HAL
open science

Towards aeraulic simulations at urban scale using the lattice Boltzmann method

Christian Obrecht, Frédéric Kuznik, Lucie Merlier, Jean-Jacques Roux, Bernard Tourancheau

To cite this version:

Christian Obrecht, Frédéric Kuznik, Lucie Merlier, Jean-Jacques Roux, Bernard Tourancheau. Towards aeraulic simulations at urban scale using the lattice Boltzmann method. Environmental Fluid Mechanics, 2015, 15 (4), pp.753-770. 10.1007/s10652-014-9381-0 . hal-01059223

HAL Id: hal-01059223

https://hal.science/hal-01059223v1

Submitted on 29 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards Aeraulic Simulations at Urban Scale Using the Lattice Boltzmann Method

Christian Obrecht · Frédéric Kuznik · Lucie Merlier · Jean-Jacques Roux · Bernard Tourancheau

Published in Environmental Fluid Mechanics (2014), DOI 10.1007/s10652-014-9381-0

Abstract The lattice Boltzmann method (LBM) is an innovative approach in computational fluid dynamics (CFD). Due to the underlying lattice structure, the LBM is inherently parallel and therefore well suited for high performance computing. Its application to outdoor aeraulic studies is promising, e.g. applied on complex urban configurations, as an alternative approach to the commonplace Reynolds-averaged Navier–Stokes (RANS) and large eddy simulation (LES) methods based on the Navier–Stokes equations. Emerging many-core devices, such as graphic processing units (GPUs), nowadays make possible to run very large scale simulations on rather inexpensive hardware. In this paper, we present simulation results obtained using our multi-GPU LBM solver. For validation purpose, we study the flow around a wall-mounted cube and show agreement with previously published experimental results. Furthermore, we discuss larger scale flow simulations involving nine cubes which demonstrate the practicability of CFD simulations in building external aeraulics.

Keywords Computational fluid dynamics · Lattice Boltzmann method · Urban flow · Large eddy simulation · High-performance computing

1 Introduction

Computational fluid dynamics (CFD) appears more and more suited and relevant to investigate aeraulic issues from the building-scale to the meso-scale [37]. Actually, three approaches are at hand when focusing on external air flows: full-scale measurements, wind tunnel simulations and CFD modelling. Each one has its own advantages and drawbacks, which make them highly complementary, yet more or less suited to specific investigations. Due to its better versatility, CFD modelling is well-suited for the investigation of various aspects linked with local urban micro-climates, such as urban wind patterns or building energy loads [2, 18]. However, the accuracy of numerical outputs greatly depends on the physical

C. Obrecht (✉) · F. Kuznik · L. Merlier · J.-J. Roux
Université de Lyon, CNRS, INSA-Lyon, CETHIL UMR5008, 69621 Villeurbanne Cedex, France
E-mail: christian.obrecht@insa-lyon.fr

B. Tourancheau
Université de Grenoble, UJF-Grenoble, LIG UMR5217, 38041 Grenoble Cedex 9, France

Nomenclature

C_S	Smagorinsky constant	
c_s	speed of sound	$[\mathrm{m}\cdot\mathrm{s}^{-1}]$
δt	time step	[s]
δx	mesh size	[m]
e	energy	[J]
$\mathbf{F}$	external force	
f	distribution function	
H	height of the cube	[m]
h	height of the channel	[m]
$\mathbf{j}$	fluid momentum	$[\mathrm{kg}\cdot\mathrm{m}^{-2}\cdot\mathrm{s}^{-1}]$
m	mass of the particle	[kg]
P	strain rate tensor	
p_{xx}	related to the strain rate tensor	
p	mean pressure	[Pa]
$\mathbf{q}$	heat flux	$[\mathrm{W}\cdot\mathrm{m}^{-2}]$
Re	Reynolds number	$[-]$
r	averaged pressure relative variation	$[-]$
s	relaxation rate	
T_0	turn-over time	[s]
u_0	maximum inlet velocity	$[\mathrm{m}\cdot\mathrm{s}^{-1}]$
$\mathbf{u}$	fluid velocity	$[\mathrm{m}\cdot\mathrm{s}^{-1}]$
x, y, z	position	[m]
Greek letters		
Ω	collision operator	
ξ_α	particle velocity	$[\mathrm{m}\cdot\mathrm{s}^{-1}]$
ρ	fluid density	$[\mathrm{kg}\cdot\mathrm{m}^{-3}]$
ε	energy square	$[\mathrm{J}^2]$
ν	kinematic viscosity	$[\mathrm{m}^2\cdot\mathrm{s}^{-1}]$
τ	relaxation time	[s]
Subscripts		
0	molecular or inflow	
α	associated to the particle velocities ξ_α	
B	bulk	
t	turbulent	
x, y, z	relative to direction	
∞	free-stream	

and numerical hypotheses. These modelling assumptions can directly impact simulation results and their usability in further studies.

In comparison to the commonplace Reynolds-averaged Navier–Stokes (RANS) models used in wind engineering, studies performed with the large eddy simulation (LES) method provide more relevant results. It should be pointed out, in particular, that simplified approaches are not satisfactory in terms of accuracy when modelling energy efficient buildings [3]. As a matter of fact, the representation of the turbulent fluctuations as well as the resolution of the largest turbulence scales can significantly improve the computed fields providing better fit to experimental results. However, the more complex a model is, the higher the computational effort will be. Consequently, the challenge is yet to enhance the quality of CFD models while controlling the computational costs. Indeed, recent advances, in both

computational fluid dynamics and high performance computing make possible to consider the actual use of explicit flow simulations in building models.

In this contribution, we shall present simulation results obtained using the lattice Boltzmann method (LBM). Being based on a mesoscopic point of view, this novel CFD approach has numerous advantages over classic macroscopic methods such as the solving of the Navier–Stokes equations [5]. Among other benefits, it is worth mentioning the high numerical stability, the ability to deal with complex geometries and the straightforwardness of various physical couplings and turbulence modelling [6]. Although parallel implementations of the LBM may be rather efficient, performing large scale simulations on mainstream architectures still requires the use of expensive clusters [28]. The present simulations were carried out using several graphics processing units (GPUs) in parallel within a single server. Performance afforded by such hardware configuration is comparable to the one obtained using large clusters at a fairly lower cost. Their use in LBM based urban flow simulations dates back to the first attempts to perform general purpose computations on these domain-specific processors [9] and is still under active investigation [26]. The present work is an improved and extended version of a contribution to the proceedings of the Building Simulation 2011 conference [23].

The remainder of the paper is organised as follows. The second section gives a brief review of some previous works showing the ability of CFD to provide a better understanding of urban air flows. It addresses current practices in CFD modelling for outdoor issues and highlights some sore points which are directly linked to modelling assumptions. The third section is a summary of the LBM, presenting the specific model we retained, together with the subgrid-scale model we added in order to enable simulations at high Reynolds number. In the fourth section, we give a short description of state-of-the-art GPU implementations of LBM solvers and of our multi-GPU LBM framework. In the fifth section, for validation purposes, we present the simulation of a fully developed flow over a wall-mounted cube in a flat channel. The simulation results are compared to experimental data. Section 6 reports the simulation of the flow over nine identical wall-mounted cubes at high Reynolds numbers. The last section gives some concluding remarks.

