Unified Gas-kinetic Scheme with Multigrid Convergence for Rarefied Flow Study

Yajun Zhu,¹[1] Chengwen Zhong,¹[2] and Kun Xu²[3]

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

(Dated: 5 July 2017)
The unified gas kinetic scheme (UGKS) is a direct modeling method based on the gas dynamical model on the mesh size and time step scales. With the implementation of particle transport and collision in a time-dependent flux function, the UGKS can recover multiple flow physics from the kinetic particle transport to the hydrodynamic wave propagation. In comparison with direct simulation Monte Carlo (DSMC), the equations-based UGKS can use the implicit techniques in the updates of macroscopic conservative variables and microscopic distribution function. The implicit UGKS significantly increases the convergence speed for steady flow computations, especially in the highly rarefied and near continuum regime. In order to further improve the computational efficiency, for the first time a geometric multigrid technique is introduced into the implicit UGKS, where the prediction step for the equilibrium state and the evolution step for the distribution function are both treated with multigrid acceleration. More specifically, a full approximate nonlinear system is employed in the prediction step for fast evaluation of the equilibrium state, and a correction linear equation is used in the evolution step for the update of the gas distribution function. As a result, convergent speed has been greatly improved in all flow regimes from rarefied to the continuum ones. The multigrid implicit UGKS (MIUGKS) is used in the non-equilibrium flow study, which includes microflow, such as lid-driven cavity flow and the flow passing through a finite-length flat plate, and high speed one, such as supersonic flow over a square cylinder. The MIUGKS shows 5 to 9 times efficiency increase over the previous implicit scheme. For the low speed microflow, the efficiency of MIUGKS is several orders of magnitude higher than the DSMC. Even for the hypersonic flow at Mach number 5 and Knudsen number 0.1, the MIUGKS is still more than 100 times faster than the DSMC method for a convergent steady state solution.
Keywords: multigrid method, unified gas kinetic scheme, rarefied flows, multiscale physics
I. INTRODUCTION

With discretized particle velocity space, the unified gas kinetic scheme (UGKS) was an extension from the gas kinetic scheme (GKS) for the Navier-Stokes (NS) solution to the flow dynamics in the entire Knudsen regimes\textsuperscript{1,2}. As a NS solver, the GKS updates macroscopic conservative flow variables only. But, the UGKS evolves both macroscopic flow variables and the microscopic gas distribution function. In both schemes, a time-dependent gas distribution function from kinetic model equation is used for the flux evaluation, and this time evolving solution covers the gas dynamics from the initial non-equilibrium state to the final hydrodynamic equilibrium one. The real solution used for the updates of macroscopic flow variables and the gas distribution function depends on the relative values of particle collision time $\tau$ and the local numerical time step $\Delta t$. The main difference between GKS and UGKS is the set-up of the initial distribution function around a cell interface at the beginning of each time step. For GKS, it is reconstructed from the updated macroscopic flow variables through the Chapman-Enskog expansion, and for UGKS it directly uses the updated gas distribution function. The UGKS can describe highly non-equilibrium flow physics due to the update of the discretized distribution function. With a variation of the ratio between the time step and local particle mean collision time, the UGKS is capable to present the Boltzmann solution in the rarefied flow regime and the NS solution in the continuum flow domain. In the transition regime, a reliable solution can be obtained by UGKS as well\textsuperscript{3}. In addition, different from the discrete velocity method (DVM)\textsuperscript{4} and the direct simulation of Monte Carlo (DSMC) method\textsuperscript{5}, the cell size and time step in UGKS are not restricted by the particle mean free path and collision time due to the implicit treatment of particle collision term inside each cell with the help of updated macroscopic flow variables. The distinguishable multi-scale feature of the UGKS makes it suitable in the gas dynamics
study with multiple flow regimes in a single computation, such as the flow passing through a nozzle from the inside highly compressed continuum flow (\(\tau/\Delta t \ll 1\)) to the outside highly rarefied one (\(\tau/\Delta t \gg 1\)).

In the past years, the UGKS has been validated extensively and it gives accurate solutions in all flow regimes. The method can be easily extended to more complex gases, such as diatomic molecule gas and multi-species flow. The methodology of direct modeling in UGKS can be used to construct numerical methods in other transport processes, such as radiation and phonon transfer and plasma physics. The UGKS provides a promising tool and shows great potentials in the engineering applications, e.g., to the micro-electro-mechanical system and spacecraft designs.

The barriers in front of UGKS for preventing its wide applications in comparison with the DSMC method are the high memory requirements and computational cost, especially for the hypersonic flow and high temperature variation. However, since the UGKS is still an equation-based method, it has advantages in comparison with particle methods in the reduction of computational cost. Many acceleration techniques in traditional computational fluid dynamics (CFD) can be directly adopted in UGKS. One way to reduce the computational cost in UGKS is to reduce the discretization points, such as adopting a moving and adaptive mesh in the physical and velocity space. With adaptive discretization techniques, the computational cost could be controlled to a tolerable level even for highly non-equilibrium flow problems. Another way is to adopt acceleration techniques. For explicit scheme, the numerical stability imposes the Courant-Friedrichs-Lewy (CFL) condition on the time step. But with implicit treatments, this constraint can be released and the computational efficiency can be greatly enhanced. The implicit GKS has been constructed for a faster convergence to the Navier-Stokes solutions. For rarefied flows, several implicit schemes have been proposed based on the iterative algorithms in updating the discretized gas distribution function. As pointed out, the direct explicit
treatment of the equilibrium state in the collision term of the kinetic equation may slow down the convergence of the implicit schemes, especially near continuum flow regime. Hence in a previous study\cite{20}, an implicit UGKS with a prediction step for the equilibrium state was developed to increase acceleration convergence. By first updating the conservative variables implicitly, the collision term in the kinetic equation can be treated in an implicit way, which drives the gas distribution function to a steady state solution efficiently. The implicit UGKS has been validated to be a robust and efficient method in all flow regimes. In order to further speed up the convergence of UGKS for a steady flow solution, the multigrid method, which is one of the most outstanding acceleration techniques in CFD, will be implemented in the implicit UGKS in this paper.

The study of multigrid technique may originate from 1960s\cite{21,22}. Since Brandt’s works\cite{23} in 1970s, the multigrid method got a fast development in practical computations. Now the multigrid method is commonly used in CFD community\cite{24} for solving the Euler and Navier-Stokes equations\cite{25,26}. It has been applied to the GKS\cite{15,27} for acceleration to the steady state solutions for the continuum flow computations. There are many monographs\cite{28,29,30} about multigrid techniques and their numerical implementations. The basic idea behind all multigrid strategies is to accelerate the solution at fine grid by computing corrections on a coarser grid\cite{32} to eliminate low-frequency errors efficiently. In general, an iterative algorithm can reduce the high-frequency errors faster than the low-frequency ones. The multiple grid method is to make the transition between the low and high frequency modes through a change of cell size, and to eliminate the low frequency error in an even coarse mesh by increasing its spatial frequency.

In this paper, we develop a multigrid method for the implicit UGKS, which further improves its convergence efficiency in rarefied flow computations. The implicit UGKS\cite{20} has the prediction stage for evaluating the implicit part of equilibrium state
in the collision term, and the evolution stage for updating the gas distribution func-
tion. Both stages of the implicit UGKS are treated with multigrid techniques to
ensure a fast convergence in all flow regimes. It turns out that the macroscopic
equations become a nonlinear system, while the implicit evolution equations for the
distribution function at discrete particle velocities are still linear ones. As a result,
the full approximation storage scheme (FAS)\textsuperscript{28} is used in the prediction step for
the conservative flow variables and the correction scheme (CS)\textsuperscript{28,29} for solving linear
equations is utilized in the evolution of the distribution function. For the first time, a
multigrid method is used in the UGKS for the rarefied flow computation. After pre-
senting the scheme, many rarefied flow cases from low to high speed ones covering a
wide range of flow regimes will be studied, such as lid-driven cavity flow, flow passing
through a finite-length flat plate, and supersonic flow over a square cylinder. In all
cases presented in the current paper, the implicit UGKS with multigrid acceleration
is much more efficient than the DSMC method with orders of magnitude differences.

The paper is organized as follows. In section II, the multigrid implicit UGKS
(MIUGKS) is presented. Section III is about the analysis and remarks on the current
multigrid method. Section IV presents the rarefied flow studies using the MIUGKS.
The last section is the conclusion.

II. MULTIGRID IMPLICIT UGKS

In this section, the implicit UGKS will be introduced first\textsuperscript{20}. The implicit UGKS
is a pseudo-time-marching scheme for steady state solution. In fact, the implicit
scheme can be interpreted as a numerical smoothing method, which can be naturally
incorporated into a multigrid framework. The basic components in the multigrid
method will be described via the detailed formulation of a two-grid cycle. The
extension to multiple grids is straightforward through a recursive way on the basis
of the two-grid cycle.

