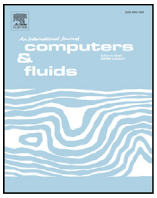
Computers and Fluids 155 (2017) 9–21

Contents lists available at ScienceDirect

Computers and Fluids

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



DUGKS simulations of three-dimensional Taylor–Green vortex flow and turbulent channel flow

Yuntian Bo[a], Peng Wang[b], Zhaoli Guo[b], Lian-Ping Wang[b,c,*]

[a] Department of Energy and Resource Engineering, Peking University, Beijing, PR China
[b] State Key Laboratory of Coal Combustion, Huazhong University of Science and Technology, Wuhan, P.R. China
[c] Department of Mechanical Engineering, 126 Spencer Laboratory, University of Delaware, Newark, Delaware 19716-3140, USA

ARTICLE INFO

Article history:
Received 3 March 2016
Revised 10 February 2017
Accepted 6 March 2017
Available online 7 March 2017

Keywords:
Discrete unified gas kinetic scheme
Lattice Boltzmann equation
Pseudo-spectral methods
3D Taylor–Green vortex flow
Turbulent channel flow

ABSTRACT

The discrete unified gas-kinetic scheme (DUGKS) is a relatively new, finite-volume formulation of the Boltzmann equation. It has several advantages over the lattice Boltzmann method (LBM) in that it can naturally incorporate multiscale physical processes and non-uniform lattice mesh. With the goal of simulating a variety of turbulent flows, we investigate two aspects of DUGKS. First, we explore a parallel implementation strategy of DUGKS using domain decomposition and MPI (Message Passing Interface), and demonstrate the scalability of the parallel DUGKS code. We validate the resulting parallel code using the 3D Taylor-green vortex flow where small eddies are generated over time from large eddies. The DUGKS results are compared to short-time analytical solution as well as to those from LBM and pseudo-spectral method. The second-order accuracy of DUGKS is confirmed by using the highest-resulotion DUGKS flow as the benchmark. Second, we consider how to incorporate solid walls and non-uniform mesh in DUGKS for three-dimensional flows, by simulating a turbulent channel flow. The statistics of the simulated turbulent channel flow are compared to those based on LBM and spectral methods. It is found that the DUGKS results, even with a coarse non-uniform mesh, are overall better than the LBM results when compared to the spectral benchmark data.

1. Introduction

One of the major advances in computational fluid dynamics (CFD) in the last three decades is the emergence, development, and applications of mesoscopic numerical methods based on solving the kinetic Boltzmann equation with a linearized collision model. As the most popular mesoscopic method for simulating nearly incompressible flows, the lattice Boltzmann method (LBM) has received a great deal of attention due to its simplicity, high computational efficiency, and capability of treating a variety of complex flows such as flows through porous media, multiphase flows with moving fluid-fluid or fluid-solid interfaces, and turbulent flows [1–7]. Although its spatial accuracy is of only the second-order, the low numerical dissipation of LBM [8,9] makes LBM a competitive CFD tool for simulating turbulent flows, and its physical accuracy has been demonstrated by comparing with the N-S based pesudo-spectral methods [7,10,11]. The reason for low numerical dissipation in LBM is the accurate treatment (by Lagrangian streaming) of the advection term. Unlike the conventional CFD methods which solve the strongly nonlinear Navier–Stokes equations, the LBM solves essentially a linear equation with local nonlinearity residing in its collision term. In LBM, the space and time discretizations are fully coupled to each other and to a highly symmetric set of discrete kinetic particle velocities. This highly efficient treatment, however, constrains the method to nearly incompressible flow under the continuum limit (*i.e.*, Knudsen number less than 10^{-3}) or, through extensions, to weak nonequilibrium flows [12–15].

More general kinetic schemes that can handle flows at all Knudsen numbers have also been developed in parallel. A great example is the unified gas kinetic scheme (UGKS) [16,17] which is designed as a multiscale simulation tool that can treat, within a same framework, both the continuum limit (i.e., the N-S equations) and the free-molecular limit [18]. In this framework, the physical modeling is directly performed at the scales of the spatial mesh size and time step size, with the use of a sufficient number of discrete kinetic velocities. The collisions of kinetic particles within the grid volume and the particle transport through the surfaces of the grid volume together determine the time evolution of the particle distribution functions.

[*] Corresponding author.
E-mail addresses: boyuntian@pku.edu.cn (Y. Bo), sklccwangpeng@hust.edu.cn (P. Wang), zlguo@hust.edu.cn (Z. Guo), lwang@udel.edu (L.-P. Wang).

http://dx.doi.org/10.1016/j.compfluid.2017.03.007
0045-7930/© 2017 Elsevier Ltd. All rights reserved.

Recently, a discrete unified gas kinetic scheme (DUGKS) [19,20] has been developed, combining the advantages of both LBM and UGKS methods. DUGKS is designed as a finite-volume scheme with flexible mesh adaptation. The key is the improved implementation of the calculations of surface transport fluxes through a transformation of the distribution function coupled with the effect of particle collisions. By using extended equilibrium distributions, DUGKS has been shown to be capable of simulating flows at all Knudsen numbers including thermal compressible flows [20] and Boussinesq flows [21]. This greatly extends the potential applications of discrete-velocity based mesoscopic schemes.

A potential application of DUGKS is direct numerical simulation of complex turbulent flows. Due to its finite-volume formulation and its feasibility of incorporating unstructured mesh [22], DUGKS could be potentially more efficient in treating inhomogeneous and wall-bounded turbulent flows. For laminar flow simulations, DUGKS has been shown to have a similar accuracy as LBM [23], but DUGKS is numerically much more stable when the flow is under-resolved. For three-dimensional time-dependent turbulent flows, a first comparative study of DUGKS and LBM was reported by Wang et al. [24], who concluded that DUGKS is capable of simulating decaying homogeneous isotropic turbulence, but is slightly more dissipative when compared to LBM. We must note that DUGKS does not employ the usual finite-volume formulation for the advection term. The novelty of DUGKS is that the flux at the cell interface is computed using the distribution functions that couple the advection and collision, namely, the flux is evaluated by solving the Boltzmann evolution equation rather than by interpolation. This coupling strategy in DUGKS ensures low numerical dissipation, as demonstrated in [25,26]. More specifically, in our recent paper [24], we show that DUGKS is capable of simulating a decaying turbulent flow (without walls) and the resolution requirement ($k_{\max}\eta > 3$) for DUGKS is only slightly higher than that for LBM ($k_{\max}\eta > 2$), where $k_{\max}$ and η are the largest wavenumber resolved and the flow Kolmogorov length, respectively.

A critical step in extending the DUGKS applications to three-dimensional turbulent flows is the scalable implementation. In this paper, we report a first parallel implementation of DUGKS using domain decompositions. The resulting parallel code is then validated using the three-dimensional energy-cascading Taylor–Green vortex flow for which a short-time analytical solution exists. Comparisons with pseudo-spectral simulation results and LBM results are also provided. Furthermore, we will demonstrate, for the first time, a successful DUGKS simulation of a turbulent channel flow using a relative coarse non-uniform mesh. It should be noted that the efficient parallel implementations of LBM have been widely studied, taking advantage of the strict local data communication in LBM. The key is to explore the best use of data structure and cache memory, and several implementation methods are available [27–32]. The nature of data communication in DUGKS depends on the exact algorithm used to compute the surface transport fluxes.

The remainder of the paper is organized as follows. In Section 2, the basic algorithm of DUGKS is described, along with parallel implementation details. The treatment of no-slip boundary and the formulation of mesoscopic forcing are also developed. The numerical results and scalability data are presented in Section 3. Conclusions and future outlook are given in Section 4.

2. The DUGKS algorithm and parallel implementation

In this section, we describe the basic DUGKS algorithm, the method to parallelize the code, and implementation of the non-uniform, time-dependent forcing.

2.1. The DUGKS algorithm

DUGKS begins with the Boltzmann equation with the BGK collision model. In this paper, we only consider the incompressible flow limit or flows with a small Mach number [19,23]. In this case, the same discrete kinetic velocities used in the D3Q19 LBM model can be adopted in DUGKS, namely,

$$\boldsymbol{\xi}_\alpha = \begin{cases} (0,0,0)c, & \alpha = 0 \\ (\pm1,0,0)c,\ (0,\pm1,0)c,\ (0,0,\pm1)c, & \alpha = 1-6 \\ (\pm1,\pm1,0)c,\ (\pm1,0,\pm1)c,\ (0,\pm1,\pm1)c, & \alpha = 7-18 \end{cases} \tag{1}$$

where the speed of the sound is $c = \sqrt{3RT}$, R is the gas constant, T is the temperature. The discrete distribution function $f_\alpha(\mathbf{x}, t)$ satisfies the following equation,

$$\frac{\partial f_\alpha}{\partial t} + \boldsymbol{\xi}_\alpha \cdot \nabla f_\alpha = \Omega_\alpha \equiv \frac{f_\alpha^{(eq)} - f_\alpha}{\tau} + \Phi_\alpha, \tag{2}$$

where τ is the relaxation time and Φ_α denotes the mesoscopic forcing used to represent the local nonuniform force $\mathbf{F}(\mathbf{x}, t)$ per unit volume in the Navier–Stokes equation.

At this point, we recall that the discrete-velocity Boltzmann equation with the BGK collision model is *designed* to reproduce the Navier–Stokes equations, by expanding the distribution function in terms of a set of Hermite ortho-normal polynomials in velocity space. This expansion allows the distribution function to be approximated in terms of conservative hydrodynamic variables and their derivatives [15]. With the BGK collision model, the Chapman–Enskog procedure implies that the continuity and Navier–Stokes (*i.e.*, macroscopic) equations depend only on a few leading moments of the distribution function instead of the full form of the distribution function. Therefore, truncation of the higher-order terms in the Hermite expansion of the distribution function has no explicit effect on the macroscopic equations [15]. Specifically, the explicit Hermite expansion of the Maxwelliam equilibrium distribution at low Mach number (or isothermal limit), up to the second order in the Hermite expansion, is [15]

$$f_\alpha^{(eq)} = \rho W_\alpha \left[1 + \frac{\boldsymbol{\xi}_\alpha \cdot \mathbf{u}}{RT} + \frac{(\boldsymbol{\xi}_\alpha \cdot \mathbf{u})^2}{2(RT)^2} - \frac{\mathbf{u}\cdot\mathbf{u}}{2RT} \right], \tag{3}$$

where the weighting coefficients W_α are $W_0 = 1/3$, $W_{1,\cdots,6} = 1/18$, and $W_{7,\cdots,18} = 1/36$; ρ is the fluid density; and $\mathbf{u}$ is the hydrodynamic fluid velocity.

