Proceedings of the Seventeenth International Conference on
Civil, Structural and Environmental Engineering Computing
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 6, Paper 13.3
Civil-Comp Press, Edinburgh, United Kingdom, 2023
doi: 10.4203/ccc.6.13.3

Simulation of simplified urban flows using Lattice Boltzmann Method

K.K. Maskey and R. Deiterding

Aerodynamics and Flight Mechanics Research Group
University of Southampton, United Kingdom

Abstract

Simulations of urban environments can have substantial benefits to public safety. This can be in scenarios of danger assessment such as those related to the release of chemical and biological hazards. The currently used simple, semi-analytical models have trouble predicting realistic cases and hence require accurate aerodynamic simulations. LBM is used here since it produces high-fidelity simulations within a reasonable timescale. The LBM code is currently being developed based on the AMROC framework, which uses adaptive mesh refinement. Simulations have been conducted on cube configurations at various Reynolds numbers to simulate flow around buildings. The results show that the solver with adaptive mesh refinement can be used to efficiently solve atmospheric problems.

Keywords: lattice Boltzmann method, Cartesian method, adaptive mesh refinement, urban flow

1 Introduction

High-fidelity, unsteady aerodynamic simulations can help substantially in hazard predictions of urban environmental safety. The prediction for atmospheric dispersion can come in the form of chemical, nuclear, or biological hazards. Examples could be an accidental release from nuclear installations or petrochemical plants [1]. In scenarios such as these hazard prediction within minutes or hours can become very useful in preventing harm to public health. Many regulatory bodies from countries around the world employ atmospheric dispersion models (ADM) to safeguard against and predict these threats. The methods employed currently consider the source information and the topography to portray the dispersion process but do not capture the full physics of the turbulent behaviour required for a truly realistic, accurate assessment with detailed time-dependent simulations [2, 3].

The current and most common method used by the industry and many of the regulatory bodies include the Gaussian models. These models – while fast – have many weaknesses and limitations with regard to producing realistic cases. Some of these include their inability to predict near a source, the simplification of turbulence and meteorological effects, and the lack of accounting for the recirculation from multiple buildings. These methods also use steady-state assumptions which do not calculate the travel time of pollutants to receptors [4].

To get around this, more robust methods such as Computational Fluid Dynamics (CFD) are increasingly used. Large-eddy simulation (LES) is used in these methods to capture the turbulent dynamics of a problem. Traditional CFD methods can have very large runtimes when solving large unsteady problems, with some taking days or weeks to fully resolve all relevant scales. This makes them unrealistic for hazard assessments which need to be conducted within much shorter timescales such as hours. Therefore, the Lattice Boltzmann Method (LBM) with adaptive mesh refinement (AMR) is proposed as a quicker solution while still preserving the accuracy provided by traditional CFD. LBM is readily suited for parallel processing, thus, allowing for quick results.

The LBM code is developed based on the AMROC framework, which was initially formed for finite volume schemes but was later adapted for the cell-based LBM. It uses AMR which allows for dynamic local mesh adaptation. It uses a block-based data structure with non-overlapping rectangular grids. The spatial mesh and time step get refined by a given factor and these refined blocks overlay coarser ones. To allow for synchronisation and creating boundary conditions, a layer of halo cells is used [5].

The following section briefly describes the methodology used in achieving dispersion simulations using LBM. For the validation cases in Section 3, we consider an isolated cube in uniform flow and a set of cube arrays to simulate simplified urban flows [6, 7, 8].

2 Method

2.1 Lattice Boltzmann Method

The Lattice Boltzmann method (LBM) is a novel approach to CFD. It solves the Boltzmann equations instead of the Navier-Stokes equations to produce the required macroscopic quantities of interest such as density or velocity. The Boltzmann equation is defined as

$$\frac{\delta f(\mathbf{x}, \boldsymbol{\epsilon}, t)}{\delta t} + \boldsymbol{\epsilon} \cdot \nabla f(\mathbf{x}, \boldsymbol{\epsilon}, t) = \Omega\big((\mathbf{x}, \boldsymbol{\epsilon}, t)\big), \tag{1}$$

where f is the probability distribution function and Ω is the collision operator. The SRT-BGK model is used within the momentum operator to simulate the flow. The SRT-BGK is a weakly-compressible isothermal model [9]. This provides us with the discrete form of the Boltzmann equation as

