tested by,

$$\lambda_a \mathbb{V}_a \leq \text{CFL} \tag{111}$$

which is a necessary, but not always sufficient condition for numerical stability.[64] During implementation within the current work however, it was shown that the CFL condition perfectly predicted the Δt stability criteria that prevented the formation of negative weights in a 1st-order time and spatial accurate CQMOM model of a free jet crossing ($\beta = 2, D = 2$).

Recall that the CQMOM convective term includes up to $\alpha = 1, ..., \mathbb{B}$ unique velocity abscissas, $\hat{\boldsymbol{v}}_\alpha$, for closure (see Section §*1. Flux Splitting*). Therefore, *each* velocity abscissa must satisfy the CFL condition individually, or rather, it suffices to check that the largest velocity abscissa at a single grid point meets the condition. Let $\max_\alpha(*)$ return the largest value present among all nodes, and $\max_a(*)$ be the largest value across spatial index a. Then for the multidimensional case ($d = 1, ..., D$), the CFL restriction is limited by the maximum local particle velocity at the local grid spacing summed across all dimensions,

$$\Delta t \left[\max_{a} \left(\sum_{d=1}^{D} \frac{\max_{\alpha}(\hat{v}_{d,\alpha,a})}{\Delta x_{d,a}} \right) \right] \leq \text{CFL} \tag{112}$$

C. Optimal Moment Set

All of the inversion sequences performed during flux-splitting require a subset of 2β moments for the 1-D Wheeler β-point inversion. The optimal moment set, $\boldsymbol{M}$, contains the union of all subsets that are required for closure. In the present work, this optimal moment set vector has length $K = 32$ in $\mathcal{X} = 3$ dimensional internal space ($D = 3, Q = 0$) with $\beta = 2$-point quadrature, defined generally by,[44]

$$K = (\mathcal{X} + 1)\beta^{D} \tag{113}$$

where K is also the number of elements in the node matrix $\mathbb{N}$ that provides a multidimensional quadrature representation of the PDF. The K corresponding moments, $M^{(\gamma)}_{l_1,l_2,l_3}$, are stored and evolved via the transport equations listed in Equation (49), providing a complete closed mapping and vice versa, $\boldsymbol{M} \leftrightarrow \mathbb{N}$. Table 9 lists the optimal combinations of l_1, l_2, l_3 that construct *all* the 0th, 1st, 2nd, and 3rd order optimal moments, and *some* of the 4th and 5th order moments,[44] providing K total moments.

Table 9. Optimal Moment Permutations for $\mathcal{X}$ =3 Dimensional State Space With $\beta = 2$-Point Quadrature, $\mathbb{E}(m, d)$

m :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
γ :	0	1			2						3										4									5		
(d=1) l_1	0	1	0	0	2	1	1	0	0	0	3	2	2	1	1	1	0	0	0	0	3	3	1	1	0	0	2	1	1	3	1	1
(d=2) l_2	0	0	1	0	0	1	0	2	1	0	0	1	0	2	1	0	3	2	1	0	0	1	3	0	3	1	1	2	1	1	3	1
(d=3) l_3	0	0	0	1	0	0	1	0	1	2	0	0	1	0	1	2	0	1	2	3	1	0	0	3	1	3	1	1	2	1	1	3

Table 9 also shows an extremely useful index conversion, because the *linearized* index m allows the moment set to be stored/referenced as a 1D vector, $M^{(\gamma)}_{[m]}$ $\Leftrightarrow M^{(\gamma)}_{l_1,l_2,l_3}$, instead of using the multidimensional indices, $\{l_1, l_2, l_3\}$. From a numerical/computational standpoint, this is advantageous and more efficient than attempting to store varying length vectors in a multi-dimensional array. Furthermore, the look-up table function $\mathbb{E}(m, d)$ is easily implementable corresponding to the dth row and mth column per Table 9. Note that the set of $K = 32$ moments in Table 9 involves only the optimal moments of order up to $\gamma \leq 5$, and is valid only for exactly $\mathcal{X} = 3$ dimensions of the internal state space and $\beta = 2$ node quadrature. If a different number are required, the table must be reconstructed to define all optimal index permutations $(l_1, l_2, l_3, l_4, ...)$ with the custom total K determined by Equation (113).

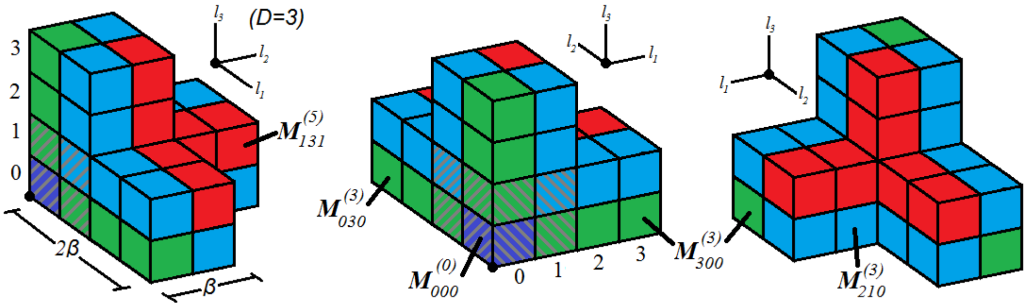


Figure 15. Visual Representation of $K = 32$ Optimal Moments for $\mathcal{X} = 3$ and $\beta = 2$

Determining the optimal moment set is a matter of combining all the moment subsets of pertinent 1-D inversions, i.e. the union of all subset produces the optimal set, $\boldsymbol{M}$. Figure 15 offers a graphical representation of Table 9. The zero order moment, $\{l_1, l_2, l_3\} = \{0, 0, 0\}$, (purple colored) is the “cornerstone” of the moment set and is the total weight (area under the PDF); the green moments are the “simple” moments because only one of the indices (l_1, l_2, or l_3) is non-zero; the blue moments are “mixed” moments that contain two non-zero indices; and the red moments are fully-mixed with all non-zero indices ($l_1, l_2, l_3 \neq 0$). Core moments are those indicated by the hatch shading and exhibit $l_1, l_2, l_3 < \beta$ for *all* indices. The total dimension $\mathcal{X}$ describes the number of “arms” in the figure, while the number of nodes, β, describes the “thickness” of each arm and 2β is the “length” of each arm. This arm length is significant because each subset of $2\beta = 4$ adjacent moments (in a straight line) provides a 1-D moment β-point inversion. Many of the inversions share the core moments among the different $\mathcal{X}$ number of “arms.” As an example, the moment subset, $\{M^{(1)}_{010}, M^{(2)}_{011}, M^{(3)}_{012}, M^{(4)}_{013}\}$, is conditioned on the y-dimension where $l_2 = 1$ and provides a z-dimension inversion where $l_3 = 0, 1, 2, 3$. In fact, there are exactly $\mathcal{X} \cdot \beta^{\mathcal{X}-1}$ unique inversions possible within the optimal moment set and all of these inversion subsets share the $\beta^{\mathcal{X}}$ *core* moments .

Figure 15 also calls-out a few selected moments for reference.

Table 10. Example Optimal Moment Counts, K

$\mathcal{X} =$	1	1	2	2	**3**	2	4	3	5	3	4
$\beta =$	2	3	2	3	**2**	4	2	3	2	4	3
$K =$	4	6	12	27	**32**	48	80	108	192	256	405

For comparison, Table 10 lists the moment count, K, needed for various internal state-space dimension $\mathcal{X}$ of the PDF given various β-point quadrature representation. The values of the single case employed in the current work are shown in the bold column, and some of the other columns have been implemented in previous works,[44] but CQMOM offers the ability to go beyond $K > 4$. As can be seen, the number of moments and thus transport equations and nested inversion sequences quickly grows with the addition of more nodes and dimensions. To date, these higher order CQMOM sets, $K > 48$, have not been tested for numerical feasibility/stability.

Recall that in Table 9, only *some* of the 3-D moment permutations are retained for the higher order moments $\gamma \geq 4$. Higher order moments are more prone to numerical errors and it is thus desirable to operate with the lowest order moments possible. In other words, first exhaust all the lower order permutations (core moments) before considering higher order permutations which are also only pertinent to one respective dimension. It is possible to achieve the total of K moments (needed for closure of $\mathbb{N}$) without extending the permutations of the mixed moments beyond the core moments in *more than one* dimension. To demonstrate this, let $\mathcal{X}$=D+Q be the total internal dimensions of the PDF, then a generalized moment, $M^{(\gamma)}_{l_1,l_2,l_3,l_4,...}$, has $\mathcal{X}$ internal dimensions represented by the multiset of exponents (written in the subscripts) and their summation produces the moment order, $l_1+l_2+l_3+l_4+... = \gamma$. An optimal moment will have permutations limited to the set $\{0, 1, 2, ..., 2\beta - 1\}$, which extends exactly to the maximum value needed for a 1-D inversion. Furthermore, some additional restrictions may apply to the multiplicity of each element. Multiplicity is defined as the number of times that an element is repeated within the multiset. An optimal moment will have only *one* element larger than or equal to $l_1, l_2, l_3, l_4 \geq \beta$, meaning that only *one* dimension extends beyond the core set of moments. Then the total number of moment permutations is given by a combinatorics approach with binomial notation,

$$\binom{\mathcal{X}}{\mathcal{L}_1, \mathcal{L}_2, ...} \equiv \frac{\mathcal{X}!}{\mathcal{L}_1!\mathcal{L}_2!...} \tag{114}$$

where $\mathcal{X}$ is the total number of dimensions, and $\mathcal{L}_i$ are the multiplicities of elements in $\boldsymbol{l} \equiv \{l_1, l_2, l_3, l_4, ...\}$, which are themselves also multiplicities (of the abscissas used to generate moments).

For example, to determine the number of optimal fourth order moments, $M^{(4)}$, let $\beta = 2$ and $\mathcal{X}$ be the arbitrary length of multiset $\boldsymbol{l}$, then its elements must always sum to $\gamma = 4$. Apply Equation (114), but allow only *one* 2 or 3 element to exist in the multiset, and fill the remaining elements with 1's or 0's to achieve $l_1 + l_2 + l_3 + l_4 + ... = 4$. It should be apparent that three types of multisets are then possible, those involving the elements $\{3, 1, 0, ..., 0\}$, $\{2, 1, 1, 0, ..., 0\}$, and $\{1, 1, 1, 1, 0, ..., 0\}$, each having their own set of unique permutations and always summing to $\gamma = 4$. Then the number of permutations from each multiset may be summed together to obtain the total quantity of permutations, and is represented mathematically as,

$$\binom{\mathcal{X}}{1, 1, \mathcal{X} - 2} + \binom{\mathcal{X}}{1, 2, \mathcal{X} - 3} + \binom{\mathcal{X}}{4, \mathcal{X} - 4} \tag{115}$$

If a negative factorial is encountered then the entire binomial term is considered zero. The last integer values in the denominators represent the multiplicity of 0's, which is simply the total dimension minus the multiplicity of all other elements. Expanding the values of Equation (115), a single non-repeated digit (multiplicity of one) reduces to $1!$ in the denominator and can be ignored, but each repeated digit (multiplicity greater than one, $\mathcal{L}_i > 1$) will appear in the denominator of Equation (115). Note that the 3rd multiset term has four 1's, yielding the "4!" value, and therefore must have a multiplicity of $\mathcal{X} - 4$ for the 0's. Additionally, the 3rd term is non-zero only when $\mathcal{X} \geq 4$, so as to ignore any negative factorials.

$$\frac{\mathcal{X}!}{(\mathcal{X}-2)!} + \frac{\mathcal{X}!}{2!(\mathcal{X}-3)!} + \frac{\mathcal{X}!}{4!(\mathcal{X}-4)!} \tag{116}$$

Thus, the quantity of 4th-order moments $M^{(4)}_{l_1,l_2,l_3,l_4,...}$ depends on the number of nodes (β) and dimensions available, i.e. the length of multiset $\{l_1, l_2, l_3, l_4, ...\}$, which is $\mathcal{X}$. Table 11 lists the quantities of optimal moments (given a fixed $\beta = 2$), for unique permutations of the multiset and the general expression is given in the 2nd column. The last row of Table 11 lists the expected total number of optimal moments (K) required for closure of $\mathbb{N}$, and is recovered by summing all quantities of unique ordered moments in the previous rows. This $\beta = 2$ case agrees with the totals listed previously in Table 10.

As a final note, it is worth mentioning that it is *not* mandatory for β to be equal in *all* dimensions, as it has been so far described. Instead, each dimension could receive a distinguished node count, β_d, for customizing the resolution of the PDF in each dimension. For example, in a 3-D boundary layer flow the wall-normal particle velocities (v_2) will exhibit more diversity than either the streamwise or spanwise velocities (v_1 and v_3); therefore, adding an extra node in only the wall-normal direction may be beneficial, e.g. $\{\beta_1, \beta_2, \beta_3\} = \{2, 3, 2\}$, as opposed to requiring a full 3-point quadrature in all

Table 11. Optimal Ordered Moment Counts of Varying $\mathcal{X}$ for $\beta = 2$

		$\mathcal{X} = D + Q =$					
γ	Expression	1	2	3	4	5	6
0	$\binom{\mathcal{X}}{\mathcal{X}}$	1	1	1	1	1	1
1	$\binom{\mathcal{X}}{\mathcal{X}-1}$	1	2	3	4	5	6
2	$\binom{\mathcal{X}}{\mathcal{X}-1} + \binom{\mathcal{X}}{2,\mathcal{X}-2}$	1	3	6	10	15	21
3	$\binom{\mathcal{X}}{\mathcal{X}-1} + \binom{\mathcal{X}}{\mathcal{X}-2} + \binom{\mathcal{X}}{3,\mathcal{X}-3}$	1	4	10	20	35	56
4	$\binom{\mathcal{X}}{\mathcal{X}-2} + \binom{\mathcal{X}}{2,\mathcal{X}-3} + \binom{\mathcal{X}}{4,\mathcal{X}-4}$	0	2	9	25	55	105
5	$\binom{\mathcal{X}}{2,\mathcal{X}-3} + \binom{\mathcal{X}}{3,\mathcal{X}-4} + \binom{\mathcal{X}}{5,\mathcal{X}-5}$	0	0	3	16	51	126
6	$\binom{\mathcal{X}}{3,\mathcal{X}-4} + \binom{\mathcal{X}}{4,\mathcal{X}-5} + \binom{\mathcal{X}}{6,\mathcal{X}-6}$	0	0	0	4	25	91
7	$\binom{\mathcal{X}}{4,\mathcal{X}-5} + \binom{\mathcal{X}}{5,\mathcal{X}-6} + \binom{\mathcal{X}}{7,\mathcal{X}-7}$	0	0	0	0	5	36
8	$\binom{\mathcal{X}}{5,\mathcal{X}-6} + \binom{\mathcal{X}}{6,\mathcal{X}-7} + \binom{\mathcal{X}}{8,\mathcal{X}-8}$	0	0	0	0	0	6
Sum:	$K = (\mathcal{X} + 1)\beta^{\mathcal{X}}$	4	12	32	80	192	448

dimensions which would be far more costly. However, the dimension containing the largest node count would need to be inverted *last* in the multidimensional sequence, and the previous formulations will need to be extended to account for unique values of β_d.

D. Time Iteration Scheme

An improved order of time-accuracy is available using the 2nd order Runge-Kutta Strong-Stability-Preserving (RK2SSP)[45,75] scheme defined as,

$$\begin{aligned}
\overset{(1)}{\boldsymbol{M}}_a &:= \overset{(n)}{\boldsymbol{M}}_a - \frac{\Delta t}{\Delta x_{d,a}} \left[\boldsymbol{G}(d, \overset{(n)}{\mathbb{N}}_{a+\frac{1}{2},\ell}, \overset{(n)}{\mathbb{N}}_{a+\frac{1}{2},r}) - \boldsymbol{G}(d, \overset{(n)}{\mathbb{N}}_{a-\frac{1}{2},\ell}, \overset{(n)}{\mathbb{N}}_{a-\frac{1}{2},r}) \right] \\
\overset{(2)}{\boldsymbol{M}}_a &:= \overset{(1)}{\boldsymbol{M}}_a - \frac{\Delta t}{\Delta x_{d,a}} \left[\boldsymbol{G}(d, \overset{(1)}{\mathbb{N}}_{a+\frac{1}{2},\ell}, \overset{(1)}{\mathbb{N}}_{a+\frac{1}{2},r}) - \boldsymbol{G}(d, \overset{(1)}{\mathbb{N}}_{a-\frac{1}{2},\ell}, \overset{(1)}{\mathbb{N}}_{a-\frac{1}{2},r}) \right] \qquad (117) \\
\overset{(n+1)}{\boldsymbol{M}}_a &:= \frac{1}{2} \left[\overset{(n)}{\boldsymbol{M}}_a + \overset{(2)}{\boldsymbol{M}}_a \right]
\end{aligned}$$

where a is the spatial index in a single dimension, and the overscript value (n) is the time step, while overscripts (1) and (2) are placeholders for internal Runge-Kutta iterations. This formulation requires that the entire CQMOM operation be carried out during iteration (1), where all $(D-1)!^D$ flux paths must be performed using Equation (92), then proceed to repeat the entire operation again for iteration (2). Note that the full domain set of moments $\overset{(n)}{\boldsymbol{M}}$ must be memorized while $\overset{(2)}{\boldsymbol{M}}$ may overwrite $\overset{(1)}{\boldsymbol{M}}$. In the final step of Equation (117), the initial and final moments are averaged together. Although this process *doubles* the computational requirements compared to a first-order time accurate scheme, it provides second-order accurate time resolution. Higher order R-K methods could be employed by extending the format of Equation (117).

E. Source Terms

Included in the moment transport equation are the additional source terms, $\mathbb{D}$ and $\mathbb{C}$, that may be added to the RHS of the equation to model additional particle interactions. Some of the terms may deal directly with the abscissas and weights ($\mathbb{N}$) while others deal with the moments ($\boldsymbol{M}$), but they can both be applied using operator-splitting technique as described in Section §*2. Operator Splitting Technique*. Firstly, the acceleration term has many contributing factors such as drag, gravity, or other body forces, all of which may be grouped together into a generic net acceleration source term $\mathbb{D}$ that deals exclusively with the nodes, $\mathbb{N}$ and must be repeated once in each flux dimension, $1, ..., D$. Secondly, the inter-particle collision term, $\mathbb{C}$, deals with the moments exclusively and thus must only be applied once per timestep, typically after the flux-splitting is complete in all dimensions.

These source terms are optional, and it is important to consider that they may operate on very different time scales as compared to the flux scheme's CFL condition, and that the stability of the model may be compromised. First, the collision model operates at the time scale τ_c, which may become very small in dense regions experiencing high granular temperature. As $\tau_c \to 0$, the collision term, $\mathbb{C}$, allows for drastic changes in $\boldsymbol{M}$ within a single time step, thus the model stability is at stake if the time step size, Δt, is not sufficiently small. Similarly, the drag term, $\mathbb{D}$, influences particles at the timescale τ_p and low inertia particles may respond rapidly to changes in the carrier fluid. Introducing a relaxation time coefficient, κ, that exponentially approaches a "steady-state" solution, can help remedy these issues. For example, it can prevent slow moving particles from accelerating past the fluid speed (overshooting).

1. Inter-Particle Collisions

A source term may be added to the Boltzmann equation to model inter-particle collisions[76] from a statistical mechanics viewpoint; see Section §*G. Viscosity Limitations*. One approach for modeling the collision effects is the Bhatnagar-Gross-Krook (BGK) approximation.[51,77] The concept assumes a mean free-path treatment that replaces a collision integral with a *collision relaxation time*, τ_c, from Equation (16). The difference between f and f_{eq} can be considered a perturbation method, and the source term is amplified by the increased collision frequency,

$$C = \frac{1}{\tau_c}\Big(f_{eq}(\boldsymbol{v}) - f(\boldsymbol{v})\Big) \tag{118}$$

where the multivariate Gaussian[1,33,44,45] distribution, $f_{eq}(\boldsymbol{v})$, is computed by copying many features from $f(\boldsymbol{v})$,

$$f_{eq}(\boldsymbol{v}) = \frac{M^{(0)}}{(2\pi\Theta_{eq})^{D/2}} \exp\left(-\frac{|\boldsymbol{v} - \langle\boldsymbol{v}\rangle|^2}{2\Theta_{eq}}\right) \tag{119}$$

The "mass", $M^{(0)}$, mean velocity, $\langle\boldsymbol{v}\rangle$, and equilibrium temperature, Θ_{eq}, are all obtained from $f(\boldsymbol{v})$. However, $f_{eq}(\boldsymbol{v})$ by definition has zero skew and equal variance in all directions: Θ_{eq}. In other words, f_{eq} is the "equivalent" representation of f with maximum entropy, centered symmetrically about the mean velocity, $\langle\boldsymbol{v}\rangle$, with variance Θ_{eq} in all dimensions.

In the special case that the multidimensional variance of $f(\boldsymbol{v})$ is zero ($\Theta_{eq} = 0$) then $f(\boldsymbol{v}) = f_{eq}(\boldsymbol{v}) = \delta$ is the same Delta function and the collision term is null. This follows since no particles are crossing trajectories and they all move uniformly with the same velocity, which also corresponds to the *lowest* entropy state possible. It is important to realize that the collision term, C, attempts to transform $f(\boldsymbol{v})$ into $f_{eq}(\boldsymbol{v})$, but the rate at which this occurs depends on $1/\tau_c$; thus is also characterized by the Knudsen number. The more frequent the collisions ($\tau_c \to 0$), the more rapidly the velocity distribution can return to an equilibrium Gaussian distribution of maximum entropy. A truly collisionless state ($\tau_c \to \infty$) is also possible in the rarefied limit $M^{(0)} \to 0$, and also nullifies the collision source term.