The body force term should be formulated in the spirit of the Hermite approximation. According to the original Boltzmann equation, the forcing term Φ in the continuous-velocity form of the Boltzmann equation is [15,33]

$$\Phi(\mathbf{x}, \boldsymbol{\xi}, t) = -\frac{F_j}{\rho} \frac{\partial f(\mathbf{x}, \boldsymbol{\xi}, t)}{\partial \boldsymbol{\xi}_j}, \tag{4}$$

where $f(\mathbf{x}, \boldsymbol{\xi}, t)$ is the continuous distribution function. The term $\partial f(\mathbf{x}, \boldsymbol{\xi}, t)/\partial \boldsymbol{\xi}_j$ should be projected onto the truncated Hermite basis. Shan et al. [15] showed explicitly, using our notations and up to the second order in the Hermite expansion, that

$$\Phi_\alpha = \frac{F_j W_\alpha}{RT} \left[\left(\boldsymbol{\xi}_{\alpha,j} - u_j\right) + \boldsymbol{\xi}_{\alpha,j} \frac{\boldsymbol{\xi}_\alpha \cdot \mathbf{u}}{RT} \right]. \tag{5}$$

In view of the leading-order terms in $f_\alpha^{(eq)}$, the above formulation can be written alternatively as

$$\Phi_\alpha = \frac{F_j}{\rho} \left[\frac{\left(\boldsymbol{\xi}_{\alpha,j} - u_j\right)}{RT} f_\alpha^{(eq)} + \mathcal{O}\left(\mathbf{u}^2\right) \right] \approx \frac{\mathbf{F} \cdot \left(\boldsymbol{\xi}_{\alpha,j} - u_j\right)}{\rho RT} f_\alpha^{(eq)}. \tag{6}$$

The $\mathcal{O}(\mathbf{u}^2)$ error term in Eq. (6) leads to an error of order $\mathcal{O}(\mathbf{u}^3)$ in the Navier–Stokes equation, which is usually ignored under the assumption of low Mach number. It is noted that the approximation, Eq. (6), can also be derived if we start with Eq. (4) and approximate $\partial f/\partial\xi_j$ by $\partial f^{(eq)}/\partial\xi_j$ [15,33]. We now write

$$\Phi_\alpha(\mathbf{x},t) \approx \phi_\alpha f_\alpha^{(eq)}(\mathbf{x},t), \qquad \text{with} \;\; \phi_\alpha \equiv \frac{F_j(\boldsymbol{\xi}_{\alpha,j}-u_j)}{\rho RT}. \tag{7}$$

The consistency with the Navier–Stokes equations requires

$$\sum_\alpha \Phi_\alpha = 0, \quad \sum_\alpha \boldsymbol{\xi}_{\alpha,j}\Phi_\alpha = F_j,$$
$$\sum_\alpha \boldsymbol{\xi}_{\alpha,j}\boldsymbol{\xi}_{\alpha,k}\Phi_\alpha = u_jF_k + u_kF_j + \mathcal{O}(\mathbf{u}^3), \tag{8}$$

which can be easily shown to be satisfied by the two alternative forms, Eq. (5) and Eq. (7). This consistency has also been demonstrated in the LBM framework in [6].

The hydrodynamic variables, namely, density ρ, the fluid macrosopic velocity $\mathbf{u}$, and pressure p are obtained as

$$\rho = \sum_i f_\alpha, \qquad \rho\mathbf{u} = \sum_\alpha \xi_\alpha f_\alpha, \qquad p = \rho RT. \tag{9}$$

The DUGKS algorithm makes use of a finite-volume formulation of Eq. (2) [19,20]. The advection term in Eq. (2) is treated as sum of surface fluxes after applying the divergence theorem. This treatment allows the lattice velocities to be decoupled from the grid structure. For the time integration, the collision term is treated by the trapezoidal rule, similar to the lattice Boltzmann method. The surface flux terms by the midpoint rule (other alternative formulations may be used). The details are now explained below.

Consider here a cuboid grid cell centered at $\mathbf{x}_{i,\,j,\,k} \equiv (x_i,\, y_j,\, z_k)$, with cell sizes Δx_i, Δy_j, and Δz_k, respectively, in the x, y, and z directions. The time evolution of the discrete distribution averaged over the cell volume $V_{i,j,k} = \Delta x_i\Delta y_j\Delta z_k$, $f_\alpha(\mathbf{x}_{i,\,j,\,k},\, t)$, is advanced from $t = t_n$ to $t = t_{n+1}$ as,

$$\tilde{f}_\alpha(\mathbf{x}_{i,j,k}, t_{n+1}) = \tilde{f}_\alpha^{+}(\mathbf{x}_{i,j,k}, t_n) - \frac{\Delta t}{V_j} Q_\alpha(\mathbf{x}_{i,j,k}, t_{n+0.5}), \tag{10}$$

where, for convenience, the transformed variables are defined as

$$\tilde{f}_\alpha = f_\alpha - \frac{\Delta t}{2}\Omega_\alpha, \quad \tilde{f}_\alpha^{+} = f_\alpha + \frac{\Delta t}{2}\Omega_\alpha. \tag{11}$$

The code computes mainly the transformed variable $\tilde{f}_\alpha$, all other forms of the distribution function can be expressed in terms of $\tilde{f}_\alpha$ and $f_\alpha^{(eq)}$, as shown below.

The net flux $Q_\alpha(\mathbf{x}_{i,j,k}, t_{n+0.5})$ is defined as

$$\begin{aligned} Q_\alpha(\mathbf{x}_{i,j,k}, t_{n+0.5})dS &= \int_{\partial V_j} (\boldsymbol{\xi}_\alpha\cdot\mathbf{n}) f_\alpha(\mathbf{x}_b, t_{n+0.5}) \\ &= \big[f_\alpha(\mathbf{x}_{i+,j,k}, t_{n+0.5}) - f_\alpha(\mathbf{x}_{i-,j,k}, t_{n+0.5})\big]\boldsymbol{\xi}_{\alpha,x}\Delta y_j\Delta z_k \\ &\quad + \big[f_\alpha(\mathbf{x}_{i,j+,k}, t_{n+0.5}) - f_\alpha(\mathbf{x}_{i,j-,k}, t_{n+0.5})\big]\boldsymbol{\xi}_{\alpha,y}\Delta x_i\Delta z_k \\ &\quad + \big[f_\alpha(\mathbf{x}_{i,j,k+}, t_{n+0.5}) - f_\alpha(\mathbf{x}_{i,j,k-}, t_{n+0.5})\big]\boldsymbol{\xi}_{\alpha,z}\Delta x_i\Delta y_j, \end{aligned} \tag{12}$$

where $f_\alpha(\mathbf{x}_b, t_{n+0.5})$ is the distribution at the cell boundary or interface $\mathbf{x}_b$ at the half time step, $\mathbf{x}_{i+,j,k} = \mathbf{x}_{i,j,k} + 0.5\Delta x_i\mathbf{e}_x$, $\mathbf{x}_{i-,j,k} = \mathbf{x}_{i,j,k} - 0.5\Delta x_i\mathbf{e}_x$, $\mathbf{x}_{i,j+,k} = \mathbf{x}_{i,j,k} + 0.5\Delta y_j\mathbf{e}_y$, $\mathbf{x}_{i,j-,k} = \mathbf{x}_{i,j,k} - 0.5\Delta y_j\mathbf{e}_y$, $\mathbf{x}_{i,j,k+} = \mathbf{x}_{i,j,k} + 0.5\Delta z_k\mathbf{e}_z$, $\mathbf{x}_{i,j,k-} = \mathbf{x}_{i,j,k} - 0.5\Delta z_k\mathbf{e}_z$, and $\mathbf{e}_x$, $\mathbf{e}_y$, $\mathbf{e}_z$ are unit vectors in x, y, and z directions, respectively.

The key step is then to properly evaluate F_α. In DUGKS, after integrating Eq. (2) along a particle path from t_n to $t_n + h$ ($h \equiv 0.5\Delta t$) with the collision term included, the distribution function at the cell boundary at $t_n + h$ can be written as

$$\overline{f}_\alpha(\mathbf{x}_b, t_n + h) = \overline{f}_\alpha^{+}(\mathbf{x}_b - h\,\boldsymbol{\xi}_\alpha, t_n), \tag{13}$$

where a second set of transformed variables are defined as

$$\overline{f}_\alpha = f_\alpha - \frac{h}{2}\Omega_\alpha, \quad \overline{f}_\alpha^{+} = f_\alpha + \frac{h}{2}\Omega_\alpha. \tag{14}$$

It is important to note that the evolution equation, Eq. (13), is exact, namely, no numerical dissipation is introduced at this stage between two different times, t_n and $t_n + h$. Next, we need an approximation of $\overline{f}_\alpha^{+}(\mathbf{x}_b - h\,\boldsymbol{\xi}_\alpha, t_n)$ at a single time t_n using spatial interpolation. There are many possibilities, such as the spectral interpolation or high-order compact finite-difference schemes. For simplicity, here we choose to utilize a local Taylor expansion, namely,

$$\overline{f}_\alpha(\mathbf{x}_b, t_n + h) = \overline{f}_\alpha^{+}(\mathbf{x}_b - h\,\boldsymbol{\xi}_\alpha, t_n) \approx \overline{f}_\alpha^{+}(\mathbf{x}_b, t_n) - h\,\boldsymbol{\xi}_\alpha\cdot\nabla\overline{f}_\alpha^{+}(\mathbf{x}_b, t_n), \tag{15}$$

where $\overline{f}_\alpha^{+}(\mathbf{x}_b, t_n)$ and its gradients $\nabla\overline{f}_\alpha^{+}(\mathbf{x}_b, t_n)$ at the cell interfaces can be approximated by the second-order linear interpolations from values at the cell centers. For example,

$$\begin{aligned} \frac{\partial\overline{f}_\alpha^{+}(\mathbf{x}_{i+,j,k}, t_n)}{\partial x} &\approx \frac{\overline{f}_\alpha^{+}(\mathbf{x}_{i+1,j,k}, t_n) - \overline{f}_\alpha^{+}(\mathbf{x}_{i,j,k}, t_n)}{x_{j+1} - x_j} \\ &= \frac{\overline{f}_\alpha^{+}(\mathbf{x}_{i+1,j,k}, t_n) - \overline{f}_\alpha^{+}(\mathbf{x}_{i,j,k}, t_n)}{(\Delta x_i + \Delta x_{i+1})/2}, \end{aligned} \tag{16a}$$

$$\overline{f}_\alpha^{+}(\mathbf{x}_{i+,j,k}, t_n) \approx \overline{f}_\alpha^{+}(\mathbf{x}_{i,j,k}, t_n) + \frac{\partial\overline{f}_\alpha^{+}(\mathbf{x}_{i+,j,k}, t_n)}{\partial x}\frac{\Delta x_i}{2}, \tag{16b}$$

where

$$\begin{aligned} \overline{f}_\alpha^{+}(\mathbf{x}_{i,j,k}, t_n) &= \frac{2\tau - h}{2\tau + \Delta t}\tilde{f}_\alpha(\mathbf{x}_{i,j,k}, t_n) \\ &\quad + \left(\frac{3h}{2\tau + \Delta t} + \frac{3h\tau}{2\tau + \Delta t}\phi_\alpha\right) f_\alpha^{(eq)}(\mathbf{x}_{i,j,k}, t_n). \end{aligned} \tag{17}$$

In Eq. (13), the half time step values at the cell interface included the particle collision effect. This coupled treatment of advection and collision makes DUGKS a self-adaptive scheme for both the continuum regime and the free-molecular regime [19].

Clearly, the hydrodynamic variables can also be evaluated using any of the transformed distributions, namely,

$$\rho = \sum_\alpha \tilde{f}_\alpha = \sum_\alpha \tilde{f}_\alpha^{+} = \sum_\alpha \overline{f}_\alpha = \sum_\alpha \overline{f}_\alpha^{+}, \tag{18a}$$

$$\begin{aligned} \rho\mathbf{u} &= \sum_\alpha \xi_\alpha\tilde{f}_\alpha + \frac{dt}{2}\mathbf{F} = \sum_\alpha \xi_\alpha\tilde{f}_\alpha^{+} - \frac{dt}{2}\mathbf{F} = \sum_\alpha \xi_\alpha\overline{f}_\alpha + \frac{h}{2}\mathbf{F} \\ &= \sum_\alpha \xi_\alpha\overline{f}_\alpha^{+} - \frac{h}{2}\mathbf{F}. \end{aligned} \tag{18b}$$

Therefore, $\rho(\mathbf{x}_b, t_n + h)$ and $\mathbf{u}(\mathbf{x}, t_n + h)$ can be obtained from $\overline{f}_\alpha(\mathbf{x}_b, t_n + h)$, which can be used to compute $f_\alpha^{(eq)}(\mathbf{x}_b, t_n + h)$. Finally, the desired distributions at half time step at the cell interface can be readily obtained by

$$\begin{aligned} f_\alpha(\mathbf{x}_b, t_n + h) &= \frac{2\tau}{2\tau + h}\overline{f}_\alpha(\mathbf{x}_b, t_n + h) \\ &\quad + \left(\frac{h}{2\tau + h} + \frac{\tau h}{2\tau + h}\phi_\alpha\right) f_\alpha^{(eq)}(\mathbf{x}_b, t_n + h). \end{aligned} \tag{19}$$

which can be used to compute the net flux $F_\alpha(\mathbf{x}_{i,j,k}, t_n + h)$ using Eq. (12). This then allows the distribution function to be advanced through Eq. (10).

