

# Bachelor Thesis

Mikael Wijaya

January 27, 2026

# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduction</b>   | <b>3</b> |
| 1.1      | Background . . . . .  | 3        |
| 1.2      | Objective . . . . .   | 4        |
| <b>2</b> | <b>Methodology</b>  | <b>5</b> |
| 2.1      | Lattice Boltzmann Method . . . . .                                  | 5        |
| 2.1.1    | Lattice Configuration and Accuracy . . . . .                        | 5        |
| 2.1.2    | Large Eddy Simulation and Turbulence Modeling . . . . .             | 6        |
| 2.1.3    | Implementation of Lagrangian Stochastic Model . . . . .             | 6        |
| 2.1.4    | Advanced Wall Boundary Conditions . . . . .                         | 7        |
| 2.1.5    | Subgrid-Scale Turbulent Kinetic Energy (SGS TKE) . . . . .          | 8        |
| 2.1.6    | Stochastic Integration and Stability . . . . .                      | 8        |
| 2.1.7    | Model Selection and Validation via Flat Plate Analysis . . . . .    | 8        |
| 2.2      | Validation and Numerical Workflow . . . . .                         | 9        |
| 2.2.1    | Benchmark Case: AIJ Case H . . . . .                                | 9        |
| 2.2.2    | Computational Implementation and Workflow . . . . .                 | 10       |
| 2.2.3    | Lagrangian Data Post-Processing and Statistical Reduction . . . . . | 11       |
| 2.3      | Simulation Parameters and Case Setup . . . . .                      | 11       |
| 2.3.1    | Data Processing and Validation Strategy . . . . .                   | 14       |
| 2.3.2    | Validation Metric Equations . . . . .                               | 14       |
| 2.4      | Factor of Two (FAC2) . . . . .                                      | 15       |
| 2.5      | Mean Absolute Error (MAE) . . . . .                                 | 15       |
| 2.6      | Root Mean Square Error (RMSE) . . . . .                             | 15       |
| 2.7      | Reynolds Number . . . . .   | 15       |
| 2.8      | TKE ( $k$ ) . . . . .   | 15       |
| 2.9      | Concentration . . . . .   | 16       |

# 1 Introduction

## 1.1 Background

Air pollution in urban areas represents an ongoing global challenge that negatively impacts individual health, ecosystem integrity, and places increasing pressure on medical systems and economies worldwide. Urban air pollution comprises multiple components, including carbon dioxide ( $\text{CO}_2$ ), carbon monoxide (CO), and particulate matter with an aerodynamic diameter of less than  $2.5 \mu\text{m}$  ( $\text{PM}_{2.5}$ ).

The health implications are severe; recent data from the World Health Organization (2024) indicates that ambient air pollution contributes to approximately 4.2 million premature deaths globally each year. While mortality is largely attributed to increased private vehicle demand and population growth, a significant contributing factor is the lack of consideration for pollutant dispersion during rapid urban development (Li et al., 2021). As noted by Li et al. (2021), "hasty and irrational construction strategies" often lead to poor ventilation, trapping pollutants in high concentrations near the ground level.

Understanding and mitigating these concentrations requires a dual-level approach. First, it involves the utilization of mechanical factors, which include forced convection by wind, natural convection by solar radiation, and traffic-induced turbulence. Second, it requires the strategic design of urban morphology. This focus on the "sound design of the urban landscape" emphasizes building geometries, urban density, and enclosure degrees to improve the city's natural capacity to flush out pollutants (Li et al., 2021).

The morphology and organization of urban areas directly influence urban traffic patterns and pollutant flow dynamics. Therefore, understanding the relationship between urban morphology, pollutant transport, and air quality is critical for developing effective mitigation strategies that move beyond qualitative observations toward quantitative air quality improvements.

## 1.2 Objective

## 2 Methodology

### 2.1 Lattice Boltzmann Method

To simulate urban air pollutant transport, this study employs the Lattice Boltzmann Method (LBM), a mesoscopic computational fluid dynamics (CFD) technique. Unlike traditional solvers that directly discretize the macroscopic Navier-Stokes equations, LBM models fluid flow by simulating the collective behavior of a distribution of particles on a discrete grid, or "lattice" (Krüger et al., 2017). The movement of these particles is governed by the Lattice Boltzmann Equation (LBE), which describes the evolution of the discrete-velocity distribution function  $f_i$ :

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = \Omega_i(f) + S_i \quad (1)$$

Where  $f_i$  is the distribution function representing the density of particles moving with velocity components  $\mathbf{c}_i$  in the  $i$ -th direction. The term  $\Omega_i$  represents the collision operator, which models the internal redistribution of particle momentum, while  $S_i$  is the source term. The inclusion of  $S_i$  is critical for this research as it allows for the integration of external body forces and the modeling of mass sources, such as vehicle emissions. The simulation proceeds through a sequence of two primary operations: streaming and collision. During the streaming step, particles move from their current node  $\mathbf{x}$  to adjacent points based on their respective velocities. Following this, the collision step redistributes the particle velocities at each node based on local interactions, effectively mimicking fluid viscosity and relaxation toward equilibrium. As noted by Krüger et al. (2017), LBM offers several distinct advantages for simulating complex urban environments. First, the underlying Boltzmann equation is essentially a hyperbolic advection equation where the source terms depend only on local values of  $f$ . Because these operations rely strictly on information from a node and its immediate neighbors, the algorithm is inherently local. This locality facilitates high-performance parallelization, which is essential for large-scale urban simulations. Furthermore, the grid-based nature of LBM allows for the efficient handling of complex building geometries and irregular urban morphologies without the need for the intensive re-meshing often required by finite-element methods.

