Modelling Techniques for Large-Eddy Simulation of Wall-Bounded Turbulent Flows

TIMOFEY MUKHA
Abstract

Large-eddy simulation (LES) is a highly accurate turbulence modelling approach in which a wide range of spatial and temporal scales of the flow are resolved. However, LES becomes prohibitively computationally expensive when applied to wall-bounded flows at high Reynolds numbers, which are typical of many industrial applications. This is caused by the need to resolve very small, yet dynamically important flow structures found in the inner region of turbulent boundary layers (TBLs). To remove the restrictive resolution requirements, coupling LES with special models for the flow in the inner region has been proposed. The predictive accuracy of this promising approach, referred to as wall-modelled LES (WMLES), requires further analysis and validation.

In this work, systematic simulation campaigns of canonical wall-bounded flows have been conducted to support the development of a complete methodology for highly accurate WMLES on unstructured grids. Two novel algebraic wall-stress models are also proposed and shown to be more robust and precise than the classical approaches of the same type.

For turbulence simulations, it is often challenging to provide accurate conditions at the inflow boundaries of the domain. Here, a novel methodology is proposed for generating an inflow TBL using a precursor simulation of turbulent channel flow. A procedure for determining the parameters of the precursor based on the Reynolds number of the inflow TBL is given. The proposed method is robust and easy to implement, and its accuracy is demonstrated to be on par with other state-of-the-art approaches.

To make the above investigations possible, several software packages have been developed in the course of the work on this thesis. This includes a Python package for post-processing the flow simulation results, a Python package for inflow generation methods, and a library for WMLES based on the general-purpose software for computational fluid dynamics OpenFOAM. All three codes are publicly released under an open-source licence to facilitate their use by other research groups.

Keywords: Wall modelling, Inflow generation, Large-eddy simulation, OpenFOAM, Computational fluid dynamics

Timofey Mukha, Department of Information Technology, Box 337, Uppsala University, SE-75105 Uppsala, Sweden.

© Timofey Mukha 2018

ISSN 1651-6214
urn:nbn:se:uu:diva-356729 (http://urn.kb.se/resolve?urn=nbn:se:uu:diva-356729)
Dedicated to the most loving and caring scientists I know: my parents, Elena and Dmitry, and my grandfathers, Genrikh and Vladimir.
List of Papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

I  T. Mukha and M. Liefvendahl,
*The generation of turbulent inflow boundary conditions using precursor channel flow simulations.*
DOI: 10.1016/j.compfluid.2017.06.020

II T. Mukha and M. Liefvendahl,
*Eddylicious: A Python package for turbulent inflow generation.*
DOI: 10.1016/j.softx.2018.04.001

III T. Mukha, M. Johansson, and M. Liefvendahl,
*Effect of wall-stress model and mesh-cell topology on the predictive accuracy of LES of turbulent boundary layer flows.*
In proceedings of the 7th European Conference on Computational Fluid Dynamics, Glasgow, UK, 2018.

IV S. Rezaeiravesh, T. Mukha, and M. Liefvendahl,
*A-priori study of wall modeling in large eddy simulation.*
In proceedings of the 7th European Conference on Computational Fluid Dynamics, Glasgow, UK, 2018.

V T. Mukha,
*Turbulucid: A Python package for post-processing of fluid flow simulations.*
arXiv: 1807.09688

VI T. Mukha, S. Rezaeiravesh, and M. Liefvendahl,
*A library for wall-modelled large-eddy simulation based on OpenFOAM technology.*
Computer Physics Communications (submitted).
arXiv: 1807.11786

Reprints were made with permission from the publishers.
Related Works

The following papers, although not included, are related to the content of this thesis. They are numbered according to their position in the list of references.

[61] T. Mukha and M. Liefvendahl,
*Large-eddy simulation of turbulent channel flow.*
Technical report 2015-014, Uppsala University, Department of Information Technology, 2015.
DiVA: diva2:812934

[62] T. Mukha, S. Rezaeiravesh, and M. Liefvendahl,
*An OpenFOAM library for wall-modelled large-eddy simulation.*
In proceedings of the 12th OpenFOAM Workshop, Exeter, UK, 2017.

