Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Fachbereich für Maschinenbau und Verfahrenstechnik
Lehrstuhl für Strömungsmechanik und Strömungsmaschinen
Prof. Dr.-Ing. Martin Böhle

Masterarbeit

Study and Implementation of Wall Modelling for Large Eddy Simulations of Turbulent Wall-Bounded Flow by Means of the Lattice Boltzmann Method

Studie und Implementierung von Wandmodellen für Large-Eddy-Simulationen turbulenter wandgebundener Strömungen mit der Lattice-Boltzmann-Methode

---

2025/015

---

von

Gagan Deep Prakash Matr.-Nr. 425919

Betreuer:

Prof. Dr.-Ing. Andreas Schneider

Kaiserslautern, Juli 2025

Abstract

This thesis presents the implementation of wall modelling with Cartesian grids for large-eddy simulations (LES) using the lattice Boltzmann method (LBM). Wall modelling enables the advantages of LBM to be utilised without the high computational costs caused by wall-resolved approaches. However, this poses challenges because the near-wall nodes lie in a wide range of $y+$ values for complex geometries, and the adjacent nodes are not perpendicular, as in the body-fitted mesh used in traditional solvers. These problems are addressed by a distribution reconstruction procedure, which consists of near-wall friction velocity interpolations for smooth and stable boundary behaviour with coarse grids. In addition, accurate explicit wall functions consistent with the turbulence model were implemented to reduce the computational effort while maintaining accuracy. After the accurate reconstruction of the distribution function using the interpolated quantities, near-wall turbulence modelling was enhanced using a hybrid RANS-LES model with a blending region to avoid the high dissipation problem of the LES close to the wall. The implemented code was verified for a turbulent channel flow case for different frictional Reynolds numbers, and showed good agreement with the DNS data in the near-wall region. This indicates that the wall modelling procedure is stable and accurate for very high Reynolds numbers and is suitable for extension to industry-level problems. This study provides a complete discussion of the required algorithms, common modelling errors, and further verifications required to optimise wall modelling in LBM.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

LBM	Lattice Boltzmann Method	–
LES	Large Eddy Simulation	–
RANS	Reynolds-Averaged Navier–Stokes	–
DNS	Direct Numerical Simulation	–
CFD	Computational Fluid Dynamics	–
IBM	Immersed Boundary Method	–
SRT	Single Relaxation Time	–
SGS	Subgrid-Scale	–
HRR	Hybrid Recursive Regularised	–
PDF	Probability Density Function	–
BGK	Bhatnagar–Gross–Krook	–

Latin Symbols

a	Flow acceleration
c_s	Lattice speed of sound
u	Velocity vector
u_τ	Friction velocity
y^+	Dimensionless wall-normal coordinate
l_{mix}	Mixing length (typically wall distance)
f_i	Particle distribution function
f_i^{eq}	Equilibrium distribution function
$f_i^{(1)}$	Non-equilibrium distribution function
n	Unit wall-normal vector
x, y, z	Spatial coordinates
t	Time
Δx, Δt	Grid spacing and time step
S_{ij}	Strain rate tensor components
K	Blending auxiliary function variable
U^+	Non-dimensional velocity (normalized by u_τ)

Greek Symbols

κ	von Kármán constant
ν	Kinematic viscosity
ν_T	Turbulent viscosity
ρ	Fluid density
β	RANS–LES blending function
α	Start-point factor of blending region
γ	Blending exponent
ϕ	Generic scalar field (for filtering)
ω	Relaxation parameter
Π	Momentum flux tensor
μ	Dynamic viscosity

1 Introduction

Computational Fluid Dynamics (CFD) is a powerful tool with enormous potential for enhancing the design of engineering systems involving fluid flow. It is also used to expand our understanding of many natural phenomena. Despite decades of method development and improvement, numerous challenges remain regarding the accuracy and computational efficiency of CFD methods in different applications [1]. In this study, we address the challenge of wall-bounded turbulence. Turbulence is ubiquitous in most technical applications involving high-speed or large-scale fluid flows, with Reynolds numbers often reaching the range of 10^7–10^9.

In this study, the focus was on the lattice Boltzmann method (LBM), as opposed to classical Navier-Stokes solvers. It has gained interest in aerodynamic simulations [2], particularly for acoustic and dynamic load predictions, owing to several advantages of the LBM, as stated below:

1. As an intrinsically time-dependent method, LBM is highly efficient for high-fidelity simulations such as Direct Numerical Simulations (DNS) and Large Eddy Simulations (LES).

2. It is easy to implement and highly parallelizable due to its node-local computations in the collision step.

3. High Reynolds number turbulent flows, including separated flows, have been successfully captured using LBM with LES, often outperforming conventional RANS simulations in accuracy.

4. Its inherently low numerical dissipation is advantageous for simulating long wakes and their downstream effects, as well as for aeroacoustics applications.

5. LBM solved on Cartesian grids and coupled with the Immersed Boundary Method (IBM), enables straightforward grid generation, avoiding the tedious process of generating body-fitted meshes manually, as required in traditional CFD solvers for complex geometries.

The final advantage using the IBM is of particular relevance to this study. Leveraging Cartesian grids for LES introduces new challenges that are not typically encountered with body-fitted meshes. In technically relevant turbulent flows, fully resolving the near-wall region becomes computationally prohibitive. The estimates presented in [3] and [1] demonstrate that the computational cost of wall-resolved LES is at least an order of magnitude greater than that of wall-modelled LES. This disparity highlights the importance of efficient wall modelling strategies, particularly in the context of high Reynolds number flows over complex geometries [4].

In comparison, Reynolds-averaged Navier–Stokes (RANS) solvers also require fine resolution in the wall-normal direction, typically targeting $y^+ \approx 1$ for wall-resolved simulations. However, a key distinction lies in the treatment of wall-parallel directions. In RANS, these directions can be discretised using significantly larger grid spacings, resulting in high-aspect-ratio cells near the wall. This flexibility reduces the overall computational burden, and consequently, the relative importance of wall models in the RANS framework is diminished. This flexibility of using high-aspect-ratio cells is unavailable on Cartesian grids, which evidently leads to a significant increase in the number of nodes and makes wall modelling very significant in this context.

IBM first proposed by Peskin [5], eliminates all the nodes of the Cartesian grid that are within the body and applies the effect of the wall on the boundary nodes (fluid nodes close to the wall) by introducing a force. This resulted in a diffuse interface where the wall force was continuously applied over several layers of nodes near the surface. However, it has been established that in the context of wall-bounded turbulence, directly applying the wall effect to the first boundary node yields accurate results, commonly referred to as sharp interface methods [6].

In this context, the incorporation of wall shear stress provided by the wall model is not straightforward to impose on the boundary node owing to the non-body-fitted nature of the Cartesian grids. Hence, special care must be taken depending on the solution method employed.

Various methods can be used to incorporate wall boundary conditions in the LBM-LES context. One commonly used approach is the direct Dirichlet velocity boundary condition at the boundary node. Here, the distribution function in the LBM is either fully reconstructed [7, 8] or partially interpolated using the macroscopic velocity prescribed by the wall function [9].

Several modifications have also been made to the standard no-slip bounce-back method to account for the unresolved near-wall behaviour. These include interpolated bounce-back schemes [10], which are adapted with linear velocity interpolation [11]. A comprehensive overview of the available options is provided in [12].

The number of different approaches indicates the complexity of applying wall modelling to LBM solvers. The reconstruction method with an advanced interpolation scheme introduced in [4] in combination with a RANS–LES hybrid approach for near-wall tur-

bulence modelling is the approach implemented here. This method has been shown to effectively capture attached wall-bounded turbulence at high Reynolds numbers.

The scope of this study is to briefly review the challenges and advantages of WM-LES with LBM and to implement a specific modelling technique for the turbulent channel flow case for validation. Implementation details that are often missing from the literature are also explained.

The contents of this thesis are as follows: The theoretical aspects of LBM and LES are briefly reviewed in Chapters 2 and 3. In Chapter 4, the theoretical justification for wall modelling is presented, along with the derivation of the specific wall function used in this study. Chapter 5 outlines the fully implemented algorithm using flowcharts. Chapter 6 provides verification results for turbulent channel flow, including comparisons with DNS data. Finally, Chapter 7 concludes the thesis by summarising the key outcomes and outlining directions for future improvements and extensions to a generalised, accurate WM-LES–LBM solver.

2 The Lattice Boltzmann Method

The Lattice Boltzmann Method (LBM) is a numerical technique for solving the Boltzmann equation, although it originated from the Lattice Gas Automata (LGA) [13].

This chapter provides an overview of the LBM concepts necessary for the understanding and execution of this thesis on wall models in LBM, following the work of [13]. First, we define the distribution function in the LBM context and then discuss the Boltzmann equation and its complexity in solving it. Then, the process of applying approximations and discretising the Boltzmann equation to arrive at the lattice Boltzmann equation is presented to provide an overview. Readers seeking mathematical details and historical background are referred to foundational literature such as [14] and [15].

2.1 Kinetic Theory and the Boltzmann Equation

The Boltzmann equation originates from the kinetic theory of gases where a statistical description of a group of gas molecules is given by a distribution function defined as

$$f(\mathbf{x}, \boldsymbol{\xi}, t) = \frac{\mathrm{d}N}{\mathrm{d}V \cdot \mathrm{d}\boldsymbol{\xi}}. \tag{1}$$

$f(\mathbf{x}, \boldsymbol{\xi}, t)$ is interpreted as the probability density of finding $\mathrm{d}N$ number of molecules in the six-dimensional phase-space volume $\mathrm{d}V \cdot \mathrm{d}\boldsymbol{\xi}$, where $\mathrm{d}V = \mathrm{d}x \cdot \mathrm{d}y \cdot \mathrm{d}z$ and $\mathrm{d}\boldsymbol{\xi} = \mathrm{d}u \cdot \mathrm{d}v \cdot \mathrm{d}w$. $\mathrm{d}V$ and $\mathrm{d}\boldsymbol{\xi}$ represent infinitesimally small regions in the physical and velocity spaces, respectively.

The evolution of this distribution function is governed by the Boltzmann equation, which is derived based on the conservation of molecules [16].

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \boldsymbol{\xi}} = \left(\frac{\partial f}{\partial t}\right)_{\mathrm{coll}}. \tag{2}$$

The left-hand side of the equation accounts for changes in time and space, whereas the right-hand side accounts for changes in the distribution caused by collisions [17]. Under the assumption that inter-molecular forces are negligible and binary collisions dominate, the Boltzmann's collision term is written as

$$\left(\frac{\partial f}{\partial t}\right)_{\text{coll}} = \int_{\boldsymbol{\xi}_1}\int_{A_C} \left(f' \cdot f_1' - f \cdot f_1\right) \left|\boldsymbol{\xi}_r\right| \mathrm{d}A_C \, \mathrm{d}\boldsymbol{\xi}_1 \tag{3}$$

This term accounts for the change in the particle distribution function $f(\mathbf{x}, \boldsymbol{\xi}, t)$ owing to binary collisions. Specifically, it models how particles with velocity $\boldsymbol{\xi}$ collide with others with velocity $\boldsymbol{\xi}_1$, represented by the distribution $f_1 = f(\mathbf{x}, \boldsymbol{\xi}_1, t)$. The term also includes post-collision distributions, denoted by a prime symbol $'$.

Additionally, the H-theorem [16] implies that, in a closed homogeneous system, every distribution tends to an equilibrium (Maxwellian) distribution over time, given by

$$f^{\text{eq}}(\mathbf{x}, \boldsymbol{\xi}, t) = \frac{n}{(2 \cdot \pi \cdot R \cdot T)^{3/2}} \cdot \exp\left(-\frac{|\boldsymbol{\xi} - \mathbf{u}|^2}{2 \cdot R \cdot T}\right). \tag{4}$$

The collision term in Equation 3 is complex and nonlinear, resulting in the need for simplification which was proposed by Bhatnagar, Gross, and Krook (BGK) in [18] as

$$\left(\frac{\partial f}{\partial t}\right)_{\text{coll}} = -\omega \cdot (f - f^{eq}) = -\frac{1}{\tau} \cdot (f - f^{eq}). \tag{5}$$

This is the BGK collision model, which approximates the collision process in the Boltzmann equation. ω is the collision frequency, and equivalently its reciprocal is the collision time τ.

2.2 The Lattice Boltzmann Equation

$$f(\mathbf{x}, \boldsymbol{\xi}, t) = m \cdot \frac{\mathrm{d}N}{\mathrm{d}V \cdot \mathrm{d}\boldsymbol{\xi}}. \tag{6}$$

As opposed to Eq. (1), the distribution function in the LBE has a molecular mass and is referred to as the density distribution function (Eq. (6)). Hereafter, it is referred to as the distribution function.

In order to solve the continuous Boltzmann equation (Eq. (2)) numerically over arbitrary geometries, both the equations and the computational domain must be discretised. The complex collision term in this integro-differential formulation is replaced by the Bhatnagar–Gross–Krook (BGK) approximation (Eq. (5)), which is then discretised in time and space to derive the Lattice Boltzmann Equation.

Discretisation in velocity space can be performed using the discrete velocity method or Hermite polynomial expansions, which are mathematically equivalent [19]. Additionally, the equilibrium distribution was expanded as a Taylor series and truncated at the second order to enable the application of appropriate quadrature rules. For the discretisation in space and time, the equation is integrated over a time interval, which is approximated using the trapezoidal rule.

After all the discretisation steps and some algebraic simplifications, this process leads to the *Lattice Boltzmann Equation* as follows:

$$f_i\left(\mathbf{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t\right) = f_i(\mathbf{x},t) + \Omega\left[f_i^{eq}(\mathbf{x},t) - f_i(\mathbf{x},t)\right]. \tag{7}$$

Here, the relaxation parameter Ω, which is a function of the kinematic fluid viscosity, is given by

$$\Omega = \frac{\Delta t}{\bar{\tau}} = \frac{\Delta t}{\tau + 0.5\,\Delta t} = \frac{c_s^2 \Delta t}{\nu + 0.5\,c_s^2 \Delta t}, \tag{8}$$

where $\bar{\tau}$ is the numerical relaxation time, τ is the physical relaxation time, ν is the kinematic viscosity, and c_s is the lattice speed of sound.

2.3 The Velocity Sets

Discretisation imposes a restriction on the velocity space, which is described by the lattices selected to compute a particular problem. These lattices are labelled according to the DdQq nomenclature [20] where q is the number of discrete particle velocities and d is the number of spatial dimensions. In this study, all verification cases were conducted using the D3Q19 lattice, as shown in the Figure.

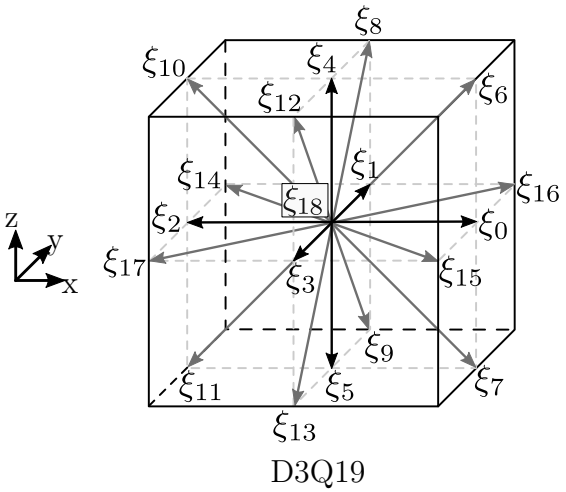


Figure 1: D3Q19 velocity set used for all simulations in this thesis.

Here, $\boldsymbol{\xi} = \frac{\Delta x}{\Delta t}$ is the *molecular velocity*, whose particle velocity vectors for the given lattice are

$$\xi_i = \begin{cases} (0,0,0) & i = 18 \\ (\pm\tilde{\xi},0,0),(0,\pm\tilde{\xi},0),(0,0,\pm\tilde{\xi}) & i = 0,\ldots,5 \\ (\pm\tilde{\xi},\pm\tilde{\xi},0),(\pm\tilde{\xi},0,\pm\tilde{\xi}),(0,\pm\tilde{\xi},\pm\tilde{\xi}) & i = 6,\ldots,17 \end{cases} \tag{9}$$

To obtain macroscopic physical quantities from these distribution functions $f(x,\xi,t)$, their moments must be calculated. In the continuous sense, these are absolute integrals of f multiplied by a function of ξ in the infinite velocity space, but in the discretised phase space, these are calculated as weighted sums using the Gauss-Hermite quadrature.

$$M(\boldsymbol{x},t) = \int_{-\infty}^{\infty} \varphi(\boldsymbol{\xi}) f(\boldsymbol{x},\boldsymbol{\xi},t)\mathrm{d}\boldsymbol{\xi} = \sum_{i=0}^{q-1} \varphi\left(\boldsymbol{\xi}_i\right) f_i$$
$$f_i = w_i f\left(\boldsymbol{x},\boldsymbol{\xi}_i,t\right) \tag{10}$$

The weights w_i are obtained such that an equivalence exists between the moments calculated on the continuous and discretised distribution functions f_i.

$$w_i = \begin{cases} \frac{1}{3}, & i = 18 \\ \frac{1}{18}, & i = 0,\ldots,5 \\ \frac{1}{36}, & i = 6,\ldots,17 \end{cases} \tag{11}$$

The isothermal speed of sound, which is $c_s = \frac{p}{\rho}$, is related to the molecular velocity ξ by $\xi = \sqrt{3}c_s$.

Equations 7 and 8 describe the BGK collision model, which applies the same relaxation rate to all physical processes, such as momentum diffusion. This makes the collision model inappropriate or highly unstable for very high Reynolds number flows, which clearly have processes occurring at different time scales. The study on wall models requires highly turbulent flows; hence, regularised collision models were used for greater numerical stability, which is briefly explained in the following section.

2.4 Regularised Collision Models

The following explanation of the Regularised Collision models is adapted from [21]. Expanding the discrete distribution function f_i as a perturbation series in powers of the Knudsen number about the local equilibrium $f_i^{(0)}$ yields the Chapman–Enskog expansion [22]:

$$f_i = f_i^{(0)} + \epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)} + \epsilon^3 f_i^{(3)} + \cdots = f_i^{(0)} + f_i^{\mathrm{neq}}. \tag{12}$$

Here, f_i^{neq} represents the sum of all deviations from equilibrium, and ϵ is the *smallness parameter* that reflects the relative order in the Knudsen number. It has been shown that retaining only the two lowest-order terms in the Taylor expansion of the Lattice Boltzmann Equation (Eq. 2) suffices to recover the Navier–Stokes equations, and thereby to reproduce macroscopic fluid behaviour.

