Fractional step lattice Boltzmann methods with coarse corrective steps

Alpar Matyas

Continental AG, Division Chassis and Safety, Salzburgstr. 8, 550018 Sibiu, Romania.

Abstract

A fractional step lattice Boltzmann method (FSLBM) is developed for incompressible fluid flow in three dimensions. The presented scheme is based on the time-evolution of velocity and pressure in contrast to the evolution of probability density functions. Furthermore, the discretization with coarse corrective steps (CCS) facilitates calculations up to large Reynolds numbers. The method is validated for the friction factor of pipes with various surface roughness, as well as, for the Reynolds stress of smooth pipes. A good agreement is found to general scaling laws and high quality DNS data respectively. Development of turbulent flow in a pipe calculated with $\approx 0.57 \times 10^9$ lattice elements on a single GPU is reported. FSLBMs can process 56.6 million elements per GB of GPU memory which is considered a near-optimum memory efficiency for 3D lattice Boltzmann calculations.

Keywords: Lattice Boltzmann method, Projection scheme, Corrective step, Periodic pipe flow, Turbulence

1. Introduction

Lattice Boltzmann methods (LBMs) play an important role for the modeling of fluid flow based on kinetic theory. In contrast to computational fluid dynamics (CFD) based on the Navier-Stokes equation (NSE), LBMs are implemented with the collision and streaming of particles in discrete lattice space [1, 2]. The collision occurs locally and the streaming step updates the density functions (DFs) at each lattice point from the neighbors. Furthermore, the choice of the lattice defines the neighbor relations by the velocity streaming direction, therefore lattices with more streaming directions from a given cell offer a more accurate description of the flow problem and get closer to the solutions of the continuum Boltzmann equation [3]. Several lattice types have been reported so far based on nearest neighbors like D3Q15, D3Q19 and D3Q27, where the number after D is the spatial dimension of the lattice while numbers after Q stand for the amount of lattice neighbors (see Figure 1. (a) for D3Q27). Additionally, lattices with more complex neighbor structure D3Q39 [4], D3Q41,D3Q125 [5] have been reported which can be applied also to compressible flow. Indeed, the latter configurations consistently preserve the relevant hydrodynamic moments of the continuum Boltzmann equation, i.e., mass, momentum and energy are conserved for Mach numbers below approximately 0.7[6]. Furthermore, lattices based on body centered cubic ordering have been reported with improved numerical accuracy compared to face centered cubic lattices [7, 8]. An equally important part for LBM calculations is the equilibrium density function which is found by Hermite polynomial expansion in the velocity up to arbitrary orders greater or equal to two [9]. However, the expansion order must match the type of lattice that is used for correctly managing the interplay between discretization errors and defects from not having Galilean invariance [3]. Particularly for the D3Q27 lattice used for incompressible flow calculations, the equilibrium DF should be truncated at order two in velocity, since energy conservation is not Galilean invariant, i.e., cubic moment defects are introduced resulting from the discretization of the continuous Maxwell-Boltzmann DF [3]. A further approach for defining the incompressible limit for the steady fluid flow is to change the equilibrium DF. Indeed, He and Luo introduced equilibrium DFs that lead to the incompressible property proven with Chapman-Enskog expansion [10] in the time step. Furthermore, their method is exact for steady flow and as it will be shown in this paper can perform very well with fractional step methods based on small time steps and coarse grid corrections.

Email address: alparmat@gmail.com (Alpar Matyas)

LBMs can be defined with collision operators having single or multiple relaxation times, where the former can have simple interpretations and the latter can be more accurate [11]. The multiple relaxation time (MRT) approach generally suppresses the hydrodynamic moments that are not needed for the particular flow problem by orthogonalization, [12, 3, 13], i.e., equations for energy conservation, heat flux and higher order hydrodynamics are not needed in the limit of incompressible flow. MRTs can be chosen to improve the LBM stability and to cover an extended Reynolds number (Re) range as compared to the case of single relaxation time. However, the relaxation times can depend on the lattice type and the flow problem, consequently they are not general. A further approach that corrects some of the issues based on MRT is based on cumulant LBMs where Galilean invariance, mode orthogonality and mode statistical independence hold [13]. In spite of the above complex but accurate collision methods we choose the simplest possible collision operator for our scheme, i.e., the Bhatnagar-Gross-Krook (BGK) model [14]. Additionally, for every flow problem we use the relaxation time set to one as has been previously performed in fractional step LBMs [15, 16]. Such choice is necessary since FSLBMs are constrained for single relaxation times in their predictive step. Consequently, the choice of the BGK single relaxation time model provides a simple relation to the fluid viscosity and for the calculation of pressure and velocity from the DFs. Furthermore, an equivalence of the LBM to the NSE equation can be proven by Chapman-Enskog expansion [10, 11, 17]. Such properties have shown that LBMs can be used as a general technique for modeling various scales of fluid flow.

A scheme optimized for GPU calculations is presented in this paper since consumer-type GPUs are cost and energy efficient while still offer parallel computation speeds between consumer-type multi-core CPUs and server farms. Consequently, we attempt to combine the industrial requirement for a large number of elements and fast calculations with the scientific requirement of having very low numerical dissipation and accurate methods; all this on GPUs to have cheap and efficient parallelism. Furthermore the choice for LBMs for flow calculation is based on its low numerical dissipation and efficient prediction for turbulence noise [18] as well as its ability of modeling complex valve stability phenomena [19, 20]. For the implementation of a CFD code that can be applied to a wide range of incompressible flow problems on single (or few) GPUs the main limitation is memory. Therefore to improve storage efficiency of the method we set the goal to have as less data stored as possible for the calculation of hydrodynamic variables. A first step in reaching this optimum storage scheme for a given LBM or Navier-Stokes finite difference method is the shift algorithm [21]. The algorithm systematically moves through xy-planes (or a bundle of xy-planes) of the 3D Cartesian domain, updates the actual sample and stores it in the same memory array, however, several points further and opposed to the update direction. Consequently, it works with a single set of hydrodynamic variables, i.e., 27 DFs for D3Q27 LBMs or 3 velocity and 1 pressure variable for the 3D finite difference NSEs. Our method is based on the above shift algorithm approach for saving memory during the update of the flow variables [16]. In spite of the shift algorithm, however, the D3Q27 LBM is severely memory-limited as compared to the finite difference NSE since it needs 27/4 times more storage space in the GPU memory. Therefore, having a LBM implementation that stores the velocity vector and pressure instead of DFs is appealing in order to extend usability for large flow problems on single GPUs as well as to improve the memory bandwidth usage. FSLBMs provide a solution for these problems [15] by facilitating storage for the velocity and pressure and not the DFs. Furthermore FSLBMs are providing high memory efficiency and calculations up to very large Reynolds numbers (Re) as well as good parallel scalability. For FSLBM, the flow problem is separated into an LBM based predictive part and a viscosity-corrective part, which is a reverse diffusion equation for the velocity vector [15]. However, such FSLBMs have been implemented only up to two dimensions until recently [15, 16], therefore their usage has been limited to symmetric flows. Furthermore, numerical tests for an equivalent 3D implementation of FSLBM show limitation to very low Re, and therefore 3D FSLBM could not be practically used.

Several methods have been developed for improving data structuring and compute efficiency of LBM implementations [21, 22, 23] that depend mainly on the hardware of choice. For CPUs the access time is very similar for all the data in the memory, however the floating point operations used in the collision step take most of the computing time. This aspect favors the so-called AoS (array-of-structures) data layout, where each lattice point has arrays that contain all DFs and the evolution equations can be updated more efficiently due to vectorization. On the other hand for GPUs the SoA (structures-of-arrays) data layout is preferable for LBMs [22], since the streaming step takes more time than the collision step and streaming to neighbor cells that are stored close in memory is faster.

