The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5–-9 June 2022, Oslo, Norway

LATTICE-BOLTZMANN SIMULATIONS OF TRAFFIC-RELATED ATMOSPHERIC POLLUTANT DISPERSION IN URBAN AREAS

Mathis Pasquier[1,2,*], Stéphane Jay[1], Pierre Sagaut[2]

[1] IFP Energies Nouvelles,
Institut Carnot IFPEN Transports Energie,
1-4 Av. du Bois Préau,
92852 Rueil-Malmaison, France

[2] Aix Marseille Univ,
CNRS, Centrale Marseille, M2P2 UMR 7340,
13451 Marseille, France

[*] corresponding author : `mathis.pasquier@IFPEN.fr`

Key words: Lattice-Boltzmann, pollutant dispersion, turbulent atmospheric flows, traffic emissions

Abstract. The developing field of urban physics includes computational fluid dynamics (CFD) as a tool to model wind comfort, heat management and pollutant dispersion in cities. In particular, road traffic emissions significantly contribute to air pollution and should be considered in atmospheric dispersion simulations. To this end, the lattice-Boltzmann method (LBM) offers a promising alternative to traditional finite-volume CFD solvers in terms of computational cost and accuracy. At IFP Energies Nouvelles (IFPEN), a recent emission model relying on real-life driving data recorded with a mobile application was used to construct urban emission maps. However, it has not been coupled yet with a precise unsteady CFD solver, which could provide local unsteady and accurate information about local concentration fields. We propose to combine the LBM open-source code OpenLB with the emission model designed at IFPEN to simulate traffic-induced pollutant dispersion in an urban-like environment. The LBM code is used to solve the Navier-Stokes equations as well as the passive scalar transport with a double distribution function (DDF) approach. The solver is successfully validated on the well-known CODASC test case and a first evaluation of the impact of a representative urban setting on pollutant dispersion with non-uniform sources is proposed.

1 INTRODUCTION

The Lattice-Boltzmann Method (LBM) [1] has gathered increasing interest in the CFD community for about a decade now and stands as a credible alternative to traditional finite-volume LES-based CFD tools. One can cite a few of the main advantages of the standard LBM: first, its ability to simulate incompressible isothermal flows with second-order space-time accuracy. Second, the fact that – contrary to traditional finite-volume methods – there is no need to solve an intermediary Poisson equation, which considerably alleviates the computational burden. Third, it is based on a local stencil – ignoring boundary conditions –, using only direct neighbours at each time step, which makes it particularly prone to parallelization.

Thus, an increasing number of studies examines atmospheric flows using the LBM : for instance Merlier et al. [2, 3] simulate the transport of neutral and dense gas in realistic urban geometries using the commercial code ProLB, and Buffa et al. [4] compute pressure loads on high-rise buildings. Lenz et al. [5] simulate wind flow in a local urban neighbourhood with real-time performance on GPUs using a rather coarse mesh (with minimal grid size of the order of one meter) and Wellman [6] further analyses passive pollutant transport. Ahmad et al. [7] carried out aerodynamic simulations in complex urban environments for wind comfort purposes, Inagaki et al. [8] analyse the atmospheric boundary layer dynamics and Onodera et al. [9] study pollutant transport with a mesoscopic meteorological model, also achieving real-time performances on GPUs with one-meter mesh size. This review highlights the fact that the LBM appears to be promising in terms of performance and accuracy for atmospheric computations on rather coarse meshes, in particular when GPUs are employed.

Accounting for transport-related air pollution in CFD simulations requires to adequately represent the emissions sources in the computational setup. Most of the CFD studies addressing road emissions employ uniform and stationary line sources to represent road segments and vehicular exhausts. Emissions rates can be estimated from dynamical emissions models or emissions factors. The accuracy of these models turns out to be very limited for urban traffic applications, where driving style and built infrastructures significantly influence the emissions and the dispersion. Advanced modelling approaches are found in the literature, for example using source terms in transport equations to account for particle drag [10] or traffic-induced turbulence [11]. The vehicles can also be represented by moving solids [12] or a viscous additional fluid [13]. The best option to accurately represent the vehicles' dynamics is to couple a microscopic traffic simulator with the CFD solver [14, 6, 15]. Such traffic models account for the complex non-linear interactions among the fleet and output realistic information about the positions and speeds of the vehicles, which can be used with emissions models to better represent the spatial and temporal variability of the road emissions. However, the speed profiles obtained with traffic models are not always representative of real behaviours, and alternative options such as GPS-based measurements can improve the accuracy of the subsequent emissions estimations [16].

This work aims at constructing a functional (but not necessarily optimally accurate) modeling chain to simulate traffic-induced pollution at the local urban scale. The thermal stratification of the atmosphere and the chemistry are ignored and an open-source LBM solver is adapted to simulate incompressible isothermal highly turbulent flows with passive scalar transport. The sources modeling is improved in comparison with uniform and stationary sources, by exploiting a GPS-based emissions simulator developed at IFPEN and coupling it for the first time with a CFD solver. Section 2 describes the LBM methodology for the simulation of turbulent dispersion, section 3 provides detail about the emissions modeling strategy and section 4 presents some validation results for the LBM solver before coupling it to the emissions model on a realistic geometry.