In light of this, the non-equilibrium distribution is typically approximated by neglecting higher-order terms $\mathcal{O}(\epsilon^n)$ for $n > 1$ in Eq. 12, and computed as:

$$f_i^{\mathrm{neq}} = f_i - f_i^{(0)} \simeq f_i^{(1)}. \tag{13}$$

This form of explicit reconstruction of the non-equilibrium part $f_i^{(1)}$, that is, the regularisation procedure applied to the pre-collision distribution with an SRT collision operator, results in the following:

$$
\begin{aligned}
f_i\left(x+\xi_i\Delta t, t+\Delta t\right) - f_i^{\mathrm{REG}} &= \Omega_i^{\mathrm{REG}} \\
f_i^{\mathrm{REG}} &= f_i^{(0)} + f_i^{(1)} \\
\Omega_i^{\mathrm{REG}} &= \tilde{\omega} f_i^{(1)} \\
\Rightarrow f_i\left(x+\xi_i\Delta t, t+\Delta t\right) &= f_i^{(0)} + (1-\tilde{\omega}) f_i^{(1)}.
\end{aligned}
\tag{14}
$$

The first-order non-equilibrium part $f_i^{(1)}$ is calculated by the Hermite series expansion

$$
f_i^{(1)} = w_i \sum_{n=1}^{2} \frac{1}{c_s^{2n} n!} \mathcal{H}_i^{(n)} : \boldsymbol{A}_1^{(n)}. \tag{15}
$$

$\boldsymbol{A}_1^{(n)}$ is the non-equilibrium expansion coefficient, which is obtained by projecting the non-equilibrium functions onto the Hermite tensors.

$$
\begin{aligned}
A_{1,\alpha}^{(1)} &= \sum_i \mathcal{H}_{i,\alpha}^{(1)} \left(f_i - f_i^{(0)}\right) = \sum_i \xi_{i,\alpha} f_i^{neq} \\
A_{1,\alpha\beta}^{(2)} &= \sum_i \mathcal{H}_{i,\alpha\beta}^{(2)} \left(f_i - f_i^{(0)}\right) = \sum_i \left(\xi_{i,\alpha}\xi_{i,\beta} - \delta_{\alpha\beta} c_s^2\right) f_i^{neq}
\end{aligned}
\tag{16}
$$

As an improvement, it was further found that any higher non-equilibrium moments higher than two can be obtained, given the macroscopic velocity u and the second-order expansion coefficients. This results in the *recursive-regularisation* (RR) model proposed in [23].

$$
\begin{aligned}
f_i^{(1)} = w_i \Bigg[\sum_{n=1}^{2} \frac{1}{c_s^{2n} n!} \mathcal{H}_i^{(n)} : A_1^{(n)} &+ \frac{1}{2c_s^6}\left(\mathcal{H}_{i,xxy}^{(3)} + \mathcal{H}_{i,yzz}^{(3)}\right)\left(A_{1,xxy}^{(3)} + A_{1,yzz}^{(3)}\right) \\
&+ \frac{1}{2c_s^6}\left(\mathcal{H}_{i,xxz}^{(3)} + \mathcal{H}_{i,yyz}^{(3)}\right)\left(A_{1,xxz}^{(3)} + A_{1,yyz}^{(3)}\right) \\
&+ \frac{1}{2c_s^6}\left(\mathcal{H}_{i,xyy}^{(3)} + \mathcal{H}_{i,xzz}^{(3)}\right)\left(A_{1,xyy}^{(3)} + A_{1,xzz}^{(3)}\right) \\
&+ \frac{1}{6c_s^6}\left(\mathcal{H}_{i,xxy}^{(3)} - \mathcal{H}_{i,yzz}^{(3)}\right)\left(A_{1,xxy}^{(3)} - A_{1,yzz}^{(3)}\right) \\
&+ \frac{1}{6c_s^6}\left(\mathcal{H}_{i,xxz}^{(3)} - \mathcal{H}_{i,yyz}^{(3)}\right)\left(A_{1,xxz}^{(3)} - A_{1,yyz}^{(3)}\right) \\
&+ \frac{1}{6c_s^6}\left(\mathcal{H}_{i,xyy}^{(3)} - \mathcal{H}_{i,xzz}^{(3)}\right)\left(A_{1,xyy}^{(3)} - A_{1,xzz}^{(3)}\right)\Bigg].
\end{aligned}
\tag{17}
$$

Here, the third-order moments $A_{1,\alpha\alpha\beta}^{(3)}$ and $\mathcal{H}_{i,\alpha\alpha\beta}^{(3)}$ are obtained as follows

$$
\begin{aligned}
A_{1,\alpha\alpha\beta}^{(3)} &= 2u_\alpha A_{1,\alpha\beta}^{(2)} + u_\beta A_{1,\alpha\alpha'}^{(2)} \\
\mathcal{H}_{i,\alpha\alpha\beta}^{(3)} &= \left(\xi_{i,\alpha}\xi_{i,\alpha} - c_s^2\right)\xi_{i,\beta}.
\end{aligned}
\tag{18}
$$

Finally the hybrid-recursive regularized collision operator (HRR), introdcued in [24], hybridizes the calculation of the non-equilibrium second-order expansion coefficient $A_1^{(2)}$

(Eq. 16), which is related to the stress tensor as

$$A^{(2)}_{1,\alpha\beta} = -2\frac{\rho c_s^2 \Delta t}{\tilde{\omega}} S_{\alpha\beta}. \tag{19}$$

The hybridised calculation is as follows:

$$\boldsymbol{A}^{(2)}_1 = \sigma \boldsymbol{A}^{(2),\mathrm{PR}}_1 + (1-\sigma)\boldsymbol{A}^{(2),\mathrm{FD}}_1 \quad \text{where } [0 \leq \sigma \leq 1]. \tag{20}$$

Here, $\boldsymbol{A}^{(2),\mathrm{PR}}_1$ is the expansion coefficient calculated directly from the projection moments as given in Equation 16, and the $\boldsymbol{A}^{(2),\mathrm{FD}}_1$ is calculated by finite difference estimation of the $S_{\alpha\beta}$ in Equation 19 which results in the following:

$$\begin{aligned} A^{(2),\mathrm{FD}}_{1,\alpha\beta} = -\frac{\rho c_s^2 \Delta t}{\tilde{\omega}} \Bigg(& \frac{u_\alpha\left(x + e_\beta \Delta x\right) - u_\alpha\left(x - e_\beta \Delta x\right)}{2\Delta x} \\ & + \frac{u_\beta\left(x + e_\alpha \Delta x\right) - u_\beta\left(x - e_\alpha \Delta x\right)}{2\Delta x} \Bigg). \end{aligned} \tag{21}$$

When the hybridisation factor σ is chosen between 0 and 1, a numerical hyperviscosity is added to the first-order non-equilibrium part $f_i^{(1)}$ calculation, which dampens the non-hydrodynamic contributions, leading to enhanced stability.

The HRR collision model was used in this study for enhanced stability when dealing with high Reynolds number flows. For a comprehensive explanation of the recursive regularised (RR) and hybrid-recursive regularised (HRR) collision models, the reader is directed to the works of Malaspinas [23] and Jacob et al. [24].

2.5 Boundary Conditions

To use LBM for any physical problem, an accurate representation of the boundary conditions is required. Unlike conventional CFD, the macroscopic quantities defined at the boundary (e.g. velocity or pressure) must be converted to their distributions. This section explains the boundary conditions used in this study. These are the halfway bounce-back method and periodic boundary conditions. The wall model and immersed boundary condition are explained separately in Chapter 3. For a full summary of all the different boundary conditions in LBM, see [25].

2.5.1 Equilibrium Schemes

Equilibrium distributions as seen before, can be calculated directly from the macroscopic quantities (Eq. 4), which means they can directly be used to set boundary conditions. As in many cases, only a single variable (pressure or velocity) is available at the boundaries, and the unknown quantity is extrapolated from the neighbouring node within the fluid. This procedure is generally employed to establish the boundary conditions in the LBM.

This results in a boundary condition that is only accurate to the first order [25]. This

can be improved by reconstructing the non-equilibrium part using either extrapolation schemes or stress tensors [26]. This type of velocity or pressure boundary condition is not required for the body-force-driven channel flow considered in this study.

However, the reconstruction of equilibrium and non-equilibrium distributions will be dealt with in detail as a wall-boundary condition in Chapter 4.

2.5.2 Periodic Boundary Conditions

Periodic boundary conditions were implemented when the flow state was expected to be periodic. In this boundary condition, the flow exits from one boundary and re-enters through another, completely conserving mass and momentum. With this boundary condition, an external source of momentum is always required to drive the flow; otherwise, the flow will subside to a homogeneous state.

Periodic boundary conditions are necessary because the validation of the wall model requires a fully developed channel flow with isotropic turbulence. Figure 2 shows an illustrative example of a periodic boundary condition.

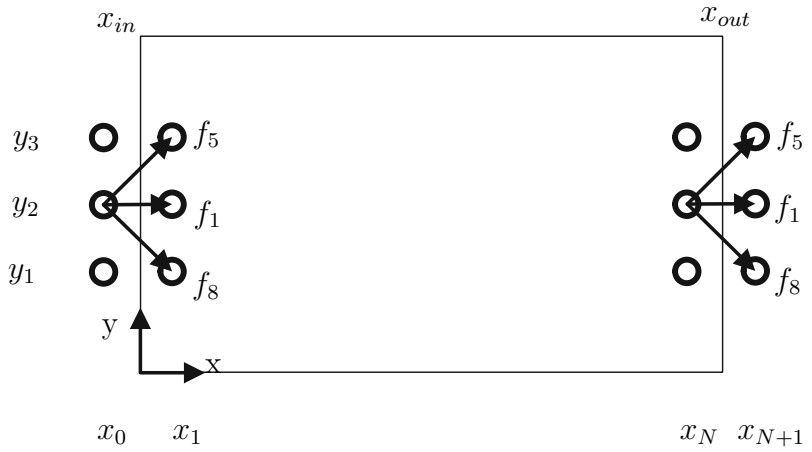


Figure 2: An illustration of the periodic boundary condition where the distributions f_5, f_1, and f_8 exit the domain at the right boundary (x_N) and re-enter at the left boundary (x_1). To implement this, additional ghost nodes are introduced at x_{N+1} and x_0. Illustration adapted from [27].

2.5.3 Bounce Back Scheme

The most common boundary condition applied at a fluid-solid interface in hydrodynamic problems is the no-slip velocity boundary condition. This can be achieved in the LBM using the *bounce-back* boundary condition. The idea is simple: the molecules that are transported towards a solid wall are bounced back in the opposite direction to the same node. This is explained by the example shown in Fig. 3, where molecules are transported in the $-y$ and $+x$ direction, which is direction number 15 in the $D3Q19$ lattice, and it is

then reflected back to the same node in the opposite direction, that is direction number 14. The mathematical form of this rule is as follows:

$$f_i\left(\mathbf{x}_f, t+\Delta t\right) = f_i\left(\mathbf{x}_f, t\right) \quad \text{with} \quad \boldsymbol{\xi}_{\hat{i}} = -\boldsymbol{\xi}_i, \tag{22}$$

which is applied to all directions $\hat{i}$ where $\xi_{\hat{i}} \cdot \mathbf{n} < 0$. To be consistent with the convective scaling, that is, distribution values must be transported from one node to another in exactly one time step, the physical wall must be placed at a distance of half grid spacing. Hence, in the Fig. 3 the wall is located at half way between the wall node x_w and the fluid node x_f.

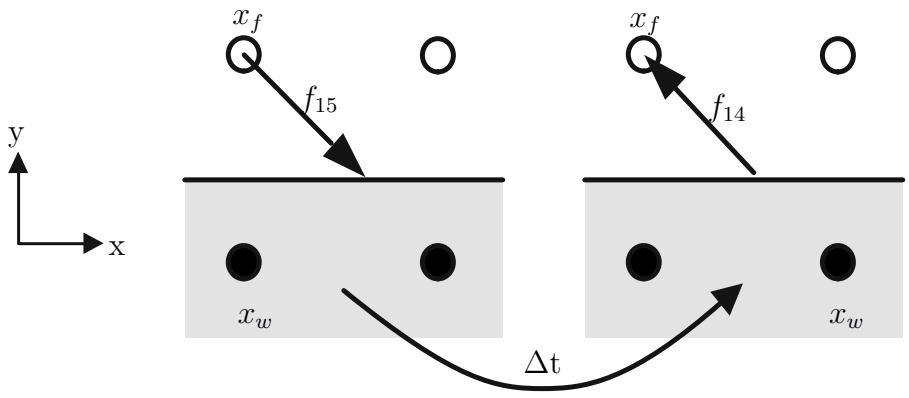


Figure 3: An illustration of the half-way bounce back boundary condition, where the distribution f_{15} is streamed from the fluid node $\mathbf{x}_f$ towards the wall location and is then reflected back in the opposite direction to the same fluid node, as shown on the right. This mechanism effectively enforces the no-slip condition at a location halfway between the fluid node $\mathbf{x}_f$ and wall node $\mathbf{x}_w$. Illustration inspired by [27].

2.6 Initial Conditions

The Lattice Boltzmann method is an initial boundary value problem. Therefore, appropriate values must be set for the distributions before the simulations can be initiated. In the case of channel flow with body force, there are essentially two options for initialising the macroscopic velocity: either set it to 0 and let the flow develop or use a starting profile to accelerate the development of turbulence. The following two methods can be used to achieve this goal. For a thorough discription of initialisation algorithms in LBM the reader is directed to [28, 29].

2.6.1 Analytical Schemes

To initialise from a zero-velocity equilibrium distribution, the equilibrium distribution can be computed using Eq. (4) is assigned to all f_i. That is,

$$f_i(\boldsymbol{x}, t=0) = f_i^{\mathrm{eq}}\left(\rho_0, \boldsymbol{u}_0(\boldsymbol{x})\right),$$

where ρ_0 is a constant and $\boldsymbol{u}_0(\boldsymbol{x}) = \mathbf{0}$.

Because the primary interest in turbulent channel flow lies in the statistical behaviour of the fully developed turbulence, the unphysical transients present at initialisation do not affect the final solution. However, a velocity profile cannot be initialised in a similar manner with simply the equilibrium distribution, because the higher-order terms of the distribution as well as the density variation would be missing and cause the profile to equalise the non-physical starting condition. For a more accurate initialisation, the non-equilibrium part of the distribution function should also be computed.

2.6.2 Iterative Schemes

The macroscopic velocity alone is insufficient to define physically accurate initial boundary conditions. Appropriate density values, ρ_0 across the boundaries must also be specified to ensure correct initialisation, particularly when a non-zero initial state is required. Because both the pressure field p_0 and the velocity field $\mathbf{u}_0$ must be accurate solutions of the Navier–Stokes equations in the initial state.

To obtain p_0 or ρ_0, the Poisson equation must be solved [30], which can be achieved using various numerical schemes [31]. An appealing alternative was proposed in [29], where an iterative scheme was introduced, as outlined in Algorithm 1.

Algorithm 1 Consistent Initialisation via Modified LB Scheme

Reference state: Given a solenoidal velocity field $\mathbf{u}_0(\mathbf{x})$, the goal is to iteratively initialise a consistent hydrodynamic state.

1: Initialise the populations f_i using $f_i^{\mathrm{eq}}(\rho_0, \mathbf{u}_0(\mathbf{x}))$.
2: Compute the density: $\rho(\mathbf{x}) = \sum_i f_i(\mathbf{x})$.
3: Perform collision using modified equilibrium: $f_i^{\mathrm{eq}}(\rho(\mathbf{x}), \mathbf{u}_0(\mathbf{x}))$ *(update ρ, keep $\mathbf{u}_0$).*
4: Stream populations to neighbouring nodes.
5: Repeat from step 2 until convergence of f_i (or macroscopic fields) is achieved.
6: **Note:** Final iteration must end with *streaming*, not collision, to ensure a correct non-equilibrium state.

Such an algorithm could be used to initialise a velocity profile with a power law to reduce the computational time of flow development. However, in this study, all the simulations were initialised from a zero-velocity equilibrium distribution.

3 Large Eddy Simulation for Lattice Boltzmann Method

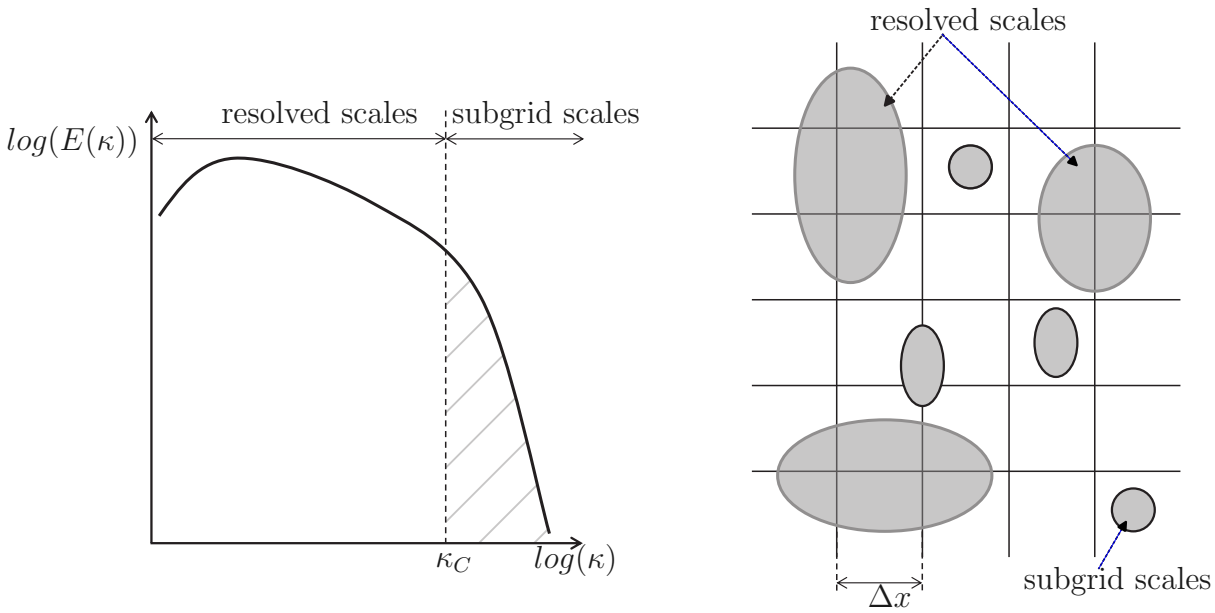
Modelling flows of technical relevance often necessitates the consideration of turbulence. Solving the Lattice Boltzmann Equation (LBE) or Navier–Stokes Equations (NSE) directly for turbulent flows requires spatial and temporal discretisations finer than the smallest dynamically active scales in the flow. Simulations with such resolution are referred to as *Direct Numerical Simulations* (DNS). It is well established that the computational cost of DNS scales cubically with the Reynolds number of the flow, that is, the computational effort grows as $\mathcal{O}(\mathrm{Re}^3)$ [32]. Owing to this steep scaling, performing DNS on technically relevant flows, where $\mathrm{Re} \sim 10^5 - 10^8$, remains computationally infeasible, even with continual advancements in high-performance computing. As a result, DNS is primarily restricted to simplified or canonical flow cases, such as turbulent channel flow [33], which was used in this study for verification.

To overcome the impractical computational requirements of DNS, various modelling techniques have been developed that enable the simulation of turbulent flows at a reduced cost [34]. Among these, this study focuses on LES which provides better insights into the unsteady details of flow than RANS models [35] with considerably less computational effort than DNS. The following section provides an overview of the LES approach, following the work of [13]. A thorough mathematical description is provided in [32].

3.1 Concept of Large Eddy Simulations

The basic principle of LES is the separation of dynamically active scales into two parts: large and small scales. The large scales, that is, the spatially larger eddies and their respective active temporal scales, are resolved directly, whereas the influence of the smaller subgrid scales is modelled using a subgrid-scale model.

The scale separation is shown in Fig. 4. In Fig. 4(a), the cut-off of the energy spectrum is shown in Fourier space, where the turbulent kinetic energy (TKE), $E(\kappa)$, is resolved below a particular wavenumber κ_C, and modelled beyond it. Similarly, Figure 4(b) depicts the scale separation in physical space, where eddies smaller than a given cut-off scale—typically related to the grid spacing Δx—are modelled, and larger eddies are directly resolved.



(a) Fourier Space

(b) Physical Space

Figure 4: Figure (a) shows a log-log plot of the turbulent kinetic energy $E(\kappa)$ versus the wave number κ, illustrating that only the energy associated with wave numbers lower than the cutoff wave number κ_c is resolved. Higher-frequency components were modelled using a subgrid-scale (SGS) model. Figure (b) represents the same scale separation in the physical space, where turbulent structures larger than the grid spacing Δx are resolved and smaller structures are modelled. Illustration adapted from [32].

This scale separation procedure is supported by several theoretical and empirical arguments [36]. Large eddies are strongly influenced by the flow configuration and are difficult to model because they tend to be anisotropic, inhomogeneous, diffusive, and persistent over longer time scales. In contrast, smaller scales are generally more isotropic and homogeneous, making them more amenable to modelling. It is commonly recommended to resolve at least 80% of the turbulent kinetic energy (TKE) to achieve accurate simulation results [37].

3.2 Filtering Operation for LES

Mathematical filtering operations are essential for achieving scale separation in LES; here, analytical functions are applied to the governing equations for filtering. The convolution product offers a convenient representation of the filtering process. In the physical space, a high-pass filter is required because the aim is to remove smaller scales. Conversely, in the Fourier space, it functions as a low-pass filter to omit high-frequency or high-wavenumber components from the solution.

This section presents an overview of filtering in the physical space through decomposition and convolution. Let $\phi(\mathbf{x}, t)$ be any flow variable that depends on the spatial coordinate $\mathbf{x}$ and time t. It can be decomposed into a resolved (filtered) component $\bar{\phi}(\mathbf{x}, t)$

and an unresolved (subgrid) component $\phi'(\mathbf{x}, t)$ as follows:

$$\phi(\mathbf{x}, t) = \bar{\phi}(\mathbf{x}, t) + \phi'(\mathbf{x}, t). \tag{23}$$

The filtering operation can be formally expressed as a convolution product [38]:

$$\bar{\phi}(\mathbf{x}, t) = \int_{-\infty}^{\infty} G\left(\mathbf{x} - \mathbf{x}'\right) \phi\left(\mathbf{x}', t\right) d\mathbf{x}' = G * \phi, \tag{24}$$

where G is the filter kernel used for spatial averaging. The filtering kernel G serves as the defining component of the filtering operation. In three-dimensional space, it is constructed by taking the product of one-dimensional kernels applied along each spatial direction, as follows:

$$G\left(\mathbf{x} - \mathbf{x}'\right) = \prod_{i=1}^{3} G_i\left(x_i - x_i'\right). \tag{25}$$

Each G_i represents a kernel acting in the i-th coordinate direction, and the full multidimensional kernel is obtained by tensorizing these one-dimensional components.

Some of the common filters for these monodimensional kernels G_i are the Gaussian filter, Fourier cutoff filter, and box filter. The latter is defined as follows:

$$G_i\left(x_i - x_i'\right) = \begin{cases} \frac{1}{\overline{\Delta_c}} & \text{if } |x_i - x_i'| \leq \frac{\overline{\Delta_c}}{2} \\ 0 & \text{else} \end{cases} \tag{26}$$

A box filter with a cutoff length $\overline{\Delta_c} = 1$ is shown in Fig. 5, the function is 0 for $(x_i - x_i') \pm \overline{\Delta_c}/2$. The cutoff length $\overline{\Delta_c}$ is primarily related to the grid spacing Δx.