$$f_i(\mathbf{x} + c\mathbf{e}_i\Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{\Delta t}{\tau_f}\big(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)\big), \tag{2}$$

where f_i is the partial distribution function and τ is the relaxation parameter. The solver also incorporates LES to capture the complex turbulent structures present within urban environments. LES is achieved in LBM by constructing an effective relaxation parameter locally instead of the standard relaxation parameters through [10]

$$\tau_{eff} = \frac{(\nu + \nu_t) + \frac{c_s^2 \Delta t}{2}}{c_s^2}, \tag{3}$$

where the eddy viscosity, ν_t, is defined using a LES model dependent constant, C, spatial step, Δ and characteristic time scale, OP_{LES} as,

$$\nu_t = (C\Delta)^2 OP_{LES} \tag{4}$$

2.2 Adaptive Mesh Refinement

The AMROC framework uses block-structured adaptive mesh refinement (SAMR) to increase the accuracy of the simulations while maintaining lower computational costs. The method is modelled after the work of Berger and Collela, which was tailored for hyperbolic conservation laws on Cartesian meshes [11]. The grids contain a layer of halo cells for implementing boundary conditions and synchronisation between the overlying sub-meshes. In the current approach, cells are clustered together into non-overlapping rectangular grids. A hierarchy of sub-meshes is created recursively as shown in Fig. 1, where G_l is the domain of an entire level [12].

The relationships of the spatial and temporal widths Δx and Δt between the coarser and finer meshes can be reduced to

$$\frac{\Delta x_c}{\Delta x_f} = \frac{\Delta t_c}{\Delta t_f} = r, \tag{5}$$

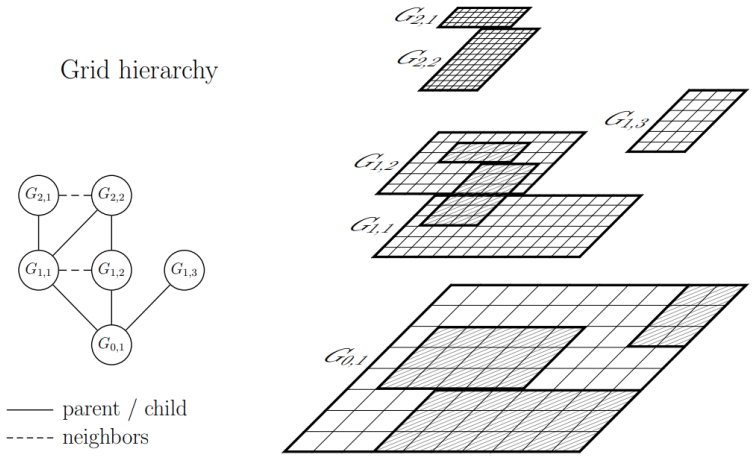

Figure 1: Hierarchy of sub-meshes used within AMR [12].

where c and f relate to the coarser and finer mesh, respectively [13]. r relates to the refinement factor, which can be set to any positive integer between each refinement level. The distribution functions of the finer cells are averaged when transitioning to a coarser cell. Whereas spatial and temporal interpolation is done to create the distribution functions for the transition from coarser to finer cells. The rescaling of the distribution functions between the levels is done using the approach described by Dupius and Chopard [14]. The implementation of AMR within the AMROC framework is detailed in previous works [15, 16].

3 Isolated Cube

The first validation case is based on an isolated cube in uniform flow to represent a simplified building. This validation is compared to the works of Castro and Robins [6] and only utilises the momentum operator. The cube has dimensions of $h = 2.5\,\mathrm{m}$ and is placed on a flat surface with an incoming uniform flow of $3\,\mathrm{m/s}$.

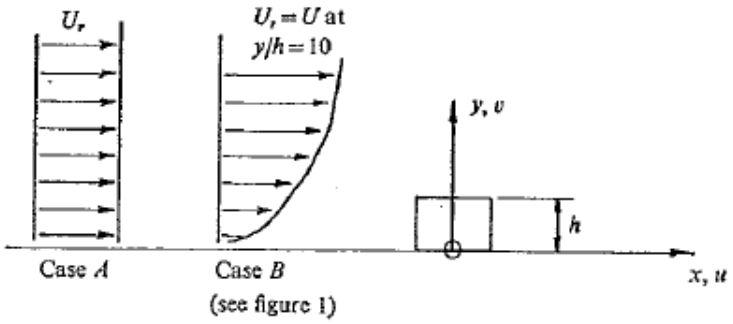

Figure 2: Set up image of isolated Cube cases. Case A and Case B represent the uniform flow and neutrally stable boundary layer flow, respectively [6].