A. Implicit UGKS

For steady flows, the governing equation of macroscopic variables averaged in a finite volume $i$ gives

$$\frac{1}{V_i} \sum_{j \in N(i)} S_{ij} F_{ij} = 0,$$

where $N(i)$ is the set of neighbors of cell $i$, and $j$ is one of the neighboring cell, and $ij$ denotes the interface between cells $i$ and $j$. Here $S_{ij}$ is the area of the interface $ij$ and $V_i$ is the volume of the cell $i$. $F_{ij}$ are the fluxes of conservative variables $W = (\rho, \rho U, \rho V, \rho e)^T$ passing through the cell interface $ij$. Eq. (1) describes the balance of interface fluxes for cell $i$ at steady state.

For the gas distribution function $f_{i,k}$ at the discretized velocity $u_k$, the governing equation can be written as

$$\frac{1}{V_i} \sum_{j \in N(i)} S_{ij} u_{k,n} \tilde{f}_{ij,k} - \frac{g_{i,k} - f_{i,k}}{\tau_i} = 0,$$

where $u_{k,n}$ is the normal component of $u_k$ along the interface $ij$. The interface distribution function $\tilde{f}_{ij,k}$ is a local physical time $\Delta t_p$ averaged quantity, which identifies different physics in different regimes. The equilibrium state $g_{i,k}$ can be the Maxwellian distribution, or a Shakhov-type model with the justification of the Prandtl number. The multiple scale nature of UGKS is fully determined by the modeling of the flux function at a cell interface $\tilde{f}_{ij,k}$, which will be presented later.

Since Eq. (1) and Eq. (2) depict the final steady state solution, which denotes time $t \to \infty$ for explicit scheme or iteration step $n \to \infty$ for iterative methods, they could be directly regarded as the implicit governing equations of the accurate final solution. In general, the solution is basically impossible to be obtained in one step. Numerical
computation will start from an approximate solution (or initial state solution) and then get more accurate solutions step by step using explicit time-marching schemes or implicit iterative methods.

For the implicit UGKS, given an approximate solution $f_{i,k}^n$ and $W_i^n$ at step $n$, the errors for macroscopic and microscopic variables can be defined as

\[ E_i^n = W_i - W_i^n, \]  

(3)

and

\[ e_{i,k}^n = f_{i,k} - f_{i,k}^n. \]  

(4)

The residuals become

\[ R_i^n = -\frac{1}{V_i} \sum_{j \in N(i)} S_{ij} F_{ij}^n, \]  

(5)

and

\[ r_{i,k}^n = \frac{g_{i,k} - f_{i,k}^n}{\tau_i} - \frac{1}{V_i} \sum_{j \in N(i)} S_{ij} u_{k,n} \tilde{f}_{ij,k}^n. \]  

(6)

As a result, the residual equations (or defect equations) for implicit iterations go to

\[ \frac{1}{V_i} \sum_{j \in N(i)} S_{ij} (F_{ij} - F_{ij}^n) = R_i^n, \]  

(7)

and

\[ \frac{e_{i,k}^n}{\tau_i} + \frac{1}{V_i} \sum_{j \in N(i)} S_{ij} u_{k,n} e_{ij,k}^n = r_{i,k}^n. \]  

(8)

If $E_i^n$ and $e_{i,k}^n$ were precisely solved from Eq. (7) and Eq. (8), we could get the exact solution $W_i$ and $f_{i,k}$ from Eq. (3) and Eq. (4). However, it is much too difficult to solve the residual equations (7) and (8) with the full UGKS terms of $F_{ij} - F_{ij}^n$ and $e_{ij,k}^n$. Moreover, it requires the unknown equilibrium state $g_{i,k}$ in evaluation of microscopic residual $r_{i,k}^n$. Therefore, we divide the solving process into two steps, i.e., prediction for equilibrium state and evolution of gas distribution function. In the
prediction step, with the simplified implicit fluxes on the left hand side of Eq. (7) we can approximately solve the Eq. (7) to get a correction of conservative variables \( \Delta W^n_i \) as an approximation of \( E^n_i \). Then the equilibrium state \( g_{i,k} \) in the evaluation of \( r^n_{i,k} \) can be approximated by \( \tilde{g}^{n+1}_{i,k} \) obtained from \( \tilde{W}^{n+1}_i = W^n_i + \Delta W^n_i \). Consequently, in the evolution step we can obtain a correction of distribution function \( \Delta f^n_{i,k} \) as an approximate \( e^n_{i,k} \) once we solve Eq. (8) using simplified fluxes \( e^n_{ij,k} \). Then the distribution function can be updated by \( f^{n+1}_{i,k} = f^n_{i,k} + \Delta f^n_{i,k} \) and the equilibrium state \( g^{n+1}_{i,k} \) and the conservative variables \( W^{n+1}_i \) can be renewed by the compatibility condition from \( f^{n+1}_{i,k} \). Following these procedures iteratively, the convergent solution can be obtained step by step from the corrections, accompanied with error smoothing and reduction. Details in these two steps will be introduced in the following.

1. **Prediction step for equilibrium state**

In order to evaluate the residuals \( r^n_{i,k} \) in Eq. (6), the equilibrium state \( g_{i,k} \) should be given first. Here we give a predicted one \( \tilde{g}_{i,k} \) by solving the implicit governing equations (7) of macroscopic variables.

For evaluation of the residual \( R^n_i \) in Eq. (5), we use

\[
F^n_{ij} = \sum_k \frac{1}{\Delta t_p} \int_0^{\Delta t_p} u_{k,n} \tilde{f}^{n}_{ij,k}(t) \psi_k dt,
\]

where \( \tilde{f}^{n}_{ij,k}(t) \) is a time-dependent distribution function constructed by the analytic solution of the kinetic model equation along a characteristic line, and \( \psi_k \) is the vector for its moments of mass, momentum and energy. \( \Delta t_p \) is the physical time step determined by the CFL condition with a Courant number less than 1, which recovers the local flow physics. The residuals \( R^n_i \) are completely evaluated by explicit UGKS fluxes, see details in papers. In order to solve Eq. (7), the fluxes on the left hand
side will be simplified by Euler equations-based flux splitting method,

\[ F_{ij} - F_{ijn} = \frac{1}{2} [T_i + T_j + \Gamma_{ij}(W_i - W_j)] - \frac{1}{2} [T^n_i + T^n_j + \Gamma_{ij}(W^n_i - W^n_j)], \]

where \( T \) is the Euler flux. \( \Gamma_{ij} \) satisfies

\[ \Gamma_{ij} \geq \Lambda_{ij} = |U_{ij} \cdot n_{ij}| + a_s, \]

where \( \Lambda_{ij} \) represents the spectral radius of the Euler flux Jacobian, which can be evaluated by the macroscopic velocity \( U_{ij} \) and speed of sound \( a_s \) at the interface \( ij \). Here \( n_{ij} \) is the normal vector of the interface along the direction from cell \( i \) to cell \( j \).

Generally, a stable factor \( s_{ij} \) related to the kinematic viscosity coefficient \( \nu \) can be introduced into the calculation of \( \Gamma_{ij} \),

\[ \Gamma_{ij} = \Lambda_{ij} + s_{ij} = \Lambda_{ij} + \frac{2\nu}{n_{ij} \cdot (x_j - x_i)}. \]

Then the residual equations can be rewritten as

\[ \frac{1}{2V_i} \sum_{j \in N(i)} S_{ij} \Gamma_{ij} E^n_i + \frac{1}{2V_i} \sum_{j \in N(i)} S_{ij} [T(W^n_j + E^n_j) - T(W^n_j) - \Gamma_{ij} E^n_j] = R^n_i. \]

For two dimensional cases on structured mesh, it will form a penta-diagonal matrix, which can be solved by LU-SGS iterations. With the correction \( \Delta W_i^n \) for conservative variables as an approximation of \( E_i^n \), we can get the predicted equilibrium state \( \tilde{g}_{i,k} \) from the newly evolved macroscopic variables \( W_i^n + \Delta W_i^n \).