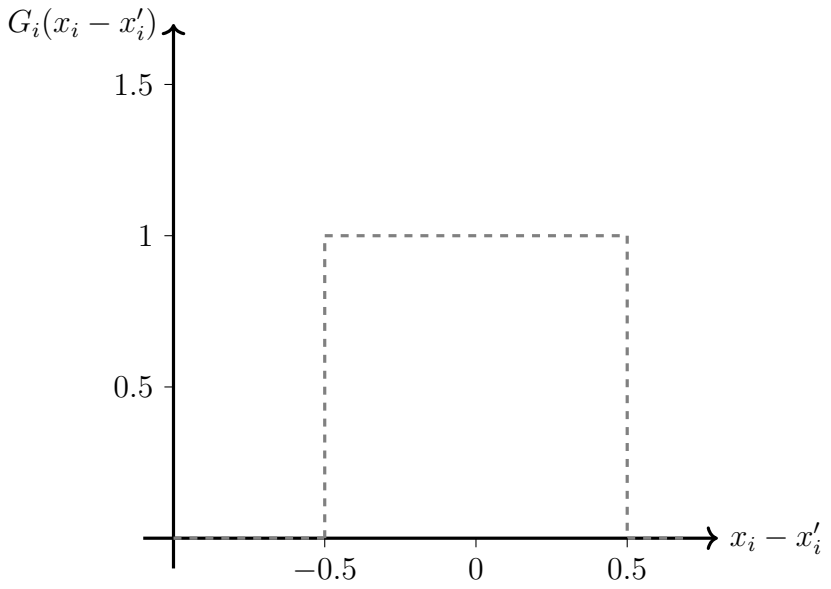


Figure 5: Box filter kernel $G_i(x_i - x_i')$, which has a constant value within the interval $[-0.5, 0.5]$ and is zero everywhere outside this range.

As mentioned before, the filtering operation shown above is applied to the governing equations, either the Navier-Stokes or the Boltzmann equation; hence, it is called an analytical filter. Then, the filtered governing equations are solved along with the subgrid-scale model to account for the unresolved lower (physical) scales. It should be noted that the true effective filter is not the analytical filter applied here, as different factors play a role. The grid size itself is a filter, which discards all length scales lower than its resolution, the approximation of the gradients with the numerical schemes also causes additional filtering, and finally, the subgrid-scale model error also contributes to the filtering. The calculation of these effects would be tedious, and often the analytical filter with cutoff length $\overline{\Delta_c} = \Delta x$ is approximated to the effective filter.

3.3 Filtering Operation Applied to the Lattice Boltzmann Method

The application of this type of analytical filter to the LBM was first performed by Hou et al. [39], and more consistent works on which the following explanations are based are [40].

The convolution filtering operation is shown in Eq. 24, applied to the Boltzmann equation, gives the filtered Boltzmann equation:

$$\frac{\partial \bar{f}}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial \bar{f}}{\partial \mathbf{x}} = \overline{\Omega(f)}, \tag{27}$$

where $\bar{f}$ is the filtered distribution given by $\bar{f} = G * f$ and $\overline{\Omega(f)}$ denotes the filtering operation applied to the collision operator. Here, for example, is applied to the BGK model:

$$\overline{\Omega(f)} = G * \left[-\frac{1}{\tau} \left(f - f^{eq} \right) \right] = -\frac{1}{\tau} \left(\bar{f} - \overline{f^{eq}} \right). \tag{28}$$

This direct filtering of the collision model is not possible because of the equilibrium distribution $\overline{f^{eq}}$, as it would require the unfiltered density and velocity values, that is, $\rho = \bar{\rho} + \rho'$ and $\mathbf{u} = \overline{\mathbf{u}} + \mathbf{u}'$. ρ' and $\mathbf{u}'$ the unresolved quantities are unknown. Hence, the equilibrium distribution is calculated only with the filtered macroscopic values $f^{eq}(\bar{\rho}, \overline{\mathbf{u}})$, and a correction term R is added, as suggested in [39].

$$\begin{aligned} \overline{\Omega(f)} &= \Omega(\bar{f}) + R = -\frac{1}{\tau} \left[\bar{f} - f^{eq}(\bar{\rho}, \overline{\mathbf{u}}) \right] + R \\ R &= \frac{1}{\tau} \left[\overline{f^{eq}} - f^{eq}(\bar{\rho}, \overline{\mathbf{u}}) \right] \end{aligned} \tag{29}$$

To obtain the correction term R, a subgrid-scale model needs to be used, since it again consists of the $\overline{f^{eq}}$ term. With the inclusion of this correction term, the discretised lattice Boltzmann equation takes the following in the following form:

$$\bar{f}_i \left(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t \right) = \bar{f}_i(\mathbf{x}, t) + \Omega \left[f_i^{eq}(\bar{\rho}(\mathbf{x}, t), \overline{\mathbf{u}}(\mathbf{x}, t)) - \bar{f}_i(\mathbf{x}, t) \right] + R_i. \tag{30}$$

There are various ways to approximate the discretised correction term R_i, and the Boussinesq approximation is the most often used. where the transport of turbulent energy is treated as equivalent to the momentum transfer; hence, the kinematic viscosity ν is replaced by an effective kinematic viscosity ν_{eff}, that is $\nu_{eff} = \nu + \nu_t$. ν_t is the eddy viscosity. This can be readily incorporated into the filtered LBE in the relaxation parameter as follows:

$$\bar{f}_i\left(\mathbf{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t\right) = \bar{f}_i(\mathbf{x}, t) + \Omega_{\text{eff}}\left[f_i^{\text{eq}}(\bar{\rho}(\mathbf{x}, t), \overline{\mathbf{u}}(\mathbf{x}, t)) - \bar{f}_i(\mathbf{x}, t)\right], \tag{31}$$

$$\Omega_{\text{eff}} = \frac{c_s^2 \Delta t}{\nu + \nu_t + 0.5\, c_s^2 \Delta t}. \tag{32}$$

The eddy viscosity term ν_t, added to the effective relaxation parameter Ω_{eff} is modelled using subgrid-scale models, as explained in the next subsection.

3.4 Smagorinsky Subgrid Scale Model

The purpose of the subgrid scale model is to approximate the energy transfer from the large scales to the smaller ones and this is done by adding the eddy viscosity term ν_t, in the effective relaxation parameter (Eq. 32). The Smagorinsky model, one of the most widely used models, was implemented in this context, where the eddy viscosity was calculated as

$$\nu_t = \left(C_s \overline{\Delta_c}\right)^2 |\overline{\mathbf{S}}| = (C_s \Delta x)^2 |\overline{\mathbf{S}}|. \tag{33}$$

C_s is the dimensionless Smagorinsky constant, which is commonly set as a calibration parameter set between 0.1-0.2. Theoretically, it has been determined to be approximately 0.18 in the case of isotropic homogeneous turbulence [40]. $\overline{\Delta_c}$, is the cutoff length scale of the filter, which is equal to the grid spacing Δx in this case. $|\overline{\mathbf{S}}|$ is the norm of the strain rate tensor, which is defined as

$$\bar{S}_{\alpha\beta} = \frac{1}{2}\left(\frac{\partial \bar{u}_\alpha}{\partial x_\beta} + \frac{\partial \bar{u}_\beta}{\partial x_\alpha}\right), \tag{34}$$

$$|\overline{\mathbf{S}}| = \sqrt{2 \bar{S}_{\alpha\beta} \bar{S}_{\alpha\beta}}. \tag{35}$$

$|\overline{\mathbf{S}}|$, can be obtained either by a finite difference stencil or calculated from the values of the distribution $\overline{f_i}$ using the following relation:

$$\overline{\mathbf{S}} = -\frac{\Omega_{eff}}{2\rho c_s^2 \Delta t} \boldsymbol{\Pi}^{\text{neq}}_{\text{discr}}, \tag{36}$$

where $\boldsymbol{\Pi}^{\text{neq.}}_{\text{discr.}}$ is:

$$\mathbf{\Pi}^{\text{neq.}}_{\text{discr.}} := \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i \left[\overline{f_i} - f_i^{eq}(\bar{\rho}, \overline{\mathbf{u}})\right]. \tag{37}$$

The expression linking the distribution to the strain rate is 36, set in Eq. 33 gives us:

$$\nu_t = (C_s \Delta x)^2 \frac{\Omega_{eff}}{2\rho c_s^2 \Delta t} |\mathbf{\Pi}^{\text{neq}}_{\text{discr}}|. \tag{38}$$

Eq. 32 and Eq. 38 provide a quadratic equation that can be solved for, and the positive (physical) solution is:

$$\nu_t = \frac{\sqrt{\left(v + \frac{1}{2}c_s^2 \Delta t\right)^2 + \frac{2}{\rho}(C_s \Delta x)^2 |\Pi^{\text{neq}}_{\text{discr}}|} - \left(v + \frac{1}{2}c_s^2 \Delta t\right)}{2}. \tag{39}$$

Equation 39 is calculated at each node for each time step, and the eddy viscosity is obtained and applied in the relaxation parameter Eq. 32. This direct calculation of the eddy viscosity ν_t is another advantage of the LBM, avoiding the more computationally intensive finite difference method. In this manner, the effect of unresolved eddies is modelled in LES.

The Smagorinsky model, in the form given above, has a few characteristic drawbacks. Because the model is dependent on the gradients of the resolved macroscopic velocity field, the eddy viscosity term, ν_t, is always greater than 0, even when the flow is laminar [32], [41]. This means that the model should only be used when the flow is fully turbulent. This issue does not concern this particular study, as the results only depend on the fully developed turbulent state, and many different solutions are available for this problem [40].

Another drawback is that the Smagorinsky model is excessively dissipative near the wall because of the non-homogeneous and non-isotropic nature of the turbulence close to the wall. To solve this issue, a van Driest damping function [32] can be incorporated as follows:

$$C_s^{vD} = C_s \left[1 - \exp\left(-y^+/A^+\right)\right]^2 \quad A^+ = 25, \tag{40}$$

$$y^+ = \frac{y u_\tau}{\nu}, \tag{41}$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}. \tag{42}$$

The physical wall distance y was non-dimensionalized with the friction velocity u_τ and kinematic viscosity ν, yielding y^+. C_s^{vD} decreases exponentially with decreasing y^+ and vanishes at the wall. The use of this expression leads to better near-wall behaviour.

Slight variations in the damping functions can be made to incorporate the $\nu_t \sim y^{+3}$ relation [32]:

$$C_s^{vD^*} = C_s \left[1 - \exp\left(-y^+/A^+\right)^3\right]^{1/2} \quad A^+ = 25. \tag{43}$$

4 Wall Modelling for Large Eddy Simulations

As mentioned in the introduction, two options are available for resolving wall boundaries in LES: wall-resolved (WR) and wall-modelled (WM) approaches. The wall-resolved approach does not require any additional modelling, and the theory of LES explained earlier can be used directly with the Dirichlet boundary conditions. However, the turbulent scale becomes smaller as we approach the wall, and a grid size that is sufficient to resolve 80% of the TKE in the outer region will be insufficient to do so close to the wall. In addition, the first fluid node must reside approximately at a y^+ of 1, which is the viscous sublayer, to resolve the velocity gradient at the wall correctly. This means that the grid must be refined progressively as it approaches the wall, which incurs additional computational costs. As mentioned in the Introduction, in conventional CFD where body-fitted meshes are used, the mesh is refined only in the wall-normal direction, resulting in cells with high aspect ratios, thereby reducing the total number of nodes required. This strategy is not feasible with the uniform grids used in LBM, further emphasising the need for wall modelling in the LBM-LES context.

According to the estimates in [42], the computational cost for the boundary layer resolution is $O\left(Re^{1.8}\right)$ compared to $O\left(Re^{0.4}\right)$ in the outer region. This limits the WR approach to cases with Reynolds numbers below 10^5 and makes the WM approach essential for higher Reynolds number flows.

The following chapter first introduces the turbulent boundary layer [13], then covers the basics of wall modelling along with conventional wall functions, and finally introduces the wall modelling algorithm implemented in this study along with its derivation.

4.1 The Canonical Turbulent Boundary Layer

The structure of the boundary layers in turbulent flows is essential for developing accurate wall models. Although canonical flows involve several simplifications, such as the absence of adverse pressure gradients, surface roughness, wall curvature, and the assumption of fully developed turbulence, they still offer valuable insights. This section introduces the key features of turbulent boundary layers and the analytical approximations commonly used to describe their behaviour.

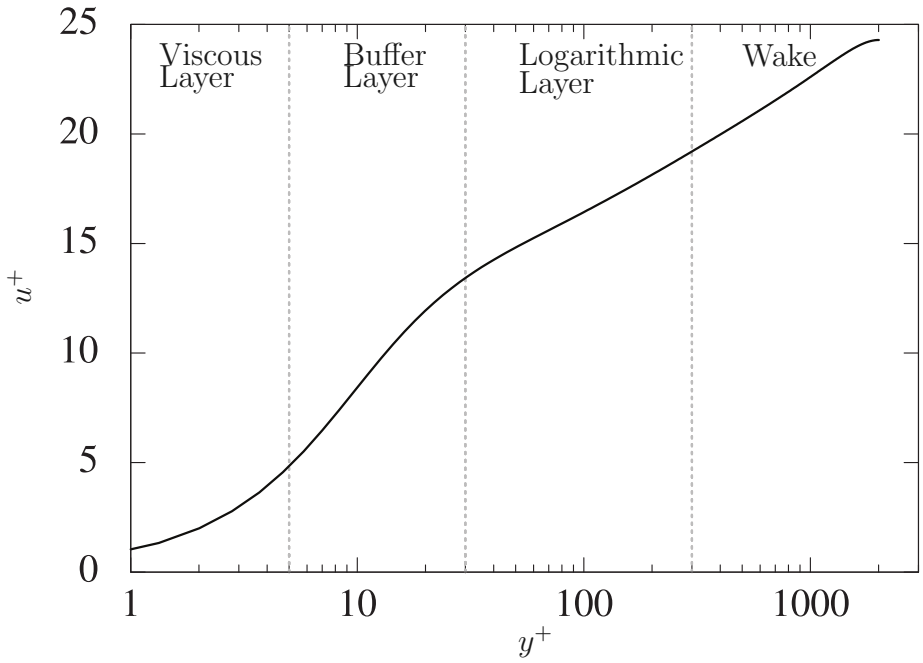


Figure 6: Sub-layers in the Turbulent Boundary Layer, adapted from [13]

.

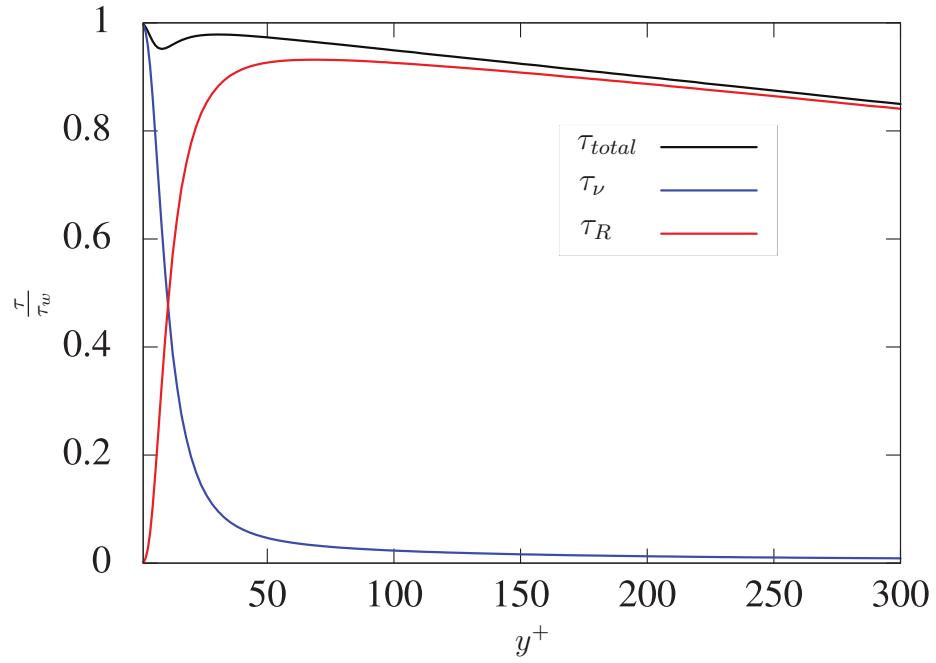


Figure 7: Distribution of normalized shear stresses in a turbulent boundary layer as a function of wall-normal coordinate y^+. The plot shows the viscous shear stress τ_ν, turbulent Reynolds shear stress τ_R, and total shear stress τ_{total}, adapted from [13].

Figure 6 illustrates the variation of the non-dimensional wall distance y^+, given in

Eq. 41, against the non-dimensional velocity u^+, which is expressed as

$$u^+ = \frac{\langle u \rangle}{u_\tau}, \tag{44}$$

where $\langle u \rangle$ denotes the mean streamwise velocity and u_τ is the friction velocity.

Figure 8 shows the variation in the different stress components within the turbulent boundary layer as a function of y^+. The viscous stress, $\tau_\nu = \mu \, d\langle u \rangle / dy$, arises from molecular viscosity μ and dominates near the wall. The Reynolds stress, $\tau_R = -\rho \, u'v'$, originates from the convective term in the Navier–Stokes equations via Reynolds averaging. Although not of molecular origin, it is interpreted similarly to viscous stress, as both contribute to momentum transfer and dissipation within the flow.

The mean velocity $\langle u \rangle$ and the fluctuating components u', v', w' are obtained through *Reynolds decomposition*, which separates the instantaneous flow variable into mean and fluctuating parts. The time-averaged mean velocity is defined as

$$\langle u(\mathbf{x}) \rangle = \frac{1}{T} \int_t^{t+T} u \left(\mathbf{x}, t' \right) \, dt' \tag{45}$$

The corresponding fluctuation is then given by

$$u'(\mathbf{x}, t) = u(\mathbf{x}, t) - \langle u(\mathbf{x}) \rangle \tag{46}$$

This decomposition applies similarly to the other velocity components v and w, leading to their respective fluctuations v' and w'.

Furthermore, the velocity profile close to the wall, that is, the boundary layer, as shown in Figure 6 can be divided into distinct regions of the inner boundary layer based on the dominant physical mechanisms:

1. **Viscous sublayer:** This is the region closest to the wall, typically valid for $y^+ \lesssim 5$. In this layer, turbulence is absent, and the flow is dominated solely by viscous shear stresses; that is, Reynolds stresses are negligible. The velocity profile in this region follows a linear law, assuming a constant total shear stress [41]:

$$u^+ = y^+ \tag{47}$$

2. **Buffer layer:** This intermediate region spans approximately in the range of $5 < y^+ \lesssim 30$. Here, both viscous and turbulent (Reynolds) stresses coexist. This is a transitional zone where turbulent eddies begin to emerge but are still significantly damped by viscosity. Sagaut [32] proposed a logarithmic law that approximates the mean velocity profile in the buffer region as

$$u^+ = A \cdot \ln(y^+) + B \tag{48}$$

with constants $A = 5.0$ and $B = -3.05$, which provides a good empirical fit.

3. **Logarithmic layer:** In this outer region, where $y^+ \gtrsim 30$, the Reynolds stresses dominate the momentum transfer and viscous effects are less. For canonical channel flows, the mean velocity profile follows the well-known logarithmic law [41]:

$$u^+ = \frac{1}{\kappa} \ln(y^+) + B \tag{49}$$

where $\kappa \approx 0.41$ is the von Kármán constant, and $B \approx 5.5$. Owing to the logarithmic nature of the profile, this region is commonly referred to as the *log-law region*.

Beyond the inner region, the remaining exterior part of the boundary layer is referred to as the *outer layer* or *wake region*, where the influence of the wall's viscous effects is no longer felt by the flow.

4.2 Overview of Wall Modelling Principles

In this subsection, the general purpose and requirements of wall models for turbulent flow simulations are introduced. Specifically, it outlines the role that wall models are expected to play. Classical wall functions, which have been widely employed for many years, are briefly discussed along with the fundamental assumptions on which they are based.

Typically, the boundary nodes in the mesh lie within the logarithmic region of the boundary layer. Consequently, the wall-normal distance from these nodes exceeded the local turbulent length scale. This poses two key challenges that must be addressed by wall models.

First, the unresolved near-wall region must be modelled because it falls below the resolution capabilities of the mesh. Second, the velocity gradients and by extension, the wall shear stress cannot be directly resolved at such high values of y^+. This is particularly critical because wall shear stress is directly linked to the force balance of the flow.

Therefore, a fundamental requirement of any wall model is to provide a reasonably accurate estimation of the wall shear stress and other near-wall quantities, compensating for the lack of resolution in the viscous sublayer and buffer region.

The turbulent boundary layer (TBL), as discussed in Section 4.1, allows the formulation of a functional relationship between the dimensionless velocity u^+ and the wall-normal coordinate y^+. This relationship enables the deduction of a wall shear stress expression of the form $\langle \tau_w \rangle = \mathcal{F}(\langle u \rangle, y, \ldots)$, commonly referred to as a *wall function*.