2 DDF LBM-LES MODEL

2.1 Software presentation

In this work we use the open-source Lattice-Boltzmann code OpenLB designed at the Karlsruhe Institute of Technology [17]. This code is written in C++ and is constructed in the most generic fashion to facilitate the implementation of new models. Among the options already available, one can simulate incompressible laminar and turbulent flows, reactive flows, flows in porous media, particulate flows and thermal flows. The code is parallelized using the MPI and OpenMP libraries for CPUs, and a recently released version is adapted to GPU architectures. The code is available for academic and industrial applications under the license GNU GPL2.

In this work we address highly turbulent flows with passive scalar transport and the initial version of OpenLB that was used in the first place (version 1.4) had two major limitations that made the code highly unsuitable for such purposes : first, the absence of mesh refinement and second, the absence of stable models for high-Reynolds number flows. The former limitation was not addressed here even though it would be very useful to have local mesh refinement to limit the computational expanses for urban flows. The latter issue was solved by implementing additional ingredients in the code. This functional numerical method is presented in the subsequent sections for highly turbulent incompressible isothermal flows and passive scalar transport using the LBM.

2.2 High-Re incompressible isothermal flows

The LBM originates from the Boltzmann equation, which describes the evolution of a probability density function f that characterizes the state of a system composed of many colliding particles in the phase space (defined by positions $\mathbf{x}$ and microscopic velocities $\mathbf{v}$ of the particles). The original equation is simplified using the BGK approximation for the collision operator, and is discretized in the velocity space using Gauss-Hermite quadrature rules. f is discretized into populations f_i's associated to a set of q discrete velocities $\mathbf{v}_i$ which compose the D3Q27 ($q = 27$) and D3Q19 ($q = 19$) lattices commonly used in 3D. The density and momentum of the fluid moving with macroscopic velocity $\mathbf{u}$ are related to the populations as:

$$\rho = \sum_i f_i \quad \text{and} \quad \rho\mathbf{u} = \sum_i \mathbf{v}_i f_i$$

Space-time and velocity discretization lead to the Lattice-Boltzmann equation (LBE)

$$f_i(\mathbf{x} + \Delta t\mathbf{v}_i, t + \Delta t) - f_i(\mathbf{x}, t) = -\bar{\omega}\Delta t \left(f_i(\mathbf{x}, t) - f_i^{\mathrm{eq}}(\mathbf{x}, t)\right) \quad \text{with} \quad \frac{1}{\bar{\omega}} = \frac{1}{\omega} + \frac{\Delta t}{2} \tag{1}$$

where ω is the relaxation rate of the f_i's to the equilibrium distributions defined by

$$f_i^{\mathrm{eq}} = w_i\rho \left[1 + \frac{\mathbf{u}\cdot\mathbf{v}_i}{c_s^2} + \frac{(\mathbf{u}\cdot\mathbf{v}_i)^2}{2c_s^4} - \frac{\mathbf{u}\cdot\mathbf{u}}{2c_s^2}\right]. \tag{2}$$

with the quadrature weights w_i. The LBE translates into the *collision-propagation algorithm* at the heart of the LBM :

1. **Collision step:** compute intermediate populations as $f_i^{\mathrm{coll}} = f_i - \bar{\omega}\Delta t \left(f_i - f_i^{\mathrm{eq}}\right)$

2. **Propagation step:** move the intermediate populations to direct neighbours along discrete velocities: $f_i(\mathbf{x} + \Delta t \mathbf{v}_i, t + \Delta t) = f_i^{\text{coll}}(\mathbf{x}, t)$

The propagation step imposes the necessity of a uniform cartesian mesh for the LBM. A Chapman-Enskog perturbative analysis shows that the hydrodynamic limit of the LBE yields the following mass and momentum conservation laws (Navier-Stokes equations, NSE):

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{3a}$$

$$\partial_t(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot (2\mu \mathbf{S}) + \mathcal{O}(\text{Ma}^3) \tag{3b}$$

with the dynamic viscosity $\mu = \rho c_s^2 \left(\frac{1}{\bar{\omega}} - \frac{\Delta t}{2}\right)$, the barotropic equation of state $p = c_s^2 \rho$ (where $c_s = \sqrt{rT_0}$ is the newtonian sound speed) and $\mathbf{S}$ the rate-of-strain tensor. The momentum additional Ma^3-like error term originates from the limited quadrature in the velocity space and justifies the adequacy of the aforementioned standard LBM to isothermal *weakly-compressible* flows.

