A three-dimensional unified gas-kinetic wave-particle solver for flow computation in all regimes

Cite as: Phys. Fluids 32, 096108 (2020); https://doi.org/10.1063/5.0021199
Submitted: 08 July 2020 . Accepted: 25 August 2020 . Published Online: 18 September 2020

Yipei Chen (陈亿沛), Yajun Zhu (朱亚军), and Kun Xu (徐昆)

COLLECTIONS

Note: This paper is part of the Special Topic, Advances in Micro/Nano Fluid Flows: In Memory of Prof. Jason Reese.
This paper was selected as an Editor’s Pick
A three-dimensional unified gas-kinetic wave-particle solver for flow computation in all regimes

Yipei Chen (陈亿沛), Yajun Zhu (朱亚军), and Kun Xu (徐昆)

AFFILIATIONS
1Department of Mathematics, Hong Kong University of Science and Technology, Hong Kong, China
2National Key Laboratory of Science and Technology on Aerodynamic Design and Research, Northwestern Polytechnical University, Xi’an, Shaanxi 710072, China
3Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology, Hong Kong, China

Note: This paper is part of the Special Topic, Advances in Micro/Nano Fluid Flows: In Memory of Prof. Jason Reese.

I. INTRODUCTION

Non-equilibrium flow appears in a wide range of applications, such as the re-entry of a spacecraft in upper planetary atmospheres, vacuum devices, and fluid–structure interaction in microelectromechanical systems. For example, for a vehicle in a near-space flight at Mach number 6 and Reynolds number 5000, the local Knudsen number defined by $Kn_{local} = \|\nabla p\|/\rho$ with the mean free path $l$ can cover a wide range of values with five orders of magnitude difference. To simulate such a multiscale flow problem, a numerical algorithm is required to capture both equilibrium and non-equilibrium flow in different regimes in a single computation efficiently, such as the hydrodynamic regime in the highly compressible leading edge, the whole transition regime across the vehicle surface, and the rarefied regime in the highly expanding trailing edge. The problem becomes more difficult when the Mach number gets to 10 and 20.

The Euler and Navier–Stokes–Fourier (NSF) equations are hydrodynamic equations, which are valid for the continuum flow regime. They become inaccurate in the near continuum and...
transition regimes. On the other hand, the Boltzmann equation models the gas dynamics in the kinetic particle mean free path and collision time scale. The direct Boltzmann solver requires the cell size and time step resolution on the kinetic scale. The high dimensionality of the Boltzmann equation, nonlinearity of the collision term, and its integro-differential nature make the deterministic Boltzmann solver to be extremely expensive in terms of memory requirement and computational cost. Instead, for practical high-speed non-equilibrium flow computation, the direct simulation Monte Carlo (DSMC) method becomes the main choice in engineering applications due to its high efficiency of stochastic particle approach in recovering the Boltzmann modeling. The separate particle transport and collision treatment in DSMC require the numerical cell size and time step to be less than the particle mean free path and collision time. The DSMC can be hardly applied to the flow simulation in the continuum regime.

Developed at Sandia National Laboratories, the stochastic parallel rarefied-gas time-accurate analyzer (SPARTA) is an open source 2 and 3D DSMC simulator optimized for exascale parallel computing and embed with both static and dynamic load balancing across processors. Particles in SPARTA advect through a hierarchical oct-tree based Cartesian grid that overlays the simulation box. Additionally, dsmcFoam and its upgrade release dsmcFoam+ have been developed within the framework of OpenFOAM, which notably feature with dynamic load balancing on a 2D/3D polyhedron mesh, molecular vibrational and electronic energy modes, chemical reactions, and gravitational force. Other DSMC codes, such as MONACO, SMILE, and DAC, with different mesh topologies and collision treatments, can be found in the literature. Apart from stochastic solvers, the deterministic numerical schemes for the Boltzmann and kinetic model equations with discrete velocity method (DVM) have been extensively studied in the last several decades. 4–10 Nesvetaev-3D is an implicit solver on an unstructured mesh with both physical and velocity space decomposed parallelization. Recently, discrete unified gas-kinetic scheme (DUGKS)11,19 with the Shakov collision model has been developed in dugsFoam.11 Unlike the traditional DVM method, DUGKS is a multiscale solver, and the time step is not restricted by the particle collision time due to the coupled treatment of particle transport and collision. Besides the traditional physical space decomposition parallel strategy, dugsFoam features a parallel computing ability based on the velocity space decomposition.

The unified gas-kinetic wave-particle (UGKWP) method is a recently developed multiscale method for all flow regimes,22,23 and it is applicable to many multiscale transport problems, such as radiative heat transfer.24 The UGKWP is constructed under the unified gas-kinetic scheme (UGKS) framework.25 Instead of adopting the DVM approach in UGKS, the UGKWP uses both hydrodynamic wave and stochastic particles to model the flow evolution in the updated macroscopic flow variables and microscopic gas distribution function inside each control volume, where a time-dependent flux function is constructed through the coupled wave and particle interaction across a cell interface. Due to the wave-particle adaptation, the hydrodynamic equilibrium flow and the kinetic non-equilibrium particle transport can be simulated efficiently by their separate representation and commonly coupled dynamics under the control of local cell Knudsen number, i.e., $Kn_{c} = \tau / \Delta t$ with the particle relaxation time $\tau$ and numerical time step $\Delta t$. The UGKWP method has unified preserving property to present the physical solution in all flow regimes from the kinetic scale transport to the Navier–Stokes wave propagation without the constraint on the numerical cell size and time step being less than the particle mean free path and collision time.

The particle number in UGKWP is proportional to $\exp(-1/Kn_{c})$, which is a function of the cell Knudsen number. In the continuum flow regime, the particle number will be reduced significantly due to the small cell Knudsen number. In the highly rarefied regime, similar to DSMC, the particle will play a dominant role in the flow evolution with the association of statistical noise. The steady-state solution can be obtained from the averaging of time-accurate evolution solution. Due to the decoupled treatment of particle transport and collision, DSMC requires the cell size to be less than the particle mean free path. In contrast, UGKS and UGKWP have no such requirement due to coupled particle transport and collision in the gas evolution process. Moreover, the DSMC method handles the collision process by selecting particle pairs. In the low Knudsen number case, intensive collisions make DSMC impractical in the near continuum flow simulation, such as the flow simulation around a flying vehicle at an altitude below 80 km. In contrast, the UGKWP method removes collisional particles and re-samples them from the updated macroscopic flow variables. Therefore, the modeling of collective effect of particles’ collision within a time step in UGKWP does not require the time step and cell size to be less than the particle collision time and mean free path. As a result, the UGKWP is suitable for multiscale flow computation. Furthermore, with the implementation of particles in UGKWP, the ray effect, which is observed in DVM-type schemes in the highly rarefied flow regime due to the inadequate numerical resolution in the particle velocity space, can be totally avoided. The UGKWP has been tested in many 1D/2D cases with a large variation of Mach and Knudsen numbers.22,23 In this work, the UGKWP method will be constructed on a 3D unstructured mesh with parallel computing via spatial decomposition. The code is applicable to 3D flow simulation with arbitrary geometries in all flow regimes, especially suitable for hypersonic flow simulation. Great effort has been paid in the development of a 3D code by resolving all problems related to the complex geometry, reconstruction, multidimensional flux from wave-particle decomposition, particle tracking, and parallelization.

The rest of the paper is organized as follows: In Sec. II, the numerical procedure of the UGKWP method on an unstructured mesh is presented. Section III covers the construction of 3D UGKWP on a parallel framework. In Sec. IV, several numerical examples, including the 3D Sod shock tube inside a square-column, lid-driven cubic cavity flow, high-speed flow around a cube, and hypersonic flow passing through a space vehicle, will be computed to demonstrate the performance of the current algorithm. Conclusion and further developments are given in Sec. V.

II. 3D UNIFIED GAS-KINETIC WAVE-PARTICLE METHOD

In this section, the unified gas-kinetic wave-particle (UGKWP) method will be introduced. In the classical kinetic theory, the Boltzmann equation reads
\[ \frac{\partial f}{\partial t} + u \cdot \nabla f = Q(f), \]  

where \( f(x, u, t) \) is the gas distribution function, which depends on the particle velocity \( u \in \mathbb{R}^3 \), physical space position \( x \in \mathbb{R}^3 \), and time \( t \in \mathbb{R}^+ \). \( Q(f) \) is the nonlinear Boltzmann collision operator. In many applications, the collision term is usually simplified by other relaxation-type collision models \( S(f) \), such as Bhatnagar–Gross–Krook (BGK),\(^5\) the ellipsoidal statistical BGK (ES-BGK),\(^6\) and the Shakhov model.\(^7\) In general, the kinetic model can be written as

\[ \frac{\partial f}{\partial t} + u \cdot \nabla f = S(f). \]  

(Equation 2)

Throughout this paper, the BGK relaxation model

\[ f_i + u \cdot \nabla f = \frac{g - f}{\tau}, \]  

(Equation 3)

will be used to construct the UGKWP method. Here \( \tau \) denotes the relaxation time, which is related to the dynamic viscosity coefficient \( \mu \) and the pressure \( p \), i.e., \( \tau = \mu / p \). The local equilibrium state \( g \) is the Maxwellian distribution

\[ g = \rho \left( \frac{\lambda}{\pi} \right)^{3/2} \exp\left[-\lambda((u - U)^2 + \xi^2)\right], \]  

(Equation 4)

with density \( \rho \), macroscopic velocity \( U \), internal degree of freedom \( K \), and the internal variable \( \xi = (\xi_1, \ldots, \xi_\ell) \). \( \lambda \) is the temperature \( T \) by \( \lambda = m/(2K_0T) = 1/(2RT) \). Here, \( m \) and \( K_0 \) represent the molecular mass and the Boltzmann constant, respectively. \( R = k_\text{B}/m \) is the specific gas constant. Typically, the relaxation parameter in the kinetic model can be calculated through

\[ \tau = \frac{\mu}{\rho} \left( \frac{T_\text{ref}}{T} \right)^\omega, \]  

(Equation 5)

where \( \mu_\text{ref}, T_\text{ref} \) are the reference viscosity coefficient and temperature, and \( \omega \) is power index for the variable hard sphere (VHS) molecules.