#### 2.1.1 Lattice Configuration and Accuracy

The choice of the D3Q27 lattice model is fundamental to the accuracy of the turbulence modeling in this research (Yokouchi et al., 2020). While the simpler D3Q19 model is often used to reduce computational costs, it lacks the rotational invariance required to capture complex secondary flows accurately. As demonstrated by Suga et al. (2015), the D3Q27 lattice provides a superior representation of turbulent structures. Their comparative analysis of turbulent pipe flow showed that the D3Q27 model produces significantly more realistic flow fields than the D3Q19 model, which tends to struggle with resolving fine-scale turbulent fluctuations in three-dimensional space. By utilizing 27 discrete velocities, the model can more effectively resolve the secondary circulations and high-frequency fluctuations necessary for an in-depth understanding of the pollutant dispersion mechanism in urban areas. The internal redistribution of particle momentum is governed by the Bhatnagar-Gross-Krook (BGK) collision operator. This operator assumes that the particle populations relax toward a local equilibrium state,  $f_i^{eq}$ , at a rate controlled by a single relaxation time  $\tau$  (Krüger et al., 2017):

$$\Omega_i(f) = -\frac{f_i - f_i^{eq}}{\tau} \Delta t$$

The equilibrium distribution  $f_i^{eq}$  is formulated as a second-order expansion of the Maxwell-Boltzmann distribution in terms of the macroscopic density  $\rho$  and velocity  $\mathbf{u}$ :

$$f_i^{eq}(\mathbf{x}, t) = w_i \rho \left( 1 + \frac{\mathbf{u} \cdot \mathbf{c}_i}{c_s^2} + \frac{(\mathbf{u} \cdot \mathbf{c}_i)^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right)$$

Here,  $c_s$  represents the isothermal speed of sound, defined as  $c_s^2 = \frac{1}{3} \frac{\Delta x^2}{\Delta t^2}$ . This formulation ensures the local conservation of mass and momentum while establishing the relationship between pressure and density as  $p = c_s^2 \rho$ .

### 2.1.2 Large Eddy Simulation and Turbulence Modeling

To capture the transient, multi-scale nature of urban wind flows, this research utilizes a Large Eddy Simulation (LES) approach. The unresolved subgrid-scale (SGS) motions are accounted for using the Coherent-structure Smagorinsky Model (CSM). In a standard Smagorinsky framework, the turbulent viscosity  $\nu_t$  is determined by the strain-rate tensor  $|S|$  and a filter width  $\Delta$ :

$$\nu_t = (C_S \Delta)^2 |S|$$

Within the LBM framework, the total viscosity  $\nu$  (the sum of molecular and turbulent viscosity) is directly linked to the relaxation time:

$$\nu = c_s^2 \left( \tau - \frac{\Delta t}{2} \right)$$

While a standard Smagorinsky constant  $C_S$  typically ranges between 0.1 and 0.2 (Blazek, 2015), the CSM used here employs a dynamic parameter based on the second invariant of the velocity gradient tensor. This approach allows the model to automatically account for wall-damping effects, ensuring more accurate flow predictions near building surfaces without requiring manual damping functions (Phuc et al., 2018).

### 2.1.3 Implementation of Lagrangian Stochastic Model

To simulate the transport and dispersion of particulate matter within the urban environment, this study utilizes a Lagrangian Stochastic Model (LSM). Unlike Eulerian methods that treat pollutants as a continuous concentration field, the LSM tracks the trajectories of thousands of discrete "passive" particles within the turbulent flow field generated by the LBM. This approach is particularly effective for modeling dispersion in the inhomogeneous and non-Gaussian turbulence characteristic of the urban planetary boundary layer (PBL). By simulating a statistically significant ensemble of particle paths, the ensemble-mean concentration can be derived, which is directly proportional to the particle number density or the probability density function (PDF) of particle positions (Weil et al., 2004). The trajectory of each particle is determined by integrating its position  $x_p$  over time. For a particle starting at an initial position  $x_0$ , the position at the next time step  $t + \Delta t$  is calculated as:

$$x_p(x_0, t + \Delta t) = x_p(x_0, t) + u_p(x_0, t) \Delta t$$

where  $u_p$  represents the instantaneous Lagrangian particle velocity. This velocity is composed of two primary components:

$$u_p = u_r + u_s$$

In this formulation,  $u_r$  is the resolved grid-scale (GS) velocity obtained directly from the LBM-LES results, representing the large-scale atmospheric motions. The term  $u_s$  represents the subgrid-scale (SGS) velocity fluctuations, which account for the smaller, unresolved turbulent motions. The resolved velocity  $u_r$  at the particle's location is determined by interpolating the LBM grid results using the weighting factors  $w_i$  of the neighboring nodes:

$$u_r = \sum_{i=0}^7 (w_i U_i)$$

To account for the unresolved fluctuations, the SGS velocity  $u_s$  is modeled as a stochastic process. Following the framework established by Weil et al. (2004) and Thomson (1987), the evolution of the SGS velocity  $du_s$  is described by a stochastic differential equation:

$$du_s = -\frac{3f_s C_0 \epsilon}{4} dt + \frac{1}{2} \left( \frac{1}{e_s} \frac{de_s}{dt} u_s + \frac{2}{3} \nabla e_s \right) dt + (f_s C_0 \epsilon)^{1/2} d\xi$$

where  $e_s$  is the local SGS turbulent kinetic energy (TKE) and  $f_s$  represents the mean contribution of the SGS TKE to the total TKE. The term  $\epsilon$  denotes the total dissipation rate, while  $C_0$  is a universal constant (set to 4.0 in this study). The stochastic component  $(f_s C_0 \epsilon)^{1/2} d\xi$  incorporates Gaussian white noise via the displacement vector  $d\xi$ , allowing the model to represent the random nature of sub-lattice turbulent dispersion.