In this paper an attempt is made to implement 3D FSLBMs based on coarse corrective steps (CCS), the shift algorithm and the AoS data layout for the velocity and pressure. Furthermore, 3D texture is used for memory layout with (hardware supported) space-filling curves to load data as 3D blocks (tiles) in GPU shared memory. It is found that

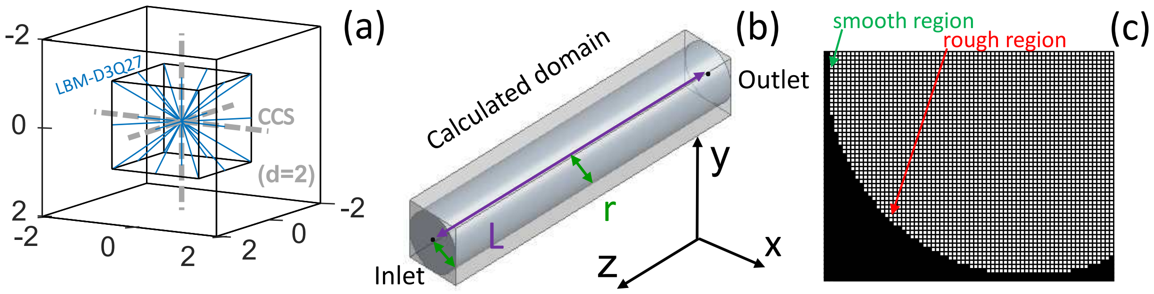

Figure 1: (a) D3Q27 stencils shown in the small box used for the LBM-predictive step. The coarse corrective step (CCS) has stencils with $d \geq 2$ shown by the dashed lines. (b) Pipe domain used for the calculations, where the radius r and length L fully define the geometry. (c) Pipe cross-sections used in the calculations with some smooth and rough regions.

CCSs extend the Re range of 3D FSLBMs without compromising the flow-physics; GPU device memory requirement is equivalent to NSE-based finite difference approach, i.e., the velocity vector and the pressure is stored after each time step and not the DFs. Additionally, the implementation of 3D FSLBM in this paper facilitates calculations with up to half billion lattice points at ≈3790 MLUPS (million lattice updates per second) on a single GTX 1080 Ti GPU with 11 GB memory and offers a good accuracy for resolving turbulence as shown by benchmark problems for pipes. All this is achieved with single precision calculations which are sufficient for the given flow problem as comparison to high quality DNS data shows.

2. Problem Statement

Fractional step methods [24] play an important role in solving CFD problems by dividing the equations into individual parts that can have separate numerical treatment [25]. In our particular case this implies a predictive step based on the incompressible LBM therefore we introduce its form based on the framework given in [10]. The evolution equation for the LBM DFs is

$$f_\alpha(\mathbf{x},t) - f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\delta_t, t-\delta_t) = \tag{1}$$

$$-\frac{1}{\tau}\left(f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\delta_t, t-\delta_t) - f_\alpha^{eq}(\mathbf{x}-\mathbf{e}_\alpha\delta_t, t-\delta_t)\right) \tag{2}$$

Here, the DFs f_α are defined on the D3Q27 lattice with velocity direction $\alpha \in \{0,1,..,26\}$ and coordinate vector $\mathbf{x}$. The above form of the evolution equation has its DFs $f_\alpha(\mathbf{x},t)$ calculated based on the streaming from $f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$ and $f_\alpha^{eq}(\mathbf{x}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$, however, its form is equivalent to the definition in [10] with substitutions $\mathbf{x} \longrightarrow \mathbf{x}+\mathbf{e}_\alpha\delta_t$ and $t \longrightarrow t+\delta_t$. Furthermore, τ is the relaxation time, $\mathbf{e}_\alpha$ is the direction stencil defined on the small cube of Figure 1. (a), δ_t is the time step and

$$f_\alpha^{eq}(\mathbf{x},t) = W_\alpha\left\{\rho + \rho_0\left[3\frac{(\boldsymbol{e}_\alpha\boldsymbol{u})}{c^2} + \frac{9}{2}\frac{(\boldsymbol{e}_\alpha\boldsymbol{u})^2}{c^4} - \frac{3}{2}\frac{\boldsymbol{u}^2}{c^2}\right]\right\} \tag{3}$$

is the equilibrium DF. Here, W_α is the lattice weight for the D3Q27 lattice, $\mathbf{u}$ is the velocity vector, ρ is the density (or equivalently pressure with $P = c_s^2\rho$), $c = \delta_x/\delta_t$ is the lattice speed and δ_x the lattice spacing. Furthermore, $\tau = 1$ is set in every predictive step of our method, therefore Eq. (2) simplifies to $f_\alpha^{eq}(\mathbf{x}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$ that can be evaluated by using Eq. (3). For the choice of $\tau = 1$ the equivalent viscosity is calculated as

$$\nu^* = \frac{2\tau-1}{6}\frac{\delta_x^2}{\delta_t}|_{\tau=1} = \frac{1}{6} \tag{4}$$

where $\delta_x^2 = \delta_t$ is set in our calculations. The above equations provide the reconstruction of DFs from the density and velocity, furthermore the pressure P and the predicted velocity $\mathbf{u}^*$ are calculated from the DFs with sum over the velocity directions as

$$P = \frac{1}{3}\sum_{\alpha} f_\alpha(\mathbf{x}, t) = \frac{1}{3}\sum_{\alpha} f_\alpha^{eq}(\mathbf{x} - \mathbf{e}_\alpha\delta_t, t - \delta_t),\ \mathbf{u}^* = \sum_{\alpha} \mathbf{e}_\alpha f_\alpha(\mathbf{x}, t) = \sum_{\alpha} \mathbf{e}_\alpha f_\alpha^{eq}(\mathbf{x} - \mathbf{e}_\alpha\delta_t, t - \delta_t). \tag{5}$$

With the above definitions the equivalent Navier-Stokes equation for small time steps (see appendix in [10]) is

$$\frac{\partial \boldsymbol{u}^*}{\partial t} = -\boldsymbol{u}^* \cdot \nabla\mathbf{u}^* - \nabla P^* + \nu^*\nabla^2\boldsymbol{u}^* \tag{6}$$

with the fixed viscosity of 1/6. Furthermore, Eq. (6) can be approximated with the time discretised form

$$\mathbf{u}^* \approx \mathbf{u}^k + \delta_t L_1(\mathbf{u}^k, \nu, P^{k+1}) \approx \mathbf{u}^k + \delta_t\left[\frac{1}{6}\nabla^2\mathbf{u}^k - \left(\mathbf{u}^k \cdot \nabla\right)\mathbf{u}^k - \nabla P^{k+1}\right], \tag{7}$$

where pressure and velocity are known from the previous time step k and $*$ stands for the predicted time step. Furthermore, in the limit of small time steps the replacement $\mathbf{u}^* \approx \mathbf{u}^k$ on the right hand side of Eq. (6) has been performed. Note, that $P^* = P^{k+1}$, i.e., pressure does not need an explicit correction, while $\mathbf{u}^*$ is the predicted velocity with viscosity $\nu^* = 1/6$. Since the LBM method with $\tau = 1$ provides solutions to Eq. (7) the solution to the general flow problem with viscosity ν is found by using a corrective step [15]. Consequently, the reverse (i. e., $\nu - 1/6 < 0$) diffusion equation

$$\mathbf{u}^{k+1} = \mathbf{u}^* + \delta_t L_2(\mathbf{u}^k, \nu) = \mathbf{u}^* + \delta_t\left(\nu - \frac{1}{6}\right)\nabla^2\mathbf{u}^k \tag{8}$$

is chosen as to cancel out the viscosity of 1/6 from Eq. (7) similarly as in Ref. [15]. It can be observed that by adding Eq. (7) and (8) the incompressible NSE is obtained with the kinematic viscosity ν for the fluid. Therefore if the reverse diffusion equation is solved for the corrective step one should obtain the solutions for the fluid flow. The above equation, however needs to be solved numerically, therefore its discrete form