### A. Unified gas-kinetic framework

The unified scheme is direct modeling in the discretized space \( \bigcup \Omega_i \subset \mathbb{R}^3 \) and time \( t^\text{ref} \in \mathbb{R}^+ \).\(^8\) The cell averaged conservative flow variables \( W_i = (\rho_i, (\rho U_i), (\rho E_i)) \) on a physical cell \( \Omega_i \) are defined as

\[ W_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} W(x) dx, \]  

(Equation 6)

and the cell averaged distribution function \( f_i \) on physical cell \( \Omega_i \) is defined as

\[ f_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} f(x) dx. \]  

(Equation 7)

The discretized conservation laws for the flow variables and gas distribution function within a time step in the control volume \( \Omega_i \) are

\[ W_{i+1} = W_i^0 - \frac{1}{|\Omega_i|} \sum_{j \in N(i)} F_{ij} |S_{ij}| \]  

(Equation 8)

and

\[ f_{i+1} = f_i^0 - \frac{1}{|\Omega_i|} \sum_{j \in N(i)} F_{ij} |S_{ij}| + \int_{t^0}^{t+1} S(f_j) dt, \]  

(Equation 9)

where \( N(i) \) denotes the set of the interface-adjacent neighboring cells of cell \( i \), and cell \( j \) is one of the neighbors. The interface between cells \( i \) and \( j \) is represented by the subscript \( ij \). Hence, \( |S_{ij}| \) is referred to the area of the interface \( ij \). \( F_{ij} \) and \( F_{ij} \) denote the macroscopic and microscopic fluxes across the interface, respectively. \( |\Omega_i| \) is the volume of cell \( i \), and \( \Delta t = t_i - t_0 \) denotes the numerical time step.

It should be noted that Eqs. (8) and (9) are the fundamental physical laws on the scale of mesh size and time step, which describe the conservations of macroscopic flow variables and microscopic gas distribution function. The macroscopic flux variables, their fluxes, and the flux for the particle transport across the cell interface are related to the moments of the gas distribution function

\[ W_i = \int f_i \psi d\mathbb{E}, \]  

(Equation 10)

\[ F_{ij} = \int_{t^0}^{t+1} \int_{\mathbb{E}} u \cdot n_j f_j(t) \psi d\mathbb{E} dt, \]  

(Equation 11)

and

\[ F_{ij} = \int_{t^0}^{t+1} u \cdot n_j f_j(t) dt, \]  

(Equation 12)

where \( n_j \) is the unit normal vector of the interface \( ij \) pointing from cell \( i \) to cell \( j \) and \( f_j(t) \) is the time-dependent distribution function on the cell interface. \( \psi = (1, u, \frac{1}{2}(u^2 + \xi^2)) \) denotes the collision invariants with \( d\mathbb{E} = dud\xi, du = dud\psi, \) and \( d\xi = d\xi_1 d\xi_2 \ldots d\xi_\ell \). The BGK relaxation term satisfies the compatibility condition

\[ \int S(f) \psi d\mathbb{E} = \int \frac{g - f}{\tau} \psi d\mathbb{E} = 0 \]  

(Equation 13)

for the mass, momentum, and energy conservations in the particle collision process.

The multiscale flow evolution in the unified algorithm relies on the construction of the flux function at the cell interfaces. The time-dependent gas distribution function \( f_j(t) \) for recovering flow physics in different regimes is based on the integral solution of the BGK model

\[ f(x_0, t) = \frac{1}{\tau} \int_{t_0}^{t} \int_{x_0}^{x(t')} g(x', t') e^{-((t-t')/\tau)} dt' \]  

\[ + e^{-((t-t_0)/\tau)} f_0(x_0 - u(t - t_0)), \]  

(Equation 14)

where \( x_0 \) is the point for the evaluation of the local gas distribution function and \( x' = x_0 - u(t - t') \) is the particle trajectory. Typically, \( x_0 \) is denoted as \( x_{ij} \), the center of a cell interface for flux evaluation. \( f_0(x) \) is the initial distribution function around \( x_0 \) at the beginning of each step \( t_0 = t', \) and \( g(x, t) \) is the equilibrium state distributed around \( x_0 \) and \( t_0 \). Specifically, for second-order accuracy, with the definition \( t_0 = 0 \) and \( x_0 = 0 \), the local expansions are

\[ g(x, t) = g_0 + g_\lambda \cdot x + g_\alpha t \]  

(Equation 15)
and

\[ f_0(x) = f_0 + f_k \cdot x. \]  

(16)

The time-dependent distribution function at the center of the cell interface \( x_0 \) can be constructed as

\[ f_j(t) = \frac{c_1 g_0 + c_2 g_x \cdot u + c_3 g_t + c_4 f_k \cdot u}{f_0'(t)} \]

(17)

with the coefficients

\[ c_1 = 1 - e^{-t/\tau}, \]
\[ c_2 = te^{-t/\tau} - \tau (1 - e^{-t/\tau}), \]
\[ c_3 = \tau (1 - e^{-t/\tau}), \]
\[ c_4 = e^{-t/\tau}, \]
\[ c_5 = -te^{-t/\tau}. \]

Note that \( f_j'(t) \) and \( f_j''(t) \) are the terms related to the evolution of the local equilibrium state \( g(x, t) \) and the initial distribution function \( f_0(x) \), respectively. The initial gas distribution function \( f_0 \) in Eq. (17) is reconstructed from the updated gas distribution function at \( f_0 \), which has the form

\[ f_0(x_0, t) = \left\{ \begin{array}{ll} f_0'' + (\nabla x f)_0'' \cdot (x_0 - x) - u \cdot (\nabla x f)_0'' (t - t_0), & n_y \cdot u \geq 0, \\ f_0' + (\nabla x f)_0' \cdot (x_0 - x) - u \cdot (\nabla x f)_0' (t - t_0), & n_y \cdot u < 0, \end{array} \right. \]

(19)

where \( f_0'' \) and \( f_0' \) are the initial distribution functions at neighboring cells around the cell interface \( i \). Here \( (\nabla x f)_0'' \) is the spatial gradient of the initial distribution function inside the cell \( i \) and can be reconstructed via least square with Venkatakrishnan’s limiter or the Barth and Jespersen limiter.\(^{21} \)

The local equilibrium state \( g_0 \) in Eq. (17) is computed from the compatibility condition

\[ W_0 = \int g_0 \psi dE = \int f_0 \psi dE, \]

(20)

and the spatial and temporal derivatives of the equilibrium state can be obtained through the micro-macro relationship

\[ W_x = \int g_x \psi dE = \int f_x \psi dE, \]
\[ W_t = - \int u \cdot g_x \psi dE. \]

(21)

Equations (14) and (17) present a transition process from the initial non-equilibrium distribution function to the equilibrium one with the occurrence of particle collision. It shows a process from the kinetic to the hydrodynamic scale flow evolution, and the specific solution depends on the local parameter \( t/\Delta t \), i.e., the local cell Knudsen number. Specifically, the integrated microscopic flux over a time step gives

\[ \mathcal{F}_y = \int_0^{\Delta t} u \cdot n_y f_0(t) dt \]
\[ = u \cdot n_y (q_1 g_0 + q_2 g_x \cdot u + q_3 g_t) + u \cdot n_y (q_1 f_0 + q_3 f_k \cdot u), \]

(22)

where \( \mathcal{F}_y \) and \( \mathcal{F}_x \) are the equilibrium microscopic flux and the free transport microscopic flux, respectively. Similarly, the macroscopic fluxes for conservative variables are split into the equilibrium flux \( \mathcal{F}_y \) and the free streaming flux \( \mathcal{F}_x \)

\[ \mathcal{F}_y = \int_0^{\Delta t} u \cdot n_y f_0(t) \psi dE dt \]
\[ = \int \mathcal{F}_y \psi dE + \int \mathcal{F}_x \psi dE, \]

(23)

with the coefficients

\[ q_1 = \Delta t - \tau (1 - e^{-t/\tau}), \]
\[ q_2 = 2 \tau (1 - e^{-t/\tau}) - \tau \Delta t - \tau \Delta t e^{-t/\tau}, \]
\[ q_3 = \frac{\Delta t^2}{2} - \tau \Delta t + \tau (1 - e^{-t/\tau}), \]
\[ q_4 = \tau (1 - e^{-t/\tau}), \]
\[ q_5 = \tau \Delta t e^{-t/\tau} - \tau (1 - e^{-t/\tau}). \]

(24)

With the variation of \( t/\Delta t \), Eqs. (22) and (23) can provide multiscale flow evolution solution. When \( \Delta t \gg \tau \), only the terms \( \mathcal{F}_y \) with \( q_1 \approx \Delta t \) and \( q_3 \approx \Delta t^2/2 \) remained for equilibrium wave interaction; when \( \Delta t \ll \tau \), \( \mathcal{F}_x \) with \( q_4 \approx \Delta t \) and \( q_5 \approx \Delta t^2/2 \) are left for non-equilibrium particle free transport.