#### 2.1.4 Advanced Wall Boundary Conditions

In LBM, the standard Bounce-Back (BB) scheme assumes that fluid particles hitting a solid boundary reverse their momentum, effectively enforcing a no-slip condition ( $\mathbf{u} = 0$ ) at the wall. While numerically stable and simple to implement, the BB scheme requires the first grid point to reside within the viscous sublayer to accurately capture wall shear stress. In urban-scale simulations, the grid resolution is typically much larger than the thickness of this sublayer, causing the BB scheme to underpredict shear drag. To resolve this, a Wall Function Boundary (WFB) is implemented, following the approach evaluated by Han et al. (2021). The WFB utilizes Spalding's Law to model the non-linear velocity profile in the logarithmic layer, allowing the simulation to approximate near-wall velocity without fully resolving the thin viscous region. This modification is applied during the collision step by introducing a momentum loss term based on the wall shear stress  $\tau_w$ :

$$f_a^* = f_a \pm \frac{\Delta t}{2\Delta y} \tau_{w,j}$$

where  $\tau_{w,j}$  is the shear stress in the  $j$ -direction ( $x$  or  $z$ ). The wall shear stress is determined by solving Spalding's implicit equation:

$$y^+ = u^+ + e^{-\kappa B} \left[ e^{\kappa u^+} - 1 - (\kappa u^+) - \frac{(\kappa u^+)^2}{2} - \frac{(\kappa u^+)^3}{6} \right]$$

Here,  $y^+$  is the dimensionless distance from the wall, and  $u^+$  is the dimensionless velocity. By calculating the friction velocity  $u_\tau = \sqrt{\tau_w/\rho}$  from this relationship, the model applies a "reverse resistance" that decelerates particles near building surfaces. As demonstrated by Han et al. (2021), this WFB approach significantly improves the accuracy of shear drag predictions in high-Reynolds-number urban flows where coarse grids are a necessity.

### 2.1.5 Subgrid-Scale Turbulent Kinetic Energy (SGS TKE)

The calculation of the SGS TKE ( $e_s$ ) is vital for the Lagrangian Stochastic Model, as it dictates the magnitude of the stochastic velocity fluctuations. While traditional LES often parameterizes  $e_s$  via the eddy viscosity and a mixing length ( $v_t = c_k l e_s^{1/2}$ ), this study utilizes a filtering approach as implemented by Suga et al. (2015) and Yokouchi et al. (2020). The SGS TKE ( $k_{sgs}$ ) is derived from the difference between the instantaneous grid-scale velocity and a filtered velocity field:

$$k_{sgs} = C_{kes} \sum_{i=1}^3 (\langle \hat{U}_i \rangle - \langle U_i \rangle)^2$$

The filtered velocity  $\langle \hat{U}_i \rangle$  is calculated using a weighted average of the central node (50% weight) and its six immediate neighbors (1/12 weight each):

$$\langle \hat{U}_i \rangle = \frac{1}{2} \langle U_i \rangle + \frac{\langle U_i^E \rangle + \langle U_i^W \rangle + \langle U_i^N \rangle + \langle U_i^S \rangle + \langle U_i^T \rangle + \langle U_i^B \rangle}{12}$$

This local filtering operation is highly compatible with the LBM's neighbor-only communication pattern, maintaining the algorithm's high parallel efficiency while providing a robust estimate of local turbulence intensity.

### 2.1.6 Stochastic Integration and Stability

The stochastic component of the LSM involves the integration of Gaussian white noise,  $\xi(t)$ , which is formally defined as the time derivative of a Wiener process  $W(t)$ :

$$W_i(t) = \int_0^t \xi(s) ds$$

In accordance with Brownian motion theory, the variance of this process increases linearly with time. Given that the SGS velocity  $u_s$  is modeled as a random walk, the stochastic term  $(f_s C_0 \epsilon)^{1/2} d\xi$  can occasionally produce unrealistically high instantaneous velocities. This is particularly prevalent near building walls where  $e_s$  is high and the gradient terms in the LSM equation are steep. To prevent numerical instability and maintain physical consistency, a velocity limitation is enforced, ensuring that the total Lagrangian particle velocity  $u_p$  remains within physically plausible bounds relative to the local mean flow Yokouchi et al., 2020.

### 2.1.7 Model Selection and Validation via Flat Plate Analysis

Before applying the model to complex urban morphologies, Yokouchi et al. (2020) conducted a validation study using particle dispersion over a flat plate to identify the most accurate combination of numerical schemes. The evaluation compared three primary configurations: (1) a baseline model using a simple Bounce-Back (BB) boundary and the standard subgrid-scale (SGS) model; (2) a model incorporating the Wall Function (WF) and Suga's refined SGS TKE calculation; and (3) an integrated model combining the WF-Suga framework with the Lagrangian Stochastic Model (LSM) and a velocity limitation. The accuracy of these models was evaluated based on their ability to replicate the theoretical logarithmic profile of particle

concentration in a neutral boundary layer:

$$\frac{C_0 - C}{C^*} = \frac{1}{\kappa} \ln \frac{z}{z_0}$$

where  $C_0$  represents the particle density at ground level,  $C^*$  is the friction density,  $\kappa$  is the von Kármán constant, and  $z_0$  is the roughness length. The results demonstrated that the WF-Suga-LSM model provided the highest correlation with the logarithmic profile. In contrast, models without the LSM or the specialized SGS TKE tended to deviate from the expected distribution, particularly in the near-wall region. Furthermore, the necessity of the speed limitation—defined as  $|u_s| < k_{sgs}^{1/2}$ —was confirmed through an analysis of the particle velocity variance. Without this limitation, the stochastic term in the LSM produced unrealistically high velocity fluctuations near the boundary. Implementing the speed limit successfully reduced this variance, ensuring a stable and physically consistent velocity distribution across the vertical profile. By adopting this validated WF-Suga-LSM framework with speed limitations, this research ensures that the fundamental transport mechanisms are correctly modeled before introducing the complexities of the AIJ Case H urban block.