Although $f_{eq}(\boldsymbol{v})$ pertains to the random motion of perfectly elastic ($\omega = 1$) fluid molecules, it does have some basis for use in the motion of granular particles too. Recall that granular particles differ from their molecular counterparts in that they are inelastic and dissipative,[7] and a time-independent Maxwellian function is no longer a solution to the Boltzmann equation, but can serve as a starting point for a perturbation expansion.[78] The assumption that $\omega \approx 1$ for granular particles is made for simplification, but inelastic models are also currently available[1,36,44] using the function, $\widetilde{f}_{eq}(\boldsymbol{v})$,

$$\widetilde{f}_{eq}(\boldsymbol{v}) = \frac{M^{(0)}}{[\det(2\pi\mathbb{F})]^{D/2}} \exp\left(-\frac{1}{2}\Big(v_i - \langle v_i\rangle\Big)\left(\mathbb{F}^{-1}\right)_{ij}\Big(v_j - \langle v_j\rangle\Big)\right) \tag{120}$$

where the repeated Latin subscripts imply summation of the various dimensions. The second order tensor, $\mathbb{F}$, is given by,[1]

$$\mathbb{F} = \Upsilon\omega^2\Theta_{eq}\boldsymbol{I} + \left(\Upsilon\omega^2 - 2\Upsilon\omega + 1\right)\boldsymbol{\theta} \tag{121}$$

The moments of f_{eq} are constructed similarly to Equation (45), but depend on the equilibrium distribution instead,

$$\mathbb{C}^{(\gamma)}_{l_1,l_2,l_3} = M^{(0)}\Lambda^{(\gamma)}_{l_1,l_2,l_3} = \int v_1^{l_1} v_2^{l_2} v_3^{l_3} f_{eq}(t, \boldsymbol{x}; \boldsymbol{v}) d\boldsymbol{v} \tag{122}$$

Mass and momentum conservation (during elastic and inelastic collisions alike) ensures that the zero and first order moments of f and f_{eq} are identical, while a sink/source term, $\mathbb{C}^{(\gamma)}_{l_1,l_2,l_3}$, will appear only in the 2nd-order and higher moment transport Equations (49). It should be noted that the kinetic theory collision approximation will be accurate for the 2nd-order moment, but higher order moments may be poorly modeled for a distribution far from equilibrium.[1] The total weight, $M^{(0)}$, remains a factor on all $\Lambda^{(\gamma)}_{l_1,l_2,l_3}$ terms and so it is kept as an external coefficient. Each of these elastic collision terms may be approximated as follows for isotropic collisions,

$$\begin{aligned}
\Lambda^{(0)}_{0,0,0} =& \quad 1 \\
\Lambda^{(1)}_{l_1,l_2,l_3} =& \quad \langle v_1 \rangle^{l_1} \langle v_2 \rangle^{l_2} \langle v_3 \rangle^{l_3} \\
\Lambda^{(2)}_{l_1,l_2,l_3} =& \quad \left(\Theta_{eq} + \langle v_1 \rangle^2\right) \delta_{l_1,2} + \left(\Theta_{eq} + \langle v_2 \rangle^2\right) \delta_{l_2,2} + \left(\Theta_{eq} + \langle v_3 \rangle^2\right) \delta_{l_3,2} + \\
& \quad \langle v_1 \rangle \langle v_2 \rangle \delta_{l_1,l_2} + \langle v_1 \rangle \langle v_3 \rangle \delta_{l_1,l_3} + \langle v_2 \rangle \langle v_3 \rangle \delta_{l_2,l_3} \\
\Lambda^{(3)}_{l_1,l_2,l_3} =& \quad ...
\end{aligned} \tag{123}$$

Table 12 shows the full terms expanded up to the 5th order moments ($\gamma \leq 5$) required for vector $\mathbb{C}$ (of length $K = 32$) and they become sophisticated. These equations also includes the notation that is compatible with *anisotropic* collisions. To obtain the more generic isotropic collision formulas of Equation (123), ignore the off-diagonal terms of the covariance matrix ($\theta_{1,2} = \theta_{1,3} = \theta_{2,3} = 0$), and replace the diagonal terms with the scalar equilibrium temperature ($\theta_{1,1} = \theta_{2,2} = \theta_{3,3} = \Theta_{eq}$). The isotropic assumption relies on the single condensed scalar temperature, Θ_{eq}, obtained by averaging all dimensional granular temperatures into one, as described in Equation (68). Consequently, an isotropic collision model may cause undesired results in regions that are highly anisotropic. In other words, the isotropic simplification provides a mechanism for momentum transfer to occur between different dimensions by equalizing the collisions in all directions, but

Table 12. Collision Terms Up to 4th-Order

$\Lambda^{(0)}_{0,0,0} = 1$	$\Lambda^{(3)}_{3,0,0} = 3\left\langle v_1\right\rangle \theta_{1,1} + \left\langle v_1\right\rangle^3$
$\Lambda^{(1)}_{1,0,0} = \left\langle v_1\right\rangle$	$\Lambda^{(3)}_{2,1,0} = 2\left\langle v_1\right\rangle \theta_{1,2} + \left\langle v_2\right\rangle \theta_{1,1} + \left\langle v_1\right\rangle^2 \left\langle v_2\right\rangle$
$\Lambda^{(1)}_{0,1,0} = \left\langle v_2\right\rangle$	$\Lambda^{(3)}_{2,0,1} = 2\left\langle v_1\right\rangle \theta_{1,3} + \left\langle v_3\right\rangle \theta_{1,1} + \left\langle v_1\right\rangle^2 \left\langle v_3\right\rangle$
$\Lambda^{(1)}_{0,0,1} = \left\langle v_3\right\rangle$	$\Lambda^{(3)}_{1,2,0} = 2\left\langle v_2\right\rangle \theta_{1,2} + \left\langle v_1\right\rangle \theta_{2,2} + \left\langle v_1\right\rangle \left\langle v_2\right\rangle^2$
$\Lambda^{(2)}_{2,0,0} = \theta_{1,1} + \left\langle v_1\right\rangle^2$	$\Lambda^{(3)}_{1,1,1} = \left\langle v_1\right\rangle \theta_{2,3} + \left\langle v_2\right\rangle \theta_{1,3} + \left\langle v_3\right\rangle \theta_{1,2} + \left\langle v_1\right\rangle \left\langle v_2\right\rangle \left\langle v_3\right\rangle$
$\Lambda^{(2)}_{1,1,0} = \theta_{1,2} + \left\langle v_1\right\rangle \left\langle v_2\right\rangle$	$\Lambda^{(3)}_{1,0,2} = 2\left\langle v_3\right\rangle \theta_{1,3} + \left\langle v_1\right\rangle \theta_{3,3} + \left\langle v_1\right\rangle \left\langle v_3\right\rangle^2$
$\Lambda^{(2)}_{1,0,1} = \theta_{1,3} + \left\langle v_1\right\rangle \left\langle v_3\right\rangle$	$\Lambda^{(3)}_{0,3,0} = 3\left\langle v_2\right\rangle \theta_{2,2} + \left\langle v_2\right\rangle^3$
$\Lambda^{(2)}_{0,2,0} = \theta_{2,2} + \left\langle v_2\right\rangle^2$	$\Lambda^{(3)}_{0,2,1} = 2\left\langle v_2\right\rangle \theta_{2,3} + \left\langle v_3\right\rangle \theta_{2,2} + \left\langle v_2\right\rangle^2 \left\langle v_3\right\rangle$
$\Lambda^{(2)}_{0,1,1} = \theta_{2,3} + \left\langle v_2\right\rangle \left\langle v_3\right\rangle$	$\Lambda^{(3)}_{0,1,2} = 2\left\langle v_3\right\rangle \theta_{2,3} + \left\langle v_2\right\rangle \theta_{3,3} + \left\langle v_2\right\rangle \left\langle v_3\right\rangle^2$
$\Lambda^{(2)}_{0,0,2} = \theta_{3,3} + \left\langle v_3\right\rangle^2$	$\Lambda^{(3)}_{0,0,3} = 3\left\langle v_3\right\rangle \theta_{3,3} + \left\langle v_3\right\rangle^3$
$\Lambda^{(4)}_{3,0,1} = 3\theta_{1,1}\theta_{1,3} + 3\theta_{1,3}\left\langle v_1\right\rangle^2 + 3\left\langle v_1\right\rangle \left\langle v_3\right\rangle \theta_{1,1} + \left\langle v_1\right\rangle \left\langle v_3\right\rangle \left\langle v_1\right\rangle^2$	
$\Lambda^{(4)}_{3,1,0} = 3\theta_{1,1}\theta_{1,2} + 3\theta_{1,2}\left\langle v_1\right\rangle^2 + 3\left\langle v_1\right\rangle \left\langle v_2\right\rangle \theta_{1,1} + \left\langle v_1\right\rangle \left\langle v_2\right\rangle \left\langle v_1\right\rangle^2$	
$\Lambda^{(4)}_{1,3,0} = 3\theta_{1,2}\theta_{2,2} + 3\theta_{1,2}\left\langle v_2\right\rangle^2 + 3\left\langle v_1\right\rangle \left\langle v_2\right\rangle \theta_{2,2} + \left\langle v_1\right\rangle \left\langle v_2\right\rangle \left\langle v_2\right\rangle^2$	
$\Lambda^{(4)}_{1,0,3} = 3\theta_{1,3}\theta_{3,3} + 3\theta_{1,3}\left\langle v_3\right\rangle^2 + 3\left\langle v_1\right\rangle \left\langle v_3\right\rangle \theta_{3,3} + \left\langle v_1\right\rangle \left\langle v_3\right\rangle \left\langle v_3\right\rangle^2$	
$\Lambda^{(4)}_{0,3,1} = 3\theta_{2,2}\theta_{2,3} + 3\theta_{2,3}\left\langle v_2\right\rangle^2 + 3\left\langle v_2\right\rangle \left\langle v_3\right\rangle \theta_{2,2} + \left\langle v_2\right\rangle \left\langle v_3\right\rangle \left\langle v_2\right\rangle^2$	
$\Lambda^{(4)}_{0,1,3} = 3\theta_{2,3}\theta_{3,3} + 3\theta_{2,3}\left\langle v_3\right\rangle^2 + 3\left\langle v_2\right\rangle \left\langle v_3\right\rangle \theta_{3,3} + \left\langle v_2\right\rangle \left\langle v_3\right\rangle \left\langle v_3\right\rangle^2$	
$\Lambda^{(4)}_{2,1,1} = 2\theta_{1,2}\theta_{1,3} + \theta_{1,1}\theta_{2,3} + \theta_{2,3}\left\langle v_1\right\rangle^2 + 2\left\langle v_1\right\rangle \left\langle v_2\right\rangle \theta_{1,3} + 2\left\langle v_1\right\rangle \left\langle v_3\right\rangle \theta_{1,2}$ $+ \left\langle v_2\right\rangle \left\langle v_3\right\rangle \theta_{1,1} + \left\langle v_2\right\rangle \left\langle v_3\right\rangle \left\langle v_1\right\rangle^2$	
$\Lambda^{(4)}_{1,2,1} = 2\theta_{1,2}\theta_{2,3} + \theta_{1,3}\theta_{2,2} + \theta_{1,3}\left\langle v_2\right\rangle^2 + 2\left\langle v_2\right\rangle \left\langle v_3\right\rangle \theta_{1,2} + 2\left\langle v_1\right\rangle \left\langle v_2\right\rangle \theta_{2,3}$ $+ \left\langle v_1\right\rangle \left\langle v_3\right\rangle \theta_{2,2} + \left\langle v_1\right\rangle \left\langle v_3\right\rangle \left\langle v_2\right\rangle^2$	
$\Lambda^{(4)}_{1,1,2} = 2\theta_{1,3}\theta_{2,3} + \theta_{1,2}\theta_{3,3} + \theta_{1,2}\left\langle v_3\right\rangle^2 + 2\left\langle v_2\right\rangle \left\langle v_3\right\rangle \theta_{1,3} + 2\left\langle v_1\right\rangle \left\langle v_3\right\rangle \theta_{2,3}$ $+ \left\langle v_1\right\rangle \left\langle v_2\right\rangle \theta_{3,3} + \left\langle v_1\right\rangle \left\langle v_2\right\rangle \left\langle v_3\right\rangle^2$	
$\Lambda^{(5)}_{3,1,1} = 6\left\langle v_1\right\rangle \theta_{1,2}\theta_{1,3} + 3\left\langle v_2\right\rangle \theta_{1,1}\theta_{1,3} + 3\left\langle v_3\right\rangle \theta_{1,1}\theta_{1,2} + 3\left\langle v_1\right\rangle \theta_{1,1}\theta_{2,3}$ $+3\left\langle v_2\right\rangle \theta_{1,3}\left\langle v_1\right\rangle^2 + 3\left\langle v_3\right\rangle \theta_{1,2}\left\langle v_1\right\rangle^2 + \left\langle v_1\right\rangle \theta_{2,3}\left\langle v_1\right\rangle^2$ $+3\left\langle v_1\right\rangle \left\langle v_2\right\rangle \left\langle v_3\right\rangle \theta_{1,1} + \left\langle v_1\right\rangle^3 \left\langle v_2\right\rangle \left\langle v_3\right\rangle$	
$\Lambda^{(5)}_{1,3,1} = 6\left\langle v_2\right\rangle \theta_{1,2}\theta_{2,3} + 3\left\langle v_2\right\rangle \theta_{1,3}\theta_{2,2} + 3\left\langle v_3\right\rangle \theta_{1,2}\theta_{2,2} + 3\left\langle v_1\right\rangle \theta_{2,2}\theta_{2,3}$ $+ \left\langle v_2\right\rangle \theta_{1,3}\left\langle v_2\right\rangle^2 + 3\left\langle v_3\right\rangle \theta_{1,2}\left\langle v_2\right\rangle^2 + 3\left\langle v_1\right\rangle \theta_{2,3}\left\langle v_2\right\rangle^2$ $+3\left\langle v_1\right\rangle \left\langle v_2\right\rangle \left\langle v_3\right\rangle \theta_{2,2} + \left\langle v_1\right\rangle \left\langle v_2\right\rangle^3 \left\langle v_3\right\rangle$	
$\Lambda^{(5)}_{1,1,3} = 6\left\langle v_3\right\rangle \theta_{1,3}\theta_{2,3} + 3\left\langle v_2\right\rangle \theta_{1,3}\theta_{3,3} + 3\left\langle v_3\right\rangle \theta_{1,2}\theta_{3,3} + 3\left\langle v_1\right\rangle \theta_{2,3}\theta_{3,3}$ $+3\left\langle v_2\right\rangle \theta_{1,3}\left\langle v_3\right\rangle^2 + \left\langle v_3\right\rangle \theta_{1,2}\left\langle v_3\right\rangle^2 + 3\left\langle v_1\right\rangle \theta_{2,3}\left\langle v_3\right\rangle^2$ $+3\left\langle v_1\right\rangle \left\langle v_2\right\rangle \left\langle v_3\right\rangle \theta_{3,3} + \left\langle v_1\right\rangle \left\langle v_2\right\rangle \left\langle v_3\right\rangle^3$	

treating all dimensions equally in collision strength when they actually exhibit quite different temperatures, $\Theta_d \neq \Theta_{eq}$, may yield unphysical results.

Finally, the full quadrature approximation of the collision term from Equation (118) depends linearly upon the difference between the equilibrium moments, $M^{(0)}\Lambda^{(\gamma)}_{l_1,l_2,l_3}$, and the actual moments, $M^{(\gamma)}_{l_1,l_2,l_3}$,

$$\mathbb{C}^{(\gamma)}_{l_1,l_2,l_3} = \frac{1}{\tau_c}\left[(M^{(0)}\Lambda^{(\gamma)}_{l_1,l_2,l_3}) - M^{(\gamma)}_{l_1,l_2,l_3}\right] \tag{124}$$

Note that the 0th and 1st-order collision terms are identical to their respective moments, $M^{(0)}\Lambda^{(\gamma)} \equiv M^{(\gamma)}$ for $0 \leq \gamma \leq 1$. Therefore, the collision term, $\mathbb{C}$, will cancel-out (and has no effect) until the 2nd-order or higher moment transport equations. This is representative of the mass and momentum conservation of (in)elastic collisions. For elastic collisions, energy is also conserved; however, since the energy is being transferred between dimensions using the average value Θ_{eq}, the individual 2nd-order moments are *not* necessarily conserved.

2. *Operator Splitting Technique*

To improve the stability of the moment transport equations and help ensure the realizability of the moment set, the contributions from source terms $\mathbb{D}$ and $\mathbb{C}$ are *not* directly added into the moment equations as shown in Equation (49), instead, both of these source terms are applied separately from the typical flux scheme. The convective term updates the moments in the PDE per Equation (117); however, before converting the nodes back into moments with $\boldsymbol{G}$, the drag term is applied while the nodes $\mathbb{N}$ are in the proper format described directly by $f(\hat{\boldsymbol{v}})$.

The operator splitting technique will *apply* the acceleration source term, $\mathbb{D}$, by influencing the abscissas directly.[42] A first-order ODE assumption allows an exponential relaxation to alter the velocities,

$$\begin{aligned} \overset{(n+1)}{\phi_\alpha} &:= \overset{(n)}{\phi_\alpha} \\ \overset{(n+1)}{\hat{\boldsymbol{v}}_\alpha} &:= \kappa_p \overset{(n)}{\hat{\boldsymbol{v}}_\alpha} + (1-\kappa_p)(\overset{(n)}{\boldsymbol{u}} + ...) \end{aligned} \tag{125}$$

where n is the time step index, $\boldsymbol{u}$ is the carrier fluid velocity, τ_p is the particle response time defined in Equation (10), and $\kappa_p = e^{-\Delta t/\tau_p}$ is the particle drag time *coefficient*. Therefore, the particle velocity $\boldsymbol{v}$ will tend toward $\boldsymbol{u}$ in an exponentially decaying manner dependent upon the time coefficient that describes how “laggy” the particle is (Stokes number). Note that the new weight $\overset{(n+1)}{\phi_\alpha}$ remains unchanged. This method is applicable for any net drag/gravity/body forces desired within the $\mathbb{D}$ term. An example specifically for the drag/gravity only case in 3-D is,

$$\begin{aligned} \overset{(n+1)}{\boldsymbol{U}} &:= \kappa_p \overset{(n)}{\boldsymbol{U}} + (1-\kappa_p)(\overset{(n)}{u_1} + \tau_p g_1) \\ \overset{(n+1)}{\boldsymbol{V}} &:= \kappa_p \overset{(n)}{\boldsymbol{V}} + (1-\kappa_p)(\overset{(n)}{u_2} + \tau_p g_2) \\ \overset{(n+1)}{\boldsymbol{W}} &:= \kappa_p \overset{(n)}{\boldsymbol{W}} + (1-\kappa_p)(\overset{(n)}{u_3} + \tau_p g_3) \\ \overset{(n+1)}{\boldsymbol{\phi}} &:= \overset{(n)}{\boldsymbol{\phi}} \end{aligned} \tag{126}$$

where $\boldsymbol{g}$ is the gravitational acceleration vector. $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$, and $\boldsymbol{\phi}$ are vectors of length $\mathbb{B}$ containing all node info from $\mathbb{N}$; see Equation (54). However, only the relevant dimension d is applied during each inversion-flux pair operation in CQMOM. It is then straightforward to add additional acceleration terms into Equation (126) as needed.

Next, the same ideology is applicable for the collision source term vector, $\mathbb{C}$, applied to the moments, $\boldsymbol{M}$, using the exponential solution to a first-order differential equation,

$$\overset{(n+1)}{\boldsymbol{M}} := \overset{(n)}{\kappa_c}\overset{(n)}{\boldsymbol{M}} + (1 - \overset{(n)}{\kappa_c})\overset{(n)}{\mathbb{C}} \tag{127}$$

The collision time, τ_c defined in Equation (13), influences the particle collision time *coefficient*, $\kappa_c = e^{-\Delta t/\tau_c}$, locally since it depends upon the local zero order moment, M_0. Again, the exponential decay ensures that the moments will not overshoot and become negative because $0 \leq \kappa \leq 1$. The zero and first order moments are not changed, but all the higher order moments are updated each time step per Table 12 (Fox *et al.*), using Equation (127).

3. *Boundary Conditions*

Many types of boundary conditions are possible in QMOM. Typically the nodes, $\mathbb{N}$, are altered or restricted to conform to certain desired conditions (using the operator splitting technique) by applying the conditions *after* updating the abscissas and weights due to the acceleration term, but *before* reconstructing the moment set (which also performs the convective term of the moment transport). Periodic boundaries simply copy the node values from the first internal cells to a ghost layer at the other end of the domain for the respective boundary.

Reflective boundaries allow particles to "bounce" off of boundary walls with little added complexity. Consider a Cartesian grid with a planar wall in the *xz*-plane such that particles contacting the wall will have their y-velocity reflected. By copying the velocities on the wall boundary (where location y_0 is a ghost layer) and negating the velocity which is perpendicular to the wall, the neighboring adjacent cells (at y_1) will begin to experience a positive variance ($\Theta_2 > 0$) when the wall-normal velocity $\boldsymbol{V}$ is reflected back (for all nodes). The ghost layer is set according to,[36,39]

$$\begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{V} \\ \boldsymbol{W} \\ \boldsymbol{\phi} \end{bmatrix}_{y_0} := \begin{bmatrix} \boldsymbol{U} \\ -e_w\boldsymbol{V} \\ \boldsymbol{W} \\ \boldsymbol{\phi}/e_w \end{bmatrix}_{y_1} \tag{128}$$

where e_w is the coefficient of restitution[7] and $e_w = 1$ for perfectly elastic reflections. Note that two nodes will immediately be exhausted near reflective boundaries and may make it difficult to model any additional physical phenomenon that is occurring on top of the reflection unless more nodes are allotted ($\beta > 2$).

F. High Performance Computing

The DNS fluid solver was converted from FORTRAN into C++ and merged with the CQMOM solver natively written in C++ based on the MATLAB 3-D Wheeler inversion provided by Yuan *et al.*[44] The CQMOM algorithm responds well to CPU parallelization when delegating work by spatial *domain decomposition*. This also ensures that the mathematics described thus far are wholly applied to every single cell, because each CPU receives a local domain which is part of the global domain topology and information is transferred between the internal boundaries shared by any two neighboring CPU's. The Message Passing Interface (MPI) protocol was employed in the current work. The only pertinent information that requires sharing are the node matrices, $\mathbb{N}$, and indices, $\mathbb{I}$, of the ghost layer(s) *after* the inversion process and *before* reconstructing the values at the cell faces with the slope limiters. A 2nd-order accurate flux scheme requires *two* ghost layers at each internal boundary. It was observed in the current work that *identical* results are obtainable independent of CPU count on a 3-D rectilinear grid. The speedup is significant and the computations show good scalability to parallelization. The present code can run on any grid which is divisible into $\mathcal{N}$ equally sized domain partitions (same number of cells), including dimensions of non-uniformly distributed cells. Figure 16 shows the

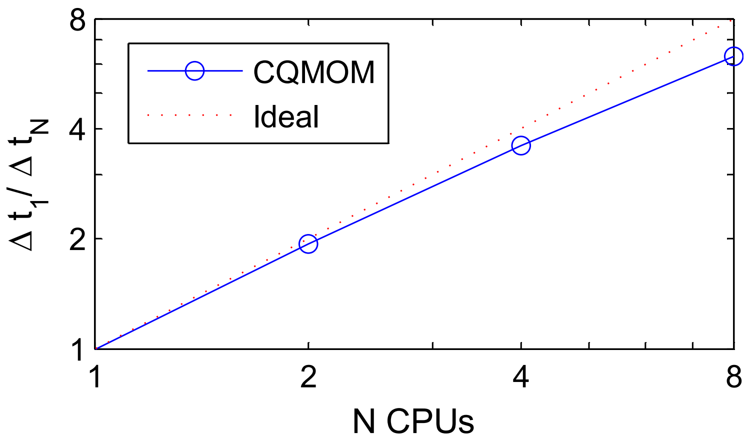


Figure 16. Speedup of CQMOM Algorithm Through CPU Parallelization

speedup obtained for the 3-D channel CQMOM code only, with isotropic collisions enabled and running on $\mathcal{N} = 1, 2, 4$, and 8 CPU's. At most, the full *internal* 3-D grid, $64 \times 64 \times 64$, was divided into $\mathcal{N} = 8$ partitions of $32 \times 32 \times 32$, each having two ghost layers on all 6 boundary faces. This could very easily be divided into more smaller partitions given the availability of better hardware.