In deterministic UGKS,\(^{21} \) the cell averaged distribution function \( f_i \) is further discretized in the particle velocity space with discrete velocity points \( u_i \) to capture the non-equilibrium distribution function. Compared with many other DVM with separate particle free streaming and collision, the mesh size and time step in UGKS are not limited by the particle mean free path and collision time due to their coupled evolution solution in the flux function. Moreover, the NS solutions can be obtained automatically by UGKS in the continuum regime even with \( \Delta t \gg \tau \), such as for the laminar boundary layer solution at high Reynolds number. For UGKWP, instead of discretizing the particle velocity space, the particle will be used directly to represent the non-equilibrium gas distribution function.

B. Particle evolution in UGKWP

The integral solution of the kinetic model equation (14) can be rewritten as

\[ f(x, t) = (1 - e^{-t/\tau}) g_0(x, t) + e^{-t/\tau} f_0(x - ut), \]

(25)

where

\[ g_0 = g_0 + \frac{te^{-t/\tau} - \tau}{1 - e^{-t/\tau}} u \cdot \nabla g + \left( \frac{t}{1 - e^{-t/\tau}} - \tau \right) q_1 g. \]

(26)

Equation (25) states that the distribution function at time \( t \) is a combination of the initial distribution function \( f_0 \) and the modified equilibrium state \( g_0 \). The probability for the particle without suffering collision at time \( t \) is \( e^{-t/\tau} \). Otherwise, it will collide with other particles and the post-collision distribution is determined by the distribution \( g_p \). The cumulative distribution for particle free streaming is given by

\[ \mathcal{F}(t) = e^{-t/\tau}. \] 

(27)
A particle $P_i(m_i, x_i, u_i, e_i)$ can be represented by its mass $m_i$, position $x_i$, velocity $u_i$, and internal energy $e_i$. Its free transport time is

$$t_f = \min(-\tau \ln(\eta), \Delta t),$$

where $\eta$ is a random number generated from a uniform distribution on the interval $(0, 1)$, i.e., $\eta \sim U(0, 1)$. Moreover, the location $x^*$ of the particle free transport up to time $t_f$ can be accurately tracked

$$x^*_k = x^*_0 + u_i t_f,$$

where the particle velocity $u_i$ keeps the same value.

According to the time $t_f$ assigned to each particle, these particles with $t_f = \Delta t$ are called collisionless particles $P_i$, and the particles with $t_f < \Delta t$ are called collisional particles $P_i$. The collisional particles $P_{i,k}$ should be deleted at the collision time $t_f$ and re-sampled from distribution function $g_i$ with the updated macroscopic quantities $\mathbf{W}_1$ via sampling, i.e.,

$$x^*_{ik} \sim U(\Omega),$$

$$u^{n+1}_{ik} = \mathbf{g}_i(\mathbf{W}^{n+1}).$$

(30)

The position of the re-sampled particle $x^*_{ik}$ is uniformly distributed inside the cell $\Omega$, where the particle collision happens. Similar to DSMC method, the internal energy $e^{n+1}_{ik}$ is sampled according to the temperature and internal degree of freedom $K$. The particles mass $m_i$ can be prescribed and will be discussed later. The above scheme is called the unified gas-kinetic particle (UGKP) method.

Theoretically, in the next time step, the re-sampled equilibrium particles will be reclassified into collisionless and collisional particles again according to the free transport time $t_f$, and only the collisionless particles will be retained at the end of the next time step. Since the collisional particle will disappear in the next time step, it is not necessary to re-sample it, and its dynamic impact, such as the contribution to the flux, can be calculated analytically. Therefore, in order to reduce the noise variance in near continuum regime and avoid re-sampling collisional particles repeatedly, only the collisionless particles in the hydrodynamic wave $\mathbf{W}^h_{ik} = \mathbf{W}^h - \mathbf{W}^p$ need to be re-sampled at the beginning of each time step. This is the basic idea of the unified gas-kinetic wave-particle (UGKWP) method. Here, $\mathbf{W}^p$ is the total conservative quantities of collisionless particles remained in cell $\Omega_i$ at the end of each time step,

$$\mathbf{W}^p = \frac{1}{\Omega_i} \sum_{x_i \in \Omega_i} \phi_k,$$

(31)

where the vector $\phi_k = m_k (1, u_k, \frac{1}{2} (u_k^2 + e_k))$ denotes the mass, momentum, and energy carried by the particle $P_k$. $\mathbf{W}_i$ is the macroscopic variables updated in cell $i$ through Eq. (8).

Based on the cumulative distribution equation (27), the proportion of the collisionless particles can be evaluated in each cell at the beginning of each time step. The total mass density of the re-sampled collisionless particle takes a portion of the updated hydrodynamic wave density $\rho^h$ from the previous time step

$$\rho^{hp} = e^{-\Delta t/\tau} \rho^h.$$

(32)

Based on this observation, the particle evolution procedure of the UGKWP method can be summarized as

1. Obtain free streaming time $t_{f,k}$ for the remaining particles $\rho^{p,k}$.
2. Sample the collisionless particles $P^{\rho,k}$ from hydrodynamic wave with distribution $g_i^{(h)}$.
3. Obtain the collisionless particles with total mass density $\rho^{hp,n} = e^{-\Delta t/\tau} \rho^{h,n}$ have the free streaming time $t_f = \Delta t$.
4. Stream all the particles and classify into two categories, i.e., collisionless particles $P^{\rho,k}$ and collisional particles $P^{\rho,k}$.
5. Keep collisionless particles $P^{\rho,k}$ and remove collisional particles $P^{\rho,k}$. Calculate the total conservative quantities of the remained collisionless particles $\mathbf{W}^{p,n+1}$ according to Eq. (31). The conservative quantities of collisional particles $\mathbf{W}^{h,n+1}$ are obtained from the updated total conservative quantities $\mathbf{W}^{n+1}$ in Eq. (8) as $\mathbf{W}^{h,n+1} = \mathbf{W}^{n+1} - \mathbf{W}^{p,n+1}$. The detailed formulation for the update of $\mathbf{W}^{n+1}$ will be presented in Subsection II C.

The interplay of waves (collisional) and particles (collisionless) in the UGKWP method is illustrated through a series of figures in Fig. 1. From the diagram, the multi-efficiency property of UGKWP is clearly indicated, i.e., the computational efficiency of UGKWP gets to the high efficient approach in the corresponding regime. For example, in the near continuum regime, i.e., $\tau \to 0$, the proportion of collisionless particle decreases exponentially. The UGKWP becomes a scheme without particles, and its computational cost is comparable to a traditional NS solver. On the other hand, for highly non-equilibrium hypersonic flow, such as $\tau \gg \Delta t$, the particles will play a dominant role to capture the non-equilibrium transport, and the efficiency of the scheme will go to the particle method, such as DSMC.

It has been shown in Ref. 22 that the UGKWP method is a kinetic equation solver in the rarefied regime and preserves the Navier–Stokes solution in the continuum regime. Even though the particles are sampled uniformly inside the control volume, the spatial accuracy can be still kept in the near continuum regime, because the portion of particles $e^{-\Delta t/\tau}$ is minimal and the hydrodynamic wave evolution is dominant by the updated $\mathbf{W}$, which is computed analytically with second-order accuracy. The DSMC method requires $\Delta t$ to be less than the particle mean collision time, which is equivalent to $t_f = \Delta t$ for all particles in its subsequent transport step. The free transport time in the UGKWP method is obtained from the sampling process in Eq. (28), where the particle collisional effect, such as evolving to the equilibrium distribution $g_i$, has been modeled in the scheme through the evolution solution in (25). Without using this evolution solution with time accumulating particle collision effect, or any other equivalent form, it is impossible to design a multiscale method, which can recover the NS solution in the continuum flow regime, and capture the solution in the transition regime faithfully.

C. Macroscopic variables update

The UGKWP updates the macroscopic variables in each control volume through Eq. (8). The equilibrium part flux $\mathbf{F}^{eq}$ is directly calculated from the macroscopic flow field as given by Eq. (23). In UGKWP, the calculation of the free streaming flux $\mathbf{F}^{hp}$ will be divided into two parts. The free streaming flux from collisional hydro-

Phys. Fluids 32, 096108 (2020); doi: 10.1063/5.0021199
Published under license by AIP Publishing
namic waves of $(1 - e^{-\Delta t/\tau})W^h$ can be calculated analytically. The free streaming flux from collisionless particles of hydrodynamic waves $e^{-\Delta t/\tau}W^h$ and the remaining particles $W^p$ from the previous time step can be evaluated by counting the particles passing through the cell interface during a time step. The free streaming flux contributed from the collisional hydrodynamic waves of $(1 - e^{-\Delta t/\tau})W^h$ on the cell interface $ij$ is

$$\mathbf{F}^{fr}_{ij} = \mathbf{F}^{fr, UGKS}_{ij}(W^h) - \mathbf{F}^{fr, DM}_{ij}(W^p)$$

$$= \int \mathbf{u} \cdot \mathbf{n} \left[ (q^h_\phi + q^h_\psi) - e^{-\Delta t/\tau} \right] \left( g^h_\phi - t \mathbf{u} \cdot \mathbf{g}^h_\phi \right) d\mathbf{x} d\mathbf{y}$$

$$= \int \mathbf{u} \cdot \mathbf{n} \left[ (q^h_\phi - \Delta t e^{-\Delta t/\tau} g^h_\phi) \right] + \left( q^h_\psi + \frac{\Delta t^2}{2} e^{-\Delta t/\tau} \right) \mathbf{u} \cdot \mathbf{g}^h_\phi \right] d\mathbf{x} d\mathbf{y},$$

where $g^h_\phi$ is the Maxwellian distribution with temperature and average velocity determined by total macroscopic variables $W$, but the density by $W^h$, $g^h_\psi$ is the spatial derivative of the Maxwellian distribution, which can be obtained from the reconstruction of $W$ and $W^h$. The free streaming flux $\mathbf{F}^{fr}_{ij}$ in Eq. (23) is computed partially by particles and partially by the contribution of $g^h_\phi(W^h)$ analytically. In addition, the subtraction of $\mathbf{F}^{fr, DM}_{ij}(W^p)$ from $\mathbf{F}^{fr, UGKS}_{ij}(W^h)$ aims to remove the free transport fluxes that are still calculated by the collisionless particle $P_0$ sampled from $W^{hp}$. The total non-equilibrium free streaming flux $\mathbf{F}^{fr}_{ij}$ also includes the contribution from the remaining particles $P_1$ from the previous time step. During the free transport process, the contribution to the numerical fluxes of cell $i$ can be obtained by counting the particles at the beginning and at the end of the time step inside the cell $i$.