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^*}{\delta_t} = L_2(\mathbf{u}^k, \nu) = \frac{\nu - \frac{1}{6}}{d^2\delta_x^2}\left[c_{0,N}\mathbf{u}^k\left(\mathbf{x}\right) - \sum_{i=1}^{N} c_{i,N}\sum_{\alpha=1}^{6}\mathbf{u}^k\left(\mathbf{x} + id\mathbf{e}_\alpha\Delta t\right)\right], \tag{9}$$

is used, where $d\delta_x$ is the grid spacing and $c_{0,N}$, $c_{i,N}$ are the coefficients for 3D central differences of order 2N. The above CCS form of the reverse diffusion equation uses exclusively stencils pointing beyond the nearest lattice cells as shown in Figure 1. (a) by the dashed lines. Such choice enables calculations for several orders of magnitude higher Re than for a conventional stencil with d=1. Furthemore, the construction for higher order (N>1) central difference correction is based on the lattice spacing d as well (see Eq. (9)). The improvement for the corrective step for $d > 1$ is attributed to reducing the time step sensitivity in the exponential solutions of reversed diffusion equations, and is discussed in the following subsection. For a complete summary we list all the relevant steps iterated for the solution of the flow problem within the FSLBM framework:

1. $f_\alpha(\mathbf{x}, t) = f_\alpha^{eq}(\mathbf{x} - \mathbf{e}_\alpha\delta_t, t - \delta_t)$ is calculated for all the directions α by using Eq. (3) with $\mathbf{u}^k$ and P^k (where $P^k = \rho^k/3$)
2. P^{k+1} and $\mathbf{u}^*$ are calculated with Eq. (5)
3. The new velocity $\mathbf{u}^{k+1}$ is corrected with $\mathbf{u}^{k+1} = (1 - b)\mathbf{u}^k + \mathrm{b}\left[\mathbf{u}^* + \delta_t L_2(\mathbf{u}^k, \nu)\right]$, where b is a blending factor smaller or equal to 1 and L_2 is defined in Eq. (9).
4. Boundary conditions are applied as shown in Figure 2 (a).

We now explain step 3. since it has the blending b and the central difference discretization L_2 that need to be clarified. Here b is equivalent to reducing the time step δ_t to $b\delta_t$ since we have

$$\mathbf{u}^{k+1} = (1 - b)\mathbf{u}^k + \mathrm{b}\left[\mathbf{u}^* + \delta_t L_2(\mathbf{u}^k, \nu)\right] \approx \mathbf{u}^k + b\delta_t\left[\nu\nabla^2\mathbf{u}^k - \left(\mathbf{u}^k \cdot \nabla\right)\mathbf{u}^k - \nabla P^{k+1}\right] \tag{10}$$

by substitution of Eq. (7) and using the definition of L_2 from Eq (8). By reducing the time step (or by performing the replacement $\delta_t \rightarrow b\delta_t$) one can improve the approximation to the flow problem by reducing the influence of higher order moments in the Chapman-Enskog expansion in terms of the small parameter $\varepsilon = b\delta_t$ [10]. Having the time

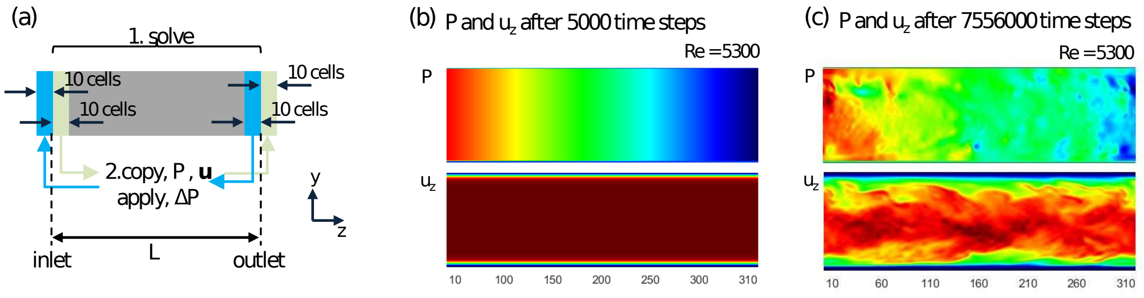

Figure 2: (a) Periodic boundary conditions used for the calculations. After each time step velocity and pressure data are copied as shown by the arrows and the pressure difference is applied. (b) Pressure and streamwise velocity plots after 5000 time steps with the applied periodic boundary conditions. The problem is initialized with a constant streamwise velocity of 0.5. (c) Same as (b) but after 7556000 time steps showing the periodic fully developed turbulent pipe flow.

reduced allows us to increase the input pressure until the average "Mach" number rises to about 1-2 (note that shocks do not form since the equilibrium DF in Eq. (3) is for incompressible flow). This allows us to increase the viscosity applied for the flow problem and aim for higher Re, therefore improve the accuracy. Finally, the discrete form of L_2 in Eq. (9) can have representations by higher order central difference stencils, which also improve the solution. For the rough pipe calculations in the results section N=1 (second order central difference) is chosen while for the smooth pipe calculations $c_{i,2,3} = 0.6c_{i,2} + 0.4c_{i,3}$ (a blend between N=2 and N=3) has been used. Such correction stencils offer improved solution accuracy while the calculations remain stable.

2.1. CCS and expansion solutions

Coarse corrective steps give the solution to 3D reverse diffusion equations in terms of expansion solutions as it is explained below. Therefore, CCS are explained mathematically for understanding the solution process. The expressions below are not explicitly used in the solution for FSLBM, they are just meant to provide a hint on the numerical behavior. For the 3D reverse diffusion Eq. 8 the assumption is that a solution can be found in the form

$$u(x, y, z, t) = X(x)Y(y)Z(z)T(t). \tag{11}$$

Furthermore, we treat only one of the velocity vector components from the reverse diffusion equation since the other have the same properties and are updated by CCS independently. Using Eq. (11) with the continuous version of the reverse diffusion equation (8) leads to the expansion solution

$$u(x, y, z, t) = \sum_{k,l,m} a_{klm} \sin\left(k\frac{\pi x}{L} + \varphi_0\right) \sin\left(l\frac{\pi y}{L} + \xi_0\right) \sin\left(m\frac{\pi z}{L} + \eta_0\right) e^{\frac{k^2+l^2+m^2}{d^2\delta_x^2}\left(\frac{\delta_t e^2}{6} - \nu\right)\pi^2 t} \tag{12}$$

which is a linear combination of orthogonal basis functions. Here, $L = 2d\delta_x$ is the cell length where we apply CCS and φ_0, ξ_0, η_0 are the phases that satisfy the boundary conditions of the discrete cubes of length $d\delta_x$. Generally the phases can be complex numbers, therefore the expansion solution is built from exponentially growing or decaying waves as well as periodic waves. Within the above solution one can observe that the larger integer d is, the slower the exponential growth in time which improves the stability of the velocity components. Furthermore, hyperbolic basis solutions (e.g. when $\sin\left(k\frac{\pi x}{L} + \varphi_0\right) \in \mathbb{C}$) have smaller variation due to L being increased in the denominator. Since $a_{klm} = a_k b_l c_m$ and the coefficients a_k, b_l and c_m correspond to Fourier modes of the general solution u, they are finite. The above properties favor increased ($d > 1$) discrete steps (CCS) for correcting the LBM viscosity.