### 2. Evolution step for updating particle distribution function

Once we get the predicted equilibrium state \( \tilde{g}_{i,k} \), the microscopic residual in Eq. (6) can be evaluated. Simplifying the numerical flux on the left hand side of Eq. (8) by
an upwind approach,

\[
e_{ij,k}^n = \frac{1}{2} \left( e_{i,k}^n + e_{j,k}^n \right) + \frac{1}{2} \text{sign}(u_k \cdot n_{ij}) \left( e_{i,k}^n - e_{j,k}^n \right)
\]
\[= \frac{1}{2} \left[ 1 + \text{sign}(u_{k,n}) \right] e_{i,k}^n + \frac{1}{2} \left[ 1 - \text{sign}(u_{k,n}) \right] e_{j,k}^n, \tag{13}\]

we get the residual equation

\[
D_{i,k} e_{i,k}^n + \sum_{j \in N(i)} D_{j,k} e_{j,k}^n = r_{i,k}^n, \tag{14}\]

where

\[
D_{i,k} = 1 + \frac{1}{2V_i} \sum_{j \in N(i)} u_{k,n} S_{ij} \left[ 1 + \text{sign}(u_{k,n}) \right],
\]
\[
D_{j,k} = \frac{1}{2V_i} u_{k,n} S_{ij} \left[ 1 - \text{sign}(u_{k,n}) \right]. \tag{15}\]

Here \(r_{i,k}^n\) is evaluated by the time-averaged UGKS flux over a physical time step \(\Delta t_p\) and the collision term with the predicted equilibrium state \(\tilde{g}_{i,k}\). By using LU-SGS iterations, a correction \(\Delta f_{i,k}^n\) approximating the error \(e_{i,k}^n\) can be obtained from Eq. (14). After obtaining \(\Delta f_{i,k}^n\), the solution \(f_{i,k}^{n+1}\) can be updated. Consequently, the macroscopic variables can be updated as well by taking moments of the renewed distribution function.

Different from the nonlinear Eq. (12), Eq. (14) can be regarded as a linear equation for the distribution function error if the mean collision time \(\tilde{\tau}_i\) is frozen locally within each iteration step, because the coefficients \(D_{i,k}\) and \(D_{j,k}\) are only related to the discretization of the physical and velocity space. Therefore, different multigrid techniques are imposed on solving the nonlinear equation (12) of the conservative variables and the linear equation (14) of the gas distribution function. Details will be introduced next.
B. A two-grid cycle implicit UGKS

A two-grid cycle method is a basis for any multigrid algorithm. It is a combination of error smoothing and coarse grid correction. Usually it consists of a pre-smoothing, a coarse grid correction, and a post-smoothing part. Here we implement two-grid cycle technique into the implicit UGKS to develop a multigrid method.

Based on a finer grid $\Omega_h$ and a coarser grid $\Omega_H$, the iteration step of the two-grid cycle for the implicit UGKS is illustrated in Fig. 1 with a prediction step and an evolution step. Since interpolations of macroscopic variables and distribution function between two grids are involved, for a better representation the solutions and interpolations will be denoted as grid functions and grid operators respectively. As shown in Fig. 1, the two stages in the implicit UGKS are considered successively with the multigrid technique.

In the prediction step, the residuals $R^n_h$ of the implicit macroscopic equations are evaluated by the UGKS fluxes on a fine grid $\Omega_h$ from a given approximate solutions of $f^n_h$ and $W^n_h$. After $\nu_1$ times of pre-smoothing on this level, the residuals and conservative variables are updated to $\hat{R}^n_h$ and $\hat{W}^n_h$. Then, both the renewed residuals
and the smoothed conservative variables are restricted from the fine grid $\Omega_h$ to the coarse grid $\Omega_H$ by a transfer operator $I_h^H$. On a coarse grid, the residual equations with restricted residuals $R_n^H$ and $W_n^H$ should be solved to get a correction of the conservative variables $\Delta W_n^H$. Consequently, the residuals and solutions on the fine grid can be renewed through a prolongated correction $\Delta W_n^H$ with a transformation operator $I_h^H$. Again, taking $\nu_2$ times for post-smoothing, the smoothed solution $\tilde{W}_{n+1}^H$ is regarded as the final result in this prediction step to give a predicted equilibrium state for the following evolution of the gas distribution function.

In the evolution step, the residual $r_n^h$ is obtained first on a fine grid $\Omega_h$ from the given approximate solution $f_n^h$ and the predicted equilibrium state $\tilde{g}_{n+1}^h$. Meanwhile, the residual will be renewed after $\nu_1$ times of pre-smoothing, and a correction $\Delta \tilde{f}_n^h$ will be obtained during these smoothing processes. As mentioned in Section II A 2, the residual equation (14) of gas distribution function is a linear equation, therefore only the residual $\tilde{r}_n^h$ is needed to be restricted on a coarse grid to get a correction. The correction $\Delta \tilde{f}_n^h$ obtained by solving the residual equation on a coarse grid will be prolongated back onto the fine grid. With the interpolated correction $\Delta \tilde{f}_n^h$, and renewed residual $\tilde{r}_n^h$, $\nu_2$ times of post-smoothing can be carried out to give an updated distribution function $f_{n+1}^h$. Then the conservative variables $W_{n+1}^H$ and equilibrium state $g_{n+1}^H$ can be updated from the moments of the gas distribution function.

So far, the two-grid cycle for the prediction and the evolution steps has been illustrated for the updating of the distribution function from $f_n^h$ to $f_{n+1}^H$. It should be noted that the only difference between the two steps is whether the intermediate smoothed solution is required and restricted on a coarse grid, i.e., the difference between the so-called full approximation storage scheme and correction scheme. In the following, each component of the multigrid method will be introduced in details.
C. Numerical procedures in the multigrid method

1. Transfer operator : Restriction

For initialization on a successive coarser grid, variables such as the residuals should be transferred (restricted) from finer grid to coarser ones.

A restriction operator $I_h^H$ maps fine-grid functions to coarse-grid functions by a volume weighted interpolation for cell-centered schemes. For a specific variable denoted by $Q$, the restricted result inside cell $I$ on coarse grid $\Omega_H$ becomes

$$
(I_h^H Q_h)_I = \frac{\sum_{j \in S(I)} (Q_h V_h)_j}{\sum_{j \in S(I)} (V_h)_j},
$$

(16)

where $S(I)$ is the set of subcells of cell $I$ and $j$ is one of the subcell members. In prediction step, both the smoothed conservative variables $\hat{W}_n^h$ and renewed residuals $\hat{R}_n^h$ should be restricted to a coarse grid to form the residual equation (12) on $\Omega_H$. As illustrated in Fig. 2, we have

$$
(W_H^n)_I = (I_h^H \hat{W}_h^n)_I = \frac{1}{V_I} \sum_{j \in S(I)} (\hat{W}_h^n)_j V_j,
$$

$$
(R_H^n)_I = (I_h^H \hat{R}_h^n)_I = \frac{1}{V_I} \sum_{j \in S(I)} (\hat{R}_h^n)_j V_j
$$

(17)

where $S(I) = \{a, b, c, d\}$. In evolution step, only the residual is needed on a coarse
Therefore, we have
\[
(r^n_H)_I = (I^H_h r^n_h)_I = \frac{1}{V_I} \sum_{j \in S(I)} (r^n_h)_j V_j.
\] (18)

2. **Smoothing**

The smoothing process, whether the pre-smoothing or post-smoothing, is to solve the residual equations by LU-SGS iterations to get a more accurate solution. It is a correction process of the solutions on a single grid. This is why one iteration step of the original implicit UGKS on a single grid is claimed as a smoothing method in Section II A. In the current method, solving the residual equations on a coarsest grid is indeed implemented by applying LU-SGS iterations, i.e., through adequate times of smoothing.

In the prediction step, the residual equations (12) are rewritten on a coarse grid \(\Omega_H\) as
\[
\frac{1}{V_I} \sum_{J \in N(I)} S_{IJ} F_{IJ}(W^n_H) - \frac{1}{V_I} \sum_{J \in N(I)} S_{IJ} F_{IJ}(W^n_H) = (R^n_H)_I.
\] (19)

where \(W^n_H\) are the restricted conservative variables and \(W_H\) are the accurate solutions of these equations. After the first time of LU-SGS iteration on this grid, the above equations can be solved to give new approximation solutions \(W^{(1)}_H\). Denoting the \(m\)-th intermediate solution as \(W^{(m)}_H\), we get the governing equations for the \((m + 1)\)-th time of smoothing, i.e.,
\[
\frac{1}{V_I} \sum_{J \in N(I)} S_{IJ} F_{IJ} - \frac{1}{V_I} \sum_{J \in N(I)} S_{IJ} F_{IJ}(W^{(m)}_H) = (P^n_H)_I - \frac{1}{V_I} \sum_{J \in N(I)} S_{IJ} F_{IJ}(W^{(m)}_H),
\] (20)

where \(P^n_H\) is the forcing function defined as the difference between the residuals directly transferred from the fine grid, and the macroscopic evolution equations-
determined residuals which are recomputed on a coarse grid, i.e.,

\[
(P^n_H)_I - R_I(W^n_H) = (I^H_h R^n_h)_I + \frac{1}{V_I} \sum_{J \in N(I)} S_{IJ} F_{IJ}(W^n_H).
\]  

(21)

This is commonly used in solving nonlinear residual equations. Here \( F_{IJ}(W^n_H) \) are calculated by a flux splitting method as in Eq. (10). Therefore, for the \((m + 1)\)-th smoothing process, the residuals on the right hand side of Eq. (20) can be updated by

\[
R^{(m)}_H = P_H + R(W^{(m)}_H).
\]  

(22)

For the first smoothing iteration the Eq. (20) is identical to the Eq. (19). By solving Eq. (20) with LU-SGS iterations, multiple smoothing processes can be carried out.