Table 13. Comparison of DNS and CQMOM Solver Variables and Equations

Velocity	D	$\mathbb{B}^D D$
Pressure	1	
Weights		$\mathbb{B}^D$
Moments		$\mathbb{B}^D(D+1)$
Transport Eqs.	D	$\mathbb{B}^D(D+1)$
Permutations	1	$(D-1)!^D$

Table 13 compares the number of variables and equations required to solve the DNS carrier fluid vs. the CQMOM particle phase respectively. The costs for CQMOM are multiplied because each permutation must repeat the conditional moment inversion for all D dimensions and perform the flux term in all D dimensions to close the flux term per Equation (52).[44] However, when $\beta < \mathbb{B}$ (for each dimension) the costs are somewhat reduced.

G. Solution Algorithm

In summary, the CQMOM solution algorithm is given in Table 14 and lists the overarching procedure as executed in the current work's numerical source code. Each indentation signifies a nested "for" loop that is repeated as indicated, and the tilde terms ($\widetilde{\boldsymbol{M}}$ and $\widetilde{\mathbb{N}}$) are temporary placeholders with scope inside the path p loop only.

Table 14. CQMOM Solution Algorithm

```
1. Apply initial conditions to abscissas ($\hat{\boldsymbol{v}}, \boldsymbol{\xi}$) and weights ($\phi$) in $\mathbb{N}(t_0, \boldsymbol{x})$
2. Apply all §*3. Boundary Conditions* to $\mathbb{N}$,
3. Initialize the moments over entire domain $(t_0, \boldsymbol{x})$, $\mathbb{N} \rightarrow \boldsymbol{M}$, Equation (52)
4. Main time index loop ($t_n$), for $n$ in 0:$\mathcal{N}$
  A. Initialize $\overset{(r=0)}{\boldsymbol{M}} = \overset{(n)}{\boldsymbol{M}}$
  B. Runge-Kutta iteration loop, for $r$ in $\{1{:}R = 2\}$,
    i. Initialize the running sum, $\overset{(r)}{\boldsymbol{M}} = \boldsymbol{0}$
    ii. Loop through all flux paths, for $p$ in $\{1{:}P\}$, Equation (92)
      a. Initialize the temporary placeholder, $\widetilde{\boldsymbol{M}} = \overset{(r-1)}{\boldsymbol{M}}$
      b. Flux-splitting loop, for dimension $d$ in $[x_{d_1}, x_{d_2}, x_{d_3}]$, Equation (91)
        • Determine sequence vector $\boldsymbol{b}$ from Table 7 for dimension $d$ and path $p$
        • Conditional-splitting loop, for dimension $i$ in $\boldsymbol{b}$, Figure 12
          ◦ Wheeler inversion in dimension $i$, $\widetilde{\boldsymbol{M}} \rightarrow \widetilde{\mathbb{N}}$, Figure 10
        • Apply body force terms to $\widetilde{\mathbb{N}}$ in dimension $d$, Equation (126)
        • Apply §*3. Boundary Conditions* to $\widetilde{\mathbb{N}}$
        • Share ghost layer(s) of $\widetilde{\mathbb{N}}$ between neighboring CPU's (HPC only)
        • Reconstruct $\widetilde{\mathbb{N}}_{a\pm\frac{1}{2}}$ at cell faces in dimension $d$ using §*4. Slope limiter*
        • Compute convective moments, $\boldsymbol{H}^{\pm}_{a\pm\frac{1}{2}}(d, ...)$, Equation (96)
        • Compute FV spatial flux terms, $\boldsymbol{G}_{a\pm\frac{1}{2}}(d, ...)$, Equation (98)
        • Advance flux term, $\widetilde{\boldsymbol{M}}$-= $\frac{\Delta t}{\Delta x_d}(\boldsymbol{G}_{a+\frac{1}{2}} - \boldsymbol{G}_{a-\frac{1}{2}})$, Equation (117)
      c. Add the result to the running sum, $\overset{(r)}{\boldsymbol{M}}$+=$\widetilde{\boldsymbol{M}}$
    iii. Average the running sum by total $P$ path permutations, $\overset{(r)}{\boldsymbol{M}}$/=$P$
    iv. Advance $\overset{(r)}{\boldsymbol{M}}$ by $\Delta t$ due to collision term $\mathbb{C}$, Equation (127)
  C. Finalize RK SSP average, $\overset{(n+1)}{\boldsymbol{M}} = [\overset{(n)}{\boldsymbol{M}} + \overset{(r=R)}{\boldsymbol{M}}]/2$, Equation (117)
5. Save results
```

VI. RESULTS

A variety of simulations were performed in the present work, all of which focused on the multivalued velocity field of particles corresponding to the PDF, $f(t, \boldsymbol{x}; \boldsymbol{v})$. At most, $\beta = 2$ node quadratures are possible in *each* dimension for all cases. The 2-D cases employ the full 3-D solver by ignoring one of the dimensions, which ensures consistency in the software methodology and is mathematically equivalent.

A. Free Jet

To demonstrate the capabilities of CQMOM operating at very large Knudsen numbers, a free jet flow in a vacuum is modeled, i.e. the Stokes number can be considered infinite, with no carrier flow or drag force present. A 40×40 grid covering a domain of length 1×1 is used to demonstrate the effects of numerical diffusion. The `minmod` and `superbee` slope limiters are compared by viewing the first four x moments, $\{M^{(0)}, M^{(1)}_{100}, M^{(2)}_{200}, M^{(3)}_{300}\}$, and the x and y velocity abscissas $\hat{\boldsymbol{v}} = \{U, V\}$ along the diagonal centerline of the jet, seen as the dashed cross-section in Figure 17(a). The 3-D solver is still employed for consistency by ignoring any z moments or z-direction inversions. Free jets in the following cases showed the best results when using $\mathbb{W}$=$\{1E^{-3}, 1E^{-2}\}$. The abscissa proximity ratio was left unrestricted ($\mathbb{P}$=1.0) to test the robustness of the solver.

Figure 17(a-b) depict the volume fraction in a 2-D plane, starting with the inlet boundary condition at the south face with inlet volume fraction $M_0 = 0.05$, nine cells wide, and moving diagonally with velocity $\{U, V\}$ toward the outlet at the east face.

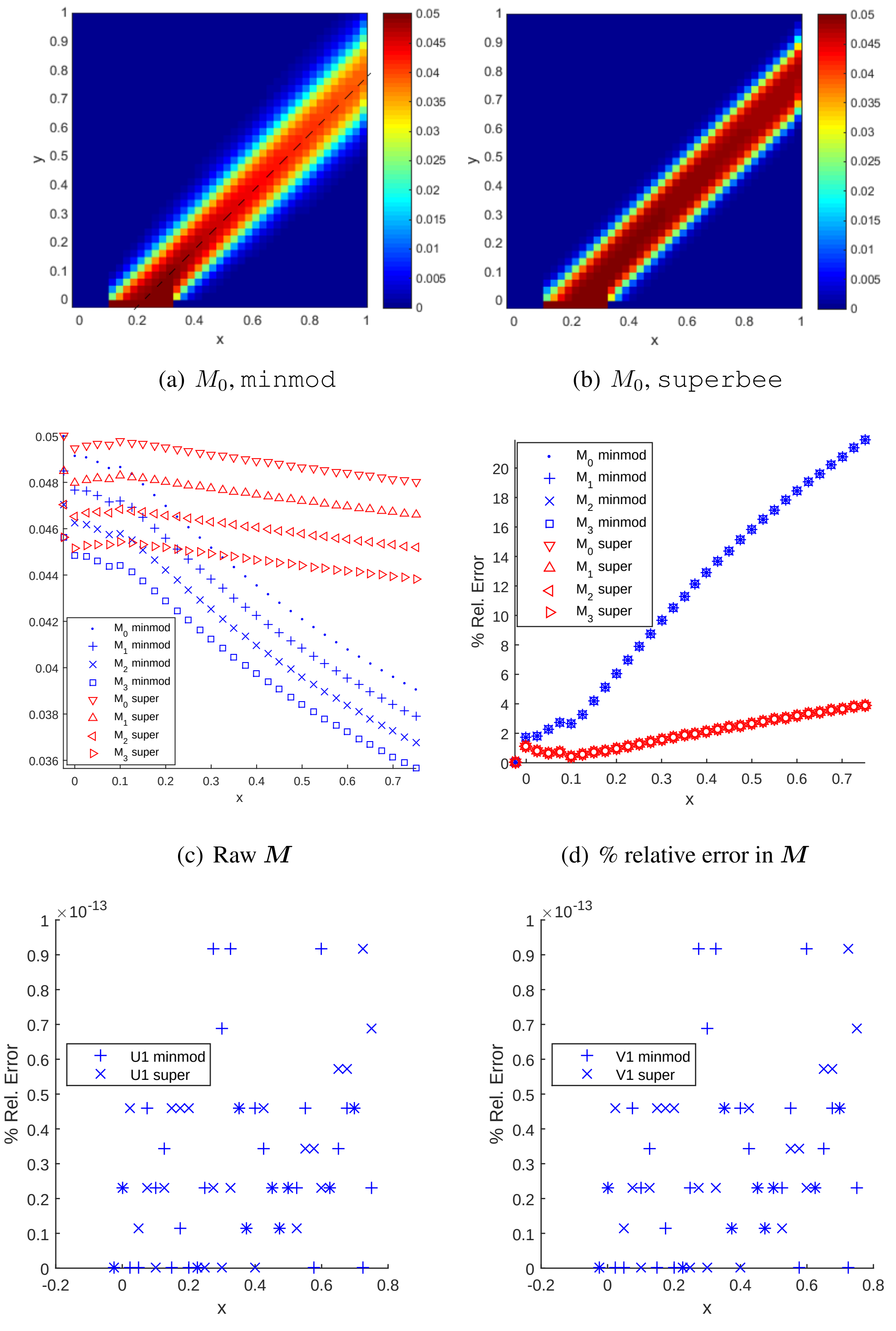
The time step size is chosen to be much smaller than the CFL condition in all the 2-D simulations by the following convention,

$$\Delta t = \frac{\min(\Delta x_{d,a})}{4\mathbb{V}2^D} \tag{129}$$

where $\min(\Delta x_{d,a})$ is the scalar value of the smallest Δx in any dimension among all the cells in the domain. Ideally, the moments and velocities will remain constant from inlet to outlet, therefore, the inlet conditions provide the “true” solution that can be used as a reference point to measure relative errors. Recall that the reconstruction between cells is only first-order accurate when neighboring node sets are different ($\mathbb{I}$), but this had little or no effect on the accuracy of the single jet. This was verified in a case which ignored $\mathbb{I}$ and was fully second-order accurate and showed identical results.

1. Moment Misalignment

Instead of setting the velocity scale to exactly unity ($\mathbb{V} = 1.0$), an inlet velocity of $\{U, V\} = \{0.97, 0.97\}$ is now chosen to highlight the non-linear effects of numerical round-off errors that lead to moment misalignment. Notice from Figure 17(c) that all the moments degrade/diffuse *in concert*, and in fact, the *relative* error of *all* the moments in $\boldsymbol{M}$ degrade identically as shown in Figure 17(d). Therefore, their relationships (such as $U = M^{(1)}/M^{(0)}$) retain extremely high accuracy even when suffering from significant degradation caused by numerical diffusion. This retained *alignment* of moments is a very powerful feature of moment methods, but it is also delicate because discriminatory errors (which affect different moments unequally) lead to *misaligned* moments that



(a) M_0, `minmod`

(b) M_0, `superbee`

(c) Raw $\boldsymbol{M}$

(d) % relative error in $\boldsymbol{M}$

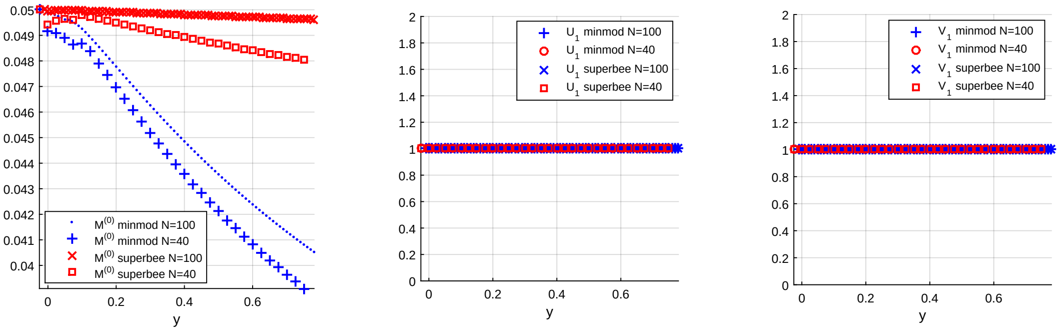
(e) % relative error in U

(f) % relative error in V

Figure 17. Numerical Diffusion of One Free Jet in 2-D Along Jet Centerline in 40×40 domain

create instabilities in the velocities as described further later. When the velocity abscissas only contain small round-off errors that misalign the moments during construction, the relative error between transported moments still remains very small and random (machine precision) as depicted in Figure 17(e-f). The *percent* relative errors are on the order of $1E^{-14}$, (which is $1E^{-16}$ true relative error), and coincides exactly with the double machine precision employed. Furthermore, the errors are identical in both dimensions, owing to the symmetry in the solver.

2. *Grid Sensitivity to Numerical Diffusion*



(a) (b) (c)

Figure 18. Numerical Diffusion of Single Jet at Time Step $t = 1600$. Columns Are Volume Fraction (M_0), x-Velocity (U_α), and y-Velocity (V_α)

Choosing the velocities $\{U, V\} = \{1, 1\}$ exactly ensures that *all* the moments are equal in value, $M^{(0)} = M^{(1)}_{100} = M^{(2)}_{200} = M^{(3)}_{300}$, and any effects of misalignment are absent. In fact, any velocity values which are exactly representable in binary (no round-off, such as 0.5) retain their initial values *exactly*, completely independent of the numerical diffusion errors in $\boldsymbol{M}$. This has been verified for the single jet case and is shown in Figure 18 by comparing two different $\mathcal{N} \times \mathcal{N}$ sized grids. In the absence of moment misalignment, these cases can be used to isolate the grid sensitivity to numerical diffusion only. The jets have inlet widths of 9 and 15 cells for $\mathcal{N} = 40$ and $\mathcal{N} = 100$ respectively. The `superbee` limiter is far superior at preserving sharp discontinuities (shocks) compared to `minmod`. However, even the coarsest mesh with the least accurate slope limiter still produces *exact* results for the velocity throughout the entire domain, $\{U, V\} = \{1.0, 1.0\}$ *exactly*! Since only 1-point quadrature is needed for the single-jet case, only the first node ($\alpha = 1$) is non-zero and the unused nodes (entirely zeros) are hidden for clarity. In summary, the velocities maintained their exact inlet values, regardless of the slope limiter or fineness of the mesh, which was possible due to perfect moment alignment.

B. Free Jet Crossing

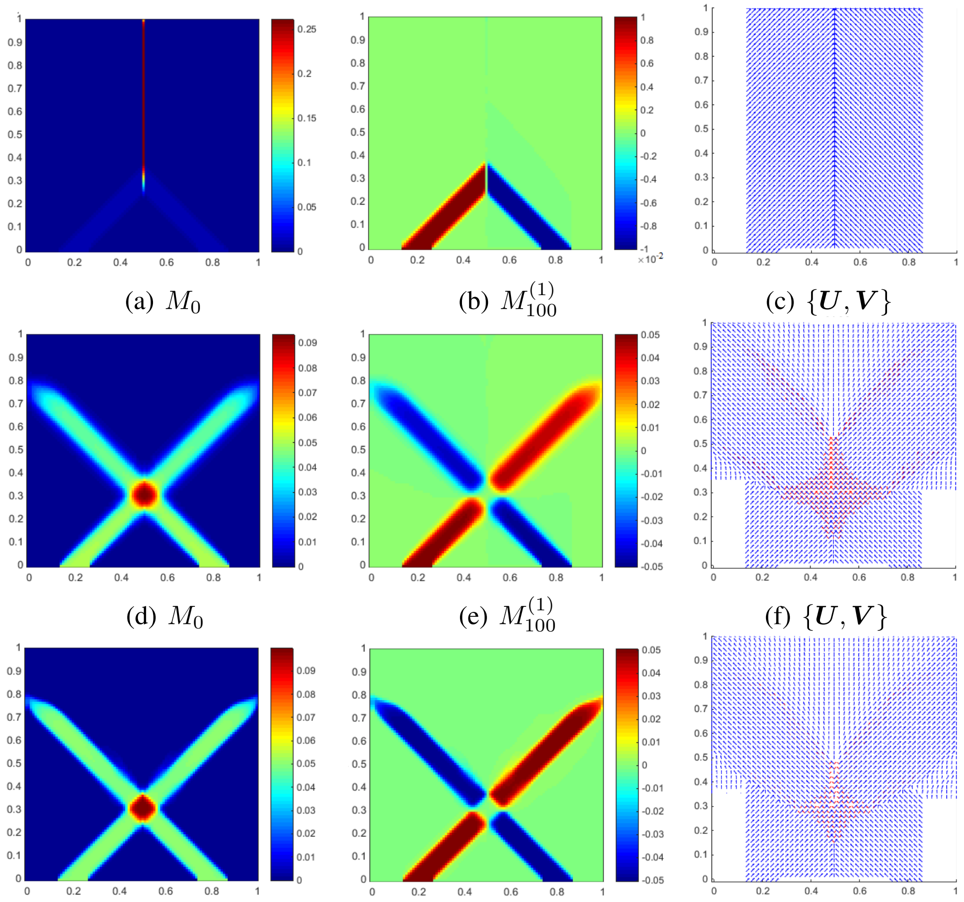
Next, a second jet is introduced into the 2-D planar 100×100 grid covering a domain of length 1×1. An inlet boundary condition of *two* particle jets is located on the south boundary with initial velocity $\hat{\boldsymbol{v}} = \{1, 1\}$ and $\hat{\boldsymbol{v}} = \{-1, 1\}$ respectively so that their trajectories will cross. Just as before, the incoming jets are perfectly collimated and experience no collisions; therefore, the Knudsen number is infinite at the inlets (and remains so for the collisionless case even at the intersection). An inlet volume fraction of $M_0 = 0.05$ is chosen arbitrarily for a collisionless case. The jet-crossing case introduces an intersection region that drastically affects all the moment fields in $\boldsymbol{M}$. At the intersection's very center, the even-ordered x-moments double in size while odd-ordered x-moments cancel perfectly, and the surrounding radius contains skewed mixtures that depend upon which of the two jets is dominant.

First, results for a uniform grid with equal cell-spacings Δx_d in all dimensions, d, are provided. As expected, the bimodal distribution occurring at the intersection of the jets is realizable using 2-point quadrature with CQMOM, allowing for the multivalued velocities to be retained. Perfect symmetry is achievable in the solution when the initial conditions are also symmetric. The vector plots skip every other grid point and may not appear perfectly symmetric, even when they are.

A restrictive example[3] is shown in Figure 19(a-c) which intentionally limits the number of nodes to $\beta = 1$, preventing any PTC or variance ($\sigma^2 = 0$). When $\beta = 1$, then only "mass" and mean momentum are considered, resulting in a non-physical delta shock where the mean x-momentum of zero, $\langle v_1 \rangle = 0$, is given to all the particles of the cell, causing an accumulation of mass in those cells that is transported upward, $\langle v_2 \rangle = 1$. This result is closer to a physical solution that might be expected for a hydrodynamic fluid at Kn $\ll 1$, as demonstrated experimentally by Keshavarz *et al.*,[79]but is not the expected result for a collisionless rarefied fluid, Kn $\to \infty$. Kah *et al.*[12] show a jet crossing of a turbulent aerosol spray exhibiting the same phenomenon.