The domain dimensions for the x, y and z planes are set to a value of $28h \times 14h \times 26h$, respectively. The larger domain size was used to remove the blockage effects. The centre of the cube is placed at $30h$ from the inlet and $20h$ from the sides. Sponge zones were placed at all domain boundaries except the bottom with a thickness of 20 cells at the coarsest level, to absorb the pressure fluctuations which occur at the startup of the simulation. Standard sea-level conditions were assumed and the boundary conditions for the domain were set to:

- Front: Inlet (uniform velocity profile)
- Back: Pressure
- Bottom: Symmetry (before cube). No-Slip Wall (after cube)
- Sides and Top: Outlet

The boundary condition on the bottom domain face was set to symmetry to produce a stress-free region to best simulate a cube in a wind tunnel placed on a false floor which has negligible boundary layer height before the cube. The boundary condition for the cube itself was set to Bouzidi bounce-back. The domain used 4 levels of refinement with each having a refinement factor of 2. The simulated pressure coefficient from the centre-line of the cube was measured against the experimental data from case A with the false floor. The experimental data were measured at a Reynolds number slightly larger than $1 \cdot 10^5$. The blockage ratio experienced in the experiment was shown to be 1.5%. The simulations were conducted at a lowered Reynolds number of $2 \cdot 10^4$ with a smaller blockage ratio. The simulations were conducted using 160 processors and required $39.585\,\mathrm{h}$ wall clock time to complete.

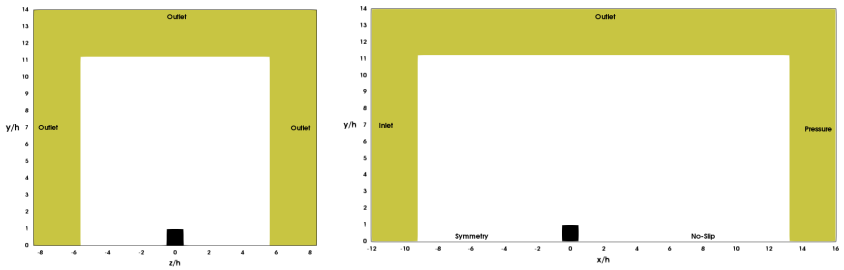

Figure 3: Domain layout with boundary conditions and sponge zone (yellow) set up around the cube. Frontal, $x/h = 0$ (left); Side, $z/h = 0$ (right)

The C_p curve for the setup for case A of the isolated cube is shown in Fig. 7. The pressure coefficient is obtained along the centreline of the cube starting at the bottom of the front face and ending at the bottom of the back face. As can be seen from the image, the general trend of the pressure coefficient matches the experimental findings, however, there are minor shortcomings in particular areas of the C_p curve. The front face has a C_p value that is marginally higher than that registered in the experiments.

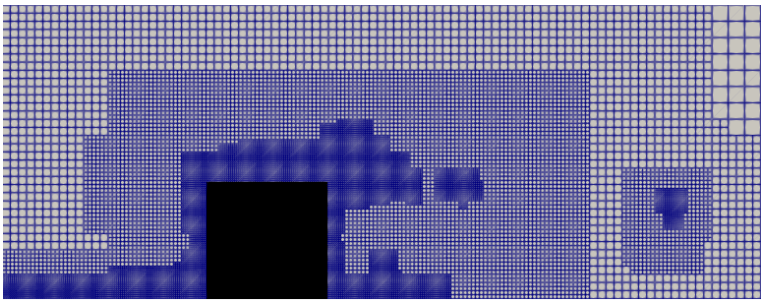

Figure 4: Mesh setup around cube using adaptive mesh refinement.