$$\mathbf{F}^{fr}_{ij} = \frac{1}{\Omega_i} \sum_{\phi} \phi^{k}_{ij} - \frac{1}{\Omega_i} \sum_{\phi} \phi^{k}_{ij}.$$  

Finally, the updates of the conservative flow variables in the UGKWP method are

$$W_i^{n+1} = W_i^n - \frac{1}{|\Omega_i|} \sum_{j \in N(i)} \mathbf{F}^{eq}_{ij} \cdot \mathbf{S}_{ij} - \frac{1}{|\Omega_i|} \sum_{j \in N(i)} \mathbf{F}^{fr}_{ij} \cdot \mathbf{S}_{ij}.$$

$$= W_i^n - \frac{1}{|\Omega_i|} \sum_{j \in N(i)} \mathbf{F}^{fr, wave}_{ij} \cdot \mathbf{S}_{ij} + \frac{\mathbf{F}^{fr, p}_{ij} \cdot \mathbf{S}_{ij}}{|\Omega_i|},$$

(D55)

### D. Miscellaneous details

1. **Time step on an unstructured mesh**

Following the implementation in Ref. 33, the time step for an unsteady flow simulation is obtained from

$$\Delta t = \min_i \frac{\Omega_i}{\Lambda_i^1 + \Lambda_i^2 + \Lambda_i^3},$$

with the Courant number $0 < C < 1$ and convective spectral radii of cell $i$

$$\Lambda_i^1 = (|U_i| + c) \Delta S_i^z,$$

$$\Lambda_i^2 = (|V_i| + c) \Delta S_i^y,$$

$$\Lambda_i^3 = (|W_i| + c) \Delta S_i^x,$$

where $c = 3\sigma = 3\sqrt{RT_i}$ is approximately the sound speed and $(U_i, V_i, W_i)$ is the macroscopic velocity. The variables $\Delta S_i^z$, $\Delta S_i^y$, and $\Delta S_i^x$, respectively, represent projections of the control volume on the $y$-$z$-, $x$-$z$-, and $x$-$y$-planes, which are given by

$$\Delta S_i^z = \frac{1}{2} \sum_{j \in N(i)} |S_i^z|,$$

(38)

$$\Delta S_i^y = \frac{1}{2} \sum_{j \in N(i)} |S_i^y|,$$

$$\Delta S_i^x = \frac{1}{2} \sum_{j \in N(i)} |S_i^x|,$$
where \( S_x, S_y, \) and \( S_z \) denote the \( x, y, \) and \( z \)-components of the face vector \( S = |S|n_x. \)

2. Particle sampling

At the beginning of each time step, the collisionless particles of hydrodynamic waves will be sampled in pairs from the Maxwellian distribution function \( g(W) \). Specifically, given the macroscopic velocity \( U = (U, V, W) \), temperature \( T \), and vector \( X \), a pair of particles with microscopic velocities \( u = U + \sqrt{RT}X \) and \( u' = U - \sqrt{RT}X \) will be sampled from the normal distribution. To determine the sampling particle number \( N_{sam} \) in the cell, a prescribed preference number \( N_{ref} \) is required. Further, the reference mass \( m_{ref} \) can be determined from the total particle mass and the reference number \( N_{ref} \),

\[
m_{ref} = \frac{(\rho^p + \rho^{hp})|Ω|}{N_{ref}} = \frac{(\rho^p + e^{-\Lambda t/\tau} \rho^{hp})|Ω|}{N_{ref}},
\]

(39)

Once the reference mass \( m_{ref} \) is available, the number of particles to be sampled symmetrically (antithetic variates method for variance reduction) is determined by

\[
N_{sam} = \left\lceil \frac{\rho^{hp}|Ω|}{2m_{ref}} \right\rceil = \left\lceil \frac{e^{-\Lambda t/\tau} \rho^{hp}|Ω|}{2m_{ref}} \right\rceil,
\]

(40)

If the reference mass \( m_{ref} \) is the same for all cells, then the total number of particles per cell would be exactly equal to \( N_{ref} \). In this way, the total number of particles in each cell can be controlled around the given reference number \( N_{ref} \) in the near continuum regime regardless of mesh distribution. Moreover, the minimum number of particles \( N_{min} \) per cell can be prescribed to adjust the sampled particles’ number such that

\[
N_{sam} = \max\{N_{sam}, N_{min} - N_{left}\},
\]

(41)

where \( N_{left} \) is the collisionless particles that remained at the beginning of each time step. Finally, the sampled mass weight \( m_{sam} \) for each sampled particle is

\[
m_{sam} = \frac{\rho^{hp}|Ω|}{N_{sam}} = \frac{e^{-\Lambda t/\tau} \rho^{hp}|Ω|}{N_{sam}},
\]

(42)

which guarantees that the total sampled mass is exactly equal to \( \rho^{hp}|Ω| \).

As for the determination of the simulation parameter \( N_{ref} \), it depends on the Knudsen number \( Kn \) and the desired accuracy. As pointed out in Ref. 34, the root mean square error (RMSE) in the Monte Carlo integration is \( E[|κ|^2]^{1/2} = \sigma N^{-1/2} \), which is of the size \( O(N^{1/2}) \) with a constant \( \sigma \) that is just the variance of the integrand \( f \). In the UGKWP method, under the same numerical setup, the noise reduces as the square root of the increasing number of particles.

Furthermore, the variance in the RMSE is no longer constant. It decreases as \( e^{-\Lambda t/\tau} \) since the portion of the particle decreases as \( e^{-\Lambda t/\tau} \) in the UGKWP method, which distinguishes the UGKWP method from other stochastic particle methods based on kinetic model equations.\(^{23,26} \)

Generally speaking, for a three-dimensional steady-state problem, the \( N_{ref} \) can range from a few hundred to a few thousand particles per cell. As the quantities of second-order moments like temperature are not emphasized, dozens of particles per cell are enough.

3. Time-averaging

For the steady-state solution, the flow field \( \bar{W} \) starts to be averaged after a given time step \( N_{avg} \),

\[
\bar{W} = \frac{\sum_{n=0}^{N_{avg}} \Delta t^n \bar{W}^n}{\sum_{n=0}^{N_{avg}} \Delta t^n},
\]

(43)

where \( \Delta t^n = t^n - t^{n-1} \). The averaged flow field \( \bar{W} \) is assumed to be convergent if the relative change in two successive steps is less than a given tolerance, such as \( \epsilon = 10^{-8} \). Then, the flow variables, such as the temperature \( \bar{T} \) and the macroscopic velocity \( \bar{U} \), can be obtained from the averaged conservative flow variables \( \bar{W} \).

4. Numerical dissipation

The UGKWP targets the continuum and rarefied flow. In the continuum flow regime, the strong shock structure is usually unresolved by the mesh size. Therefore, numerical dissipation is added through an enhanced relaxation time to enlarge the shock thickness to the mesh size scale,

\[
\tau_{num} = \frac{\mu}{\bar{p}} + C_2 |P_1 - P_2| |P_1 + P_2| \Delta t,
\]

(44)

where \( P_1 \) and \( P_2 \) are the reconstructed pressures at the left and right side of the cell interface and \( C_2 \) is a constant, such as \( C_2 = 10 \) for strong shock in the continuum regime.

5. Boundary condition

The proper treatment of the boundary condition is crucial for a numerical scheme. For a diffusive wall condition with a normal direction \( n \) pointing toward the inside of the computational domain, the incoming distribution function \( f_{in}(t) \) at the boundary is given by Eq. (17). The distribution function of the emitted particles from the wall has a Maxwellian distribution

\[
g_w = \rho_w \left( \frac{1}{2\pi RT_w} \right)^{3/2} \exp \left( \frac{-(u - U_w)^2 + v^2}{2RT_w} \right),
\]

(45)

where \( T_w \) and \( U_w \) are the prescribed wall temperature and velocity. Based on the non-penetration condition, \( \rho_w \) in the above Maxwellian is given by

\[
\int_0^{\Delta t} \int_{n_0}^{n_0} \int_{n_0}^{n_0} \mathbf{n} \cdot (u - U_w) f_{in}(t) dt du dv = \Delta t \int_{n_0}^{n_0} \mathbf{n} \cdot (u - U_w) g_w du.
\]

(46)

III. 3D UGKWP CODE AND PARALLELIZATION

The UGKWP solver is constructed under a finite volume framework on a 3D unstructured mesh. It includes not only the reconstruction and flux evaluation module as in the traditional finite volume solver, but also the particle sampling and tracking module as in the pure particle method.
A. Structure of the solver

The main components of the UGKWP solver are sketched in Fig. 2. As shown in the diagram, the program starts with the pre- and post-processor module, where the mesh partition, initialization, setup of boundary condition, and parallel input/output (I/O) are handled inside. In the UGKWP solver, the numerical procedures are organized into a macroscopic field level and a microscopic particle level. Accordingly, the macroscopic components surrounded by the blue dashed line consist of the parallel data transfer, reconstruction of the macroscopic variable gradient, and macroscopic flux calculation with boundary treatment. The green dashed block contains the components for microscopic particles that are stored in the doubly linked list. There are frequent operations, such as tracking particles and calculating the macroscopic fluxes. The insert/delete operation is efficient in the scenario of parallel transfer and sampling/elimination of the particles.