1. Limiter Comparison

To capture PTC, a 2-point quadrature approach is required at minimum and shown in Figure 19(d-f) with the `minmod` slope limiter. Notice that the x-momentum, $M_{100}^{(1)}$, at the intersection decreases to zero because the two jets cancel one another, while the energy,



(a) M_0 (b) $M_{100}^{(1)}$ (c) $\{\boldsymbol{U}, \boldsymbol{V}\}$

(d) M_0 (e) $M_{100}^{(1)}$ (f) $\{\boldsymbol{U}, \boldsymbol{V}\}$

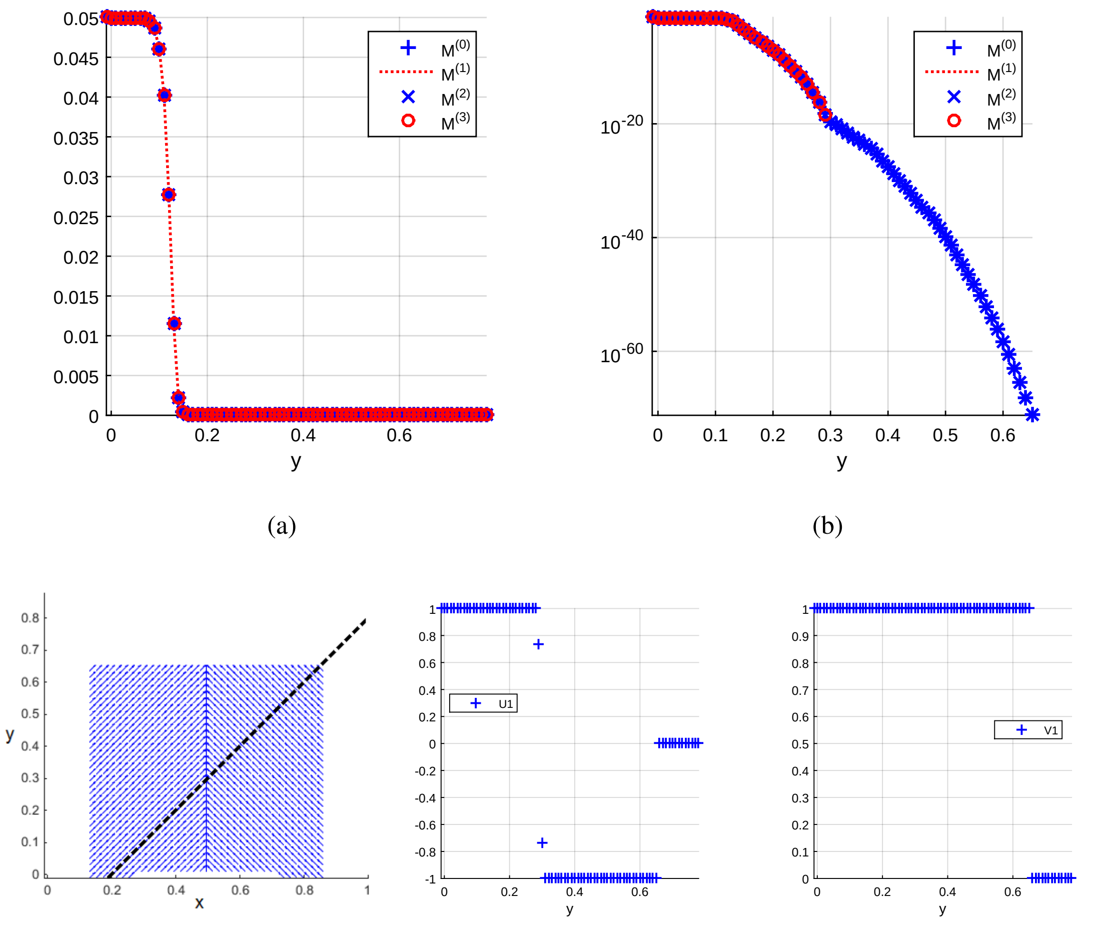
(g) M_0 (h) $M_{100}^{(1)}$ (i) $\{\boldsymbol{U}, \boldsymbol{V}\}$

Figure 19. ***x*-*y* Planar Contours of Collisionless Free Jet Crossing on Uniform Grid. Slope Limiters: `minmod` (d-f); Otherwise `superbee`**

$M_{200}^{(2)}$, doubles in magnitude. Numerical diffusion is clearly noticeable farther from the inlet, amplified by the 2-D diagonal motion of the jet. The inlet concentration, $M^{(0)} = 0.05$, uses a step function to test the shock-preserving capabilities. The mass, momentum, and higher order moments all exhibit similar effects of numerical diffusion.

In an attempt to reduce the numerical diffusion, the `superbee` slope limiter is also applied, shown in Figure 19(g-i). This improved limiter is capable of preserving sharper shocks, and the coefficient $\varepsilon = 1.5$ was employed (see section §*4. Slope limiter*).

Notice also that fewer 2-point quadratures are required when comparing Figure 19(f) and Figure 19(i) because less mass has been expelled by numerical diffusion away from the jets, i.e. the vector field has a narrower more accurate representation of the jets. The peak mass, $M^{(0)}$, at the outlet boundary suffered a 22.5% loss with the `minmod` limiter and exhibits noticeable spreading; however, the `superbee` limiter only suffered a 3.5% loss and a qualitatively sharper edge. The results in Figure 19 confirm the previous results of others[42,44] and also add a new improved shock treatment.



(a)

(b)

(c)

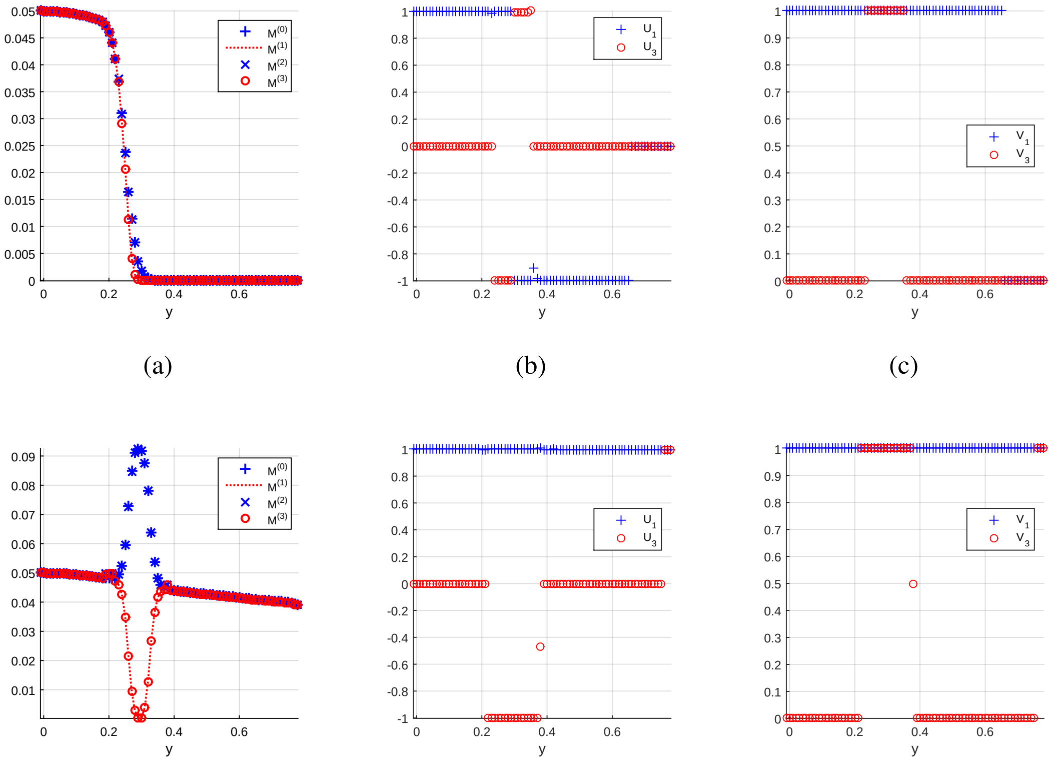
(d)

(e)

Figure 20. Early Evolution of the `superbee` Limiter at Time Step $t = 200$

The volume fraction along the diagonal centerline of the right-moving jet in Figure 20(a) shows the early ($t = 200$) forefront of the large “mass” shock ($y \approx 0.13$) before it reaches the intersection. Notice in Figure 20(d-e) however, that the velocity abscissas are already determined far in advance, up to $y \approx 0.65$, far beyond the forefront of the large “mass” shock. In actuality the real shock is far ahead of the apparent “mass” of particles, and Figure 20(b) indicates this by showing an equivalent log plot of Figure 20(a). An extremely small “mass” of particles, on the order of $M_0 \approx 1E^{-72}$, is being convected far ahead of the jets, yet still perfectly retains the exact velocity abscissas. This “colonization” of the domain lays the foundation for the abscissas far before any significant amount of “mass” occupies it. However, since the “mass” is well below the weight threshold ($M_0 \ll \mathbb{W}_1$), only a simple 1-point quadrature inversion is performed in those cells, e.g. $U_1 = M^{(1)}_{100}/M^{(0)}$, and only the lowest two moments are required. The slope discontinuity at $y \approx 0.3$ in Figure 20(b) is the shift as the right-moving jet relinquishes dominance to the left-moving jet as shown in Figure 20(c) near x=0.5. Therefore, the log plot does not show $M^{(1)}_{100}$ or $M^{(3)}_{300}$ beyond $y \approx 0.3$ because they become negative where the left-moving jet is dominant, but are exactly the same magnitude as M_0 and $M^{(2)}_{200}$ and retain the exact velocity abscissas, $V = -1.0$. Beyond the forefront of the shock ($y > 0.65$) the nodes still retain their untouched zero-valued initial conditions, which also do not appear on a log scale plot.

Figure 21 depicts the further evolution of the right-moving jet at two different time steps ($t = 400$ and 1400), once right at the initiation of the jet crossing and once after the jet has crossed completely and exited the domain at the east outlet. The onset of the jet



(a)

(b)

(c)

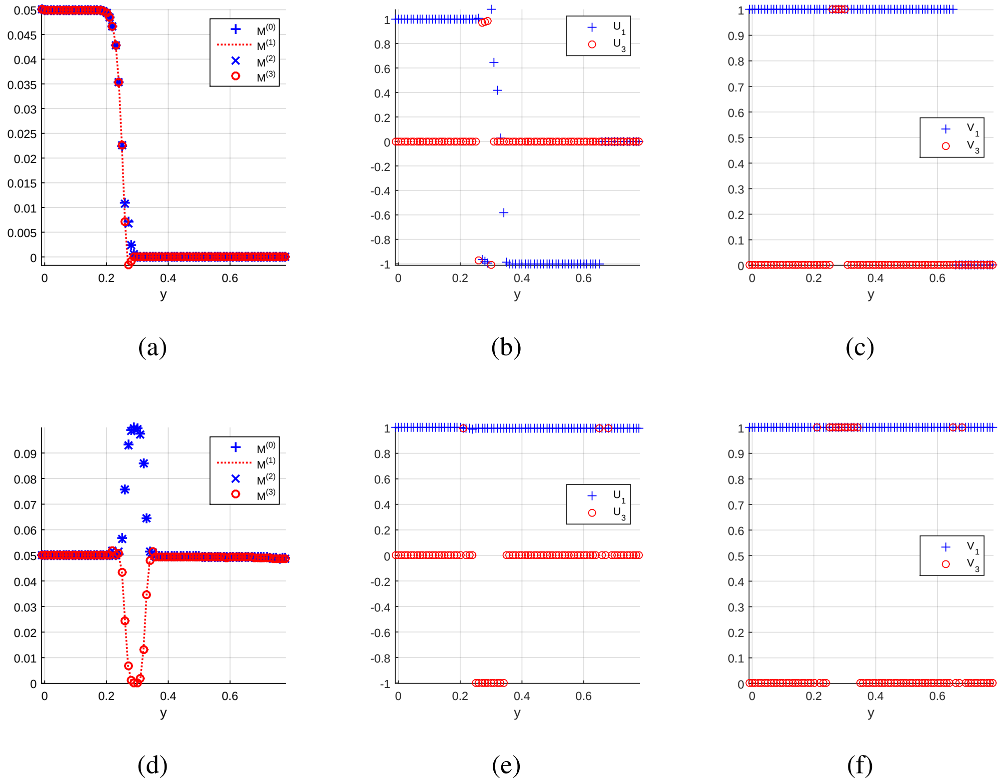
(d)

(e)

(f)

Figure 21. Numerical Diffusion From `minmod` Limiter at Time Steps $t = 400$ (First Row) and $t = 1400$ (Second Row). Columns Are Moments (M), x-Velocity (U_α), and y-Velocity (V_α) Respectively

intersection occurs in Figure 21(a) and Figure 22(a) near $y \approx 0.25$, where the odd ordered moments interfere destructively and the even ordered moments interfere constructively, causing the two pairs of moments to separate in value, and the odd ordered moments may become negative. This significant difference between moments immediately causes the inversion to return a 2-point quadrature of the PDF. Therefore the values of U_1 and U_3 are nonzero (the zeros of the other unused nodes, U_α, are hidden for clarity). The linearized subscript index shown here is that of Equation (88).



(a)

(b)

(c)

(d)

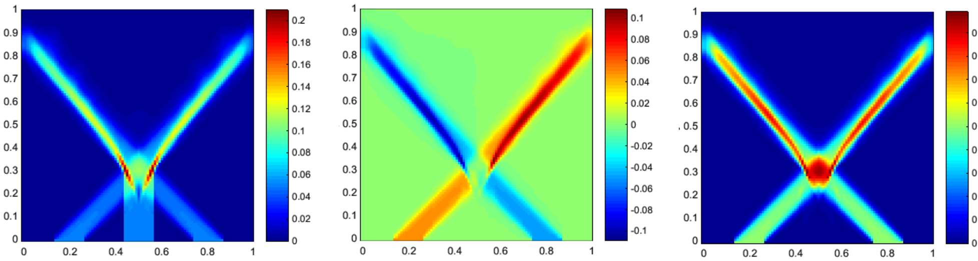
(e)

(f)

Figure 22. Numerical Diffusion From `superbee` Limiter at Time Steps $t = 400$ (First Row) and $t = 1400$ (Second Row). Columns Are Moments (M), x-Velocity (U_α), and y-Velocity (V_α) Respectively

2. *Under-Representation Errors*

Notice that Figure 20(d), Figure 21(b,e,f), and Figure 22(b) contain rare occurrences of outlier velocity abscissas which are clearly *not* -1, 0, or 1. These are erroneous values occur in cells which experience PTC, but whose weights are too small to be given a 2-point quadrature, thus a 1-point quadrature inherits the averaged velocity. Unfortunately, cells lying at the interface between regions of 1-point and 2-point quadratures will commonly suffer from this phenomenon (and may benefit from special treatment or consideration to mitigate these effects in future work). This 1-to-2-point interface is unavoidable if a 1-point quadrature simplification is to be used in regions with very low weights. The $\mathbb{W}_1$ threshold determines where the cutoff is made and where the interface will occur. It should be chosen small enough that the errors at the interface do not readily affect the solution. The oversimplification of a PDF into 1-point quadrature may cause the single abscissa to acquire strange values. For example, if the $M^{(1)}_{100}$ value decreases because it has experienced flux cancellation from the neighboring cells, but $M^{(0)}$ grows in value, then the moments become misaligned for a 1-point representation and the resulting inverted single velocity $U_1 = M^{(1)}_{100}/M^{(0)}$ acquires the cell-average value, which is different in magnitude than either of the 2-point quadrature velocities would be, and explains the observed outlier abscissas.



(a) M_0

(b) $M^{(1)}_{100}$

(c) $M^{(2)}_{200}$

Figure 23. x-y Planar Contours of Collisionless Triple Jet Crossing on Uniform Grid With `minmod`

Lastly, in an attempt to illustrate the limitations of a 2-point quadrature, three jets with unique x-velocities, (v_1 is 1, 0, and -1 respectively), are directed to intersect simultaneously with y-velocity $v_2 = 1$. In the intersection region, the three distinct velocities will be forcibly reduced into two ($\beta = 2$) distinct velocities that conserve mass, momentum, and energy. It is reasonable to expect that only two distinct jets will emerge from the intersection with decreased x-velocity ($|\hat{v}_1| < 1$). Obviously, this is not the correct physical solution, and it resembles the results of Figure 19(a-c) that is restricted based on an inadequate β-point limitation. Employing $\beta = 3$ nodes should results in the proper expected solution because the PDF at the intersection will be fully resolved (consisting of three delta functions).

3. Crossing Errors

Figure 24 shows a direct comparison of the numerical diffusion of M_0 taken from Figure 18(a), Figure 21(d), and Figure 22(d), indicating that the jet loses some cohesion after passing through the intersection as compared to the single jet case which did not pass through an intersection. The moments degrade noticeably more when passing through the jet intersection because it introduces large slopes and an irregular convex

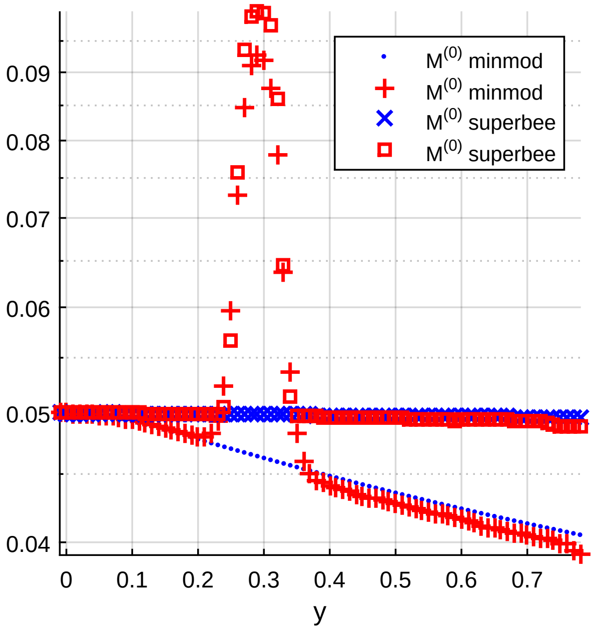
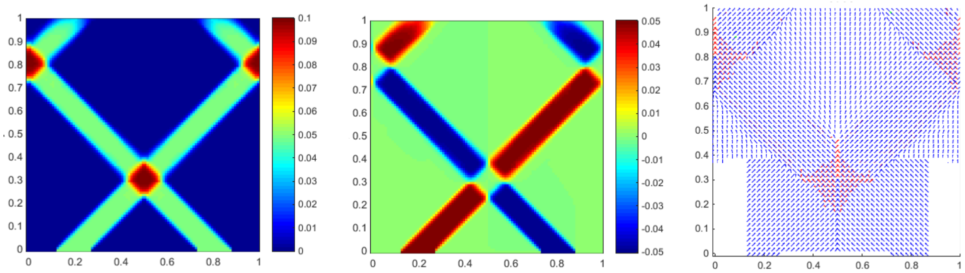


Figure 24. Comparison of M_0 Between Single Jet and Dual Crossing Jets for Both Slope Limiters (Log Scale)

bulge in all the moment fields. Additionally, small errors occur at the 1-to-2-point quadrature interfaces due to under-representation as described previously, and these are present in the jet-crossing case. Consequently, the jet is never perfectly recovered after passing through the intersection and the M_0 computed using `minmod` is only 97.9% as large after the dual crossing as compared to the single jet. For `superbee`, 99.5% of M_0 is recovered. However, the velocity abscissas are retained with *extremely* small error because all the moments in $\boldsymbol{M}$ were on the same order and the curved shock was similar in shape such that moment alignment was maintained extremely well. The worst velocity abscissa errors in Figure 21(e) and Figure 22(e) were nearly identical at 0.154% and 0.159% respectively, ignoring velocity outliers which appear rarely and are usually insignificant in weight ($M_0 \ < \ \mathbb{W}_1$). However, if the flux error in $M^{(3)}$ or $M^{(2)}$ were much different than the error in $M^{(1)}$ or $M^{(0)}$ (hypothetically speaking) then the flux scheme would selectively introduce more error to certain moments and lead to moment misalignment. The `superbee` limiter may be marginally more likely to introduce moment misalignment because it allows for larger slopes (steeper shocks) than the conservative `minmod` method (see Section §*6. Volume-averaging error*), but `superbee` also provides drastically reduced numerical diffusion of the moments.

4. Boundary Conditions

Next, the boundary conditions were tested by imposing a “reflective wall” treatment at the west and east x borders of the domain. As can be seen in Figure 25(a-c), the free jets are reflected back into the domain, having their x-velocities perfectly reversed at



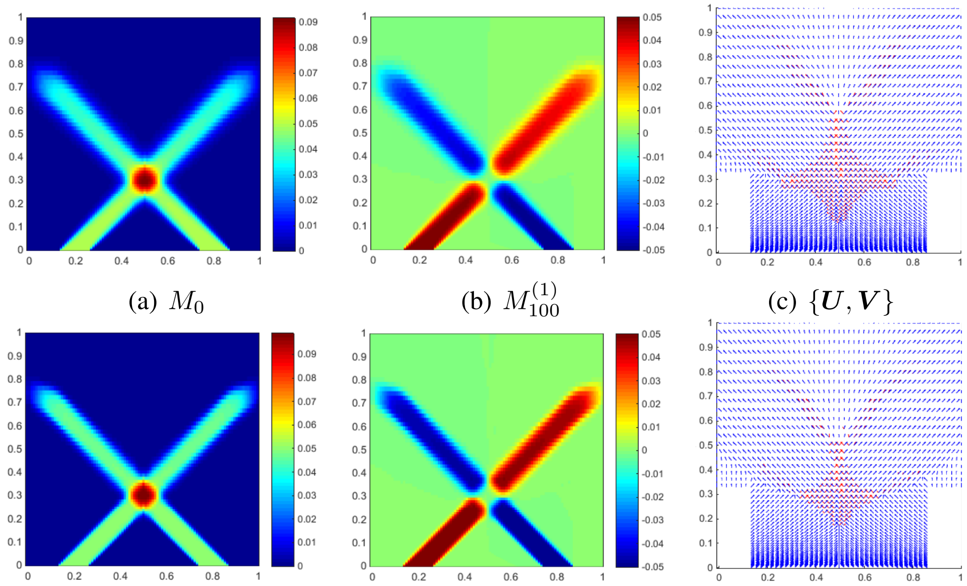
(a) M_0 (b) $M_{100}^{(1)}$ (c) $\{\boldsymbol{U}, \boldsymbol{V}\}$

Figure 25. Symmetrical x-y Planar Contours of Collisionless Free Jet Crossing with Reflective/Periodic x-Sidewalls and `superbee` Slope Limiter

the sidewalls, since $e_w = 1$. This creates two more regions of strong PTC before both jets exit at the north (Neumann) boundary. Interestingly, when replacing the reflective walls with periodic boundary conditions, the *exact* same indistinguishable results are observed as in Figure 25(a-c)! This is because the wall boundary effectively mirrors the internal boundary layer onto the ghost layer, which is the same (for a symmetrical case) as copying the values from one end of the domain to the ghost layer at the other end. In other words, the reflective wall ghost layer acts as an "inbound" mirror image of whatever is "outgoing" from the domain at the boundary (when $e_w = 1$).