The front face C_p curve adheres to the experimental findings much more readily past this initial point and matches the transition in C_p seen at the leading edge of the top face. This can even be seen in the velocity images shown previously, where a recirculation region is formed in front of the cube in a more circular nature, displacing the attributes of the horseshoe vortex and sticking closer to the cube. The region past the leading edge of the top surface is highly Reynolds number dependent and the difference in the velocity profiles can be attributed to the high Reynolds number, mesh resolution, and the boundary condition of the cube. A greater agreement was obtained in the detachment region due to the use of the more complex boundary condition compared to a simple no-slip boundary and a higher resolution. The agreement can be seen using the mean velocity variation above the cube shown in Fig. 7. It shows that the minimum value of the velocity close to the wall still does not reach the value achieved in the experiments. However, the boundary layer height seems to be very close to the experimental value. The maximum velocity achieved, and the location of the inflection point is also comparable to the experimental findings, with the simulation returning to the free-stream velocity at a slightly higher height than in the experimental findings.

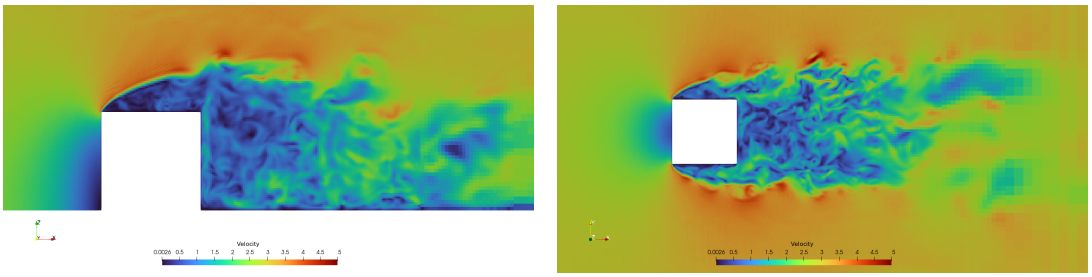

Figure 5: Profile view of instantaneous velocity around isolated cube case A. Side, $z/h = 0$; (left) Topdown, $y/h = 0$ (right)

The back surface seems to display a divergence from the experimental findings. As mentioned in the literature, the back face is very sensitive to the flow conditions from the front of the cube. As the boundary condition, that was chosen for the bottom face of the domain, is not entirely representative of the experiment, the turbulent structures

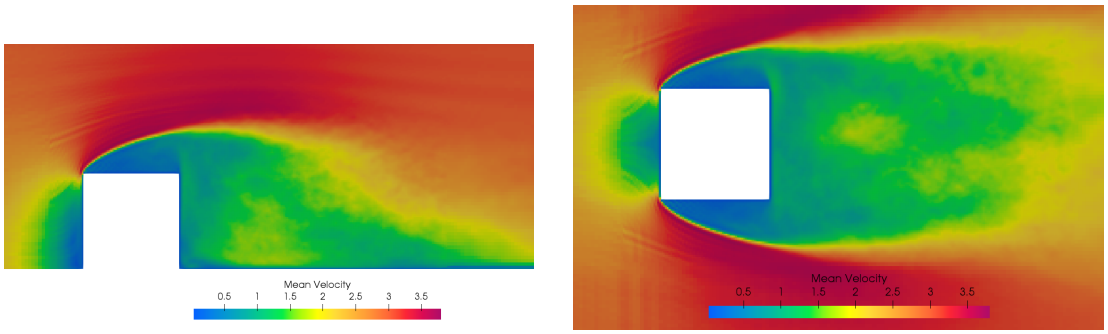

Figure 6: Profile view of mean velocity around isolated cube case A. Side, $z/h = 0$; (left) Topdown, $y/h = 0$ (right)

interacting with the wake are not truly accurate. The lack of refinement further downstream in the wake region and the presence of large horseshoe vortices on the sides of the cube also lead to the discrepancy.

The simulation as a whole, however, shows a strong agreement and trend with the experimental findings especially close to the cube. The major regions of difference are near the recirculation region along the top surface and the wake region, where the lack of resolution is not able to fully capture all turbulent details.