## 2.2 Validation and Numerical Workflow

### 2.2.1 Benchmark Case: AIJ Case H

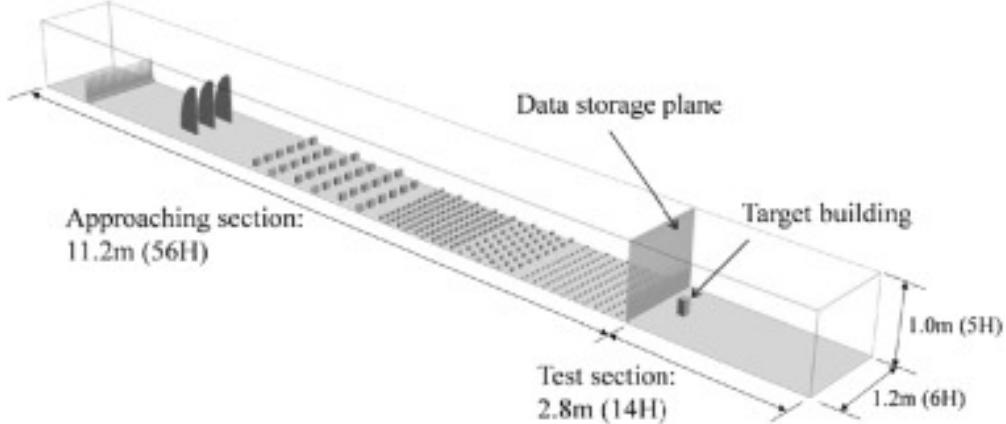


Figure 1: Experimental setup for AIJ Case H at Tokyo Polytechnic University, isolated building model and the arrangement of spires and roughness elements in the approach section to induce a turbulent boundary layer (adapted from Architectural Institute of Japan, 2021; Okaze et al., 2021).

The numerical validation of the Lattice Boltzmann Method (LBM) in this study is based on Case H of the Architectural Institute of Japan (AIJ) benchmark, which involves an isolated building model within a turbulent boundary layer. The physical experiment was conducted at Tokyo Polytechnic University using a 1:250 scale model. The building is a rectangular prism with a height-to-width ratio of 1 : 1 : 2 ( $H = 0.20$  m). The wind tunnel test section measures  $2.8\text{ m} \times 1.2\text{ m} \times 1.0\text{ m}$ , corresponding to normalized dimensions of  $14H \times 6H \times 5H$ . Turbulence in the approach section was induced using spires and roughness elements to simulate a realistic atmospheric boundary layer. For the pollutant dispersion study, a tracer gas release point was located 50 mm ( $0.25H$ ) from the leeward side of the building. To ensure comparability across different scales, the reference velocity  $U_H$  is defined as the mean inflow velocity at the building height. The

LBM results in this research are cross-validated against the comprehensive Large Eddy Simulation (LES) study conducted by Okaze et al. (2021). Their study utilized OpenFOAM to investigate the influence of various subgrid-scale (SGS) models—including the Standard Smagorinsky (S), Dynamic Smagorinsky (DS), Wall-Adapting Local Eddy-Viscosity (WALE), and Coherent Structure (CS) models—on turbulent statistics. Notably, Okaze et al. (2021) evaluated grid sensitivity across 10, 20, and 40-grid resolutions relative to the building width. This thesis focuses on comparing LBM performance against the high-resolution (40-grid) data and the quality acceptance criteria established by the AIJ, specifically aiming for a hit rate ( $q$ ) of  $\geq 0.66$  and a factor of two observations (FAC2) of  $> 0.5$ . By comparing the LBM-LSM results directly to these established OpenFOAM benchmarks, the predictive accuracy of the mesoscopic approach for urban pollutant dispersion can be rigorously quantified.

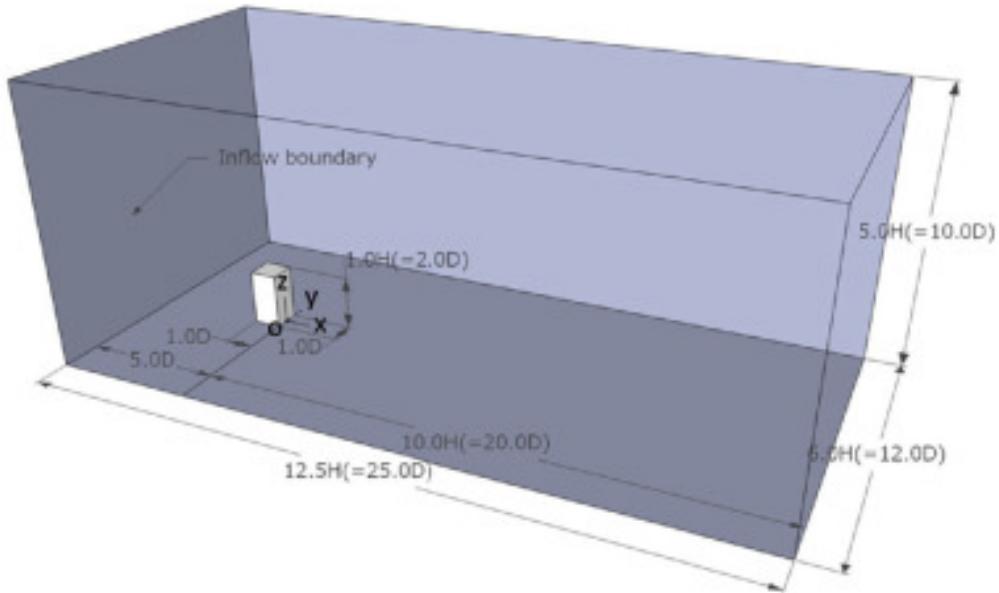


Figure 2: Numerical domain configuration and measurement locations for the LES validation study (adapted from Okaze et al., 2021).