3. Results

FSLBMs with coarse corrective steps have been validated on pipe domains as shown schematically in Figure 1. (b) and (c). Here, the relevant geometry parameters are the radius r and length L which define the number of lattice points for FSLBM in the (x,y,z) directions. Furthermore, the pipe domains have surfaces that vary in roughness, therefore

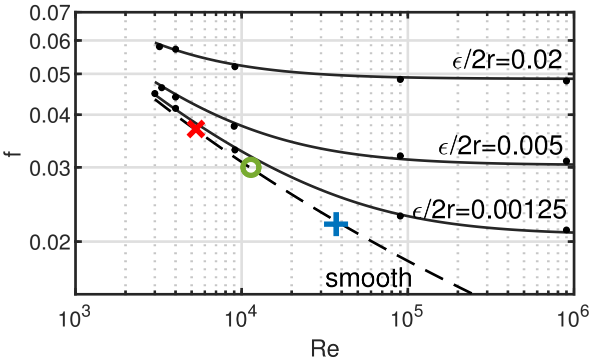

Figure 3: Comparison for the FSLBM friction factors f (symbols) to the reference results from the Colebrook-White equation (lines). Solid lines show f for the Colebrook-White equation with roughness ratio $\epsilon/2r$, while the dashed line gives f for the smooth pipe. All the calculated FSLBM results compare well to the reference results. Further details about the used pipe domains for FSLBM can be found in the text.

the approximation of a smooth surface depends mainly on the number of elements in the pipe cross section and the Reynolds number. Additionally, periodic boundary conditions have been defined by having a small pipe region with a length of 10 elements before the inlet as well as after the outlet as shown schematically in Figure 2 (a). Here, after solving the marked domain, the velocity vector is simply copied according to the arrows, while the pressure is also copied, however shifted with ΔP when copying from outlet to inlet and with $-\Delta P$ for the other case [26]. These small regions provide data for the LBM predictive step stencils as well as for the CCS stencils up to central difference order 10 ($N = 5$ with $d = 2$) extending outside the solved region. Consequently, the definitions above (and illustrated in Figure 2 (a)) provide periodic boundary conditions for the flow as can be observed for pressure and streamwise velocity in Figure 2 (b) and (c). Here, $Re = 5300$ and the first case shows a state close to the initialization, where only a few time steps are calculated, while the second case shows developed periodic turbulent flow. Furthermore, $d = 2$ for all of the calculations in this paper and $\Delta P = 0.01933$, $\nu = 0.0116$, $b = 0.03$, $L = 300$ and $r = 50$ for this initial test case.

3.1. Comparison for the friction factors

Pipe friction factors are calculated with FSLBM for rough pipe domains on Cartesian grids as shown in Figure 1 (c). Here, the surface roughness to diameter ratio ($\epsilon/2r$) influences the friction factors, therefore the calculations with different domain dimensions are appropriate for validating FSLBM. For comparing to the Colebrook-White equation [27] the FSLBM friction factors are defined as $f = 4r(\bar{P}_1 - \bar{P}_2)/3\rho_1\bar{u}_z^2 L$. Where, the average pressures $\bar{P}_1$ and $\bar{P}_2$ are defined at the inlet and outlet respectively (see Figure 2 (a)). Furthermore the average streamwise velocity $\bar{u}_z$ as well as the density $\rho_1 = 1$ is used. Additionally, the Reynolds number is defined as $Re = 2r\bar{u}_z/\nu$, and can be modified by changing r, or changing the lattice viscosity ν, or implicitly through $\bar{u}_z$ by adjusting ΔP=$\bar{P}_1 - \bar{P}_2$. Figure 3 shows the friction factor based comparison for FSLBM calculated pipe domains with various surface roughness (symbols) to results from the Colebrook-White equation (lines). The dots show FSLBM results for three different pipe domains, i.e., r=25, L=150 (top dots), r=50, L=300 (mid dots) and r=100, L=600 (bottom dots). Furthermore, for the above three pipe domain calculations $\Delta P = 0.01$, $d = 2$, N=1 (order 2 accurate coarse corrective steps are used), and Re is changed through the lattice viscosity ν. A very good agreement is found with the results from the Colebrook-White equation (thin solid lines) for all the cases, which shows that FSLBM can calculate the influence of surface roughness for pipes. Since the radius $r = 25$ for the first pipe is doubled and quadrupled for generating the second and third pipe domain, the relative roughness to-area changes from 1 to 1/4 and 1/16 respectively. Therefore the roughness ratio $\epsilon/2r = 0.02$ scales to its 1/4-th when r is doubled and to 1/16-th when r quadrupled as shown in Figure 3. Further calculations on smooth pipe domain-Re configuration are also shown in Figure 3 with $r = 50$, $L = 300$ (× symbol), $r = 100$, $L = 600$ (O symbol) and $r = 200$, $L = 1200$ (+ symbol) and compare well to the reference smooth result (dashed line). For such calculations the parameters are explained in the next subsection.

Re	L	r	b	d	$3\Delta P$	ν	$\bar{u}_z$	Re_τ	y^+
5300	300	50	0.01	2	0.058	0.0111	0.59	181	1.8
11300	600	100	0.01	2	0.05	0.0108	0.61	350	1.75
35200	1200	200	0.01	2	0.12	0.0125	1.1	936	2.3

Table 1: Smooth pipe FSLBM calculation test cases (rows) and their corresponding parameters ($L, r, b, d, \Delta P, \nu$). The mean streamwise velocity $\bar{u}_z$, friction Reynolds number Re_τ and y^+ are calculated result quantities.

3.2. Smooth pipe calculations

The smooth pipe domains in this paper have radii and lengths of $r = 50$, $L = 300$; $r = 100$, $L = 600$ and $r = 200$, $L = 1200$ respectively while for FSLBM the corrective step central difference coefficients $c_{i,2,3} = 0.6c_{i,2} + 0.4c_{i,3}$ have been used. The calculated Reynolds numbers and the list of corresponding parameters can be found in Table 1. Here, b has been set to 0.01 for all cases and the CCS step length $d = 2$. Additionally, the calculations for the first two rows in Table 1 have similar average velocities, while for the last case the velocity is $1.1\,\mathrm{ms}^{-1}$ in physical units that corresponds to 1.1 Ma in lattice units since $c = \delta_x/\delta_t = 1$. Since FSLBM is incompressible, the large Ma does not induce shocks, furthermore, the average velocity shows robustness for such calculations. The above parameters have been chosen to yield Re and friction Reynolds numbers (Re_τ) close to pipe flow calculations and experiments in literature [28, 29, 30]. Furthermore, the calculated y^+ values between 1.75 and 2.3 show that FSLBM calculations with parameters in Table 1 correspond to implicit LES. Based on the three set of results with parameters in Table 1 we have validated the statistically relevant quantities like turbulence spectra, average streamwise velocity, average pressure, and the root-mean-square (RMS) of the turbulent fluctuations, i.e., the Reynolds stresses. Furthermore, the turbulence is fully developed since the mean velocity and mean Reynolds stresses do not change in time in our analysis (not shown). The FSLBM calculated turbulence spectra can be found in Figure 4 (a) for the three cases. Here, the time dependent streamwise velocity from the symmetry point of the pipes, (e.g. (25,25,150) for the smallest pipe case) has been used for calculations. The turbulent kinetic energy is calculated as $E(\kappa) = \bar{u}_z \left|\mathcal{F}\left\{u_z(r, r, L/2, t)\right\}\right|^2 / (4\pi\Delta t)$ where $\bar{u}_z$ is the average streamwise velocity, $\mathcal{F}\left\{\cdot\right\}$ is the Fourier transform and $\Delta t = 50b\delta t$ is the time step. Furthermore, $\delta t = 1$, the sampling time interval is 50 time-steps, $b = 0.01$ and the Taylor hypothesis (i.e.($E(\kappa) \propto \left|\mathcal{F}\left\{u_z\right\}\right|^2$) has been used. The results in Figure 4 (a) are normalized to the velocity scale $u_\eta = (\epsilon\nu)^{0.25}$ and length scale $\eta = \left(\nu^3/\epsilon\right)^{0.25}$ where the dissipation rate $\epsilon = 4\pi^2\nu \int f_1^2 \left|\mathcal{F}\left\{u_z(r, r, L/2, t)\right\}\right|^2 \mathrm{d}f_1 / \left(\bar{u}_z^2\Delta t\right)$, $\kappa = 2\pi f_1/\bar{u}_z$ and f_1 is the temporal frequency. The above scale normalization approximately collapses the inertial sub-range for all three cases into the thin solid line shown in Figure 4 (a), which has the $-5/3$ Kolmogorov scaling. The integral scales before and the dissipated scales after the inertial sub-range can be identified as well which qualitatively validates our results to typical spectra for turbulent flows. The validation for the average velocity has been performed in Figure 4 (b) where the cases for $Re \approx 5300$ and $Re \approx 35200$ are shown. The FSLBM calculations (solid lines) compare well to the corresponding sampled DNS data from Ref. [28] shown by the dashed lines. Furthermore, the laws of the wall with corresponding expressions are shown for reference and delimit the flow into the linear and logarithmic range. The calculation for the average velocity is based on the spatial and temporal average $\bar{\bar{u}}_z$ on the longitudinal mid-section ($x = r$) of the pipe domain, i.e. $\bar{\bar{u}}_z(y) = \int\int u_z(r, y, z, t)\mathrm{d}t\mathrm{d}z/L$ (see Figure 2 (a) for the illustration of this domain section). Furthermore, the friction velocity $u_\tau = \sqrt{\nu\bar{\bar{u}}_z(0.5\delta_x)/\,(0.5\delta_x)}$ with $u^+ = \bar{\bar{u}}_z(y)/u_\tau$ and $y^+ \in [0.5\delta_x u_\tau/\nu, r\delta_x u_\tau/\nu]$, where $\delta_x = 1$.