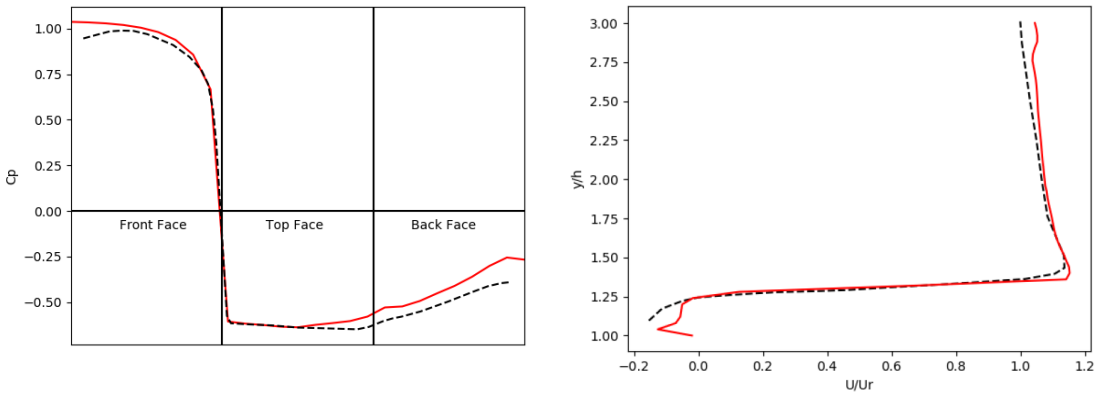

Figure 7: Simulated (line) vs experimental data (dashed) around isolated cube case A. C_p curve around centreline (left), Mean velocity variation for centreline at $x/h = 0$ (right).

4 DIPLOS Cube Array

The next set of cases was conducted on an array of cubes based on Project DIPLOS (Dispersion of Localised Releases in a Street Network). The DIPLOS project aimed to determine the nature of the flow and dispersion process in street networks that contain localised scalar sources [8].

The simulations of the flow around the cube array were compared to the experi-

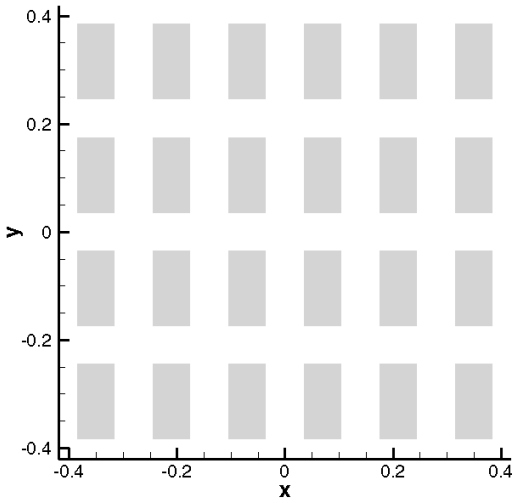

Figure 8: Top down view of DIPLOS cube array setup.

mental works such as those shown by Castro et al. [7]. The computational setup for the simulations considers a small section of the experimental setup. The cubes themselves have a dimension of $h \times 2h \times h$, where $h = 70\,\mathrm{mm}$. The domain dimensions for the uniform array case are set to $12h \times 12h \times 12h$ w.r.t the x, y and z planes. The cube itself has dimensions as detailed in the experiment. Within the domain is an array of 4×6 cubes spaced at one cube length, h, away from each other. The streets are differentiated as "short streets" which are $1h$ long and parallel to the x-axis, while the "long streets" are $2h$ long and parallel to the y-axis. The velocity was set to $2\,\mathrm{m/s}$ with the incoming wind direction of 0^o. The Reynolds Number was $\mathrm{Re}_\tau = u_\tau h/\nu \approx 1000$ for this problem. The domain boundaries were set to be periodic in the x and y planes, while the top of the domain was set to an outlet and the floor to a no-slip boundary conditions. The flow itself was then driven using pressure forcing methods, taking into account the friction velocity stated by the Reynolds number above. The mesh itself used 3 layers of static refinement with a refinement factor of 2 which encapsulated the cube layer. The simulations were conducted using 512 processors and required $28.743\,\mathrm{h}$ wall clock time to complete.