The standard LBM is known to suffer from severe stability issues as the Reynolds number grows (which numerically translates into the *zero-viscosity limit* with $\bar{\omega}\Delta t \to 2$). One option to overcome these issues is to use a modified collision kernel with enhanced stability properties. OpenLB offers only multiple relaxation times (MRT/TRT), entropic (ELBM) and projected regularized (PR) [18] models, which are unstable in our case. In the literature the state-of-the art options are the cumulant [19] and hybrid-recursive regularized (HRR) [20] models, which considerably increase the ability of the LBM to address high Reynolds numbers flows. Yet, none of them is available in OpenLB and both methods require an increased computational effort to work properly. Thus, in order to limit the implementation difficulties and the additional computational overhead, we chose to implement the recursive-regularized (RR) model which was designed by Malaspinas [21] to improve the stability of the D3Q27 LBM. In our case we use the D3Q19 lattice for performance purposes, for which the RR kernel is not stable enough. Thus we use a subgrid-scale model to further filter out the spurious oscillations occurring in the macroscopic fields. Here the Shear-Improved Smagorinsky model (SISM) [22] was chosen for its near-wall improved accuracy. Furthermore, spurious pressure oscillations are generated at the initialization of any simulation, which eventually lead to divergence over long simulation times. To cure this issue we implemented sponge zones adapted to the LBM [23] close to the boundaries of the computational domain to damp spurious pressure waves.

All these ingredients added to each other contribute to make the code OpenLB functional for high-Reynolds flows without mesh refinement. The computational overhead induced by our implementation of the RR kernel, the SISM model and the sponge zones is not significant compared to plain BGK with D3Q19.

2.3 Passive scalar transport

The transport of a passive scalar in the wind is described by a simple advection-diffusion equation (ADE) of the form:

$$\partial_t \chi + \mathbf{u} \cdot \nabla \chi = \kappa \nabla^2 \chi \tag{4}$$

where χ is the molecular concentration of pollutant (in ppm) and κ is the molecular diffusivity. In order to solve equation (4), two options are available : either solve NSE with LBM and ADE with finite differences (hybrid approach) [9] or solve both NSE and ADE with LBM (double distribution function approach, DDF) [24]. In OpenLB the second option is readily available, which is why we choose a DDF approach. The idea is to introduce an additional family of populations g_i such that

$$\sum_i g_i = \chi \tag{5}$$

The g_i's are governed by the following LBE equation:

$$g_i(\mathbf{x} + \Delta t\mathbf{v}_i, t + \Delta t) - g_i(\mathbf{x}, t) = -\bar{\omega}_g \Delta t \left(g_i(\mathbf{x}, t) - g_i^{\text{eq}}(\mathbf{x}, t)\right) \quad \text{with} \quad \frac{1}{\bar{\omega}_g} = \frac{1}{\omega_g} + \frac{\Delta t}{2}. \tag{6}$$

where ω_g is the relaxation rate of the g_i's toward their equilibria defined similarly to (2) with χ instead of ρ. A Chapman-Enskog analysis shows that the corresponding hydrodynamic limit of (6) is

$$\partial_t \chi + \mathbf{u} \cdot \nabla \chi = \kappa \nabla^2 \chi - \frac{\kappa}{c_s^2} \nabla \left(\frac{\chi}{\rho} \nabla p\right) \tag{7}$$

which is a regular ADE with an error term that can be shown to scale like the square of the Mach number compared to the diffusive term. It is thus negligible for low-Mach flows. The molecular diffusivity reads $\kappa = c_s^2 \left(\frac{1}{\bar{\omega}_g} - \frac{\Delta t}{2}\right)$. For the ADE, only the first moment of the g_i's is conserved, which means that a D3Q7 lattice is sufficient in 3D and first-order truncated discrete equilibria (2) can be used. Yet, for stability purposes, it is better to use a D3Q19 lattice since the D3Q7 is shown to be equivalent to a centred finite-difference scheme [25] which is known to be unstable at high Péclet numbers. Additionally, we use a subgrid-scale model as for the f_i's with a turbulent scalar diffusivity coupled to the turbulent kinematic viscosity through the turbulent Schmidt number chosen here as $\text{Sc}_t = 0.7$. Finally, we also apply the recursive regularization procedure to the g_i's, which was never done in the literature so far, in order to improve the stability and avoid spurious oscillations in the scalar field.

2.4 Note on the boundary conditions

The boundary conditions used here should be robust enough at high Reynolds numbers. The bounce-back scheme is used for no-slip Dirichlet velocity conditions and for zero-flux Neumann scalar conditions [26]. At the inlet, we impose a velocity profile using a projected regularization of the populations and at the outlet we impose constant pressure using a finite-difference velocity gradient estimation to reconstruct the populations (BC3 and BC4 from Latt et al. [27] respectively). For oblique no-slip velocity boundaries we use the interpolated bounce-back scheme designed by Bouzidi et al. [28] and Dirichlet conditions are imposed for the scalar sources using BC3 from Latt et al.

The inlet turbulence is generated using the synthetic eddy method in its original formulation proposed by Jarrin et al. [29]. This method was chosen for its ease of implementation and because the generated turbulence sustains itself rather satisfactorily in the computational domain.