2 Application of CFD to external aeraulic issues

With the present and future growth of the world's urban population, new challenges arise in terms of environment, energy, as well as human health and comfort. The development of urban areas, with the increase of population density, imply sensible micro-climate modifications. As a matter of fact, urban thermal and mass balances are changed, which might lead to the intensification of urban heat island as described in Ref. [25], as well as to more local phenomena, such as specific wind events which are of great importance for many urban applications.

Simple empirical relations might provide valuable indications and trends regarding the outdoor environment of a building. However, as already stated, to go further than preliminary investigations, it is necessary to resort to either on site measurement campaigns, wind tunnel simulations or CFD modelling (see [18] for a more in-depth discussion). The first method is the only one which includes all turbulence scales and takes into account all complex interactions occurring within the area of interest. It allows for a direct observation of reality.

Wind tunnel tests and CFD simulations are models, which are generally composed of still obstacles contained in a bounded domain. However, field experiments do not enable parametric studies to be carried out, and their boundary conditions are inherently undefined.

On the contrary, all boundary conditions of wind tunnels and CFD models are known. Several versions of a geometrical configuration can be investigated, which is useful during the design process. In practice, each one can advantageously be used to evaluate interactions between a building or an urban planning project and its immediate environment (see for instance [2]). Nevertheless, these approaches must address important methodological and modelling requirements, such as similarity criteria or accuracy assessment by model validation and verification.

Due to its versatility and relatively low cost, CFD is increasingly used in urban studies. Its application domain goes beyond the building boundaries, and is currently extended to the study of the outdoor environment. Actually, numerical simulations can be performed to investigate many applications linked with urban physics issues [3, 18], such as:

- urban pedestrian wind environment: wind comfort and wind danger;
- wind-driven rain on buildings;
- convective heat and mass transfer coefficients for external building walls which directly impact building energy loads;
- pollutant dispersion within the urban area from car, domestic and industrial exhausts, or toxic releases.

CFD can also provide information on natural ventilation potential and free cooling, which are interesting passive strategies for building thermal control.

The temporal and spatial resolution of CFD models provide the whole flow field data. As a consequence, CFD simulations can be linked with other numerical models, and supply useful boundary conditions. For example, it can provide mean pressure distribution and convective heat transfer coefficients at building outer walls for a building energy simulation. It can also give pollutant concentrations close to ventilation devices to allow for a better building design.

Nevertheless, because of turbulence phenomena, external air flows are usually complex, even for apparently simple configurations. This hardship comes in addition to the consequences of the CFD modelling assumptions. Consequently, most of research studies have been carried out on quite simple geometries, for which comprehensive experimental validation datasets are available. Those generic cases, typically cubes, ideal street canyons or array of cubes already show intricate flow recirculation phenomena, which would then be met in more complex configurations. Their understanding is in essence preliminary to further urban studies.

CFD studies for urban issues are often carried out using a steady RANS approach, based on time-averaged Navier–Stokes equations together with a closure scheme, typically two transport equations for the turbulent kinetic energy k and the dissipation ε. All turbulence scales are thus taken into account. Such models are widely used in industrial applications because of their computational efficiency. However, for urban wind investigations, even though this kind of approach usually provides satisfactory global mean wind fields, it is often deficient to predict accurately recirculation in building wakes, concentration fields, or wind driven rain impact.

The LES method is a time-dependent approach which resolves all the turbulence scales but the smallest ones. The smallest eddies are taken into account by a sub-grid scale model. The largest eddies, which contain most of the turbulent kinetic energy, are highly anisotropic and dependent of the geometric and climatic conditions, whereas the smallest one have a more universal behaviour [18]. Consequently, LES simulations usually reproduce empirical observations well.

To evaluate the accuracy of CFD approaches, several studies compare standard and modified RANS models and LES for urban wind issues [11, 35, 37]. They show the advantage of reproducing transient features and vortex shedding. Such comparisons were performed for variously complex geometries, allowing different levels of analysis and interpretation. Some of them are briefly reviewed hereafter, as representative examples of wind velocity fields or pollutant dispersion patterns investigations.

In Ref. [37], the authors compare several numerical approaches (mostly RANS but also LES for the simplest cases) on different geometric configurations, from an isolated sharp edged element to a complex urban configuration. Pedestrian wind environment is investigated. For the tested models, RANS predictions for strong wind zones are found to be fairly satisfactory, but thus fail to reproduce wind patterns in the wake region and close to buildings. This lack of accuracy is attributed to incapacity of RANS to reproduce the vortex shedding in building wakes, which can be achieved with an LES model. Furthermore, instantaneous wind velocity maximum could be computed by LES, allowing for the identification of dangerous gusts zones.

To analyse in details the mechanisms leading to the differences between RANS and LES, Tominaga and Stathopoulos performed a detailed comparative study of dispersion around a cube [35]. It is shown that the tested RANS model tends to underestimate the turbulence diffusion around the cube, and that the reattachment length downstream the obstacle is overpredicted. In terms of concentration fields, LES provides better results with regards to experimental measurements.

In Ref. [11], considering a more complex geometric configuration, the authors study the near field pollutant dispersion for a stack release in Montreal. Simulations using the standard RANS k–ε model and the LES approach are compared to a wind tunnel experiment. LES provides better results than RANS for the two wind directions tested. One of the main conclusions is that RANS output is highly dependent of the turbulent Schmidt number which is a priori unknown. Because of its time-dependent nature, LES does not need such input parameter. Moreover, the flow separation reproduction is found to be important to provide more accurate results, which is known to be often one of the weaknesses of RANS.

All of these studies emphasise the importance of modelling transient flow features, turbulence anisotropy, and turbulent dispersion of pollutant, to get accurate predictions. However, *they* also stress the high computational resources needed to carry out a mesoscopic LES simulation even on a simple case. As a consequence, significant advances in the efficiency of simulations are required for the LES to become more than a research tool, despite of its identified benefits. In the following sections we shall thus present an innovative approach to perform high-performance LES simulations, combining the computational power of GPUs with the efficiency of the lattice Boltzmann method.

3 Lattice Boltzmann method

3.1 Lattice Boltzmann equation

Although originating from the lattice gas automata theory [10], the lattice Boltzmann method is nowadays usually interpreted as a discrete version of the Boltzmann transport equation [16]:

$$\partial_t f + \xi \cdot \nabla_{\mathbf{x}} f + \frac{\mathbf{F}}{m} \cdot \nabla_{\xi} f = \Omega(f), \tag{1}$$

where $f(\mathbf{x},\xi,t)$ describes the evolution in time of the distribution of one particle in phase space, $\mathbf{F}$ is the external force field, m the mass of the particle, and Ω the collision operator. It is well known, using Chapman-Enskog expansion, that the incompressible Navier–Stokes equations can be recovered from the Boltzmann transport equation for small Knudsen numbers (see for instance [4]).

Discretisation occurs both in time, with constant time steps δt, and phase space, generally using a regular orthogonal grid of mesh size δx and a finite set of $N+1$ particle velocities ξ_α with $\xi_0 = \mathbf{0}$. The later is commonly a subset of the velocities linking any node of the grid to its nearest neighbours as the D3Q19 stencil we used for our simulations (see Fig. 1).