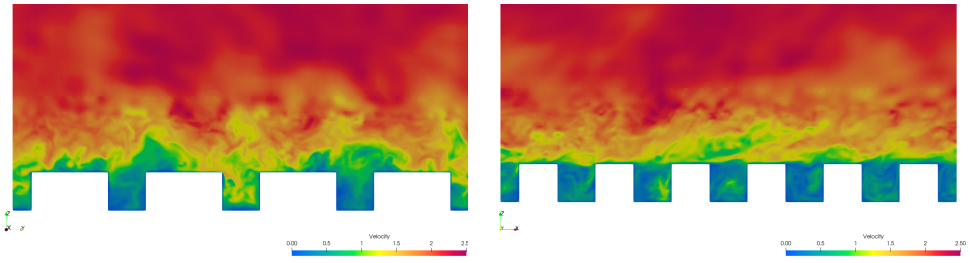

Figure 9: Instantaneous velocity distribution of DIPLOS case at $x = -0.2$ (right) $y = -0.1$ (left).

The velocity distribution for the DIPLOS array is shown in Figs. 10 and 9. The mean velocity profiles are shown in Fig. 11. The profile is obtained by spatially and

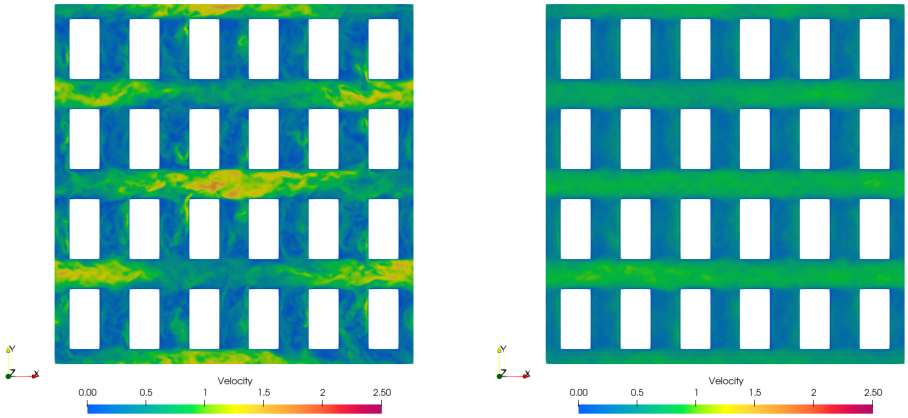

Figure 10: Velocity distribution of DIPLOS case at $z = 0.05$. Instantaneous (left), averaged (right).

temporally averaging the velocity profile. The initial region within the array layer (below $z/h = 1$) is more sensitive to the boundary conditions of the cubes and therefore the mean velocity curve do not correlate as strongly with the experimental data as in the region above the array layer. This can be seen in both normalised velocity curves. The data within the layer above the cube array between $1 < z/h < 4$ agrees far more strongly with the experimental findings than even the LES data when looking at the normalisation with the freestream velocity. This behaviour is not seen in the profile normalised with the friction velocity, however, this may be due to the method in which the friction velocity was defined. The profile at the very top of the domain, nonetheless, still agrees quite well to the experimental findings.

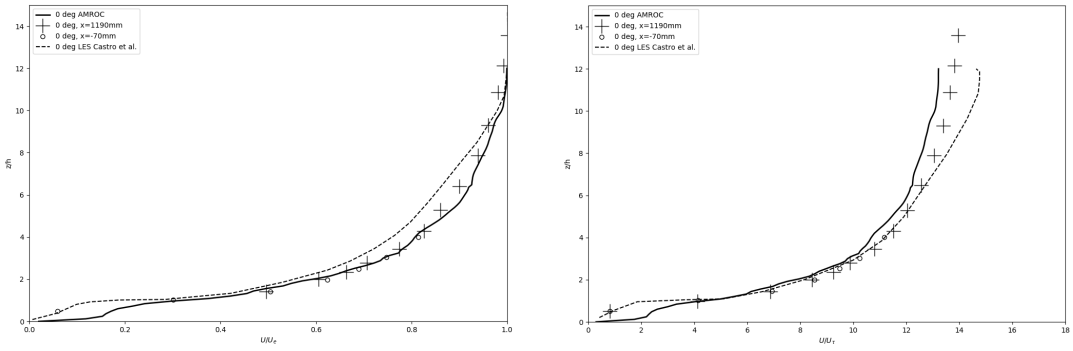

Figure 11: Mean velocity variation (in space and time) for the DIPLOS cube array from experimental and comparison. Normalised with freestream velocity (left) and friction velocity (right).