B. Parallelization

The parallelization of the current code adopts Message Passing Interface (MPI) based on the physical mesh decomposition. Every MPI process deals with a non-overlapping sub-domain, and the information like conservative variables and particles are communicated with the neighboring domain through corresponding boundaries. Since the macroscopic solver has second-order accuracy, no padding area of sub-domain is required.

The code has been tested on Tianhe-2, located in the National Supercomputer Center in Guangzhou, China. Tianhe-2 contains 16 000 nodes each of which possesses one Intel® Xeon® E5-2692 12 Cores at 2.2 GHz central processing unit (CPU) and 88 gigabytes of memory (64 used by the Ivy Bridge processors, and 8 gigabytes for each of the Xeon Phi processors). The computing nodes of Tianhe-2 are interconnected by the TH Express-2 network. To test the parallel efficiency and scalability of the UGKWP, the code is compiled using Intel C/C++ compiler of version 18.0.0 with -O3 optimization flag, and are linked to the MPICH2 with a customized GLEX channel. Since the scaling is problem-specific (depending on the Knudsen number and the preference number of particles per cells \(N_{ref}\)), the factors affecting the performance will be analyzed in a series of three-dimensional lid-driven cavity flow tests.

1. Macroscopic field computation

First, only the parallel speedup of pure macroscopic field computation using different MPI processes is measured to eliminate the computation of particles and communication of particle parallel transfer. Without the involvement of particle generation and particle transportation, the UGKWP degenerates to the gas-kinetic scheme (GKS) for the continuum flow computation.\(^{37}\) As it becomes a deterministic flow solver, for simplicity, the Knudsen number is fixed at \(10^{-4}\) in the following parallel computation. The averaged running time (wall clock time) of a single iteration step is measured, and the measurement is ensured to be over 100 s, and no I/O time is counted.

To investigate Amdahl’s law (strong scaling) at different fixed problem sizes, the physical domain is discretized as \(D^3\), where \(D\) is the number of cells along each direction with the values 64, 128, and 256 separately. To test Gustafson’s law (weak scaling), we concern the speedup for a scaled problem size to the number of processors. Hence, \(D = 64\) on one node with 24 cores is chosen as the baseline, i.e., keeping the number of cells per processors as \(64^3/24\), and the grid size increased simultaneously as the increment of the number

![FIG. 2. The structure and main components of the UGKWP solver.](image-url)
of processors $P$. The corresponding speedup is measured based on the averaged single-node simulation time, i.e., $S_P = 24 T_p / T_{24}$.

Both strong and weak scaling analyses are plotted in Fig. 3. The solid red line represents the ideal linear speedup. From the diagram, the overall computational time of the various physical grid sizes scales well with the number of processing cores (or MPI processes). Although strong scaling is very sensitive toward the serial fraction of the program and the communication overhead (e.g., synchronization) could further degrade the performance, the worst efficiency still has 73.8% for the case $D = 64$ with $P = 960$ number of processors. Moreover, it is observed that the strong scaling performance increases considerably with the increase in grid size. The strong scaling parallel efficiency for the largest problem size $D = 256$ can reach 85.4%, despite the usage of $P = 1536$ number of processing cores. Finally, the weak scaling is also verified by increasing both the job size and the number of processing cores, and a satisfactory weak scaling efficiency up to 92% has been achieved even with $P = 1536$ number of processing cores.

### 2. Involvement of microscopic particles

Next, the particles are included in the experiment to explore the scalability and parallel efficiency of the implementation. Problems with different numbers of cells $D^3$, Knudsen number $Kn$, and preference number of particles per cell $N_{ref}$ are run across different nodes. The maximum problem size is limited by the total memory available on each node, and no I/O time is recorded. The average computing time (wall clock time) for a single iteration step is averaged from 1001 to 1100 steps to ensure sufficient running time and reach a steady-state solution.

The averaged wall-clock time (second) per step against the number of cores is shown in Fig. 4. The results indicate that the scaling performance is good for all simulations because the wall-clock time decreases almost linearly as the increase in the number of processing cores. Actually, from Tables I and II, the parallel efficiency $E_P = S_P / P$ for cases with $D \geq 72$ is over 86% even with $P = 864$ number of processing cores.

Furthermore, several interesting patterns have been observed. First, for the cases of both $N_{ref} = 50$ and $N_{ref} = 500$, we can observe that the wall-clock time raises not only as of the increment of grid size but also as of the Knudsen number. The reason is that the mean free path of the particles becomes large at a high Knudsen number, which produces the unbalanced distribution of particles among different sub-domains as well as in the processing cores. Secondly, the scaling performance deteriorates as the shrinkage of grid size, especially in high-Knudsen number cases, because the proportion of communication time would increase as the declining number of cells per core, and the uneven effect of distributing particles among
Processing cores would also be amplified. Accordingly, the worst parallel efficiency $E_{864} = 68.7\%$ is observed for the case $D = 36$ at $Kn = 1$ with respect to $P = 864$ number of cores. Lastly, the parallel efficiency would increase as the reference number of particles becomes larger.

Another interesting phenomenon is that the maximum parallel efficiency can be greater than one. For instance, the maximum parallel efficiency observed is $E_{864} = 123.2\%$ and achieved at the cases of $D = 72$, $Kn = 10^{-2}$ and $N_{ref} = 500$ with $P = 864$ cores. Actually, the parallel efficiency in transition regime $Kn = 10^{-2}$ is even higher than that in the near continuum regime $Kn = 10^{-4}$ with the same grid size $D$ and the reference number of particles $N_{ref}$. Besides, the worst parallel efficiency is even larger than 86.3% for all cases at $Kn = 10^{-4}$, $10^{-2}$. This counterintuitive parallel efficiency in the transition regime is probably due to the doubly linked list data structure for storing particles. In the transition or near continuum regime, the collision between particles is intensive, and the frequent elimination/resampling of particles involves frequent deleting/inserting operation in memory. Nonetheless, the bottleneck caused by the implemented data structure can be alleviated through the replacement of a sequence container, like the STL vector.

In summary, intensive parallel tests of cavity flow show a satisfactory strong and weak scaling performance of the UGKWP code. The current implementation becomes a valuable tool for simulating complex flow problems across thousands of processing cores in parallel computation. The actual parallel efficiency might vary for a particular simulation setup.

**IV. NUMERICAL EXAMPLES**

In this section, the accuracy and the computational efficiency of the UGKWP solver will be evaluated through many test cases with a wide range of Knudsen and Mach numbers. The numerical Sod shock tube problem in 3D, lid-driven cubic cavity flow, high-speed flow passing through a cube, and the flow around a space vehicle, are tested. The results are compared with those from UGKS/DUGKS and DSMC. Without a special statement, the diffusive boundary condition with full accommodation is applied for the isothermal walls. The code is compiled with GCC version 7.5.0, and all computations are carried out on a workstation with (Dual CPU) Intel Xeon Platinum 8168 at 2.70 GHz with 48 cores and 270 GB memory unless indicated otherwise.

**A. Sod shock tube inside a square-column**

The Sod shock tube problem is simulated inside a square-column for diatomic gas at different Knudsen numbers to validate the current UGKWP method, and the result is compared with the 1D UGKS solution.

In this test case, the following non-dimensionalization is used:

\[
\hat{\rho} = \frac{\rho}{\rho_\infty}, \quad \hat{U} = \frac{U}{C_\infty}, \quad \hat{V} = \frac{V}{C_\infty}, \quad \hat{W} = \frac{W}{C_\infty}, \quad \hat{T} = \frac{T}{T_\infty}, \quad \hat{P} = \frac{P}{\rho_\infty C_\infty^2},
\]

\[
\hat{t} = \frac{t}{t_\infty}, \quad \hat{x} = \frac{x}{L}, \quad C_\infty = \sqrt{\frac{2kB T_\infty}{m}}, \quad t_\infty = \frac{L}{C_\infty},
\]
and the initial condition for the non-dimensional variables is
\[
(\hat{\rho}, \hat{U}, \hat{V}, \hat{W}) = \begin{cases} 
(1, 0, 0, 0.1), & 0 < \hat{x} < 0.5 \\
(0.125, 0, 0, 0.1), & 0.5 < \hat{x} < 1.
\end{cases}
\] (47)

For the UGKWP simulation, the physical domain is a \([0, 1] \times [−0.1, 0.1] \times [−0.1, 0.1]\) square-column tube, which is discretized by 100 × 5 × 5 uniform mesh points. The preset reference number of particles are \(N_{ref} = 200, 400, 1000, 2000, 3200, 3200\) for the cases at \(Kn = 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10\), respectively. Least square reconstruction with the Venkatakrishnan limiter is utilized. For the UGKS simulation, the 1D physical domain \([0, 1]\) is discretized uniformly with 100 cells. Composed Newton–Cotes quadrature with 101 velocity points in the range \([-6, 6]\) is fixed to discretize the one-dimensional velocity space. The van Leer limiter is used for the reconstruction of both conservative variables and discrete distribution function. The left and right boundaries are treated as far-field, and the others are treated as symmetric planes. The Courant–Friedrichs–Lewy (CFL) number for both UGKWP and UGKS simulations is 0.9, and the reference viscosity is given in Eq. (5) with \(\omega = 0.74\). The results at time \(t = 0.12\) in all flow regimes are presented.