In the above formulation, we have defined five sets of discrete distributions: f_α, $\tilde{f}_\alpha$, $\tilde{f}_\alpha^{+}$, $\overline{f}_\alpha$, $\overline{f}_\alpha^{+}$. Any one of the five can be used to determine the equilibrium distribution $f_\alpha^{(eq)}$ and then the remaining four. The most convenient choice is to follow the evolution of $\tilde{f}_\alpha$, which can be used to determine others as follows [19]

$$f_\alpha = \frac{2\tau}{2\tau + \Delta t}\tilde{f}_\alpha + \left(\frac{\Delta t}{2\tau + \Delta t} + \frac{\tau\Delta t}{2\tau + \Delta t}\phi_\alpha\right) f_\alpha^{(eq)}, \tag{20a}$$

$$\bar{f}_\alpha = \frac{2\tau + h}{2\tau + \Delta t}\tilde{f}_\alpha + \left(\frac{h}{2\tau + \Delta t} + \frac{\tau h}{2\tau + \Delta t}\phi_\alpha\right) f_\alpha^{(eq)}, \tag{20b}$$

$$\begin{aligned}\tilde{f}_\alpha^{+} &= \frac{2\tau - \Delta t}{2\tau + \Delta t}\tilde{f}_\alpha + \left(\frac{2\Delta t}{2\tau + \Delta t} + \frac{2\Delta t\tau}{2\tau + \Delta t}\phi_\alpha\right) f_\alpha^{(eq)} \\ &= \frac{4}{3}\bar{f}_\alpha^{+} - \frac{1}{3}\tilde{f}_\alpha,\end{aligned} \tag{20c}$$

in addition to the relation Eq. (17).

It must be noted that the mid-point treatment (in time) of the advection term is only one of the many possibilities. Likewise, there are other possibilities of treating the off-grid distributions at $t = t_n$ in Eq. (15) [19,20]. When compared to the treatment of the nonlinear advection term in the conventional Navier–Stokes solvers, we observe two important differences: (1) the advection or related flux term in DUGKS is linear and as such the time evolution from t_n for $t_n + h$, Eq. (13), is formally precise, (2) the interpolation in our treatment is over space only at a single time t_n while the interpolation in the Navier-Stokes solvers involves both space and time. The coupled collision and streaming treatment in Eq. (13) is exact, and the space-only interpolation in Eq. (15) maintains a low numerical dissipation of the overall scheme [25,26]. Should a spectral interpolation be used in Eq. (15), then the numerical dissipation could be further reduced. A mathematical analysis of the numerical dissipation in DUGKS [34] shows that the numerical dissipation depends on both the mesh size and the time step size (or equivalently the CFL number), but is smaller than that uses a direct interpolation without considering the collision term.

2.2. Parallel implementation

The parallel code makes use of two-dimensional domain decomposition. Assume the physical domain is divided into $nx \times ny \times nz$ cells as shown in Fig. 1. We divide the y direction into N_{py} parts and the z direction into N_{pz} parts. A total of $N_p = N_{py}N_{pz}$ processors are used, each handles a sub-domian of $nx \times ly \times lz$ cells, where $ly = ny/N_{py}$ and $lz = nz/N_{pz}$. The main variable for each processor is a four-dimensional (*i.e.*, particle index and *x, y, z* locations) array $\tilde{f}$ [f(0:np - 1,1:nx,1:ly,1:lz)], which is defined at the cell centers, as shown in Fig. 1, where np = 19 is the number of kinetic particles.

The algorithm is summarized using a block flow diagram in Fig. 2, where the large red box marks the main loop. There are 4 main subroutines: interface, flux, evolve, and macrovar. The *interface* subroutine performs the following functions. First $\bar{f}^{+}$ [fb(0:np - 1,0:nx+1,0:ly+1,0:lz+1)] is computed at the cell centers using Eq. (17). Here the hydrodynamic variables at $(\mathbf{x}_{i,\ j,\ k},\ t_n)$ need to be computed to obtain $f^{(eq)}(\mathbf{x}_{i,\ j,\ k},\ t_n)$ in Eq. (17). Then it is extended to include ghost cell centers as shown in Fig. 1, by using the periodic boundary condition, or for no-slip wall by extrapolation. For interior sub-domains, this extension requires data communication with neighboring sub-domains. This extension is the only data communication of the algorithm, which is handled by the MPI (Message Passing Interface) library. Now the values of $\bar{f}^{+}(\mathbf{x}_b, t_n)$ at the cell interface nodes can be computed by interpolation from the values at the cell centers and ghost cell centers. There are three types of interface nodes: $x-$interface nodes, $y-$interface nodes, and $z-$interface nodes; the values of $\bar{f}^{+}(\mathbf{x}_b, t_n)$ at these three types of interface nodes are saved as fix(0:np - 1,0:lx,0:ly+1,0:lz+1), fiy(0:np - 1,0:lx+1,0:ly,0:lz+1), and fiz(0:np - 1,0:lx+1,0:ly+1,0:lz).

The *flux* subroutine performs the computation of $f_\alpha(\mathbf{x}_b, t_n + h)$ using Eqs. (15) and (19). Additional consideration for the no-slip boundary condition will be explained in Section 2.3. The values of $\bar{f}_\alpha(\mathbf{x}_b, t_n + h)$ are saved in a temporary array fi(0:np - 1,0:lx+1,0:ly+1,0:lz+1). Here the hydrodynamic variables at $(\mathbf{x}_b, t_n + h)$ need to be computed to obtain $f^{(eq)}(\mathbf{x}_b, t_n + h)$ in Eq. (19). This process is repeated for the *x*, *y*, and *z* interface nodes. The same arrays, fix, fiy, and fiz, can be used to store $f_\alpha(\mathbf{x}_b, t_n + h)$ at the three types of interface nodes.

The *evolve* subroutine performs the net flux calculation according to Eq. (12). At this point, $\tilde{f}_\alpha(\mathbf{x}_{i,j,k}, t_n + \Delta t)$ can be readily obtained by Eq. (10). Finally, the hydrodynamic variables at $(\mathbf{x}_{i,j,k}, t_n + \Delta t)$ are obtained from $\tilde{f}_\alpha(\mathbf{x}_{i,j,k}, t_n + \Delta t)$ in the *macrovar* subroutine.

The above description shows that only the *interface* subroutine requires local sub-domain data communication. A total of five large arrays are used in the program, which consumes more memory than the LBM approach. The main advantage over LBM is that DUGKS allows the use of any non-uniform and anisotropic grid structure, as the grid structure is not coupled with the particle velocities. Two examples of using non-uniform grids will be provided in Section 3.2 and Section 3.3.

2.3. Implementation of the no-slip boundary condition on the channel walls

Here we explain several other necessary details. First, for a solid wall boundary such as the channel walls inSecs. 3.2 and 3.3, the $\bar{f}^{+}$ values at the ghost cell centers are determined by extrapolation, similar to the extrapolation method used in LBM in [35]. The no-slip boundary condition is done using the concept of bounce-back. However, it is performed on the cell interfaces rather than along lattice links in LBM. This interface surface-based treatment has the potential benefit of eliminating corner singularity, for example, for the case of cavity flow simulations [23]. Consider a cell interface whose outward normal is $\mathbf{m}$, since in the examples we considered in Section 3, the walls are fixed. The general no-slip bounce-back discussed in [19] reduces simply to

$$\bar{f}_\alpha(\mathbf{x}_b, t + h) = f_{\bar{\alpha}}(\mathbf{x}_b, t + h), \quad \text{for all } \boldsymbol{\xi}_\alpha \cdot \mathbf{m} < 0, \tag{21}$$

which should be executed after the reconstruction of $\bar{f}_\alpha(\mathbf{x}, t_n + h)$ in Eq. (13), but before the transformation back to the original variable $f_\alpha(\mathbf{x}, t_n + h)$, Eq. (19). This order ensures a consistent computation of wall velocity needed for the transformation.

While we assumed a structured grid in the above discussions, the finite-volume DUGKS formulation can be extended relatively easily to unstructured grids, as has already been done in [22].

3. Results

In this section, we will perform three-dimensional flow simulations using the parallel DUGKS code. We will compare the results from the pseudo-spectral method and the LBM method. For periodic flows, the pseudo-spectral method has been described in [42,43], and for turbulent channel flow it is presented in [36,37]. The LBM method follows the details provided in [38,39].

3.1. 3D Taylor-Green vortex flow with energy cascade

The 3D Taylor-Green vortex flow [40] is one of the very few analytical solutions of three-dimensional time-dependent Navier–Stokes equations. Starting from a simple incompressible 3D initial condition of the form

$$u(\mathbf{x}, t = 0) = \cos(x)\sin(y)\sin(z), \tag{22a}$$

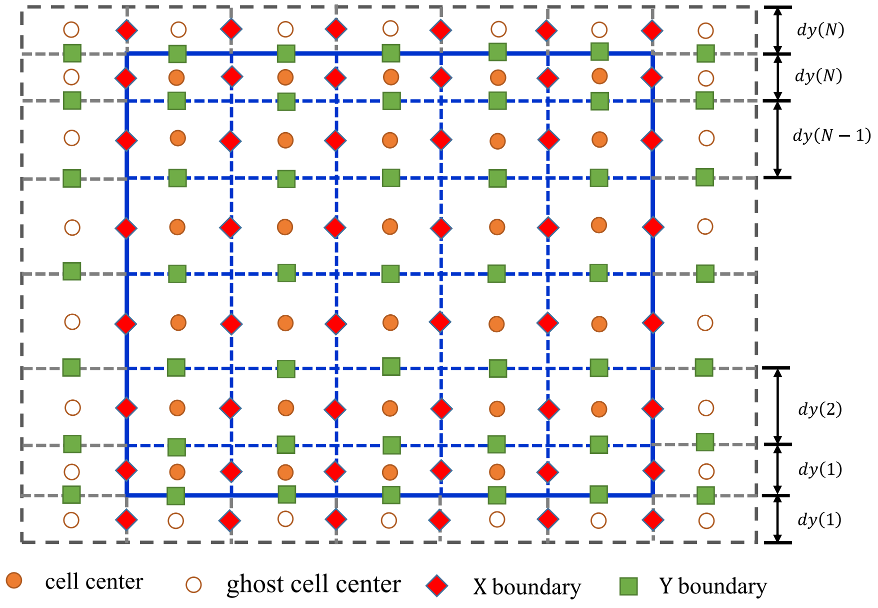


Fig. 1. A sketch showing the locations of cell-center nodes (filled circles) and related boundary nodes (diamonds for boundary nodes along the *x* direction, and squares for boundary nodes along the *y* direction) and the ghost nodes (open circles). The grid spacing can be non-uniform in general. The distributions to be solved are defined only at the cell-center nodes. The distributions at the ghost nodes are extended from the center-center nodes either by periodic boundary conditions or by extrapolation (for a no-slip wall) from the cell-center nodes.

