Flow Turbulence Combust (2015) 94:495–512
DOI 10.1007/s10494-014-9584-y

The Lattice Boltzmann Method Implemented on the GPU to Simulate the Turbulent Flow Over a Square Cylinder Confined in a Channel

Yusuke Koda · Fue-Sang Lien

Received: 2 June 2014 / Accepted: 4 November 2014 / Published online: 16 December 2014

Abstract The lattice Boltzmann method (LBM) is a relatively new method for fluid flow simulations, and is recently gaining popularity due to its simple algorithm and parallel scalability. Although the method has been successfully applied to a wide range of flow physics, its capabilities in simulating turbulent flow is still under-validated. Hence, in this paper, a 3D LBM code was developed to investigate the validity of the LBM for turbulent flow simulations through large eddy simulations (LES). A GPU enabled LBM code was developed, and validated against a benchmark test case involving the flow over a square cylinder in square channel. The flow results showed good agreement with literature, and speedups of over 150 times were observed when two GPUs were used in parallel. Turbulent flow simulations were then conducted using LES with the Smagorinsky subgrid model. The methodology was first validated by computing the fully developed turbulent channel flow, and comparing the results against direct numerical simulation results. The results were in good agreement despite the relatively coarse grid. The code was then used to simulate the turbulent flow over a square cylinder confined in a channel. In order to emulate a realistic inflow at the channel inlet, an auxiliary simulation consisting of a fully developed turbulent channel flow was run in conjunction, and its velocity profile was used to enforce the inlet boundary condition for the cylinder flow simulation. Comparison of the results with experimental and numerical results revealed that the presence of the turbulent flow structures at the inlet can significantly influence the resulting flow field around the cylinder.

Keywords GPU programming · CUDA · Lattice Boltzmann method · Parallel computing · Grid refinement · LES · Turbulent flows

Y. Koda · F.-S. Lien (✉)
Department of Mechanical and Mechatronics Engineering, University of Waterloo,
200 University Avenue West, Waterloo, ON, Canada, N2L 3G1
e-mail: fslien@uwaterloo.ca

1 Introduction

In Computational Fluid Dynamics (CFD), one of the most challenging tasks is to accurately and efficiently simulate turbulent flow. The main aspect about turbulent flows that make them difficult to predict is that it contains a wide range of length and time scales. Although it is possible to resolve all scales of motion by employing an extremely fine mesh through a direct numerical simulation (DNS), the resulting computational costs can become astronomical for practical flows. Therefore, it is more common to use turbulence models to simplify the computation. The most widely used and practical method of such is the Reynolds Averaged Navier-Stokes (RANS) approach, where the unsteadiness of the flow is modeled. This approach has proven to work well for certain flows, but the fact that a wide range of physics is accounted for by predetermined models make them non-ideal for a universal representation of turbulent flows. Large eddy simulations (LES) on the other hand, attempts to resolve the large scale motions, and use subgrid scale models to incorporate the effects of the small scale motions. This approach is more realistic than the RANS approach, since the features of the small eddies in a flow are relatively universal, and independent of the flow geometry.

The lattice Boltzmann method (LBM) is a numerical technique derived from the Boltzmann equation and kinetic theory, and is being recognized as an alternative to the methods based on the Navier-Stokes equations for flow computations. Contrary to the conventional CFD methods that solve the Navier-Stokes equations, the LBM employs discretized particle velocity distribution functions based on microscopic fluid physics to emulate the hydrodynamic flow field. In the LBM, fictitious particles are assumed at each node on the computational domain, where the particles are allowed to advect along restricted directions, and collide with other particles. The movement and collision rules are designed such that "coarse graining" the particle distributions will recover the weakly compressible Navier-Stokes equations. Recently, the LBM has been gaining popularity, due to its algorithmic simplicity, as well as its suitability for running on many-core systems such as the graphics processing unit (GPU).

In the context of LBM applied to turbulent flows, there have been some notable work on some fundamental and complex flows using LES [1–11], following the pioneering work by Hou et al. [12], who successfully incorporated the Smagorinsky subgrid model [13] to the LBM framework. However, investigations involving the application of LBM for LES on the GPU are still at an early stage, and further evaluation of its accuracy and efficiency is necessary.

Hence, this paper aims to investigate the validity of the LBM for simulating the incompressible turbulent flow over bluff bodies. This paper is organized as follows: §2 outlines the LBM algorithm; §3 consists of a validation study of our GPU code in terms of accuracy and performance; §4 presents the results from simulating the turbulent flow over a square cylinder confined in a channel.

2 The Lattice Boltzmann Method

The LBM is based on the lattice Boltzmann equation (LBE), which can be written as [14]:

$$f_i(\vec{r} + \vec{c}_i, t + 1) = f_i(\vec{r}, t) + \Omega_i \tag{1}$$

where $f(\vec{r}, t)$ is the probability distribution function of the particle populations at site $\vec{r}$, and Ω is the collision operator. For this study, the D3Q19 lattice, which consists of the following discrete velocities, was employed:

$$\vec{c}_i = \begin{cases} (0,0,0), & i = 0 \\ (\pm 1,0,0),(0,\pm 1,0),(0,0,\pm 1), & i = 1,2,\ldots,6 \\ (\pm 1,\pm 1,0),(\pm 1,0,\pm 1),(0,\pm 1,\pm 1), & i = 7,8,\ldots,18, \end{cases} \tag{2}$$

For the collision operator, the standard Bhatnagar-Gross-Krook (BGK) model was used. The BGK collision operator can be written as:

$$\Omega_i = -\omega(f_i(\vec{r}, t) - f_i^{eq}(\vec{r}, t)), \tag{3}$$

where f represents the distribution functions, f^{eq} represents the local equilibrium distribution functions, and ω is the relaxation rate. ω defines the viscosity of the fluid, as:

$$\nu = \frac{c^2}{3}\left(\frac{1}{\omega} - \frac{1}{2}\right)\delta t. \tag{4}$$

c is the lattice speed, which is defined as $\delta x/\delta t$, where δx is the lattice spacing, and δt is the time increment, and is unity for uniform lattice structures. f_i^{eq} can be written as:

$$f_i^{eq} = \rho t_i \left(1 + \frac{1}{c_s^2}\vec{c}_i \cdot \vec{u} + \frac{1}{2c_s^2}\mathbf{Q}_i : \vec{u}\,\vec{u}\right). \tag{5}$$

t_i is a constant weighting to take into account the different magnitudes of the lattice vectors, c_s is the speed of sound of the model, and the tensor $\mathbf{Q}_i$ is defined as:

$$\mathbf{Q}_i = \vec{c}_i\vec{c}_i - c_s^2\mathbf{I} \tag{6}$$

where $\mathbf{I}$ is the identity matrix.