The density, velocity, and temperature obtained by the UGKS and the UGKWP methods at different Knudsen numbers are plotted in Figs. 5–10, where the three-dimensional flow field computed by UGKWP is projected to the x-direction by taking the ensemble average over the cells on y–z plane. No time-averaging is applied, and the statistical noise is satisfactory for this unsteady flow simulation. For all cases in different flow regimes, the 3D UGKWP solutions agree well with the 1D UGKS data. The slight difference is due to different limiters, i.e., the van Leer limiter for UGKS and the Venkatakrishnan limiter for UGKWP. The capability of the UGKWP method for numerical simulations in both continuum and rarefied regimes is confirmed.

The distinguishable feature of multi-efficiency\(^{22}\) can be demonstrated here as well. For UGKS, the computational costs for all Knudsen number cases are almost the same since its discretization of the particle velocity space is the same. While for the UGKWP method, the computational cost is reduced at small Knudsen number, e.g., \(Kn = 10^{-4}\). In the near continuum regime, the hydrodynamic wave is dominant, and few particles are sampled and tracked. The computational cost of UGKWP for 3D simulation is admissible because only a few hundred or thousand particles are enough to adaptively discretize the velocity space, whereas it becomes possible that 101\(^7\) mesh points in the velocity space may be required in DVM-based UGKS for the high speed flow. For the steady-state simulation, the number of particles can be reduced further since the statistical noise can be eliminated through the temporal ensemble.

### B. Lid-driven cubic cavity flow

For the low-speed flow, the UGKWP method is used to study the three-dimensional lid-driven cubic cavity flow in the transition regime, and the results are compared with the solutions predicted by dugksFoam.\(^{21}\)

The side length of the cubic cavity is \(L = 1\) m with the computational domain \([0, 1] \times [0, 1] \times [0, 1]\), which is divided non-uniformly into 40\(^3\) hexahedrons with the cell size gradually increased toward the cavity center. The ratio of the cell size in the center and the boundary is about 2. The lid (top boundary) of the cavity moves in the positive x-direction with a constant velocity \(U_w = 50\) m/s, while the other walls are kept fixed. All sidewalls have the diffusive boundary condition and keep a uniform temperature \(T_w = 273\) K. The cavity is assumed to consist of monatomic argon gas with a molecular mass \(m = 6.63 \times 10^{-26}\) kg and diameter \(d = 4.17 \times 10^{-10}\) m. The Knudsen number is \(Kn = \lambda/L = 0.075\), where the mean free path \(\lambda\) is calculated from the initial uniform gas density by \(\lambda = m/(\sqrt{2}\pi d^2 \rho)\). The gas viscosity depends on the temperature by Eq. (5) with the reference temperature \(T_{ref} = T_w = 273\) K and the reference viscosity \(\mu_{ref}\) given by variable hard sphere (VHS) model with \(\omega = 0.81\).

Since it is a low-speed flow with a small temperature variance, \(N_{ref} = 5000\) reference number of the simulation particles is used. The time-averaging is starting from 1000 steps in order to reduce the statistical noises of high moments quantities, such as the temperature. The CFL number is set to be 0.95, and the least square reconstruction with the Venkatakrishnan limiter is employed for the gradient...
FIG. 6. Sod shock tube at $Kn = 10^{-3}$. (a) Density, (b) X-velocity $U$, and (c) temperature.

FIG. 7. Sod shock tube at $Kn = 10^{-2}$. (a) Density, (b) X-velocity $U$, and (c) temperature.

FIG. 8. Sod shock tube at $Kn = 0.1$. (a) Density, (b) X-velocity $U$, and (c) temperature.
calculation. Physical space parallelization with 48 cores is adopted for UGKWP.

In the dugksFoam simulation, the three-dimensional velocity space is discretized using 28 half-range Gauss–Hermit quadrature points in each direction. The CFL number is set to be 0.8. The gradients are calculated by the least square method. The Prandtl number is fixed as $Pr = 1.0$ in the DUGKS simulation to eliminate the model difference since the BGK model is used in the current UGKWP. The velocity space decomposition approach is adopted for dugksFoam with 48 cores on the same machine.

Figure 11 presents the temperature iso-surfaces predicted by UGKWP and dugksFoam. Even though the UGKWP solution exhibits a strong fluctuation, the two results agree well in general. To compare the solutions more precisely, the contours on the symmetric X–Z plane are shown in Fig. 12, where the low order quantities between these two schemes, such as density and x- and y-direction velocity ($U$ and $V$), match well. For the temperature, as a high moment quantity, the UGKWP solutions generally agree with that of dugksFoam, but still exhibit relatively large statistical noise, although a long time-averaging has been performed in UGKWP. It is also noteworthy that the noise incurred by three-dimensional particles in a real three-dimensional simulation is larger than that in the two-dimensional simulation with particles without the Z-direction velocity, e.g., 2D cavity flow.

The computational time for dugksFoam is around 154.1 h with 5000 iterations to reach a velocity residual of $2.4 \times 10^{-7}$. The UGKWP solution takes 93.2 h, including 23,000 steps of averaging. The total memory consumption of dugksFoam reaches 205 GB, whereas of UGKWP is 70.1 GB. For the low-speed flow calculation in the transition regime, the UGKWP method is as expensive as the explicit DUGKS. Techniques such as low variance DSMC 

![FIG. 9. Sod shock tube at Kn = 1. (a) Density, (b) X-velocity U, and (c) temperature.](image1)

![FIG. 10. Sod shock tube at Kn = 10. (a) Density, (b) X-velocity U, and (c) temperature.](image2)
FIG. 11. Comparison of the temperature iso-surfaces predicted by DUGKS (left) and UGKWP (right).

FIG. 12. Symmetric X–Z cut-plane contour of cavity flow at Kn = 0.075. Background: UGKWP; black lines with label: dugksFoam. (a) Density contour, (b) temperature contour, (c) U-velocity contour, (d) W-velocity contour.
C. Flow passing through a cube in rarefied and transition regimes

1. Supersonic flow in rarefied regime

The first case is a supersonic rarefied gas flow passing through a cube at $Ma = 2$ and $Kn = 1$. The cube center is located at (0, 0, 0), and the cube volume is $1 \text{ m}^3$. The surfaces of the cube are diffusive wall boundary condition with a constant temperature $T_w = 273 \text{ K}$. Due to the symmetry, only a quadrant of the cube is simulated by UGKWP. The computational domain $[−6, 8] \times [0, 8] \times [0, 8]$ is discretized by $(32 + 14 + 34) \times (7 + 34) \times (7 + 34) = 80 \times 41 \times 41$ cells with uniformly distributed grids on the surface of the cube. The cell size is stretched from the cube surface with a ratio of 1.0764 up to the front side and a ratio of 1.083 at the rear and lateral sides of the cube. The inflow is monatomic argon gas with molecular mass $m = 6.63 \times 10^{-26} \text{ kg}$ and diameter $d = 4.17 \times 10^{-10} \text{ m}$. The CFL number for UGKWP simulation is 0.9, and the reference viscosity is given by the variable hard sphere (VHS) model with $\omega = 0.81$. To capture the lowest temperature that appears at the rear of the cube caused by the expanding flow, a large number of particles $N_{ref} = N_{min} = 5000$ is required. The simulation is carried out with first-order spatial accuracy to reduce the noises caused by unreliable wave reconstruction. The time-averaging starts from 1400 steps with an initial field computed by 1000 steps GKS. The simulation runs 120.8 h with 48 cores and consumes 183 GB memory, including 8600 steps of averaging.

Figure 13 presents the temperature, density, and velocity distributions on the $X$–$Z$ symmetry plane that are compared with the benchmark DSMC result, which is obtained using dsmcFoam. Similar to UGKWP simulation, a quadrant of the cube is simulated in DSMC with a much finer physical grid of $191 \times 91 \times 91$. Each cell has 50 particles on average, and the time step is $2.0 \times 10^{-7} \text{ s}$. The averaging begins from 1000 steps and continues for 68 000 steps, which takes 128.5 h on 128 CPU cores [Xeon E5-2680v3 (Haswell) at 2.5 GHz]. The results have a satisfactory agreement overall, especially the flow field near the cube wall. However, regarding temperature contour, visible differences can be observed at the front of the bow shock and at the rear part of the cube. The differences come from different kinetic models in UGKWP and DSMC. The UGKWP uses the BGK model and the DSMC solves the full Boltzmann collision term.