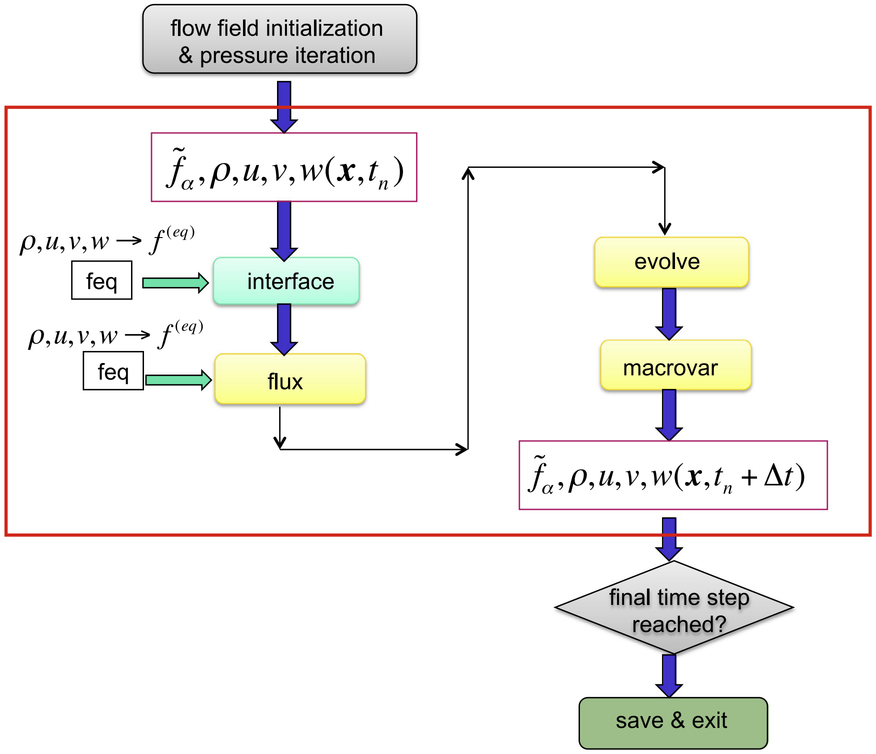

flow field initialization & pressure iteration
$\tilde{f}_\alpha, \rho, u, v, w(\boldsymbol{x}, t_n)$
$\rho, u, v, w \rightarrow f^{(eq)}$
feq
interface
$\rho, u, v, w \rightarrow f^{(eq)}$
feq
flux
evolve
macrovar
$\tilde{f}_\alpha, \rho, u, v, w(\boldsymbol{x}, t_n + \Delta t)$
final time step reached?
save & exit

Fig. 2. Flow diagram of the parallel DUGKS code.

$$v(\mathbf{x}, t = 0) = -\sin(x)\cos(y)\sin(z), \tag{22b}$$

$$w(\mathbf{x}, t = 0) = 0, \tag{22c}$$

and assuming periodic conditions in a cubic domain: $0 \le x \le 2\pi$, $0 \le y \le 2\pi$, $0 \le z \le 2\pi$, the three-dimensional time-dependent Navier–Stokes equation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho_0}\nabla p + \frac{1}{R}\nabla^2 \mathbf{u} \tag{23}$$

can be solved analytically, at small times, using the short-time perturbation expansion. Here all quantities have been properly normalized by the initial maximum velocity magnitude U_0 in the *x* or *y* direction, and $L_B/2\pi$, where L_B is the physical domain size, $R = L_B U_0/(2\pi\nu)$ is the flow Reynolds number divided by (2π), ν

is the kinematic viscosity. The pressure p has been normalized by ρU_0^2.

Taylor and Green (TG37 hereafter) [40] obtained a perturbation expansion of the velocity field, up to $\mathcal{O}(t^5)$. Their solution for the velocity field contains 8 distinct modes and is given as

$$\begin{aligned} u(\mathbf{x},t) = {} & \delta_1 \cos(x)\sin(y)\sin(z) + \alpha_1 \sin(2x)\cos(2z) \\ & + \frac{2}{3}\gamma_2 \cos(3x)\sin(y)\sin(z) \\ & + \gamma_2 \cos(x)\sin(3y)\sin(z) \\ & - \gamma_3 \cos(x)\sin(y)\sin(3z) \\ & - \beta_1 \cos(3x)\sin(y)\sin(3z), \end{aligned} \tag{24a}$$

$$\begin{aligned} v(\mathbf{x},t) = {} & -\delta_1 \sin(x)\cos(y)\sin(z) + \alpha_1 \sin(2y)\cos(2z) \\ & - \gamma_2 \sin(3x)\cos(y)\sin(z) \\ & - \frac{2}{3}\gamma_2 \sin(x)\cos(3y)\sin(z) \\ & + \gamma_3 \sin(x)\cos(y)\sin(3z) \\ & + \beta_1 \sin(x)\cos(3y)\sin(3z), \end{aligned} \tag{24b}$$

$$\begin{aligned} w(\mathbf{x},t) = {} & -\alpha_1[\cos(2x) + \cos(2y)]\sin(2z) \\ & - \gamma_2 \sin(3x)\sin(y)\cos(z) \\ & + \gamma_2 \sin(x)\sin(3y)\cos(z) \\ & + \beta_1 \sin(3x)\sin(y)\cos(3z) \\ & - \beta_1 \sin(x)\sin(3y)\cos(3z), \end{aligned} \tag{24c}$$

where

$$\begin{aligned} \delta_1 = {} & \Bigg[1 - \frac{3t}{R} + \left(\frac{9}{R^2} - \frac{1}{16}\right)\frac{t^2}{2} - \left(\frac{27}{2R^2} - \frac{23}{32}\right)\frac{t^3}{3R} \\ & + \left(\frac{31}{132\cdot 64} - \frac{185}{48R^2} + \frac{27}{2R^4}\right)\frac{t^4}{4} \\ & - \left(\frac{555}{44\cdot 32\cdot 8} - \frac{2575}{192R^2} + \frac{81}{8R^4}\right)\frac{t^5}{5R}\Bigg], \end{aligned} \tag{25a}$$

$$\alpha_1 = -\frac{1}{8}\left[t - \frac{7t^2}{R} + \left(\frac{74}{3R^2} - \frac{43}{44\cdot 12}\right)t^3 - \left(\frac{175}{3R^3} - \frac{43}{44}\right)\frac{t^4}{4}\right], \tag{25b}$$

$$\beta_1(\mathbf{x}, t=0) = -\frac{1}{16}t^2\left(\frac{1}{2} - \frac{13}{2}\frac{t}{R}\right), \tag{25c}$$

$$\gamma_2(\mathbf{x}, t=0) = -\frac{15}{176}t^2\left(\frac{1}{2} - \frac{31}{6}\frac{t}{R}\right), \tag{25d}$$

$$\gamma_3(\mathbf{x}, t=0) = -\frac{1}{16}t^2\left(\frac{1}{2} - \frac{31}{6}\frac{t}{R}\right). \tag{25e}$$

Clearly, as smaller scale fluid motion is generated over time, and the process of energy transfer from large to small scales was the original motivation for TG37. Based on this perturbation solution, TG37 provided prediction of the flow kinetic energy and dissipation rate at short times. Note that this energy transfer across scales is not present in the 2D Taylor-Green vortex flow often used to benchmark numerical methods in two spatial dimensions [41].

The parameter settings for our simulation are shown in Table 1. All the simulations are set with the physical Reynolds number $Re = U_0 L_B/\nu = 600\pi$. The velocity scale in LBM is chosen such that, at the same grid resolution N, one LBM time step is equivalent to one spectral time step. The LBM MRT parameters are [39]: $s_9 = dt/(3\nu + 0.5dt)$, $s_1 = 1.19$, $s_2 = 1.4$, $s_4 = 1.2$, $s_{10} = 1.4$, $s_{13} = s_9$, $s_{16} = 1.98$, $\omega_\varepsilon = 0.0$, $\omega_{\varepsilon j} = -475/63$, and $\omega_{xx} = 0.0$.

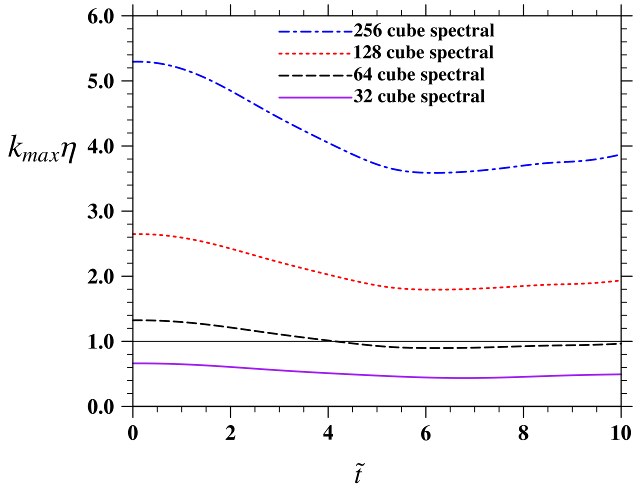


Fig. 3. The resolution parameter $k_{\max}\eta$ as a function of time, for different grid resolutions in the spectral simulation. The spectral truncation radius $k_{\max}$ is set to $k_{\max} = N/3$ by the 2/3 rule.

For the spectral simulation, as shown in Fig. 3, the resolution parameter $k_{\max}\eta$ changes from 0.662, 0.965, 1.935, 3.871 at $\tilde{t} = 0$ to 0.494, 1.32, 2.65, 5.29 at $\tilde{t} = 10$, for the different grid resolutions $N = 32, 64, 128, 256$, respectively, where $k_{\max} = N/3$ is the spectral truncation radius in the spectral code and η is the Kolmogorov length. The minimum $k_{\max}\eta$ occurs around $\tilde{t} = 6.2$ and is 0.437, 0.897, 1.79, 3.59, for $N = 32, 64, 128, 256$, respectively (Fig. 3). Here the dimensionless time is defined as $\tilde{t} = 2\pi U_0 t/L_B$. This indicates the flow is not resolved at $N = 32$, barely resolved at $N = 64$, and well resolved at $N = 128, 256$ in the spectral simulation [42]. As confirmed below, the results from $N = 64$ and $N = 128$ spectral simulations are almost identical. Note that the requirement of $k_{\max}\eta > 1$ for a well-resolved spectral simulation [42] was developed based on fully-developed forced homogeneous isotropic turbulence, and should only be taken as a reference. For example, the initial flow here is not a turbulent flow and contains only the large-scale, so the $k_{\max}\eta$ parameter is not meaningful.

The time evolutions of the kinetic energy and dissipation rate are shown in Fig. 4. Four groups of results are shown: the Taylor–Green short-time theoretical results [40], and numerical results from three different codes: spectral, LBM, and DUGKS. The spectral results at $N = 128$ can be viewed as the benchmark for others. Note that the spectral results at $N = 256$ (not shown) are essentially identical to the spectral results at $N = 128$. A number of observations can be made. First, all results are identical for $\tilde{t} < 2.0$, showing that the theoretical short-time expansion is accurate for $\tilde{t} < 2.0$ as long as the kinetic energy and dissipation rate are concerned. Also all codes at the two resolutions ($N = 64, 128$) shown adequately resolve the flow for $\tilde{t} < 2.0$, since at the early times the flow is dominated by large-scale initial flow, and the energy-cascading process has not yet generated the motion at the smallest scales. Second, the flow appears to be adequately resolved by the spectral code even at $N = 64$ as the results from the spectral code at $N = 64$ are essentially identical to those at $N = 128$, for both the kinetic energy and the dissipation rate, at all times shown in the plots. This confirms that the results from the spectral code at $N = 128$ and above can be used as the benchmark. The kinetic energy decays monotonically. However, the dissipation rate first increases due to production of small-scale flow structures, reaches a maximum at $\tilde{t} \approx 6$, and then decreases again due to reduction of the overall flow Reynolds number. Third, LBM and DUGKS results

Table 1
Parameter settings for the 3D Taylor–Green vortex flow.