After the above basic validation a more detailed analysis is performed for the average pressure and Reynolds stress of the turbulent flow. Here, we consider in detail the case of $Re = 5300$ from Table 1 and compare the average pressure, pressure fluctuations and velocity fluctuations normalized by the friction velocity to sampled results from Ref. [28]. The average for the velocity (Figure 4 (b)) and pressure (Figure 5 (a)) is calculated by using $\bar{\bar{X}}(y) = \int\int X(r, y, z, t)\mathrm{d}z\mathrm{d}t/L$. For the case of Figure 5 (a), X is replaced with p and we have $p^+ = \bar{\bar{p}}(y)/\,(u_\tau)^2$. Furthermore, the RMS for the pressure and the velocity vector components are defined as

$$X_{rms}(y) = \sqrt{\int\int \left|X(r, y, z, t) - \bar{\bar{X}}(y)\right|^2 \mathrm{d}t\mathrm{d}z/L}, \tag{13}$$

and used for the normalized pressure $p^+_{RMS} = p_{RMS}/\,(u_\tau)^2$ and velocities $u^+_{x,y,z,RMS} = u_{x,y,z,RMS}/u_\tau$. Finally, the

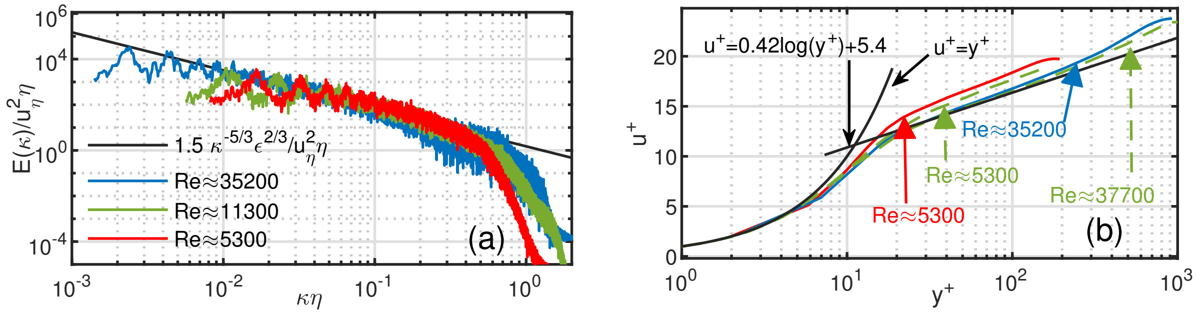

Figure 4: (a) Turbulence spectra of the streamwise velocity u_z at the x,y,z symmetry point of the pipe domains for $Re \approx 5300, 11300$ and 35200. (b) Average velocity calculated with FSLBM for $Re \approx 5300$ and $Re \approx 35200$ (solid) compared to spectral DNS data sampled from Ref. [28] and to the law of the wall (solid line marked with expressions).

normalized average mixed velocity fluctuation are calculated as $\left\langle u_z u_y \right\rangle^+ = \int\int \left| u_z(r, y, z, t) u_y(r, y, z, t) \right| \mathrm{d}t\mathrm{d}z / \left((u_\tau)^2 L \right)$ since $\bar{\bar{u}}_x(y) = \bar{\bar{u}}_y(y) = 0$. The above normalized quantities have been compared to DNS results sampled from Ref. [28] shown by the dashed lines in Figure 5 (a)-(f). Here a very good agreement is found considering that FSLBM for the presented grid is an implicit LES. For p^+ there are small differences in the minimum pressure dip which can result from slightly underestimated dissipation of the FSLBM method, i.e., the CCS order is slightly high and underestimates the required viscosity (see the definition of $c_{i,2,3}$ in Section 2 for the CCS order). Furthermore a second cause is the under-resolved wall region since there is a $y^+ = 1.8$ and there are only 3 elements for $y^+ < 10$ which is generally not recommended [31]. The overestimation for the streamwise fluctuations as well as the mixed velocity (Figure 5 (c) and (f)) in the buffer layer ($y^+ \approx 15$) is a result of the under-resolved wall region. Such overestimation is not observed if the wall region is refined or Re is reduced (not shown). The good qualitative and quantitative agreement for FSLBM and the reference results in Figure 5 (a)-(f) shows promising behavior of FSLBM for the calculation of Reynolds stress which is a good benchmark for turbulence, as well as for the pressure fluctuation that can be relevant for the noise sources.

A further analysis can be observed in Figure 6 (a)-(f), where on the top the isosurfaces for the Q-criterion ($Q = 0.001$) colored by the stream-wise velocity and on the bottom the streamwise velocity for the pipe crossections are shown. Furthermore, three Reynolds number cases from Table 1 have been considered: Figure 6 (a), (d) have $Re = 5300$, Figure 6 (b), (e) have $Re = 11300$ and for Figure 6 (c), (f) $Re = 35200$. In Figure 6 (a)-(c) it can be observed that vortex structures are formed in the close proximity of the wall. This paring of vortices starts from the buffer layer and approximately decays after $y^+ \approx 100$ for all the cases. The corresponding streamwise velocity for the pipe crossections is shown in Figure 6 (d)-(f) for the three cases. Again it can be seen that sizes for the flow vortices are smallest in the close proximity of the wall. Furthermore, the size of the structures becomes smaller as Re and the grid size is increased. Such behavior of the scales is observed in several wall bounded flows for channels and pipes [28, 29, 33] as well as external flows over wings [34, 35].

Finally, the validation for the Reynolds stress for all the considered smooth pipe cases can be found in Figure 7 (a)-(c). Here, the average streamwise, normal and mixed velocity fluctuations are compared and the amount of fluctuation increases as Re increases as marked by the arrows. Additionally, solid lines are for FSLBM results and dashed lines are for the sampled results from Ref. [28]. Since the DNS data shown by the dashed lines is considered very accurate, the FSLBM streamwise fluctuations are overestimated between $10 < y^+ < 30$ in Figure 7 (a) and some irregular behavior is observed in this region in Figure 7 (c). These effects are a consequence of the under-resolved wall region as for the previously explained case of $Re = 5300$. Furthermore, there are generally small differences between the reference results (dashed lines) and FSLBM (solid lines) for the case of $y^+ > 30$ and especially for increased Re. Here, if we consider that $Re = 5300$, 11700 and 37700 have been reported for the reference case and our results have $Re = 5300$, 11300 and 35200 respectively the slight differences in Re are the explanation for the underestimated fluctuations. In spite of the mentioned small differences we can observe a very good qualitative and quantitative agreement between the reference case and FSLBM calculations, showing that FSLBM is a reliable solution for resolving turbulent fluctuations with GPUs.