The incompressible Navier-Stokes equations can be recovered from the LBE through the Chapman-Enskog analysis, which is essentially a formal multi-scaling expansion [14, 15]:

$$\frac{\partial}{\partial t} = \epsilon\frac{\partial}{\partial t_1} + \epsilon^2\frac{\partial}{\partial t_2} + \cdots \tag{7}$$

$$\frac{\partial}{\partial x} = \epsilon\frac{\partial}{\partial x_1} + \cdots. \tag{8}$$

The expansion parameter, ϵ, is the Knudsen number, defined as the ratio between the mean free path of a gas molecule and a macroscopic length scale [16]. Using this technique, the particle distribution function, f_i, can be expanded about the local equilibrium function, f_i^{eq}, as:

$$f_i = f_i^{eq} + \epsilon f_i^{neq}. \tag{9}$$

Here, f_i^{eq} is defined from Eq. 5, and must satisfy:

$$\sum_{i=0}^{q-1} f_i^{eq} = \rho \tag{10}$$

$$\frac{1}{\rho}\sum_{i=0}^{q-1} \vec{c}_i f_i^{eq} = \vec{u}, \tag{11}$$

where q is the number of discrete velocities associated with the node. The non-equilibrium distribution function, $f_i^{neq} = f_i^1 + \epsilon f_i^2 + \mathcal{O}(\epsilon^2)$, has the following constraints to ensure conservation of mass and momentum in the collision operator:

$$\sum_{i=0}^{q-1} f_i^k = 0 \tag{12}$$

$$\sum_{i=0}^{q-1} \vec{c}_i f_i^k = 0. \tag{13}$$

for both $k = 1$ and $k = 2$. Equation 3 can be Taylor expanded, and rewritten in the consecutive order of ϵ as [17]:

$$f_i^0 = f_i^{eq} : \mathcal{O}(\epsilon^0) \tag{14}$$

$$\left(\partial_{t_0} + \vec{c}_i \cdot \nabla\right) f_i^0 = -\tfrac{1}{\tau} f_i^1 : \mathcal{O}(\epsilon^1) \tag{15}$$

$$\partial_{t_1} f_i^0 + \left(\frac{2\tau - 1}{2\tau}\right)\left(\partial_{t_0} + \vec{c}_i \cdot \nabla\right) f_i^1 = -\tfrac{1}{\tau} f_i^2 : \mathcal{O}(\epsilon^2) \tag{16}$$

Using the constraints shown in Eqs. 10–13, and summing the first order expansion (15) for all i, the continuity equation can be obtained as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{u} = 0. \tag{17}$$

Next, by multiplying $\vec{c}_i$ to both sides of Eqs. 15 and 14 and combining them, the momentum equation is derived as:

$$\frac{\partial(\rho \vec{u})}{\partial t} + \nabla \cdot \left(\mathbf{\Pi}^0 + \frac{2\tau - 1}{2\tau}\mathbf{\Pi}^1\right) = 0, \tag{18}$$

where $\mathbf{\Pi}^0$ and $\mathbf{\Pi}^1$ are momentum flux tensors defined as:

$$\mathbf{\Pi}^0 = \sum_{i=0}^{q-1} \vec{c}_i \vec{c}_i f_i^0 = p\delta_{\alpha\beta} + \rho u_\alpha u_\beta \tag{19}$$

$$\mathbf{\Pi}^1 = \sum_{i=0}^{q-1} \vec{c}_i \vec{c}_i f_i^1 = \nu\left(\nabla_\alpha(\rho u_\beta) + \nabla_\beta(\rho u_\alpha)\right). \tag{20}$$

p is the pressure, and is related to density as:

$$p = c_s^2 \rho = \frac{1}{3}\rho. \tag{21}$$

The resulting momentum equation becomes:

$$\rho\left(\frac{\partial \vec{u}}{\partial t} + \vec{u}.\nabla\vec{u}\right) = -\nabla p + \rho\nu\nabla^2\vec{u}, \tag{22}$$

which is identical to the Navier-Stokes equations, given that the density fluctuations are small enough [14].

2.1 Smagorinsky subgrid model for the lattice Boltzmann method

In order to model the unresolved scales of motion at high Reynolds numbers, subgrid models are often employed. For LES, the subgrid models are applied after a spatial filtering operation, defined as [18]:

$$\overline{w}(x) = \int w(x)G(x, x')dx' \tag{23}$$

where w is a spatially dependent quantity, and G the kernel function. Based on this filtering function, the filtered particle distribution function, $\overline{f}_i$, can be defined as:

$$\overline{f}_i(x) = \int f_i(x)G(x, x')dx'. \tag{24}$$

Thus, the filtered LBE becomes:

$$\overline{f}_i(\vec{r} + \vec{c}_i, t + 1) = \overline{f}_i(\vec{r}, t) + \overline{\Omega}_i. \tag{25}$$

From here, it is assumed that the filtered particle distribution will relax towards a local filtered equilibrium distribution, which only depends on the local filtered macroscopic variables ($\overline{\rho}$ and $\overline{\vec{u}}$) [12]. Hence,

$$\overline{\Omega_i(f(\vec{r}, t))} = \Omega_i(\overline{f(\vec{r}, t)}), \tag{26}$$

where the equilibrium distribution function, $\overline{f}_{eq}$, is:

$$\overline{f}_i^{eq} = \overline{\rho} t_i \left(1 + \frac{1}{c_s^2}\vec{c}_i \cdot \overline{\vec{u}} + \frac{1}{2c_s^2}\overline{\mathbf{Q}}_i : \overline{\vec{u}}\,\overline{\vec{u}}\right). \tag{27}$$

For subgrid closure, the Smagorinsky model [13], which relates the eddy viscosity to the local strain rate tensor, was used. The governing equation for this model is:

$$\nu_{total} = \nu_0 + C\Delta^2 \left|\overline{\mathbf{S}}\right|, \tag{28}$$

where ν_{total} is the total effective viscosity, ν_0 is the molecular viscosity, Δ is the filter size, C is the Smagorinsky constant, and $\left|\overline{\mathbf{S}}\right|$ is related to the magnitude of the filtered strain rate tensor as:

$$\left|\overline{\mathbf{S}}\right| = \sqrt{2\overline{\mathbf{S}} : \overline{\mathbf{S}}}, \tag{29}$$

where

$$\overline{\mathbf{S}} = \frac{1}{2}(\nabla\overline{\vec{u}} + (\nabla\overline{\vec{u}})^\top). \tag{30}$$

In the LBM, the viscosity of the fluid is governed by the relaxation time. Hence, the Smagorinsky subgrid model can be implemented by locally adjusting the relaxation time in the LBE. From Eq. 4,

$$\nu_{total} = c\delta x \frac{2\tau_{total} - 1}{6}. \tag{31}$$

Combining this with Eq. 28,

$$\begin{aligned} \tau_{total} &= \frac{3}{c\delta x}\left(\nu_0 + C\Delta^2\left|\overline{\mathbf{S}}\right|\right) + \frac{1}{2} \\ &= \tau_0 + \frac{3}{c\delta x}C\Delta^2\left|\overline{\mathbf{S}}\right|, \end{aligned} \tag{32}$$

where τ_0 is the relaxation time obtained from the molecular viscosity. The filtered strain rate tensor can be computed directly from the non-equilibrium momentum flux tensor, $\boldsymbol{\Pi}^1$ [12]. Using Eq. 29, it can be seen that Eq. 20 can be written as:

$$\overline{\boldsymbol{\Pi}}^1 = \sum_{i=0}^{q-1} \vec{c}_i \vec{c}_i \overline{f}_i^1 = -\frac{2\overline{\rho}\tau_{total} c\delta x}{3} \overline{\mathbf{S}}. \tag{33}$$