3 TRAFFIC EMISSIONS MODELLING STRATEGY

We consider a generic road network $\mathcal{R}$ composed of several road segments and compute non-uniform stationary traffic-related emission rates in g/s/km on each of these segments. The method used here is a simplification of the procedure designed by De Nunzio et al. [30] and is used to construct a complete modeling workflow when coupled to the LBM CFD model described above. The method is decomposed into three main parts:

- **prior estimation of macroscopic traffic data on the network.** De Nunzio et al. recover information about road infrastructure, topology and average traffic speed from a Geographic Information System (GIS) in order to compute the emissions on real road networks. In our study, we want to exhibit control parameters that encapsulate traffic-related uncertainty for future uncertainty quantification studies. Thus we use a microscopic traffic simulator (the SUMO library available in Python3) that outputs segment-wise (space and time-) average vehicular density $\langle \varrho \rangle$ and vehicle speed $\langle v \rangle$ from a specified road network $\mathcal{R}$ and a given value for the overall traffic flow q (which is considered as an aleatory variable). The traffic flow is assumed here to be the same on every possible route taken by the vehicles.

- **prediction of driving data.** Driving data consists in realistic speed profiles. A statistical model was trained by Laraki et al. [31] on real-life GPS data recorded in Paris and Lyon (France) and is able to generate realistic driving data for an arbitrary network just from GIS data. The driving data is clustered into several categories depending on macroscopic features such as the level of congestion, the presence of a traffic light, an intersection with priority, a stop and yield sign, or the level of curvature of the road. Due to the remaining statistical dispersion of speed profiles amongst the categories, they are further grouped into several clusters for each category depending on the dynamical drivers' behaviours (presence/absence of stopping point and congestion level). The driving data are generated by the machine learning model from the value of the average traffic speed $\langle v \rangle$ using markovian processes, the probability distribution of which is different for each cluster of each category. For more detail about the generation of the driving data, see [31]. Ultimately, each segment is associated to a category and the value of the velocity v_k for the cluster k of this category at a given location x depends on the road network, the length L of the segment and the underlying imposed traffic flow.

- **computation of the emissions.** The NOx and CO2 emission rates are computed with a physical engine model designed by Thibault et al. [16]. The model is able to deal with 1Hz GPS-sampled real-life driving data and was successfully validated against experimental data by Thibault et al. It computes emissions data from vehicle specifications (engine type, etc.), the topography of the road segment and the driving behaviour (a velocity profile). The emissions profile obtained from a velocity profile v_k is denoted e_k and is associated to a type of driving behaviour (a cluster) among those composing the category of the considered road segment. Since the fleet is composed of several drivers with different behaviours and of several types of vehicles (we consider petrol and diesel engines in this study) we can aggregate the emissions to obtain the total emissions associated to a given segment:

$$e = \sum_{k=1}^{N_c} \alpha_i \left(\beta e_k^{\text{gas}} + (1-\beta)\, e_k^{\text{diesel}} \right) \tag{8}$$

where β is the proportion of petrol engines vehicles and α_i are weights depending on the time of the day and the congestion level (which is assumed here to be determined by $\langle v \rangle$). Since the v_k's are stochastic profiles, the emissions profiles are also stochastic. Their statistical variability can be reduced by averaging the emissions over several profiles before using it as such in CFD simulations. The emissions computed so far are emissions rates per vehicle, so the final emissions rates are obtained by multiplying the vehicular emission rate e with the average traffic density :

$$E = \langle \varrho \rangle e.$$

Although the underlying method to generate driving and emissions data is that of De Nunzio et al., a major difference with their work is that here we keep track of the spatial variability of the emissions. De Nunzio et al. aggregated emissions amongst clusters but also averaged them over each segment, thus losing information about possible emission peaks before traffic lights for instance. The main interest to keep track of the non-uniformity is that here we consider an urban neighbourhood instead of a whole city, so the spatial resolution is higher. Yet, the spatial variability is not fully considered, since the traffic density is averaged out. Moreover, the temporal variability of the emissions is ignored, which makes it debatable to use them in unsteady CFD simulations. This temporal variability seems to be directly determined by the value of ϱ. Thus, these traffic density-related simplifying assumptions leave room for improvement and shall be addressed in future studies.