5. *Non-Uniform Grid*

To further test the versatility of the FV methods, a non-uniform grid stretched with a hyperbolic tangent skew ($\varepsilon_g = 2$) in the y-direction only, was employed; see Section §*5. Non-Uniform Grid*. The results in Figure 26 show that the moment transport through the non-uniform cells performs as expected. Numerical diffusion is increased compared to the uniform grid in Figure 19 because the cells are larger and the diagonal jet motion is not aligned with the staggering of the grid. The first case employed the `minmod` limiter,



(a) M_0 (b) $M_{100}^{(1)}$ (c) $\{\boldsymbol{U}, \boldsymbol{V}\}$

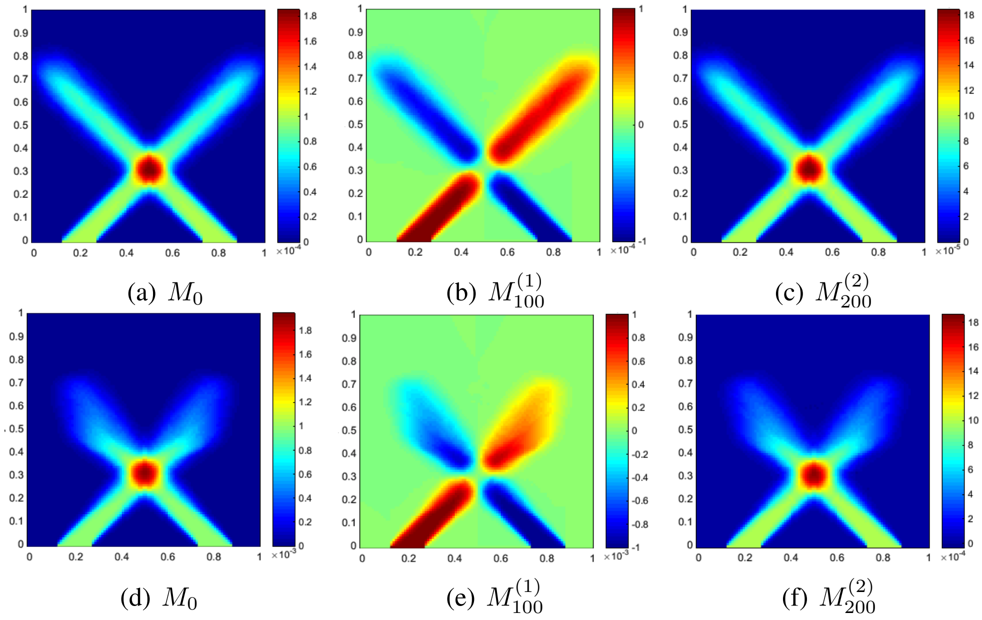
(d) M_0 (e) $M_{100}^{(1)}$ (f) $\{\boldsymbol{U}, \boldsymbol{V}\}$

Figure 26. x-y **Planar Contours of Collisionless Free Jet Crossing on Non-Uniform Grid:** `minmod` **(First Row) vs.** `superbee` **(Second Row)**

Figure 26(a-c), and the second case the `superbee` limiter, Figure 26(d-f), still showing improved shock preservation. This was a critical test before implementing CQMOM in boundary layer and channel flows that both feature a similar non-uniform grid in the wall-normal direction. Note that the vector plots skip every other cell to improve visual clarity of the arrows. A worthwhile goal for future work will be to implement a CQMOM flux scheme for an unstructured grid.

6. Collisions

One of CQMOM's advantages is its powerful stochastic modeling collision term, which operates over an extremely wide range of length scales, time scales, and arbitrary Knudsen numbers. The collision term is implemented for the crossing of two free jets in the current work. This is a very powerful test of the method because Kn $=\infty$ at the inlets and Kn $\approx O(1)$ at the jet intersection for sufficiently large particle volume fractions because it experiences a large temperature, $\Theta > 0$. The particle diameter is $L_p = 3.5714E^{-3}$ for all the cases and two different perfectly *elastic* collision models have been tested: isotropic and anisotropic.

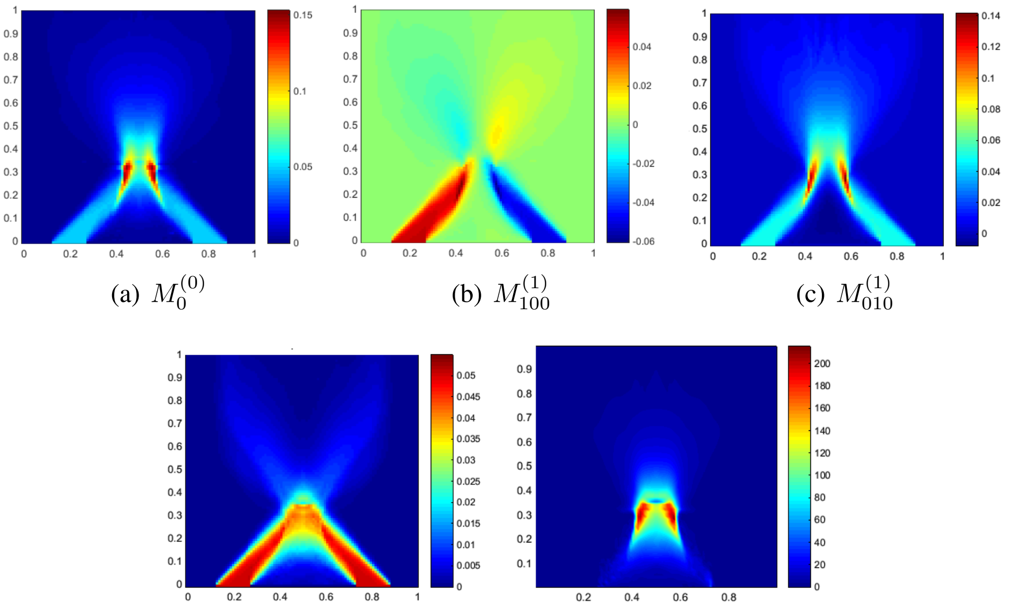


(a) M_0 (b) $M_{100}^{(1)}$ (c) $M_{200}^{(2)}$

(d) M_0 (e) $M_{100}^{(1)}$ (f) $M_{200}^{(2)}$

Figure 27. x-y Planar Contours of Free Jet Crossing With Isotropic Collisions

First, the effects of *isotropic* collisions are considered and shown in Figures 27 and 28. To moderate the collision strength at the jet intersection, the particle volume fraction, $M^{(0)}$, is adjusted at the inlets and particle diameter is held constant. The two rows of Figure 27 represent inlet volume fractions of $1E^{-4}$ and $1E^{-3}$ respectively, while Figure 28 uses $5E^{-2}$. Larger particle volume fractions result in a smaller Kn and more frequent isotropic collisions as seen by comparing Figure 27 and 28; the higher volume fractions have more prominent collision effects.

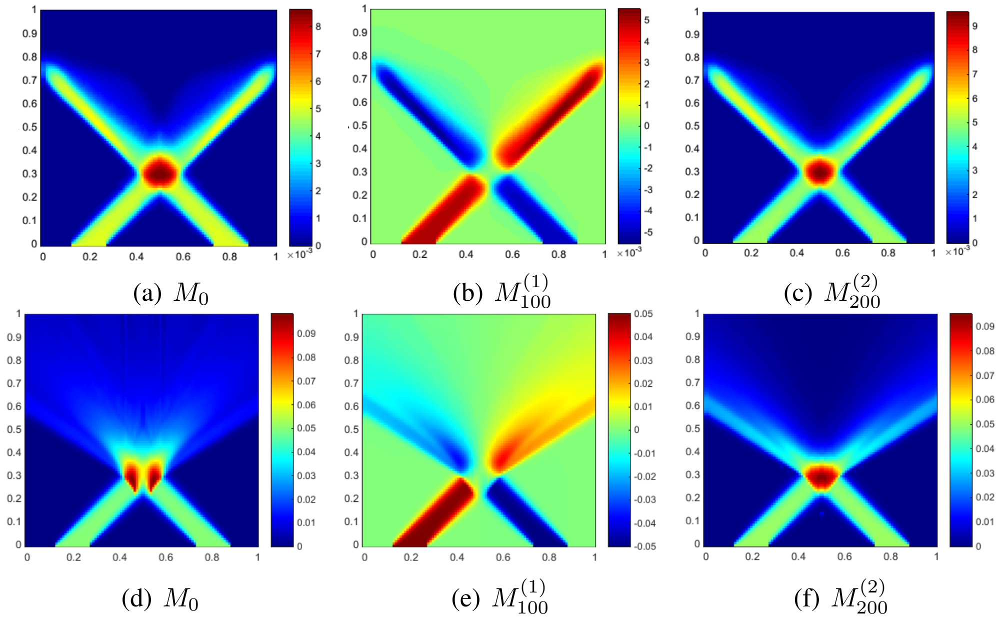


(a) $M_0^{(0)}$ (b) $M_{100}^{(1)}$ (c) $M_{010}^{(1)}$

(d) $M_{200}^{(2)}$ (e) $1/\tau_c$

Figure 28. x-y Planar Contours of Free Jet Crossing With Dense Isotropic Collisions

A noticeable effect of the elastic isotropic collision model is that collisions occur in *all* dimensions, even those which initially have very little temperature. For example, in the 2-D jet case, both jets share the same y velocity, therefore $\Theta_2 = 0$, while collisions are very prominent in the x-direction, $\Theta_1 \gg 0$, at the jet intersection. Consequently, the average of these produces the isotropic temperature, Θ_{eq}, which induces collisions in the y-direction while decreasing the collision strength in the x-direction, effectively transferring energy from the x to the y dimension. Figure 28(d) demonstrates the loss of x-energy at the jet intersection. This is a realistic result because particles often exchange energy between dimensions after an “off-center” collision.[4,7] Even for the 2-jet case in a perfect vacuum with perfectly homogenous collimated jets in which every particle has identical velocity, the isotropic assumption is still appropriate. As a consequence, backscattering will occur at the jet intersection during isotropic collisions. Figure 28 shows prominent backscattering and a wide dispersion of particles in every direction; an astounding result for the limited 2-point quadrature. The collision frequency, $1/\tau_c$, is shown in Figure 28(e) and it demonstrates the wide range of Knudsen numbers present in the domain, where larger values indicate stronger collisions that depend on both temperature and volume fraction. Experimental comparisons of high Knudsen number dual underexpanded free jets are shown by Fujimoto *et al.*[80] for various orientations, however, the jets in the lab experiment are not collimated.



(a) M_0 (b) $M^{(1)}_{100}$ (c) $M^{(2)}_{200}$

(d) M_0 (e) $M^{(1)}_{100}$ (f) $M^{(2)}_{200}$

Figure 29. x-y **Planar Contours of Free Jet Crossing With Anisotropic Collisions**

The anisotropic model will have a less pronounced effect of backscattering; therefore, the *isotropic* model may actually provide better results for comparisons with real experiments because backscattering is an expected stochastic non-homogenous physical behavior. Figure 29 shows *anisotropic* results, with inlet volume fractions of $5E^{-3}$ and $5E^{-2}$ in the two rows respectively. The numerical stability was better for the anisotropic collision model compared to the isotropic model.

C. Taylor-Green Flow

The Taylor-Green (T-G) flow was an excellent case for testing symmetry, and directly led to the development and implementation of the improved stepping method (see Section §*2. Dimension Stepping*). This T-G flow uses a carrier fluid with a time-*independent* velocity field to drive the particles by drag force only. This zero-divergence ($\nabla \cdot u = 0$) 2-D velocity field is shown in Figure 30 and defined by,

$$\begin{aligned} u_1(x,y) &= -\mathbb{U}\cos(2\pi\frac{x}{L_x})\sin(2\pi\frac{y}{L_y}) \\ u_2(x,y) &= \mathbb{U}\sin(2\pi\frac{x}{L_x})\cos(2\pi\frac{y}{L_y}) \end{aligned} \tag{130}$$

where the domain lengths $\{L_x, L_y\} = \{1, 1\}$ in the present work. Just as in the free jet cases, a 100×100 grid is employed for a $\beta = 2$-point quadrature, and the boundaries are all periodic. The initial condition of particle concentration consists of a symmetrical multidimensional Gaussian distribution placed at the center of the central vortex (at the exact center of the domain) with peak volume fraction $M^{(0)} = 5E^{-2}$. Each vortex acts as a centrifuge that forces particles away, and in-between the vortices are stagnation points where particles may accumulate and/or remain dormant.

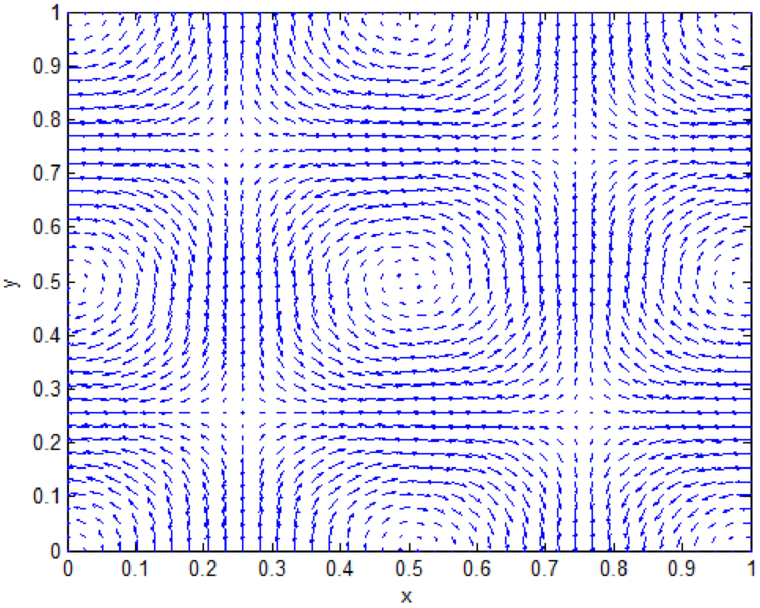
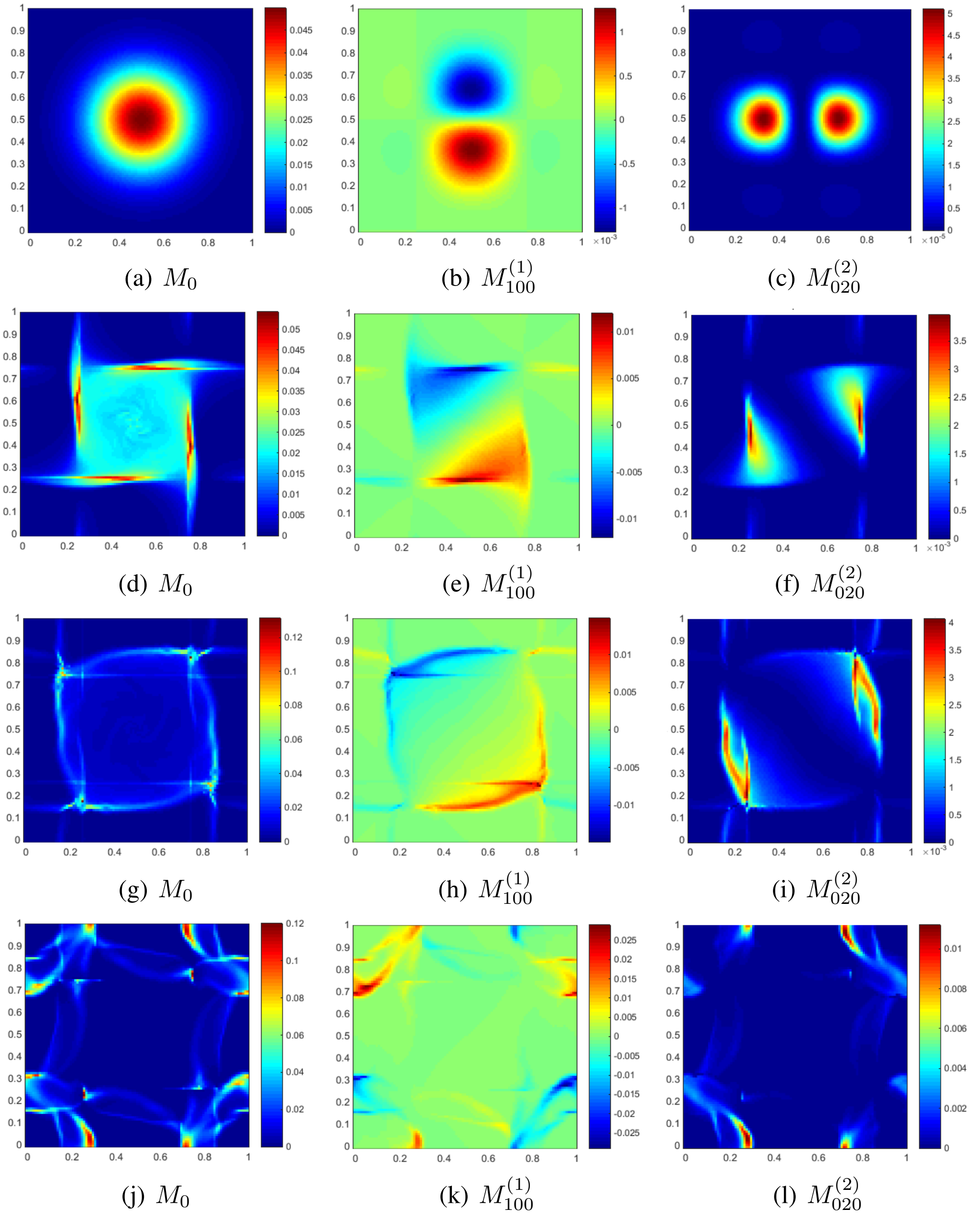


Figure 30. Taylor-Green Flow Field Vectors Spanning $\{L_x, L_y\} = \{1, 1\}$, Equation (130)

1. Collisionless

The collisionless T-G case is expected to quickly exhaust the 2-point quadrature because the particle velocities will be very diverse at the local scale where a bimodal representation of the PDF may be inadequate. However, the method's robustness and ability to maintain symmetry are revealed. Figure 31 lists four different rows corresponding to four different time steps (n =1k, 2k, 3k, 5k respectively), and each column represents the mass, x-momentum, and y-energy respectively. The thresholds and parameters were St=2.5, $\mathbb{P}$=1.2, $\mathbb{W}_1$=$1E^{-4}$, $\mathbb{W}_2$=$8E^{-3}$. The main mass of particles in Figure 31(a-c) is initially centrifuged outward, while particles lying along the square edges connecting the stagnation points (around the center vortex), form "strips" of particles that begin moving parallel to the square edges; Figure 31(d-f). The 2-point



(a) M_0 (b) $M_{100}^{(1)}$ (c) $M_{020}^{(2)}$

(d) M_0 (e) $M_{100}^{(1)}$ (f) $M_{020}^{(2)}$

(g) M_0 (h) $M_{100}^{(1)}$ (i) $M_{020}^{(2)}$

(j) M_0 (k) $M_{100}^{(1)}$ (l) $M_{020}^{(2)}$

Figure 31. x-y **Planar Contours of Collisionless T-G Flow. Each Row Represents** n **=1k,2k,3k,5k Time Steps Respectively**

quadrature realizes this and the wave-front of centrifuged outward moving particles passes through, leaving behind masses of “strips” and stagnation points; Figure 31(g-i). The wavefront continues outward until it reaches the periodic boundaries and begins the journey inward back toward the center, at which point it begins to overlap with itself near the corners of the domains, Figure 31(j-l).

D. TBL Flow Simulation

A DNS fluid solver models the incompressible gas-phase carrier flow, and the dispersed particle phase is being modeled in two different ways: Lagrangian (DPS) and an Eulerian (CQMOM) treatment. A Reynolds number of $\mathrm{Re}_\tau = 2800$ is chosen based on the boundary layer non-dimensional thickness. The full 3-D mesh contains $800 \times 128 \times 98$ structured cells with overall non-dimensional lengths of, $L_1 \times L_2 \times L_3 = 80 \times 6 \times 4$ and is shown in Figure 32. For comparative purposes, the two particle phases run concurrently and do not interact with one another, and both particle treatments are one-way coupled to the same fluid field. The Stokes number (*St*), initial homogenous volume fraction, $M^{(0)}$, and inter-collision states (enabled/disabled) are identical between the Eulerian and Lagrangian particle simulations to enable appropriate side-by-side comparisons. For the initial condition, 32 million ($\mathcal{N} = 32E^6$) monodisperse DPS particles are seeded randomly throughout the domain with particle diameter $L_p = 3.57E^{-3}$. This allows for approximately three DPS particles per cell, but the mesh is non-uniform in the wall-normal direction so that ratio may vary widely. Conversely, the CQMOM field receives a homogenous particle volume fraction in all cells matching the average DPS concentration given by $M_0 = \frac{\pi L_p^3 \mathcal{N}}{6 L_1 L_2 L_3} = 3.94E^{-4}$, ensuring that the collision model provides the proper influence of collisions for comparisons between DPS and CQMOM.

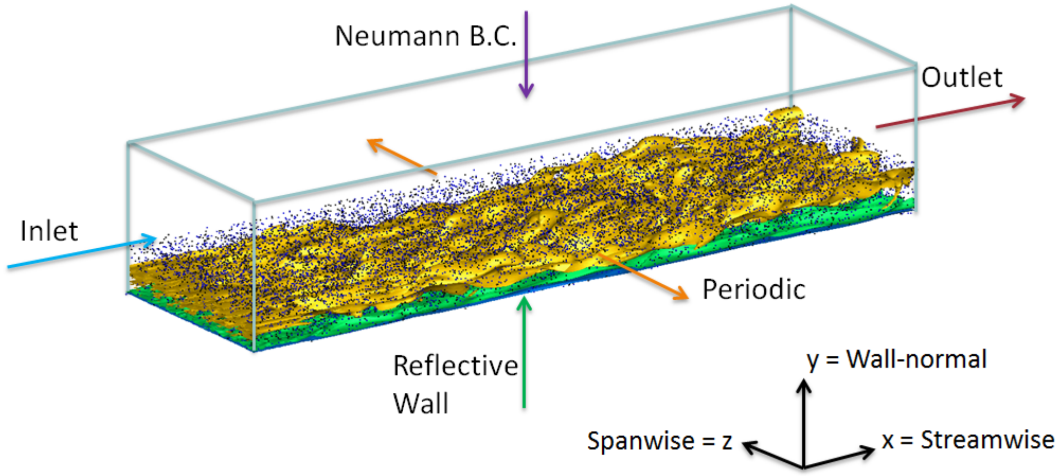
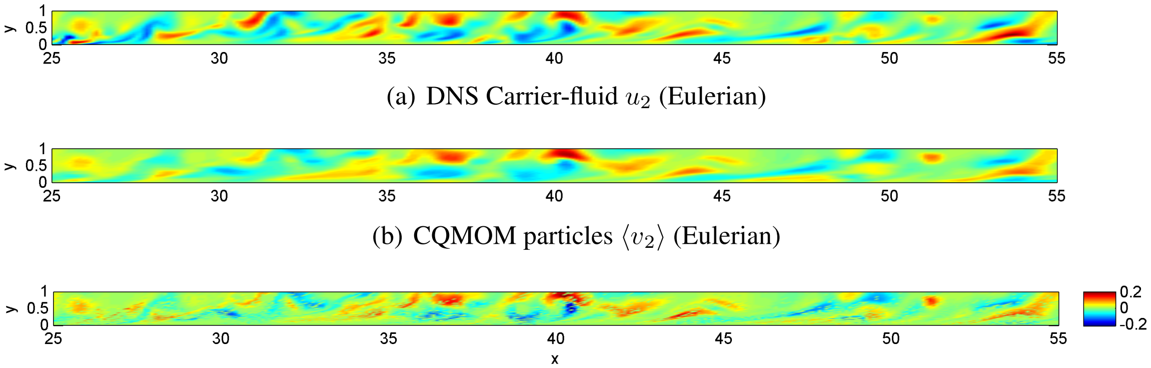


Figure 32. TBL domain and boundary conditions

The two Eulerian phases exploit the same 3-D structured Cartesian boundary layer mesh, oriented in the streamwise x-direction, wall-normal y-direction, and spanwise z-direction. The initial condition volume fraction for the Eulerian particles is maintained constant at the $x = 0$ flow inlet. All methods employ periodic spanwise boundaries, and the upper y boundary employs a Neumann boundary condition for the two Eulerian phases. The lower $y = 0$ boundary acts as an elastic reflective wall for both particle phases, and enforces a no-slip condition for the carrier-fluid phase.