Taking the tensorial magnitude, and defining $Q^{1/2} = \sqrt{\overline{\boldsymbol{\Pi}}^1 : \overline{\boldsymbol{\Pi}}^1}$,

$$Q^{1/2} = \frac{\sqrt{2}\tau_{total}\overline{\rho} c\delta x}{3} \left|\overline{\mathbf{S}}\right| \tag{34}$$

Substituting (32) into (34) leads to a quadratic equation for $\left|\overline{\mathbf{S}}\right|$, which can be solved to obtain:

$$\left|\overline{\mathbf{S}}\right| = \frac{-\tau_0\overline{\rho} c\delta x + \sqrt{(\tau_0\overline{\rho} c\delta x)^2 + 18\sqrt{2}\overline{\rho} C\Delta^2 Q^{1/2}}}{6\overline{\rho} C\Delta^2}. \tag{35}$$

Substituting (35) into (32) and assuming $\delta x = \Delta$ (implicit filtering), the final expression for τ becomes:

$$\tau_{total} = \frac{\tau_0}{2} + \frac{\sqrt{(\tau_0\overline{\rho} c)^2 + 18\sqrt{2}\overline{\rho} C Q^{1/2}}}{2\overline{\rho} c}. \tag{36}$$

Contrary to the finite volume Navier-Stokes solvers that require finite difference schemes to compute $\left|\overline{\mathbf{S}}\right|$, the LBM allows direct computation of $\left|\overline{\mathbf{S}}\right|$ using local variables.

3 Validation and Performance Evaluation

The LBM was implemented on the GPU using NVIDIA's CUDA, and validated using the benchmark case involving the 3D flow over a square cylinder in a channel. This problem was chosen for its simple and well defined geometry, as well as its abundance in simulation results. The results obtained from the GPU LBM code were compared against those summarized by Schäfer and Turek [19]. For this study, two NVIDIA Tesla M2070 GPUs were used in parallel. Although this GPU does not have graphics outputs, it has features such as ECC-protected memory and a larger device memory, making them ideal for large scale general purpose calculations. The technical specifications of this GPU are given in Table 1. The ECC feature was enabled for all test cases run in this study. The two GPUs were connected by PCIe x16 Gen2 to the Intel E5607 CPU at 2.26GHz, running Linux kernel 2.6.32, with CUDA version 5.0 and nvcc release 5.0 version 0.2.1221. Details on the GPU implementation can be found in [20].

The geometry of the simulation domain consisted of a square cylinder confined in a channel with a square cross section. The height and width of the channel was $4.1d$,

Table 1 Technical Specifications for the NVIDIA Tesla M2070 [21]

Compute Capability	2.0
Peak Single Precision Floating Point Performance	1030 GFLOPS
CUDA Cores	448
Memory Size	6 GB
Memory Bandwidth (ECC off)	150 GB/ s

where d is the diameter of the cylinder. The cylinder was placed so that its center is $5.0d$ from the inlet, and $2.0d$ from the bottom wall. The total length of the channel is $25.0d$. The four side walls of the channel, as well as the cylinder walls, were set to be no-slip walls. A schematic of the geometry is shown in Fig. 1. The inflow condition is specified with a prescribed velocity profile as:

$$U_{in} = 16U_m yz(H - y)(H - z)/H^4, V_{in} = 0, W_{in} = 0, \tag{37}$$

where H is the height of the channel. The Reynolds number, defined as $Re = \overline{U}d/\nu$, where $\overline{U} = 4/9U_m$, was set to 20, where the flow is expected to be steady. The summary parameters used for validation are the lift and drag coefficients, defined as $C_D = 2F_x/(\rho\overline{U}^2 dH)$ and $C_L = 2F_y/(\rho\overline{U}^2 dH)$, respectively.

Simulations were conducted using a uniform mesh with 10 nodes along the side of the cylinder, resulting in a total of 420,250 ($= 4.1 \times 4.1 \times 25 \times 10^3$) nodes by reference to Fig. 1. The moment extrapolation method [22] was used to apply the velocity inflow condition as well as the constant pressure outflow condition. Bounce back conditions were used to model the no-slip walls, and the lift and drag forces were computed based on the momentum exchange method [23]. The characteristic velocity, $\overline{U}$, was set to $0.04c$, and the outlet density was set to 1.0. Simulations were run for 20,000 time steps, which corresponds to 80 non-dimensional time units ($\tilde{t} = \overline{U}t/d$). The strong scalability of the multi-GPU implementation was also measured by running the simulations on one and two GPUs. Further increasing the number of GPUs can be achieved by coupling CUDA with OpenMPI, but due to the limited availability of the hardware, this study only concerns the use of up to two GPUs. Table 2 summarizes the numerical results obtained, as well as the total run time and execution speed in million lattice updates per second (MLUPS).

Both C_D and C_L were well within the range of values presented in [19]. In terms of performance, the current GPU implementation resulted in impressive execution speeds, with

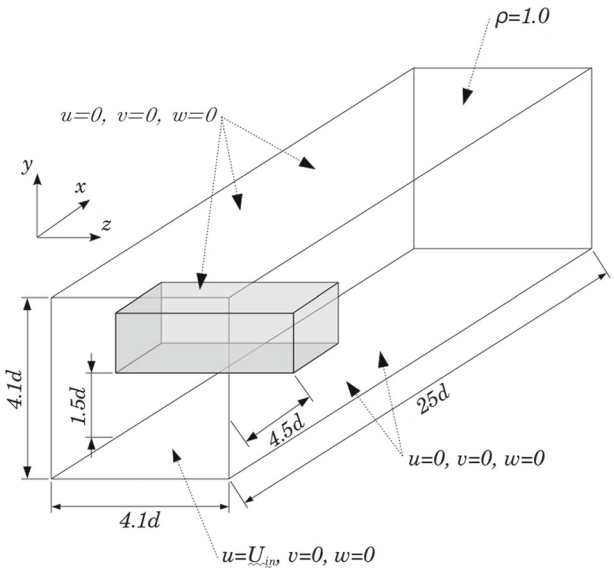

Fig. 1 Domain geometry and boundary conditions for 3D laminar flow over a square cylinder confined in a channel

Table 2 Numerical results for flow over a square cylinder at Re=20

	GPUs	C_D	C_L	time [s]	MLUPS
Uniform	1	7.614	0.0658	21	441
	2			12	788
Schäfer and Turek [19]		$7.50 - 7.70$	$0.06 - 0.08$		

441 MLUPS and 788 MLUPS for one and two GPUs, respectively. For reference, a serial code on the CPU was implemented, which showed an execution speed of 4.7 MLUPs when run on Intel E5607 CPU at 2.26GHz, compiled with the GNU g++ compiler with O2 optimization. This showed that speedups of two orders of magnitude were obtained from the GPU. The multi-GPU code showed a strong scaling efficiency of over 90 %, suggesting that the inter-GPU communication is mostly hidden by the computation. The scaling efficiency is expected to improve if a larger domain was used, or if the domain was split in a more sophisticated way that would minimize the amount of communication at the zone interface. In this study however, it was sufficient to prove that the code was capable of managing two GPUs in a reasonably efficient manner.