### 2.2.2 Computational Implementation and Workflow

The simulation framework is a high-performance heterogeneous system specifically optimized for GPU execution. It follows a modular design architecture where the Eulerian wind solver (LBM) and the Lagrangian particle tracker (LPT) are coupled in a one-way data flow. Based on the system's build configuration and modular .mak files, the architecture is categorized into six functional pillars:

1. **Fluid Core:** Manages the D3Q27 Lattice Boltzmann kernels, including the GPU-accelerated streaming and collision operators and Sub-Grid Scale (SGS) turbulence modeling.
2. **Particle Engine:** A dedicated subsystem for Lagrangian tracking that handles particle generation, GPU-based advection, and inter-node MPI communication for particle data.
3. **Boundary and Solid Interface:** Resolves the interactions between the fluid phase and complex geometries (STL) or topography maps.

4. **State and Parameter Manager:** Handles the initialization of the MPI environment, domain decomposition, and fundamental fluid properties.
5. **Utility and Math Layer:** Provides the foundational logic for index mapping, memory allocation, and specialized LBM mathematical functions.
6. **I/O Subsystem:** Orchestrates data serialization into Paraview-compatible formats and custom user-defined results.

The operational workflow follows a parametric cycle that transitions from static configuration to high-performance execution. Initially, physical constants (such as surface roughness or latitude) and output sampling intervals are defined in the `Define_user.h` header, while topographic assets are specified within `paramFluidProperty.cu`.

During the compilation phase, the `Makefile` utilizes the NVIDIA CUDA Compiler (`nvcc`) to compile the modular source list into a binary targeted for the specific GPU architecture (e.g., `sm_90` for NVIDIA Hopper). This process "bakes" user-defined constants directly into the machine code, maximizing runtime performance. Finally, the execution is launched via the `runlbm.sh` script using `mpirun`, which passes dynamic arguments such as reference wind speed and memory allocation flags for the 30-million-particle Lagrangian array.

### 2.2.3 Lagrangian Data Post-Processing and Statistical Reduction

Due to the high temporal and spatial resolution of the Lagrangian tracking, the raw output is generated in a compact binary format to minimize I/O overhead. To translate these large-scale datasets into physically interpretable results, a dedicated post-processing suite was developed in C++. This suite follows a modular architecture, where specific physical quantities are computed by discrete analytical engines coordinated through a central execution framework. The post-processing workflow is orchestrated by a multi-threaded C++ executable, compiled via a modular `Makefile` that links specialized modules for density calculation, flux analysis, vertical profiling, and flux footprint prediction (FFP). The system is designed for high flexibility; rather than hard-coding simulation parameters, the tool utilizes a Command-Line Interface (CLI) where domain dimensions, grid size, time steps, and output intervals are passed as dynamic arguments at runtime. This allows the same analytical core to be applied across various experimental cases, ranging from local isolated building tests to large-scale urban simulations without the need for frequent re-compilation. Execution is managed through batch scripts (`run.bat`). For instance, the framework can simultaneously extract particle density profiles at specified heights ( $z_{out}$ ) and compute the source area contributions at defined sensor locations. By leveraging OpenMP for multi-threading (`OMP_NUM_THREADS`), the post-processor efficiently iterates through millions of Lagrangian trajectories stored in the binary files, aggregating them into averaged statistical summaries (CSV or text format) suitable for final visualization in Python.

## 2.3 Simulation Parameters and Case Setup

The specific configurations for the validation and research cases are summarized in Table 1. These cases represent the transition from local prototyping to large-scale supercomputer execution.