To implement this expression within the LES framework, two key assumptions must be made. First, it is assumed that the relation holds instantaneously, that is, the mean quantities $\langle \tau_w \rangle$ and $\langle u \rangle$ are replaced by their instantaneous counterparts τ_w and u. Second, because the wall function typically provides only the magnitude of the wall shear stress and not its vectorial direction, this direction must be approximated. A commonly used

approach is to align it with the direction of the instantaneous tangential velocity at the wall, as described in [43].

$$\boldsymbol{\tau}_w = \frac{\mathbf{u}_t}{|\mathbf{u}_t|}\,\tau_w. \tag{50}$$

This approximation ensured that the wall shear stress vector was oriented tangentially to the wall in the direction of the local flow.

Because Cartesian grids over complex geometries inevitably place boundary nodes across various regions of the boundary layer—not solely within the logarithmic region—it is essential to ensure that the wall model delivers reasonable predictive accuracy throughout all layers. To this end, a piecewise function combining the characteristic expressions from Section 4.1 offers a straightforward yet effective wall modelling approach:

$$u^+ = \begin{cases} y^+ & \text{for } y^+ \leq 5, \\ 5\ln(y^+) - 3.05 & \text{for } 5 < y^+ \leq 28.178, \\ \dfrac{1}{\kappa}\ln(y^+) + 5.5 & \text{for } y^+ > 28.178, \end{cases} \tag{51}$$

where κ is the von Kármán constant, typically taken as $\kappa = 0.41$. This expression captures the behaviour of the velocity profile across the viscous sublayer, buffer layer, and logarithmic region in a unified formulation, as shown in Figure 8.

A reasonable agreement was observed between the three analytical expressions and DNS data. Notably, the transition points between the piecewise segments were selected based on the intersections of the curves to maintain continuity in the velocity profile. However, it should be noted that this piecewise function is not smooth and can cause problems in the velocity gradient calculations.

Alternatively, a commonly used simplification of the three-layer approach is a two-layer formulation in which only the linear and logarithmic laws are retained. The transition between the two occurs at their points of intersection (Figure 9). However, this simplification introduces significant errors in the buffer region, where neither law accurately captures the flow behaviour.

Regardless of whether a three- or two-layer formulation is employed, the computation of the wall shear stress, τ_w, remains necessary. This typically involves iteratively solving implicit equations derived from the respective velocity–distance relations. However, this approach can be computationally expensive and may introduce convergence issues, particularly in complex flow geometries. In this study, we adopted the explicit wall function proposed in [35], which is described in detail in the following sections.

For completeness, the power-law-based velocity profile proposed by Werner and Wengle [43] is also presented in Figure 9. This approach differs from the standard two-layer formulation in two key ways. First, both the logarithmic and buffer layers are represented using a continuous power-law expression of the form $u^+ = Ay^{+B}$, where A and B are

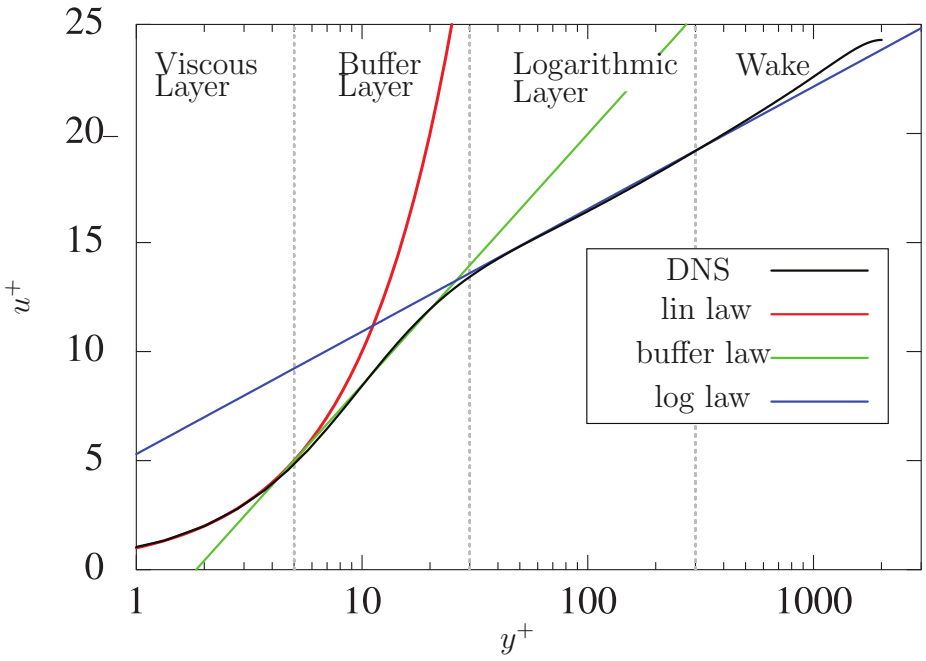


Figure 8: Turbulent boundary layer velocity profile plotted in dimensionless units (u^+ vs. y^+), highlighting the distinct regions: viscous sublayer, buffer layer, logarithmic layer, and wake region. The curves show data from DNS and analytical approximations: linear law, buffer layer approximation, and logarithmic law. Figure adapted from [13].

empirical constants. Second, the parameters vary across layers, as follows:

$$u^+ = \begin{cases} y^+ & y^+ \leq 7.1 \\ 2.73247\,(y^+)^{0.487215} & 7.1 < y^+ \leq 25.19 \\ 8.3\,(y^+)^{1/7} & y^+ > 25.19 \end{cases} \tag{52}$$

This formulation reduces the overall deviation from the DNS data and smoothens the velocity profile by mitigating the sharp transitions between the layers. Additionally, Equation 4.2 provides an explicit formulation for wall shear stress τ_w, thereby avoiding the need for iterative procedures.

4.3 Advanced wall models

As discussed in the previous section, it is possible to construct functional relationships between y^+ and u^+ and estimate the wall shear stress τ_w. However, such formulations must be consistent with the underlying LES framework and provide physically meaningful boundary conditions for the LES solver. In this section, a detailed wall treatment strategy is presented based on the methodologies proposed in the works of [4], [44], and [35]. These approaches offer explicit and practical solutions for incorporating wall modelling in LES, particularly in the presence of complex geometries and under-resolved near-wall regions.

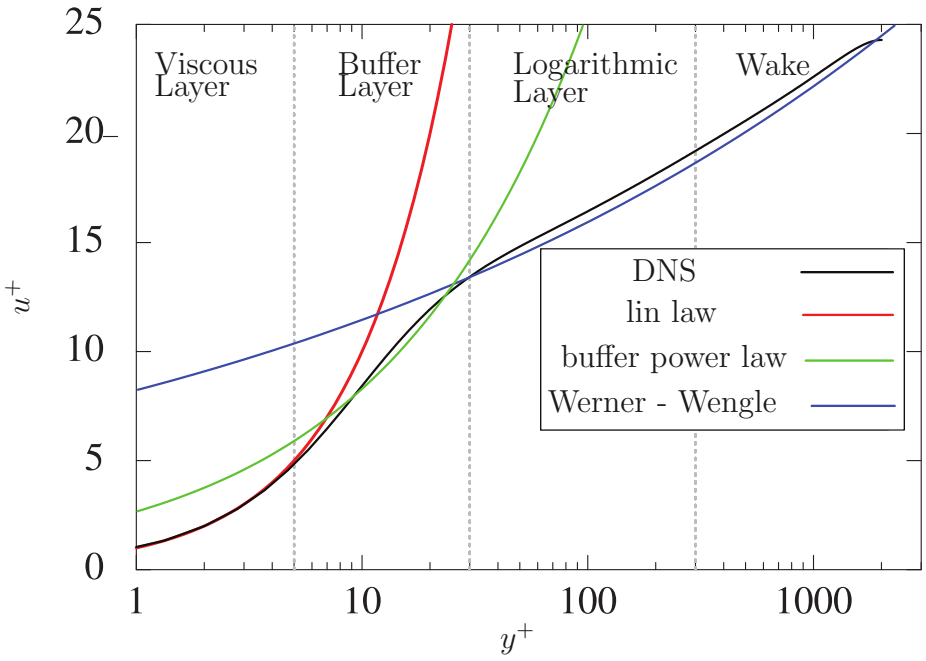


Figure 9: Turbulent boundary layer velocity profile plotted in dimensionless units (u^+ vs. y^+), highlighting the distinct regions: viscous sublayer, buffer layer, logarithmic layer, and wake region. The curves show data from DNS and analytical approximations: linear law, buffer layer approximation, and Werner and Wengle law. Figure adapted from [13].

4.3.1 RANS-Explicit Wall Models for LES

To overcome the over-dissipative nature of the LES near the wall, a RANS-based wall model can be employed to provide the required wall shear stress τ_w. Assuming a parallel flow with negligible pressure gradients close to the wall, the total shear stress in the RANS context is defined as

$$\rho\left(\nu + \nu_t^{\text{RANS}}\right)\frac{dU}{dy} = \tau_w, \tag{53}$$

where U is the mean tangential velocity in the RANS approach. Because the wall shear stress is related to the friction velocity, we can equivalently express:

$$\tau_w \equiv \rho u_\tau^2. \tag{54}$$

Equations 54 and 53 can be rewritten in viscous wall units by applying standard non-dimensionalization. The RANS equilibrium momentum equation is as follows:

$$\rho\left(\nu + \nu_t^{\text{RANS}}\right)\frac{dU}{dy} = \tau_w = \rho u_\tau^2, \tag{55}$$

We divide both sides by ρu_τ^2 to obtain

$$\frac{\nu + \nu_t^{\text{RANS}}}{u_\tau^2}\frac{dU}{dy} = 1. \tag{56}$$

We now express this in viscous (wall) units using the following definition:

$$U^+ = \frac{U}{u_\tau}, \quad y^+ = \frac{y u_\tau}{\nu}, \quad \nu_t^+ = \frac{\nu_t^{\mathrm{RANS}}}{\nu}. \tag{57}$$

Note that the velocity gradient transforms as

$$\frac{dU}{dy} = \frac{d(U^+ u_\tau)}{dy} = u_\tau \frac{dU^+}{dy}. \tag{58}$$

Substituting this into the previous equation (Eq. 56) gives:

$$\frac{\nu + \nu_t^{\mathrm{RANS}}}{u_\tau^2} \cdot u_\tau \frac{dU^+}{dy} = 1, \tag{59}$$

which simplifies to:

$$\frac{\nu + \nu_t^{\mathrm{RANS}}}{u_\tau} \frac{dU^+}{dy} = 1. \tag{60}$$

Dividing the numerator and denominator by ν, we obtain

$$\left(1 + \nu_t^+\right) \frac{\nu}{u_\tau} \frac{dU^+}{dy} = 1. \tag{61}$$

Recognising that $dy^+ = \frac{u_\tau}{\nu} dy$, we find:

$$\left(1 + \nu_t^+\right) \frac{dU^+}{dy^+} = 1. \tag{62}$$

This represents the final form of the equilibrium wall model in viscous units, which can be integrated to obtain a continuous expression for the velocity profile:

$$U^+ = \int \frac{1}{1 + \nu_t^{\mathrm{RANS},+}} \, dy^+. \tag{63}$$

Here the RANS eddy viscoisty is calculated using a Prandtl mixing length model [45] with the near wall eddy viscosity damping as given here:

$$\nu_t^{\mathrm{RANS},+} = \kappa y^+ \left(1 - e^{-y^+/A^+}\right)^2, \tag{64}$$

where $A^+ = 19$ and $\kappa = 0.41$ is the von Kármán constant. Even RANS turbulence models require a damping function to ensure an accurate reduction of eddy viscosity in the near-wall region, where turbulence is suppressed by the viscous effects.

However, substituting Equation 64 into Equation 63 and solving the resulting integral analytically is not straightforward. Therefore, a fitted expression was proposed by Cai *et al.* [4] using numerical integration as follows:

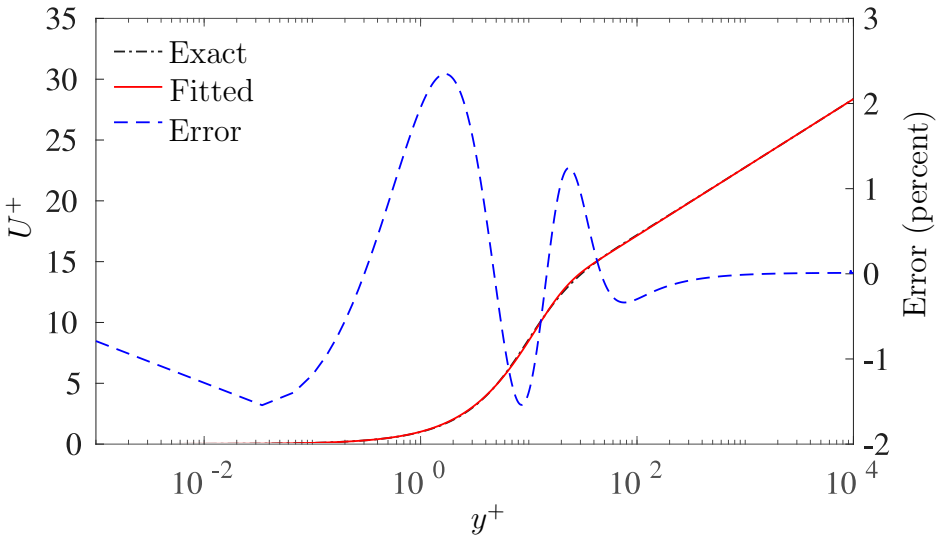


Figure 10: Comparison of the piecewise velocity profile with DNS data in the turbulent boundary layer. Adapted from [4].

$$U^+ = \left[1 - \tanh\left(\frac{y^+}{10.71}\right)\right]^{1.526} y^+ + \left[\tanh\left(\frac{y^+}{10.71}\right)\right]^{1.526} \frac{1}{\kappa}\log\left(Ey^+\right), \quad \text{where } E = 11.27. \tag{65}$$

This wall model (Eq. 65) offers several advantages: firstly, it is consistent with the RANS eddy viscosity model employed near the wall; secondly, it exhibits a lower absolute error compared to the wall models discussed in the previous subsection; and thirdly, the velocity profile is smooth and continuous without sharp transitions, finally its fully explicit. For these reasons, this model was adopted in the present study.

We still require the friction velocity u_τ, which can be obtained by iteratively solving equation 65, for example, using the Newton method, because the expression is implicit in u_τ. However, this procedure can be computationally expensive for complex geometries and may be sensitive to the initial guess.

Therefore, an explicit expression for the friction velocity, as proposed in [35], was implemented in this study. This formulation is consistent with the mixing length model and is derived as follows:

First, the local Reynolds number, defined at a particular point in the flow, is given by

$$Re_y = \frac{Uy}{\nu} = U^+ y^+, \tag{66}$$

where y is the distance from the wall.

Since, in the viscous sublayer, a linear relationship exists between U^+ and y^+ as

discussed in subsection 4.1, we can relate the local Reynolds number as

$$U^+ = y^+ = (Re_y)^{1/2}, \tag{67}$$

which is valid for $Re_y \lesssim 10$.

Similarly, in the logarithmic layer, the following relationship was obtained:

$$\kappa E Re_y = E y^+ \log\left(E y^+\right), \tag{68}$$

where $E = 11.27$ and κ is the von Kármán constant.

Substituting $x = \kappa E Re_y$ and $W = \log(E y^+)$, equation 68 can be rewritten using the Lambert W function:

$$x = W(x) e^{W(x)}, \tag{69}$$

where $W(x)$ is the Lambert W function. It can be approximated using the following expression:

$$W(x) \approx \log\left(\frac{x}{\log\left(\frac{x}{\log(x)}\right)}\right) = \log(x) - \log\left(\log(x) - \log(\log(x))\right). \tag{70}$$

Finally, using the definition $W = \log(E y^+)$, the relation in the logarithmic layer becomes

$$y^+(Re_y) = \frac{1}{E} e^{W(\kappa E Re_y)}. \tag{71}$$

Equations 67 and 71, corresponding to the viscous sublayer and logarithmic layer, respectively, are then smoothly blended to yield an explicit formulation for y^+ as a function of the local Reynolds number. This expression, proposed in [4], takes the form:

$$\begin{aligned} y^+(Re_y) = & \left[1 - \tanh\left(\frac{Re_y}{180.8}\right)\right]^{0.789} (Re_y)^{1/2} \\ & + \left[\tanh\left(\frac{Re_y}{180.8}\right)\right]^{0.789} \cdot \frac{1}{E} e^{W(\kappa E Re_y)}. \end{aligned} \tag{72}$$

Utilising Equation 72, the required friction velocity can then be readily calculated from its definition as

$$u_\tau = \frac{y^+(Re_y)\,\nu}{y}. \tag{73}$$

Equations 65 and 72 are the main ingredients required for the implementation.

4.3.2 Immersed Boundary Method

After establishing the explicit wall functions, which theoretically perform well even when the boundary node is situated in any sublayer of the TBL, we discuss the Immersed Boundary Method (IBM). IBM utilises a simple Cartesian grid covering the entire geom-

etry, allowing for straightforward meshing without the need for tedious body-fitted mesh generation.

Typically, either a layer of solid nodes within the geometry is used to implement various bounce-back schemes, or the wall effect is directly imposed on the fluid nodes adjacent to the wall. This approach can be categorised into two common variants [4]: first, the *sharp interface IBM*, where wall effects are computed and enforced at the first off-wall node; and second, the *diffuse interface IBM*, in which wall effects are distributed across a band of fluid nodes near the boundary.

The former approach was implemented in this study, as it provides greater accuracy in wall modelling, particularly for flows at very high Reynolds numbers.

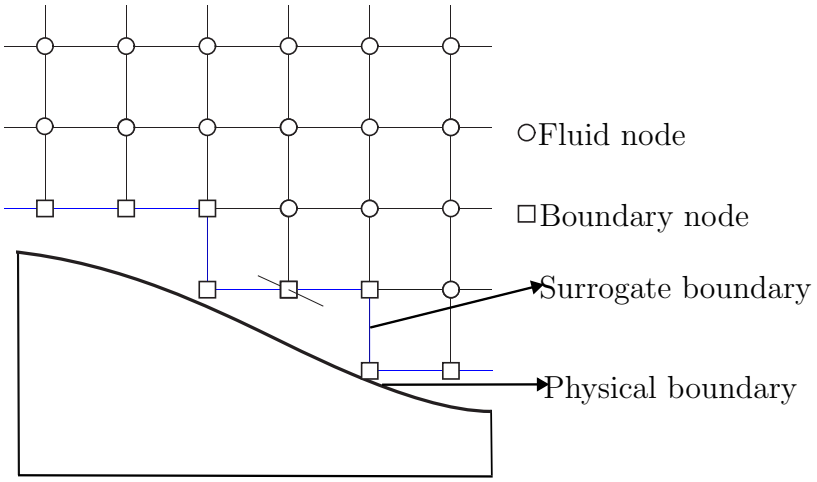


Figure 11: Illustration of the immersed boundary method showing a Cartesian grid over an arbitrary solid surface. Grid nodes inside the solid are removed, and the first layer of fluid nodes adjacent to the wall forms the surrogate boundary, consisting of boundary and fluid nodes. Adapted from [44].

The *sharp interface IBM* employed in this study requires precise identification of different types of lattice nodes. In Figure 11, an arbitrary solid geometry is overlaid with uniform Cartesian mesh. The interior solid nodes—those entirely contained within the solid object—are omitted as they are not required for the type of wall model considered here.

All remaining nodes belong to the fluid domain; however, a distinction must be made between the two subcategories. First, the *boundary nodes* are defined as fluid nodes that share at least one face with a neighbouring solid node. These form the first layer of fluid nodes adjacent to the physical boundary and collectively define the *surrogate boundary*. All wall modelling procedures are applied at these surrogate boundary nodes.

Second, the *fluid nodes* are fully embedded within the fluid domain and are not directly adjacent to any solid nodes. These nodes evolve strictly according to the fluid solver without requiring special treatment of the wall. The classification is illustrated in

Figure 11.

4.3.3 Near-Wall Interpolation Scheme

To evaluate the wall-tangential velocity at a boundary node using an expression such as Equation 65, a valid estimate of the friction velocity u_τ at that time step is required. In several studies, such as [8, 46], the velocity field was first interpolated at a reference point located at a fixed wall-normal distance. Using this interpolated velocity and the modelling height, the friction velocity is then computed by either iteratively solving one of the wall models introduced in Subsection 4.2 or by employing explicit expressions, as discussed in Section 4.3.1.

Once the friction velocity is determined, it can be mapped back to the corresponding boundary node located on the same wall-normal line from the reference point because u_τ is inherently a wall-shear-based quantity and is assumed to be constant in the wall-normal direction.

However, direct velocity interpolation in the near-wall region induces oscillations in the interpolated velocity field. These instabilities may arise owing to the strongly nonlinear nature of the near-wall velocity profile or from inaccuracies in the interpolation scheme itself, as noted in [47].

To address this, an alternative interpolation approach based on the friction velocity was proposed in [44] and adopted in this study. Interpolating u_τ instead of the velocity offers several advantages: first, it has a smoother variation across the domain [47]; second, because u_τ is assumed to be constant along the wall-normal direction by definition, performing an interpolation weighted by the inverse of the wall-tangential distance becomes physically meaningful and more robust.

A simple two-dimensional example of the friction velocity interpolation procedure is illustrated in Figure 12. A local stencil centred on the current boundary node is considered, which typically extends to two layers of surrounding nodes. Every neighbouring node within the stencil that is neither a solid node (not shown in the Figure) nor another boundary node is treated as a donor node.