[51] M. Liefvendahl, T. Mukha, and S. Rezaeiravesh,
*Formulation of a wall model for LES in a collocated finite-volume framework.*
Technical report 2017-001, Uppsala University, Department of Information Technology, 2017.
DiVA: diva2:1069325

[63] T. Mukha, S. Rezaeiravesh, and M. Liefvendahl,
*Wall-modelled large-eddy simulation of the flow over a backward-facing step.*
In proceedings of the 13th OpenFOAM Workshop, Shanghai, China, 2018.

[80] S. Rezaeiravesh, T. Mukha, and M. Liefvendahl,
*Systematic study of accuracy of wall-modeled large eddy simulation using uncertainty quantification techniques.*
(In preparation)
Contents

1 Introduction ........................................................................................................ 11

2 Turbulent Boundary Layers ............................................................................. 16
  2.1 Turbulent Boundary Layer over a Flat Plate .............................................. 16
  2.2 Turbulent Channel Flow .............................................................................. 20

3 Computational Fluid Dynamics ..................................................................... 23
  3.1 Equations Governing Fluid Motion .......................................................... 23
  3.2 Reynolds-Averaged Navier-Stokes ............................................................ 24
  3.3 Large-Eddy Simulation ............................................................................... 25
  3.4 Numerical Methods .................................................................................... 29

4 Inflow Generation ........................................................................................... 32
  4.1 Review of Existing Approaches .................................................................. 32
  4.2 Strong Recycling of Channel Flow for TBL Simulations ......................... 37

5 Wall Modelling ............................................................................................... 42
  5.1 Definition of Wall-Stress Modelling .......................................................... 43
  5.2 Relationship Between $\tau_w$ and $\overline{\tau}_w$ ...................................... 45
  5.3 Methods for Computing $\overline{\tau}_w$ ............................................................ 47
  5.4 Influence of the Distance to the Sampling Point ....................................... 54
  5.5 Influence of Other Simulation Parameters .............................................. 58
  5.6 Simulation on Unstructured Grids .............................................................. 61
  5.7 Integrated Formulation of Algebraic Wall Models ..................................... 63
  5.8 Dynamically Adjusting Wall Model Constants ......................................... 64

6 Developed Software ....................................................................................... 66
  6.1 A Python Package for Inflow Generation ............................................... 66
  6.2 A Python Package for Post-Processing of CFD Simulations .................... 68
  6.3 A Library for WMLES based on OpenFOAM Technology ....................... 70

7 Conclusions ...................................................................................................... 72

8 Summary of Papers ............................................................................................ 74

9 Sammanfattning på svenska ............................................................................ 76

10 Acknowledgements ......................................................................................... 79
1. Introduction

Apart from idiomatic expressions used to indicate the state of turmoil and disarray, one is most likely to encounter the word *turbulence* on board an aircraft prior to a period of unpleasant jolts and heavy vibrations. This may, perhaps, lead to the idea that turbulent flow is a troublesome physical peculiarity sometimes taking place in the atmosphere. In reality, however, the majority of both naturally and industrially occurring flows are turbulent. Examples include flows in rivers, the flow of the wind around buildings, the flows in pipes and combustion engines, and those around cars, ships, aircraft, and endless others. While sometimes the effects of turbulence are harmful, they are far from exclusively so. For instance, turbulent flows are very effective at mixing, which is advantageous for many applications.

The ubiquity of turbulence has prompted its extensive study by the scientific community. This has turned out to be a notoriously challenging task, as reflected in the words attributed to the prominent fluid dynamist Horace Lamb, ‘...when I die ...there are two matters on which I hope for enlightenment. One is quantum electrodynamics and the other is the turbulent motion of fluids. And about the former I am really rather optimistic’.\(^1\) Turbulence is an extremely rich and complex physical phenomenon, and a full fundamental understanding of it is still far from being achieved. However, significant progress has been made, and many properties of turbulent flow have been discovered.