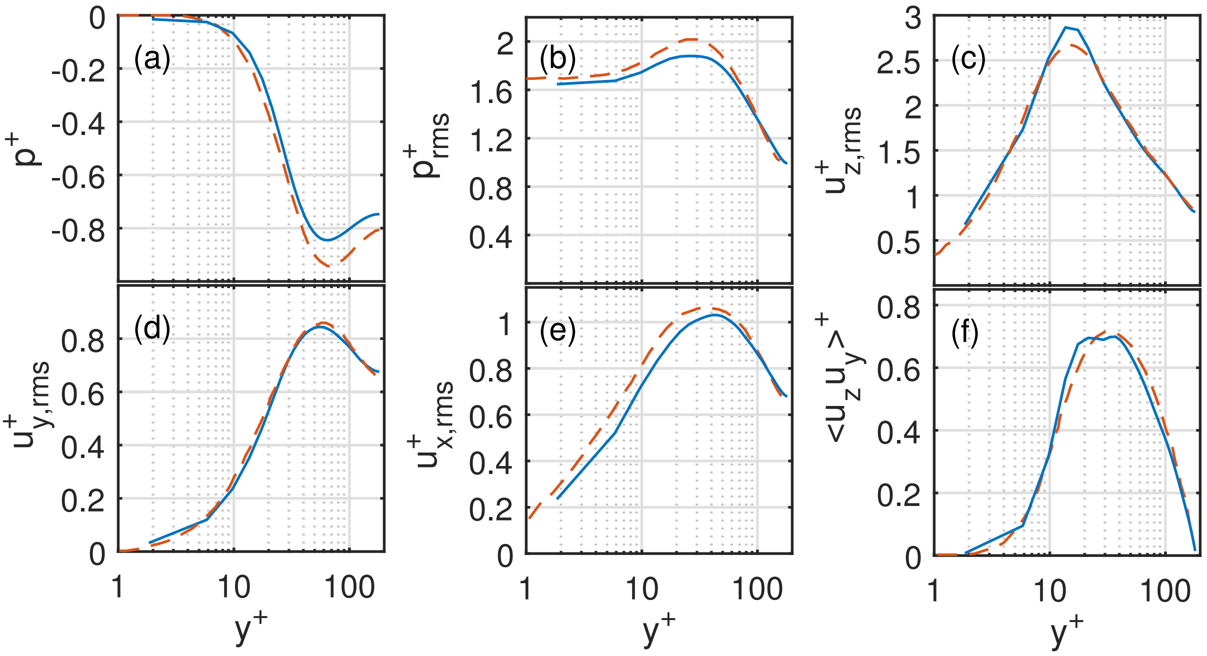

Figure 5: Comparison of average FSLBM results (solid) to sampled data from Ref. [28] (DNS results). The average pressure (a), average fluctuating pressure (b), average streamwise fluctuating velocity (c), average normal fluctuating velocity (d), average tangential fluctuating velocity (e) and average mixed y-z velocity term (f) agree well between FSLBM and the reference results. The main differences are related to the under-resolved wall region in FSLBM.

3.3. Performance

The performance a CFD code can be influenced by several factors: how the governing equations are formulated, the discretization order (lattice type for LBM), the hardware used for calculations and the memory access patterns are some of most relevant. For the implementation of LBMs these factors influence the maximum amount of elements that can be stored as well as the million lattice updates per second (MLUPS) that are quantities used to measure efficiency in this paper. For the maximum uniform grid size of LBMs an ideal estimate is calculated by using M/(4·V), where M is the memory in bytes V is the number of stored velocity DFs or hydrodynamic variables and 4 stands for the size of single precision floating point in bytes. If we consider M=1 GB, a D3Q27 LBM can have ≈ 9.26 million elements/GB of memory, while for D3Q27 FSLBM this number is ≈ 62.5 million elements/GB (in practice 56.6 Million/GB with the 1-byte 3D mask containing the boundary conditions and with 2D helper-memory blocks for the output of velocity and pressure). Consequently, FSLBM can perform on larger grids than most current implementation of LBMs. A LES on the maximum possible grid (580x580x1693 elements) with the Nvidia GTX 1080 Ti GPU is shown in Figure 8 for *Re* ≈ 11000. Here, longitudinal and transverse section planes are used for illustrating the streamwise velocity of fully developed turbulent flow. Furthermore, for the above example ≈ 3790 MLUPS is measured, based on all the fluid and boundary points. A comparison of FSLBM performance on single GPUs in terms of grid size and MLUPS to recently reported calculations can be found in Table 2. Here, the lattice type, used GPU, grid size, MLUPS and the year of publication of the reported work is shown. It can be observed that to the knowledge of the author FSBM outperforms all the previously reported single GPU LBMs in terms of grid size as well as MLUPS, however it should be mentioned that the GTX 1080 Ti GPU used in this work is faster and has more memory bandwidth than some of the GPUs used for the other reported cases. Furthermore, the calculation with comparable single GPU performance (2960 MLUPS vs. the 3790 MLUPS for FSLBM) from Riesinger et al. uses a less connected lattice (D3Q19 vs. the D3Q27 for FSLBM) and a GPU with higher memory bandwidth (Tesla P100 vs. the GTX 1080 Ti for FSLBM). Since most of the performance becnchmarks have been evaluated on a D3Q19 lattice, FSLBM should perform even better than 3790 MLUPS on such. Indeed, preliminary benchmarks show that FSLBM has 4485 MLUPS on D3Q19 lattices

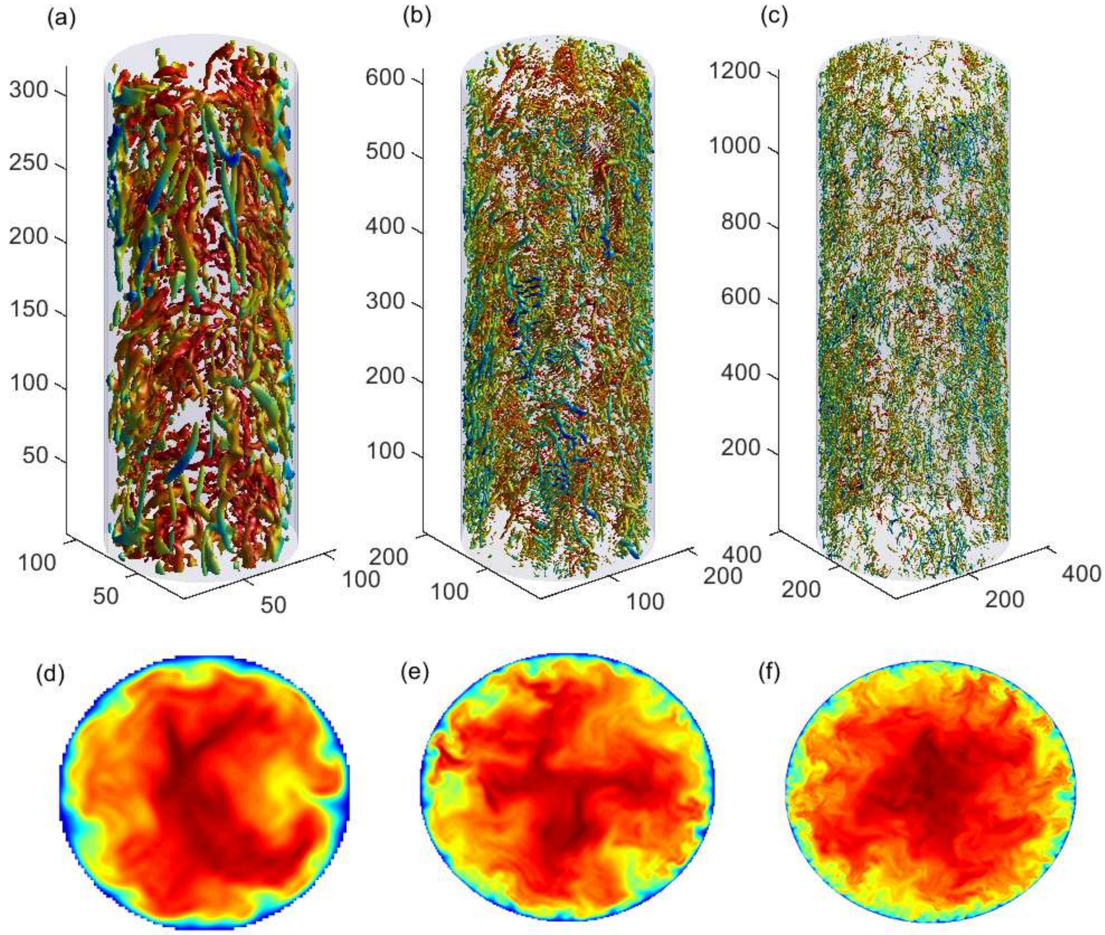

Figure 6: Streamwise velocity colored $Q = 0.001$ criterion [32] (top) showing turbulent structures for $Re \approx 5300$ (a), $Re \approx 11300$ (b) and $Re = 35200$ (c). Correspongig instantaneous streamwise velocity (bottom) in the mid-section of the pipe domain for $Re \approx 5300$ (d), $Re \approx 11300$ (e) and $Re \approx 35200$ (f).