4 LES Using the LBM

This section is concerned about the performance of the LBM for turbulent flow simulations using LES. First, the popular benchmark case of the fully developed turbulent flow in an infinite channel was simulated using a periodic domain. The time averaged velocity and RMS velocities were compared against DNS results to assess the validity of the LES-LBM code in simulating turbulent flows. Then, the LES-LBM code was used to predict the flow over a square cylinder confined in a channel. For both cases, the Smagorinsky subgrid scale model with implicit grid filtering was used for turbulence modeling.

4.1 Turbulent channel flow at $Re_\tau = 180$

4.1.1 Problem description

The fully developed turbulent flow in a plane channel is one of the most fundamental cases for studying the nature of turbulence, and has been extensively studied both experimentally and numerically. It is an ideal benchmark test case for turbulence models, since DNS data is available for lower Reynolds numbers. For this flow, it is customary to define the flow in terms of the shear Reynolds number:

$$Re_\tau = \frac{u_\tau \delta}{\nu}, \tag{38}$$

where δ is half of the channel height, H, and u_τ is the friction velocity. The friction velocity is related to the wall shear stress, τ_w, by:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}. \tag{39}$$

Since the flow is fully developed, the stream-wise forces must balance, as:

$$-\frac{\partial p}{\partial x}\delta = \tau_w. \tag{40}$$

Combining (39) and (40),

$$-\frac{\partial p}{\partial x} = \frac{\rho u_\tau^2}{\delta}. \tag{41}$$

The effect of the pressure gradient can be incorporated into the LBM framework by introducing an additional step after the collision step, as [24]:

$$\tilde{f}_i^* = \tilde{f}_i + w_i \delta t \frac{3}{c^2}\frac{\partial p}{\partial x} c_i \cdot \hat{x}. \tag{42}$$

For this study, the shear Reynolds number was set to 180. The dimensions of the computational domain were set according to the comprehensive work by Moser et al. [25], with $2\pi H$ in the stream-wise direction, $\frac{2}{3}\pi H$ in the spanwise direction and H in the vertical direction. In this work, 62 nodes were allocated along H, yielding a mesh of $400 \times 130 \times 62$ nodes in the stream-wise, spanwise and vertical directions, respectively. The upper and lower surfaces of the domain were set to be no-slip by using the bounce back scheme, and the stream-wise and spanwise boundaries were set to be periodic. u_τ was set to $0.005c$, which leads to a $\frac{\partial p}{\partial x}$ of 4.03×10^{-7}. The standard Smagorinsky model with $C = 0.01$ was used for the subgrid scale model. For simplicity of implementation on the GPU, wall damping was omitted. The flow was initialized with a uniform stream-wise velocity, and a cubical obstruction was placed to initiate turbulence. The obstruction was removed after several thousand time-steps, and the simulation was run for a total of 3 million time steps. An additional run with 6 million time steps was also run to check for statistical convergence.

4.2 Results

The instantaneous flow field is shown in Fig. 2 in terms of spanwise and stream-wise velocities. Both quantities were normalized based on u_τ. The figure shows that the simulated flow is unsteady with significant 3D flow structures as indicated by the spanwise velocities even after three million time steps, which shows that the turbulence is self-sustaining.

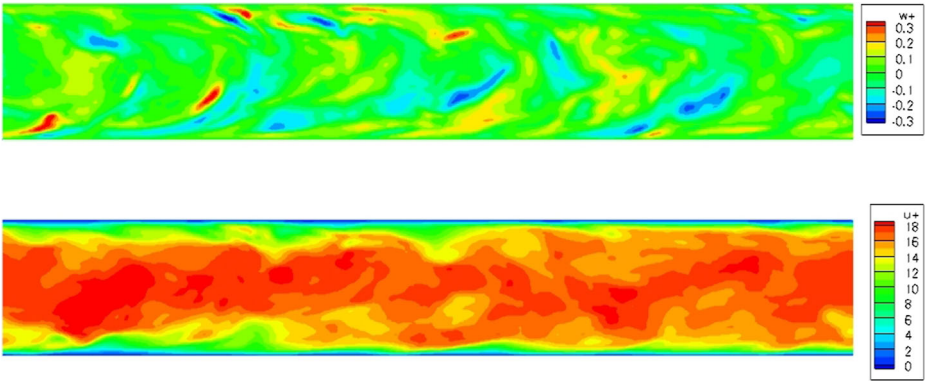

Fig. 2 Spanwise and stream-wise velocity contours on spanwise cross section. Both velocity components normalized based on u_τ

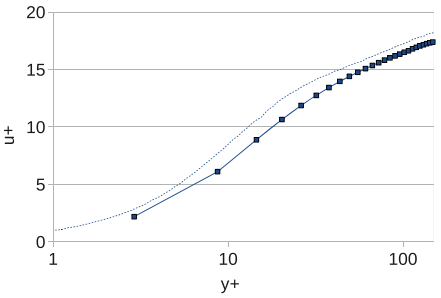

Fig. 3 Mean streamwise velocity profile in wall normalized units. Symbols: present work; *dotted line*: DNS [25]

The LES-LBM results were quantitatively compared against DNS results presented by Moser et al. [25] in terms of time averaged flow statistics (Figs. 3–4). For the mean streamwise velocity profile, the LES-LBM simulation predicted a notably lower velocity compared to the DNS results. However, the profile clearly shows the transition from the viscous sublayer to the log-law region, despite the coarse grid (minimum y^+ of 2.2). Since the wall shear stress is supposed to balance the pressure gradient, the under-prediction of mean streamwise velocity profile in Fig. 3 is a result of under-prediction of mass flow rate, likely caused by a combination of coarse meshes used and omission of a wall-damping function in the standard Smagorinsky's subgrid scale (SGS) model. The RMS profiles were in reasonable agreement with the DNS results. The peak location of the streamwise velocity fluctuations showed some discrepancy, but the cross-stream and spanwise velocity components showed very good correspondence. Although it would be interesting to investigate how much the results will improve with a finer grid, the present results were deemed adequate to validate the LES-LBM code in simulating wall-bounded turbulent flow.

4.3 Turbulent flow over a square cylinder in a channel at $Re_d = 3000$

4.3.1 Problem description

In the previous section, it was shown that the LES-LBM is capable of simulating the fully developed turbulent channel flow problem. In this section, the LES-LBM was used to predict the flow over a square cylinder confined in a channel. The Reynolds number, defined as $Re = U_{in}d/\nu$, where U_{in} is the average inlet velocity, was 3000, and the blockage ratio,

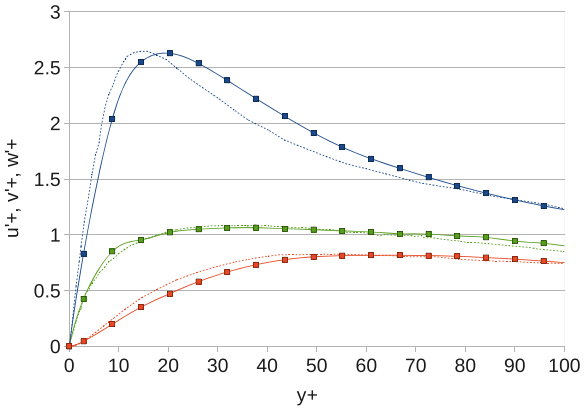

Fig. 4 RMS velocity profiles in wall normalized units. *Blue*: u'^+; *orange*: v'^+; *green*: w'^+. Symbols: present work; dotted line: DNS [25]

defined as d/H, where H is the channel height, was 20 %. This particular flow configuration was studied experimentally by Nakagawa et al. [26], and numerically using a finite volume method based LES by Kim et al. [27].