4 VALIDATION RESULTS

In order to validate the ability of OpenLB to handle passive scalar transport with a DDF approach in highly turbulent flows, we consider a test case of traffic-induced pollution in a canyon-like geometry, namely the CODASC wind tunnel experiment [32]. This consists in a constant and uniform release of SF_6 between two long buildings and was originally designed to evaluate the influence of tree-planting on pollutant dispersion The geometry and our results are presented on figure 1. We use a numerical setup similar to that of Merlier et al. [2], without trees. The inlet velocity is set using a power-law velocity profile with synthetic turbulence characterized from the experimental velocity statistics (mean streamwise velocity and turbulent intensities), and 1000 synthetic eddies are injected. 25 nodes are used over the separation distance between the two obstacles, and a total of about 4.3×10^7 grid nodes are used in total. The simulated time is set to 400 seconds with a time step equal given by the acoustic scaling $\Delta t = \Delta x/20$, which leads to an effective computation time of about 34 hours on 360 CPUs. The Reynolds number is set to 4.6×10^4 and the molecular Schmidt number based on the diffusivity of SF_6 is set to 0.68. The figure 1 highlights the presence of a strong canyon vortex at the center of the canyon, which seems to be accurately captured by OpenLB. It is driven by shear forces induced by the detachment of the flow on the first obstacle and its reattachment before the edge. The vortex enhances air exchange between the interior of the canyon and the exterior, by bringing the pollutant against the upstream wall (wall A), thus lowering the pollution level on the downstream wall (wall B). The pollutant is then evacuated outside of the canyon, which can be observed on figure 1. The ability of the code to predict dispersion is assessed by comparing experimental and numerical data on both of the interior walls of the canyon. This is done in the table given in table 1, where statistical metrics described by Chang and Hanna [33] are evaluated gathering wall A and wall B on one hand and separating them on the other hand. We can see that the accuracy of the results is very acceptable on wall A whereas wall B concentration levels show slight discrepancies. Further decomposition of the data on several height levels along the building (not shown here) reveal that the concentration levels are particularly overestimated on wall B close to the ground close to the extremities of the canyon. The values at this location underestimate the reference data by more than factor of two, which damages the geometric mean bias (MG) indicator, which are based on a logarithmic scale. Also, negative values of the fractional bias (FB) indicate a systematic over-estimation of the reference data on both walls, particularly on wall B, which damages MG. This might be due to the fact that the concentration levels on the downstream wall close to the ground are harder to capture with a coarse mesh. It is not uncommon in the literature to struggle more to capture the concentration levels on wall B, so the results obtained with OpenLB are considered to be valid for passive scalar transport.

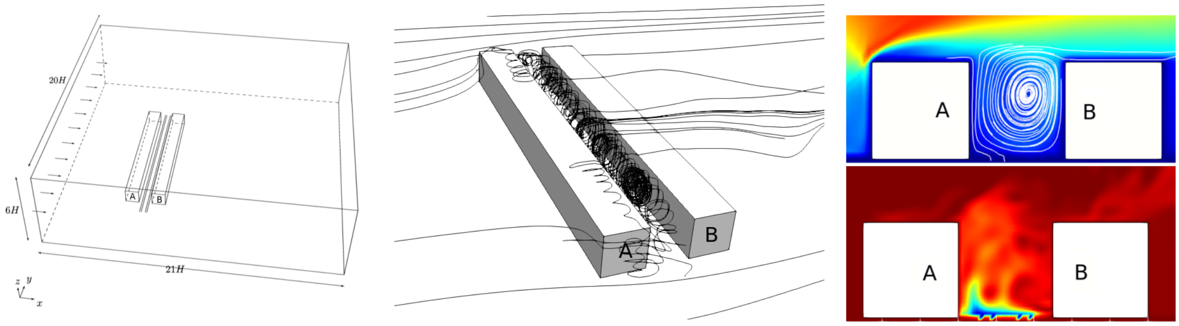

Figure 1: Computational domain (left), streamlines (center), canyon vortex + instantaneous concentration (right)

	FAC2	FB	RNMSE	MG	VG	R
validity range	> 0.5	$]-0.3;0.3[$	< 1.2	$]0.7;1.3[$	< 4	> 0.8
A + B	0.802 ✓	−0.22 ✓	0.404 ✓	0.713 ✓	1.28 ✓	0.911 ✓
A only	0.788 ✓	−0.2 ✓	0.354 ✓	0.765 ✓	1.24 ✓	0.82 ✓
B only	0.816 ✓	−0.289 ✓	0.38 ✓	0.664 ✗	1.31 ✓	0.934 ✓

Table 1: Statistical indicators reviewed by Chang and Hanna [33] for scalar transport validation purposes, and their value obtained for the simulation results with OpenLB gathering or separating wall A and wall B data.