5 Concluding remarks

This papers described an approach to predicting simplified urban flows using Lattice Boltzmann Method. The cases conducted on the AMROC framework used a block-based adaptive mesh refinement alongside LES capabilities to produce an efficient solver for high-fidelity aerodynamic simulations. Two cases were investigated to simulate simplified urban flows. The isolated cube in uniform flow and the DIPLOS cube array were able to show good computational accuracy in comparison to the experimental results. The intricate turbulent structures present within the cases highlight the need for such a method in producing realistic, unsteady, physics-based cases for works involving urban flows and hazard predictions.

Acknowledgements

The project is funded by dstl and we would like to thank them for their collaboration. The authors acknowledge the use of the IRIDIS High-Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

References

[1] M. Caputo, M. G. & Schlamp, M., ”Intercomparison of atmospheric dispersion models”, Atmospheric Environment 37, 2435–2449. pages 1, 2, 9., 2003.

[2] Macdonald, R., THEORY AND OBJECTIVES OF AIR DISPERSION MODELLING. pages 1., 2003.

[3] Boris, J. et al. “Validation of an les urban aerodynamics model for homeland security,” 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition [Preprint], 2009.

[4] Holmes, N. S. & Morawska, L., ”A review of dispersion modelling and its application to the dispersion of particles: An overview of different dispersion models available”, Atmospheric Environment 40, 5902–5928. pages 1, 2, 9., 2006.

[5] Deiterding, R., ”Block-structured Adaptive Mesh Refinement - Theory”, Implementation and Application, ESAIM: Proceedings 34, 97–150., 2011.

[6] Castro, I. P., Robins, A. G., ”The flow around a surface-mounted cube in uniform and turbulent streams”, Journal of Fluid Mechanics 79(2), 307–335. pages i, 2, 29, 32, 33, 43, 45, 58., 1977.

[7] Castro, I. P., Xie, Z.-T., Fuka, V., Robins, A. G., Carpentieri, M., Hayden, P., Hertwig, D. & Coceal, O., ”Measurements and computations of flow in an urban street system”, Boundary-Layer Meteorology 162, 207–230. pages 52, 53, 54., 2016.

[8] Fuka, V., Xie, Z.-T., Castro, I. P., Hayden, P., Carpentieri, M. & Robins, A. G., ”Scalar fluxes near a tall building in an aligned array of rectangular buildings”, Boundary-Layer Meteorology 167, 53–76. pages 3, 52, 53, 54., 2017.

[9] Bhatnagar, P. L., Gross, E. P. & Krook, M., "A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems", Physical Review 94(3), 511–525. pages 13., 1954.

[10] Gkoudesnes, C. and Deiterding, R., "Verification of the wale large eddy simulation model for adaptive lattice boltzmann methods implemented in the AMROC framework," Cartesian CFD Methods for Complex Applications, pp. 123–144., 2020.

[11] M.J. Berger and P. Colella. "Local adaptive mesh refinement for shock hydrodynamics". In: Journal of Computational Physics 82.1, pp. 64– 84., May 1989.

[12] Deiterding, R., Domingues, M.O. & Schneider, K. "Multiresolution analysis as a criterion for effective dynamic mesh adaptation – a case study for Euler equations in the SAMR framework AMROC," Computers & Fluids, 205, p. 104583., 2020.

[13] Grondeau, M. and Deiterding, R. "Direct prediction of flow noise around airfoils using an adaptive lattice Boltzmann method," Handbook of Wind Energy Aerodynamics, pp. 1463–1486., 2022.

[14] A. Dupuis and B. Chopard. "Theory and applications of an alternative lattice Boltzmann grid refinement algorithm". In: Physical Review E 67.6, p. 066707., 2003.

[15] R. Deiterding and S. L. Wood. "An adaptive lattice Boltzmann method for predicting wake fields behind wind turbines". In: New Results in Numerical and Experimental Fluid Mechanics X. Ed. by A. Dillmann et al. Vol. 132. Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer, pp. 845–857., 2016.

[16] K. Feldhusen, R. Deiterding, and C. Wagner. "A dynamically adaptive lattice Boltzmann method for thermal convection problems". In: J. Applied Math. and Computer Science 26, pp. 735–747., 2016.