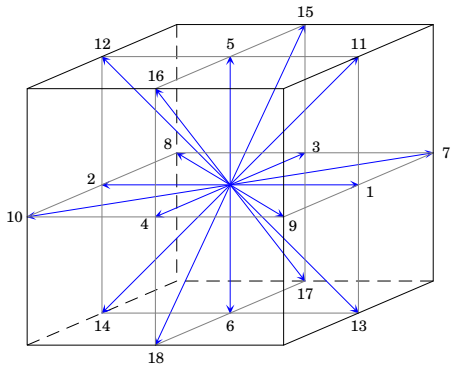


Fig. 1: The D3Q19 stencil — *The elements of this stencil link any node of the grid to 19 of its 27 nearest neighbours (including itself).*

The discrete counterpart of the distribution function f is a finite set of functions $f_\alpha(\mathbf{x},t)$ associated to the particle velocities ξ_α. Let us denote:

$$\left|a_\alpha\right\rangle = (a_0,\dots a_N)^\mathsf{T},$$

T being the transpose operator. The lattice Boltzmann equation (LBE) is expressed as:

$$\left|f_\alpha(\mathbf{x}+\delta t\xi_\alpha,t+\delta t)\right\rangle - \left|f_\alpha(\mathbf{x},t)\right\rangle = \Omega\left[\left|f_\alpha(\mathbf{x},t)\right\rangle\right]. \tag{2}$$

The mass density ρ and the velocity $\mathbf{u}$ of the fluid are given by:

$$\rho = \sum_\alpha f_\alpha\,, \qquad \mathbf{u} = \frac{1}{\rho}\sum_\alpha f_\alpha \xi_\alpha\,. \tag{3}$$

3.2 Multiple-relaxation-time LBM

The simplest (and most commonly used) way to express the collision operator is the LBGK approach [29], which uses the Bhatnagar-Gross-Krook approximation [1]. We instead chose

to use the multiple-relaxation-time (MRT) approach [7]. Although of higher computational cost, MRT was shown of better accuracy and numerical stability than LBGK [13].

In the MRT approach, collision is performed in moment space. The particle distribution is mapped to a set of moments $\{m_\alpha \,|\, i = 0, \ldots N\}$ by an orthogonal matrix M:

$$\big|f_\alpha(\mathbf{x},t)\big\rangle = \mathsf{M}^{-1}\big|m_\alpha(\mathbf{x},t)\big\rangle \tag{4}$$

where $\big|m(\mathbf{x},t)\big\rangle$ is the moment vector. For the D3Q19 stencil, the orthogonal matrix M can be found in appendix A of [8].

The moment vector corresponding to the D3Q19 vector is given by:

$$\big|m_\alpha(\mathbf{x},t)\big\rangle = \big(\rho, e, \varepsilon, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{zx}, m_x, m_y, m_z\big)^\mathsf{T} \tag{5}$$

where e is energy, ε is energy square, $\mathbf{j} = (j_x, j_y, j_z)$ is the momentum, $\mathbf{q} = (q_x, q_y, q_z)$ is the heat flux, p_{xx}, p_{xy}, p_{yz}, p_{zx}, p_{ww} are related to the strain rate tensor S, π_{xx}, π_{ww} are fourth-order moments and m_x, m_y, m_z are third-order moments with respect to the particle velocities. The mass density and the momentum are the conserved moments.

The LBE is thus expressed as:

$$\big|f_\alpha(\mathbf{x}+\delta t\xi_\alpha, t+\delta t)\big\rangle - \big|f_\alpha(\mathbf{x},t)\big\rangle = -\mathsf{M}^{-1}\Lambda\Big[\big|m_\alpha(\mathbf{x},t)\big\rangle - \big|m_\alpha^{(\mathrm{eq})}(\mathbf{x},t)\big\rangle\Big] \tag{6}$$

where Λ is a diagonal collision matrix and the $m_\alpha^{(\mathrm{eq})}$ are the equilibrium values of the moments. For the sake of isotropy, Λ obeys:

$$\Lambda = \mathrm{diag}(0, s_1, s_2, 0, s_4, 0, s_4, 0, s_4, s_9, s_{10}, s_9, s_{10}, s_{13}, s_{13}, s_{13}, s_{16}, s_{16}, s_{16}). \tag{7}$$

We additionally set $s_9 = s_{13}$. The relaxation rate s_9 is linked to the kinematic viscosity ν of the model by:

$$\frac{1}{s_9} = \frac{1}{c_s^2}\nu + \frac{1}{2}, \tag{8}$$

where the speed of sound c_s is set to:

$$c_s = \frac{1}{\sqrt{3}} \times \frac{\delta x}{\delta t}. \tag{9}$$

The other rates are set according to [13], i.e. $s_1 = 1.19$, $s_2 = s_{10} = 1.4$, $s_4 = 1.2$, and $s_{16} = 1.98$.

3.3 Large-eddy simulation

For large-eddy simulation (LES), the kinematic viscosity is $\nu = \nu_0 + \nu_t$ where ν_0 is the molecular viscosity and ν_t is the turbulent viscosity. In the Smagorinsky model [32], the turbulent viscosity is given by:

$$\nu_t = |\mathsf{S}|\,(C_S\delta x)^2, \quad |\mathsf{S}| = \sqrt{2\,\mathsf{S}:\mathsf{S}}, \tag{10}$$

where C_S is the Smagorinsky constant, which in the present work is set to $C_S = 0.1$. Adding eddy viscosity to the MRT model is achieved by replacing the relaxation rate s_9 with:

$$s_9^* = \frac{1}{\tau_0 + \tau_t} \tag{11}$$

where τ_0 and τ_t are the molecular and turbulent relaxation times:

$$\tau_0 = \frac{1}{c_s^2}\nu_0 + \frac{1}{2}, \quad \tau_t = \frac{1}{c_s^2}\nu_t. \tag{12}$$

Following [12], the second order moments obey:

$$\mathsf{P}_{ij} = \sum_\alpha \xi_{\alpha i}\xi_{\alpha j} f_\alpha = c_s^2 \rho \delta_{ij} + \rho u_i u_j - \mathsf{Q}_{ij} \tag{13}$$

with:

$$\mathsf{Q} = \frac{2c_s^2\rho}{s_9^*}\mathsf{S}. \tag{14}$$

Thus, the strain rate tensor may be computed from the moment vector. For the D3Q19 stencil, we obtain:

$$\mathsf{P}_{xx} = \frac{1}{57}(30\rho + e) + p_{xx}, \tag{15}$$

$$\mathsf{P}_{yy} = \frac{1}{57}(30\rho + e) + \frac{1}{2}(p_{ww} - p_{xx}), \tag{16}$$

$$\mathsf{P}_{zz} = \frac{1}{57}(30\rho + e) - \frac{1}{2}(p_{xx} + p_{ww}), \tag{17}$$

$$\mathsf{P}_{xy} = p_{xy}, \quad \mathsf{P}_{yz} = p_{yz}, \quad \mathsf{P}_{zx} = p_{zx}. \tag{18}$$

Finally, assuming that ν_t depends on S at current time, we have:

$$\tau_t = \frac{1}{2}\left(\sqrt{\tau_0^2 + 18\,|\mathsf{Q}|\,(C_S\delta x)^2} - \tau_0\right). \tag{19}$$

4 Multi-GPU solver

4.1 Algorithmic aspect

From an algorithmic standpoint, the LBE (Eq. 2) naturally breaks in two elementary steps:

$$\left|\widetilde{f}_\alpha(\mathbf{x},t)\right\rangle = \left|f_\alpha(\mathbf{x},t)\right\rangle + \boldsymbol{\Omega}\left[\left|f_\alpha(\mathbf{x},t)\right\rangle\right] \tag{20}$$

$$\left|f_\alpha(\mathbf{x} + \delta t\,\xi_\alpha, t + \delta t)\right\rangle = \left|\widetilde{f}_\alpha(\mathbf{x},t)\right\rangle \tag{21}$$

Equation 20 describes the *collision* step in which an updated particle distribution is computed. Equation 21 describes the *propagation* step in which the updated particle populations are transferred to the neighbouring nodes as outlined by Fig. 2 (in two dimensions for the sake of simplicity).