**FIG. 13.** Comparison of distributions between UGKWP and DSMC on the $X$–$Z$ symmetric cut-plane at $Ma = 2$ and $Kn = 1$. Dashed red lines with colored background represent the UGKWP result, and the solid white lines denote the DSMC solution. (a) Temperature contour, (b) density contour, (c) contour of $U$ ($X$-component velocity), and (d) contour of $W$ ($Z$-component velocity).
Another notable point is that UGKWP requests at least 5000 particles per cell in the rear part of the cube to get the temperature field at such a low-density region. In contrast, using roughly 500 particles per cell in UGKWP is enough to capture the non-equilibrium shock structure in front of the cube. The reason is that in the case of $\tau \gg \Delta t$, the simulation particle increases, and the stochastic noise becomes significant. Instead of choosing collision pairs in DSMC, UGKWP re-samples the collisional particles according to the cell averaged temperature and macroscopic velocity. When the temperature has a small variance, such as in the low-speed cavity flow, the inadequate particle number could deteriorate the temperature with noise. Then, the re-sampled collisional particles even inherit the inaccuracy and poison the low-temperature area at the rear part of the cube, where artificial heating with over-estimated temperature appears. Even with the above weakness, UGKWP can perform simulation on a coarse mesh than that used in the DSMC. Consequently, UGKWP and DSMC have comparable computational cost in the rarefied regime.

For DVM method, the implicit discretization with the memory reduction technique on GPU is implemented. The full cube is simulated with a physical grid $191 \times 181 \times 181$ and a velocity grid $48^3$. The velocity points are distributed uniformly to cover a range of $[-4\sqrt{2RT_w}, 4\sqrt{2RT_w}]^3$, and the trapezoidal rule is used to calculate the moments. The simulation takes $\sim 20$ h, with 41 iteration steps on the Tesla K40 GPU. Figure 14 shows the detailed comparisons of the temperature, density, and velocity distributions on the $X$–$Z$ symmetry plane with DVM solutions. Similar to the comparison with DSMC, the shock thickness and the separation distance between the density and temperature profiles have small variations between UGKWP and implicit DVM solutions. While the Shakhov model is used in DVM, the UGKWP uses the BGK model. The differences in the shock structure solution between the BGK and the Shakhov model have been presented in Ref. 42. As for computational time, owing to the implicit treatment and implementation of GPU acceleration, the DVM simulation is one order faster than the current UGKWP simulation at such a low Mach number case.

Like UGKS, more realistic models, such as the Shakhov and the full Boltzmann collision term, can be used in the construction of the UGKWP method via the implementation of importance sampling technique. The research in this direction is under investigation. The result with incorporating the Shakhov model into the

![Temperature Contour](image1.png)

![Density Contour](image2.png)

![Component Velocity U](image3.png)

![Component Velocity W](image4.png)

**FIG. 14.** Comparison of distributions between UGKWP and DVM on the X–Z symmetric cut-plane at $\text{Ma} = 2$ and $\text{Kn} = 1$. Dashed red lines with colored background represent the UGKWP result, and the solid white lines denote the DVM solution. (a) Temperature contour, (b) density contour, (c) contour of $U$ (X-component velocity), and (d) contour of $W$ (Z-component velocity).
UGKWP method is given in Fig. 15, which shows the temperature distributions on the X–Z symmetry plane at $Pr = 2/3$ in comparison with the DSMC and the implicit DVM results. Even though both UGKWP and DVM are based on an identical Shakhov model, UGKWP solution matches better with the result of DSMC rather than DVM. This surprising observation may come from the particle nature of UGKWP, which is closer to the DSMC in the rarefied regime.

2. Hypersonic flow in transition regime

To highlight the efficiency and capability of UGKWP at hypersonic flow study, a $Ma = 20$ flow is simulated in the transition regime ($Kn = 0.05$). All the parameters are the same as the previous case except that the inflow temperature is 56 K and the wall temperature is 300 K. The cube is contained in a volume with base and top side lengths of $a = 16$ m, $b = 10$ m and height $h = 14$ m. As shown in Fig. 16, an unstructured mesh with 420 702 cells is generated, which is composed of 8305 hexahedra, 52 prisms, 25 030 pyramids, and 387 315 tetrahedra with a minimum cell height of 0.0248 m near the cube wall. Distinguishable from the rarefied case, the number of particles required drops dramatically. Here, the reference and minimum number of particles per cell $N_{ref} = N_{min} = 400$ are used. The simulation is conducted with the least square reconstruction and Barth and Jespersen limiter. An initial field is first computed with 5000 steps by GKS, and after 8000 steps of UGKWP calculation, the time-averaging of the flow field starts for the steady-state solution. The simulation runs 59.9 h with 48 cores and consumes 68.1 GB memory, including 7000 steps of averaging.

Figure 17 shows the distributions of temperature, density, and velocities on the X–Z symmetry plane. The hypersonic flow computation in the transition regime is a challenge for both stochastic and deterministic methods. For the DSMC, an extremely fine mesh in physical space is required. For the DVM-type deterministic solvers, a tremendous amount of discrete velocity points becomes necessary. The UGKWP is an idealized method for the hypersonic flow in all flow regimes. Due to the high Mach number, even with a low inflow temperature of 56 K, the maximum temperature inside the shock region can get to 6500 K and over. In the future, the physics associated with a high temperature, such as ionization and chemical reaction, will be added in UGKWP.

To further illustrate the multiscale nature of the simulation, the local Knudsen number $Kn_{GLL}$ based on the gradient $Kn_{GLL} = |\nabla \rho|/\rho$ for the above two cases are presented in Fig. 18, which presents five orders of magnitude difference.

The decline of parallel efficiency in the rarefied regime, as presented in Sec. III B, can be visualized through the averaged number of particles per cell in the simulation. Figure 19 shows the
FIG. 17. Symmetric X–Z cut-plane contour of various flow fields at $Ma = 20$ and $Kn = 0.05$. (a) Temperature contour, (b) density contour, (c) contour of $U$ (X-component velocity), and (d) contour of $W$ (Z-component velocity).

FIG. 18. Local Knudsen number contour on the symmetric X–Z cut-plane at (a) $Ma = 2$ and $Kn = 1$ and (b) $Ma = 20$ and $Kn = 0.05$.

FIG. 19. Distribution of normalized particle number per cell on the symmetric X–Z cut-plane at (a) $Ma = 2$ and $Kn = 1$ and (b) $Ma = 20$ and $Kn = 0.05$. 
distribution of normalized particle number per cell $N/N_{ref}$ on the symmetric X–Z cut-plane at $Ma = 2$, $Kn = 1$ and $Ma = 20$, $Kn = 0.05$. The probability of particle collision in the cell becomes lower with the increment of $\tau/\Delta t$. Therefore, the particle tends to keep free streaming and concentrate in the rearward of the computational domain, especially in the cell with a large volume. This mechanism causes an imbalance in the distributions of particles across different CPU cores. Nevertheless, this problem can be mitigated by...
implementing dynamic load balancing as used in the DSMC implementation. At such a high Mach number, even in different flow regimes, the UGKWP becomes the top choices for the simulations.

D. Hypersonic flow over a space vehicle

The last example is a hypersonic flow at Mach numbers 6 and 10 over a space vehicle in the transition regime \( Kn = 10^{-3} \). This case shows the efficiency and capability of UGKWP for simulating three-dimension hypersonic flow over complex geometry configuration. The angle of attack is 20° in this case. As seen in Fig. 20, the unstructured mesh of 560,593 cells consists of 15,277 pyramids and 545,316 tetrahedra with the minimum cell height of 0.001 m near the front of the vehicle surface. The reference length for the definition of the Knudsen number is \( L = 0.28 \) m. The boundary condition on the vehicle surface is a diffusive one, on which the temperature maintains at \( T_w = 300 \) K. Due to the symmetry, only half of the vehicle is simulated. The inflow is monatomic argon gas with the molecular mass \( m = 6.63 \times 10^{-26} \) kg and diameter \( d = 4.17 \times 10^{-10} \) m at \( T_\infty = 300 \) K. The CFL number for the simulation is 0.95, and the reference viscosity is given by the variable hard sphere (VHS) model with \( \omega = 0.81 \).

The least square reconstruction with the Venkatakrishnan limiter is used in the simulation.

Figure 21 presents the distribution of the temperature, heat flux, pressure, local Knudsen number, and streamlines around the vehicle at Mach number 6. Figure 22 shows the solutions at Mach number 10. Even the free-stream Knudsen number is relatively small, no vortex flow is observed in the rear part of the vehicle, see Figs. 21(d) and 22(d), which is observed in the simulation of the near continuum flow. Meanwhile, from Figs. 21(c) and 22(c), the density-based local Knudsen number \( Kn_{GLL} \) can cover a wide range of values with five orders of magnitude difference. Therefore, a multiscale method, like UGKWP, is necessary to capture the flow physics in different regimes correctly. As presented in Figs. 21(a) and 22(a), a high-temperature region is detected at the leeward side, despite the low intensity of heat exchange upon the vehicle surface. This is mainly caused by particle collisions in the strong recompression region with a relatively low free-stream Knudsen number.

![Fig. 22](image-url)
As for the computational cost, for the $Ma = 6$ case, the initial field is obtained by GKS with 6000 local time stepping, and the time-averaging starts after 12,000 steps of UGWP computation. The simulation runs 24.87 h with 48 cores and consumes 35 GB memory, including 8000 steps of averaging. For the case of $Ma = 10$, $N_{ref} = N_{min} = 400$ particles is used. The simulation is conducted on Tianhe-2 with 8 nodes or 192 cores, and it takes 22.8 h, including 5000 steps GKS calculation with local time stepping for the initial field, and 9000 steps of time-averaging after 10,000 steps UGKP calculation for the steady-state solution.