Label	N	U_0	L_B	ν	$R \equiv \frac{U_0 L_B}{2\pi\nu}$	dt	$U_0 dt/dx$	$K(0)$	$D(0)$
Spectral	32/64/128/256	1.0	2π	$\frac{1}{300}$	300	$\frac{0.64}{N}$	0.1018591	0.125	0.0025
LBM	32/64/128	0.1018591	N	$\frac{U_0 N}{600\pi}$	300	1.0	0.1018591	$0.125U_0^2$	$0.0025 \cdot \frac{2\pi U_0^3}{L_B}$
DUGKS	32/64/128	0.1018591	N	$\frac{U_0 N}{600\pi}$	300	0.5	$\frac{0.1018591}{2}$	$0.125U_0^2$	$0.0025 \cdot \frac{2\pi U_0^3}{L_B}$

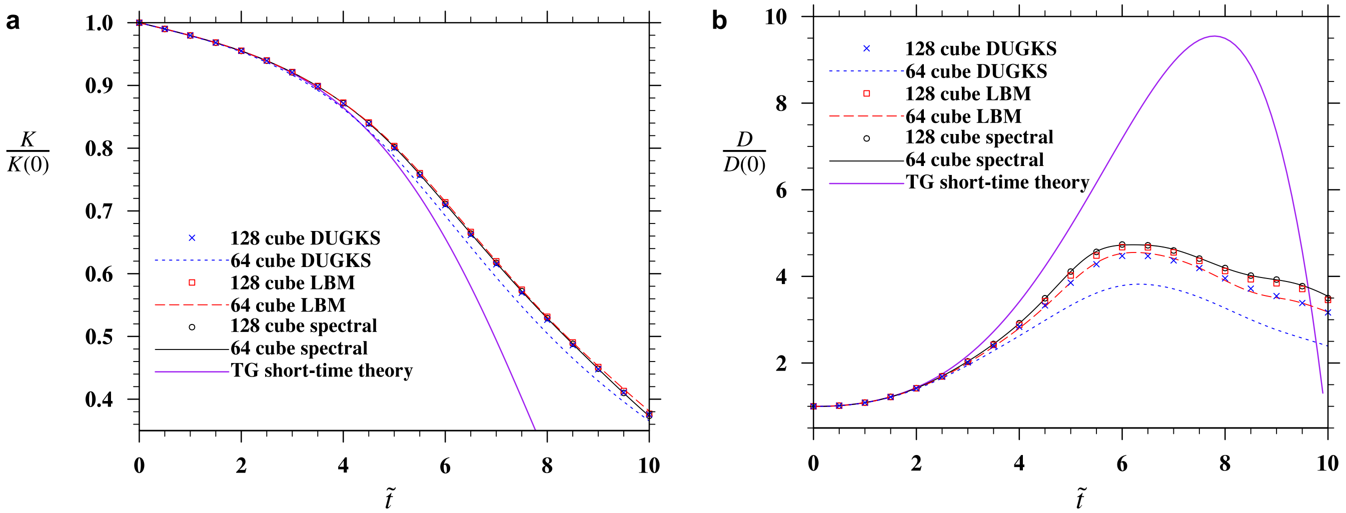


Fig. 4. (a) Kinetic energy and (b) dissipation rate as a function of time, for the 3D Taylor–Green vortex flow at $Re = 300$. All quantities are normalized as indicated.

Table 2
Relative errors in percentages in the predicted kinetic energy and dissipation rate using the 256^3 as the benchmark.

	$K(t = 1.0)$	$K(t = 5.0)$	$K(t = 10.0)$	$D(t = 1.0)$	$D(t = 5.0)$	$D(t = 10.0)$
Spectral (32^3)	0.0023	0.69	−1.13	0.0042	−7.64	−7.68
Spectral (64^3)	0.0006	0.03	−0.123	0.0009	−0.0143	1.
Spectral (128^3)	0.0001	0.0001	0.00742	0.0002	0.0055	0.0190
LBM (32^3)	0.0143	0.435	2.79	−0.345	−.3	−33.2
LBM (64^3)	0.0163	0.299	2.08	−0.0921	−6.18	−9.14
LBM (128^3)	0.0261	0.215	0.894	−0.0183	−2.16	−1.16
DUGKS (32^3)	−0.394	−8.42	−10.1	−1.14	−44.6	−59.7
DUGKS (64^3)	−0.024	−1.83	−2.16	−0.229	−19.0	−31.6
DUGKS (128^3)	0.024	−0.186	0.632	−0.0528	−6.28	−9.46

show some departures from the spectral results, but they converge to the spectral results at $N = 128$. Forth, compared to LBM results, DUGKS clearly exhibits a larger numerical dissipation, which is consistent to the observation in [23] based on the simulations of decaying homogeneous isotropic turbulence. Finally, it is interesting to note that the errors in the kinetic energy for the LBM and DUGKS are much smaller than the errors in the predicted dissipation rate. The reason is that the numerical dissipation in LBM and DUGKS causes a slower growth of small scales and as such a smaller physical dissipation, as shown in Fig. 4(b). This error combined with the numerical dissipation itself tends to provide a reasonable net total dissipation, and consequently, a reasonable prediction of the kinetic energy.

Using the 256^3 spectral data as the benchmark, we show in Table 2 the relative errors in percentage of the predicted kinetic energy and dissipation rate at three different times. First, the relative errors in the spectral method at lower resolutions are small if the resolution is sufficient. Second, once again, the results of LBM and DUGKS converge to the spectral data as the mesh resolution is increased. Third, as a given mesh resolution, the relative errors in DUGKS are larger than those in LBM. Fourth, the relative errors increase with time for a given resolution. Finally, there is a systematic error in DUGKS and LBM due to flow initialization, although the iterative approach in [44] was used to initialize the mesoscopic distribution functions.

As a measure of the accuracy of the simulated small-scale flow structures, in Fig. 5 we show the time evolutions of the velocity-derivative skewness and flatness. Here the skewness and flatness are defined as

$$S = \frac{\left\langle \frac{1}{3}\left[\left(\frac{\partial u}{\partial x}\right)^3 + \left(\frac{\partial v}{\partial y}\right)^3 + \left(\frac{\partial w}{\partial z}\right)^3 \right] \right\rangle}{\left\{ \left\langle \frac{1}{3}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2 \right] \right\rangle \right\}^{3/2}},$$

$$F = \frac{\left\langle \frac{1}{3}\left[\left(\frac{\partial u}{\partial x}\right)^4 + \left(\frac{\partial v}{\partial y}\right)^4 + \left(\frac{\partial w}{\partial z}\right)^4 \right] \right\rangle}{\left\{ \left\langle \frac{1}{3}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2 \right] \right\rangle \right\}^{2}}. \qquad (26)$$

We developed the short-time theory ourselves based on the analytical solution for the velocity field in Eq. (24). First, we observe that the short-time theory provides a good prediction of skewness only for $\tilde{t} < 1.5$, and a good prediction of flatness only for $\tilde{t} < 1.0$. There are two reasons for the degraded prediction performance: (a) the flatness and skewness depend more sensitively on the resolution of the small scales so the numerical dissipation will have a severe impact; (b) the energy cascading system is highly nonlinear,

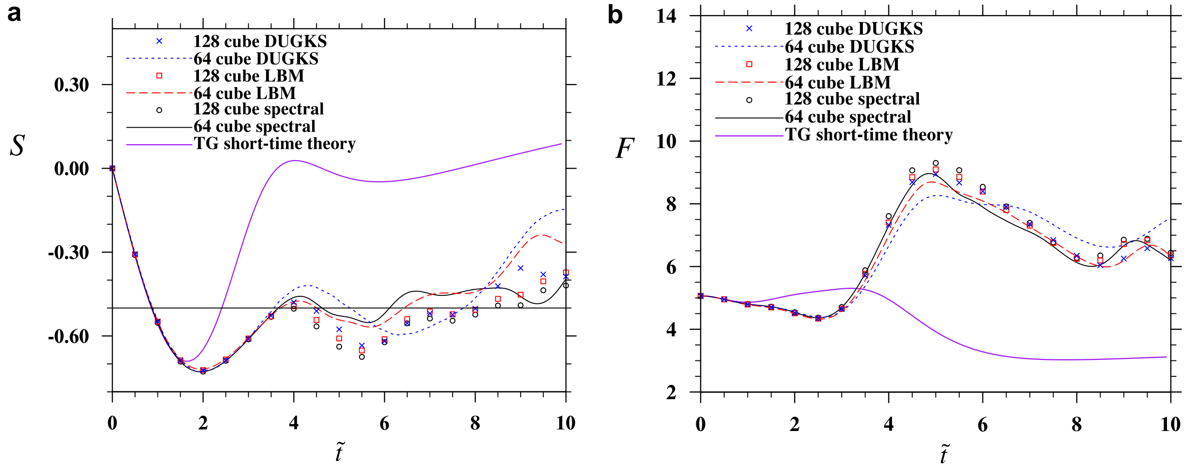


Fig. 5. (a) Velocity-derivative skewness and (b) velocity-derivative flatness as a function of time, for the 3D Taylor–Green vortex flow at $Re = 300$. The horizontal line on (a) indicates a level of -0.5, a value typically observed for forced homogeneous isotropic turbulence.

Table 3
The DUGKS order of accuracy based on the L1 norm, measured from the 3D Taylor–Green vortex flow simulations using DUGKS 256^3 simulation as the benchmark. At the same grid resolution and same time, the corresponding error norm for v velocity is identical to that of the error norm for u velocity.

	32^3		64^3		128^3
$2\pi U_0 t/L_B$	$u(L1)/w(L1)$	Order of accuracy	$u(L1)/w(L1)$	Order of accuracy	$u(L1)/w(L1)$
1.0	$6.98 \times 10^{-3}/3.93 \times 10^{-2}$	2.09 / 2.16	$1.64 \times 10^{-3}/8.80 \times 10^{-3}$	1.34 / 2.23	$6.47 \times 10^{-4}/1.87 \times 10^{-3}$
5.0	0.154/0.211	1.75 / 1.47	$4.57 \times 10^{-2}/7.62 \times 10^{-2}$	2.23 / 2.24	$9.77 \times 10^{-3}/1.63 \times 10^{-2}$

Table 4
The DUGKS order of accuracy based on the L2 norm, measured from the 3D Taylor-Green vortex flow simulations using DUGKS 256^3 simulation as the benchmark. At the same grid resolution and same time, the corresponding error norm for v velocity is identical to that of the error norm for u velocity.

	32^3		64^3		128^3
$2\pi U_0 t/L_B$	$u(L2)/w(L2)$	Order of accuracy	$u(L2)/w(L2)$	Order of accuracy	$u(L2)/w(L2)$
1.0	$6.44 \times 10^{-3}/3.90 \times 10^{-2}$	2.07 / 2.16	$1.53 \times 10^{-3}/8.71 \times 10^{-3}$	1.35 / 2.24	$6.01 \times 10^{-4}/1.85 \times 10^{-3}$
5.0	0.182/0.286	1.49 / 1.18	$6.49 \times 10^{-2}/0.126$	2.02 / 2.12	$1.60 \times 10^{-2}/2.90 \times 10^{-2}$

and over time, even a small round-off error may lead to production of different local structures. In fact, the results of two spectral simulations agree only for $\tilde{t} < 3.5$, and are different at later times, possibly due to emergence of different local small-scale flow structures with different grid resolutions. We note that the flatness is slightly larger for the $N = 128$ spectral run when compared to the $N = 64$ spectral run. The amplitude of the skewness reaches a maximum at $\tilde{t} = 2$, corresponding to the strongest nonlinear energy transfer. When the small-scale motions are developed, say, for $6 < \tilde{t} < 8$, the skewness reaches a value of about -0.5, typical for a fully-developed homogeneous isotropic turbulence [43]. The flatness increases rapidly during the period (*i.e.*, $3 < \tilde{t} < 5$) of rapid development of small-scale flow, then decreases afterwards due to the reduction of flow Reynolds number. Both LBM and DUGKS yield reasonable results, with LBM data being closer to the spectral data, at early times and overall. The trends are all in good quantitative agreement with the spectral results, and the three numerical methods tend to converge as N is increased.