In the experimental work by Nakagawa et al. [26], the unsteady turbulent velocity field was measured using laser Doppler velocimetry (LDV). The time-averaged and phase-averaged statistics were summarized for various cylinder aspect ratios (width to height). Their experimental setup consisted of a closed water channel with streamwise, cross-stream, and spanwise dimensions of $30d$, $5d$, and $35d$, respectively. A rectangular cross-sectioned cylinder was installed on the centerline, where the distance between the test section inlet and the front side of the cylinder was $3d$. The flow was generated by a constant head tank, and the water was allowed to pass through a contraction before entering the test section with a mean velocity of U_{in}. The stream-wise and cross-stream velocity components were measured using a two-color four-beamed LDV.

Kim et al. [27] then conducted a numerical study in order to verify the experimental results of Nakagawa et al. [26]. They solved the three-dimensional incompressible Navier-Stokes equations, which were filtered by a box filter, on a non-uniform staggered Cartesian grid, using the finite volume method with the fractional time-stepping method for time integration. Their domain dimensions were made identical to that used in [26], except for the spanwise dimension, which was set to $1d$ with periodic boundary conditions. At the inlet, a uniform mean velocity profile with a thin boundary layer was imposed, and 'jittered' by using random numbers that are 6 % of U_{in} in RMS magnitude. A convective condition was employed to model the outlet. For turbulence modeling, a dynamic subgrid scale model, which uses two filters to correlate the subgrid scale stresses with the resolved turbulent stresses [28], was used.

In this study, the LES-LBM was used on a uniform grid, with the standard Smagorinsky model for subgrid closure. The stream-wise and cross-stream dimensions of the domain were set to $20d$ and $5d$, respectively. Two separate cases with spanwise dimensions of $1d$ and $3d$ were conducted, to evaluate its effect on the results. The cylinder location with respect to the inlet location was made identical to [26] and [27] (Fig. 5). Bounce-back conditions were used to apply no-slip conditions on the channel and cylinder walls, and the outlet was set to have constant pressure. For the inlet condition, instead of prescribing a predetermined velocity profile, a time accurate velocity profile based on a fully developed turbulent channel flow was used. This was done by running an infinite channel flow simulation in

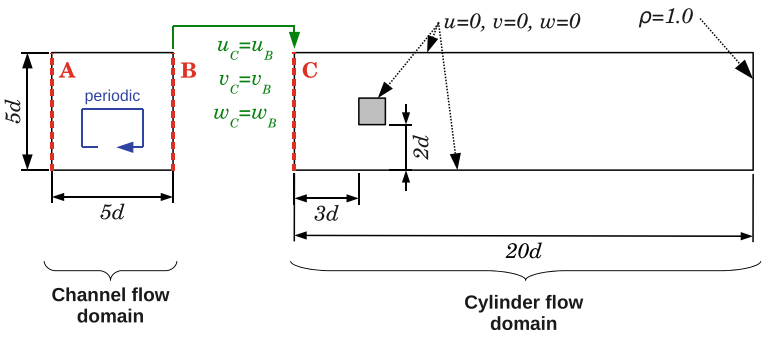

Fig. 5 Domain geometry and boundary conditions for 3D turbulent flow over square cylinder confined in a channel. Lines A and B indicate the periodic coupling for the infinite channel flow domain, and line C indicates the inlet location for the cylinder flow domain

parallel with the square cylinder simulation. At each time step, the velocity profile from the channel flow simulation was extracted, and prescribed as an inlet velocity condition for the square cylinder simulation, to allow for an accurate emulation of turbulent inflow in the channel.

To achieve this, the following method was used. For a 3D Cartesian grid, let the nodes be indexed by (i, j, k). Assuming the steam-wise coordinates for the lines B and C in Fig. 5 are $i = B$ and $i = C$, respectively, the distribution functions at the inlet of the cylinder flow domain can be written as:

$$f_i(C, j, k) = f_i^{eq}\left(\rho = \rho(C+1, j, k), \vec{u} = \vec{u}(B, j, k)\right). \tag{43}$$

This ensures that the velocity is prescribed based on the channel flow solution, while the density (pressure) is set to be consistent with the cylinder flow domain.

4.3.2 Results for turbulent channel flow

Prior to simulating the flow over a cylinder in a channel, the fully developed channel flow at the Reynolds number of interest was studied. Since the Reynolds number based on the diameter of the cylinder is 3000, the equivalent Reynolds number based on the channel height (Re_H) is 15000. The stream-wise dimension was set to H, and the spanwise dimension was set to $0.2H$ and $0.6H$, where $H = 200$ nodes. For example, in the case of $z = 0.6H$ (or $3d$), the grid size for the channel flow domain in Fig. 5 is 4,800,000 nodes ($= 200 \times 200 \times 120$ in the stream-wise, vertical and span-wise directions, respectively). Turbulence was initiated by placing a cubical obstacle in the channel, similar to the previous section. The main challenge in this simulation was that since the Reynolds number is specified in terms of the channel height, the shear Reynolds number (Re_τ) is unknown. This means that the pressure gradient required to drive the flow cannot be derived *a priori*. Hence, in this study, a proportional controller, which dynamically adjusts the pressure gradient to achieve the desired flow rate, was used.

The controller was designed to adjust the pressure gradient dynamically, based on the current flow rate in the channel. Letting the current cross-sectionally averaged stream-wise velocity be $u_{current}$, the target stream-wise velocity to be u_{target}, the stream-wise pressure gradient for the next time step to be $\partial p/\partial x|_{next}$, the current stream-wise pressure gradient to be $\partial p/\partial x|_{current}$, and the initial estimate for the stream-wise pressure gradient to be $\partial p/\partial x|_{init}$, $\partial p/\partial x|_{next}$ was adjusted as:

$$\frac{\partial p}{\partial x}|_{next} = \frac{\partial p}{\partial x}|_{current} + \frac{u_{target} - u_{current}}{u_{target}} K_P \frac{\partial p}{\partial x}|_{init} \tag{44}$$

where K_P is a constant. In this study, the above correction was applied every 1000 time steps, with K_P set to 0.05.

u_{target} was set to $0.06c$, and the simulation was run for a total of 5 million time steps. The average cross-sectional stream-wise velocity and the applied pressure gradient was plotted for each time step for the $z = 1d$ case (Fig. 6). It can be seen that the flow rate approaches the target value in the first 2 million time steps, but does not completely converge to a singular value. This behavior was expected, since the turbulent flow is random in nature, and applying a constant pressure gradient does not lead to a constant flow rate. Therefore, to obtain the pressure gradient that achieves the desired average flow rate, the time averaged value of the final 3 million time steps was taken.