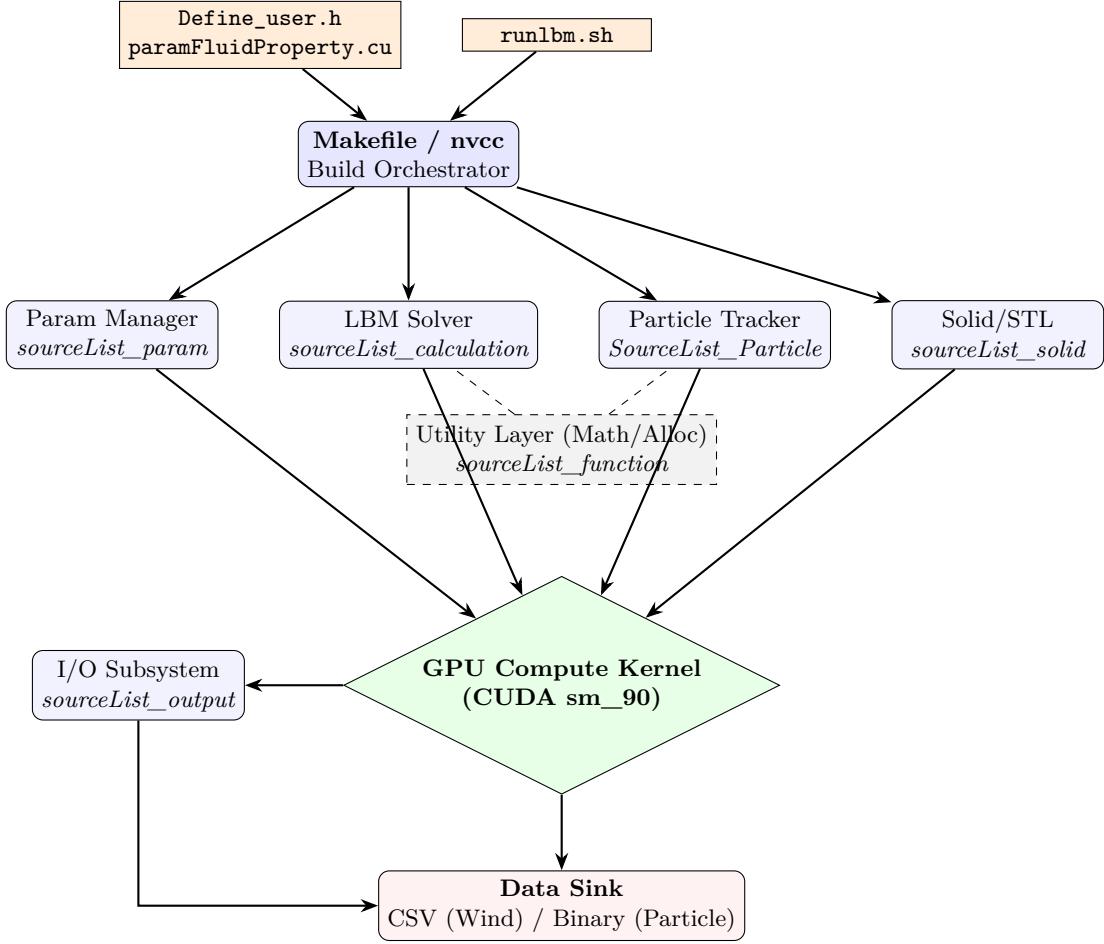


Figure 3: System architecture of the LBM simulation framework

Table 1: Detailed Simulation Case Parameters

| Case | Environment  | Domain Size ( $x, y, z$ in m) | Duration (s) | Approach | $H_{cube}$ (m) | $z_0$ (m) | $U_0$ (m/s) | Flow Rate |
|------|--------------|-------------------------------|--------------|----------|----------------|-----------|-------------|-----------|
| 1    | Local Server | $832 \times 384 \times 240$   | 2400         | No       | —              | 0.001     | 6.0         | 150 p/s   |
| 2    | TSUBAME 4.0  | $4352 \times 384 \times 320$  | 6000         | Yes      | 8              | 0.001     | 6.0         | 150 p/s   |
| 3    | TSUBAME 4.0  | $4352 \times 384 \times 320$  | 6000         | Yes      | 16             | 0.001     | 6.0         | 150 p/s   |
| 4    | TSUBAME 4.0  | $4352 \times 384 \times 320$  | 6000         | Yes      | 20             | 0.001     | 6.0         | 150 p/s   |
| 5    | TSUBAME 4.0  | $4352 \times 384 \times 320$  | 6000         | Yes      | 24             | 0.001     | 6.0         | 150 p/s   |
| 6    | TSUBAME 4.0  | $4352 \times 384 \times 320$  | 6000         | Yes      | 32             | 0.001     | 6.0         | 150 p/s   |
| 7    | TSUBAME 4.0  | $4352 \times 384 \times 320$  | 6000         | Yes      | 16             | 0.1       | 6.0         | 150 p/s   |
| 8    | TSUBAME 4.0  | $4352 \times 384 \times 320$  | 6000         | Yes      | 20             | 0.1       | 6.0         | 150 p/s   |