It is worth mentioning that in the first step, computations only involve informations local to each node. Moreover, in the second step, data transfers only require proper synchronisation with the nearest neighbours. As a matter of consequence, the LBM is fairly well suited for parallel implementations.

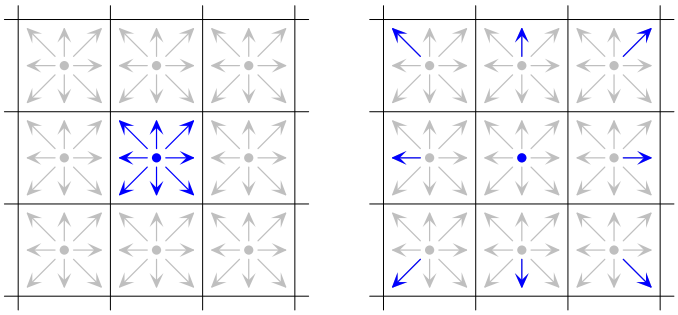


Fig. 2: Propagation step — *After collision (left-hand side), the updated particle densities are simply advected to the neighbouring nodes with respect of the chosen stencil.*

4.2 CUDA implementations

During the last decade, the computational power of GPUs has grown exponentially, reaching 1.35 Tflop/s single precision peak performance with the latest generation of NVIDIA processors [19]. Early attempts to implement the LBM on such hardware [9] were quite promising. With the introduction of the CUDA technology by NVIDIA in 2007, general purpose programming on GPUs became more practicable. Several successful CUDA implementations of the three-dimensional LBM [20, 34] were reported from then on.

On recent hardware, single GPU implementations are able to handle up to about 7.7×10^8 nodes per second, whereas multithreaded CPU implementations handle at most about 8.5×10^7 nodes per second using a single quad core processor [14]. It is also worth mentioning that performance of GPU implementations is communication bound [21], while performance of CPU implementations is computation bound. Thus, making use of a model of higher algorithmic complexity (e.g. MRT instead of LBGK) has in general little impact on performance.

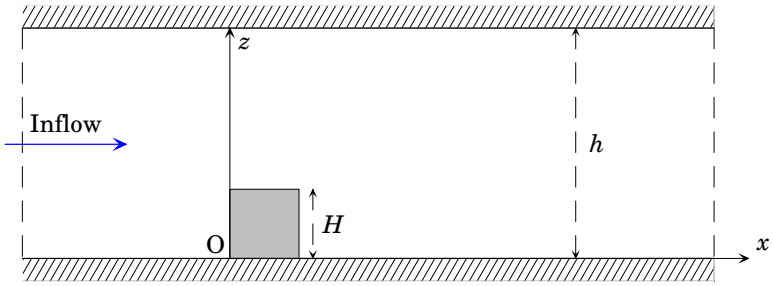
4.3 TheLMA framework

GPUs provide large computational power at fairly low cost. Yet, although growing more versatile at each generation, CUDA enabled GPUs still have numerous drawbacks. The CUDA tool chain for instance, due to hardware limitations, is unable to link several GPU binaries. In cases like LBM, this forbids the use of library oriented development techniques. The limited amount of on board memory may also be problematic. Using the latest computation devices, a single GPU implementation of the D3Q19 scheme may handle at most about 4.2×10^7 nodes in single precision.

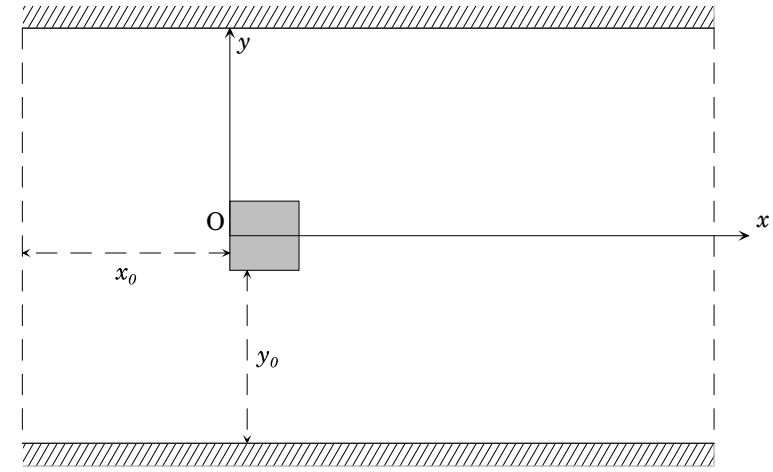
To address both issues, we created the TheLMA framework [33]. TheLMA stands for *Thermal LBM on Many-core Architectures*, thermal simulations being our main topic of interest. The TheLMA framework is designed to improve code reusability. Setting up a new simulation usually only requires minor code modifications. Moreover, TheLMA provides native multi-GPU support, for single-node servers [22] as well as for multi-node GPU clusters [24].

5 Flow around a single cube

In order to validate our MRT-LBM solver, we chose to simulate a fully developed flow around a wall-mounted cube in a channel. The simulation results are compared to experimental data from [17]. Figure 3 outlines the simulation setup.



(a) Side view



(b) Top view

Fig. 3: Simulation setup for a single cube

The channel is represented by a cavity containing $1\,024 \times 768 \times 192 \approx 1.51 \times 10^8$ nodes. Solid walls are simulated using half-way bounce-back boundary condition (see for instance [27]). A uniform velocity profile at the inlet is imposed by adding the corresponding non-equilibrium values to the distribution functions. The outlet condition is obtained by imposing null velocity gradient. The size of the cube is set to $H = 58\,\delta x$ in order to have $h/H \approx 3.3$ as in our reference, and the position of the cube is such as $x_0 = 4H$. It should be mentioned that, in order to save memory, y_0 is less in our setup than in the experimental one. This allows to improve the resolution of the obstacle, with little impact on the flow since we have $y_0 > 6H$.

In their work, the authors of [17] give the time-averaged streamwise velocity of the flow in the vertical symmetry plane, obtained through laser Doppler anemometry (LDA). The measurements were conducted at Reynolds number Re = 4 440 where:

$$\mathrm{Re} = \frac{u_B H}{\nu} \tag{22}$$

and u_B is the bulk velocity of the inflow. In our simulation, to ensure temporal convergence to a statistically steady state, we averaged the streamwise velocity over time from $50T_0$ to $200T_0$, where $T_0 = H/u_0$ is the turn-over time and u_0 is the maximum inlet velocity. The overall computation time was less than six hours using a Tyan B7015 server with eight Tesla C1060 computing devices.