with a GTX 1080 Ti and 6332 MLUPS with an RTX 2080 Ti. Finally, the efficiency of FSLBM can be attributed to the formulation of the flow problem with velocity and pressure evolution, the AoS implementation that can store the velocity and pressure into float4 CUDA memory and the usage of 3D threads with 3D texture and with float4 that have hardware support for space filling curves and are memory-bandwidth efficient. Additionally the tiling in 3D shared memory of sub lattices read using 3D texture also boosts MLUPS by 25%. In spite of the optimizations performed for the FSLBM kernel, the GPU calculations are mostly bandwidth limited since more than 75% of the memory bandwidth is constantly used in benchmarks. A further addressing of bandwidth limitation would require the optimization of the coarse corrective steps and perhaps new formulations for the equations.

4. Conclusion

Calculations with 3D fractional step lattice Boltzmann methods using coarse corrective steps for the reverse diffusion equation have been performed. The advantage of the method lies in the treatment of the correction step stencils that skip the nearest neighbors of the updated lattice cell, therefore enable increased Re range for the computations. Furthermore, periodic turbulent pipe flow calculations have been used for validating FSLBM friction factors against the Colebrook-White equation as well as validating the average velocity and Reynolds stresses against high quality

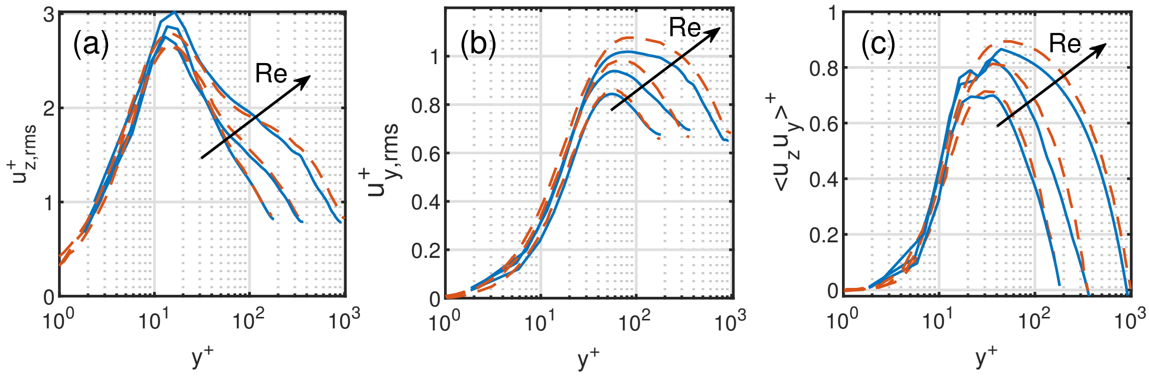

Figure 7: Comparison for streamwise (a), normal (b), and mixed (c), fluctuating velocity components for FSLBM (solid) and sampled DNS data from Ref. [28] and Ref.[30]. Three *Re* cases (5300, 11300 and 35200) are shown, where the amount of fluctuations increase with *Re* as marked by the arrows.

Reference	Lattice	GPU	Grid size	MLUPS	Year
McIntosh-Smith et al. [36]	D3Q19	GTX 780 Ti	$16.8 \cdot 10^6$	$\approx$ 1600	2014
Tran et al. [37]	D3Q19	Nvidia Tesla K20	$16.8 \cdot 10^6$	1210.63	2017
Riesinger et al. [38]	D3Q19	Nvidia Tesla P100	$56.6 \cdot 10^6$	2960	2017
Harwood et al. [39]	D3Q19	Nvidia GTX 1080 Ti	$7.2 \cdot 10^6$	1700	2017
Khani et al. [40]	D3Q19	Nvidia Tesla K40c	$7.1 \cdot 10^6$	1248 (d)	2018
Obrecht et al. [41]	D3Q19	Nvidia GTX Titan	$201 \cdot 10^6$	$\approx$2500	2014
FSLBM (this work)	D3Q27	Nvidia GTX 1080 Ti	$570 \cdot 10^6$	3790	2019

Table 2: Comparison of FSLBM performance based on grid sizes and MLUPS to recently reported calculations. Data are shown for single GPU and single floating point precision. (d) in the MLUPS column marks the calculations where double precision performance has been reported.

(spectral) DNS data. Additionally, the turbulence spectrum calculated from time-sampled velocity in the pipe midpoints has shown correct Kolmogorov scaling. The analysis has found good qualitative and quantitative agreement of FSLBM to the reference results proving that FSLBM can be a useful method for LES/DNS calculations. Additionally, FSLBMs enable calculations with $\approx 56.6 \cdot 10^6$ uniform elements per GB of GPU memory which is considered a storage efficiency record for current uniform 3D LBM implementations. From a computing efficiency standpoint FSLBM reaches 3790 MLUPS on an Nvidia GTX 1080 Ti GPU, which is again among the highest reported efficiency numbers on single GPUs.

References

[1] G. R. McNamara, G. Zanetti, Use of the Boltzmann equation to simulate lattice-gas automata, Phys Rev Lett 61 (1988) 2332–2335.

[2] A. Matyas, Validation of axisymmetric lattice Boltzmann methods for internal incompressible flow in complex 2d geometries, AIP Conf Proc 1863 (1) (2017) 560049.

[3] P. J. Dellar, Lattice Boltzmann algorithms without cubic defects in galilean invariance on standard lattices, J Comput Phys 259 (2014) 270 – 283.

[4] K. Suga, Lattice Boltzmann methods for complex micro-flows: applicability and limitations for practical applications, Fluid Dyn Res 45 (3) (2013) 034501.

[5] S. S. Chikatamarla, I. V. Karlin, Lattices for the lattice Boltzmann method, Phys Rev E 79 (2009) 046701.

[6] X. Nie, X. Shan, H. Chen, 47th AIAA Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics, 2009, Ch. A Lattice-Boltzmann / Finite-Difference Hybrid Simulation of Transonic Flow, pp. AIAA 2009–139.

[7] U. R. Alim, A. Entezari, T. Moller, The lattice-Boltzmann method on optimal sampling lattices, IEEE T Vis Comput Gr 15 (4) (2009) 630–641.

[8] M. Namburi, S. Krithivasan, S. Ansumali, Crystallographic lattice Boltzmann method, Sci Rep-UK 6 (2016) 27172 EP –.

[9] H. Chen, X. Shan, Fundamental conditions for n-th-order accurate lattice Boltzmann models, Physica D 237 (14) (2008) 2003 – 2008.

[10] X. He, L.-S. Luo, Lattice Boltzmann model for the incompressible Navier–Stokes equation, J Stat Phys 88 (3) (1997) 927–944.

[11] C. K. Aidun, J. R. Clausen, Lattice-Boltzmann method for complex flows, Annu Rev Fluid Mech 42 (1) (2010) 439–472.

[12] M. E. McCracken, J. Abraham, Multiple-relaxation-time lattice-Boltzmann model for multiphase flow, Phys Rev E 71 (2005) 036701.