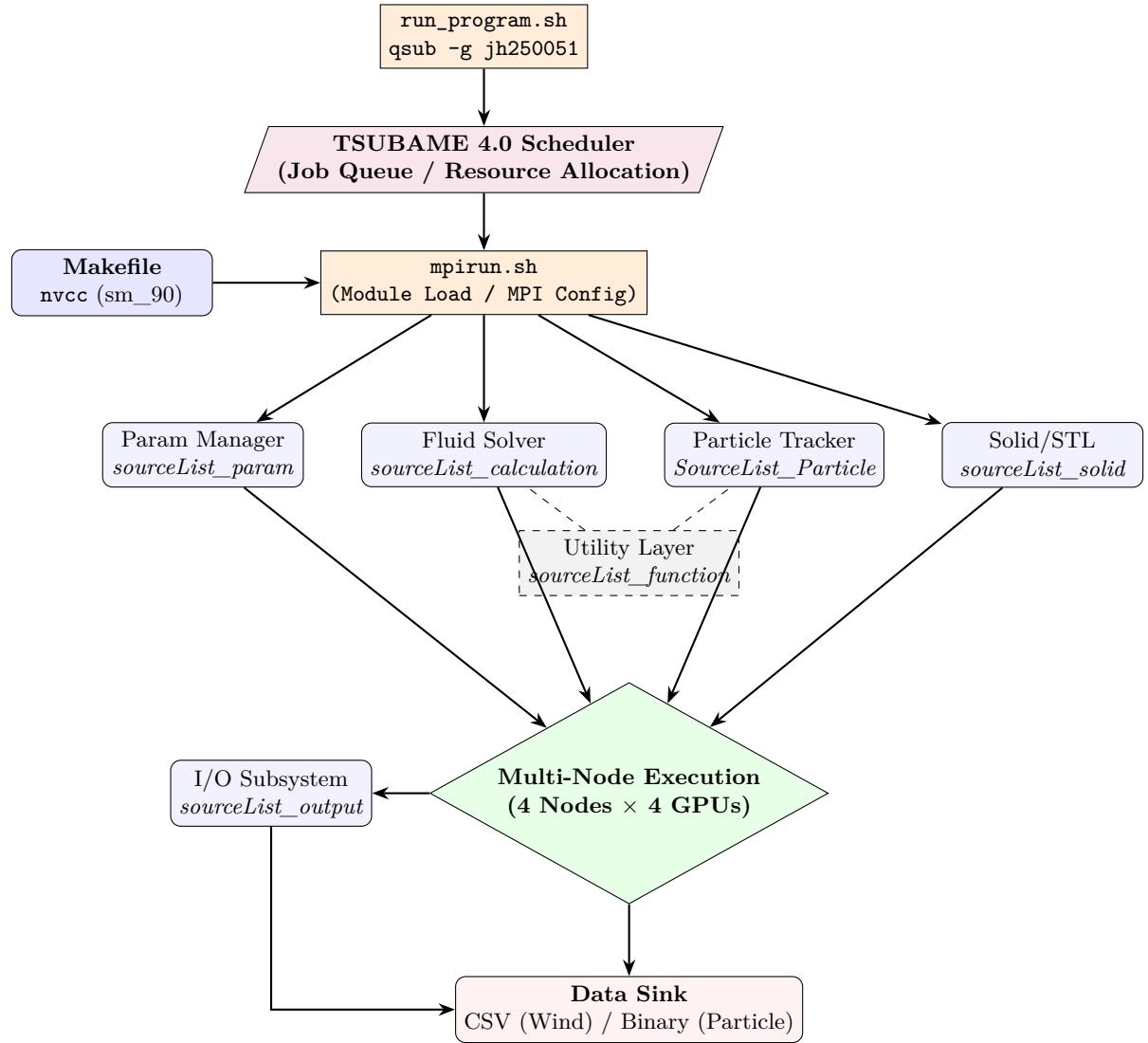


Figure 4: System architecture of the LBM simulation framework executed using the TSUBAME 4.0 Supercomputer

### 2.3.1 Data Processing and Validation Strategy

Table 2: Previous zemi, no approach.

| Metric               | N  | Hit Rate ( $q$ ) | FAC2  | MAE    | RMSE   | (D)  | (W)   |
|----------------------|----|------------------|-------|--------|--------|------|-------|
| $u_m/u_H$            | 90 | 0.578            | 0.756 | 0.1713 | 0.2431 | 0.25 | 0.050 |
| $v_m/u_H$            | 90 | 0.967            | 1.000 | 0.0178 | 0.0232 | 0.25 | 0.050 |
| $w_m/u_H$            | 90 | 0.289            | 0.311 | 0.0980 | 0.1165 | 0.25 | 0.050 |
| $u_{\text{rms}}/u_H$ | 90 | 0.511            | 0.644 | 0.0783 | 0.0995 | 0.50 | 0.005 |
| $v_{\text{rms}}/u_H$ | 90 | 0.544            | 0.644 | 0.0572 | 0.0680 | 0.50 | 0.005 |
| $w_{\text{rms}}/u_H$ | 90 | 0.322            | 0.678 | 0.0581 | 0.0650 | 0.50 | 0.005 |
| TKE/ $u_H^2$         | 90 | 0.222            | 0.600 | 0.0281 | 0.0321 | 0.50 | 0.005 |

Table 3: 2 km approach. Calculated  $U_H$  for normalization: 3.4744 m/s.

| Metric               | N  | Hit Rate ( $q$ ) | FAC2  | MAE    | RMSE   | (D)  | (W)   |
|----------------------|----|------------------|-------|--------|--------|------|-------|
| $u_m/u_H$            | 90 | 0.856            | 0.944 | 0.0706 | 0.1392 | 0.25 | 0.050 |
| $v_m/u_H$            | 90 | 0.933            | 1.000 | 0.0215 | 0.0264 | 0.25 | 0.050 |
| $w_m/u_H$            | 90 | 0.489            | 0.578 | 0.0565 | 0.0686 | 0.25 | 0.050 |
| $u_{\text{rms}}/u_H$ | 90 | 1.000            | 1.000 | 0.0313 | 0.0396 | 0.50 | 0.005 |
| $v_{\text{rms}}/u_H$ | 90 | 1.000            | 1.000 | 0.0426 | 0.0519 | 0.50 | 0.005 |
| $w_{\text{rms}}/u_H$ | 90 | 0.922            | 1.000 | 0.0375 | 0.0470 | 0.50 | 0.005 |
| TKE/ $u_H^2$         | 90 | 0.689            | 0.978 | 0.0255 | 0.0321 | 0.50 | 0.005 |

Table 4: 3 km approach. Calculated  $U_H$  for normalization: 4.0868 m/s.

| Metric               | N  | Hit Rate ( $q$ ) | FAC2  | MAE    | RMSE   | (D)  | (W)   |
|----------------------|----|------------------|-------|--------|--------|------|-------|
| $u_m/u_H$            | 90 | 0.722            | 0.878 | 0.0871 | 0.1563 | 0.25 | 0.050 |
| $v_m/u_H$            | 90 | 0.956            | 0.978 | 0.0221 | 0.0278 | 0.25 | 0.050 |
| $w_m/u_H$            | 90 | 0.544            | 0.544 | 0.0609 | 0.0737 | 0.25 | 0.050 |
| $u_{\text{rms}}/u_H$ | 90 | 1.000            | 1.000 | 0.0424 | 0.0525 | 0.50 | 0.005 |
| $v_{\text{rms}}/u_H$ | 90 | 1.000            | 1.000 | 0.0363 | 0.0420 | 0.50 | 0.005 |
| $w_{\text{rms}}/u_H$ | 90 | 0.956            | 1.000 | 0.0370 | 0.0419 | 0.50 | 0.005 |
| TKE/ $u_H^2$         | 90 | 0.667            | 0.744 | 0.0213 | 0.0248 | 0.50 | 0.005 |

### 2.3.2 Validation Metric Equations

Here are the formal definitions for the validation metrics, where  $P_i$  is the predicted (LBM) value,  $O_i$  is the observed (AIJ) value, and  $N$  is the total number of data points.