In terms of the resulting flow field, it was confirmed that the turbulent flow was self-sustaining, with coherent structures developing in the near-wall regions for both cases

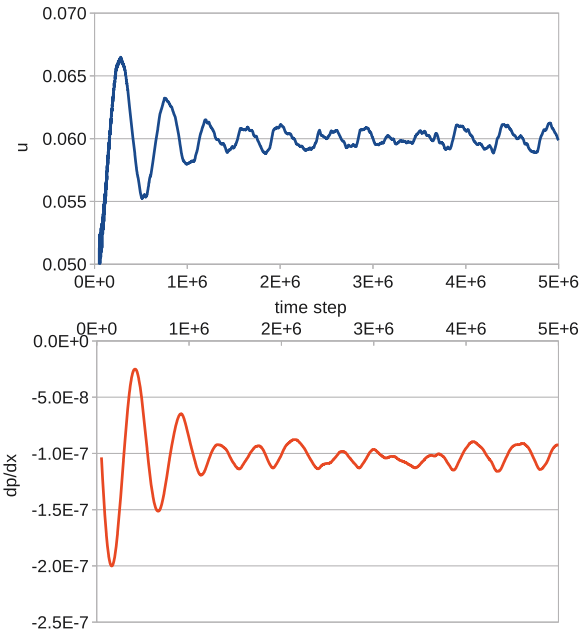

Fig. 6 Cross-sectionally averaged stream-wise velocity and stream-wise pressure gradient, plotted against time steps

(Fig. 7). Qualitatively, the general flow structures near the wall are similar, but the $z = 3d$ case displayed a higher concentration of the stream-wise vortical structures. This observation is reasonable considering the larger range of wavelengths that are allowed to exist in the spanwise direction for the $z = 3d$ case.

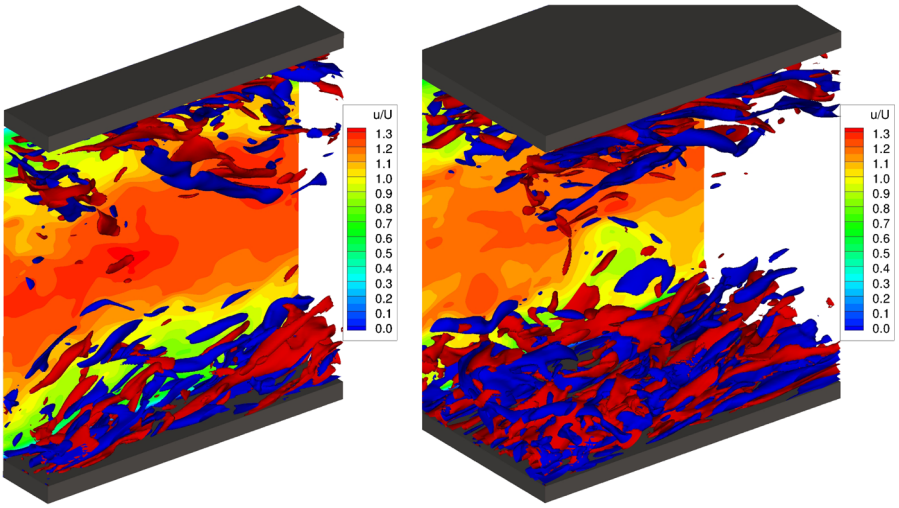

Fig. 7 Vorticity and velocity plots for fully developed turbulent channel flow at $Re_H = 15000$. Isosurface plots show the stream-wise vorticity (red: $\omega_x = 6.67$; blue: $\omega_x = -6.67$), normalized based on U_{in} and H. Contour plot shows the stream-wise velocity magnitude, normalized based on U_{in}. Left: $z = 1d$; right: $z = 3d$

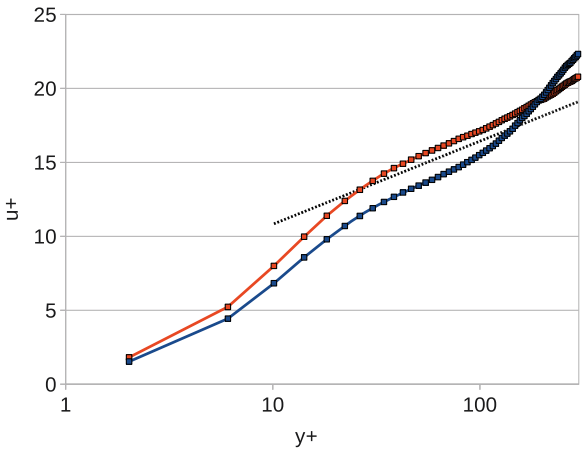

Fig. 8 Mean stream-wise velocity profiles in wall normalized units. Symbols: present work; dotted line: log-law. Blue: $z = 1d$; orange: $z = 3d$

In Fig. 8, the mean stream-wise velocity profiles for the two cases were plotted, and compared against the log-law profile, defined as:

$$u^+ = \frac{1}{\kappa} ln(y^+) + B, \tag{45}$$

where $\kappa = 0.41$ and $B = 5.2$. It was found that the $z = 3d$ case showed better agreement with the log-law profile than the $z = 1d$ case. It was interesting to see that although the $z = 1d$ case showed qualitatively acceptable results, the mean velocity profile displayed

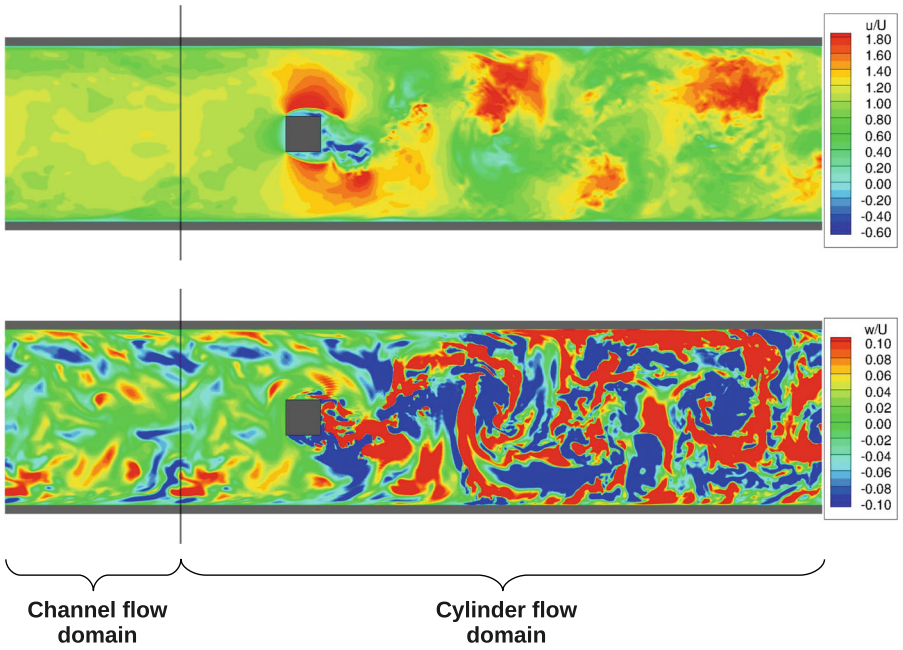

Fig. 9 Instantaneous stream-wise *(top)* and spanwise *(bottom)* velocity components. Black vertical line shows the boundary between the channel flow domain and the cylinder flow domain

an obvious discrepancy for $y^+ > 100$, where the velocity gradient rose sharply above the log-law profile.