Next, we shall investigate the order of accuracy of the DUGKS using this 3D Taylor-Green vortex flow. For this purpose, the L1 and L2 error norms of the instantaneous flow field must be computed. Initially, we thought to use the 256^3 spectral flow field, but quickly we realized that the systematic errors due to the flow initialization in DUGKS tend to contaminate the error norms. Instead, we use the 256^3 DUGKS flow field as the benchmark for the simulated flows at lower resolutions. The L1 and L2 error norms at two different times are shown in Table 3 and 4, respectively. Then the order of accuracy can be estimated based on the ratio of error norms between two consecutive mesh resolutions, the results are shown in columns 3 and 5 in Table 3 and 4. Since this is a time-dependent nonlinear flow, the error norms increase with time. The averaged order of accuracy from the L1 norm is 1.94 (Table 3), and this is 1.83 (Table 4) if the L2 norm is used. Considering the strong nonlinear nature of the 3D Taylor-Green flow, This clearly demonstrates that the DUGKS scheme has a second-order accuracy.

Finally, we examine the scalability of our MPI code by plotting the measured wall clock time against the number of processors used in Fig. 6. The code was run on the IBM cluster Yellowstone at the NCAR-Wyoming Supercomputing Center. On this log-log plot, an ideal scalability would follow a -1 slope. Indeed, this is approximately the case, showing that our MPI code scales properly with the number of processors used. The wall clock times for the simulation of 3D Taylor-Green vortex flow on the 128^3 grid, using the three approaches, are compared in Table 5. In all cases, two-dimensional domain decompositions are used. Overall, the CPU time per time step for DUGKS and LBM are comparable, with DUGKS taking 19% longer time. Since DUGKS takes twice time steps, the overall wall clock time for DUGKS is equal to 2.43 times the wall clock of the LBM code. We should note that the spectral code has been highly optimized [45], LBM is relatively well optimized, and DUGKS has not been optimized.

Table 5
Comparison of wall clock times with 16 processors, for simulations of 3D Taylor–Green vortex flow on the 128^3 grid for a time period of $2\pi U_0 t/L_B = 10.0$.

	$2\pi U_0 dt/L_B$	total time steps	Total CPU time (s)	Wall clock time per step (s)	Ratio / time step	Overall Ratio
Spectral	0.005	1000	2969	0.185	1	1
LBM	0.005	1000	12,519	0.782	4.2	4.2
DUGKS	0.0025	2000	30,052	0.939	5.1	10.2

Table 6
Parameter settings for the laminar channel flow simulations.

Label	$N_1 \times N_2 \times N_3$	$u_{\max}$	r_G	$dx_{\min}/dx_{\max}$	$2H$	ν	$Re \equiv \frac{2u_{\max}H}{\nu}$	dt	$\sqrt{6}\cdot c_s dt/dx_{\min}$
Case 1	40 × 4 × 4	0.1	0.0	1.0	30.4891	0.2186	18.3	0.35355	0.5
Case 2	40 × 4 × 4	0.1	3.0	0.1938	24.1825	0.1322	18.3	0.077182	0.5632

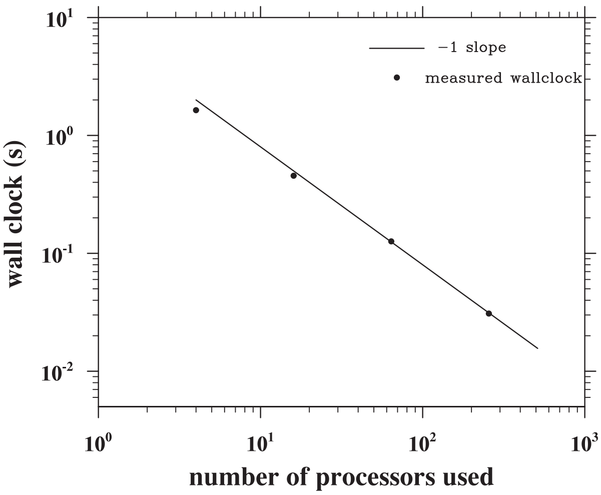


Fig. 6. Measured wall clock time as a function of the number of processors used. The line shows a power law of −1 slope.

3.2. The laminar channel flow

Before reporting the results for a turbulent channel flow in Section 3.3, we demonstrate briefly the simulation of transient laminar channel flow, as in this case the flow can be solved analytically.

A non-uniform mesh in the wall-normal direction is defined the by locations of the cell faces as

$$x_f(i) = \frac{\tanh(0.5r_G) + \tanh\left[r_G \frac{(i-N_1/2-1)}{N_1}\right]}{\tanh\left(\frac{r_G}{N_1}\right)}, \quad \text{for } i = 1, 2, \ldots, N_1+1, \tag{27}$$

where r_G is a parameter used to adjust the degree of non-uniformity. This defines a symmetric mesh with respect to $i = N_1/2+1$. The mesh is defined such that the maximum mesh size is one unit, namely, $dx_{\max} = x_f(N_1/2+1) - x_f(N_1/2) = 1$. The minimum mesh size is $dx_{\min} = x_f(2) - x_f(1) = [\tanh(0.5r_G) + \tanh(r_G/N_1 - 0.5r_G)]/\tanh(r_G/N_1)$. The above mesh specifies the channel half-width as

$$H = \frac{\tanh(0.5r_G)}{\tanh\left(\frac{r_G}{N_1}\right)}. \tag{28}$$

The limiting case of $r_G \to 0$ represents the uniform mesh. Two meshes with $r_G = 0$ and $r_G = 3$ are considered (Table 6). N_1 is set to 40. Since there is no dependence on y and z, N_2 and N_3 can be set to any value.

The analytical solution of the laminar transient flow is

$$\frac{v(x,t)}{v_{max}} = F(x_n, t_n) = \left(1 - x_n^2\right) - \sum_{k=0}^{\infty} \frac{(-1)^k \cdot 4}{[(k+0.5)\pi]^3} \times \cos[(k+0.5)\pi x_n]\exp\left[-(k+0.5)^2\pi^2 t_n\right], \tag{29}$$

where $x_n = x/H$ and $t_n \equiv \nu t/H^2$.

Fig. 7 shows the simulated velocity profiles at five different times with both the uniform grid and non-uniform grid. The theoretical solutions are also shown. The theoretical solutions are obtained using 300 terms in the summation in Eq. (29). Clearly, in all cases, the DUGKS results are in excellent agreement with the theoretical solutions. The difference between the simulated velocity and the theoretical value is plotted in Fig. 8. In both cases, there appears to be a positive fixed error at the wall for the specific no-slip boundary treatment we used. The errors inside are mostly positive, except at early times that they may be negative near the channel center. The shape and magnitude of the error profiles are affected by the mesh structure, but overall the errors for the non-uniform mesh case (Case 2) are comparable to those for the uniform mesh case (Case 1).

3.3. The turbulent channel flow

The parameter settings for the turbulent channel flow simulation are shown in Table 7. Specifically, the same nonuniform mesh defined by Eq. (27) is used in the wall normal (x) direction, with $N_1 = 128$ and $r_G = 4.0$. This setting yields $dx_{\min} = 0.0728$ and a channel half width of $H = 30.8589$. A 128^3 mesh resolution is used. The DUGKS domain size matches the spectral domain size, therefore, $dy = 3.03$, and $dz = 1.01$. The CFL number is $dt\sqrt{2}/dx_{\min} = 0.90$. In terms of the wall unit, $dx^+_{\min} = 0.425$, which is sufficient to resolve the wall layer [46]. With the parameters shown in Table 7, roughly 6 DUGKS time steps correspond to one LBM time step.

In Fig. 9(a), we show the mean flow speed (averaged over the whole flow domain) as a function of non-dimensional time $t^* \equiv tu^*/H$. A nonuniform force field described in [39] was added to the constant physical driving force, to excite the development of turbulent flow during $0 < t^* < 0.526$. The same perturbation forcing with identical parameters was applied in the LBM simulation. The difference in the mean flow speed at $t^* = 0$ between DUGKS and LBM was due to the use of slightly different initial mean flow profiles in the two simulations. Initially, the mean flow speed decreases as the kinetic energy is transferred from the mean flow to the turbulent fluctuations. After $t^* = 0.526$, the nonuniform force field is removed, and the mean flow speed continues to evolve, first reaches a minimum mean flow speed at around $t^* = 4.5$, then rebounds and gradually reaches a stationary value. For both the DUGKS and LBM runs, the flow appears to be stationary at around

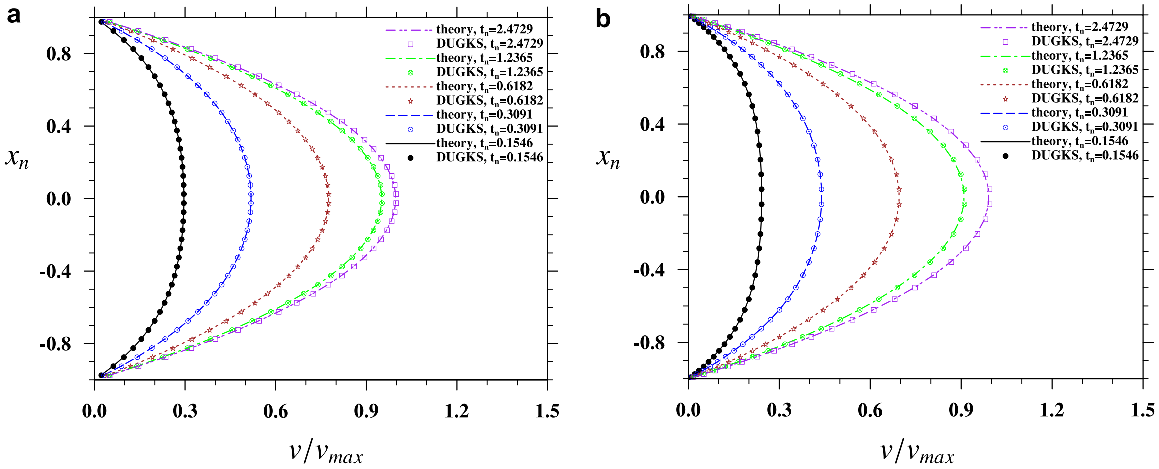


Fig. 7. Velocity profiles for the transient laminar flow: (a) uniform grid with $2H = 40.0$, and (b) nonuniform grid with $2H = 24.1825$. The symbols are the simulated DUGKS data and the lines are the theoretical solutions. $Re = 18.3$.

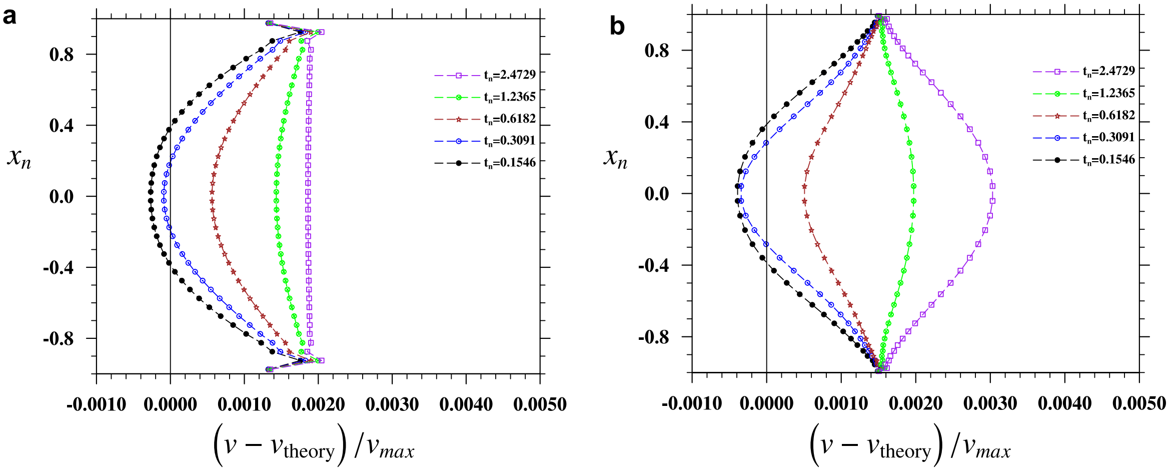


Fig. 8. The relative errors for the DUGKS transient laminar flow: (a) uniform grid with $2H = 40.0$, and (b) nonuniform grid with $2H = 24.1825$.