Figures 5 and 6 show upstream and downstream normalised velocity profiles with respect of x/H for both simulation and measurements. Agreement of simulation data with experimental data is satisfactory since uncertainties on both position and value of measurements should be taken into account. Unfortunately, our reference does not provide such informations, nor does it give detailed data regarding the inlet velocity profile. Although not perfect, agreement with measurements is by far better than in previously published work (see for instance Fig. 9 of [36]).

The upstream velocity profiles (Fig. 5) show very similar variations. For $z/H \leq 0.13$, both the experiment and the simulation show that the streamwise velocity reaches a minimum near the obstacle. While agreeing on the magnitude of this minimum, they disagree on its location which is about $x/H = -0.35$ for the experiment and $x/H = -0.45$ for the simulation. Moreover, for $z/H \geq 0.3$, the simulation tends to overestimate the streamwise velocity with respect to experimental data when drawing away from the obstacle. We assume these discrepancies are mainly caused by the uncertainties regarding the approach flow. The downstream velocity profiles (Fig. 6) are in good agreement, both from a qualitative and a quantitative standpoint.

6 Flow around nine cubes

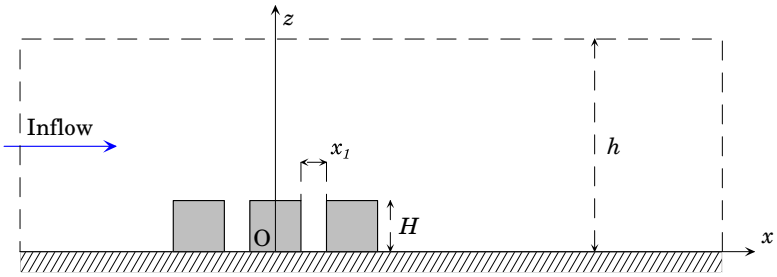
To illustrate the possible use of multi-GPU LBM solvers in building aeraulics, we chose to simulate the flow around nine identical wall-mounted cubes. Figure 4 outlines the simulation setup. Even if quite simple, this configuration implies complex turbulent flows.

For this simulation we used the same Tyan B7015 server, this time equipped with seven Tesla C2075 computing devices. The larger available device memory enabled us to use a simulation domain of size $1\,288 \times 768 \times 240$ which amounts approximatively to 2.38×10^8 nodes. The size of the cubes is set to $H = 48\delta x$, and the position is such that $x_0 = 3H$ and $x_1 = H/2$. Thus, we have $y_0 = 6H$ and $h = 5H$. We impose logarithmic velocity profile for the inflow. In this configuration, we define the Reynolds number as:

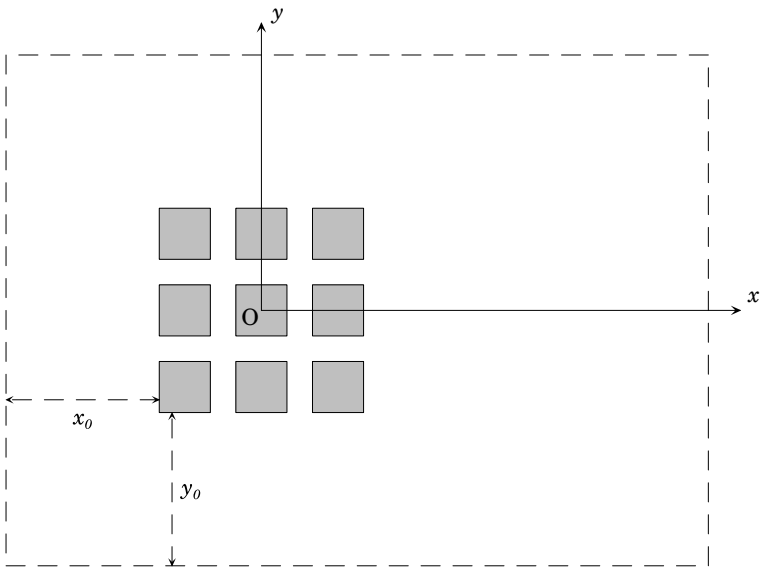
$$\mathrm{Re} = \frac{u_1 H}{\nu} \tag{23}$$

where u_1 is the inflow velocity at obstacle height. Furthermore, to reduce the impact of lateral faces on the flow, we apply the same boundary condition as for the outlet, i.e. null velocity gradient in the direction normal to the face.

We chose to run simulations at $\mathrm{Re}_1 = 4 \times 10^4$ and $\mathrm{Re}_2 = 10^6$. Smagorinsky subgrid-scale models were reported satisfactory in similar situations, for Reynolds numbers up to at least Re_1 with LBM flow solvers [12], and at least Re_2 for Navier-Stockes solvers [31].



(a) Side view



(b) Top view

Fig. 4: Simulation setup for nine cubes

Although the LBM part in our implementation differs from the former, we may be confident in the results at $\mathrm{Re} = \mathrm{Re}_1$. The simulation at $\mathrm{Re} = \mathrm{Re}_2$ is more relevant at building scale, however the results should be considered with greater care.

As in the single cube simulation, we averaged density and velocity over time from $50T_0$ to $200T_0$, the turn-over time being set to $T_0 = 4H/u_0$. The overall computation time was about 14 h 56 min for 768 000 time steps. The corresponding performance is approximatively 3.4×10^9 node updates per second, which is about a $40\times$ speedup over optimised multi-threaded CPU implementations.

Figures 7 and 8 display the pressure relative variation r and velocity streamlines of the time-averaged flow in the horizontal plane at $z = H/2$ and in the vertical streamwise plane at $y = 0$, with r defined as:

$$r = \frac{p - p_0}{p_0} \tag{24}$$

where p_0 is the pressure near the inflow.

Both simulations lead to very similar mean flow patterns. Considering the gaps upstream and downstream the central cube, the horizontal sections display two weakly linked counter-rotating cells, whereas the vertical stream-wise sections display a well-delimited cell above a more quiet region. A closer inspection of these zones, using plane sections at various locations, shows that these patterns correspond in fact to horse-shoe flow structures. Similar phenomena are known to occur in the wake of a single wall-mounted cube [15]. The obtained flow features seem compatible with the well-known street canyon phenomenology.

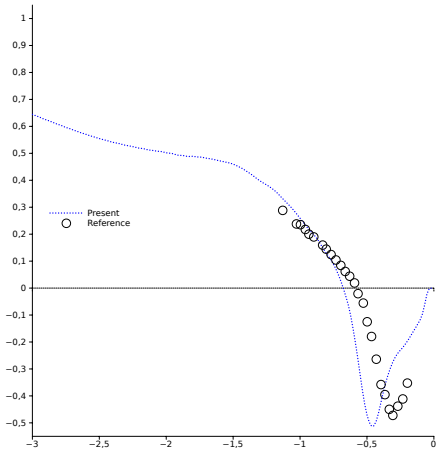
The most striking point revealed by the horizontal sections is the fact that the flow patterns are asymmetric and almost enantiomorphic, i.e. the mirror image of one pattern with respect to the median stream-wise plane closely coincides with the other one. We infer that a symmetric mean flow pattern is either inexistent or metastable. Because of the turbulent nature of the flow arriving at the obstacles, the simulation leads to a chiral ground-state with either one of the two possible orientations.