For each valid donor node, the friction velocity $u_{\tau,i}$ is computed using the explicit expression derived earlier. These values are then interpolated to the boundary node location $\mathbf{x}_\mathrm{b}$ using weights based on the inverse of the squared tangential distances:

$$u_{\tau,\mathrm{b}} = \frac{\sum_i u_{\tau,i}\, a_i}{\sum_i a_i}, \quad \text{with} \quad a_i = \frac{1}{\left\| (\mathbf{x}_i - \mathbf{x}_b) - (\mathbf{x}_i - \mathbf{x}_b) \cdot \mathbf{n}_i \right\|^2}. \tag{74}$$

Here, i denotes the index looping over all valid donor nodes, $\mathbf{x}_i$ is the position of the donor node, and $\mathbf{n}_i$ is the wall-normal unit vector at the node.

It is important to note the details of this study: because the validation is performed using a turbulent channel flow with flat walls, the wall-normal vector $\mathbf{n}_i$ remains constant. However, this can be easily extended to curved walls by replacing the surface normals of

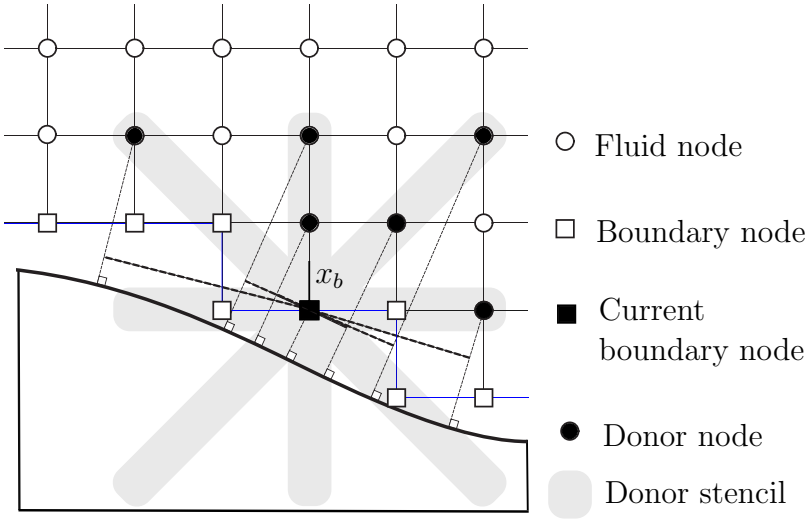


Figure 12: Stencil used to determine donor nodes for interpolating friction velocity at a boundary node. The example highlights a two-layer neighbourhood on a Cartesian grid, where the donor nodes are defined as fluid nodes that are neither solid nor boundary nodes. Adapted from [44].

the geometry.

Moreover, the density at the boundary node is still required to complete the reconstruction of the distribution function. Under the assumption of isothermal flow and negligible pressure gradients in the wall-normal direction, the density can also be considered constant along the wall-normal direction, similar to the friction velocity [44]. Therefore, the same interpolation scheme can be applied to the density using inverse tangential distance weights:

$$\rho_\mathrm{b} = \frac{\sum_i \rho_i \, a_i}{\sum_i a_i}, \quad \text{with} \quad a_i = \frac{1}{\left\| (\mathbf{x}_i - \mathbf{x}_b) - (\mathbf{x}_i - \mathbf{x}_b) \cdot \mathbf{n}_i \right\|^2}. \tag{75}$$

4.3.4 Near-wall turbulence modelling

In WM-LES, as previously discussed, turbulent structures close to the wall are not resolved but are modelled, which means that turbulence modelling should be considered separately from the bulk of the flow close to the wall. This modelling directly impacts the near-wall velocity fluctuations and mean velocity profile, which are difficult to predict, as shown in [48].

In this study, a hybrid RANS–LES approach was employed to ensure the correct modelling of eddy viscosity in the near-wall region, consistent with the explicit wall functions derived from the RANS momentum equation [4]. Specifically, the RANS-computed wall shear stress was imposed at the modelling height, which corresponded to the height of the boundary node. This zonal approach mitigates the highly dissipative behaviour of the LES in the immediate vicinity of the wall. Without such treatment, a coarse grid resolution near the wall would lead to significant subgrid-scale (SGS) errors. By applying

a more stable RANS model closer to the wall, these errors can be effectively avoided. The application of hybrid RANS/LES strategies has been extensively studied in the context of body-fitted grids for traditional Navier–Stokes solvers [49].

For simple implementation, a mixing-length model is employed in the RANS zone, defined as

$$\nu_T = (\kappa l_{\text{mix}})^2 \, |S|, \tag{76}$$

where ν_T is the turbulent eddy viscosity, κ is the von Kármán constant, l_{mix} is the mixing length (i.e. wall distance), and $|S|$ is the magnitude of the strain-rate tensor.

To accurately capture the near-wall behaviour, a damping function was incorporated into the model. This leads to the formulation given in Equation 64, as discussed in Section 4.3.1, which is the version implemented in this study.

The blending between the RANS and LES turbulence models can be incorporated using various strategies [4], and here, the empirical blending model is implemented which was presented by Kawai and Larsson [50]. This is a linear function that blends the two regions and is given as

$$\beta = 1 - K^{\gamma}, \quad K = \max\left[\min\left(\frac{y_{bl} - y}{y_{bl} - y_c}, 1\right), 0\right], \quad y_c = \alpha y_{bl}, \tag{77}$$

$$\nu_t^{\text{hybrid}} = \beta \nu_t^{\text{LES}} + (1 - \beta)\nu_t^{\text{RANS}}. \tag{78}$$

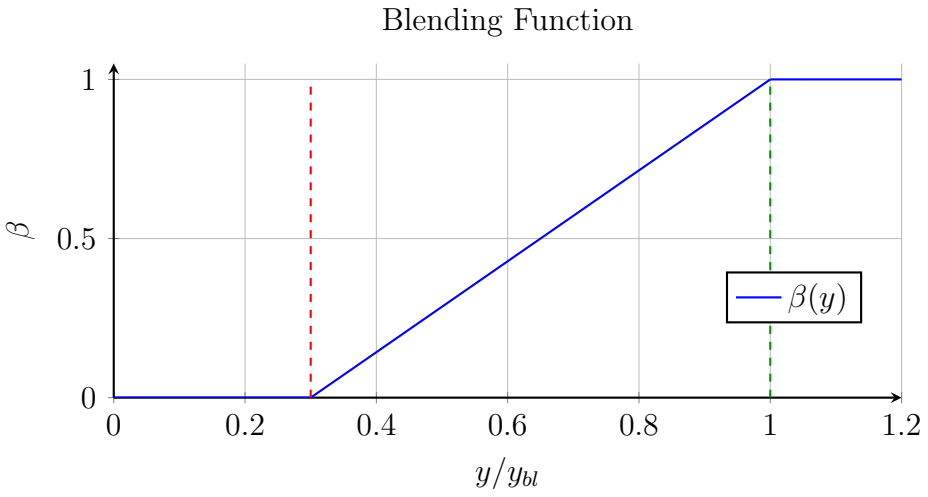


Figure 13: Blending function $\beta(y)$ used in hybrid RANS–LES models. The parameter y_{bl} denotes the blending height, representing the total thickness of the region over which the transition from RANS ($\beta = 0$) to LES ($\beta = 1$) occurs. Blending begins at $y_c = \alpha y_{bl}$, where α defines the relative start of the transition. Beyond y_{bl}, the function remains constant at $\beta = 1$, indicating a fully LES-resolved region.

4.3.5 Application of the Wall Model to LBM

Thus far, only the behaviour of macroscopic quantities has been discussed. However, to incorporate a wall model within the LBM, it is necessary to establish a link between the modelled macroscopic quantities and particle distribution functions. This connection can be achieved through various methods. In this study, the particle distribution functions were entirely reconstructed at the boundary node, discarding all distributions that would otherwise arrive at the wall.

As already discussed in Chapter 2, the particle distribution function $f_i(t, \mathbf{x})$ can be decomposed into two components: the equilibrium part f_i^{eq} and the non-equilibrium part $f_i^{(1)}$.

$$f_i(t, \mathbf{x}) = f_i^{\text{eq}}(\rho, \mathbf{u}) + f_i^{(1)}$$

The equilibrium part depends only on the local macroscopic density $\rho(\mathbf{x}, t)$ and velocity $\mathbf{u}(\mathbf{x}, t)$, which is calculated from the interpolation and application of the wall function.

The simplest scheme, referred to here as the equilibrium scheme [9] and denoted in [51], assumes that the non-equilibrium component can be neglected entirely. This implies that the distribution function at the boundary is set equal to its equilibrium counterpart.

In more advanced schemes, the non-equilibrium part of the distribution can be recovered in multiple ways.

1. It can be simply extrapolated from a neighbour node in the wall-inward direction, as originally proposed in [52] and later applied in [9]. This extrapolation uses the modelled macroscopic quantities at the wall to reconstruct the equilibrium component and copies the non-equilibrium part from the adjacent fluid node:

$$f_i(t, \mathbf{x}) = f_i^{\text{eq}}(\rho, \mathbf{u}) + f_i^{(1)}\left(t, \mathbf{x} - |\mathbf{c}_j|\mathbf{n}\right), \tag{79}$$

   where $\mathbf{u}$ is the wall model velocity (typically tangential) at the boundary node, ρ is the density, and $\mathbf{n}$ is a unit wall-normal vector. Here, $\mathbf{c}_j$ refers to the lattice velocity vector opposite to the direction i. This expression effectively pulls the nonequilibrium contribution from the adjacent donor node while applying the correct wall-modelled equilibrium part.

2. Alternatively, the non-equilibrium part can be reconstructed entirely from lower-order moments, as discussed in Chapeter 2. The regularisation approach requires wall-parallel velocity and local strain rate tensor. A similar boundary reconstruction method based on HRR was adopted in [44].

In [4], the second option—reconstructing the nonequilibrium part of the distribution using finite difference approximations of the strain rate tensor—was implemented. However, the interpolation-based approach for estimating the non-equilibrium part is simpler

to implement. To the best of the author's knowledge, the zeroth-order extrapolation method proposed in [52] for the non-equilibrium distribution has not yet been tested in conjunction with friction velocity interpolation. Therefore, this combination was investigated in the present study.

5 Implementation of the Wall-Modelling Algorithm

The implementation of the wall model algorithm was performed in `C++` using an object-oriented programming (OOP) paradigm. This approach offers several advantages in the current context, including the structured management of variables and functions, ease of handling different node types, and enhanced code reusability. `C++` is inherently object-oriented and provides increased memory control and flexibility, making it well-suited for the demands of this thesis.

An existing turbulent channel flow solver was extended to incorporate a wall-modelling algorithm, along with several related enhancements. Additional features required were, for example, the calculation of moving averages and variance, the ability to restart the channel flow simulation from previously saved distribution values, and control over the data collection period.

The theoretical basis of the implemented wall modelling algorithm, as discussed in Chapter 4, is summarised here, along with practical implementation details essential for replicating this work.

5.1 Concept of the Turbulent Channel Flow Programme

The primary objective of the program is to implement and evaluate a robust and stable wall model in the context of a canonical turbulent channel flow. This case study provides a sound foundation for understanding the theory and implementation, enabling the future integration of the algorithm into a more advanced CFD tool based on the LBM coupled with the IBM.

The simulation framework adheres to the standard CFD workflow comprising preprocessing, solving, and postprocessing, as outlined below.

Preprocessing Given the simplicity of the channel geometry, the discretisation and initialisation of the nodes are relatively straightforward. All physical and numerical parameters, such as density, viscosity, lattice Mach number, grid spacing, and wall modelling settings, were defined in a text-based configuration file (`Boundaryconditions.txt`). Using these inputs, the simulation domain was generated, and the initial conditions were

applied accordingly.

Solving The numerical scheme employed was the Lattice Boltzmann Method augmented with LES turbulence modelling and the implemented wall model. The solver is inherently transient and progresses iteratively for a user-defined number of time steps, as follows: the simulation output is written at specified intervals and includes both instantaneous macroscopic fields and statistical averages. The latter were computed using an online averaging technique based on Welford's method, which ensured numerical stability and minimal memory usage.

Postprocessing This study focuses on near-wall quantities, such as the mean velocity profile and turbulence intensities, and their comparison with reference DNS data. A dedicated Python script was developed to postprocess the solver output. It performs surface averaging of the time-averaged results and plots them alongside the benchmark DNS data corresponding to the relevant friction Reynolds number. Additionally, the script includes functionality for temporal Fourier transforms to evaluate the energy spectra and investigate the turbulent energy cascade.

Figure 14 illustrates the various processes of the program in the form of a flowchart. The overall workflow can be categorised into distinct stages. As previously discussed, the input handling, preprocessing, and postprocessing components are relatively self-explanatory in this study.

The remainder of this chapter is dedicated to the implementation of the wall-modelling algorithm within the solver, providing detailed insight into its structure, functionality, and integration with the existing codebase.

5.2 Solver

5.2.1 Solving the LBE

As shown in Figure 14, the solver module is structured around a time-stepping loop that solves the Lattice Boltzmann Equation (LBE). This is achieved by decomposing the LBE into a two-step process: (1) the transport step and (2) the collision step. For the case of a Single Relaxation Time (SRT) collision model, the formulation is as follows:

$$\text{Transport:} \quad \tilde{f}_i\left(\mathbf{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t\right) = f_i(\mathbf{x},t), \tag{80}$$

$$\begin{aligned}\text{Collision:} \quad f_i\left(\mathbf{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t\right) &= \tilde{f}_i\left(\mathbf{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t\right) \\ &\quad + \Omega\left[\tilde{f}_i^{eq}\left(\mathbf{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t\right) - \tilde{f}_i\left(\mathbf{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t\right)\right].\end{aligned} \tag{81}$$

The first step, commonly referred to as the *streaming* or *push* step, involves transferring the distribution functions to neighbouring lattice nodes along their respective discrete

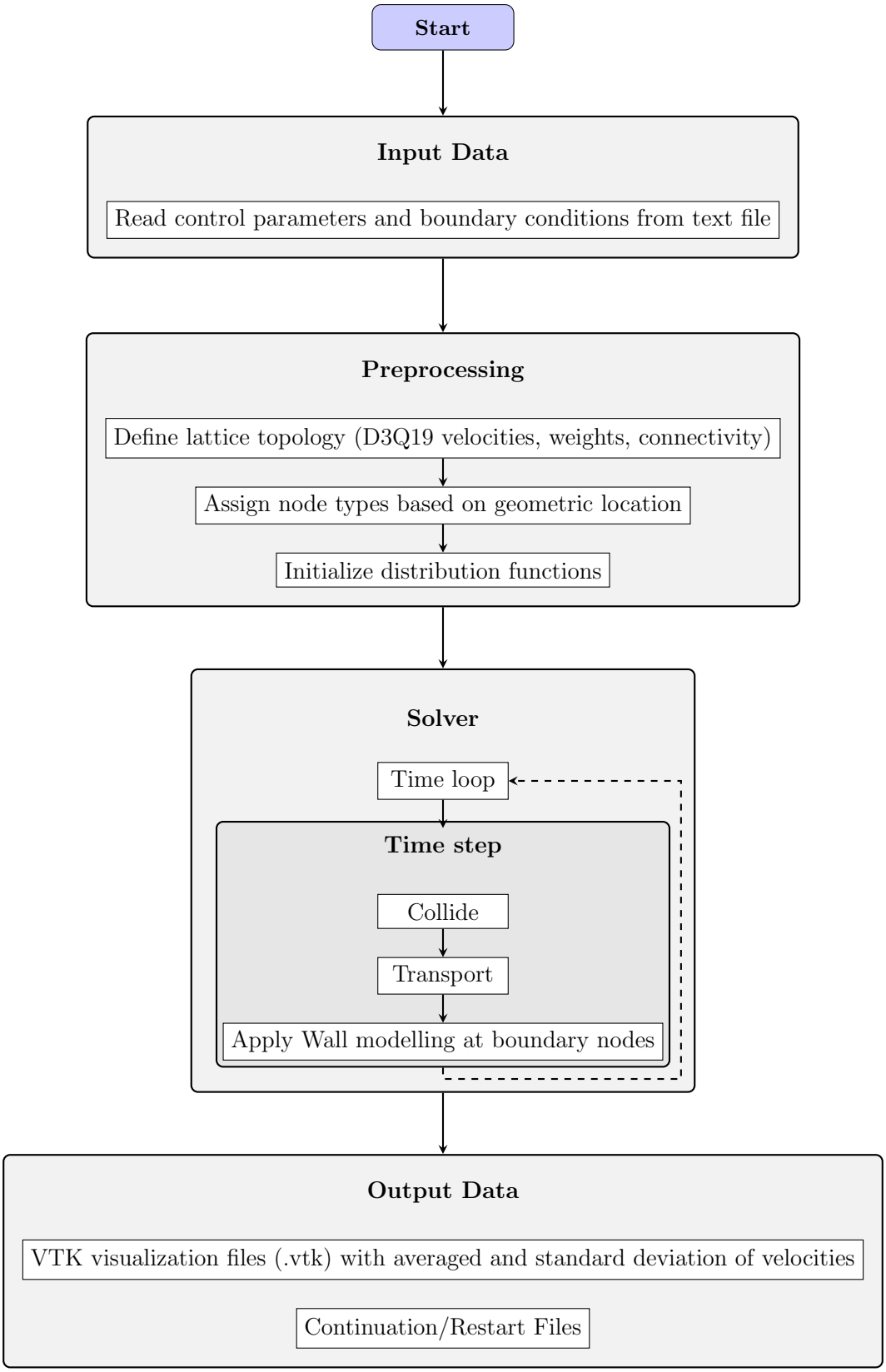


Figure 14: High-level flowchart of the LBM solver.

velocity directions. This step is purely data movement and does not introduce numerical errors, making it advantageous in terms of numerical dissipation.

The second step, the *collision* step, is a local operation, wherein the distribution functions are relaxed towards their equilibrium states. The locality of this operation makes it highly efficient and naturally parallelizable. Additionally, because the overall scheme is explicit in time, there is no need to solve linear systems or perform matrix inversions, resulting in a straightforward and efficient implementation. The details of the LBM solver implementation can be found in [27].

5.2.2 Wall Modelling

The wall modelling approach adopted in this study involved the complete reconstruction of all distribution functions at the boundary nodes, effectively discarding the incoming distributions resulting from the transport step. This reconstruction process is inherently nonlocal and requires the interpolation and evaluation of the wall function. Although the general approach could involve solving implicit equations iteratively, an explicit wall function was selected here to simplify the implementation and reduce the computational cost. The workflow of this wall modelling scheme is illustrated in detail in Figure 15.

Because distribution reconstruction must be performed for every boundary node, the algorithm begins with a loop over all such nodes. For each boundary node, the donor nodes are first identified using a predefined stencil, as shown in Figure. 12. This is facilitated by the program's object-oriented structure, where nodes are categorised into fluid nodes (those within the channel), periodic nodes (for periodic boundary conditions), and first off-wall nodes (which are technically fluid nodes, but serve as surrogate boundaries where reconstruction occurs). The donor-finding routine iterates over two layers of neighbouring nodes surrounding the first off-wall node and uses dynamic type casting to verify whether each neighbour is a valid fluid node. Only valid fluid nodes were considered donor nodes.

Once all donor nodes for a given boundary node are identified, the wall friction velocity u_τ is computed at each donor location using the wall model. Here, the implementation decision was made to first calculate and store all the friction velocities of the donor nodes and then interpolate them to avoid recalculations, as the boundary nodes share many donor nodes. This process is illustrated in detail in Figure 16. Notably, each donor node may have a different wall-normal distance, and the friction velocity is evaluated individually at each location to reflect this variation. These friction velocities were then interpolated tangentially along the wall, allowing for a smooth and physically realistic estimation of u_τ even in high Reynolds number regimes. During interpolation error handling, the zero tangentail distance must be implemented, as a donor node perpendicular to the boundary node with respect to the wall will cause a zero tangentail distance, causing a division by zero error in the interpolation. The density was interpolated similarly.