V. CONCLUSION AND FURTHER IMPROVEMENTS

In this paper, a unified gas-kinetic wave-particle (UGWP) method is constructed on a three-dimensional unstructured mesh with parallel computing on a supercomputer. The scheme is validated for flow simulation in both continuum and rarefied regimes at different flow speeds. Compared with other popular flow solvers, such as the DSMC method and the deterministic DVM-based Boltzmann solver, the UGKP has multiscale property and is efficient in simulating 3D supersonic/hypersonic flow, especially in the transition and near continuum flow regimes. However, for practical engineering applications, further optimization and extension have to be implemented in the code to make it more efficient and comprehensive. It is expected that the dynamic load balancing implementation can enhance the parallel efficiency in the rarefied regime considerably as in conventional DSMC implementation. Moreover, coalescing of small weight particles can also save the memory substantially in the rarefied regime. Besides, the improved sampling technique can be employed to moderate the noise for low-speed or small temperature variance simulations. On the physical modeling side, instead of the BGK, the Shakhov or ellipsoidal statistical model for monatomic gas and the Rykov model for diatomic gas can also be used in the construction of UGKP in 3D computation. Further extensions can be the coupling between the UGKP particle re-sampling with DSMC collision model and the inclusion of complex physical processes, such as ionization and chemical reaction, for the high speed and high temperature flow. In comparison with DVM-based UGKS, the UGWP has distinguishable advantages for the supersonic and hypersonic flow simulations. For the low speed microflows, the UGKS is still preferred, especially with the implementation of acceleration techniques, such as implicit and x multigrid.

AUTHORS’ CONTRIBUTIONS

All authors contributed equally to this work.

ACKNOWLEDGMENTS

This work was supported by the Hong Kong research grant council (Grant No. 16206617), the National Natural Science Foundation of China (Grant Nos. 11772281 and 91852114), and the National Numerical Windtunnel project.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

1. D. Jiang, M. Mao, J. Li, and X. Deng, “An implicit parallel UGKS solver for flows covering various regimes,” Adv. Aerodyn. 1 (1), 8 (2019).
2. G. A. Bird and J. Brady, Molecular Gas Dynamics and the Direct Simulation of Gas Flows (Clarendon Press, Oxford, 1994), Vol. 5.
3. M. Pfeiffer, P. Nizenson, A. Mirza, and S. Fasoulas, “Direct simulation Monte Carlo modeling of relaxation processes in polyatomic gases,” Phys. Fluids 28 (2), 027103 (2016).
4. S. K. Stefanov, “On the basic concepts of the direct simulation Monte Carlo method,” Phys. Fluids 31 (6), 067104 (2019).
5. J. Plimpton, S. G. Moore, A. Borner, A. K. Stagg, T. P. Koehler, J. R. Torczynski, and M. A. Gallis, “Direct simulation Monte Carlo on petaflop supercomputers and beyond,” Phys. Fluids 31, 086101 (2019).
6. T. J. Scanlon, E. Roohi, C. White, M. Darbandi, and J. M. Reese, “An open source, parallel DSMC code for rarefied gas flows in arbitrary geometries,” Comput. Fluids 39 (10), 2078–2089 (2010).
7. C. White, M. K. Borg, T. J. Scanlon, S. M. Longshaw, R. John, D. R. Emerson, and J. M. Reese, “dsmcFoam+: An OpenFOAM based direct simulation Monte Carlo solver,” Comput. Phys. Commun. 224, 22–43 (2018).
8. H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, “A tensorial approach to computational continuum mechanics using object-oriented techniques,” Comput. Phys. 12 (6), 620–631 (1998).
9. C. J. Greenshields, OpenFOAM—The Open Source CFD Toolbox-User Guide, OpenFOAM Foundation Ltd. (2015).
10. See https://openfoam.org/ for the OpenFOAM Foundation, Openfoam; accessed 12 January 2020.
11. S. Dietrich and I. D. Boyd, “Scalar and parallel optimized implementation of the direct simulation Monte Carlo method,” J. Comput. Phys. 126 (2), 328–342 (1996).
12. M. Ivanov, G. Markelov, and S. Gelmishel, “Statistical simulation of reactive rarefied flows—numerical approach and applications,” in 7th AIAA/ASME Joint Thermophysics and Heat Transfer Conference (AIAA, 1998), p. 2669.
13. G. J. LeBeau, “A parallel implementation of the direct simulation Monte Carlo method,” Comput. Methods Appl. Mech. Eng. 174 (3–4), 319–337 (1999).
14. Y. Zhu, C. Zhong, and K. Xu, “Unified gas-kinetic scheme with multigrid convergence for rarefied flow study,” Phys. Fluids 29 (9), 096102 (2017).
15. Y. Zhu, C. Zhong, and K. Xu, “An implicit unified gas-kinetic scheme for unsteady flow in all Knudsen regimes,” J. Comput. Phys. 386, 190–217 (2019).
16. Z.-H. Li, A.-P. Peng, Q. Ma, L.-N. Dang, X.-W. Tang, and X.-Z. Sun, “Gas-kinetic unified algorithm for computer modeling of Boltzmann equation and application to aerothermodynamics for falling disintegration of uncontrolled Tiangong-1 spacecraf,” Adv. Aerodyn. 1 (1), 4 (2019).
17. Ñ. Titarev, M. Dumbser, and S. Utyuzhnikov, “Construction and comparison of parallel implicit kinetic solvers in three spatial dimensions,” J. Comput. Phys. 256, 17–33 (2014).
18. Z. Guo, K. Xu, and R. Wang, “Discrete unified gas kinetic scheme for all Knudsen number flows: Low-speed isothermal case,” Phys. Rev. E 88 (3), 033305 (2013).
19. Z. Guo, R. Wang, and K. Xu, “Discrete unified gas kinetic scheme for all Knudsen number flows. II. Thermal compressible case,” Phys. Rev. E 91 (3), 033313 (2015).
20. E. M. Shakhov, “Generalization of the Krook kinetic relaxation equation,” Fluid Dyn. 3 (5), 95–96 (1968).
21. L. Zhu, S. Chen, and Z. Guo, “dugksFoam: An open source OpenFOAM solver for the Boltzmann model equation,” Comput. Phys. Commun. 213, 155–164 (2017).
22. C. Liu, Y. Zhu, and K. Xu, “Unified gas-kinetic wave-particle methods I: Continuum and rarefied gas flow,” J. Comput. Phys. 401, 108977 (2020).
23. Y. Zhu, C. Liu, C. Zhong, and K. Xu, “Unified gas-kinetic wave-particle methods II. Multiscale simulation on unstructured mesh,” Phys. Fluids 31 (6), 067105 (2019).
24. W. Li, C. Liu, Y. Zhu, J. Zhang, and K. Xu, “Unified gas-kinetic wave-particle methods III: Multiscale photon transport,” J. Comput. Phys. 408, 109280 (2020).
25. K. Xu and J.-C. Huang, “A unified gas-kinetic scheme for continuum and rarefied flows,” J. Comput. Phys. 229 (20), 7747–7784 (2010).

Phys. Fluids 32, 096108 (2020); doi: 10.1063/5.0021199
Published under license by AIP Publishing
Z. Guo, J. Li, and K. Xu, “On unified preserving properties of kinetic schemes,” arXiv:1909.04923 (2019).

K. Xu, Direct Modeling for Computational Fluid Dynamics: Construction and Application of Unified Gas-Kinetic Schemes (World Scientific, 2015).

Y. Zhu, C. Zhong, and K. Xu, “Ray effect in rarefied flow simulation,” J. Comput. Phys. 422, 109751 (2020).

P. L. Bhatnagar, E. P. Gross, and M. Krook, “A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems,” Phys. Rev. 94(3), 511 (1954).

L. H. Holway, Jr., “New statistical models for kinetic theory: Methods of construction,” Phys. Fluids 9(9), 1658–1673 (1966).

V. Venkatakrishnan, “Convergence to steady state solutions of the Euler equations on unstructured grids with limiters,” J. Comput. Phys. 118(1), 120–130 (1995).

T. Barth and D. Jespersen, “The design and application of upwind schemes on unstructured meshes,” in 27th Aerospace Sciences Meeting (AIAA, 1989), p. 366.

P. Vijayan and Y. Kallinderis, “A 3D finite-volume scheme for the Euler equations on adaptive tetrahedral grids,” J. Comput. Phys. 113(2), 249–267 (1994).

R. E. Caflisch, “Monte Carlo and quasi-Monte Carlo for the Boltzmann equation,” Phys. Fluids 17(5), 051703 (2005).

L. Zhu, P. Wang, S. Chen, Z. Guo, and Y. Zhang, “GPU acceleration of an iterative scheme for gas-kinetic model equations with memory reduction techniques,” Comput. Phys. Commun. 245, 106861 (2019).

K. Xu, “A gas-kinetic BGK scheme for the Navier–Stokes equations and its connection with artificial dissipation and Godunov method,” J. Comput. Phys. 171(1), 289–335 (2001).

L. Zhu, P. Wang, S. Chen, Z. Guo, and Y. Zhang, “GPU acceleration of an iterative scheme for gas-kinetic model equations with memory reduction techniques,” Comput. Phys. Commun. 245, 106861 (2019).

K. Xu and J.-C. Huang, “An improved unified gas-kinetic scheme and the study of shock structures,” J. Thermophys. Heat Transfer 22(1), 38–49 (2008).

V. A. Rykov, “A model kinetic equation for a gas with rotational degrees of freedom,” J. Thermophys. Heat Transfer 22(1), 38–49 (2008).

L. Wu, C. White, T. J. Scanlon, J. M. Reese, and Y. Zhang, “A kinetic model of the Boltzmann equation for non-vibrating polyatomic gases,” J. Fluid Mech. 763, 24–50 (2015).