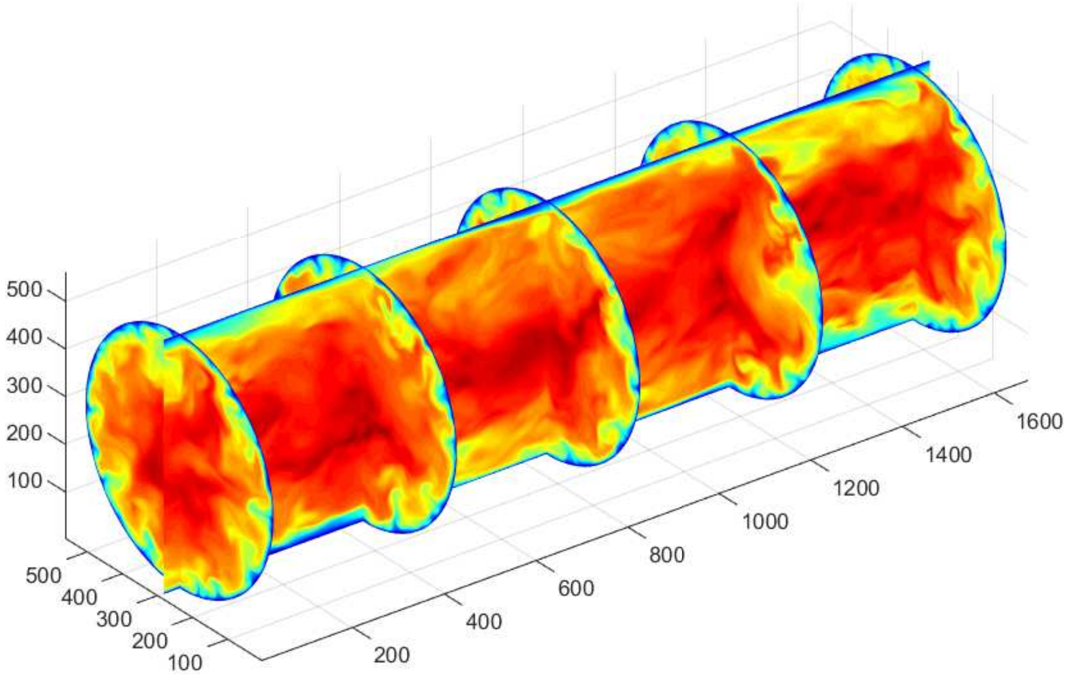

Figure 8: Streamwise velocity in a longitudinal and transverse section planes for fully developed turbulent pipe flow at $Re \approx 11000$.

[13] M. Geier, M. Schönherr, A. Pasquali, M. Krafczyk, The cumulant lattice Boltzmann equation in three dimensions: Theory and validation, Comput Math Appl 70 (4) (2015) 507 – 547.

[14] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems, Phys Rev 94 (1954) 511–525.

[15] C. Shu, X. Niu, Y. Chew, Q. Cai, A fractional step lattice Boltzmann method for simulating high reynolds number flows, Math Comput Simulat 72 (2) (2006) 201 – 205.

[16] A. Matyas, Large eddy simulation of flow around a cylinder with fractional step lattice Boltzmann methods, AIP Conf Proc 1978 (2018) 030029.

[17] S. Chen, G. D. Doolen, Lattice Boltzmann method for fluid flows, Annu Rev Fluid Mech 30 (1) (1998) 329–364.

[18] M. F. Barad, J. G. Kocheemoolayil, C. C. Kiris, 23rd AIAA Computational Fluid Dynamics Conference, AIAA AVIATION Forum, American Institute of Aeronautics and Astronautics, 2017, Ch. Lattice Boltzmann and Navier-Stokes Cartesian CFD Approaches for Airframe Noise Predictions, pp. AIAA 2017–4404.

[19] A. Matyas, S. Olajos, Z. Balint, S. Jones, Revised ansatz for ball-shape valve hydraulic force analysis, AIP Conf Proc 1648 (1) (2015) 850053.

[20] A. Matyas, B. Nagy, Numerical study of bipolar coordinate Jeffery-Hamel flow, AIP Conf Proc 1863 (1) (2017) 560077.

[21] K. Mattila, J. Hyväluoma, J. Timonen, T. Rossi, Comparison of implementations of the lattice-Boltzmann method, Comput Math Appl 55 (7) (2008) 1514 – 1524.

[22] C. Obrecht, F. Kuznik, B. Tourancheau, J. J. Roux, A new approach to the lattice Boltzmann method for graphics processing units, Comput Math Appl 61 (12) (2011) 3628 – 3638.

[23] M. Astorino, J. B. Sagredo, A. Quarteroni, A modular lattice Boltzmann solver for gpu computing processors, SeMA Journal 59 (1) (2012) 53–78.

[24] A. J. Chorin, A numerical method for solving incompressible viscous flow problems, J Comput Phys 2 (1) (1967) 12 – 26.

[25] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, J Comput Phys 59 (2) (1985) 308 – 323.

[26] J. W. Thomas, Numerical Partial Differential Equations: Finite Difference Methods, Springer New York, New York, NY, 1995, Ch. Hyperbolic Equations, pp. 205–259.

[27] C. F. Colebrook, Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws., J I Civil Eng 11 (4) (1939) 133–156.

[28] G. K. El Khoury, P. Schlatter, A. Noorani, P. F. Fischer, G. Brethouwer, A. V. Johansson, Direct numerical simulation of turbulent pipe flow at moderately high reynolds numbers, Flow Turbul Combust 91 (3) (2013) 475–495.

[29] O. Marin, R. Vinuesa, A. V. Obabko, P. Schlatter, Characterization of the secondary flow in hexagonal ducts, Phys Fluids 28 (12) (2016) 125101.

[30] M. P. Schultz, K. A. Flack, Reynolds-number scaling of turbulent channel flow, Phys Fluids 25 (2) (2013) 025104.

[31] Z. Ghiasi, D. Li, J. Komperda, F. Mashayek, Near-wall resolution requirement for direct numerical simulation of turbulent flow using multidomain chebyshev grid, Int J Heat Mass Tran 126 (2018) 746 – 760.

[32] J. Jeong, F. Hussain, On the identification of a vortex, J Fluid Mech 285 (1995) 69–94.

[33] J. Jimenez, S. Hoyas, Turbulent fluctuations above the buffer layer of wall-bounded flows, J Fluid Mech 611 (2008) 215–236.

[34] R. Vinuesa, R. Örlü, P. Schlatter, Characterisation of backflow events over a wing section, J. Turbul. 18 (2) (2017) 170–185.

[35] S. Hosseini, R. Vinuesa, P. Schlatter, A. Hanifi, D. Henningson, Direct numerical simulation of the flow around a wing section at moderate reynolds number, Int. J. Heat Fluid Fl. 61 (2016) 117 – 128.

[36] S. McIntosh-Smith, D. Curran, Evaluation of a performance portable lattice boltzmann code using opencl, in: Proceedings of the International Workshop on OpenCL 2013 &#38; 2014, IWOCL '14, ACM, New York, NY, USA, 2014, pp. 2:1–2:12.

[37] N.-P. Tran, M. Lee, S. Hong, Performance Optimization of 3D Lattice Boltzmann Flow Solver on a GPU, Sci Programming-Neth 2017 (2017) 16.

[38] C. Riesinger, A. Bakhtiari, M. Schreiber, P. Neumann, H.-J. Bungartz, A holistic scalable implementation approach of the lattice boltzmann method for cpu/gpu heterogeneous clusters, Computation 5 (4).

[39] A. R. Harwood, P. Wenisch, A. J. Revell, A real-time modelling and simulation platform for virtual engineering design and analysis, in: Proceedings of 6th European Conference on Computational Mechanics (ECCM 6) and 7th European Conference on Computational Fluid Dynamics (ECFD 7), Vol. MS17B, 2018, p. Paper 504.

[40] M. Khani, T.-H. Wu, A D3Q19 lattice Boltzmann solver on a GPU using constant-time circular array shifting, in: Proceedings of the 2018 international conference on parallel and distributed processing techniques & applications (PDPTA′18), Vol. 1-60132-487-1, 2018, pp. 383 – 389.

[41] C. Obrecht, P. Asinari, F. Kuznik, J.-J. Roux, High-performance implementations and large-scale validation of the link-wise artificial compressibility method, J Comput Phys 275 (2014) 143–153.