After the friction velocity and density interpolation, the final step is the reconstruc-

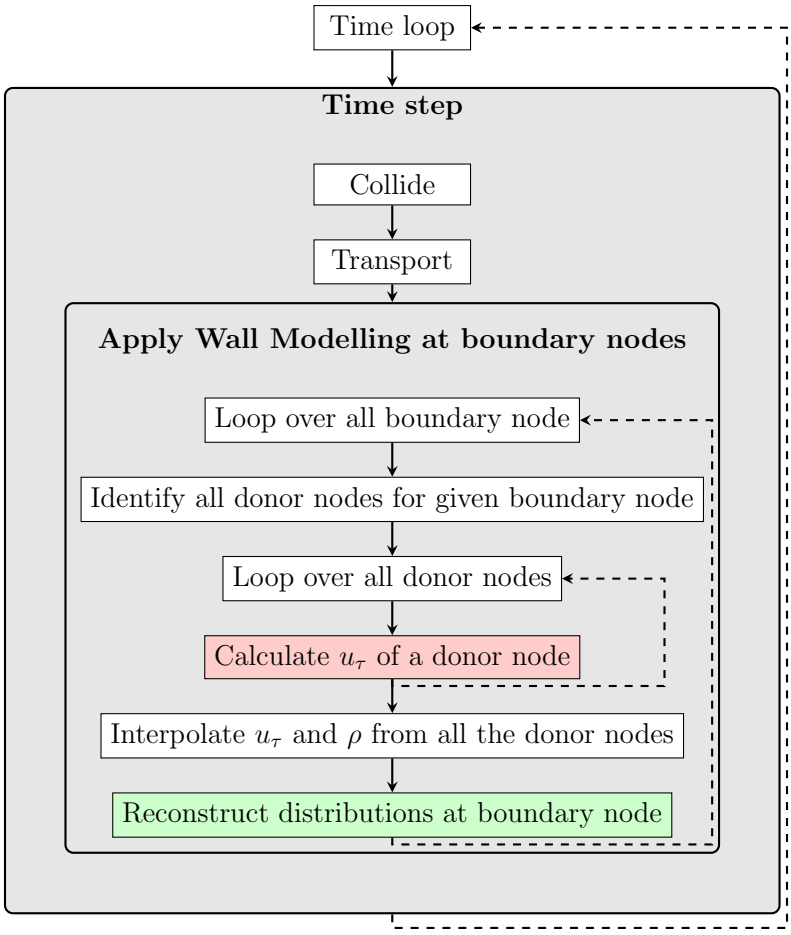


Figure 15: Detailed flowchart of wall modelling algorithm within the time step of the LBM solver.

tion of the distribution functions at the current boundary node. As discussed in earlier chapters, the distribution function is conceptually decomposed into two components: an equilibrium part and a non-equilibrium part.

The equilibrium component is computed based on the wall-modelled macroscopic boundary velocity obtained from the interpolated friction velocity and the interpolated density. This procedure is depicted on the left branch of Figure 17.

The non-equilibrium component can be computed using different approaches, as discussed in Section 4.3.5. In the current implementation, a method based on extrapolation from the inward wall normal is used owing to its ease of implementation. This procedure is summarised in the right branch of Figure 17. The complete reconstruction algorithm is applied to all boundary nodes in each time step and repeated until the specified number of time iterations is reached.

Further details of the turbulence model are not included here, as the description in Section 4.3.4 is sufficient for implementation.

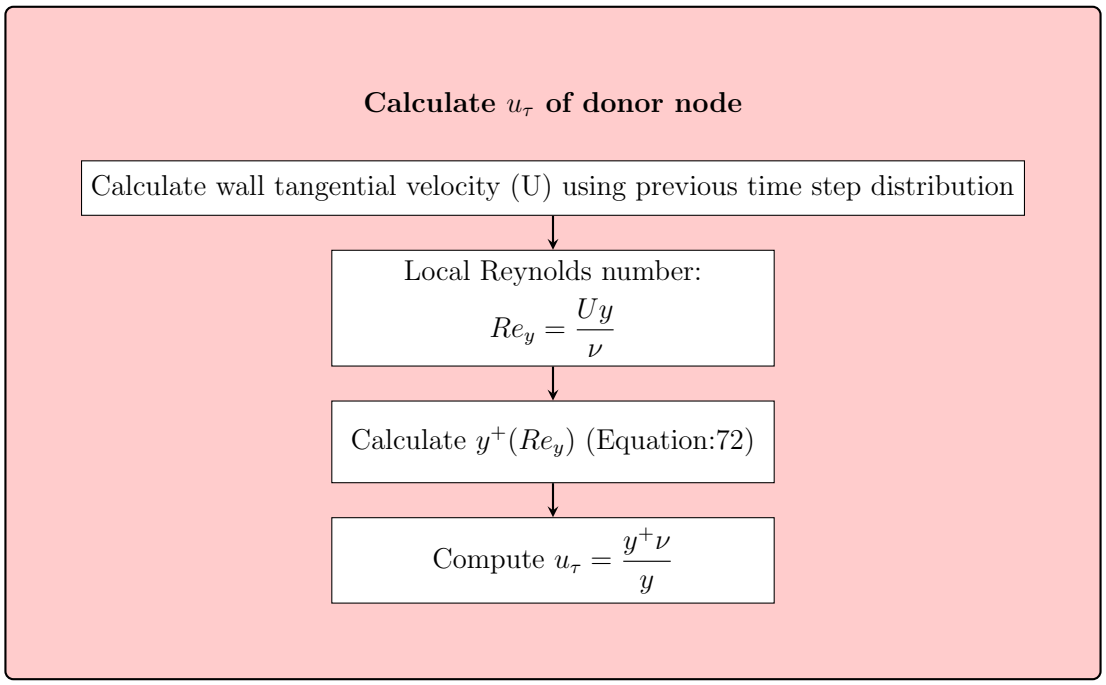


Figure 16: Sequential flowchart of friction velocity calculation using an explicit wall model.

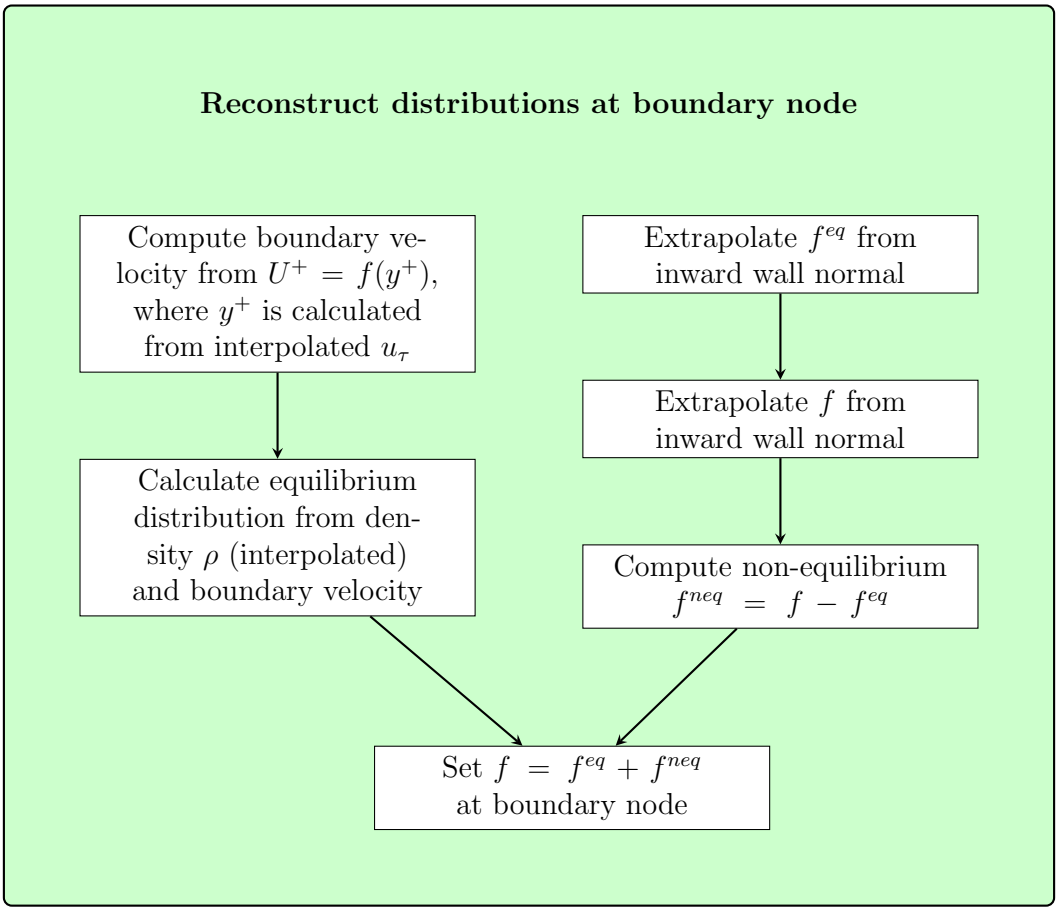


Figure 17: Flowchart for reconstructing distribution at the boundary node using interpolated values and neighbour information. Two branches feed into the final distribution composition.

6 Turbulent Channel Flow Verification Case

After the successful implementation of the numerical scheme, verification is essential to demonstrate that the method accurately solves the underlying governing equations. In this study, the wall modelling algorithm integrated into the LBM-LES solver for turbulent channel flow was verified against a well-established DNS database, [33] to assess its accuracy and robustness. A comparison of these results can be made directly with the recent literature on WM-LES, such as [4], [9] and [53].

The verification process is presented in two sections. First, the performance of the wall-modelled LBM-LES was evaluated across a range of Reynolds numbers and grid resolutions. Second, two different approaches for reconstructing the non-equilibrium part of the distribution function were compared, providing insight into their relative accuracy.

6.1 Setup of the Wall-Modelled LBM-LES Simulations

This subsection outlines the complete setup used in the wall-modelled LBM-LES simulations. For all cases presented in this study, the standard Smagorinsky model served as the subgrid-scale turbulence model, with a Smagorinsky constant of $C_S = 0.1$. The simulations were performed on single-level uniform Cartesian grids, which means that the computational grid was not refined. Consequently, the surrogate boundary used for wall modelling was consistently positioned at a distance of $0.5\,\Delta x$ from the physical wall, where Δx denotes the grid spacing. This is similar to the case of a halfway bounce back, but can certainly be varied for complex geometries, and the effect of varying the model height was studied in [4].

The computational domain used for the simulations is shown in Figure 18. Its dimensions correspond to the minimal domain size required to sustain a fully developed turbulent channel flow, as calculated in [54]. Specifically, the domain extended $4H$ in the streamwise (x) direction and $2H$ in both the wall-normal (z) and spanwise (y) directions, where H denotes the channel half-height. The flow is driven in the x-direction by a constant body force. Periodic boundary conditions were applied in both the x and y directions, whereas no-slip wall boundary conditions were enforced on the planes normal to the z-axis.

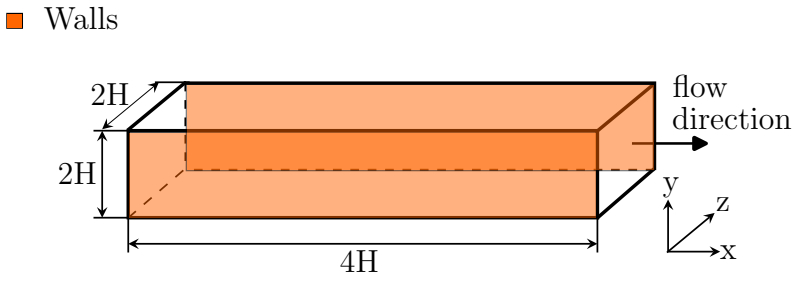


Figure 18: Schematic of the computational domain showing the dimensions of the channel, along with the flow direction and the walls.

The body force required to drive the flow is determined as

$$\mathbf{f} = \rho \mathbf{a} = -\frac{dp}{dx}\,\mathbf{x} = \frac{\tau_w}{H}\,\mathbf{x} = \frac{\rho u_\tau^2}{H}\,\mathbf{x}, \tag{82}$$

where H is the channel half-height and u_τ denotes the friction velocity, defined by the target friction Reynolds number (Eq. 84).

In this study, the driving acceleration was adjusted dynamically to reach the target friction velocity. Instead of prescribing a fixed pressure gradient, the acceleration was updated at each time step based on the deviation from the desired wall shear stress. This control approach, following the method proposed in [4], modifies the acceleration as

$$a = \frac{u_\tau^2}{H} + \frac{u_\tau - u_\tau^{\text{avg}}}{\Delta t}, \tag{83}$$

where u_τ^{avg} is the current average friction velocity and Δt is the time step. The second term acts as a feedback correction to progressively adjust the body force until the target friction velocity is achieved. This ensures that the wall model is tested for the intended u_τ.

The simulations were compared with the DNS data provided by [33], which are available for friction Reynolds numbers of 180, 950, and 2000, which are the same values employed in this study. The friction Reynolds number, denoted as Re_τ, is defined as

$$\text{Re}_\tau = \frac{H u_\tau}{\nu}, \tag{84}$$

where H is the channel half-height, u_τ is the friction velocity, and ν is the kinematic viscosity. The corresponding bulk Reynolds number can be calculated as follows:

$$\text{Re}_{\text{bulk}} = \frac{2H u_m}{\nu}, \tag{85}$$

where u_m represents the bulk mean velocity across the channel.

To ensure an accurate statistical analysis, the turbulence statistics were collected only

after the flow reached a statistically steady state, defined by the stabilisation of both the mean velocity and friction velocity.

The non-dimensionalized quantities were averaged over time and space over the entire domain. Welford's scheme, as discussed previously, was employed to perform the temporal moving average and compute the corresponding fluctuations.

6.2 Reynolds Number Studies

As discussed in Chapter 4, there are two distinct approaches for reconstructing the non-equilibrium part of the distribution function. These are compared in the following section. The method based on extrapolating the non-equilibrium distribution is referred to as f_{neq}-*extrapolation*, while the alternative approach using finite differences is termed f_{neq}-*finite difference*. In this section, the results obtained using the f_{neq} extrapolation method are presented for various values of Re_τ.

Simulations were conducted at friction Reynolds numbers $Re_\tau = 950$, and 2000. The corresponding bulk Reynolds numbers, computed based on the centreline velocity and channel half-height, were approximately $Re_c \approx 35000$ to 80000.

6.2.1 Results for Wall-Modelled LES at $Re_\tau = 950$

The results of the simulation at a friction Reynolds number of 950 are presented in Figures 20–24. In all cases shown, the blending height of the RANS-LES region y_{bl} was chosen as $4\,\Delta x$, that is, the blending occurred at the first four grid points close to the wall. The plots include the mean streamwise velocity u^+ as well as the root-mean-square velocity fluctuations u'^+, v'^+ and w'^+. These are compared to the corresponding DNS reference data, as shown in Figures 20, 21, 22, and 23. The simulations were conducted on three different grid sizes, as shown in Table 1 along with their corresponding Mach numbers.

Table 1: Different grid sizes studied for $Re_\tau = 950$

Grid size	$\Delta x/m$	Ma
Coarse	0.1	0.2
Medium	0.05	0.1
Fine	0.025	0.05

Figure 19 shows the fully developed turbulence for the $Re_\tau = 950$ case.

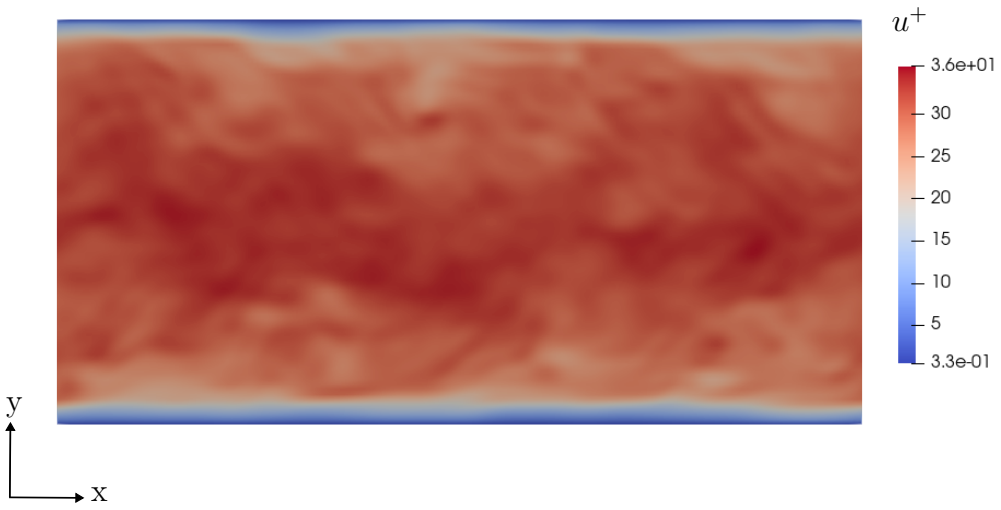


Figure 19: Non-dimensionalized instantaneous streamwise velocity contour from wall-modelled LES of turbulent channel flow at $Re_\tau = 950$.

Figure 20 shows that the streamwise velocity (u^+) in the near-wall region of both the medium and coarse grids is in good agreement with the DNS data, indicating the correct velocity and shear stress boundary conditions set by the wall model. The fine grid near the wall nodes shows that the velocity is underpredicted, which could be due to the higher error of the wall function in the buffer region. It should be noted that a continuous wall function is implemented without any switch case, and all sublayers are covered by one fitted expression. In the pure LES region, the first four near-wall nodes showed a clear over-prediction of the mean streamwise velocity, as observed in [55], and is referred to as the "log-layer mismatch". This overprediction was primarily due to the insufficiency of the LES grid to resolve the flow. The exact minimum necessary number of nodes in LES is given in [42], a grid refinement with the boundary node placed in the logarithmic layer is expected to eliminate this velocity mismatch.

In Figure 21, the fluctuations of the streamwise velocity component were in good agreement with the DNS data for the fine grid after the RANS-LES blending region. However, the near-wall fluctuation is grossly mismatched owing to the RANS turbulence model close to the wall, which suppresses near-wall fluctuations. This can be improved by studying the blending height parameters and adjusting the near-wall damping. In addition, a critical point of discussion is the overall prediction of a higher TKE than the DNS data, which is unphysical from the theoretical standpoint of an LES. This is another indication of poor grid resolution or other turbulence modelling inaccuracies.

Figure 24 shows the spectrum of the turbulent kinetic energy. The calculation is performed as follows: First, the time series of the three fluctuations of the velocity components at one particular node, in this case, close to the wall of the channel, is collected. Then, the total TKE time series for the data collection period was calculated. Then, by applying Fourier transformation to this time-series data, the transformation to the

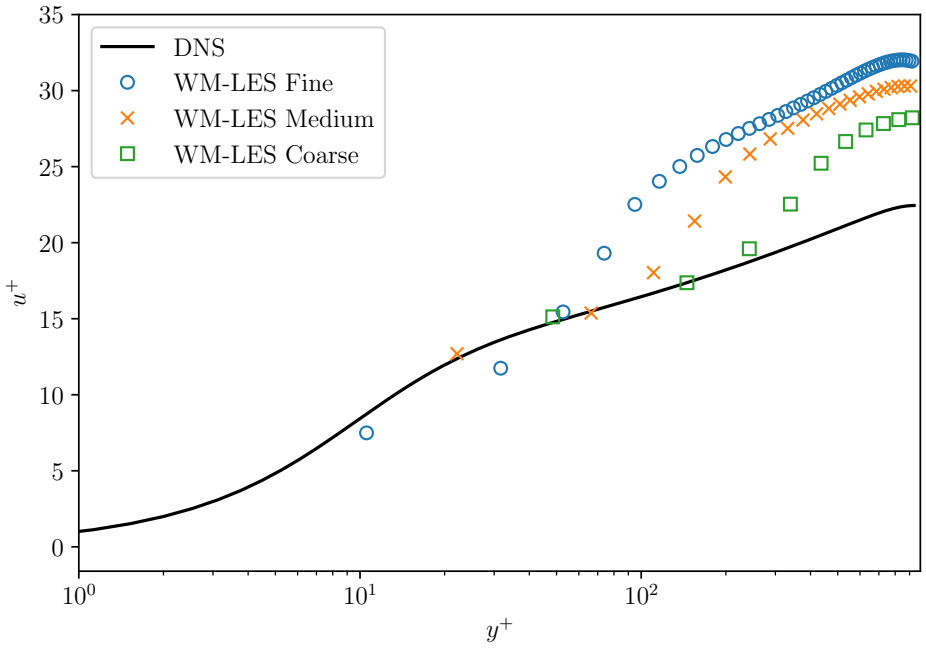


Figure 20: Temporally and spatially averaged non-dimensional streamwise velocity profiles (u^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 950$ for three different grid resolutions (coarse, medium, fine), compared with DNS reference data.

frequency domain was carried out using Taylor's hypothesis. A detailed explanation of this calculation can be found in [41]. Finally, after the transformation, the spectrum was plotted with the wave number instead of the frequency for better interpretation. The wavenumber is calculated as follows:

$$\kappa = \frac{2\pi f}{u_{\max}}. \tag{86}$$

The energy spectrum consists of three subsections [41]: the energy containing, inertial, and dissipation ranges. The inertial range of the spectrum is expected to have a slope of -5/3 on the log-log plot of the TKE, which is achieved well, as shown in Figure 24.