Similarly, for the \((m + 1)\)-th smoothing process of the distribution function in the evolution step, the residual equation (14) can be rewritten on a coarse grid as

\[
D_I e^{(m)}_I + \sum_{J \in N(I)} D_J e^{(m)}_J = (I^H_h R^n_h)_I - \left( D_I \Delta f^{(m)}_I + \sum_{J \in N(I)} D_J \Delta f^{(m)}_J \right),
\]  

(23)

where \( e^{(m)}_I = f_I - f^{(m)}_I \) and \( \Delta f^{(m)}_I = f^{(m)}_I - f^{(0)}_I \). Here \( f_I \) is the accurate solution of Eq. (23) and \( f^{(0)}_I \) is the initial distribution function on \( \Omega_H \) imaginarily restricted from \( \Omega_h \). The purpose for us to give the expression of \( e^{(m)}_I \) and \( \Delta f^{(m)}_I \) by the intermediate distribution function \( f^{(m)}_I \) is just for a better understanding of Eq. (23). It should be noted that the distribution function \( f_I \) is indeed not a necessity in computations, while only \( \Delta \)-quantities are actually involved. In computation, the residual on the right hand side of Eq. (23) for each smoothing process is updated by

\[
(i^{(m)}_H)_I = (I^H_h R^n_h)_I - \sum_{\alpha=1}^m \left( D_I \delta f^{(\alpha)}_I + \sum_{J \in N(I)} D_J \delta f^{(\alpha)}_J \right),
\]  

(24)

where \( \delta f^{(\alpha)}_I = f^{(\alpha)}_I - f^{(\alpha-1)}_I \) is the correction of the distribution function obtained from each smoothing process.

17
Taking sufficient times in the smoothing process, the residual equations (20) and (23) are supposed to be solved to give the coarse-grid corrections. In prediction step, the corrections of the conservative variables are obtained from \( \Delta W_H^n = W_{H}^{n+1} - W_{H}^n \) while in evolution step the total correction of the distribution function is computed by a summation of the intermediate corrections \( \delta f^{(\alpha)} \) in each smoothing process. Up to this point, we have obtained the corrections on the coarsest grid, which will be prolongated to finer grids to reduce the low-frequency solution error on finer grids.

### 3. Transfer operator: Prolongation

The bilinear interpolation is used to prolongate the corrections from the coarser grids to finer ones. As shown in Fig. 3(a) the interpolated result of a specific variable \( Q_H \) on fine grid is

\[
(I_H^i Q_H)_i = \frac{\sum_{J \in S(i)} w_J Q_J}{\sum_{J \in S(i)} w_J}, \tag{25}
\]
where $S(i)$ is the set of the coarse-grid stencil cells for the fine-grid cell $i$. The weights $w_J$ are

$$
\begin{align*}
  w_A &= \frac{h_{BC}h_{CD}}{S_{ABCD}}, &
  w_B &= \frac{h_{CD}h_{AD}}{S_{ABCD}}, \\
  w_C &= \frac{h_{AD}h_{AB}}{S_{ABCD}}, &
  w_D &= \frac{h_{AB}h_{BC}}{S_{ABCD}},
\end{align*}
$$

where $S_{ABCD} = (h_{AB} + h_{CD})(h_{BC} + h_{AD})$, and $h$ is the distance between the center of fine-grid cell $i$ and the line which connects the cell centers of two neighboring members in $S(i)$.

For instance, the prolongated correction of distribution function gives

$$
(\Delta \hat{f}_n^i) = (I_h^h \Delta f_H^n)_i = w_A \Delta f_A^n + w_B \Delta f_B^n + w_C \Delta f_C^n + w_D \Delta f_D^n.
$$

Specifically, on Cartesian grids the weights give

$$
\begin{align*}
  w_A &= \frac{9}{16}, &
  w_B &= \frac{3}{16}, \\
  w_C &= \frac{1}{16}, &
  w_D &= \frac{3}{16}.
\end{align*}
$$

The weights given in Eq. (26) are also available for extrapolation treatment of the cells near boundaries. In the cases of extrapolation some of the distance $h$ may be negative. For the cell $i$ near boundary shown in Fig. 3(b), $h_{AD}$ is negative while for the corner cell illustrated in Fig. 3(c) both $h_{AB}$ and $h_{AD}$ are negative. Specifically on Cartesian grids, the weights satisfy

$$
\begin{align*}
  w_A &= \frac{15}{16}, &
  w_B &= \frac{3}{16}, \\
  w_C &= -\frac{1}{16}, &
  w_D &= \frac{5}{16}.
\end{align*}
$$

for boundary cells and

$$
\begin{align*}
  w_A &= \frac{25}{16}, &
  w_B &= -\frac{5}{16}, \\
  w_C &= \frac{1}{16}, &
  w_D &= -\frac{5}{16}.
\end{align*}
$$

for corner cells.
D. Extension to multiple grids

Each component in the multigrid method of implicit UGKS has been described above. In this subsection, the multigrid algorithm will be constructed from recursion of the two-grid cycle.

First we define the multigrid cycle in prediction step as
\[ W_{l}^{n+1} = \text{FAS} (l, W_{l}^{n}, R_{l}^{n}, \text{Eq. (20)}, \nu_1, \nu_2), \]  
and the multigrid cycle in evolution step as
\[ \Delta f_{l}^{n} = \text{CS} (l, r_{l}^{n}, \text{Eq. (23)}, \nu_1, \nu_2). \]

where \( l \) is the level index, \( \nu_1 \) and \( \nu_2 \) are the times of pre-smoothing and post-smoothing respectively.

In details, the recursive descriptions of an FAS cycle, i.e., Eq. (31), in prediction step are the following.

(a): Pre-smoothing

- calculate the forcing function on this level of grid \( \Omega_l \) by Eq. (21).
- get a better approximation \( \hat{W}_{l}^{n} \) and the updated residual \( \hat{R}_{l}^{n} \) by applying \( \nu_1 \) times of smoothing through two procedures, i.e.,
  - get the intermediate approximation \( \hat{W}_{l}^{(m)} \) by solving Eq. (20) with LU-SGS iterations.
  - renew the residuals by Eq. (22) with forcing function.

(b): Coarse grid correction

get the initial approximate solution \( W_{l+1}^{n} \) and residual \( R_{l+1}^{n} \) by restricting \( \hat{W}_{l}^{n} \) and \( \hat{R}_{l}^{n} \) from the fine grid \( \Omega_l \) to the coarser grid \( \Omega_{l+1} \), see Eq. (17).
• compute a new approximate solution $W_{l+1}^{n+1}$ on the coarse grid $\Omega_{l+1}$, which may be one of the following two cases
  
  – if $l+1 = N_l$, solve the residual equations by sufficient times of smoothing, see Eqs. (20), (21) and (22)
  
  – if $l + 1 < N_l$, apply another FAS cycle on this level

$$W_{l+1}^{n+1} = \text{FAS} (l+1, W_{l+1}^n, R_{l+1}^n, \text{Eq. (20)}, \nu_1, \nu_2),$$ (33)

• get the correction $\Delta W_{l+1}^n$ from the difference $W_{l+1}^{n+1} - W_{l+1}^n$.

• interpolate the corrections to the finer grid $\Omega_l$ obtaining $\Delta \bar{W}_l^n$ by Eq. (25).

• update the solution to $\bar{W}_l^n$ on $\Omega_l$

(c): Post-smoothing

• get a smoothed approximate solution $W_l^{n+1}$ by applying $\nu_2$ steps of smoothing, with the following two steps:

  – update the residual with the approximate solution and forcing function by Eq. (22).

  – get the smoothed solution by solving Eq. (20).

The recursive CS cycles in Eq. (32) for the evolution step can be described as follows.

(a): Pre-smoothing

• get the approximate correction $\Delta \hat{f}_l^n$ and update the residual to $\hat{r}_l^n$ by applying $\nu_1$ times of smoothing with two procedures, i.e.

  – get the intermediate correction $\Delta \hat{f}_l^{(m)}$ by solving Eq. (23) with LU-SGS iterations.
– renew the residuals by Eq. (24).