**Hit Rate ( $q$ )** The fraction of data points where the relative error is within a tolerance  $D$  OR the absolute error is within a tolerance  $W$ .

$$q = \frac{1}{N} \sum_{i=1}^N n_i \quad (2)$$

where

$$n_i = \begin{cases} 1 & \text{if } \left| \frac{P_i - O_i}{O_i} \right| \leq D \quad \text{or} \quad |P_i - O_i| \leq W \\ 0 & \text{else} \end{cases} \quad (3)$$

## 2.4 Factor of Two (FAC2)

The fraction of data points where the ratio of prediction to observation is between 0.5 and 2.0, OR \*both\* the prediction and observation are below a low-value threshold  $W$ .

$$\text{FAC2} = \frac{1}{N} \sum_{i=1}^N n_i \quad (4)$$

where

$$n_i = \begin{cases} 1 & \text{if } 0.5 \leq \frac{P_i}{O_i} \leq 2.0 \quad \text{or} \quad (|P_i| \leq W \text{ and } |O_i| \leq W) \\ 0 & \text{else} \end{cases} \quad (5)$$

## 2.5 Mean Absolute Error (MAE)

The average of the absolute differences between prediction and observation.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |P_i - O_i| \quad (6)$$

## 2.6 Root Mean Square Error (RMSE)

The square root of the average of the squared differences. This metric penalizes large errors more heavily.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (P_i - O_i)^2} \quad (7)$$

## 2.7 Reynolds Number

$$Re = \frac{U \cdot L}{\nu} \quad (8)$$

$$Re_{exp} = 4.3 \cdot 10^4$$

$$Re_{LBM} = \frac{4.940 \cdot 64}{1.512 \cdot 10^{-5}} = 2.091 \cdot 10^7$$

Adapted to the experiment and simulation scenario, choosing building height  $H$  as the

$$Re = \frac{U_H \cdot H}{\nu} \quad (9)$$

## 2.8 TKE (k)

$$u_{rms}^2 = \bar{u}^2 - \bar{u}^2 \quad (10)$$

$$v_{rms}^2 = \bar{v}^2 - \bar{v}^2 \quad (11)$$

$$w_{rms}^2 = \bar{w}^2 - \bar{w}^2 \quad (12)$$

$$k = \frac{1}{2}(u_{rms}^2 + v_{rms}^2 + w_{rms}^2) \quad (13)$$

## 2.9 Concentration

$$C_{norm} = \frac{C_o \cdot U_H \cdot H^2}{Q} \quad (14)$$

## References

- Architectural Institute of Japan. (2021). *Guidebook for CFD Predictions of Urban Wind Environment: Benchmark Case H*. Retrieved January 23, 2026, from [https://www.aij.or.jp/jpn/publish/cfdguide/index\\_e.htm](https://www.aij.or.jp/jpn/publish/cfdguide/index_e.htm)
- Blazek, J. (2015). *Computational fluid dynamics: Principles and applications* (3rd). Elsevier. <https://doi.org/10.1016/C2014-0-03823-3>
- Han, M., Ooka, R., & Kikumoto, H. (2021). Effects of wall function model in lattice Boltzmann method-based large-eddy simulation on built environment flows. *Building and Environment*, 195, 107764. <https://doi.org/10.1016/j.buildenv.2021.107764>
- Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., & Viggen, E. M. (2017). *The lattice boltzmann method: Principles and practice*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-44649-3>
- Li, Z., Ming, T., Liu, S., Peng, C., de Richter, R., Li, W., Zhang, H., & Wen, C.-Y. (2021). Review on pollutant dispersion in urban areas-part a: Effects of mechanical factors and urban morphology. *Building and Environment*, 190, 107534. <https://doi.org/10.1016/j.buildenv.2020.107534>
- Okaze, T., Terada, A., Kono, T., & Kato, S. (2021). Large-eddy simulation of flow around an isolated building: A step-by-step analysis of influencing factors on turbulent statistics. *Building and Environment*, 198, 108021. <https://doi.org/10.1016/j.buildenv.2021.108021>
- Phuc, P. V., Inagaki, A., & Kanda, M. (2018). Evaluation of a coherent structure smagorinsky model for the lattice boltzmann method in complex urban environments. *Fluids*, 6(2), 32. <https://doi.org/10.3390/fluids3020032>
- Suga, K., Kuwata, Y., Takashima, K., & Chikasue, R. (2015). A D3Q27 multiple-relaxation-time lattice Boltzmann method for turbulent flows. *Computers & Mathematics with Applications*, 69(6), 518–531. <https://doi.org/10.1016/j.camwa.2015.01.011>
- Thomson, D. J. (1987). Criteria for the selection of stochastic models of particle trajectories in turbulent flows. *Journal of Fluid Mechanics*, 180, 529–556. <https://doi.org/10.1017/S0022112087001940>
- Weil, J. C., Sullivan, P. P., & Moeng, C.-H. (2004). The Lagrangian stochastic model for settling particles applied to a Large-Eddy Simulation boundary layer. *Journal of the Atmospheric Sciences*, 61(23), 2810–2834. <https://doi.org/10.1175/JAS-3302.1>
- World Health Organization. (2024, October 24). *Ambient (outdoor) air pollution*. World Health Organization. Retrieved January 15, 2026, from [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)
- Yokouchi, H., Inagaki, A., Kanda, M., & Onodera, N. (2020). Simulation of Lagrangian pollutant in Jakarta urban district using Lattice Boltzmann method [格子ボルツマン法によるジャカルタ都市街区のラグランジュ粒子状汚染物質シミュレーション]. *Journal of Japan Society of Civil Engineers, Ser. B1 (Hydraulic Engineering)*, 76(2), I\_253-I\_258. [https://doi.org/10.2208/jscejhe.76.2\\_I\\_253](https://doi.org/10.2208/jscejhe.76.2_I_253)