6.2.2 Results for Wall-Modelled LES at $Re_\tau = 2000$

With the same setup, the results for a friction Reynolds number of 2000 are presented in this section. A similar behaviour was observed in these results. In the mean steam-wise velocity plot, shown in Figure 26, the near-wall nodes agree well with the DNS data, proving the successful verification of the wall model for the $Re_\tau = 2000$ case. Combining both the $Re_\tau = 950$ and $Re_\tau = 2000$ results, we see that the wall model ensures correct velocities at the near-wall nodes, but the coarse grid fails to resolve the bulk velocity accurately, and hence an over-prediction is observed in both the cases. In addition, the velocity fluctuations for the $Re_\tau = 2000$ case, as shown in Figures. 27, 28 and 29 show the same behaviour as before, where the near-wall fluctuations are underpredicted, and

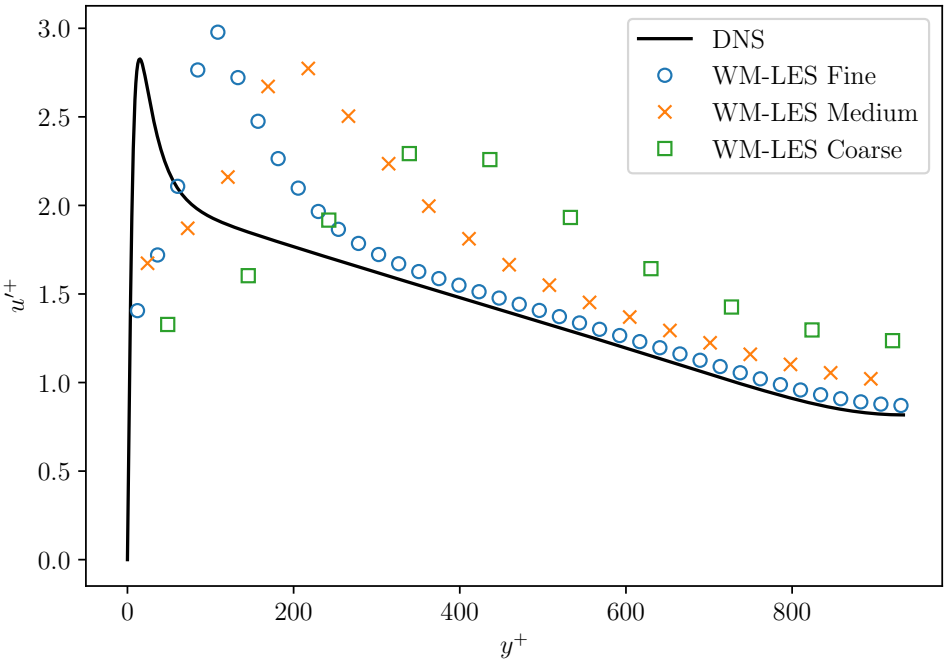


Figure 21: Streamwise velocity fluctuations (u'^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 950$ for coarse, medium, and fine grid resolutions, compared with DNS data.

the total kinetic energy is slightly higher than that of the DNS.

Table 2: Different grid sizes studied for $Re_\tau = 2000$

Grid size	$\Delta x/m$	Ma
Coarse	0.1	0.2
Medium	0.05	0.1
Fine	0.025	0.05

6.3 Evaluation of Non-Equilibrium Wall Modeling Approaches

As previously discussed in the theory of wall modelling in Chapter 4, after the reconstruction of the equilibrium part of the distribution function on the surrogate boundary using the interpolated quantities and wall function, there are various ways in which the non-equilibrium part of the distribution can be incorporated. Two such methods have been implemented in this study: one in which f_{neq} is calculated on the wall inward normal neighbour of the boundary node and simply extrapolated, and another in which it is reconstructed from the analytical expression using the strain rate tensor calculated by the finite difference method (see Eq. 21). To the best of the author's knowledge, such a comparison is not available in the literature; hence, it was performed in this study to provide a comparison between the two modelling approaches.

Figure 30 shows that the f_{neq} finite-difference method slightly underpredicts the stream-

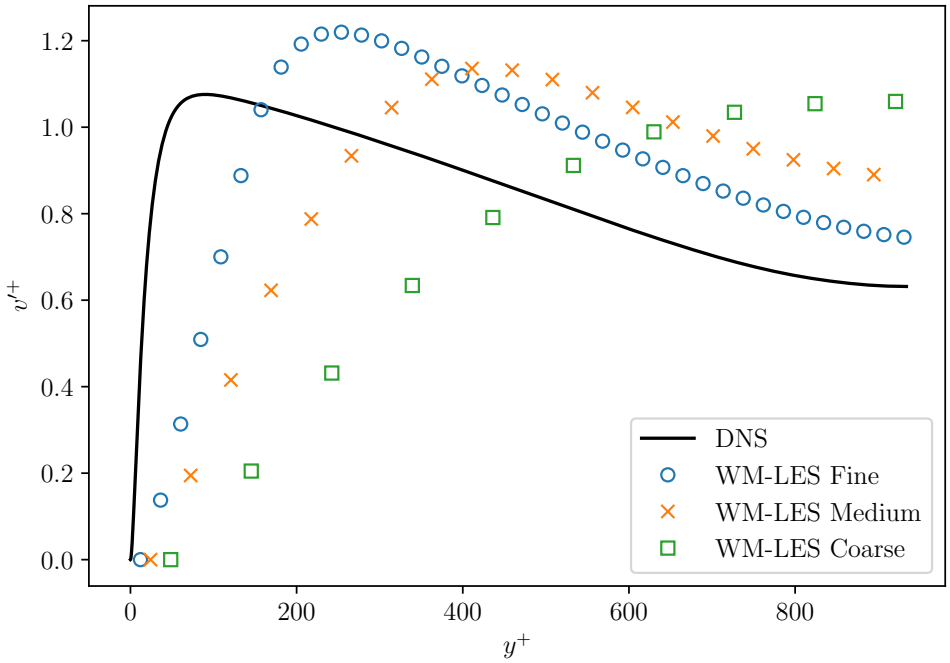


Figure 22: Spanwise velocity fluctuations (v'^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 950$ for coarse, medium, and fine grid resolutions, compared with DNS data.

wise velocity at the boundary nodes, and the f_{neq} extrapolation method shows better agreement with the DNS data. It is important to note that both methods are plotted according to the mean friction velocity achieved after the simulation reached statistical equilibrium. Because the wall model uses this friction velocity to prescribe the correct velocity at the boundary node, and this mainly depends on the equilibrium part of the distribution, both methods show correct velocities at the boundary nodes for their respective friction velocities. It was observed that the f_{neq} finite-difference method achieved a slightly smaller friction velocity, hence explaining the lesser turbulence and therefore higher velocity at the centre of the channel.

Although the Extrapolation method is more heuristic than finite-difference reconstruction, it performs better. This is because of the approximation of the strain rate tensor S_{ij}, which is calculated on a very coarse grid near the wall using the central finite difference method, and the velocity gradient is under-predicted. This underprediction of the gradient in the strain rate tensor can be corrected further using the analytical gradients in the wall-normal direction given in [35], which should improve the results. The slight under calculation of the gradients will rersult in lower total viscosity which also explains the increased fluctutations seen in Figure 31, 32 and 33. Hence, the extrapolation method proves to be a better modelling approach for boundary node reconstruction than the finite difference method, although other velocity gradient correction alternatives can be considered.

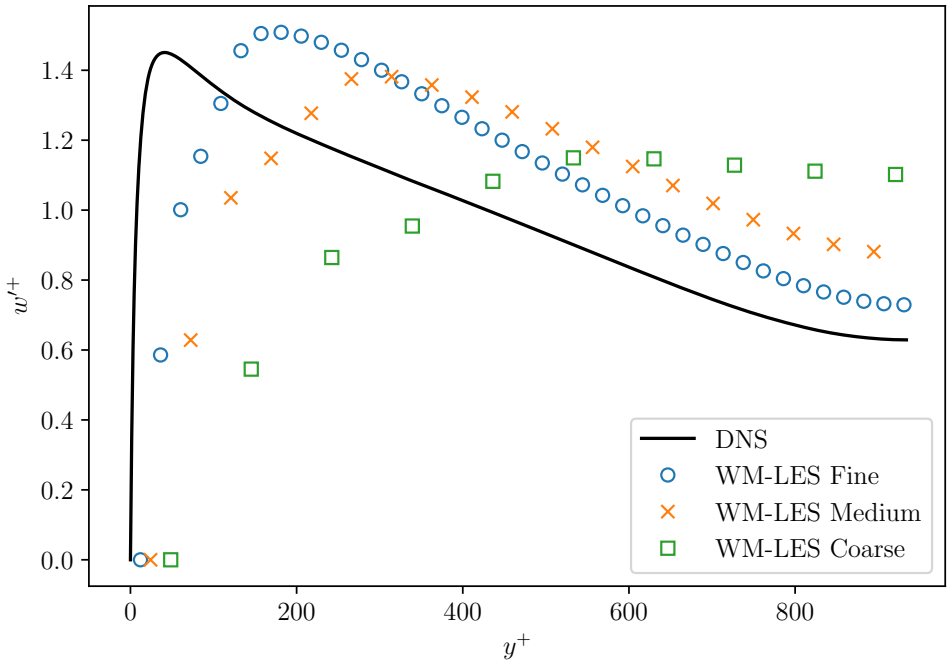


Figure 23: Wall-normal velocity fluctuations (w'^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 950$ for coarse, medium, and fine grid resolutions, compared with DNS data.

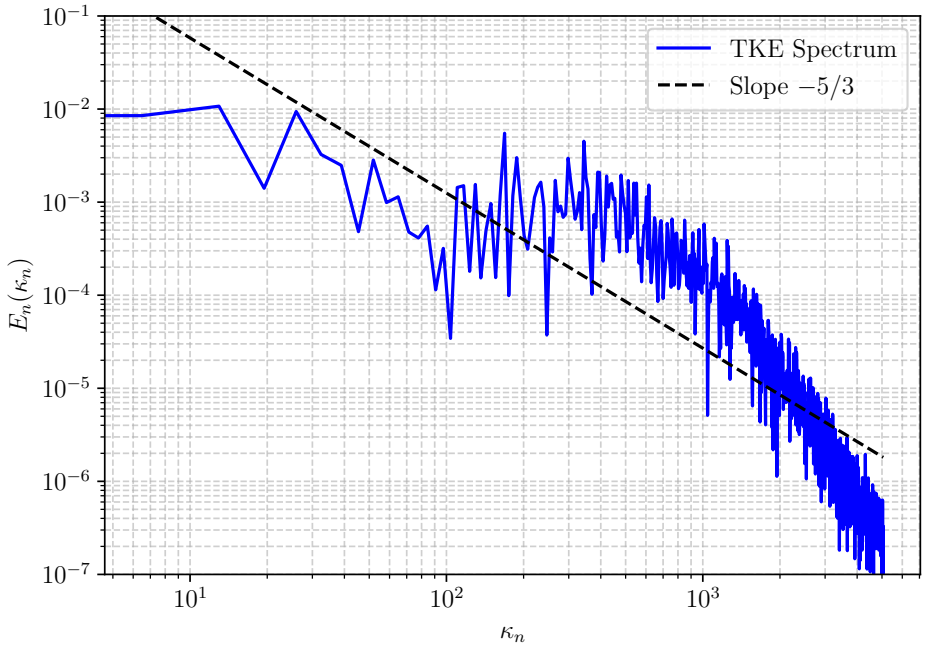


Figure 24: Turbulent kinetic energy (TKE) spectrum obtained from a direct Fast Fourier Transform (FFT) of the velocity time series at the channel center in a wall-modelled LES of turbulent channel flow at $Re_\tau = 950$. The spectrum is non-dimensionalized and compared against the expected $-5/3$ slope in the inertial subrange.

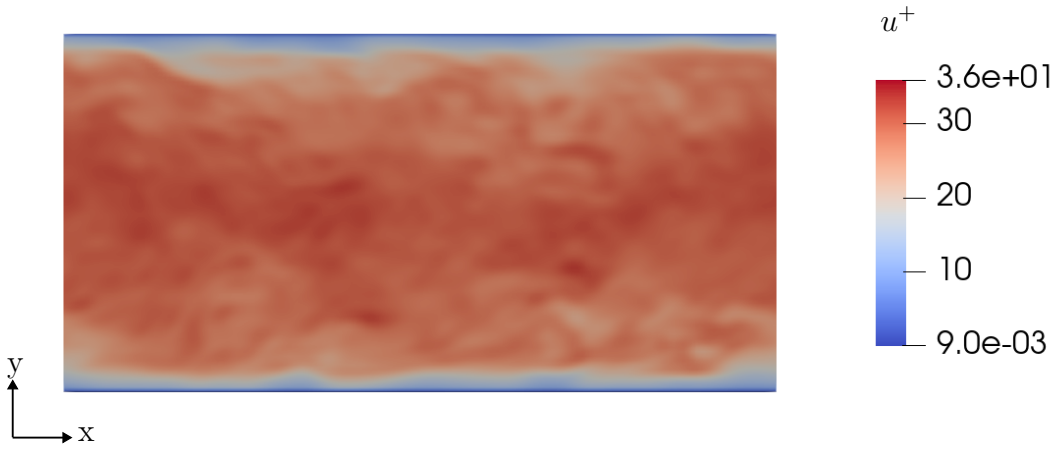


Figure 25: Non-dimensionalized instantaneous streamwise velocity contour from wall-modelled LES of turbulent channel flow at $Re_\tau = 2000$.

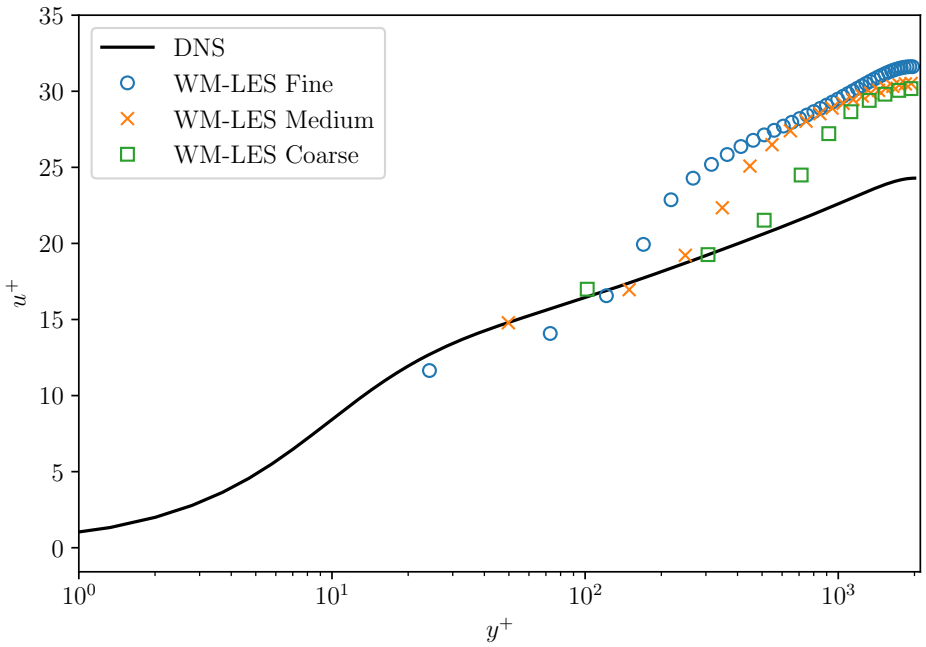


Figure 26: Temporally and spatially averaged non-dimensional streamwise velocity profiles (u^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 2000$ for three different grid resolutions (coarse, medium, fine), compared with DNS reference data.

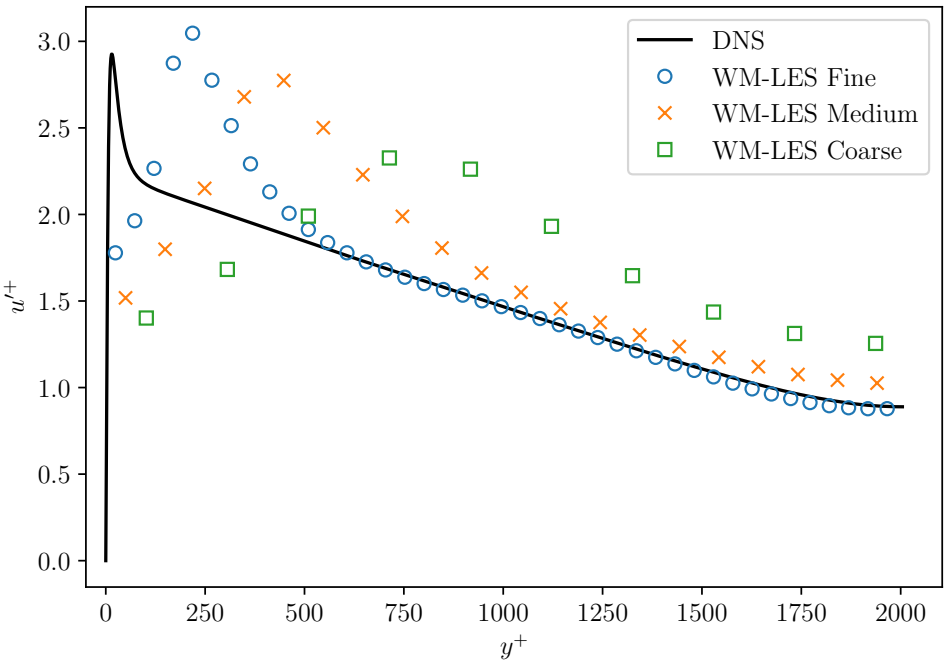


Figure 27: Streamwise velocity fluctuations (u'^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 2000$ for coarse, medium, and fine grid resolutions, compared with DNS data.

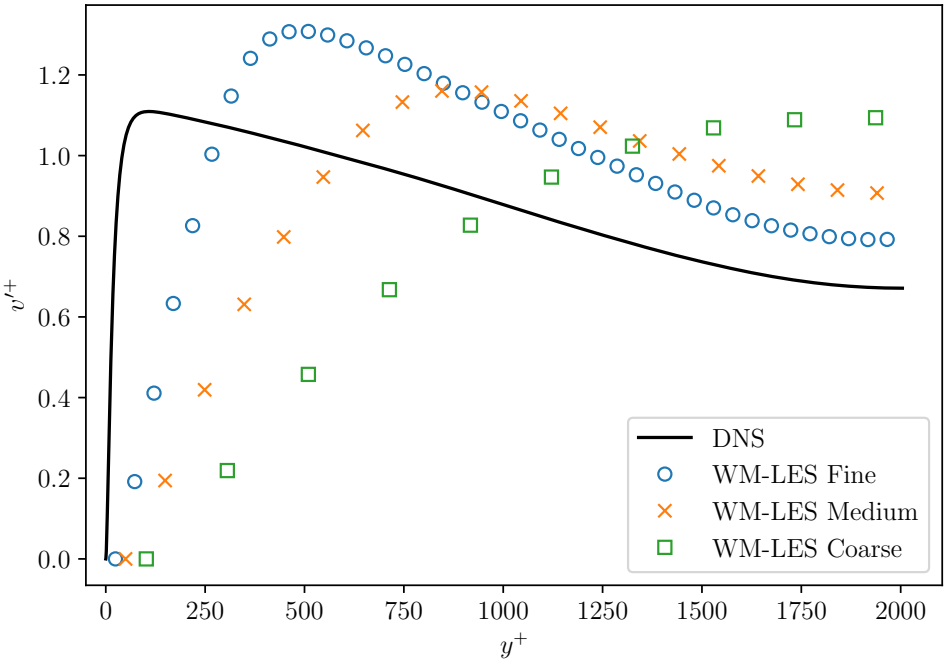


Figure 28: Spanwise velocity fluctuations (v'^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 2000$ for coarse, medium, and fine grid resolutions, compared with DNS data.

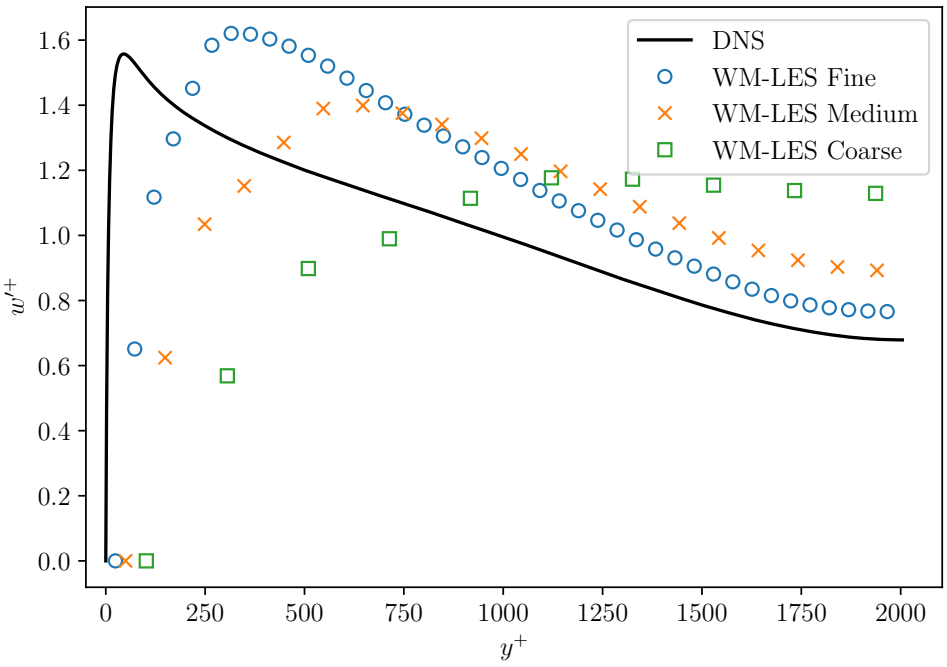


Figure 29: Wall-normal velocity fluctuations (w'^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 2000$ for coarse, medium, and fine grid resolutions, compared with DNS data.