If we assume $p_0 = p_\infty$, where p_∞ is the free-stream pressure, then r equals to the pressure coefficient. The obtained values are within the range of coefficients used in practice and seem therefore reasonable.

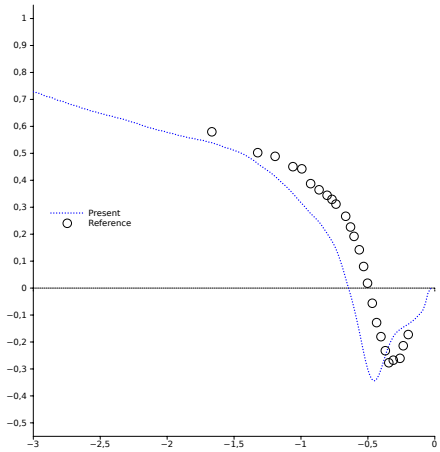
7 Conclusions

In the present work, we provide building scale flow simulation results obtained using our multi-GPU implementation of the LBM. We show that the required computation times remain below reasonable limits, even for multi-obstacle cases. Since efficient LES can be performed to accurately study external aeraulics, we believe this contribution is a significant step towards the use of effective CFD simulations in building models. To a greater extent, its suitability for further investigations (of wind patterns, pollutant dispersion process, or heat transfers at building façade) on complex urban geometrical configurations is very promising.

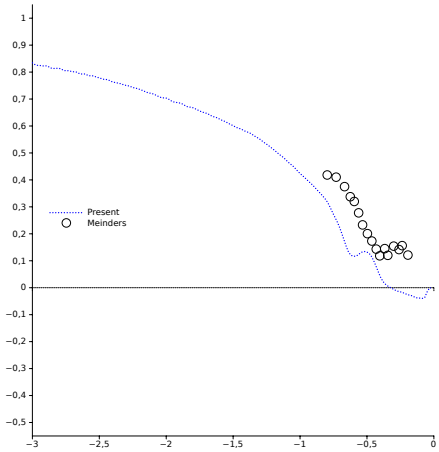
Several improvements to our approach, regarding both performance and accuracy are within reach. From a physical standpoint, the use of more elaborate subgrid-scale models than the standard Smagorinsky model we implemented would be desirable. On-going research founded on the same mesoscopic point of view as the LBM might provide advances on this issue [30]. From a computational standpoint, porting grid refinement techniques to the GPU would be of highest practical interest and we plan to add such a feature to our framework in near future.