Table 7
Parameter settings for the turbulent channel flow simulations.

Run	$N_1 \times N_2 \times N_3$	Domain size	H	ν	u^*	Re_τ	$\delta x_{\min}/(\nu/u^*)$ (< 2.25 [46])	$u^{*2}dt/\nu$
Spectral [36]	129 × 192 × 160	$2H \times 4\pi H \times 4\pi H/3$	-	-	-	180	0.054	-
LBM [39]	200 × 400 × 200	$2H \times 4H \times 2H$	100	0.0036	0.00648	180	1.806	0.01166
DUGKS	128 × 128 × 128	$2H \times 4\pi H \times 4\pi H/3$	30.86	0.0012	0.007	180	0.425	0.001893

$u^*t/H = 25$. The average mean speed averaged for $u^*t/H > 25$ is 15.71 from the DUGKS simulation and 15.80 for the LBM simulation, and relative difference is about 0.6%. Due to the use of a much large computational domain size in DUGKS, the variations of the mean flow speed with time in DUGKS are much smaller than those in LBM. As a comparison, the mean flow speed in the spectral simulation reported in [36] at the same flow Reynolds number was 15.63. Therefore, the DUGKS mean flow speed is 0.5% larger than the spectral benchmark, while the LBM mean flow speed is 1.1% larger than the spectral benchmark, implying that the DUGKS yields a better result in terms of the mean flow speed. The relative error in DUGKS is due only to the numerical truncation error, while the relative error in LBM is due to both the domain size effect and numerical truncation error.

Next, we show profiles averaged over time after $t^* = 25$. Fig. 9(b) displays the mean velocity profiles on a log-linear plot. Here, the legend "Spectral" refers to the data from the Chebychev-spectral simulation done by the Stanford group [36,37] using a domain size $4\pi H \times 2H \times 4\pi H/3$. In wall units, the channel center is at $x^+ = 180$. The DUGKS result is very close to the LBM result, and both are in reasonable agreement with the spectral results. The profile fits well the standard linear viscous sublayer scaling for $y^+ < 5$, and the inertial sublayer scaling starting at $y^+ > 30$. In the inertial sublayer, the mean velocity from the DUGKS and LBM is slightly larger than the spectral benchmark data.

The averaged Reynolds stress profiles are shown in Fig. 10, in both linear-linear and log-linear plots. The center of the channel is at $x/(2H) = 0.5$. The sum of Reynolds stress and viscous stress varies linearly from the channel center to the channel wall. All data are in good agreement. When compared to the spectral result, the DUGKS result is clearly better than the LBM result in the near wall and buffer regions ($x/(2H) < 0.15$).

Finally, the rms velocity profiles are shown in Fig. 11. Overall, the profiles from the DUGKS and LBM are in reasonable agreement. In the near-wall regions, the streamwise rms velocity is the largest and the transverse rms velocity is the smallest. The DUGKS results

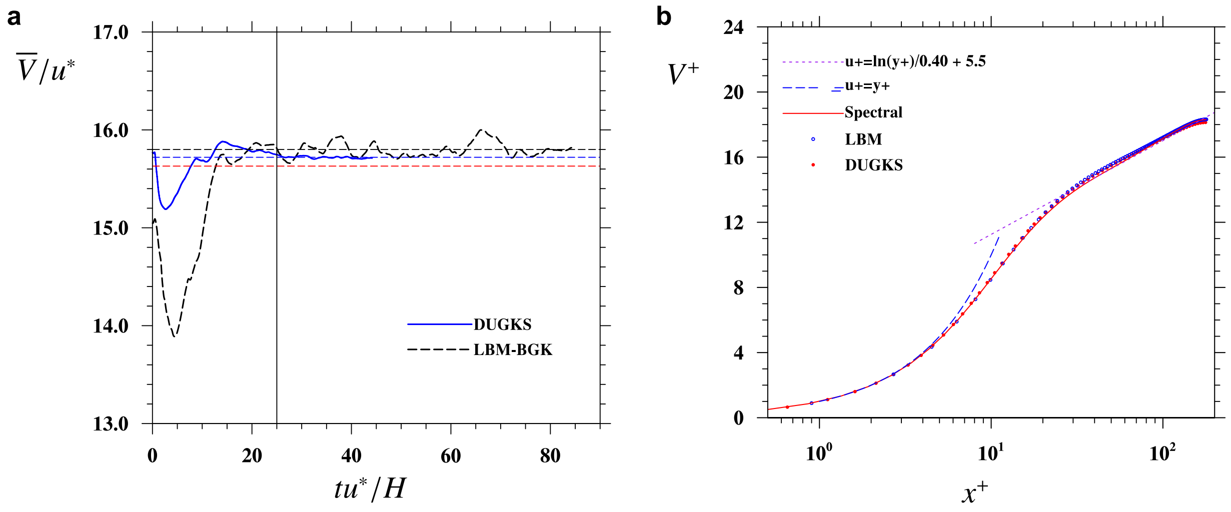


Fig. 9. (a) Time evolution of the mean flow velocity for the simulated single-phase turbulent channel flow. (b) The mean flow velocity profiles, averaged over $t^* > 25$.. The thin dash blue line in (a) marks the long-time average of $\langle U\rangle/u^* = 15.71$ for the GUGKS simulation, the thin dash black line in (a) marks the long-time average of $\langle U\rangle/u^* = 15.80$ for the LBM simulation, and the thin red dash line in (a) marks the average mean speed from the spectral simulation reported in [36]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

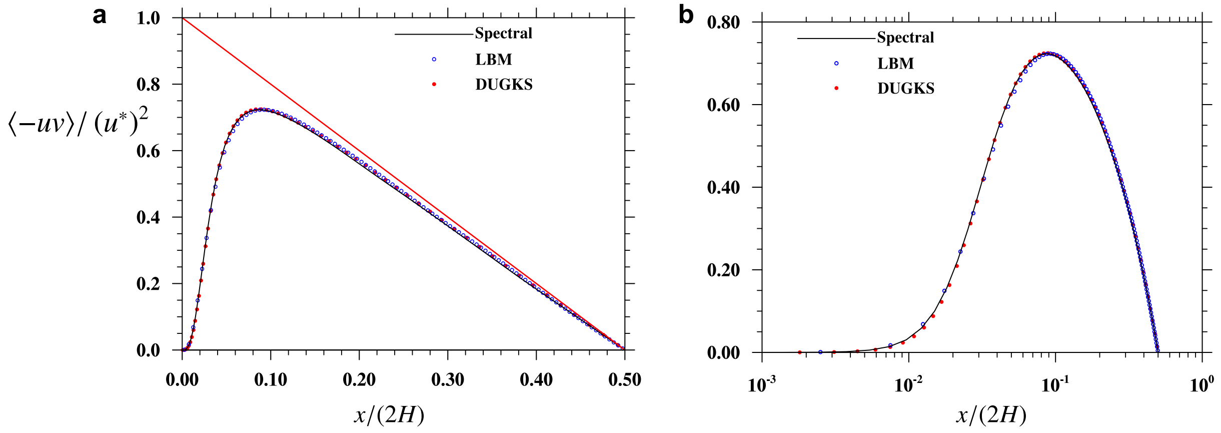


Fig. 10. Turbulent Reynolds stress profiles in half the channel: (a) linear-linear plot, (b) log-linear plot. The data are averaged over $t^* > 25$. All quantities are normalized by u^{*2}. The thin red line in (a) denotes the total shear stress, so the difference between this straight line and the data represents the viscous shear stress due to the mean flow. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

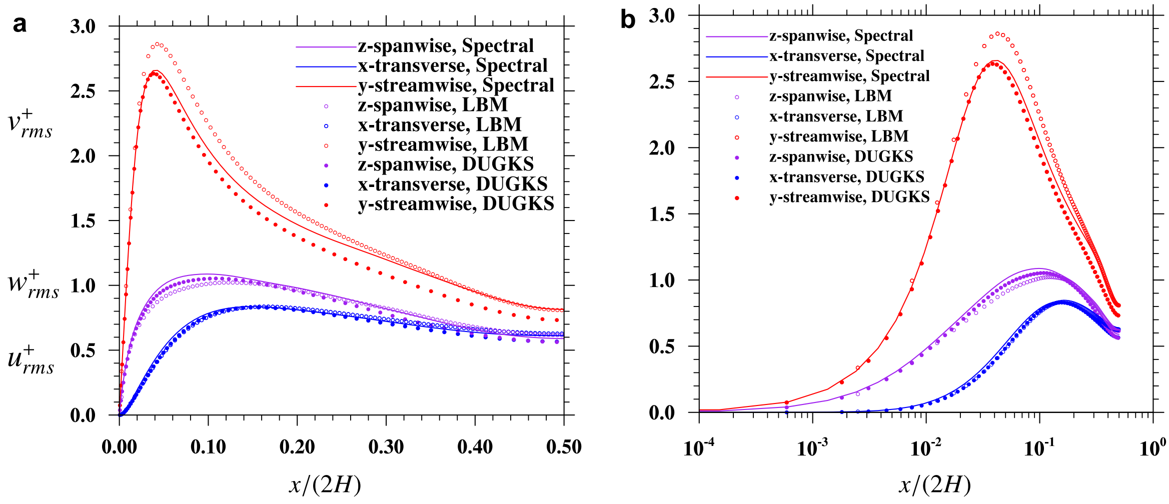


Fig. 11. The r.m.s. fluctuation velocity profiles: (a) linear-linear plot; (b) log-linear plot. The data are averaged over $t^* > 25$. All quantities are normalized by u^{*2}.

for the streamwise and spanwise rms velocities in the near-wall region ($x/(2H) < 0.15$) are better than the LBM results when compared to the spectral benchmark. In the center region of the channel, the DUGKS result in the streamwise direction appears to deviate more from the spectral data for two reasons. First, the DUGKS resolution is perhaps too coarse in the channel center region in both transverse and streamwise directions and the numerical dissipation is significant. Second, the LBM result at the peak location overshoots due to the domain size effect [39]; this combined with the numerical dissipation in LBM makes the results appear better in the center region. Overall, we may still claim that the DUGKS results are better given the coarser resolution used in DUGKS (see Table 7) to cover a larger domain. Different distribution of grid points in the transverse direction and different grid resolutions should be tested, which could be a topic for future work.

We note that the numerical dissipation in DUGKS depends on the time step size (or the CFL number), as shown in Zhu et al. [34]. A smaller time step implies a smaller numerical dissipation and more accurate solution. Since in our turbulent channel flow simulation, the time step size is derived from the CFL number based on the smallest lattice spacing near the channel wall, the time step is indeed very small, which could be the reason for accurate turbulence statistics from DUGKS.

In summary, the first simulation of turbulent channel flow using DUGKS shows that the results are reasonably accurate. Of significance is that these preliminary DUGKS results are obtained with a coarse grid resolution covering a large domain size, thus the grid cell aspect ratios near the wall are large. The maximum aspect ratio measured in terms of streamwise lattice spacing to transverse lattice spacing near the wall is 41.62. Since the standard LBM uses a cubic lattice, it is costly to cover the same domain size, so a smaller domain size was used here. Recently, we have shown that it is possible to extend the D3Q19 LBM to a cuboid grid [47], but the cuboid model could have numerical stability issues at large aspect ratios. The DUGKS code is numerically stable for grid cells with large aspect ratios.

4. Summary and outlook

In this paper, we report on parallel implementation of the recently developed mesoscopic DUGKS scheme [19,20] for simulating three-dimensional flows. When dealing with incompressible flows, the same D3Q19 lattice velocity model used in LBM can be adopted. However, the transport term is treated as the sum of fluxes across the interfaces of a grid cell volume. This removes the restriction on the grid structure of the standard LBM, making it feasible to incorporate non-uniform and irregular grids. The treatment of the transport term is more involved compared to the streaming operation in LBM, but the transformation from the linkwise streaming in LBM to surface fluxes in DUGKS has potential benefits of removing corner singularity and generating a much more stable scheme. In this paper, we consider only non-uniform structured grids, and the parallel implementation is relatively straightforward. Besides the details of DUGKS algorithm implementation, we have also documented the implementations of a non-uniform grid, no-slip wall and non-uniform time-dependent forcing. The MPI code based on two-dimensional domain decomposition shows a good scalability.