Two separate cases are considered - the first uses “lighter” particles ($\mathrm{St} = 4$) while neglecting inter-particle collisions. Figure 33 shows an instantaneous snapshot over about 25% of the extent of the streamwise domain for the carrier fluid and both models of the particle phase, at the same instant in time. The shear from the no-slip condition on the wall creates complex vortical structures in the boundary layer. The figure shows that the



(a) DNS Carrier-fluid u_2 (Eulerian)

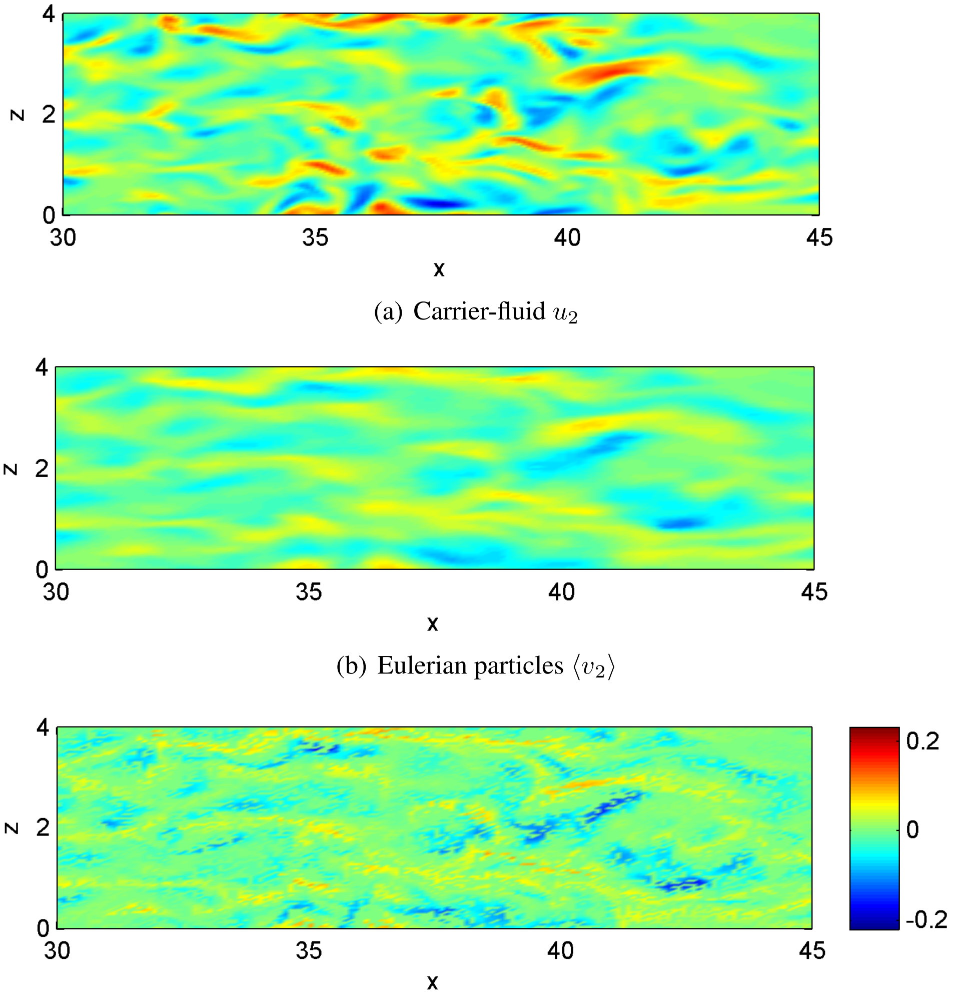
(b) CQMOM particles $\langle v_2 \rangle$ (Eulerian)

(c) DPS particles $\langle v_2 \rangle$ (Lagrangian)

Figure 33. Instantaneous Wall-Normal Velocity at Half-Span, St=4, Without Inter-Particle Collisions.

lighter particles depicted follow the fluid flow field fairly closely with the largest velocity discrepancies occurring in regions of rapid change (large velocity gradients) or small localized perturbations, owing to the inertia-induced particle lag. Importantly, the figure also shows good, qualitative agreement between the Lagrangian and Eulerian models of the particle motion, in turn providing a useful validation of the Eulerian-based approach for predicting dispersed phase properties under one-way coupled fluid drag.

Both the Lagrangian and Eulerian particle treatments exhibit large regions of correlated motion that is driven by the interaction of the particles with the motion of the fluid flow. The wall-normal particle velocity in Figure 34(b-c) are about half as large as those in the fluid shown in Figure 34(a), though they all exhibit very similar structure owing to the fairly responsive particles. An important validation of CQMOM is realized as the two particle models are in good qualitative agreement with one another, more so than they are in agreement with the fluid field.



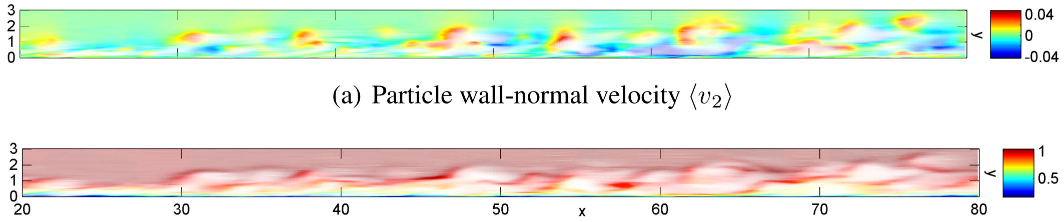
(a) Carrier-fluid u_2

(b) Eulerian particles $\langle v_2 \rangle$

(c) Lagrangian particles $\langle v_2 \rangle$

Figure 34. Instantaneous wall-normal cell-averaged velocity in a wall-parallel plane at y $\approx$ 0.31, St=4, without inter-particle collisions.

The second case employs “heavier” larger-inertia particles, St=16, that result in greater PTC, and an inter-particle collision model is enabled. As will be observed for these larger inertia particles, the fluid and particle velocities can deviate more significantly than for the smaller Stokes number case, and the particle field is less susceptible to responding to small scale structure perturbations of the fluid field.



(a) Particle wall-normal velocity $\langle v_2 \rangle$

(b) Particle streamwise velocity $\langle v_1 \rangle$

Figure 35. CQMOM (Eulerian) particle instantaneous cell-averaged velocity at half-span, St=16, with inter-particle collisions. Opacity indicates density M

Instantaneous snapshots are available for selected 2-D planes within the computational domain. Figure 35 shows nearly the entire streamwise domain ($20 \leq x \leq 80$) with an opacity mask indicating the volume fraction $M^{(0)}$. Note that the mask is identical for both (a) and (b) in the figure. Contour colors which are more transparent (white) indicate regions of lower particle volume fractions, $M^{(0)} \rightarrow 0$. The dispersion and accumulation of particles through interaction with the turbulent carrier flow quickly leads to sparse regions with $M^{(0)} \approx 0$ and higher concentration regions where $M^{(0)}$ achieves at least three times its initial condition value. A strong correlation between concentration and wall-normal velocity is visible in Figure 35(a) indicating that particle uplift disperses the particles leaving behind low concentration structures. Additionally, regions where velocities become superimposed (e.g. both nodes are being exhausted and PTC exists) will often exhibit higher concentrations because the interpenetrating particles are compounded, similar to what is also seen in the test case Figure 19(d) at the jet intersection.

The DPS instantaneous visualizations have difficulty describing the overarching structure when cells are devoid of any DPS particles, in which case zero velocity is

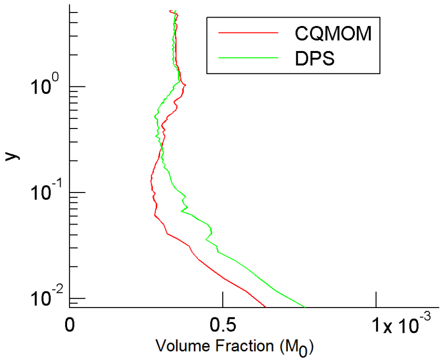


Figure 36. Spanwise-Averaged and Time-Averaged Particle Volume Fraction $M^{(0)}$ for St=16 Without Inter-Particle Collisions at $x \approx 24$.

assumed for that cell. However, cells which are devoid of Lagrangian particles do not contribute to the velocity samples to avoid artificially lowering the Lagrangian statistics, where an undefined particle velocity (assumed to be zero) would detrimentally lower the averages. The quantized nature of discrete particle counts ranging from 0-8 typically, produces “noisy” instantaneous results. One advantage that Eulerian methods offer over sparse Lagrangian methods, is that they guarantee all cells will contain an instantaneous unquantized/analog probability value, generally providing smoother contours, though shocks are still possible. To overcome these issues in sparse DPS plots, spatial-averages (spanwise) and time-averages are employed as shown in Figure 36. Very similar profiles are exhibited throughout the entire domain, $0 \leq x \leq 80$. The time average consists of 330 samples which were captured every 15 time steps, and the spanwise average consists of all 98 spanwise cells. The particle volume fraction comparison of Figure 36 shows that the two models are in qualitative agreement, with the largest discrepancy located nearer to the wall. This is largely due to the need for more statistical DPS samples. It is also possible that the CQMOM model is over-exhausted in the wall-adjacent cells, which are *always* fully exhausted to represent an incoming and oppositely outgoing wall-normal velocity, leaving no additional nodes for extra velocity diversity. If more than β unique prominent velocities seek representation in the PDF they become averaged together similar to what is depicted in Figure 19(a-c), where a $\beta = 1$-point quadrature is attempting to model the diversity of a bimodal PDF.

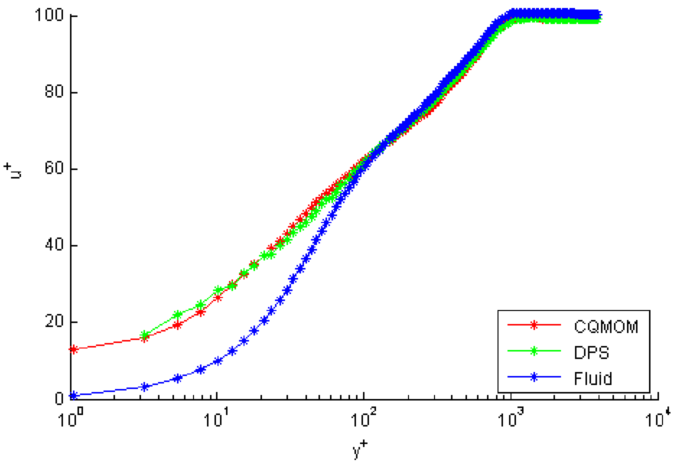


Figure 37. Spanwise-Averaged and Time-Averaged Particle Streamwise Velocity $\langle v_1 \rangle$ for St=16 at $x \approx$ 30.

Figure 37 plots the law of the wall, comparing the DPS and CQMOM streamwise particle velocities in the wall-normal plane (spanwise and time averaged). The two particle phase models match closely in the non-collision case, both deviating together from the carrier fluid velocity (blue line). This is a very promising result for CQMOM as the model shows excellent agreement with DPS streamwise velocities.

Wall time usage costs were tracked for the three different phase solvers and the fluid solver required about 4% of the computational costs, DPS about 3%, and CQMOM approximately 92%. However, when collisions were enabled, the costs for DPS increased fivefold, while the costs for CQMOM remained nearly constant (decreasing to 81% of total). Costs associated with collisional DPS solvers tends to increase almost exponentially with population, while CQMOM costs will remain nearly constant, independent of population density. Although CQMOM does impose an expensive initial overhead cost, adding additional particles or complexity such as periodic/reflective/Neumann boundaries, adds very little additional computational cost. This is advantageous for CQMOM in simulations with significantly denser and non-uniformly distributed particle concentrations.

E. Channel Flow Simulation

Analyzing the statistical aspects of the channel flow provides a reduced dataset that reveals underlying physics that are useful for comparison against other methods. Spatial averages are performed along the spanwise dimension (z) and time-averages are taken until a steady-state solution is reached. The simulation initial conditions consist of introducing the CQMOM particles with a homogenous volume fraction and are given the same initial instantaneous monokinetic velocities of the DNS carrier fluid field. After the CQMOM statistics converge once, the statistics are restarted once more to remove any artifacts from the initial transient behavior. The statistics shown below are those obtained from a second convergence. The fluid has a Reynolds number of $\mathrm{Re}_\tau = 200$ and the particle phase Stokes number is $\mathrm{St} = 2.5$. The channel domain spans non-dimensional lengths of $\{L_1, L_2, L_3\} = \{2\pi, 2, \pi\}$ in the streamwise, wall-normal, and spanwise dimensions, respectively, in terms of the channel half-height being unity, $L_2/2 = L_f = 1$.

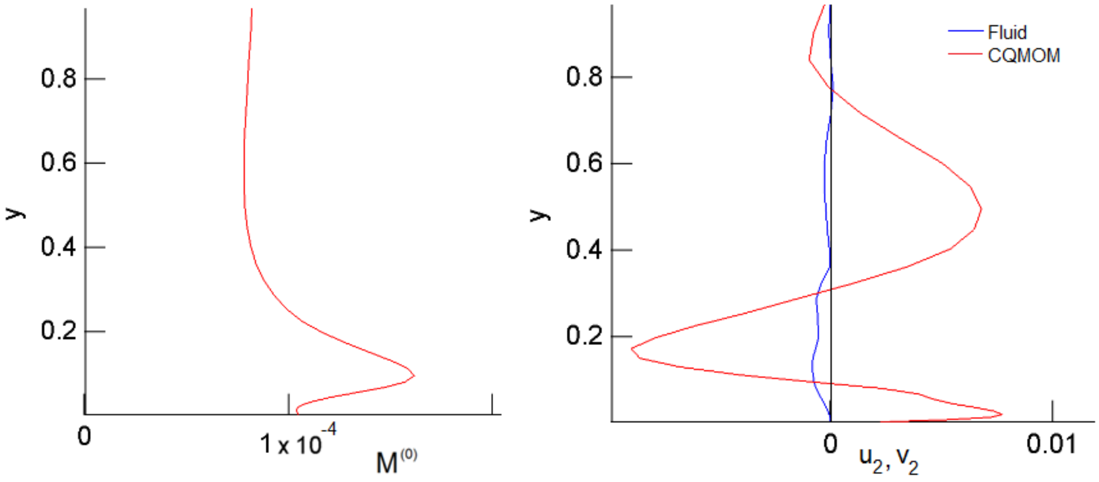
The grid contains $64 \times 64 \times 64$ internal cells (excluding ghost layers), making it readily applicable for domain decomposition in parallel computing.

1. Threshold Sensitivity

This section provides results that are erroneous in nature due to Wheeler algorithm thresholds which are too large (stringent), specifically, $\mathbb{W}_2 \gtrsim 5E^{-2}$ and $\mathbb{P} \gtrsim 1.2$. The minimum weight, $\mathbb{W}_1 = 1E^{-4}$, was less sensitive to instabilities or changes in the solution, and employing smaller values ($\mathbb{W}_1 = 1E^{-6}$) did not significantly improve the results. In general, larger thresholds offer a more conservative approach that provides increased stability because the localized PDF more cautiously tends toward a 1-point quadrature when the authenticity of the PDF is questionable. However, if these thresholds are *too* large then the PDF is too confined and cannot adequately represent the velocity diversity that may be necessary to mimic important physical features (PTC). In other words, the model may tend toward a monokinetic field too easily when it should retain significant granular temperatures and skew.

To demonstrate this, the particle volume fraction, $M^{(0)}$, is plotted in Figure 38(a) both spanwise and time averaged. It is observed through animations that a vast majority of particles quickly accumulate near the wall, gathering into fewer cells and reaching five to ten times their initial homogenous volume fraction. Then the particles reflect off the wall and tend to stagnate near $y \approx 0.1$ with mean wall-normal velocity approximately zero, $\langle v_2 \rangle \approx 0$. This dominating presence of particles creates a “barrier” structure that remains relatively dormant near $y \approx 0.1$ after having reflected off the wall once.

Additionally, this high-concentration barrier impedes the movement of particles from passing through it in the wall-normal direction, and continues to accumulate particles. The PDF quadrature in the y-direction becomes narrow (smaller variance in v_2) and the barrier's large weight dominates the field locally. Figure 38(b) supports the appearance of a local maximum in density at $y \approx 0.1$ by illustrating that the average wall-normal velocity (v_2) is negative above and positive below it.



(a) M_0

(b) $\langle v_2 \rangle$

Figure 38. Time and Span Averaged Erroneous Results

Reduced threshold values help overcome the problem, and those results are provided in Section §*4. 2-Node Statistics*. Thresholds of $\mathbb{W}_2$=1E^{-2} or $\mathbb{W}_2$=8E^{-3} significantly reduce accumulation errors resulting from the erroneously narrow PDF's shown above, but does result in more occurrences of erroneously large abscissas that must be captured and prevented by the CFL enforcement. Moreover, allowing $\mathbb{P} \rightarrow 1$ had the most significant improvement upon the ability of particles to accumulate directly at the wall.

Smaller values ($\mathbb{P} = 1.01$) allow for more accumulation of particles, whereas larger values ($\mathbb{P} \gtrsim 1.2$) tend to produce those erroneous effects seen in Figure 38.

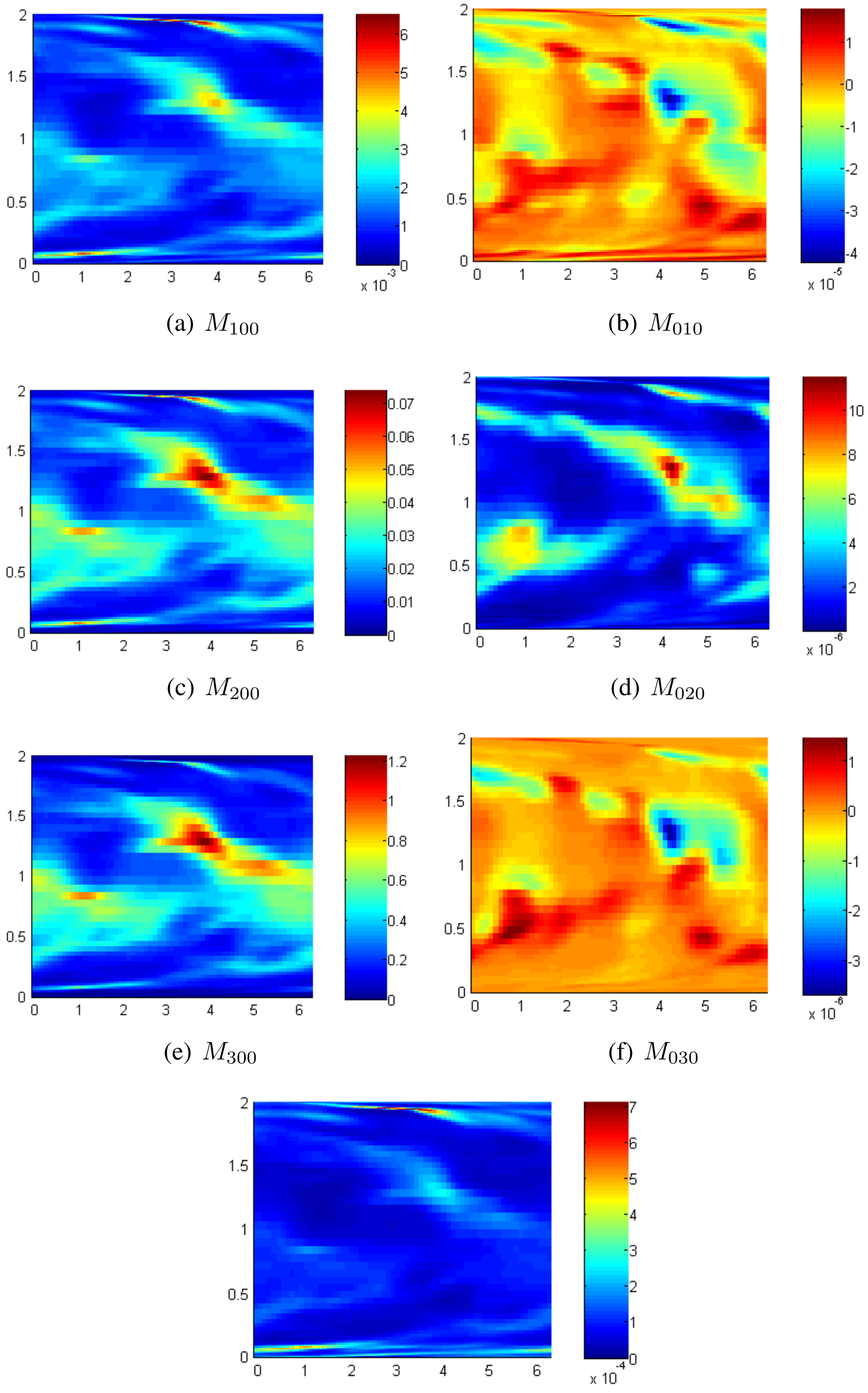
Lastly, a mere 2-point quadrature assumption is somewhat limiting for a collisionless turbulent case in which the actual particle velocities of the PDF may be extremely diverse and sparse. At best, a 2-point quadrature can represent a bimodal distribution with considerable accuracy; however, it is possible that a trimodal (or higher) distribution is necessary near the wall. If more nodes are required than available ($\beta > \mathbb{B}$), which is probable for a collisionless case with large St, then the 2-node assumption ($\mathbb{B} = 2$) will artificially reduce the strength of PTC that could otherwise be physically possible. Over-exhausting the nodes can lead to an over-accumulation of particles as described earlier by Figure 19(a-c), though not manifesting itself as a delta shock, but instead like the erroneous accumulation encountered in Figure 38 or Figure 23. To summarize, an undesired feature exists that allows for the erroneous "clumping" or accumulation of particles whenever the model is unable to represent the necessary variance in a discretized PDF. This mechanism occurs either when there are insufficient nodes available (β), or when the PDF is reduced to a 1-point quadrature for stability reasons – caused by numerical diffusion in the flux term that leads to moment misalignment. Either way, it is reasonable to say that the solution will have suffered from some type of erroneous accumulation of particles after a large number of time steps have elapsed. Once these particles gain a considerable "mass" (and momentum) they become increasingly dominant over the lesser features of the flow.