(b): Coarse grid correction

- get the residual $r_{l+1}^n$ by restricting $\hat{r}_l^n$ from the fine grid $\Omega_l$ to the coarser grid $\Omega_{l+1}$ by Eq. (18)
- compute a new approximate correction $\Delta f_{l+1}^n$ on the coarse grid $\Omega_{l+1}$, which may be one of the following two cases
  - if $l + 1 = N_l$, solve the residual equation by sufficient times of smoothing, see Eqs. (23) and (24).
  - if $l + 1 < N_l$, apply another one CS cycle on this level

\[ \Delta f_{l+1}^n = \text{CS} \left( l + 1, r_{l+1}^n, \text{Eq. (23)}, \nu_1, \nu_2 \right), \tag{34} \]

- interpolate the correction back to finer grid $\Omega_l$ by Eq. (25) obtaining $\Delta \tilde{f}_l^n$.
- update the total correction $\Delta \tilde{f}_l^n + \Delta \hat{f}_l^n$ on $\Omega_l$.

(c): Post-smoothing

- get a smoothed correction $\Delta f_l^n$ by applying $\nu_2$ steps of smoothing, similarly following two steps:
  - update the residual by Eq. (24).
  - get the smoothed correction by solving Eq. (23).

With these two recursive definition of multigrid cycles of prediction step and evolution step, the implicit UGKS on multiple grids can be described as

**Step 1**: calculate the time-averaged fluxes over physical time step and get the residual of conservative variables by Eq. (5):
Step 2: obtain the predicted conservative variables from

\[ \tilde{W}_{l=0}^{n+1} = \text{FAS} \left( l = 0, W_{l=0}^n, R_{l=0}^n, \text{Eq. (20)}, \nu_1, \nu_2 \right) ; \] (35)

Step 3: obtain the predicted equilibrium state \( \tilde{g}_{l=0}^{n+1} \) from \( \tilde{W}_{l=0}^{n+1} \);

Step 4: get the residual \( r_{l=0}^n \) on the finest grid from Eq. (6) with predicted equilibrium state;

Step 5: obtain the total correction of distribution function \( \Delta f_{l=0}^n \) from

\[ \Delta f_{l=0}^n = \text{CS} \left( l = 0, r_{l=0}^n, \text{Eq. (23)}, \nu_1, \nu_2 \right) ; \] (36)

Step 6: get the updated the distribution function \( f_{l=0}^{n+1} \) by the total correction;

Step 7: update the conservative variables \( W_{l=0}^{n+1} \) and equilibrium state \( g_{l=0}^{n+1} \) from compatible condition;

Step 8: check the residual of conservative variables

- if the convergent state is reached, stop the calculation,
- if not, go to Step 1

III. REMARKS AND DISCUSSIONS

A. Mesh generation

Different from the algebraic multigrid method (AMG) which is based on mathematic treatment, the current multigrid method adopts geometric multigrid technique, so the concrete multiple grids should be generated first. For a given problem defined on a specific resolution (i.e., on a given discretized mesh), the multiple grids could
be generated from the given discretization in physical space by coarsening algorithm level by level. Another way is to start with a coarsest grid to generate a satisfactory finest grid by refinement. In current paper the coarsening method from finest mesh is chosen to ensure that the convergent solution is defined on the original numerical discretization.

For structured grid, it is straightforward to generate the coarse meshes by deleting the grid points on every second line in each direction, see in Fig. 4. After \( N_l - 1 \) times of coarsening, \( N_l \) levels of grids would be obtained. In this situation, the finest grid should satisfy

\[
N_p = 2^n + 1, \quad n \geq N_l - 1.
\]

where \( N_p \) is the number of grid points in each direction. From Fig. 4, it can be observed that the coarsening method can retain more information of the finest grid, e.g., the growth rate of cell size along the x- and y-directions. For unstructured mesh, generation methods, such as nonnested grids, topological methods and agglomeration methods, were introduced in [24,32], about which we will not further discuss in current paper.
B. Boundary condition

Generally, boundary conditions in explicit schemes can be imposed by employing ghost cells. To solve the corrections from residual equations, the quantities in the delta-form in ghost cells should be given to start the sweeps for LU-SGS iterations. As described in [20], the boundary conditions for the implicit scheme on each level of grid are derived from those in the explicit UGKS.

For the conservative flow variables, the relation between the ghost cell $j$ and the inner cell $i$ can be expressed in the following form

$$W_j = B(W_i), \quad (37)$$

where $B$ represents a specific transformation relation. The linearization of the above equation gives

$$\Delta W_{j}^{n+1} - \left( \frac{\partial B}{\partial W} \right)_{i}^{n} \Delta W_{i}^{n+1} = 0, \quad (38)$$

which is the macroscopic governing equation of the boundary conditions adopted in the smoothing process of the prediction step. Here, we give the boundary condition for the isothermal walls to illustrate the treatment. For a solid wall moving velocity $U_w = (U_w, V_w)$ at temperature of $T_w$, the macroscopic variables in the ghost cells are

$$\begin{align*}
\rho_j &= \rho_i, \\
\rho_j U_j &= 2\rho_i U_w - \rho_i U_i, \\
\rho_j V_j &= 2\rho_i V_w - \rho_i U_i, \\
\lambda_j &= 2\lambda_w - \lambda_i, \quad (39)
\end{align*}$$

where $\lambda = 1/2RT$. After linearization, the changes of the conservative variables in
the ghost cell vary with the values in the inner cell by

\[
\begin{bmatrix}
\Delta \rho_j \\
\Delta (\rho U)_j \\
\Delta (\rho V)_j \\
\Delta (\rho \varepsilon)_j
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 & 0 \\
2U_0 & -1 & 0 & 0 \\
2V_0 & 0 & -1 & 0 \\
\frac{\partial (\rho \varepsilon)_j}{\partial \rho_i} & \frac{\partial (\rho \varepsilon)_j}{\partial (\rho U)_i} & \frac{\partial (\rho \varepsilon)_j}{\partial (\rho V)_i} & \frac{\partial (\rho \varepsilon)_j}{\partial (\rho \varepsilon)_i}
\end{bmatrix}
\begin{bmatrix}
\Delta \rho_i \\
\Delta (\rho U)_i \\
\Delta (\rho V)_i \\
\Delta (\rho \varepsilon)_i
\end{bmatrix},
\tag{40}
\]

where \((\rho, \rho U, \rho V, \rho \varepsilon)\) are the conservative variables. And we have

\[
\frac{\partial (\rho \varepsilon)_j}{\partial \rho_i} = 2(U_i^2 + V_i^2) - \frac{1}{2}(U_i^2 + V_i^2)(1 + \Upsilon^2) + \frac{K}{4\lambda_i}(\Upsilon^2 - \Upsilon)
\]

\[
\frac{\partial (\rho \varepsilon)_j}{\partial (\rho U)_i} = U_i - 2U_w + U_i \Upsilon^2,
\]

\[
\frac{\partial (\rho \varepsilon)_j}{\partial (\rho V)_i} = V_i - 2V_w + V_i \Upsilon^2,
\]

\[
\frac{\partial (\rho \varepsilon)_j}{\partial (\rho \varepsilon)_i} = -\Upsilon^2
\]

where \(K\) is the total dimensions of degree of freedom and \(\Upsilon = \lambda_i/(\lambda_i - 2\lambda_w)\). Similarly, the boundary conditions for gas distribution function in the smoothing process of the evolution step can be derived from

\[
f_{j,k} = B(f_i).
\tag{41}
\]

We have

\[
\Delta f_{j,k} = (\partial B/\partial f_i)\Delta f_{i,k'},
\tag{42}
\]

where \(f_{i,k'}\) is the distribution function at velocity \(u_{k'}\) corresponding to that in the ghost cells at velocity \(u_k\). For horizontal symmetric interfaces, we have \(\Delta f_{j,k} = \Delta f_{i,k'}\) for \(u_{k,x} = u_{k',x}\) and \(u_{k,y} = u_{k',y}\). For an isothermal solid wall with moving velocity \(U = (U_w, V_w)\) and temperature \(T_w\), for the diffusive reflection boundary condition the reduced distribution function in ghost cells is

\[
f_{j,k} = \frac{\rho_j}{2\pi RT_w} \exp \left[ -\frac{1}{2RT_w} \left( (u_k - U_w)^2 + (v_k - V_w)^2 \right) \right] = \rho_j C_k,
\tag{43}
\]

26
where $C_k$ is a constant for each discrete velocity. Therefore, the variation of gas distribution function in ghost cells is determined by

$$
\Delta f_{j,k} = C_k \Delta \rho_j,
$$

(44)

where $\Delta \rho_j$ is computed by no-transmission condition

$$
\Delta \rho_j \int_{u_n<0} u_n \frac{1}{2\pi RT_w} \exp \left[ -\frac{1}{2RT_w} \left( (u - U_w)^2 + (v - V_w)^2 \right) \right] dudv = - \sum_{u_n,k>0} u_{n,k} \Delta f_{i,k} w_k,
$$

(45)

where $w_k$ is the weight at velocity $u_k$ for numerical integrations.

The boundary conditions should be imposed when the LU-SGS sweeps come to the boundary cells and when the residuals need to be re-evaluated before interpolations.