The resulting parallel code was first tested for the 3D scale-evolving Taylor–Green vortex flow, and DUGKS results are compared to the Taylor-Green short-time analytical solution [40] and numerical results from pseudo-spectral method and LBM. The accuracy of the Taylor–Green short-time analytical solution can be accessed by the high-resolution spectral solution. It was found that DUGKS can provide accurate results with adequate grid resolution. The numerical errors in DUGKS are larger than those in LBM, showing that DUGKS is more dissipative when compared to LBM. This is consistent with the conclusion in [24]. Using the DUGKS 256^3 flow as the benchmark, we have demonstrated the second-order accuracy of the DUGKS scheme.

DUGKS results for a transient laminar channel flow agree well with the theoretical solution, with both uniform and non-uniform grids in the wall-normal direction. Here we only considered the extrapolated bounced back for the no-slip wall. The numerical error on the wall is related to how the distribution on the boundary is constructed, which needs further investigation.

A first DUGKS simulation of the turbulent channel flow was successfully performed, and results were compared to LBM and spectral results. We find that even with the use of a cuboid grid of very large aspect ratio (in our case the largest aspect ratio is 41.62 near the wall), the DUGKS code is numerically stable. The DUGKS results are better than the LBM results, in terms of the simulated mean flow speed and turbulence statistics in the near wall and buffer regions. This is very encouraging given that a relatively coarse grid was used for a large computational domain. LBM with a uniform cubic grid could not handle the same domain size with limited computational resources [39]. Furthermore, a 3D cuboid LBM model has recently been developed, but even at grid aspect ratio of 2.0, the cuboid LBM code was found to be numerically unstable [47]. Although the time step size in DUGKS has to be small due to the use of non-uniform grid, but the reduced grid size makes the overall computational cost of DUGKS to be comparable to LBM. This combined with our recent work on using DUGKS to simulate decaying homogeneous isotropic turbulence [24] clearly shows that DUGKS can be used as a DNS tool for turbulent flows, particularly compressible and thermal turbulent flows. Nevertheless the work here is considered preliminary, further examples of turbulent flow simulations using DUGKS at higher flow Reynolds numbers and different flow configurations are needed to further document the capabilities and accuracy of DUGKS relative to LBM and other methods. While in this paper, DUGKS was used as a DNS tool since the local grid spacings in all spatial directions were made sufficiently small, to adequately resolve all scales of the flow. Due to its nature as a finite-volume formulation, DUGKS has the potential to be used as an implicit large-eddy simulation (LES) tool for high-Reynolds number flows at a given grid resolution.

Acknowledgements

This work has been supported by the U.S. National Science Foundation (NSF) under grants CNS1513031, CBET-1235974, and AGS-1139743 and by Air Force Office of Scientific Research under grant FA9550-13-1-0213. LPW also acknowledges support from the Ministry of Education of P.R. China and Huazhong University of Science and Technology through Chang Jiang Scholar Visiting Professorship. LPW would like to acknowledge the travel support from U.S. National Science Foundation (NSF) to attend ICMMES-2015, held in CSRC (www.csrc.ac.cn), Beijing, July 20, – 24, 2015, under the Grant CBET-1549614. Computing resources are provided by National Center for Atmospheric Research through CISL-P35751014, and CISL-UDEL0001 and by University of Delaware through NSF CRI 0958512.

References

[1] Eggels JG. Direct and large-eddy simulation of turbulent fluid flow using the lattice Boltzmann scheme. Int J Heat Fluid Flow 1996;17(3):307–23.
[2] Chen S, Doolen G. Lattice Boltzmann method for fluid flows. Annu. Rev. Fluid Mech. 1998;30:329–64.
[3] Yu D, Mei R, Luo L-S, Shyy W. Viscous flow computations with the method of lattice Boltzmann equation. Prog Aerosp Sci. 2003;39:329–67.
[4] Chen H, Kandasamy S, Orszag S, Shock R, Succi S, Yakhot V. Extended Boltzmann kinetic equation for turbulent flows. Science 2003;301(5633):633–6.
[5] Yu H, Luo L-S, Girimaji SS. LES Of turbulent square jet flow using an MRT lattice Boltzmann model. Comput Fluids 2006;35(8):957–65.

[6] Guo Z, Shu C. Lattice boltzmann method and its applications in engineering (advances in computational fluid dynamics). World Scientific Publishing Company; 2013.

[7] Wang L-P, Peng C, Guo ZL, Yu ZS. Flow modulation by finite-size neutrally buoyant particles in a turbulent channel flow. ASME J Fluids Engr 2016a;138:041103. doi:10.1115/1.4031691.

[8] Marié S, Ricot D, Sagaut P. Comparison between lattice Boltzmann method and Navier–Stokes high order schemes for computational aeroacoustics. J Comp Phys 2009;228:1056–70.

[9] Peng Y, Liao W, Luo L-S, Wang L-P. Comparison of the lattice Boltzmann and pseudo-spectral methods for decaying turbulence: low-order statistics. Comput Fluids 2010;39(4):568–91.

[10] Gao H, Li H, Wang L-P. Lattice Boltzmann simulation of turbulent flow laden with finite-size particles. Comput Math Appl 2013;65(2):194–210.

[11] Wang L-P, Ayala O, Gao H. Study of forced turbulence and its modulation by finite-size solid particles using the lattice Boltzmann approach. Comp Math Appl 2014;67:363–80.

[12] Guo ZL, Zhao TS, Shi Y. Physical symmetry, spatial accuracy, and relaxation time of the lattice boltzmann equation for microgas flows. J Appl Phys 2006;99:074903.

[13] Zhang YH, Gu XJ, Barber RW, Emerson DR. Capturing Knudsen layer phenomena using a lattice Boltzmann model. Phys Rev E 2006;74:046704.

[14] Guo ZL, Zheng CG, Shi BC. Lattice Boltzmann equation with multiple effective relaxation times for gaseous microscale flow. Phys Rev E 2008;77:036707.

[15] Shan X, Yuan XF, Chen H. Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation. J Fluid Mech 2006;550:413–41.

[16] Xu K, Huang JC. A unified gas-kinetic scheme for continuum and rarefied flows. J Comput Phys 2010;229:7747–64.

[17] Xu K. Direct modeling for computational fluid dynamics: construction and application of unified gas-kinetic schemes. World Scientific, Singapore; 2015a.

[18] Xu K. Direct modeling for computational fluid dynamics. Acta Mech Sin 2015b;31:303–18.

[19] Guo Z, Xu K, Wang R. Discrete unified gas kinetic scheme for all knudsen number flows: low-speed isothermal case. Phys Rev E 2013;88(3):033305.

[20] Guo Z, Wang R, Xu K. Discrete unified gas kinetic scheme for all Knudsen number flows. II Thermal compressible case. Phys Rev E 2015;91(3):033313.

[21] Wang P, Wang L-P, Guo Z. A coupled discrete unified gas-kinetic scheme for Boussinesq flows. Comput Fluids 2015a;120:70–81.

[22] Zhu L, Guo Z, Xu K. Discrete unified gas kinetic scheme on unstructured meshes. Comput Fluids 2016;127:211–25.

[23] Wang P, Guo LZZ, Xu K. A comparative study of LBE and DUGKS methods for nearly incompressible flows. Commun Comput Phys 2015b;17(3):657–81.

[24] Wang P, Wang L-P, Guo Z. Comparison of the LBE and DUGKS methods for DNS of decaying homogeneous isotropic turbulence. Phys Rev E 2016b;94:043304.

[25] Ohwada T. On the construction of kinetic schemes. J Comput Phys 2002;177:156–75.

[26] Chen S, Xu K. A comparative study of an asymptotic preserving scheme and unified gas-kinetic scheme in continuum flow limit. J Comp Phys 2015;288:52–65.

[27] Amati G, Succi S, Piva R. Massively parallel lattice-Boltzmann simulation of turbulent channel flow. In J Modern Phys C 1997;8:869–77.

[28] Pohl T, Kowarschik M, Wilke J, Iglberger K, Rde U. Optimization and profiling of the cache performance of parallel lattice Boltzmann codes. Parallel Process Lett 2003;13:549–60.

[29] Wellein G, Zeiser T, Hager G, Donath S. On the single processor performance of simple lattice Boltzmann kernels. Comput Fluids 2006;35:910–19.

[30] Mattila K, Hyväluoma J, Rossi T, Aspnäs M, Westerholm J. An efficient swap algorithm for the lattice Boltzmann method. Comput Phys Commun 2007;176:200–10.

[31] Bailey P, Myre J, Walsh SDC, Lilja DJ, Saar MO. Accelerating lattice Boltzmann fluid flow simulations using graphics processors. In: Proceeding of the International Conference on Parallel Processing; 2009. p. 550–7.

[32] Wittmann M, Zeiser T, Hager G, Wellein G. Comparison of different propagation steps for lattice Boltzmann methods. Comput Math Appl 2013;65:924–35.

[33] He XY, Shan XW, Doolen GD. Discrete Boltzmann equation model for nonideal gases. Phys Rev E 1998;57:R13–16.

[34] Zhu L, Wang P, Guo ZL. Performance evaluation of the general characteristics based off-lattice Boltzmann scheme and DUGKS for low speed continuum flows. J Comp Phys 2017;333:227–46.

[35] Chen SY, Martinez D, Mei RW. On boundary conditions in lattice Boltzmann methods. Phys Fluids 1996;8:2527–36.

[36] Kim J, Moin P, Moser R. Turbulence statistics in fully-developed channel flow at low Reynolds-number. J FluidMech 1987;177:133–66.

[37] Moser R, Kim J, Mansour NN. Direct numerical simulation of turbulent channel flow up to Re-Tau = 590. Phys Fluids 1999;11:943–5.

[38] d'Humieres D, Ginzburg I, Karfczyk M. Multiple-relaxation-time lattice Boltzmann models in three dimensions. Phil Trans R Soc Lond A 2002;360:437–51.

[39] Wang L-P, Peng C, Guo ZL, YU ZS. Lattice Boltzmann simulation of particle-laden turbulent channel flow. Computers and Fluids 2016c;124:226–36.

[40] Taylor GI, Green AE. Mechanism of the production of small eddies from large ones. Proc Royal Sco London, A 1937;158:499–521.

[41] Zong Y, Peng C, Guo Z, Wang L-P. Designing correct fluid hydrodynamics on a rectangular grid using MRT lattice Boltzmann approach. Comp Math Appl 2015;72:288–310.

[42] Eswaran V, Pope SB. An examination of forcing in direct numerical simulations of turbulence. Comput Fluids 1988;16:257–78.

[43] Wang LP, Maxey MR. Settling velocity and concentration distribution of heavy particles in a forced isotropic and homogeneous turbulence. J Fluid Mech 1993;256:27–68.

[44] Mei R, Luo L-S, Lallemand P. Consistent initial conditions for lattice Boltzmann simulations. Comp Fluids 2006;35:855–62.

[45] Ayala O, Wang L-P. Parallel implementation and scalability analysis of 3d fast fourier transform using 2d domain decomposition. Parallel Comput 2013;39:58–77.

[46] Lammers P, Beronov KN, Volkert R, Brenner G, Durst F. Lattice BGK direct numerical simulation of fully developed turbulence in incompressible plane channel flow. Comput Fluids 2006;35:1137–53.

[47] Wang L.-P., Min H., Peng C., Genevaa N., Guo Z.L. A lattice-Boltzmann scheme of the Navier–Stokes equation on a three-dimenional cuboid lattice. Comput Fluids. 10.1016/j.camwa.2016.06.017.