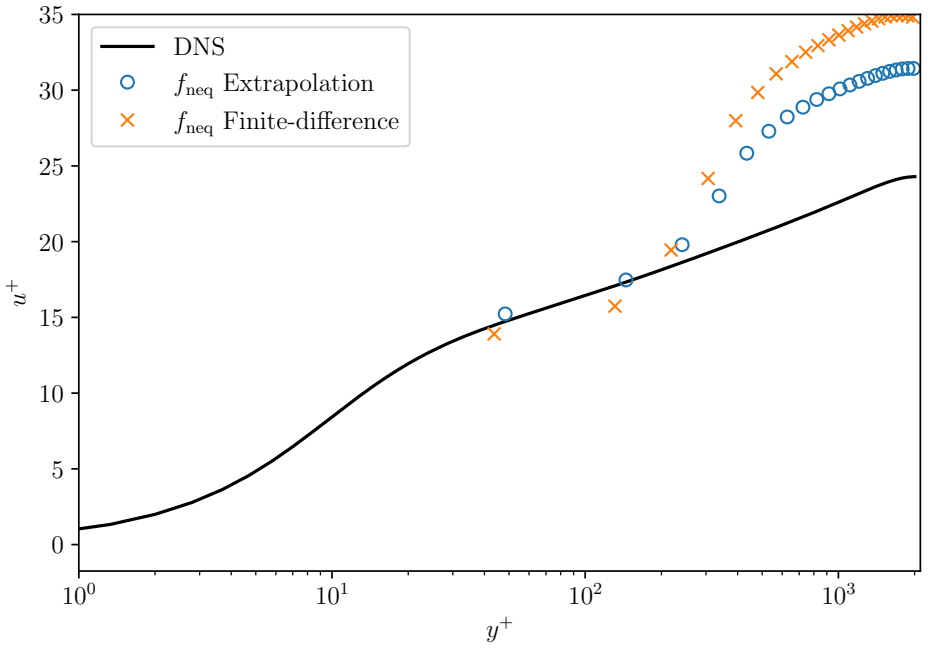


Figure 30: Temporally and spatially averaged non-dimensional streamwise velocity profiles (u^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 2000$ for three different grid resolutions (coarse, medium, fine), compared with DNS reference data.

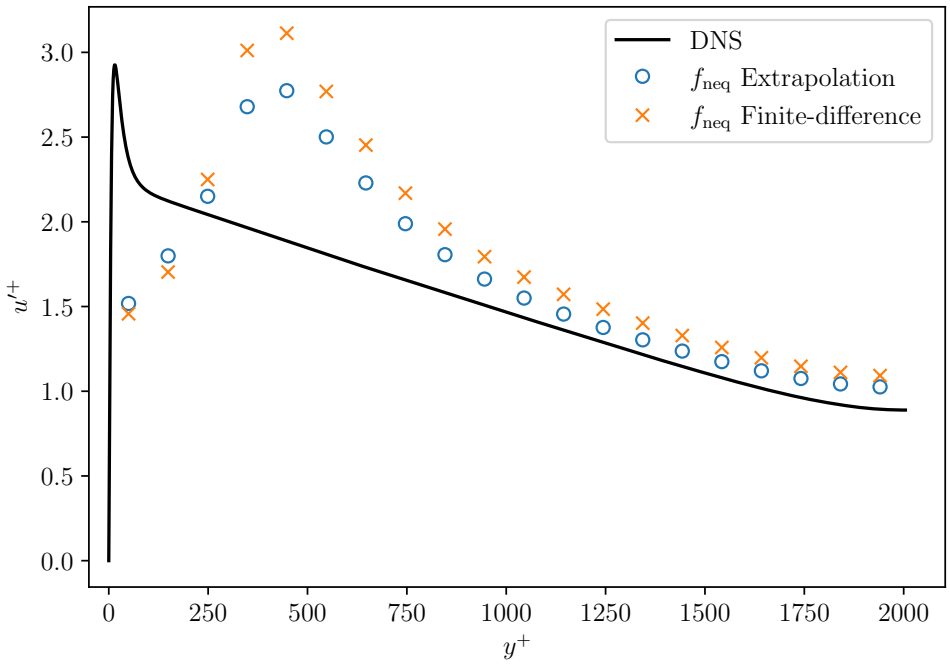


Figure 31: Streamwise velocity fluctuations (u'^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 2000$ for coarse, medium, and fine grid resolutions, compared with DNS data.

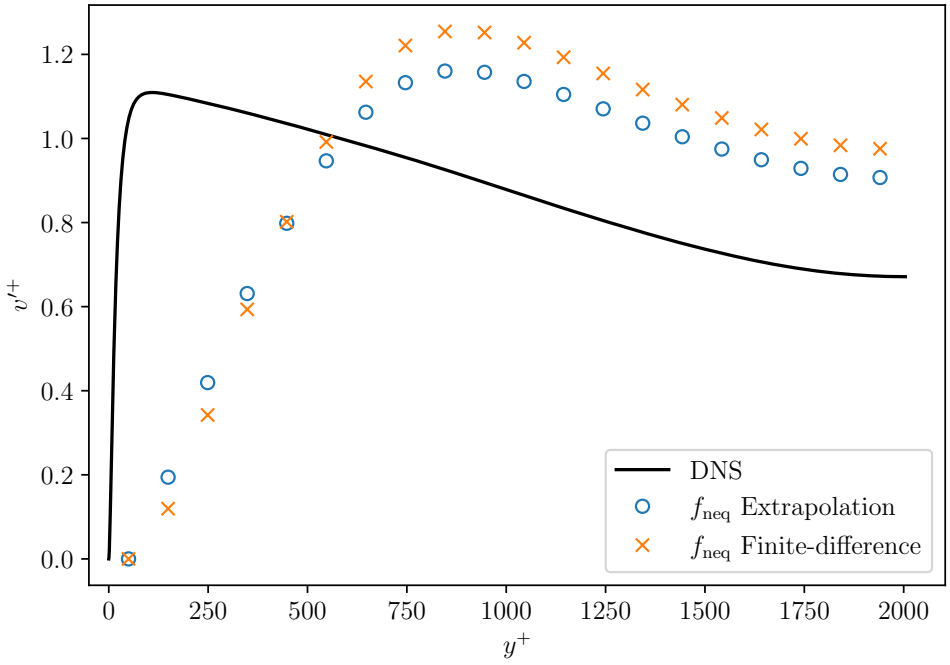


Figure 32: Spanwise velocity fluctuations (v'^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 2000$ for coarse, medium, and fine grid resolutions, compared with DNS data.

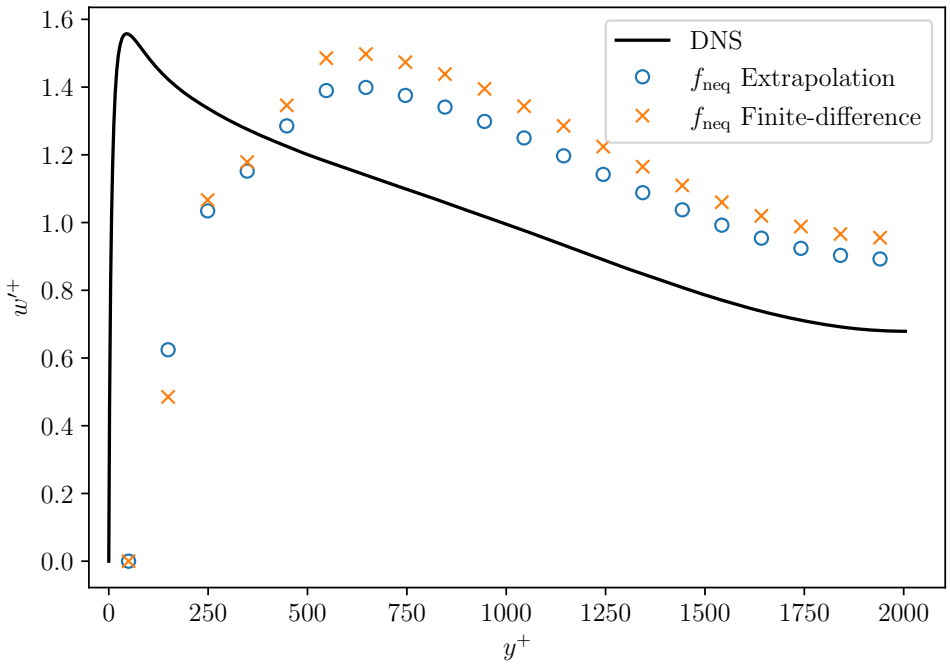


Figure 33: Wall-normal velocity fluctuations (w'^+ vs. y^+) from wall-modelled LES of turbulent channel flow at $Re_\tau = 2000$ for coarse, medium, and fine grid resolutions, compared with DNS data.

7 Conclusion

In this study, a standalone turbulent channel flow LBM-LES solver was extended with a RANS-LES hybrid wall-modelling algorithm, in which various subroutines, such as interpolation algorithms for near-wall quantities, blending functions for the RANS-LES turbulence model, wall functions, and van-Driest damping function, were implemented to complete the wall-modelling algorithm.

The implemented wall model can be readily incorporated into a general 3-D CFD package based on the LBM using the implementation details presented in this study. All the necessary modelling steps were studied and detailed to provide a simple guide to the immersed boundary conditions for WM-LES with LBM. The results for the various configurations clearly show that the implemented wall model performs reasonably well in the near-wall region across different $y+$ values and for different friction Reynolds number flows. However, further studies are required for accurate modelling to improve the results, as listed below.

1. Grid refinement according to the minimum requirement of the LES, and still placing the surrogate boundary at different modelling heights other than the standard $0.5\,\Delta x$

2. Other advanced turbulence modells can be incorporated to check its effects on the improvement of the results

3. Empirical constants such as the blending height of the RANS-LES region, and the wall function constants can be further fine tuned

4. The computational efficiency and memory handling of the various routines implemented can be analysed and improved.

Overall, wall-modelled LES with the immersed boundary method in the LBM context is an advanced and efficient tool that can be used in many different engineering applications and in research. It also has good compatibility with moving boundaries. The near-wall modelling, as seen in this study, includes many ad hoc functions and implementation details. Minor changes in these processes significantly affect the simulation

results near the wall. Hence, careful physical reasoning and precise implementation with understanding are required for wall modelling to be accurate in the general application to highly turbulent flows.

References

[1] H Choi, P Moin Physics of fluids, and undefined 2012. Grid-point requirements for large eddy simulation: Chapman's estimates revisited. *pubs.aip.org*.

[2] Alexandre Suss. Couplage des methodes navier-stokes et lattice boltzmann pour les simulations aerodynamiques instationnaires.

[3] Dean R Chapman. Computational aerodynamics development and outlook. *AIAA journal*, 17(12):1293–1313, 1979.

[4] Shang Gui Cai, Jérôme Jacob, and Pierre Sagaut. Immersed boundary based near-wall modeling for large eddy simulation of turbulent wall-bounded flow. *Computers and Fluids*, 259, 6 2023.

[5] Charles S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10:252–271, 10 1972.

[6] J Mohd-Yusof Annual Research Briefs-1998 and undefined 1998. Methods for complex geometries. *academia.edu*.

[7] O Malaspinas, P Sagaut Journal of Computational Physics, and undefined 2014. Wall model for large-eddy simulation based on the lattice boltzmann method. *ElsevierO Malaspinas, P SagautJournal of Computational Physics, 2014•Elsevier*.

[8] S Wilhelm, J Jacob, P Sagaut Physics of Fluids, and undefined 2018. An explicit power-law-based wall model for lattice boltzmann method–reynolds-averaged numerical simulations of the flow around airfoils. *pubs.aip.org*.

[9] Marc Haussmann, Alejandro CLARO BARRETO, Gislain LIPEME KOUYI, Nicolas Rivière, Hermann Nirschl, and Mathias J. Krause. Large-eddy simulation coupled with wall models for turbulent channel flows at high reynolds numbers with a lattice boltzmann method — application to coriolis mass flowmeter. *Computers and Mathematics with Applications*, 78:3285–3302, 11 2019.

[10] Dazhi Yu, Renwei Mei, and Wei Shyy. A unified boundary treatment in lattice boltzmann method. 2003.

[11] Shinsuke Nishimura, Kentaro Hayashi, Shoeji Nakaye, Minoru Yoshimoto, Kazuhiko Suga, and Takaji Inamuro. Implicit large-eddy simulation of rotating and non-rotating machinery with cumulant lattice boltzmann method aiming for industrial applications. *AIAA Aviation 2019 Forum*, pages 1–16, 2019.

[12] J. Husson, M. Terracol, S. Deck, and T. Le Garrec. A comprehensive framework for robust hybrid rans/les simulations of wall-bounded flows in lbm. *Journal of Computational Physics*, 502, 4 2024.

[13] Andreas Schneider. A consistent large eddy approach for lattice boltzmann methods and its application to complex flows. Technical report, Universität Kaiserslautern, 2015.

[14] S Succi. The lattice boltzmann equation: for fluid dynamics and beyond. 2001.

[15] Dieter A Wolf-Gladrow, Springer Berlin, Heidelberg Newyork, Hong Kong, London Milan, and Paris Tokyo. Lattice-gas cellular automata and lattice boltzmann models: an introduction. 2004.

[16] Graeme Austin Bird. Molecular gas dynamics. *NASA STI/Recon Technical Report A*, 76:40225, 1976.

[17] Friedemann Klaß. *Boundary conditions for multi-speed Lattice Boltzmann methods*. PhD thesis, Dissertation, Wuppertal, Bergische Universität Wuppertal, 2024, 2024.

[18] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94:511, 5 1954.

[19] Xiaowen Shan and Xiaoyi He. Discretization of the velocity space in the solution of the boltzmann equation. *APSX Shan, X HePhysical Review Letters, 1998•APS*, 80:65–68, 1998.

[20] Marzia Bisi, Maria Groppi, Giorgio Martalò, H Baaaolu, S Melchionna, and S Succi. Lattice bgk models for navier-stokes equation. *iopscience.iop.orgYH Qian, D d'Humières, P LallemandEurophysics letters, 1992•iopscience.iop.org*.

[21] Alexander Schukmann, Andreas Schneider, Viktor Haas, and Martin Böhle. Analysis of hierarchical grid refinement techniques for the lattice boltzmann method by numerical experiments. *Fluids*, 8, 3 2023.

[22] S Chapman and TG Cowling. The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases. 1990.

[23] Orestis Malaspinas. Increasing stability and accuracy of the lattice boltzmann scheme: recursivity and regularization. 5 2015.

[24] Jérôme Jacob, Orestis Malaspinas, and Pierre Sagaut. A new hybrid recursive regularised bhatnagar–gross–krook collision model for lattice boltzmann method-based large eddy simulation. *Journal of Turbulence*, 19:1051–1076, 2019.

[25] Z Guo and C Shu. Lattice boltzmann method and its application in engineering. 2013.

[26] Jonas Latt et al. *Hydrodynamic limit of lattice Boltzmann equations*. PhD thesis, Verlag nicht ermittelbar, 2007.

[27] Timm Krüger, Halim Kusumaatmaja, Alexandr Kuzmin, Orest Shardt, Goncalo Silva, and Erlend Magnus Viggen. The lattice boltzmann method. 2017.

[28] A Caiazzo Journal of statistical physics and undefined 2005. Analysis of lattice boltzmann initialization routines. *Springer*, 121:37–48, 10 2005.

[29] R Mei, LS Luo, P Lallemand, D d'Humières Computers, Fluids, and undefined 2006. Consistent initial conditions for lattice boltzmann simulations. *ElsevierR Mei, LS Luo, P Lallemand, D d'HumièresComputers and Fluids, 2006•Elsevier*.

[30] Z Guo and C Shu. Lattice boltzmann method and its application in engineering. 2013.

[31] Alfonso Caiazzo. Asymptotic analysis of lattice boltzmann method for fluid-structure interaction problems. 2007.

[32] Large eddy simulation for incompressible flows. *Large Eddy Simulation for Incompressible Flows*, 2006.

[33] Sergio Hoyas and Javier Jiménez. Reynolds number effects on the reynolds-stress budgets in turbulent channels. In *Physics of Fluids*, volume 20. American Institute of Physics Inc., 2008. This is the DNS Data paper, copy the link in abstract and paste in google to get the site where the DNS data is available!

[34] JH Ferziger, M Perić, JH Ferziger, M Perić Methods for Fluid Dynamics, and undefined 1996. Solution of the navier-stokes equations. *SpringerJH Ferziger, M Perić, JH Ferziger, M PerićComputational Methods for Fluid Dynamics, 1996•Springer*, pages 149–208, 1996.

[35] Shang Gui Cai and Pierre Sagaut. Explicit wall models for large eddy simulation. *Physics of Fluids*, 33, 4 2021.

[36] J Fröhlich. Large eddy simulation turbulenter strömungen. 2006.

[37] J Fröhlich, D Von Terzi Progress in Aerospace Sciences, and undefined 2008. Hybrid les/rans methods for the simulation of turbulent flows. *Elsevier*.

[38] Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice boltzmann method. Technical report.

[39] S Hou, J Sterling, S Chen Pattern formation, ..., and undefined 1996. A lattice boltzmann subgrid model for high reynolds. *books.google.comS Hou, J Sterling, S Chen, GD DoolenPattern formation and lattice gas automata, 1996•books.google.com*, 1994.

[40] O Malaspinas, P Sagaut Journal of Fluid Mechanics, and undefined 2012. Consistent subgrid scale modelling for lattice boltzmann methods. *cambridge.orgO Malaspinas, P SagautJournal of Fluid Mechanics, 2012•cambridge.org*, 700:514, 2012.

[41] Stephen B Pope. Turbulent flows. *Measurement Science and Technology*, 12:2020, 11 2001.

[42] Ugo Piomelli. Wall-layer models for large-eddy simulations, 8 2008.

[43] H. Werner and H. Wengle. Large-eddy simulation of turbulent flow over and around a cube in a plate channel. *Turbulent Shear Flows 8*, pages 155–168, 1993.

[44] Johan Degrigny, Shang Gui Cai, Jean François Boussuge, and Pierre Sagaut. Improved wall model treatment for aerodynamic flows in lbm. *Computers and Fluids*, 227, 9 2021.

[45] DC Wilcox. Turbulence modeling for cfd. 1998.

[46] Motoshi Harada, Yoshiharu Tamaki, Yuichi Takahashi, and Taro Imamura. Simple and robust cut-cell method for high-reynolds-number-flow simulation on cartesian grids. *arc.aiaa.orgM Harada, Y Tamaki, Y Takahashi, T ImamuraAIAA journal, 2017•arc.aiaa.org*, 55:2833–2841, 2017.

[47] Shang Gui Cai, Johan Degrigny, Jean François Boussuge, and Pierre Sagaut. Coupling of turbulence wall models and immersed boundaries on cartesian grids. *Journal of Computational Physics*, 429, 3 2021.

[48] J Jiménez, C Vasco Proceedings of the Summer Program, and undefined 1998. Approximate lateral boundary conditions for turbulent simulations. *CiteseerJ Jiménez, C VascoProceedings of the Summer Program, 1998•Citeseer*.

[49] W. Cabot and P. Moin. Approximate wall boundary conditions in the large-eddy simulation of high reynolds number flow. *Flow, Turbulence and Combustion*, 63:269–291, 2000.

[50] S Kawai and J Larsson. A dynamic wall model for large-eddy simulation of high reynolds number compressible flows. *CTR Annual Research Briefs*, pages 25–37, 2010.

[51] Jonas Latt, Bastien Chopard, Orestis Malaspinas, Michel Deville, and Andreas Michler. Straight velocity boundaries in the lattice boltzmann method. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 77, 5 2008.

[52] G Zhao-Li, Z Chu-Guang, S Bao-Chang Chinese physics, and undefined 2002. Non-equilibrium extrapolation method for velocity and pressure boundary conditions in

the lattice boltzmann method. *iopscience.iop.orgG Zhao-Li, Z Chu-Guang, S Bao-ChangChinese physics, 2002•iopscience.iop.org*.

[53] Malav Soni, Roland Ewert, Jan Delfs, and Kannan Masilamani. Towards wall-modeled les with lattice boltzmann method for aeroacoustics: Application and understanding. In *28th AIAA/CEAS Aeroacoustics Conference, 2022*. American Institute of Aeronautics and Astronautics Inc, AIAA, 2022.

[54] J Jiménez, P Moin Journal of Fluid Mechanics, and undefined 1991. The minimal flow unit in near-wall turbulence. *cambridge.orgJ Jiménez, P MoinJournal of Fluid Mechanics, 1991•cambridge.org*.

[55] Pierre Sagaut, Marc Terracol, and Sebastien Deck. *Multiscale and multiresolution approaches in turbulence-LES, DES and Hybrid RANS/LES Methods: Applications and Guidelines*. World Scientific, 2013.

Eidesstattliche Erklärung

Ich, **Gagan Deep, Prakash**, erkläre hiermit an Eides statt, die vorliegende Arbeit selbstständig angefertigt und alle benutzten Quellen und Hilfsmittel vollständig und genau angegeben zu haben. Weder die vorliegende Arbeit, noch Teile von ihr, wurden für eine staatliche oder andere wissenschaftliche Prüfung eingereicht. Auch wurde weder die vorliegende Arbeit, noch eine andere Abhandlung von mir, bei einem anderen Fachbereich oder einer anderen Universität eingereicht.

Kaiserslautern, den 23. Juli 2025

Gagan Deep, Prakash