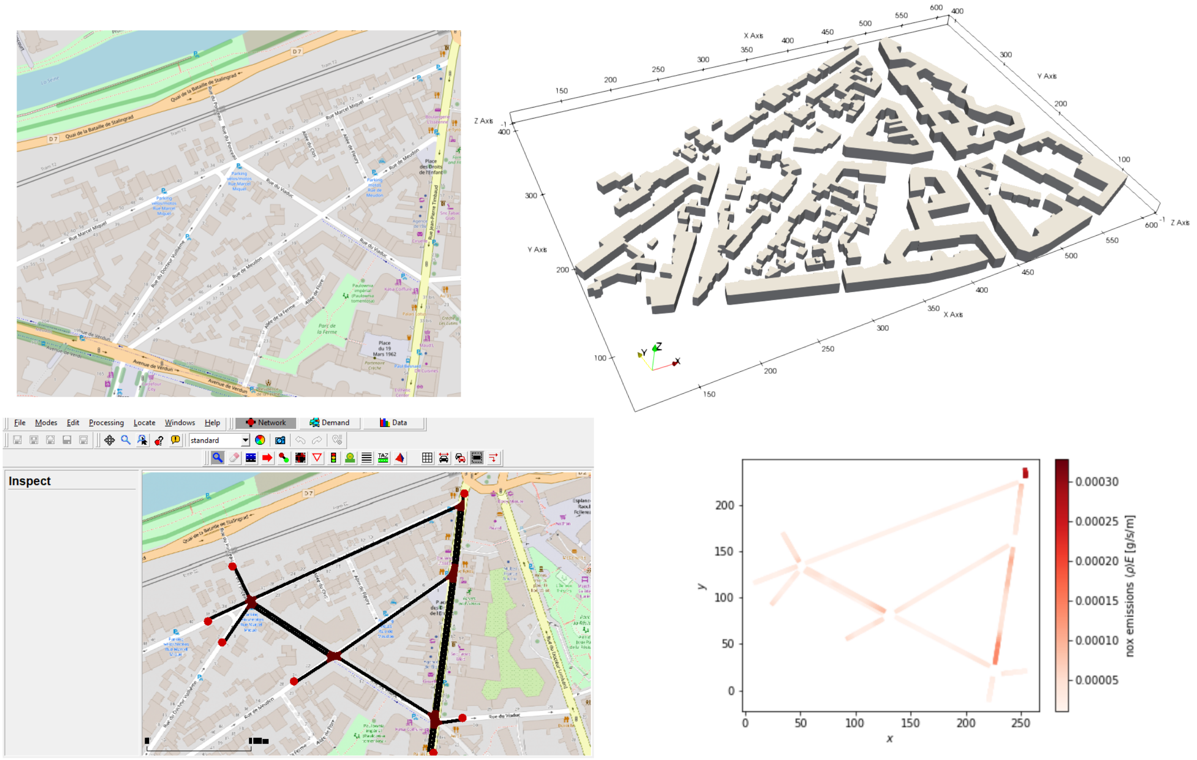

Figure 2: Location map (top left), extruded geometry from map (right), microscopic traffic simulation setup (bottom left), non-uniform stationary emissions on the road network (bottom right)

5 APPLICATION TO A REALISTIC CASE

The realistic application presented here couples the LBM solver with the emissions modeling strategy detailed in section 3, with the microscopic traffic simulator (SUMO), the machine-learning driving data generator and the physical engine emissions model. The geometry of the test case is extracted from an urban neighbourhood of southern Parisian suburbs as depicted on figure 2. The buildings are built by extrusion from the map, with arbitrary heights set between 5 and 10 meters, and the geometry scale is kept to 1:1. Then, the road network is built using the editor NETEDIT available with SUMO, (bottom left on figure 2) using field information such as traffic lights locations, road signs and intersections priorities. The traffic flow q is set to 200 vehicles per hour on each route taken by the vehicles, which are defined as admissible sequences of network links joining two extreme nodes. The microscopic traffic simulation is run for 800 physical seconds, which is enough for the average speeds and densities on each segment to be converged. The average speed values are used to determine the macroscopic congestion level on each segment and to generate speed profiles. Then the emissions are computed on the network using the driving data and the average traffic densities (bottom right on figure 2).

The emissions are then used as an input in OpenLB to impose Dirichlet conditions for the concentration at the road level. Dirichlet conditions for the upward velocity are also imposed on the sources to effectively impose the correct emission rates. The physical simulated time is set to 600 seconds and the Reynolds number is chosen equal to 10^6. The mesh size is set equal to 1 meter, which is comparable to what was done by Lenz et al. [5] and Ondera et al. [9]. Using the same acoustic scaling as for the CODASC test case, the effective computation time on 360 CPUs is thus about 8.5 hours. The instantaneous concentration field at pedestrian height is presented on figure 3, and we can see that the non-uniformity of the emissions significantly impacts the spatial distribution of the concentration levels in the neighbourhood of the road axes. For instance, one can note the particularly high pollution level at the bottom-right intersection which is due to the fact that the corresponding avenue is a double lane with a traffic light at its end. Thus, the vehicles produce higher emission rates locally, and such an information would be lost by spatially averaging the emission rates over the road segments. Note that the LBM solver for passive scalar transport appears to be stable enough although negative concentration values locally occur. This is due to the fact that the numerical scheme does not guarantee positivity, which can be bypassed by cropping local negative values. This occurs in particular close to the line sources, where concentration and velocity discontinuities are caused by the rather coarse mesh size. This could be cured by using local mesh refinement, which is not yet available in OpenLB.