2. Instantaneous Fields

Recall that 2-point quadrature in 3-D velocity space operates with $K = 32$ moment fields in CQMOM, comprising the vector $\boldsymbol{M}$. It is compelling that these field quantities are capable of transporting themselves independently by numerical hyperbolic PDEs and still remain aligned. Figure 39 illustrates the six "simple" x and y moments and density in a wall normal plane at $z = L_z/2$. The relationship between these fields yields valuable information about the PDF. For example, if the second-order moment is large, and the first-order moment is small, this indicates that significant PTC is occurring in that region. In Figure 39(g), the high density "arms" are visible near the walls with far lower density structures near the centerline.

The length scales of the particle field structures are larger than that of the fluid structures because St > 1. Comparing the CQMOM density field against the fluid streamwise velocity field shows a strong correlation. It is expected that regions of high velocity will be devoid of particles since they tend to accumulate in regions of lower carrier fluid velocity. Figure 40 illustrates a wall-parallel plane located at near the wall, $y \approx 0$, and confirms that the regions of higher particle concentration pertain to lower fluid velocities. Observing the instantaneous particle concentrations in Figure 40(b) from the wall-normal perspective near the wall highlights the utility of moment methods and its unique ability to model the dispersed phase as an Eulerian continuum.

The results from[81] were run with a very similar fluid solver, but with a DPS treatment of the dispersed particles, providing a useful comparison for the CQMOM



(a) M_{100}

(b) M_{010}

(c) M_{200}

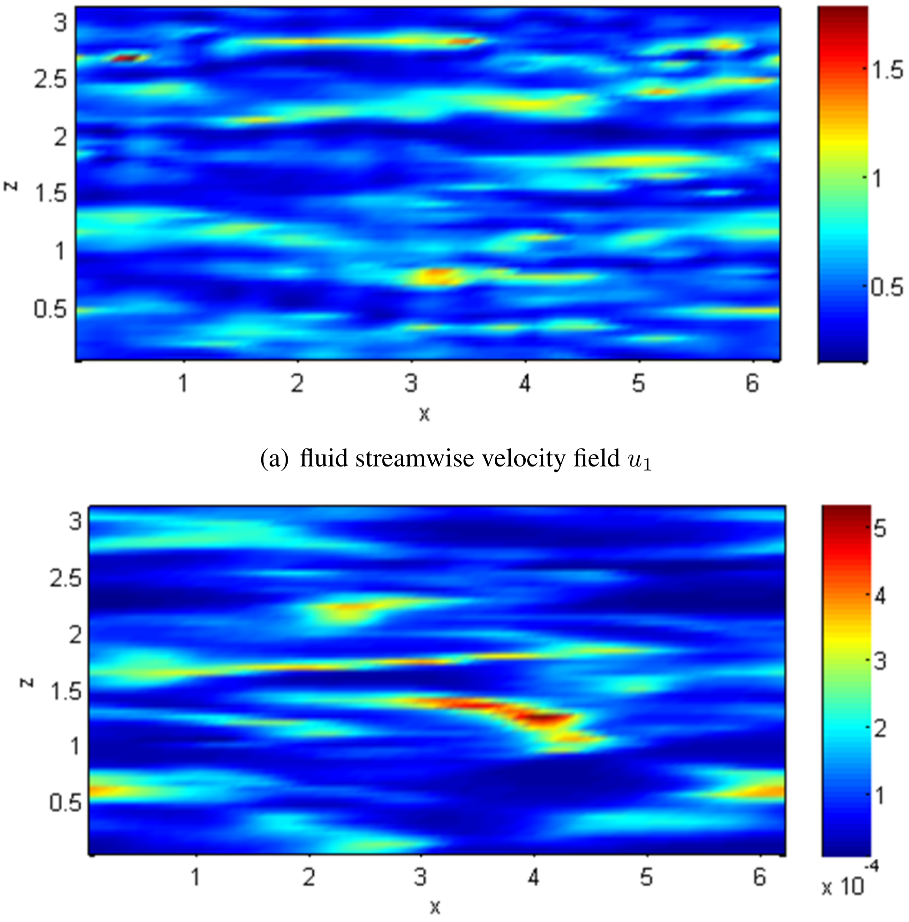
(d) M_{020}

(e) M_{300}

(f) M_{030}

(g) M_{000}

Figure 39. All Six "Simple" x and y Moment Fields and Volume Fraction



(a) fluid streamwise velocity field u_1

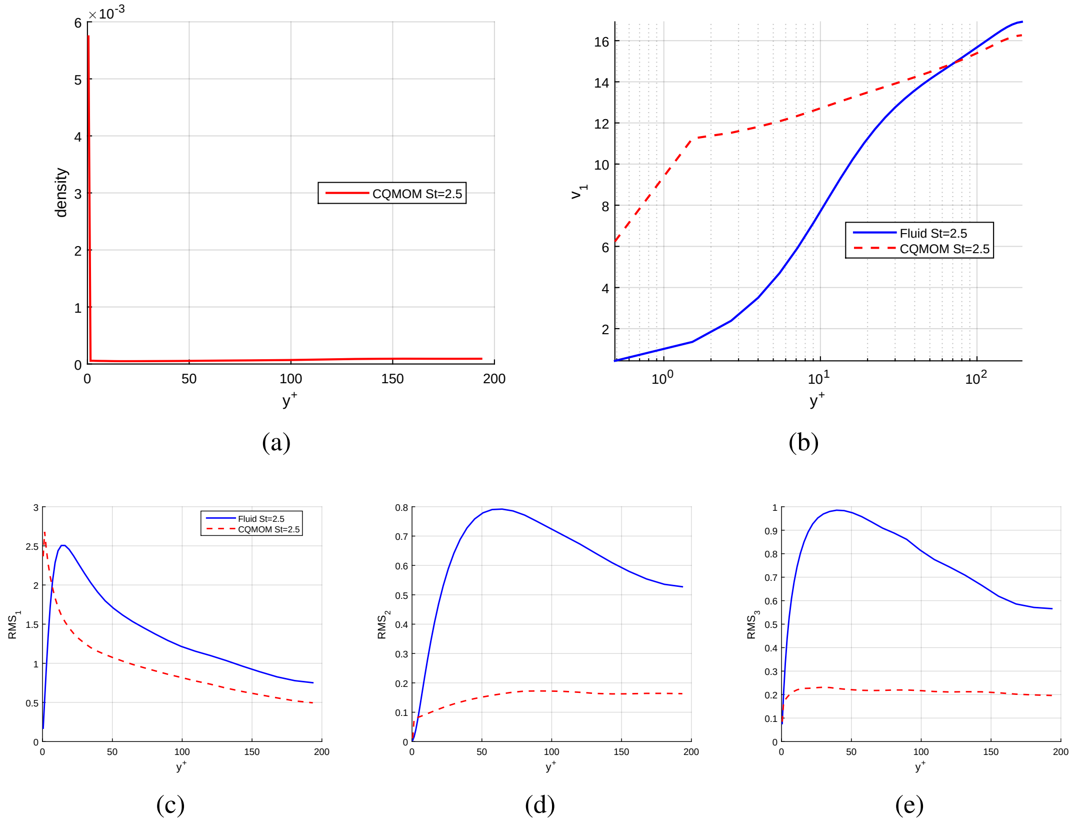
(b) CQMOM particle volume fraction M_0

Figure 40. Correlation Between Particle Volume Fraction and Fluid Velocity

results. The buildup of density near the walls is an effect of *turbophoresis*, the tendency of particles to migrate in the direction of decreasing turbulence.[82] Kuerten *et al.*[82] and Sardina *et al.*[83] determined that larger Stokes number generally increases the accumulation of collisionless particles at the wall (due to turbophoresis) up until a certain point.

3. 1-Node Statistics

A simplified 1-point quadrature restriction was imposed ($B = 1$) to test the initial capabilities of the solver at St $= 2.5$. This method is *not* be able to capture PTC because the PDF is a single (3-D) Delta function, but it may be able to approximate some of the cell-averaged quantities and will be a useful baseline result for comparison with the 2-point quadrature results in the next section. Consequently, the collision term, $\mathbb{C}$, will always be null and a “collisionless” 1-node case has no real significance. The 1-node assumption is bound to introduce artificial collisions merely by restricting the velocities to 1-point quadratures. Figure 41 shows that the 1-point quadrature has difficulty with the



(a)

(b)

(c)

(d)

(e)

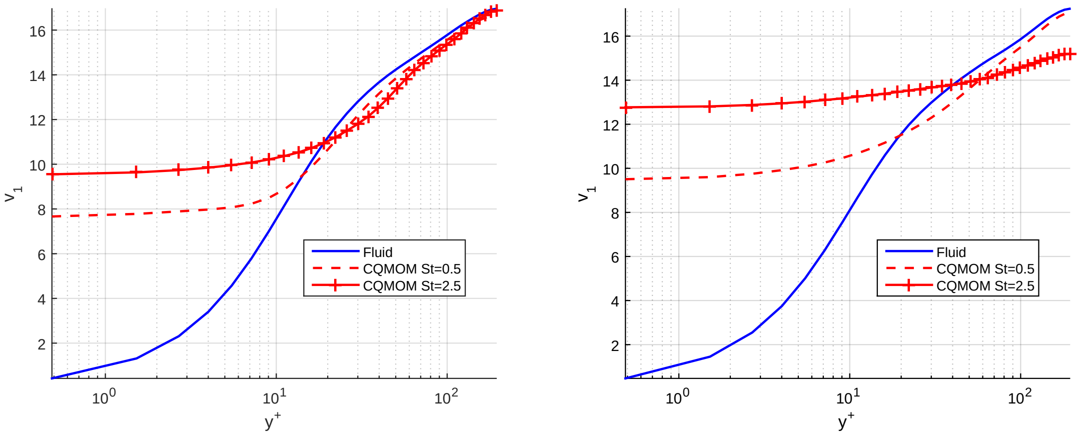
Figure 41. Collisionless 1-Point Quadrature, z and t-Averaged Statistics.

near-wall behavior and this is to be expected because the physical particles near the wall have a very complex PDF and experience significant PTC. Interestingly, the cell velocities and RMS values directly at the wall boundary are seemingly accurate, but the neighboring cell shows a discontinuity in all the measured quantities. This is because an unphysically large over-accumulation of M_0 occurs directly at the wall and momentum conservation dictates such a discontinuity in velocity too. Once the particles move toward the wall, they cannot escape because the 1-point simplification does not provide any wall-normal convective velocity back toward the flow centerline. The particles move on a one-way path toward the wall where they remain trapped.

The shapes and trends of the RMS profiles are very similar to previous results[81] (collisionless St $= 2.6$), but the CQMOM magnitude is too small for RMS_1 and too large for RMS_2 and RMS_3. This indicates that the DPS results correctly retain a better anisotropic velocity PDF with variances that can vary widely in each dimension. Obviously, the 1-node CQMOM case cannot capture any variance, and the RMS values are strictly resulting from the time-dependent fluctuations of the mean velocity, $\langle v \rangle$, which does not match the fluid exactly because the particles have a slower response time. Furthermore, to confirm that the CQMOM solver is behaving as expected, a single test case in the limit St $\rightarrow 0$ was performed, and CQMOM accurately predicts the exact same dispersed phase statistics matching the carrier fluid (in both velocity and RMS) even for a 2-point quadrature case.

4. 2-Node Statistics

A 2-point quadrature assumption is the minimum requirement for realizing PTC ($\sigma^2 > 0$) so that the particle velocities are multivalued within a cell. Stable results were obtained for statistical analysis using threshold vectors: $\mathbb{P}$=$\{1.05, 1.02, 1.0\}$ and $\mathbb{W}$=$\{1E^{-6}, 9E^{-3}\}$. The following statistics compare the *collisionless* case with the elastic (ω=1) isotropic *collision* case, using two different Stokes numbers (St=0.5 and St=2.5). Spanwise and time-averaged statistics were selected to provide a compacted form of the overall flow behavior and so that they can be compared to DPS results in previous literature.[18,23,81,82,84]

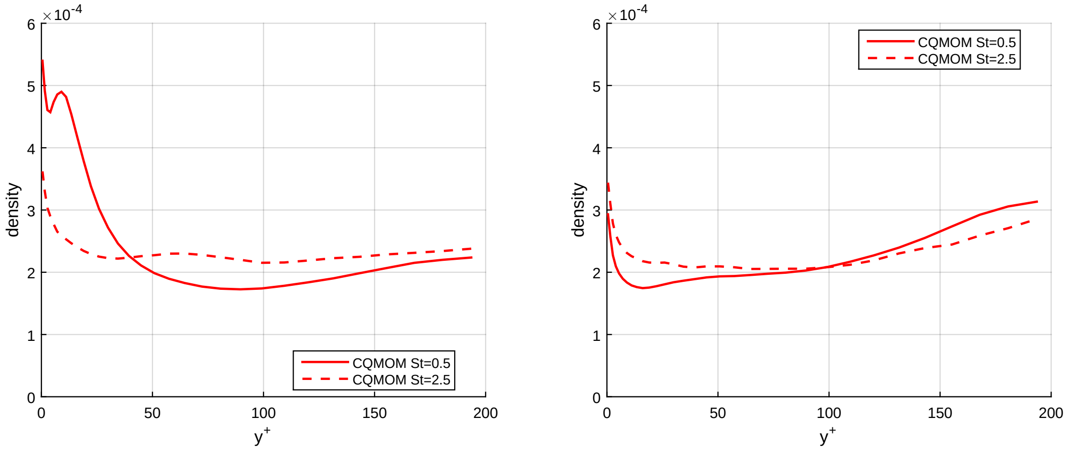


(a) Collisionless (b) Collisional

Figure 42. Streamwise Velocity z and t-Averaged Statistics. Without and With Isotropic Collisions

Figure 42 depicts the streamwise velocity profile plot for comparing the DNS carrier fluid and the CQMOM dispersed particles. The carrier fluid's viscosity enables the streamwise fluid momentum to readily transfer between the wall and the centerline in the wall-normal direction, producing a typical law-of-the-wall profile with no-slip condition. The dispersed particles do *not* have that same restriction and are able to move much faster in the streamwise direction at the wall. As expected, the heavy particles (St $= 2.5$) do *not* follow/mimic this fluid profile too closely. The larger Stokes number yields the most deviation from the fluid and flattest streamwise velocity profiles, where the change in velocity between maximum and minimum is less reduced. In the collisionless case, this is due to the increased random distribution of particles that retain memory of their previous trajectories at larger Stokes number. For the collisional case, the profile is even flatter because momentum is more quickly transferred between particles in the wall-normal direction and slip at the wall is prevalent. Frequent inter-particle collisions and decreased drag effectiveness allow for increased momentum transfer in the wall-normal direction that generates the "flatter" streamwise velocity profile. Therefore, particles are prevented from reaching the full speed of the carrier fluid at the centerline while showing increased particle velocity at the wall. Even with the 2-node limitation (which seems to be the most restrictive in the wall-normal dimension), CQMOM still produces convincing mean streamwise velocity profiles that agree with previous works. Recall that the mean velocity is obtained from the moment fields: $\langle v_d \rangle \equiv M_d^{(1)}/M^{(0)}$.

The volume fraction is fairly uniform as depicted in Figure 43, with some accumulation at the wall, but not as pronounced as the results of Vance *et al.*[81] Enabling the collisions tends to decrease the particle accumulation at the wall, and larger Stokes number increases the uniformity of the concentration distribution. The 3-D turbulent channel flow was much less susceptible to ill-conditioned inversions when $\mathbb{P} > 1.05$, but



(a) Collisionless (b) Collisional

Figure 43. Volume Fraction z and t-Averaged Statistics. Without and With Collisions

this hindered the accumulation of particles at the wall. Values as low as $\mathbb{P}$=1.0001 were also tested and yielded better physical results. The collisionless St=0.5 case shows a local minimum in concentration near the wall. This is not a physical result, but is most likely due to the over-exhaustion of nodes.

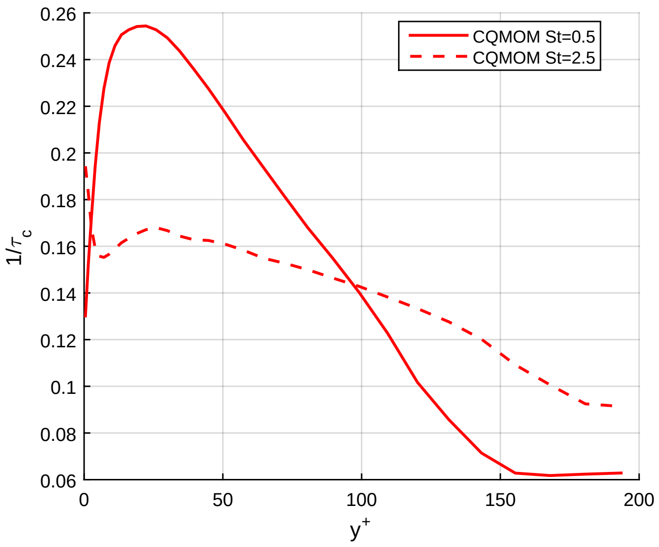
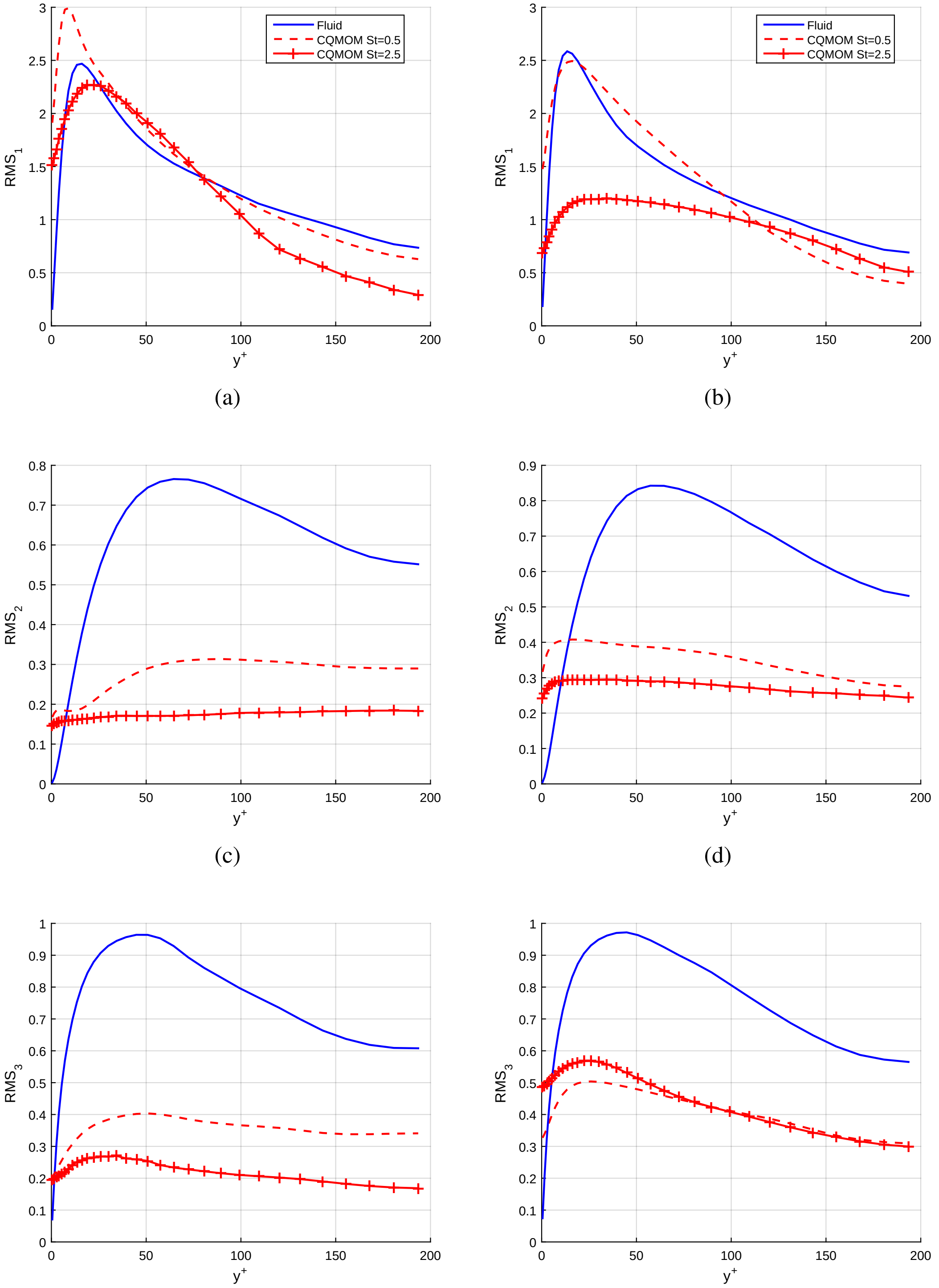


Figure 44. Collision Frequency $(1/\tau_c)$ z and t-Averaged

The collision frequency plotted in Figure 44 depends on the density and granular temperature (which is related to the average RMS in all dimensions). The St=0.5 case generally exhibited larger collision frequency at the steady-state, mainly due to the larger shear in the (steeper) v_1 profile, which generates a larger RMS$_1$. The larger Stokes number has a spike in collisions directly at the wall due to the larger density present there. Initially the collision frequency is larger during the transient period, but then the v_1 profile eventually flattens and the collisions become less frequent to maintain the steady-state solution.

Lastly, the RMS results are plotted. The streamwise fluctuations (RMS$_1$) are the largest because the corresponding velocities (u_1 and v_1) are also the largest. It is generally known that the spanwise fluctuations (RMS$_3$) are larger than the wall-normal fluctuations (RMS$_2$)[18,81] and CQMOM confirms this trend. CQMOM is also able to reproduce large RMS values near the wall which are larger than the fluid RMS. This is possible in the streamwise dimension because the particle phase is allowed to slip at the wall and more PTC leads to increased RMS$_1$. However, compared to DPS, CQMOM generally under-predicts the RMS$_1$ magnitudes while it over-predicts both the RMS$_2$ and RMS$_3$, suggesting that CQMOM may have difficulty maintaining highly anisotropic multidimensional PDF's. This may be partly due to the thresholds of the inversion method which intentionally restrict the PDF to increase stability by preventing the formation of spurious abscissas. Although the RMS values predicted by CQMOM generally differ in magnitude compared to DPS results, the qualitative trends are similar.