4.3.3 Results for flow over a square cylinder

In the previous section, a dynamic controller was developed to simulate the fully developed turbulent channel flow at $Re_H = 15000$. The resulting flow fields showed reasonable agreement with the log-law, and the pressure gradient required to achieve the desired flow rate was obtained. With this knowledge, it is possible to simulate the flow over a square cylinder confined in a channel, as outlined in Section 4.3.1 (Fig. 5). Consistent with the channel flow domain shown in Fig. 5, in the case of $z = 0.6H$ (or $3d$), the grid size for the cylinder flow domain is 19,200,000 nodes ($= 800 \times 200 \times 120$ in the stream-wise, vertical and span-wise directions, respectively). Simulations were run for a total of 2 million time steps, where turbulent statistics were collected after 1 million time steps. Similar to the previous section, the effect of the spanwise dimension was analyzed by setting up two simulations with spanwise dimensions of $1d$ and $3d$. Simulations were also run with a simplified inlet condition, where the inlet velocity is set to U_{in}, to investigate its influence on the results.

The qualitative validity of the present setup was evaluated by observing the instantaneous flow fields. Figure 9 shows the instantaneous stream-wise and spanwise velocity components at a spanwise cross section. The channel flow and the cylinder flow domains are connected, to highlight the smooth transfer of the flow conditions at the domain boundary. Careful observation shows that the flow field in the inlet region of the cylinder flow domain

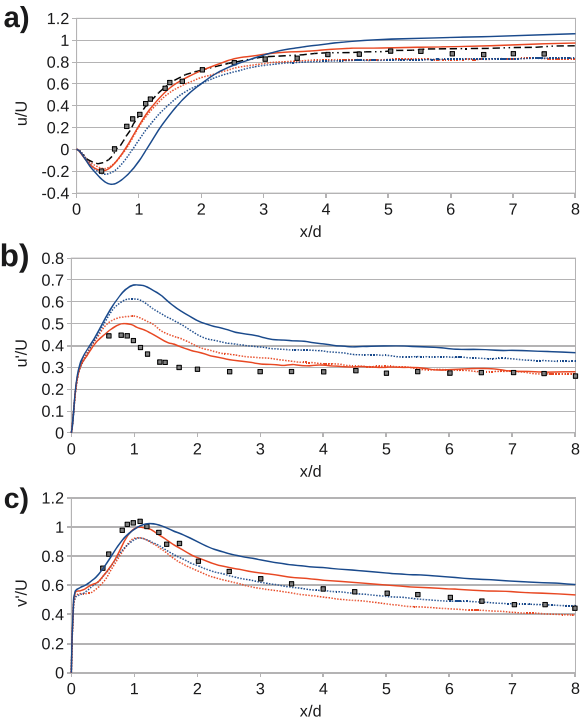

Fig. 10 Time averaged velocity profiles along the cylinder wake line. *Solid lines*: present work (turbulent inlet); *dotted line*: present work (uniform inlet); *broken line*: LES by Kim et al. [27]; *symbols*: experimental by Nakagawa et al. [26]. *Blue*: $z = 1d$; *orange*: $z = 3d$. **a**) mean stream-wise velocity; **b**) RMS stream-wise velocity fluctuations; **c**) RMS crosss-tream velocity fluctuations

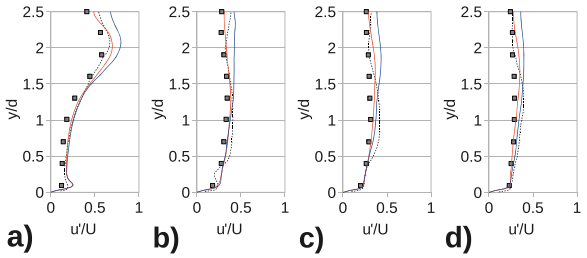

Fig. 11 RMS stream-wise velocity fluctuation profiles at four cross-stream locations. *Solid lines*: present work (turbulent inlet); *dotted line*: LES by Kim et al. [27]; *symbols*: experimental by Nakagawa et al. [26]. *Blue*: $z = 1d$; *orange*: $z = 3d$. **a)** $x = 1d$; **b)** $x = 3.5d$; **c)** $x = 6d$; **d)** $x = 8.5d$; measured from back the face of cylinder

is almost identical to that of the inlet region of the channel flow domain, which further validates the consistency of the present inlet boundary condition.

The time averaged flow field was compared against previous studies. Figure 10 shows the mean velocity profiles along the wake line of the cylinder. Note that Kim et al. [27] did not present the turbulent statistics along the wake line. From Fig. 10a, it can be seen that with the turbulent inlet condition, the $z = 1d$ cases significantly over-predicted the recirculation length and maximum negative velocity, while the $z = 3d$ cases showed good correspondence with both experimental [26] and numerical [27] results. The over-prediction of the recirculation zone in the $z = 1d$ cases also affected the stream-wise velocity fluctuations, where it displayed a peak value that is higher by more than 50 % of the experimental value (Fig. 10b). In comparison, the $z = 3d$ case showed only a 10 % over-prediction in the peak value. From Fig. 10c, it was found that both cases showed good agreement with the experimental results. The results obtained from the uniform inlet condition (dotted lines) also showed good correspondence with experimental results. In general, the uniform inlet cases produced better results compared to the $z = 1d$ case with the turbulent inlet, but the $z = 3d$ case with the turbulent inlet showed highest correspondence with literature.

The cross-sectional velocity profiles at $1d$, $3.5d$, $6d$, and $8.5d$, were also compared in Figs. 11 and 12. For clarity, only the turbulent inlet condition cases are shown. It was found that the $z = 1d$ case consistently over-predicted the turbulent fluctuations, while the $z = 3d$ case displayed excellent correspondence with experimental results. This showed that the

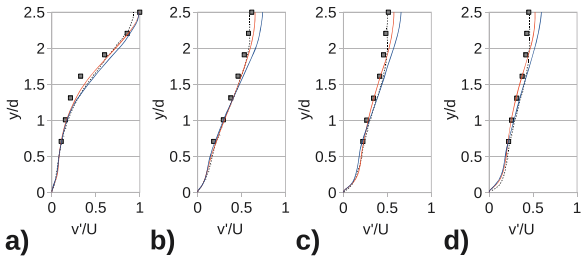

Fig. 12 RMS cross-stream velocity fluctuation profiles at four cross-stream locations. *Solid lines*: present work (turbulent inlet); *dotted line*: LES by Kim et al. [27]; *symbols*: experimental by Nakagawa et al. [26]. *Blue*: $z = 1d$; *orange*: $z = 3d$. **a)** $x = 1d$; **b)** $x = 3.5d$; **c)** $x = 6d$; **d)** $x = 8.5d$; measured from back the face of cylinder

Table 3 Force coefficients and Strouhal numbers for flow over square cylinder in a channel at $Re_d = 3000$

	$\overline{C}_D$	C'_D	C'_L	St
Present				
$z = 1d$, Turbulent inlet	3.40	0.50	1.97	0.136
$z = 3d$, Turbulent inlet	3.10	0.29	1.96	0.133
$z = 1d$, Uniform inlet	2.67	0.48	1.75	0.120
$z = 3d$, Uniform inlet	2.72	0.39	1.87	0.118
Kim et al. [27] (LES)	2.76	0.49	2.06	0.124
Nakagawa et al. [26] (exp.)	–	–	–	0.13

turbulent inlet boundary condition may have detrimental effects if the spanwise dimension is not large enough. Comparison with the LES results from Kim et al. [27] indicated that for both the stream-wise and cross-stream fluctuations, the present results with $z = 3d$ showed better agreement in the near wake region ($x = 1d$).