C. Full multigrid (FMG) method

Generally, the evolution of residuals for a given case during calculations can be separated into three regions as shown in Fig. 5. In the first region, residuals will increase up to a peak value and start to decrease, representing the initial evolution of the flow field. In the second region, residuals decrease exponentially with iteration steps during which the high-frequency error is mainly eliminated. And for the last region, residuals continue decreasing but with a low efficiency because the low-frequency error needs more iterations to attenuate. The multigrid techniques speed up the second and third stages by eliminating low-frequency error by enlarging the cell size. The full multigrid (FMG) method takes the first region into considerations as well to achieve a better efficiency.

The FMG method starts from the coarsest level of grid to provide the initial approximate solutions for finer grids, which can reduce the evolution time of the
FIG. 5. Typical convergence characteristics for steady state solution with three stages, including (a) initial flow evolution stage, (b) high frequency error reduction stage, and (c) low frequency error reduction stage.

FIG. 6. Structure of (a) the full multigrid method in comparison with (b) the typical V-type cycle. Black circle: smoothing process; white circle: solving of residual equations; double slash: FMG interpolation; backslash: restriction; slash: prolongation.

flow field due to the lower computations on coarse grids. The structure of the FMG method in comparison with the V-type cycle is illustrated in Fig. 6. The flow field will evolve on the coarser grids before being interpolated onto finer grids. Different from prolongation in multigrid cycles, the FMG interpolation denoted by double
slash in Fig. 6(a) transfers the conservative variables and gas distribution functions from coarse grids to fine grids instead of their corrections, because it requires all flow variables to initialize the flow field on finer grids. It can be seen that the FMG method will be more effective in complex flows, for which more CPU time will be taken to evolve the initial flow fields using the explicit and implicit schemes on a single grid. With the similar idea, the algorithms with low order of accuracy can be also adopted to get a fully evolved approximate solution before taking a higher-order scheme.

D. Miscellaneous factors

There are many factors that should be considered in multigrid method to ensure the stability and convergence efficiency, such as the type of cycles, number of smoothing steps, the accuracy of the transfer operators, and the chosen of time steps. In the following, a brief discussion about these factors will be given.

There are several types of cycles, such as V-type and W-type, that are commonly used in multigrid methods. All types of cycles can be derived from the basic cycle, i.e., the two-grid cycle. Through recursion of two-grid cycles, the most natural derivation is the V-type cycle. The V-cycle of three-grid methods can be regarded as using a deeper two-grid cycle to solve the residual equation on the coarser grid of the two-grid cycle. Other types of multigrid cycle, such as W-cycle, F-cycle and adaptive cycle, can be constructed by different combination of the two-grid cycles for a better convergence efficiency. In the current paper, the influence of the cycle types will not be further discussed. Generally, the V-cycle is fast enough, but for complex situations advanced types may get better convergence efficiency.

Another factor that will influence the convergence speed is the number of the pre-smoothing and the post-smoothing on each level of grid. Although more smoothing
steps may bring better convergence in one multigrid cycle, it will be more efficient
not to smooth the error too much but rather carrying out a few more multigrid
cycles. As demonstrated in\textsuperscript{29}, common choices are $\nu_1 + \nu_2 \leq 3$ in practice. In this
paper, two pre-smoothing steps are carried out before the restriction of residuals and
one post-smoothing step is carried out after the prolongation for an upwind spatial
discretization\textsuperscript{24}.

The restriction and prolongation should also satisfy certain accuracy requirements\textsuperscript{24,29},
i.e.,
\begin{equation}
m_R + m_P > m_E, \tag{46}
\end{equation}
where $m_R$ and $m_P$ are the orders of the accuracy of the restriction and prolongation
operators, respectively. $m_E$ is the order of the numerical scheme. As given in Section
\textsuperscript{II.C.1} and \textsuperscript{II.C.3} the restriction by using the volume weighted interpolation and the
prolongation by using bilinear interpolation give $m_R = m_P = 2$, and the second-order
UGKS gives $m_E = 2$.

As described in paper\textsuperscript{20}, there are two time steps in the implicit UGKS. The
physical time step $\Delta t_p$, which is used to calculate the time-averaged fluxes in the
evaluation of residuals, is related to the cell size through the CFL condition. The
other one is a pseudo-time step, namely the numerical time step $\Delta t_n$, which is ap-
plied in the temporal discretization of the governing equations. For steady state
solutions, the numerical time step is not a necessity, therefore it does not appear in
the description of the current multigrid method. However, the numerical time step
does help to improve the stability of the implicit schemes for tough numerical cases.
If necessary, a term of $1/\Delta t_n$ could be added into the coefficients before the errors $E_i^n$
and $e_{i,k}^n$ in Eq. (12) and Eq. (14). For instance, a numerical time step which increases
exponentially with iteration steps is used in the calculation of the hypersonic flow
around the square cylinder in Section \textsuperscript{IV.C}. 
IV. RAREFIED FLOW STUDIES

In this section, the multigrid implicit UGKS will be used to study rarefied flow phenomena from low and high speed at various Knudsen numbers. All UGKS computations in this section are carried out on a single machine with a processor of Intel(R) Core(TM) i5-4570 CPU@3.2GHz, and no parallel technique is adopted here. The comparison in terms of accuracy and computational efficiency between MIUGKS and DSMC, whenever available, will be presented.

A. Lid-driven cavity flow

Simulations of lid-driven cavity flows are studied at different Knudsen numbers. Following the previous work\textsuperscript{[33,38]}, the gas in the cavity is argon with molecular mass $m_0 = 6.63 \times 10^{-26} \text{kg}$ and with an initial temperature $T_0 = 273K$. The cavity has a fixed wall temperature $T_w = 273K$ and a moving lid at a constant velocity $U_w = 50 \text{m/s}$. The Knudsen number is defined as the ratio of mean free path to the length of cavity side wall. The dynamic viscosity is evaluated by $\mu = \mu_0(T/T_0)^\omega$ where $\omega = 0.81$. Cases at three different Knudsen numbers, i.e., $Kn = 10, 1.0, 0.075$ have been tested.

In physical space, the computational domain is discretized with a mesh of $64 \times 64$ cells. In velocity space, $120 \times 120$, $100 \times 100$ and $80 \times 80$ discrete velocity points are used respectively for cases at $Kn = 10, 1.0$ and $0.075$. In all three cases, the trapezoidal rule is used in the integration of the discretized distribution function to get macroscopic variables. The steady state is thought to be reached when the mean squared residuals of the conservative variables are reduced to a level being less than 31
FIG. 7. Temperature distribution in the cavity at different Knudsen numbers: (a) $Kn = 10$, (b) $Kn = 1$ and (c) $Kn = 0.075$. Background: the current multigrid method; white solid line: the implicit UGKS; black solid line: the DSMC results from paper 33.

$1.0 \times 10^{-6}$, where the residuals are computed by

$$R^n = \sqrt{\sum_{i=1}^{N_c} R_i^2 / N_c}$$

which denotes the variation rate of the conservative variables. Here $N_c$ is the total number of discrete cells in the computational domain.

The results of the temperature distribution in the cavity at different Knudsen numbers have been plotted in Fig. 7. The results of the multigrid method are consistent with those of the original implicit UGKS with a single level of grid, and agree well with the DSMC results obtained from paper 33. We also plot the distribution of the normalized velocities along the vertical and horizontal central lines comparing to the results of DSMC in Fig. 8 which also shows good agreement between the present results and reference data. In Fig. 9, the convergence histories of the energy density with respect to CPU time are given to show the acceleration of the multigrid method. Obvious accelerating effects of the multigrid method on the implicit UGKS can be observed. Moreover, it can be seen that the multigrid methods with
FIG. 8. Normalized X-velocity distribution along vertical central line and normalized Y-velocity distribution along horizontal central line at different Knudsen numbers: (a) $Kn = 10$, (b) $Kn = 1$ and (c) $Kn = 0.075$.

three-level grids and four-level grids converge faster than that with two-level grids. In comparison with the three-level grid method, the computation efficiency of the four-level scheme doesn’t further increase because of the extra computations for another coarser level of grids. In the higher Knudsen number cases at $Kn = 10$ and 1, the multigrid method is about 3 times faster than the original implicit scheme and in the case at $Kn = 0.075$ the acceleration rate can be increased up to 8 times. With a single machine (Intel(R) Core(TM) i5-4570 CPU@3.2GHz), at $Kn = 0.075$ the current scheme can get convergent solution with the CPU time being less than 3 minutes, where the DSMC solution needs parallel supercomputers for that.\[38\]

B. Flat plate flow

Following the studies in paper\[39\], subsonic flow passing over a flat plate with zero thickness at zero angle of attack is studied at different Reynolds numbers. The flat plate has a finite length with a fixed temperature of $T_w = 295K$. The freestream is
FIG. 9. Convergence history of the cavity flow indicated by the residual of energy density at different Knudsen numbers: (a) $Kn = 10$, (b) $Kn = 1$ and (c) $Kn = 0.075$. Black solid line: the implicit UGKS; red dashed line: two-level multigrid method; green dash-dotted line: three-level multigrid method; blue dash-dot-dotted line: four-level multigrid method.