(a)

(b)

(c)

(d)

(e) Collisionless

(f) Collisional

Figure 45. RMS z and t-Averaged Statistics. Left Column is Collisionless and Right Column With Isotropic Collisions

In conclusion, it is likely that the quadrature nodes are over-exhausted, especially at the wall in the wall-normal direction where a 2-point quadrature (bimodal) PDF may be insufficient. CQMOM shows better numerical stability and agrees more favorably with DPS results when the isotropic collisions are *enabled*. The collision term offers at least one advantage in that the velocity PDF will naturally return to an equilibrium (Gaussian) distribution which a 2-point quadrature assumption is more capable of representing. The equilibrium distribution is simpler to represent than the sophisticated velocity PDF's that are possible far from equilibrium. Furthermore, the multidimensional BGK collision model provides a "redistribution effect" where highly anisotropic regions attempt to become isotropic by redistributing energy from dimensions with more variance to dimensions with less variance. It is a great achievement that this level of predictability is possible with only a 2-point quadrature. It will be worthwhile in future work to measure the solution sensitivity to PDF refinement by using 3-point quadrature especially in the wall-normal direction.

VII. SUMMARY

Numerous research efforts have focused on developing Quadrature-Based Moment Method models, but it has not yet seen wide-spread use in applications, especially in 3-D. Legacy methods such as the N-S equations (Kn $\ll 1$) or Lagrangian methods (Kn $\gg 1$) are optimal for their respective Knudsen regimes, but QBMM models have been demonstrated to predict dispersed particle phase flows as an Eulerian continuum in finite Knudsen (Kn $\approx O(1)$) regimes of single and dispersed phases, as well as in large Stokes number regimes (St > 1) of multiphase flows. Traditional multi-species diffusion models are only applicable at St $\ll 1$, but QBMM offers a two-fluid (Eulerian-Eulerian) formulation for arbitrary Stokes number. Lagrangian particle-tracking models (DPS) offer hybrid Eulerian-Lagrangian solutions for multiphase flows which are easily capable of realizing PTC. However, DPS is computationally prohibitive for large particle populations, challenging due to the load balancing of non-uniformly distributed concentrations (clustering), and has difficulty modeling collisions in high concentration dispersions. In general, CQMOM does not have those shortcomings and can realize particle trajectory crossing (PTC) which is a vital requirement for generating physical solutions in non-hydrodynamic fluids or dispersed multiphase flows.

In the present work, CQMOM has been demonstrated to model multivalued ($\beta = 2$) multidimensional ($D = 3$) velocity field flows which extend beyond the capabilities of previous QMOM models. An abscissa proximity ratio ($\mathbb{P}$) is suggested to improve the stability of CQMOM, especially applicable in higher dimensional velocity space ($D = 3$).

Implementation of the Wheeler inversion algorithm and second-order accurate upwind spatial flux and temporal schemes were successful. The `superbee` slope limiter was implemented with great success and offers better shock preservation than the `minmod` limiter from previous works. A non-uniform grid spacing was also successfully modeled and the results behaved as expected for an Eulerian formulation, which was a critical step toward modeling boundary layer and channel flows within a rectilinear grid. Some major obstacles were overcome during the computational model design phase and many optimizations were successfully added into a high-performance C++ code, including parallelization through domain decomposition. 2-D free jet cases were carried out with compelling results demonstrating that the 2-node assumption offers useful predictive capabilities despite its limitations on the PDF, especially with the collision model enabled. Additionally, 2-D intersecting free jets and Taylor-Green flows both revealed that a dimension-stepping scheme improved the flow symmetry.

Both collisionless and collisional (isotropic) particles were tested. Dispersed particle phases were modeled in two different particle-laden turbulent flows that were simulated with two varying Stokes numbers. The first case was a turbulent boundary layer that offered direct comparisons between CQMOM and a DPS model, and it showed great agreement in instantaneous volume fraction and velocity fields. The second simulation was a turbulent channel flow that offered statistical comparisons with a DNS fluid solver that also showed good agreement in the time-averaged velocity fields with “flatter” streamwise profiles for the dispersed particles than for the fluid phase. The wall-normal direction showed signs of node over-exhaustion, indicating that possibly more nodes are required, $\beta > 2$ in the y-direction. Wall-normal density and RMS profiles suffered as a result, but still showed qualitative agreement with previous works.

Future work may benefit from implementing two-way coupling schemes for high mass loading on the fluid, adding additional nodes ($\beta \geq 3$), expanding the internal state space to include other attributes such as particle diameter (polydisperse, $Q > 0$), reducing the inherent errors in higher order FV flux schemes that yield moment misalignment, broadening the flux schemes to encompass unstructured grids, and extending the quadrature nodes beyond simple Dirac delta functions (EQMOM).

REFERENCES

[1]Marchisio, D. L. and Fox, R. O., *Computational Models for Polydisperse Particulate and Multiphase Systems*, Cambridge University Press, Cambridge, 2013.

[2]Bird, G. A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Vol. 42, Clarendon Press, Oxford, 1994.

[3]Fréret, L., Laurent, F., de Chaisemartin, S., Kah, D., Fox, R. O., Vedula, P., Reveillon, J., Thomine, O., and Massot, M., "Turbulent Combustion of Polydisperse Evaporating Sprays with Droplet Crossing: Eulerian Modeling of Collisions at Finite Knudsen and Validation," *Center for Turbulence Research*, 2009.

[4]Crowe, C., Sommerfeld, M., and Tsuji, Y., *Multiphase Flows with Droplets and Particles*, CRC Press, Boca Raton, Florida, 1998.

[5]Gad-El-Hak, M., "Gas and Liquid Transport at the Microscale," *Heat Transfer Engineering*, Vol. 27, No. 4, 2006, pp. 13–29.

[6]Kalarakis, A., Michalis, V., Skouras, E., and Burganos, V., "Mesoscopic Simulation of Rarefied Flow in Narrow Channels and Porous Media," *Transport in Porous Media*, Vol. 94, No. 1, 2012, pp. 385–398.

[7]Brilliantov, N. and Pöschel, T., *Kinetic Theory of Granular Gases*, Oxford University Press, Oxford, 2004.

[8]McGraw, R., "Description of Aerosol Dynamics by the Quadrature Method of Moments," *Aerosol Science and Technology*, Vol. 27, 1997, pp. 255–265.

[9]McGraw, R., Leng, L., Zhu, W., Riemer, N., and West, M., "Aerosol Dynamics Using the Quadrature Method of Moments: Comparing Several Quadrature Schemes with Particle-Resolved Simulation," *Journal of Physics: Conference Series*, Vol. 125, 2008, pp. 012020.

[10]Wright, D. L., "Numerical Advection of Moments of the Particle Size Distribution in Eulerian Models," *Journal of Aerosol Science*, Vol. 38, No. 3, 2007, pp. 352–369.

[11]Laurent, F., Massot, M., and Villedieu, P., "Eulerian Multi-Fluid Modeling for the Numerical Simulation of Coalescence in Polydisperse Dense Liquid Sprays," *Journal of Computational Physics*, Vol. 194, No. 2, 2004, pp. 505–543.

[12]Kah, D., Laurent, F., Fréret, L., de Chaisemartin, S., Fox, R. O., Reveillon, J., and Massot, M., "Eulerian Quadrature-Based Moment Models for Dilute Polydisperse Evaporating Sprays," *Flow, Turbulence and Combustion*, Vol. 85, No. 3, 2010, pp. 649–676.

[13] Buffo, A., Vanni, M., and Marchisio, D., "Multidimensional Population Balance Model for the Simulation of Turbulent Gas-Liquid Systems in Stirred Tank Reactors," *Chemical Engineering Science*, Vol. 70, 2011, pp. 31–44.

[14] Buffo, A., Vanni, M., Marchisio, D. L., and Fox, R. O., "Multivariate Quadrature-Based Moments Methods for Turbulent Polydisperse Gas-Liquid Systems," *International Journal of Multiphase Flow*, Vol. 50, 2013, pp. 41.

[15] Marchisio, D. L., Vigil, R. D., and Fox, R. O., "Quadrature Method of Moments for Aggregation-Breakage Processes," *Journal of Colloid and Interface Science*, Vol. 258, 2002, pp. 322–334.

[16] Marchisio, D. L., Vigil, R. D., and Fox, R. O., "Implementation of the Quadrature Method of Moments in CFD Codes for Aggregation-Breakage Problems," *Chemical Engineering Science*, Vol. 58, 2003, pp. 3337–3351.

[17] Ghosh, S., *Configurational Effect on Dust Cloud Formation and Brownout*, Ph.D. thesis, Iowa State University, 2010.

[18] Wang, Q. and Squires, K. D., "Large Eddy Simulation of Particle-Laden Turbulent Channel Flow," *Physics of Fluids*, Vol. 8, No. 5, 1996, pp. 1207.

[19] Vance, M. W. and Squires, K. D., "An Approach to Parallel Computing in an Eulerian-Lagrangian Two-Phase Flow Model," *Proceedings of ASME FED 2002 ASME Fluids Engineering Division Summer Conference*, Montreal, Quebec, Canada, July 2002, pp. 420–430.

[20] Fan, L.-S. and Zhu, C., *Principles of Gas-Solid Flows*, Cambridge University Press, Cambridge, 1998.

[21] Syal, M. and Leishman, J. G., "Modeling of Bombardment Ejections in the Rotorcraft Brownout Problem," *AIAA Journal*, Vol. 51, No. 4, 2013, pp. 849–866.

[22] Février, P., Simonin, O., and Squires, K. D., "Partitioning of Particle Velocities in Gas-Solid Turbulent Flows Into a Continuous Field and a Spatially Uncorrelated Random Distribution: Theoretical Formalism and Numerical Study," *Journal of Fluid Mechanics*, Vol. 533, 2005, pp. 1–46.

[23] Mallouppas, G. and van Wachem, B., "Large Eddy Simulations of Turbulent Particle-Laden Channel Flow," *International Journal of Multiphase Flow*, Vol. 54, 2013, pp. 65–75.

[24] Dorao, C. A. and Jakobsen, H. A., "Numerical Calculation of the Moments of the Population Balance Equation," *Journal of Computational and Applied Mathematics*, Vol. 196, No. 2, 2006, pp. 619–633.

[25] David H. Richter and Peter P. Sullivan, "Momentum Transfer in a Turbulent, Particle-Laden Couette Flow," *Physics of Fluids*, Vol. 25, No. 5, 2013, pp. 053304–053304–20.

[26] Gualtieri, P., Picano, F., Sardina, G., and Casciola, C. M., "Clustering and Turbulence Modulation in Particle-Laden Shear Flows," *Journal of Fluid Mechanics*, Vol. 715, 2013, pp. 134.

[27] Harting, J., Frijters, S., Ramaioli, M., Robinson, M., Wolf, D. E., and Luding, S., "Recent Advances in the Simulation of Particle-Laden Flows," *The European Physical Journal Special Topics*, Vol. 223, No. 11, 2014, pp. 2253–2267.

[28] Li, W. and Ladeinde, F., "Analysis of Interacting, Under-Expanded, Rarefied Jets," July 2010.

[29] Capecelatro, J. and Desjardins, O., "An Euler-Lagrange Strategy for Simulating Particle-Laden Flows," *Journal of Computational Physics*, Vol. 238, 2013, pp. 1–31.

[30] Yoon, C. and McGraw, R., "Representation of Generally Mixed Multivariate Aerosols by the Quadrature Method of Moments: I. Statistical Foundation," *Journal of Aerosol Science*, Vol. 35, No. 5, 2004, pp. 561–576.

[31] Yoon, C. and McGraw, R., "Representation of Generally Mixed Multivariate Aerosols by the Quadrature Method of Moments: II. Aerosol Dynamics," *Journal of Aerosol Science*, Vol. 35, No. 5, 2004, pp. 577–598.

[32] Yoon, C., *Representation of Generally-Mixed Multivariate Aerosol by the Quadrature Method of Moments*, Ph.D. thesis, Stony Brook University, 2005.

[33] Gidaspow, D., *Multiphase Flow and Fluidization: Continuum and Kinetic Theory Descriptions*, Academic Press, Boston, 1994.

[34] Michaelides, E., *Particles, Bubbles and Drops: Their Motion, Heat and Mass Transfer*, World Scientific, Hackensack, NJ, 2006.

[35] Fox, R. O., "On Multiphase Turbulence Models for Collisional Fluid-Particle Flows," *Journal of Fluid Mechanics*, Vol. 742, 2014, pp. 368.

[36] Passalacqua, A. and Fox, R., "Documentation of Open-Source MFIX-QMOM Software for Gas-Solids Flows," 2012.

[37] Carnahan, N. F. and Starling, K. E., "Equation of State for Nonattracting Rigid Spheres," *The Journal of Chemical Physics*, Vol. 51, No. 2, 1969, pp. 635.

[38] Lun, C. K. K., Savage, S. B., Jeffrey, D. J., and Chepurniy, N., "Kinetic Theories for Granular Flow: Inelastic Particles in Couette Flow and Slightly Inelastic Particles in a General Flowfield," *Journal of Fluid Mechanics*, Vol. 140, No. 1, 1984, pp. 223–256.

[39] Passalacqua, A., Fox, R. O., Garg, R., and Subramaniam, S., "A Fully Coupled Quadrature-Based Moment Method for Dilute to Moderately Dilute Fluid-Particle Flows," *Chemical Engineering Science*, Vol. 65, No. 7, 2010, pp. 2267–2283.

[40] Balachandar, S. and Eaton, J. K., "Turbulent Dispersed Multiphase Flow," *Annual Review of Fluid Mechanics*, Vol. 42, No. 1, 2010, pp. 111–133.

[41] Haar, D. t., *Elements of Statistical Mechanics*, Butterworth-Heinemann, Oxford, 1995.

[42] Desjardins, O., Fox, R. O., and Villedieu, P., "A Quadrature-Based Moment Method for Dilute Fluid-Particle Flows," *Journal of Computational Physics*, Vol. 227, No. 4, 2008, pp. 2514–2539.

[43] Mohamad, A. A., *Lattice Boltzmann Method*, Springer, New York, 2011.

[44] Yuan, C. and Fox, R. O., "Conditional Quadrature Method of Moments for Kinetic Equations," *Journal of Computational Physics*, Vol. 230, No. 22, 2011, pp. 8216–8246.

[45] Vikas, V., Wang, Z. J., Passalacqua, A., and Fox, R. O., "Realizable High-Order Finite-Volume Schemes for Quadrature-Based Moment Methods," *Journal of Computational Physics*, Vol. 230, No. 13, 2011, pp. 5328–5352.

[46] Mazzei, L., Marchisio, D. L., and Lettieri, P., "New Quadrature-Based Moment Method for the Mixing of Inert Polydisperse Fluidized Powders in Commercial CFD Codes," *American Institute of Chemical Engineers*, 2012.

[47] Mazzei, L., *Eulerian Modelling and Computational Fluid Dynamics Simulation of Mono and Polydisperse Fluidized Suspensions*, Phd thesis, University College London, Department of Chemical Engineering, October 2008.

[48] Mazzei, L., Marchisio, D., and Lettieri, P., "CFD Simulations of the Mixing of Inert Polydisperse Fluidized Suspensions Using QMOM and DQMOM," 2009.

[49] Williams, F. A., "Spray Combustion and Atomization," *Physics of Fluids*, Vol. 1, No. 6, 1958, pp. 541.

[50] Chalons, C., Kah, D., and Massot, M., "Beyond Pressureless Gas Dynamics: Quadrature-Based Velocity Moment Models," , 2010.

[51] Fox, R. O., “Higher-Order Quadrature-Based Moment Methods for Kinetic Equations,” *Journal of Computational Physics*, Vol. 228, No. 20, 2009, pp. 7771–7791.

[52] Fox, R. O., “A Quadrature-Based Third-Order Moment Method for Dilute Gas-Particle Flows,” *Journal of Computational Physics*, Vol. 227, No. 12, 2008, pp. 6313–6350.

[53] Laurent, F., Vié, A., Chalons, C., Fox, R., and Massot, M., “A Hierarchy of Eulerian Models for Trajectory Crossing in Particle-Laden Turbulent Flows Over a Wide Range of Stokes Numbers,” .

[54] Chalons, C., Fox, R. O., and Massot, M., “A multi-Gaussian Quadrature Method of Moments for Gas-Particle Flows in a LES Framework,” *Center for Turbulence Research*, 2010, p. 347.

[55] Vié, A., Chalons, C., Fox, R. O., Laurent, F., and Massot, M., “A Multi-Gaussian Quadrature Method of Moments for Simulating High Stokes Number Turbulent Two-Phase Flows,” *Center for Turbulence Research*, 2011, p. 309.

[56] Fox, R. O. and Vedula, P., “Quadrature-Based Moment Model for Moderately Dense Polydisperse GasParticle Flows,” *Industrial & Engineering Chemistry Research*, Vol. 49, No. 11, 2010, pp. 5174–5187.

[57] Bodrova, A. and Brilliantov, N., “Self-Diffusion in Granular Gases: an Impact of Particles’ Roughness,” *Granular Matter*, Vol. 14, No. 2, 2012, pp. 85–90.

[58] Reif, F., *Fundamentals of Statistical and Thermal Physics*, McGraw Hill, New York, 1965.

[59] Bardow, A., Karlin, I. V., and Gusev, A. A., “Multispeed Models in Off-Lattice Boltzmann Simulations,” *Physical review. E, Statistical, nonlinear, and soft matter physics*, Vol. 77, No. 2 Pt 2, 2008, pp. 025701.

[60] Tang, G. H., Tao, W. Q., and He, Y. L., “Three-Dimensional Lattice Boltzmann Model for Gaseous Flow in Rectangular Microducts and Microscale Porous Media,” *Journal of Applied Physics*, Vol. 104, No. 10, 2005, pp. 104918–104918–8.

[61] Beylich, A. E., “Solving the Kinetic Equation for All Knudsen Numbers,” *Physics of Fluids*, Vol. 12, No. 2, 2000, pp. 444.

[62] Wheeler, J. C., “Modified Moments and Gaussian Quadratures,” *Rocky Mountain J. Math*, Vol. 4, No. no. 2, 1974, pp. 287–296.

[63] David P. Doane and Lori E. Seward, “Measuring Skewness: A Forgotten Statistic?” *Journal of Statistics Education*, Vol. 19, No. 2, 2011.

[64] LeVeque, R. J., *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, New York, 2002.

[65] Passalacqua, A. and Fox, R. O., “Multiphase CFD for Gas-Particle Flows: Beyond the Two-Fluid Model,” 2009.

[66] Gordon, R. G., “Error Bounds in Equilibrium Statistical Mechanics,” *J. Math. Phys.*, Vol. 9, 1968, pp. 655.

[67] Marchisio, D. L., Pikturna, J. T., Fox, R. O., Vigil, R. D., and Barresi, A. A., “Quadrature Method of Moments for Population-Balance Equations,” *American Institute of Chemical Engineers*, Vol. 49, No. 5, 2003, pp. 1266.

[68] Thein, F., *On the Efficiency and Condition of the Core Routine of the Quadrature Methods of Moments (QMOM)*, Ph.D. thesis, Weierstrass-Institut für Angewandte Analysis & Stochastik Berlin, 2011.

[69] Mazzei, L., “Limitations of Quadrature-Based Moment Methods for Modeling Inhomogeneous Polydisperse Fluidized Powders,” *Chemical Engineering Science*, Vol. 66, 2011, pp. 3628–3640.

[70] John, V. and Thein, F., “On the Efficiency and Robustness of the Core Routine of the Quadrature Method of Moments (QMOM),” *Chemical Engineering Science*, Vol. 75, 2012, pp. 327–333.

[71] Fox, R. O., “Optimal Moment Sets for Multivariate Direct Quadrature Method of Moments,” *Ind. Eng. Chem. Res.*, Vol. 48, 2009, pp. 9686–9696.

[72] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in C. The Art of Scientific Computing.*, Cambridge University Press, 2nd ed., 1992.

[73] Fan, R., Marchisio, D. L., and Fox, R. O., “Application of the Direct Quadrature Method of Moments to Polydisperse Gas-Solid Fluidized Beds,” *Powder Technology*, Vol. 139, 2004, pp. 7–20.

[74] Marchisio, D. L. and Fox, R. O., “Solution of Population Balance Equations Using the Direct Quadrature Method of Moments,” *Aerosol Science*, Vol. 36, 2005, pp. 43–73.

[75] Gottlieb, S. and Gottlieb, L. J., “Strong Stability Preserving Properties of Runge-Kutta Time Discretization Methods for Linear Constant Coefficient Operators,” *Journal of Scientific Computing*, Vol. 18, No. 1, Feb. 2003, pp. 83–109.

[76] Passalacqua, A., Galvin, J. E., Vedula, P., Hrenya, C. M., and Fox, R. O., "A Quadrature-Based Kinetic Model for Dilute Non-Isothermal Granular Flows," *Commun. Comput. Phys.*, Vol. 10, 2011, pp. 216–252.

[77] Bhatnagar, P. L., Gross, E. P., and Krook, M., "A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems," *Phys. Rev.*, Vol. 94, May 1954, pp. 511–525.

[78] Cody, G. D., Goldfarb, D. J., Storch, G. V., and Norris, A. N., "Particle Granular Temperature in Gas Fluidized Beds," *Powder Technology*, Vol. 87, No. 3, 1996, pp. 211–232.

[79] Keshavarz, B. and McKinley, G. H., "When Viscous Jets Collide; Liquid Chains, Threads, Webs, Fishbones and Balloons," 2013.

[80] Fujimoto, T. and Niimi, T., "Three-Dimensional Structures of Interacting Freejets," *AIAA Journal*, 1988, pp. 391–406.

[81] Vance, M. W., Squires, K. D., and Simonin, O., "Properties of the Particle Velocity Field in Gas-Solid Turbulent Channel Flow," *Physics of Fluids*, Vol. 18, No. 6, 2006, pp. 063302–063302–13.

[82] Kuerten, J., "Subgrid Modeling in Particle-Laden Channel Flow," *Physics of Fluids*, Vol. 18, No. 2, 2006, pp. 025108–025108–13.

[83] Sardina, G., Schlatter, P., Brandt, L., Picano, F., and Casciola, C. M., "Wall Accumulation and Spatial Localization in Particle-Laden Wall Flows," *Journal of Fluid Mechanics*, Vol. 699, 2012, pp. 50.

[84] Arcen, B., Tanière, A., and Oesterlé, B., "On the Influence of Near-Wall Forces in Particle-Laden Channel Flows," *International Journal of Multiphase Flow*, Vol. 32, No. 12, 2006, pp. 1326–1339.