Mathematically, the motion of a fluid, independent of whether it is turbulent, is governed by the Navier-Stokes equations [75]. Unfortunately, it is generally not possible to solve these non-linear partial differential equations *analytically* (i.e. ‘with pen and paper’). However, in recent decades, significant progress has been made in finding their approximate solutions using *numerical methods*. In application to differential equations, this term generally refers to applying a discretisation procedure, which makes it possible to represent the solution using a finite number of values. The approach employed here is the finite volume method (FVM). This is motivated by its widespread use in applications for finding solutions for complex-geometry flows.

In the FVM, a discrete form of the governing equations is obtained by decomposing the computational domain into polyhedral cells [21], which

\(^1\)A very similar quote is also attributed to Werner Heisenberg.
are collectively referred to as the computational grid. The solution is then sought in the centroids of the cells,\(^2\) which implies that using a finer grid leads to more information about the flow and increased precision of the numerical solution.

In practice, the discretised equations are always solved using computers. Due to that, obtaining a numerical solution is often referred to as conducting a (computer) simulation. The time the machine spends on calculating the results is referred to as the \textit{computational cost}. As the number of cells in the grid increases, so does the cost; therefore, a balance between the desired accuracy of the solution and the resources that are spent on procuring it must be found.

Examination reveals that turbulent flow consists of coherent three-dimensional vortical flow structures, so-called \textit{eddies}, that evolve in space and time via interaction with each other. The physics of the interaction is still not entirely understood, but it is established that it leads to the presence of eddies over a continuous range of sizes, or \textit{scales}. The scales of the largest eddies are commonly determined by the geometry of the flow, but the characteristic size of the smallest ones is typically several orders of magnitude less.

An accurate simulation of a turbulent flow problem requires a computational grid that is dense enough to resolve the whole range of the present turbulent scales. Consequently, the computational cost is often so high that even leadership class supercomputers fail to provide a solution within a reasonable time frame. This issue has led to the development of \textit{turbulence modelling}. This concept refers to deriving different sets of equations, based on Navier-Stokes, where the effect of turbulent fluctuations (or part of them) on the directly computed motions is represented by new terms. In other words, an explicit mathematical representation of the effect of some or all turbulent fluctuations on the flow is introduced.

As the computing power increases, more attention is being drawn to turbulence models in which a significant part of the turbulent motion is resolved by the grid. Compared to lower fidelity approaches, these \textit{scale-resolving} methods have the potential to offer a higher level of accuracy and significantly richer information about the flow, allowing the simulation of complex physical phenomena, such as cavitation, flow noise, and others. The most commonly used scale-resolving approach is large-eddy simulation (LES). As evident by its name, the fundamental idea of LES is to separate all the turbulent scales into ‘large’ and ‘small’ \([82]\). The former scales are then resolved, whereas the effect of the latter is modelled.

The focus of this work is on LES of wall-bounded turbulent flows. Thus, before continuing with the overview of the main contributions of the

\(^2\)Formulations of FVM where the solution is instead sought in the vertices of the grid also exist \([21]\).
thesis, a brief discussion of the physics of near-wall turbulence is necessary. At a solid boundary (a wall), the flow loses part of its momentum due to friction. This gives rise to boundary layers, i.e. thin regions across which the velocity of the fluid rapidly transitions from zero at the wall to the velocity of the free stream. From a computational point of view, turbulent boundary layers (TBLs) represent a major challenge, in particular when a scale-resolving turbulence modelling approach, such as LES, is employed. The reason is that, inside the TBL, a thin layer of fluid exists called the inner region, where the time and length scales of motion are orders of magnitude smaller than outside of it. Yet, these scales are dynamically important, which means that they must be resolved by the LES grid. This is in contrast to flow away from walls where, generally, the smaller flow scales are associated with lower kinetic energy. As a result, the majority of the computational effort is dedicated to the near-wall region. Moreover, for a given fluid and geometric configuration, this effect increases with the velocity of the flow, thus prohibiting the simulation of many flow configurations of practical interest.