Air with a temperature of $T_\infty = 295K$ at Mach number 0.2. The dynamic viscosity coefficient is calculated by $\mu = \mu_{ref}(T/T_{ref})^\omega$ with a temperature model index $\omega = 0.77$. The global Knudsen number and Reynolds number are defined with respect to the length of the plate and have a relation of $Kn \approx 1.19Ma/Re$.

Cases at different Reynolds numbers are studied using the current multigrid method and to explore its computation efficiency. In all cases, the farfield is 20 times of the plate length far away from the leading edge and trailing edge. As illustrated in Fig. 10, the distances satisfy $L_{AB} = L_{CD} = L_{AF} = 20L_{BC}$. For each case, we adopt different discretization of the physical space and velocity space, see those listed in Table I. Symmetric boundary condition is used for the axes in the upstream and downstream, and diffusive reflection with full thermal accommodation coefficient is adopted in the boundary condition for the isothermal solid wall.

The distributions of the temperature around the plate for all cases are given in Fig. (11). The current multigrid method can get consistent solutions with the implicit
FIG. 10. Illustration of the computational domain for the subsonic flow around flat plate.

TABLE I. Discretization in physical space and velocity space for the flat plate flow simulation.

| Re   | Physical space | N_{BC} | N_{AF} | N_{total} | δy_{min} | N_u × N_v c |
|------|----------------|--------|--------|-----------|-----------|-------------|
| 0.2  |                | 32     | 32     | 3072      | 0.02      | 100 × 100   |
| 0.5, 1|                | 48     | 48     | 6912      | 0.01      | 90 × 90     |
| 2, 5 |                | 48     | 48     | 6912      | 0.005     | 80 × 80     |
| 10, 20, 50 |        | 64     | 48     | 7680      | 0.002     | 70 × 70     |

a The numbers of discrete cells on edges AF, AB and CD are equal, i.e., N_{AF} = N_{AB} = N_{CD}.

b δy_{min} denotes the minimum height of cells near the solid boundary.

c The numbers of discrete velocity points in phase space.

UGKS. It can be seen that the temperature varies more evidently in the cases at the lower Reynolds numbers than that at the higher Reynolds numbers. In order to compare with results of the DSMC method and the information preservation (IP) method obtained from paper[39], the drag coefficient on the plate has been calculated.
FIG. 11. The temperature distribution around the flat plate at different Reynolds numbers. The background with white line: the implicit UGKS solutions; the dashed line: the current multigrid method results.
by an integration of the skin friction coefficient over both sides of the plate, i.e.,

\[ C_d = \frac{1}{L} \int_0^L (C_{f,\text{upper}} + C_{f,\text{lower}}) \, dl, \]  

(48)

where the skin coefficient is computed by

\[ C_f = \frac{\tau_w}{\frac{1}{2} \rho_\infty U_\infty^2}, \]  

(49)

in which the shear stress \( \tau_w \) is the rate of the momentum transferred from gas to the solid wall for unit area. The drag coefficients varying with the Reynolds numbers are shown in Fig. 12. The multigrid UGKS gives acceptable results matching well with the data from both the IP method and the DSMC. Specifically, the UGKS solutions match better with DSMC results in the cases with lower Reynolds numbers while get closer to IP method data in near continuum flow. The fitting formula as a reference in Fig. 12 is computed by

\[ \ln(C_d \cdot Ma) = 0.225 - 0.333 \times \ln^2 \left( \sqrt{Re/Ma^{0.8}} \right) + 0.031 \times \ln^3 \left( \sqrt{Re/Ma^{0.8}} \right). \]  

(50)
The skin friction coefficients distributed on the flat plate are shown in Fig. 13, where the solutions at $Re = 10$ and $50$ are compared with that from the DSMC and results in Fig. 13(a), and the rest results are compared with those from the implicit UGKS in Fig. 13(b). Good agreements have been obtained between the present results and the reference data.

In order to illustrate the efficiency of the multigrid method, we plot the convergence history of the implicit scheme and the multigrid method in Fig. 14. The calculation stops when the mean squared residuals as defined in Eq. (47) get lower than $1.0 \times 10^{-6}$. For a better description, the iteration steps and the total CPU time that cost by the implicit UGKS, three-level, four-level and five-level multigrid UGKS are listed in Table II and Table III. Generally, the multigrid method can speed up
FIG. 14. The convergence history of the implicit UGKS and the multigrid UGKS for the flow around the flat plate, indicated by the variation of the residual of X-momentum at different Reynolds numbers.

TABLE II. Total iteration steps cost by the implicit UGKS and multigrid methods.

| Re   | IUGKS | MIUGKS | MIUGKS | MIUGKS | Speedup |
|------|-------|--------|--------|--------|---------|
|      |       | $N_l=3$ | $N_l=4$ | $N_l=5$ |         |
| 0.2  | 118   | 18     | 14     | 14     | 8.4     |
| 0.5  | 245   | 30     | 20     | 18     | 13.6    |
| 1    | 334   | 36     | 24     | 23     | 14.5    |
| 2    | 504   | 52     | 37     | 34     | 14.8    |
| 5    | 841   | 79     | 61     | 55     | 15.3    |
| 10   | 1285  | 127    | 100    | 92     | 14.0    |
| 20   | 1845  | 176    | 142    | 130    | 14.2    |
| 50   | 2854  | 260    | 211    | 195    | 14.6    |
TABLE III. Total CPU time (min) cost by the implicit UGKS and multigrid methods.

| Re | IUGKS | MIUGKS | Speedup |
|----|-------|--------|---------|
| 0.2| 13.5  | 3.3    | 2.5     | 2.5     | 5.3     |
| 0.5| 51.5  | 9.8    | 6.6     | 6.0     | 8.6     |
| 1  | 70.2  | 11.8   | 8.0     | 7.6     | 9.2     |
| 2  | 83.0  | 13.5   | 9.7     | 8.9     | 9.3     |
| 5  | 138.6 | 20.5   | 16.0    | 14.4    | 9.6     |
| 10 | 179.2 | 27.9   | 22.1    | 20.4    | 8.8     |
| 20 | 257.2 | 38.7   | 31.5    | 28.9    | 8.9     |
| 50 | 398.8 | 57.1   | 46.7    | 43.2    | 9.2     |

the computation by about 9 times for the case with five levels of grids. From these tables it can be known that the CPU time cost by the multigrid method (e.g., with 5-level grids) is about 57% more than the implicit UGKS, due to the consumption on the coarser grids and interpolation between different grid levels, and the multismoothing process on each level. In the current test case, the convergence speed gets reduced with the increase of the Reynolds number.

C. Hypersonic flow past a square cylinder

Following the research in paper[40], the multigrid implicit UGKS is used to study hypersonic flow around a square cylinder. The freestream is argon gas at $Ma = 5$ with an initial temperature of $T_\infty = 273K$. The Knudsen number of the freestream is 0.1, defined relative to the diameter of the square cylinder by the VHS model with $\omega = 0.81$. The solid surface of the square cylinder is taken as isothermal wall with
a fixed temperature of $T_w = 273K$. Due to the symmetric property, only half of the physical domain is considered.

The physical domain is discretized into 12096 cells with a nearest distance of 0.005m to the square surface. The multiple grids used in this case are shown in Fig. 15. In velocity space, $101 \times 101$ discrete velocity points are used for the integration of distribution function by the Newton-Cotes rules. Since a sudden start of a hypersonic flow in the whole computation domain with the same speed imposes great challenges in the initial simulation at the rear part of the square cylinder, in this case we initialize the rear domain behind the cylinder with $\rho_r = 0.1\rho_\infty$ and $U_r = 0$ initially for the convergence evolution.

Fig. 16 shows the flow field at the steady state, including the distributions of temperature, horizontal velocity and vertical velocity. The results of the multigrid UGKS are compared with the DSMC results obtained by dsmeFoam in 40. The present results agree well with the reference ones for each contour. The normalized surface quantities, such as normal pressure, shear stress, and heat flux, are plotted in Fig. 17. And the distribution of the flow variables along the symmetric axis in the upstream are presented in Fig. 18 and compared with DSMC results. Basically, the present results obtained from the multigrid UGKS match well with the reference data.

For this case, the numerical time step is employed, which increases exponentially
FIG. 16. Steady state of the hypersonic flow around the square cylinder at $Ma = 5$ for (a) temperature, (b) X-velocity and (c) Y-velocity. Upper: the current multigrid UGKS. Lower: DSMC data computed by dsmcFoam\textsuperscript{41}.