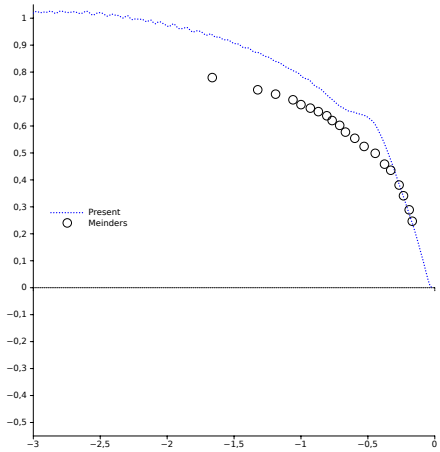
(a) $z/H = 0.07$



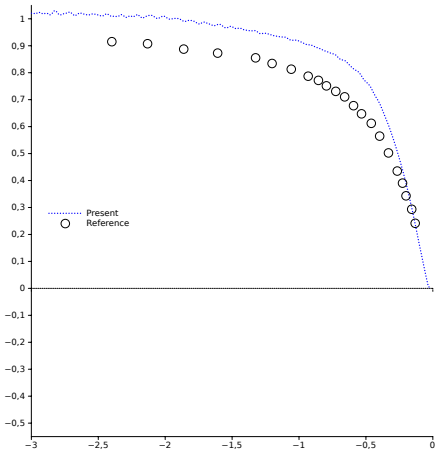
(b) $z/H = 0.1$



(c) $z/H = 0.13$

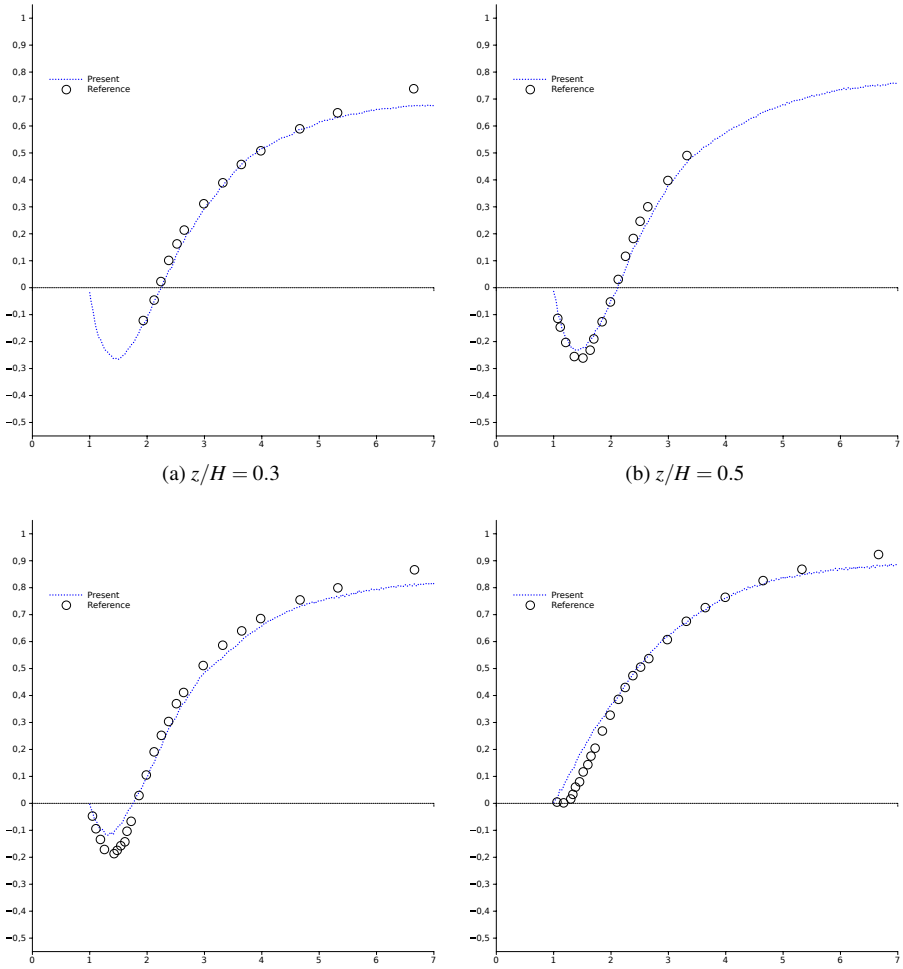


(d) $z/H = 0.3$



(e) $z/H = 0.5$

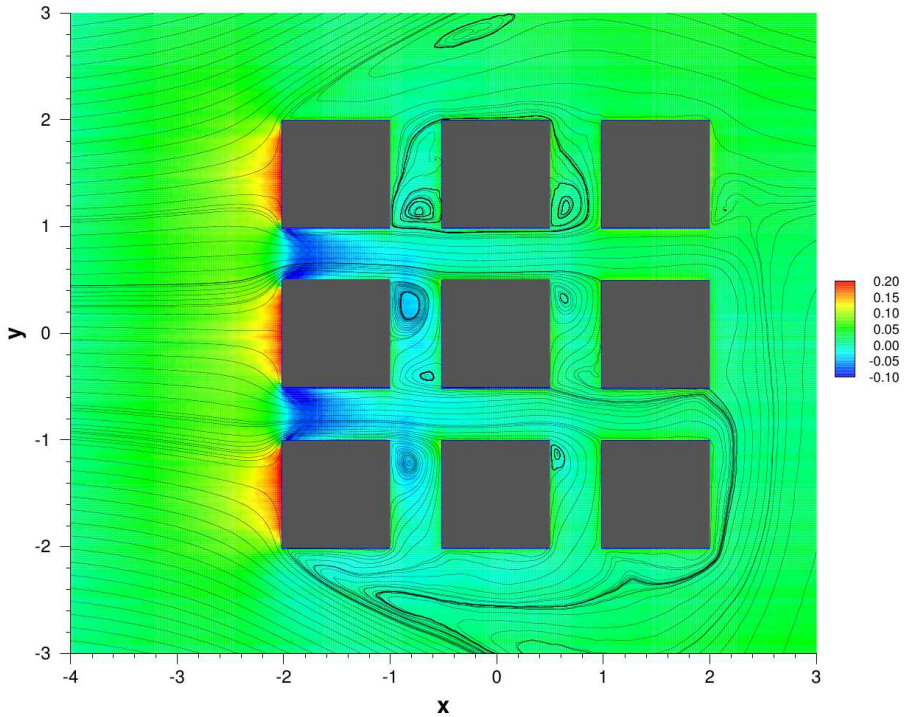
Fig. 5: Upstream normalised velocity profiles with respect of x/H — *Comparison between simulation and experimental reference data by Meinders* et al. *[17].*



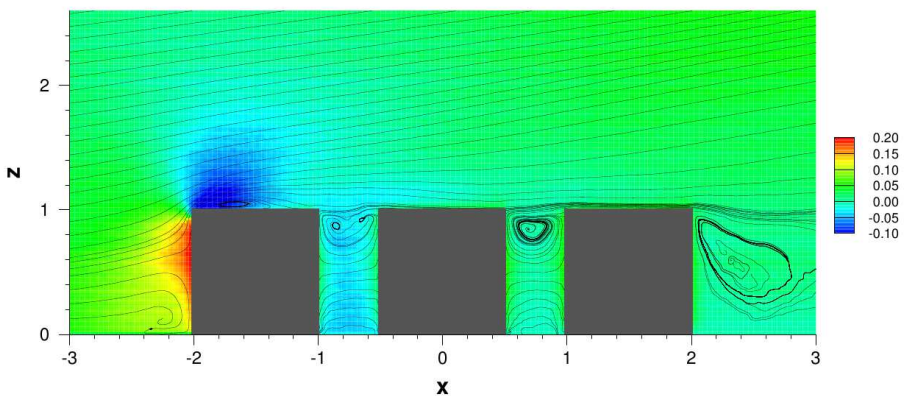
(a) $z/H = 0.3$ (b) $z/H = 0.5$

(c) $z/H = 0.7$ (d) $z/H = 0.9$

Fig. 6: Downstream normalised velocity profiles with respect of x/H — *Comparison between simulation and experimental reference data by Meinders* et al. *[17].*

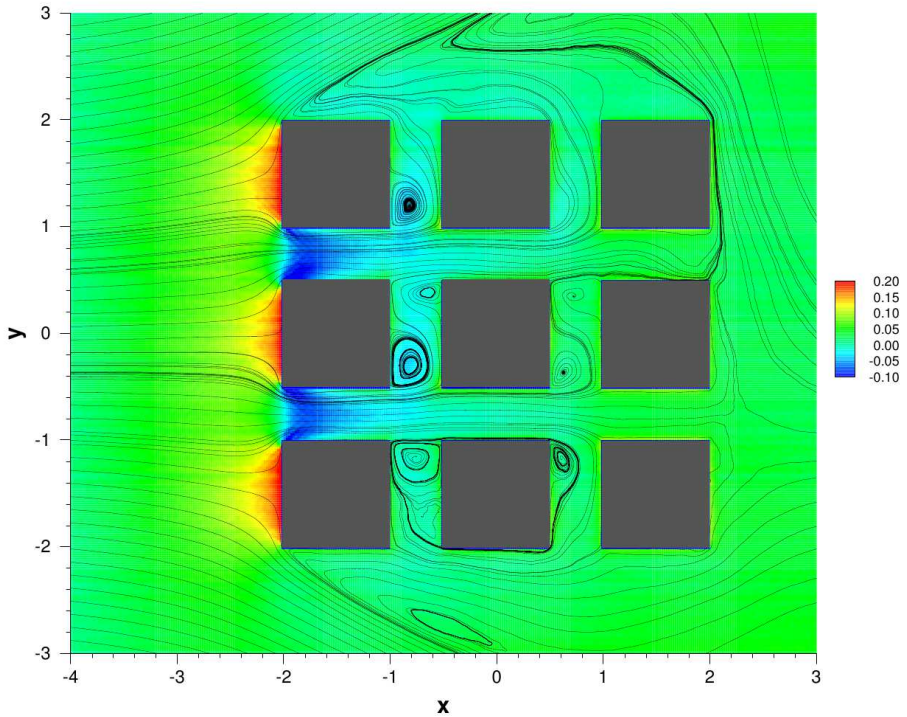


(a) Horizontal plane at $z = H/2$

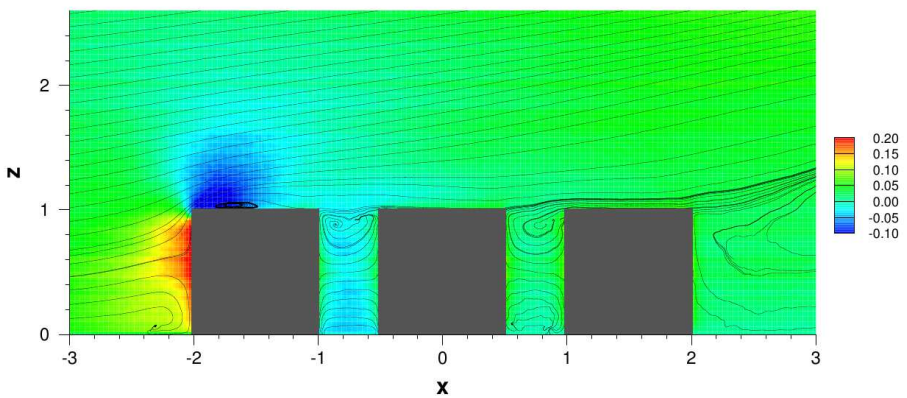


(b) Vertical stream-wise plane at $y = 0$

Fig. 7: Streamlines and pressure relative variation of the time-averaged flow for $\mathrm{Re} = 4 \times 10^4$