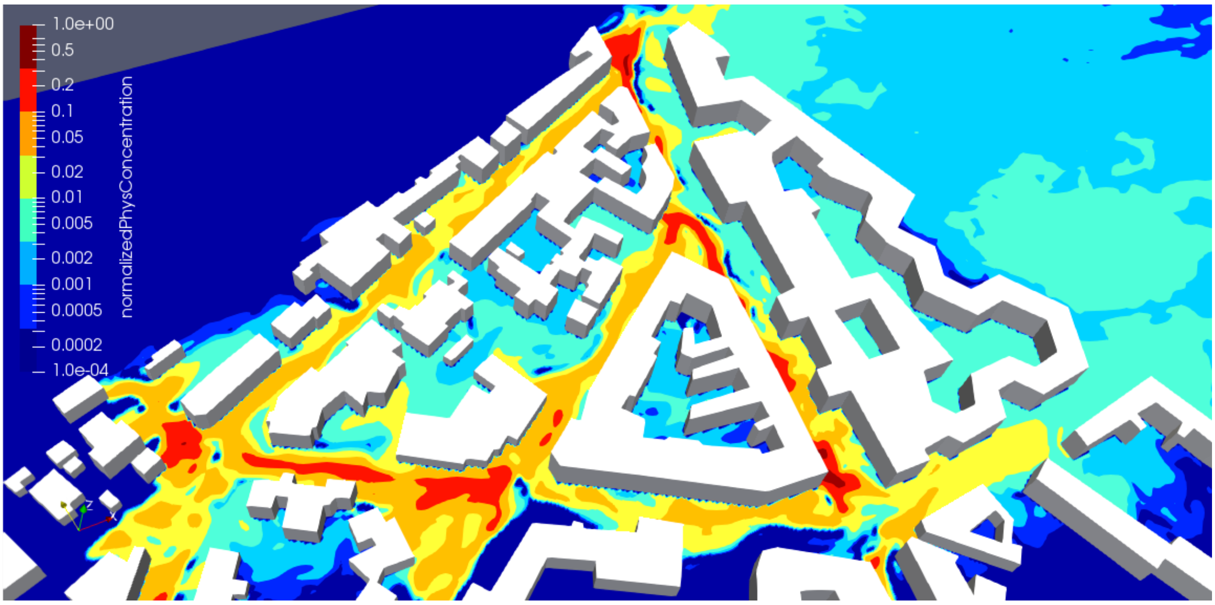

Figure 3: Floor level ($z = 1$ m) instantaneous pollutant concentration level (log scale)

6 CONCLUSION

We successfully demonstrated that the emissions model developed at IFPEN can be coupled with a CFD solver to perform sufficiently accurate and efficient pollutant dispersion in urban areas at the local scale. It is important to keep in mind that the aim was to build a complete modeling chain, the bricks of which could be replaced by more efficient elements in future studies (for instance, use a LBM solver with mesh refinement).

Further improvement is to be carried out as to how to keep track of the spatial and temporal variability of the traffic density on the road, which would lead to a more accurate representation of the emissions dynamics in unsteady simulations. The concentration level at particular locations will be chosen as a quantity of interest to evaluate the impact of aleatory uncertainties associated to the inlet wind intensity and orientation and to traffic-related parameters such as the fleet composition β or the global traffic flow q.

REFERENCES

[1] Shiyi Chen and Gary Doolen. Lattice boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30:329–364, 2003.

[2] Lucie Merlier, Jerome Jacob, and Pierre Sagaut. Lattice-boltzmann large-eddy simulation of pollutant dispersion in street canyons including tree planting effects. *Atmospheric Environment*, 195, 2018.

[3] Lucie Merlier, Jerome Jacob, and Pierre Sagaut. Lattice-boltzmann large-eddy simulation of pollutant dispersion in complex urban environment with dense gas effect: Model evaluation and flow analysis. *Building and Environment*, 148, 2019.

[4] Elisa Buffa, Jerome Jacob, and Pierre Sagaut. Lattice-boltzmann-based large-eddy simulation of high-rise building aerodynamics with inlet turbulence reconstruction. *Journal of Wind Engineering and Industrial Aerodynamics*, 212:104560, 2021.

[5] Stephan Lenz, Martin Geier, Manfred Krafczyk, Andrea Pasquali, Andreas Christen, Marco Giometto, and Martin Schönherr. Towards real-time simulation of turbulent air flow over a resolved urban canopy using the cumulant lattice boltzmann method on a gpgpu. *Journal of Wind Engineering and Industrial Aerodynamics*, 189:151–162, 2019.

[6] Anna Wellmann. Entwicklung und integration eines verkehrsmodells in eine strömungssimulation zur ermittlung von schadstoffausbreitungen im urbanen raum. *31. Forum Bauinformatik*, 2019.

[7] Nurul Ahmad, Atsushi Inagaki, Manabu Kanda, Naoyuki Onodera, and Takayuki Aoki. Large-eddy simulation of the gust index in an urban area using the lattice boltzmann method. *Boundary-Layer Meteorology*, 163, 2017.

[8] Atsushi Inagaki, Manabu Kanda, Nurul Ahmad, Ayako Yagi, Naoyuki Onodera, and Takayuki Aoki. A numerical study of turbulence statistics and the structure of a spatially-developing boundary layer over a realistic urban geometry. *Boundary-Layer Meteorology*, 164, 2017.

[9] Naoyuki Onodera, Yasuhiro Idomura, Yuta Hasegawa, Hiromasa Nakayama, Takashi Shimokawabe, and Takayuki Aoki. Real-time tracer dispersion simulations in oklahoma city using the locally mesh-refined lattice boltzmann method. *Boundary-Layer Meteorology*, 179, 2021.

[10] José Huertas and Daniel Prato. Cfd modeling of near-roadway air pollution. *Environmental Modeling & Assessment*, 25, 2020.