The force coefficients and the Strouhal numbers were compared against literature (Table 3). The table shows that the $z = 3d$ turbulent inlet case was in excellent agreement with experiment [26] in terms of Strouhal number. The $z = 1d$ turbulent inlet case in comparison, showed a reasonable lift prediction, but overpredicted the drag forces and Strouhal number. The $z = 1d$ uniform inlet case showed the closest correspondence with the LES results by Kim et al. [27]. This was not surprising, considering they used the same domain dimensions. It was also observed that for the uniform inlet case, increasing the spanwise dimension to $3d$ only resulted in minor changes in the results. Furthermore, considering that Kim et al. enforced the inlet condition as a mean velocity profile with fluctuations created by random numbers, it can be inferred that the Strouhal number is significantly affected by the presence of physically coherent turbulent structures at the inlet.

5 Conclusions

In this study, the LBM was implemented on the GPU, and the effectiveness of using LBM to simulate incompressible turbulent flow over bluff bodies was investigated. Preliminary simulations in the laminar regime showed that the LBM code is accurately capturing the flow physics involved with flows over bluff bodies, and that the multi-GPU code allows for speedups of over 150 times compared to a serial run by using two GPUs in parallel. The results from the fully developed turbulent channel flow simulation were in good agreement with DNS results, corroborating that the LBM is a viable tool for conducting LES. Finally, from the simulation of the turbulent flow over a square cylinder in a channel showed that having an auxiliary flow domain consisting of a periodic channel flow allows for a physically realistic inflow condition. Comparisons of the current results with literature suggested that accounting for the turbulent structures at the inlet location can significantly improve the results.

Acknowledgments The first author is grateful to the Natural Sciences and Engineering Research Council of Canada (NSERC) and Bombardier Aerospace for the financial support towards his MASc degree through the NSERC Industrial Postgraduate Scholarships (IPS) Program. The computations involved in this study were made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET).

References

1. Banerjee, S., Premnath, K., Pattison, M.: Turbulence simulations using the generalized lattice Boltzmann equation on massively parallel architectures. In: APCOM'07 in conjunction with EPMESC XI. Citeseer, Kyoto (2007)
2. Krafczyk, M., Tolke, J., Luo, L.: Large-eddy simulations with a multiple-relaxation-time LBE model. Int. J. Mod. Phys. B **17**(1), 33–40 (2003)
3. Premnath, K.N., Pattison, M.J., Banerjee, S.: Generalized lattice Boltzmann equation with forcing term for computation of wall-bounded turbulent flows. Phys. Rev. E **79**(2), 026703 (2009)
4. Stiebler, M., Krafczyk, M., Freudiger, S., Geier, M.: Lattice Boltzmann large eddy simulation of subcritical flows around a sphere on non-uniform grids. Comput. Math. Appl. **61**(12), 3475–3484 (2011)
5. Touil, H., Ricot, D., Lévêque, E.: Direct and large-eddy simulation of turbulent flows on composite multi-resolution grids by the lattice Boltzmann method. Hyper Articles en Ligne (2013)
6. Yu, H., Luo, L., Girimaji, S.: LES of turbulent square jet flow using an MRT lattice Boltzmann model. Comput. Fluids **35**(8), 957–965 (2006)
7. Sagaut, P.: Toward advanced subgrid models for Lattice-Boltzmann-based Large-eddy simulation: theoretical formulations. Comput. Math. Appl. **59**(7), 2194–2199 (2010)
8. Kang, S.K., Hassan, Y.A.: The effect of lattice models within the lattice Boltzmann method in the simulation of wall-bounded turbulent flows. J. Comput. Phys. **232**(1), 100–117 (2013)
9. Zhuo, C., Zhong, C.: LES-based filter-matrix lattice Boltzmann model for simulating turbulent natural convection in a square cavity. Int. J. Heat Fluid Flow **42**, 10–22 (2013)
10. Chikatamarla, S., Karlin, I.: Entropic lattice boltzmann method for turbulent flow simulations: Boundary conditions. Phys. A: Stat. Mech. Appl. **392**(9), 1925–1930 (2013)
11. Malaspinas, O., Sagaut, P.: Wall model for large-eddy simulation based on the lattice Boltzmann method. J. Comput. Phys. **275**, 25–40 (2014)
12. Hou, S., Sterling, J., Chen, S., Doolen, G.: A lattice Boltzmann subgrid model for high Reynolds number flows. Pattern Form. Lattice Gas Autom. **6**, 151–166 (1994)
13. Smagorinsky, J.: General circulation experiments with the primitive equations. Mon. Weather Rev. **91**(3), 99–164 (1963)
14. Chen, S., Doolen, G.: Lattice Boltzmann method for fluid flows. Ann. Rev. Fluid Mech. **30**(1), 329–364 (1998)
15. Frisch, U., Hasslacher, B., Pomeau, Y.: Lattice-gas automata for the Navier-Stokes equation. Phys. Rev. Lett. **56**(14), 1505–1508 (1986)
16. Lätt, J., et al.: Hydrodynamic limit of lattice Boltzmann equations. Ph.D. Thesis. Université de Genève (2007)
17. Zhao, Y.: Modeling natural phenomena with lattice Boltzmann method. Ph.D. Thesis. State University of New York at Stony Brook (2006)
18. Pope, S.: Turbulent flows. Cambridge University Press, Cambridge (2000)
19. Schäfer, M., Turek, S., Durst, F., Krause, E., Rannacher, R.: Benchmark computations of laminar flow around a cylinder. Notes on Numer. Fluid Mech. **52**, 547–566 (1996)
20. Koda, Y.: Lattice Boltzmann method for simulating turbulent flows. Master's Thesis. University of Waterloo (2013)
21. NVIDIA: Tesla M-Class GPU Computing Modules. http://www.nvidia.com/docs/IO/105880/DS-Tesla-M-Class-Aug11.pdf (2011)
22. Mussa, A., Asinari, P., Luo, L.: Lattice Boltzmann simulations of 2D laminar flows past two tandem cylinders. J. Comput. Phys. **228**(4), 983–999 (2009)
23. Lallemand, P., Luo, L.: Lattice Boltzmann method for moving boundaries. J. Comput. Phys. **184**(2), 406–421 (2003)
24. Mei, R., Luo, L., Shyy, W.: An accurate curved boundary treatment in the lattice Boltzmann method. J. Comput. Phys. **155**(2), 307–330 (1999)
25. Moser, R., Kim, J., Mansour, N.: Direct numerical simulation of turbulent channel flow up to Re= 590. Phys. Fluids **11**, 943 (1999)
26. Nakagawa, S., Nitta, K., Senda, M.: An experimental study on unsteady turbulent near wake of a rectangular cylinder in channel flow. Exp. Fluids **27**(3), 284–294 (1999)
27. Kim, D., Yang, K., Senda, M.: Large eddy simulation of turbulent flow past a square cylinder confined in a channel. Comput. Fluids **33**(1), 81–96 (2004)
28. Yang, K., Ferziger, J.: Large-eddy simulation of turbulent obstacle flow using a dynamic subgrid-scale model. AIAA J. **31**(8), 1406–1413 (1993)