(a) Horizontal plane at $z = H/2$



(b) Vertical stream-wise plane at $y = 0$

Fig. 8: Streamlines and pressure relative variation of the time-averaged flow for $\mathrm{Re} = 10^6$

References

1. Bhatnagar PL, Gross EP, Krook M (1954) A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. Physical Review 94(3):511–525
2. Blocken B, Persoon J (2009) Pedestrian wind comfort around a large football stadium in an urban environment: CFD simulation, validation and application of the new Dutch wind nuisance standard. Journal of Wind Engineering and Industrial Aerodynamics 97(5):255–270
3. Blocken B, Stathopoulos T, Carmeliet J, Hensen J (2011) Application of computational fluid dynamics in building performance simulation for the outdoor environment: an overview. Journal of Building Performance Simulation 4(2):157–184
4. Cercignani C (1987) The Boltzmann equation ans its applications. Springer-Verlag
5. Chen S, Doolen GD (1998) Lattice boltzmann method for fluid flows. Annual review of fluid mechanics 30(1):329–364
6. Crouse B, Krafczyk M, Kühner S, Rank E, Van Treeck C (2002) Indoor air flow analysis based on lattice Boltzmann methods. Energy and buildings 34(9):941–949
7. d'Humières D (1994) Generalized lattice-Boltzmann equations. Rarefied gas dynamics – Theory and simulations pp 450–458
8. d'Humières D, Ginzburg I, Krafczyk M, Lallemand P, Luo L (2002) Multiple-relaxation-time lattice Boltzmann models in three dimensions. Philosophical Transactions: Mathematical, Physical and Engineering Sciences pp 437–451
9. Fan Z, Qiu F, Kaufman A, Yoakum-Stover S (2004) GPU cluster for high performance computing. In: Proceedings of the 2004 ACM/IEEE Conference on Supercomputing, IEEE Computer Society, p 47
10. Frisch U, Hasslacher B, Pomeau Y (1986) Lattice-Gas Automata for the Navier-Stokes Equation. Physical Review Letters 56(14):1505–1508
11. Gousseau P, Blocken B, Stathopoulos T, van Heijst G (2011) CFD simulation of near-field pollutant dispersion on a high-resolution grid: A case study by LES and RANS for a building group in downtown Montreal. Atmospheric Environment 45(2):428–438
12. Krafczyk M, Tölke J, Luo L (2003) Large-eddy simulations with a multiple-relaxation-time LBE model. International Journal of Modern Physics B 17(1):33–40
13. Lallemand P, Luo L (2000) Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. Physical Review E 61(6):6546
14. Lee V, Kim C, Chhugani J, Deisher M, Kim D, Nguyen A, Satish N, Smelyanskiy M, Chennupaty S, Hammarlund P (2010) Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU. In: ACM SIGARCH Computer Architecture News, ACM, vol 38, pp 451–460
15. Martinuzzi R, Tropea C (1993) The flow around surface-mounted, prismatic obstacles placed in a fully developed channel flow. ASME Journal of Fluids Engineering 115:85–85
16. McNamara GR, Zanetti G (1988) Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. Physical Review Letters 61:2332–2335
17. Meinders E, Hanjalic K, Martinuzzi R (1999) Experimental study of the local convection heat transfer from a wall-mounted cube in turbulent channel flow. Journal of Heat Transfer 121:564
18. Moonen P, Defraeye T, Dorer V, Blocken B, Carmeliet J (2012) Urban Physics: Effect of the micro-climate on comfort, health and energy demand. Frontiers of Architectural Research 1(3):197–228

19. NVIDIA (2010) Compute Unified Device Architecture Programming Guide version 3.2
20. Obrecht C, Kuznik F, Tourancheau B, Roux JJ (2011) A New Approach to the Lattice Boltzmann Method for Graphics Processing Units. Computers & Mathematics with Applications 12(61):3628–3638
21. Obrecht C, Kuznik F, Tourancheau B, Roux JJ (2011) Global Memory Access Modelling for Efficient Implementation of the Lattice Boltzmann Method on Graphics Processing Units. In: Lecture Notes in Computer Science 6449, High Performance Computing for Computational Science – VECPAR 2010 Revised Selected Papers, Springer, pp 151–161
22. Obrecht C, Kuznik F, Tourancheau B, Roux JJ (2011) The TheLMA project: Multi-GPU Implementation of the Lattice Boltzmann Method. International Journal of High Performance Computing Applications 25(3):295–303
23. Obrecht C, Kuznik F, Tourancheau B, Roux JJ (2011) Towards Urban-Scale Flow Simulations Using the Lattice Boltzmann Method. Proceedings of the BS2011 Conference
24. Obrecht C, Kuznik F, Tourancheau B, Roux JJ (2013) Scalable Lattice Boltzmann Solvers for CUDA GPU Clusters. Parallel Computing 39(6-7):259–270
25. Oke T (1987) Boundary Layer Climates. Routledge
26. Onodera N, Aoki T, Shimokawabe T, Kobayashi H (2013) Large-scale LES Wind Simulation using Lattice Boltzmann Method for a 10 km$\times$10 km Area in Metropolitan Tokyo. Tech. rep.
27. Pan C, Luo L, Miller C (2006) An evaluation of lattice Boltzmann schemes for porous medium flow simulation. Computers & fluids 35(8-9):898–909
28. Pohl T, Deserno F, Thurey N, Rude U, Lammers P, Wellein G, Zeiser T (2004) Performance evaluation of parallel large-scale lattice boltzmann applications on three supercomputing architectures. In: Proceedings of the 2004 ACM/IEEE conference on Supercomputing, IEEE Computer Society, p 21
29. Qian YH, d'Humières D, Lallemand P (1992) Lattice BGK models for Navier-Stokes equation. Europhysics Letters 17(6):479–484
30. Sagaut P (2010) Toward advanced subgrid models for Lattice-Boltzmann-based Large-eddy simulation: Theoretical formulations. Computers & Mathematics with Applications 59(7):2194–2199
31. Šarić S, Jakirlić S, Djugum A, Tropea C (2006) Computational analysis of locally forced flow over a wall-mounted hump at high-Re number. International Journal of Heat and Fluid Flow 27(4):707–720
32. Smagorinsky J (1963) General circulation experiments with the primitive equations. Monthly weather review 91(3):99–164
33. TheLMA (2010–2013) Thermal LBM on Many-core Architectures. www.thelma-project.info
34. Tölke J, Krafczyk M (2008) TeraFLOP computing on a desktop PC with GPUs for 3D CFD. International Journal of Computational Fluid Dynamics 22(7):443–456
35. Tominaga Y, Stathopoulos T (2010) Numerical simulation of dispersion around an isolated cubic building: Model evaluation of RANS and LES. Building & Environment 45(10):2231–2239
36. Yakhot A, Liu H, Nikitin N (2006) Turbulent flow around a wall-mounted cube: A direct numerical simulation. International Journal of Heat and Fluid Flow 27(6):994–1009
37. Yoshie R, Mochida A, Tominaga Y, Kataoka H, Harimoto K, Nozu T, Shirasawa T (2007) Cooperative project for CFD prediction of pedestrian wind environment in the Architectural Institute of Japan. Journal of Wind Engineering and Industrial Aerodynamics 95(9):1551–1578