[11] Jiri Pospisil, Jiří Huzlík, Roman Licbinsky, and Michal Špiláček. Dispersion characteristics of pm10 particles identified by numerical simulation in the vicinity of roads passing through various types of urban areas. *Atmosphere*, 11:454, 2020.

[12] Zhengtong Li, Jie Xu, Tingzhen Ming, Chong Peng, Jiayue Huang, and Tingrui Gong. Numerical simulation on the effect of vehicle movement on pollutant dispersion in urban street. *Procedia Engineering*, 205:2303–2310, 2017.

[13] Huw Woodward, Marc Stettler, Dimitrios Pavlidis, Elsa Aristodemou, Helen ApSimon, and Christopher Pain. A large eddy simulation of the dispersion of traffic emissions by moving vehicles at an intersection. *Atmospheric Environment*, 215:116891, 2019.

[14] Tom Grylls, Clémence Le Cornec, Pietro Salizzoni, Lionel Soulhac, Marc Stettler, and Maarten van Reeuwijk. Evaluation of an operational air quality model using large-eddy simulation. *Atmospheric Environment: X*, 3:100041, 2019.

[15] Laëtitia Thouron, Youngseob Kim, Bertrand Carissimo, Christian Seigneur, and B. Bruge. Intercomparison of two modeling approaches for traffic air pollution in street canyons. *Urban Climate*, 27:163–178, 2019.

[16] Laurent Thibault, Philippe Dégeilh, Olivier Lepreux, Luc Voise, G. Alix, and Gilles Corde. *A new GPS-based method to estimate real driving emissions.* 2016.

[17] Mathias J. Krause, Adrian Kummerländer, Samuel J. Avis, Halim Kusumaatmaja, Davide Dapelo, Fabian

Klemens, Maximilian Gaedtke, Nicolas Hafen, Albert Mink, Robin Trunk, Jan E. Marquardt, Marie-Luise Maier, Marc Haussmann, and Stephan Simonis. Openlb—open source lattice boltzmann code. *Computers & Mathematics with Applications*, 81:258–288, 2021.

[18] Jonas Latt. *Hydrodynamic limit of lattice Boltzmann equations.* PhD thesis, 2007.

[19] Martin Geier, Martin Schönherr, Andrea Pasquali, and Manfred Krafczyk. The cumulant lattice boltzmann equation in three dimensions: Theory and validation. *Computers & Mathematics with Applications*, 222, 2015.

[20] Jerome Jacob and Pierre Sagaut. Wind comfort assessment by means of large eddy simulation with lattice boltzmann method in full scale city area. *Building and Environment*, 139, 2018.

[21] Orestis Malaspinas. Increasing stability and accuracy of the lattice boltzmann scheme: recursivity and regularization. 2015.

[22] Emmanuel Lévêque, Federico Toschi, Liang Shao, and Jean-Pierre Bertoglio. Shear-improved smagorinsky model for large-eddy simulation of wall-bounded turbulent flows. *Journal of Fluid Mechanics*, 570:491–502, 2007.

[23] Hui Xu and Pierre Sagaut. Analysis of the absorbing layers for the weakly-compressible lattice boltzmann schemes. *Journal of Computational Physics*, 245, 2013.

[24] Xiaoyi He, Shiyi Chen, and Gary Doolen. A novel thermal model for the lattice boltzmann method in incompressible limit. *Journal of Computational Physics*, 146:282–300, 1998.

[25] Michael Junk. A finite difference interpretation of the lattice boltzmann method. *Numerical Methods for Partial Differential Equations*, 17, 2001.

[26] Raoyang Zhang, Hongli Fan, and Hudong Chen. A lattice boltzmann approach for solving scalar transport equations. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 369:2264–2273, 2011.

[27] Jonas Latt, Bastien Chopard, Orestis Malaspinas, Michel Deville, and Andreas Michler. Straight velocity boundaries in the lattice boltzmann method. *Physical Review E*, 77, 2008.

[28] M'hamed Bouzidi, Mouaouia Firdaouss, Mouaouia, Lallemand, and Pierre. Momentum transfer of a boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13:3452, 2001.

[29] N. Jarrin, Sofiane Benhamadouche, Dominique Laurence, and R. Prosser. A synthetic-eddy-method for generating inflow conditions for large-eddy simulation. *International Journal of Heat and Fluid Flow*, 27:585–593, 2006.

[30] Giovanni deNunzio, Mohamed Laraki, and Laurent Thibault. Road traffic dynamic pollutant emissions estimation: From macroscopic road information to microscopic environmental impact. *Atmosphere*, 12:53, 2020.

[31] Mohamed Laraki, Giovanni de Nunzio, and Laurent Thibault. *Vehicle speed trajectory estimation using road traffic and infrastructure information.* 2020.

[32] Christof Gromke. Codasc: a database for the validation of street canyon dispersion models. *The 15th International Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes (HARMO*, 2013.

[33] Joseph Chang and Steven Hanna. Air quality model performance evaluation. *Meteorology and Atmospheric Physics*, 87:167–196, 2004.