The complications introduced by the inner region of TBLs outlined above have led to the search for novel modelling techniques that would enhance LES with special treatment of this part of the flow. Several approaches have been proposed [6, 43], and the established term for simulations employing them is wall-modelled LES (WMLES). Simplified modelling of near-wall turbulence is challenging, and none of the approaches found in the literature are perfect. Here, a particular class of methods referred to as wall-stress models is considered. Their shared property is that the dynamics of the inner layer are accounted for by enforcing the correct local wall shear stress (i.e. the friction force between the fluid and the wall per unit contact area). To approximate it, the solution to the LES equations sampled from a single point located at some distance \( h \) away from the wall is typically used. In spite of the first wall-stress model being introduced as early as 1975 [86], there remains substantial room for further development of this approach and understanding of its properties and limitations. Yet another issue is that the established practices for the choices of other influential LES modelling parameters (e.g. the model for the unresolved turbulent fluctuations) must be reconsidered and modified to couple well with the wall modelling approach employed.

In this thesis (Papers III, IV, VI), the properties of WMLES are studied through systematic simulation campaigns of several canonical turbulent flows. Trends regarding the effect of different modelling choices on the predictive accuracy are revealed and further used to propose a WMLES methodology, which leads to excellent results for important flow characteristics, such as the mean velocity field and the mean wall shear stress. Moreover, in Paper III, it is shown that the proposed approach can be used for simulations on computational grids with an unstructured topol-
ogy. This property is important because such grids are typically employed in simulations of complex-geometry flows.

Two novel wall-stress modelling techniques are introduced in Paper IV, both based on classical approaches discussed in Section 5.3.1 below. The suggested enhancements are based on predicting the local wall shear stress using the LES solution from several points in the domain, instead of just one. In the first proposed approach, this is used to reduce the sensitivity of the model accuracy to the choice of \( h \), which in this setting is defined as the distance to the sampling point farthest from the wall. In the other, it allows the introduction of a procedure for dynamically adjusting the parameters of the model, which leads to accurate predictions of the fluctuations that the wall shear stress exhibits.

Another challenge addressed in this thesis is that of inflow generation. In a simulation, the spatial domain of the studied problem is always bounded. In engineering, this may, for instance, correspond to some component of a more complex device or system. The behaviour of velocity, pressure, and other flow values at the boundaries must be modelled in such a way that it accurately represents the real-life behaviour of the flow. When a significant part of the turbulent scales is resolved (as in LES), one of the most challenging types of boundaries to model is the inflow. The prescribed values of the unknowns should change with time and contain structures that are an accurate representation of the upstream turbulent flow in question.

The existing methods for inflow generation for LES can be classified into two categories [99, 106]: precursor simulation-based methods and synthesis methods. The former rely on an auxiliary simulation, referred to as the precursor, to generate the inflow data. The latter instead generate it using a stochastic procedure, based on some known characteristics of the flow (e.g. the statistical moments of the velocity field). Precursor-based methods generally offer the best accuracy [99, 39] but at the cost of the overhead of the extra simulation.

In Paper I, the specific case of an inflow TBL is considered, and a complete description of a novel precursor-based methodology for generating the corresponding velocity values is given. In particular, using a precursor simulation of turbulent channel flow is proposed (see Section 2.2 for the definition of this flow). The velocity values are sampled from a plane spanning half of the channel flow domain in the wall-normal direction and oriented perpendicular to the direction of the mean flow. A procedure for how to specify the parameters of the precursor simulation based on the desired properties of the inflow TBL is suggested. Practical advantages of the proposed approach are its robustness and that it requires a minimum time to implement if the employed solver already includes functionality for surface sampling and reading boundary values from the disk. The
accuracy of the method is shown to be similar to other state-of-the-art approaches.