With iteration steps by

\[ \Delta t_n = a^n \Delta t_p. \]

During calculations, it is found that the multigrid method with multiple smoothing steps is more robust than the original implicit UGKS. So $a = 3$ and $a = 1.2$ are used for the multigrid UGKS and the implicit scheme, respectively. The convergence history is plotted in Fig. 19. In this case, the full multigrid method is used to give a better initial approximate solution for finer grids. It can be observed that the full multigrid method does improve the convergence efficiency and it is about 5 times faster than the implicit UGKS on a single grid for this case. For the same case, the DSMC method obtained by dsmcFoam\textsuperscript{41} takes about 8 hours to get the steady state solution by parallel computing on two server nodes, each of which has two processors of Intel(R) Xeon(R) E5-2680 v3@2.5GHz with 12 cores. However, the multigrid implicit UGKS needs only 50 minutes by serially computing on a single machine with a CPU of Intel(R) Core(TM) i5-4570 CPU@3.2GHz. The improvement of the efficiency in the current scheme in comparison with the DSMC method is significant.
FIG. 17. Distribution of the surface quantities along the square cylinder wall from the stagnation point to the trailing edge, (a) pressure, (b) shear stress and (c) heat flux. The pressure and shear stress are normalized by $\rho_\infty C^2_\infty$, and the heat flux is normalized by $\rho_\infty C^3_\infty$ where $C_\infty = \sqrt{2RT_\infty}$.

In the current high speed flow study, the MIUGKS is about 300 times faster than the DSMC.

V. CONCLUSIONS

In this paper, we present a geometric multigrid method for the implicit unified gas-kinetic scheme for rarefied flow computations. Both stages in the implicit UGKS, i.e., the prediction of equilibrium state and the evolution of distribution function, are treated with multigrid techniques. In prediction step, the governing equations of macroscopic conservative flow variables are solved by the full approximation scheme on each level of grid. While in the evolution step, the governing equations of microscopic distribution function are solved by the correction scheme, by which the distribution function is not required on coarser grids so that the increasing of the computational cost can be well controlled. With a recursive definition of the FAS cy-
FIG. 18. Flow variables along the symmetric axis in the upstream for (a) density, (b) velocity and (c) temperature. The density, velocity and temperature are normalized by $\rho_\infty$, $\sqrt{2RT_\infty}$ and $T_\infty$ respectively.

In circle and CS cycle, the multigrid method for the implicit UGKS has been constructed from the two-grid method.

The multigrid implicit UGKS has been applied to the study of non-equilibrium flows, such as lid-driven cavity flow at different Knudsen numbers, subsonic flow around a plate, and the hypersonic flow past a square cylinder, and the accuracy and efficiency of the multigrid method has been well demonstrated. In comparison with the implicit UGKS with a single level of grid, which is already hundreds times faster than the explicit UGKS, the convergence efficiency of multigrid implicit UGKS has been further improved in all flow regimes from low and high speed flows. In general, the multigrid UGKS with 5-level grids takes about 57% more CPU time than the implicit UGKS with a single level of grid within one iteration step, but overall it is about 5 to 9 times more efficient than the implicit scheme. As a further development, the AMG technique can be employed here as well to remove the generation of multiple grids. The multigrid UGKS can be also developed on unstructured mesh for the applications with complex geometry.
FIG. 19. Convergence history of the flow around a square cylinder at $Ma = 5$.

For rarefied flow computations, especially for the hypersonic flow, the DSMC is currently the dominant method in the engineering rarefied flow applications. However, with the implementation of the implicit and multigrid techniques, the UGKS becomes more efficient than the DSMC method, at least in all cases presented in this paper. Even for the high speed flow at Mach number 5 and Knudsen number 0.1, the UGKS is two orders of magnitude more efficient than the DSMC method. For low speed flows in the transition and near continuum regimes, the efficiency differences between UGKS and DSMC get even larger. With the implementation of acceleration techniques, such as implicit, preconditioning, local time, and multigrid, the UGKS becomes an accurate, reliable, and efficient method for rarefied flow computations. It has been successfully used in rarefied flow applications\textsuperscript{22}, and been extended to other non-equilibrium transport processes, such as radiative transfer and plasma\textsuperscript{12,43}. With computational efficiency increase, the equation-based flow solver will become
an alternative choice in the non-equilibrium flow study and practical engineering applications.

ACKNOWLEDGMENTS

The authors would like to thank Mr. Lianhua Zhu for providing the DSMC results for hypersonic flow around the square cylinder, and the detailed computational cost. This work of Zhu and Zhong was supported by National Natural Science Foundation of China (Grant No. 11472219) and National Pre-Research Foundation of China, as well as the 111 Project of China (B17037). The research work of Xu is supported by Hong Kong research grant council (16207715,16211014,620813) and NSFC (91330203, 91530319).

REFERENCES

1K. Xu, Journal of Computational Physics 171, 289 (2001).
2K. Xu and J.-C. Huang, Journal of Computational Physics 229, 7747 (2010).
3K. Xu and C. Liu, Physics of Fluids 29, 026101 (2017).
4L. Mieussens, Journal of Computational Physics 162, 429 (2000).
5G. A. Bird, Molecular gas dynamics and the direct simulation of gas flows (Oxford University Press, USA, 1994).
6S. Chen, K. Xu, C. Lee, and Q. Cai, Journal of Computational Physics 231, 6643 (2012).
7K. Xu, Direct modeling for computational fluid dynamics: construction and application of unified gas-kinetic schemes (World Scientific, Singapore, 2015).
8S. Liu, P. Yu, K. Xu, and C. Zhong, Journal of Computational Physics 259, 96 (2014).
9R. Wang and K. Xu, AIP Conference Proceedings 1628, 970 (2014).
10W. Sun, S. Jiang, K. Xu, and S. Li, Journal of Computational Physics 302, 222 (2015).
11Z. Guo and K. Xu, arXiv:1602.01680.
12C. Liu and K. Xu, arXiv:1609.05291v2 (2017).
13Q. Li and S. Fu, International Journal of Computational Fluid Dynamics 20, 453 (2006).
14K. Xu, M. Mao, and L. Tang, Journal of Computational Physics 203, 405 (2005).
15J. Jiang and Y. Qian, Computers & Fluids 66, 21 (2012).
16W. Li, M. Kaneda, and K. Suga, Computers & Fluids 93, 100 (2014).
17J. Yang and J. Huang, Journal of Computational Physics 120, 323 (1995).
18M. Mao, D. Jiang, J. Li, and X. Deng, Chinese Journal of Theoretical and Applied Mechanics 47, 822 (2015).
19L. Mieussens, Mathematical Models and Methods in Applied Sciences 10, 1121 (2000).
20Y. Zhu, C. Zhong, and K. Xu, Journal of Computational Physics 315, 16 (2016).
21R. P. Fedorenko, USSR Computational Mathematics and Mathematical Physics 1, 1092 (1962).
22R. P. Fedorenko, USSR Computational Mathematics and Mathematical Physics 4, 227 (1964).
23A. Brandt, Mathematics of computation 31, 333 (1977).
24J. Blazek, Computational fluid dynamics: principles and applications (Elsevier, 2001).
25A. Jameson, Applied Mathematics and Computation 13, 327 (1983).
26S. Yoon and A. Jameson, AIAA-87-0600.
27K. Xu, L. Martinelli, and A. Jameson, Journal of Computational Physics 120, 48 (1995).
A. Brandt and O. E. Livne, *Multigrid techniques: 1984 guide with applications to fluid dynamics (Revised Edition)* (SIAM, 2011).

U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid* (Academic Press, 2000).

K. Stüben and U. Trottenberg, in *Multigrid methods* (Springer, 1982) pp. 1–176.

P. Wesseling, *An introduction to multigrid methods* (Wiley, New York, 1992).

D. J. Mavriplis, “Multigrid techniques for unstructured meshes.” Tech. Rep. (DTIC Document, 1995).

J.-C. Huang, K. Xu, and P. Yu, Communications in Computational Physics 12, 662 (2012).

R. Chen and Z. Wang, AIAA Journal 38, 2238 (2000).

A. Jameson and S. Yoon, AIAA Journal 25, 929 (1987).

S. Yoon and A. Jameson, AIAA Journal 26, 1025 (1988).

K. Stüben, Journal of Computational and Applied Mathematics 128, 281 (2001).

B. John, X.-J. Gu, and D. R. Emerson, Computers & Fluids 45, 197 (2011).

Q. Sun and I. D. Boyd, AIAA Journal 42, 1066 (2004).

S. Chen, C. Zhang, L. Zhu, and Z. Guo, Science Bulletin 62, 119 (2017).

T. Scanlon, E. Roohi, C. White, M. Darbandi, and J. Reese, Computers & Fluids 39, 2078 (2010).

D. Jiang, PhD Thesis, China Aerodynamics Research and Development Center (2016).

W. Sun, S. Jiang, and K. Xu, Communications in Computational Physics, to appear (2017).