To make the above investigations possible, several software packages have been developed in the course of the work on this thesis. In Paper II, the Python package **eddylicious** is presented. The purpose of the package is to collect existing inflow generation methods in a single code base and thus contribute to accurate comparative studies of their properties. In Paper V, another Python package, **turbulucid**, is introduced. This software can be used to produce publication-quality visualisations of one- and two-dimensional simulation data. Several data extraction routines are also provided. Examples of plots produced with **turbulucid** can be found in Paper VI, which in turn describes a C++ library implementing a wide range of wall-stress models. The library is designed to be used with the general-purpose LES solvers included in **OpenFOAM**, which is an open-source software for fluid flow simulations with a large user base. Consequently, the library contributes to a wider adoption of wall-stress modelling and further development and validation of this turbulence modelling approach. All of the above programs are well-documented and released to the public under an open-source licence, thus facilitating their use by other research groups.

The remaining part of the thesis is structured as follows. In Chapter 2, the physical properties of TBLs are reviewed. Chapter 3 introduces the Navier-Stokes equations and formally defines the turbulence modelling approaches relevant for this thesis. In particular, a detailed description of LES is given. In the chapter’s last section, the FVM and associated discretisation schemes are discussed. Following that, Chapter 4 reviews precursor-based methods for inflow generation and provides a detailed description of the method proposed in Paper I. In Chapter 5, various aspects of wall modelling are considered, and the corresponding contributions of this thesis are presented. The three developed software packages are then described in Chapter 6. Finally, concluding remarks are given in Chapter 7, and a summary of the contributions of each paper included in the thesis is given in Chapter 8.
2. Turbulent Boundary Layers

In this chapter, a short review of the physical properties of TBLs is given, setting the stage for the discussion of inflow generation methods in Chapter 4 and wall modelling in Chapter 5. The introduced concepts are exemplified by two particular wall-bounded flows that play a key role in the present work: the zero-pressure-gradient (ZPG)-TBL over a flat plate and the fully developed turbulent channel flow. For a more comprehensive discussion of wall-bounded turbulent flows, see [75, pp. 264–333].

Some notation that will be used in this thesis is now introduced. A Cartesian coordinate system \((x, y, z)\) will be employed. The three coordinates correspond to the streamwise, wall-normal, and spanwise directions, respectively. The associated components of the velocity field are \((u, v, w)\). Angular brackets \(\langle \cdot \rangle\) will be used to denote the ensemble- and/or time-averaged value of a quantity. Bold font will be used to denote vectors.

2.1 Turbulent Boundary Layer over a Flat Plate

2.1.1 Definition of the Flow

Consider a uniform-velocity non-turbulent stream that encounters a smooth flat plate of infinite length and width aligned with the direction of the flow. This setup gives rise to one of the simplest types of boundary layers: the ZPG-TBL over a flat plate.

At the plate, the fluid loses momentum due to friction. Yet at some distance \(\delta\), referred to as the boundary layer thickness, the velocity is undisturbed by the presence of the plate and is equal to the velocity of the free stream, \(U_0\). Gradually, momentum is increasingly lost at the wall, leading to the growth of \(\delta\) with \(x\). The flow is thus spatially developing and statistically inhomogeneous in both the streamwise and wall-normal directions. In the spanwise direction, however, the statistics remain unchanged. The ZPG-TBL is, therefore, statistically two-dimensional. A snapshot of the velocity magnitude taken from an LES of a ZPG-TBL is shown in Figure 2.1. The resolved turbulent structures can be clearly seen.

The boundary layer thickness \(\delta\) is a theoretical quantity, which is impossible to measure. Therefore, three alternative methods to define a characteristic length scale for the boundary layer exist.
\( \delta_{99} \) — the value of \( y \), where \( \langle u \rangle = 0.99U_0 \).

\( \theta = \int_0^\infty \frac{\langle w \rangle}{U_0} \left( 1 - \frac{\langle u \rangle}{U_0} \right) dy \) — the momentum thickness, which is a measure of the loss of the momentum flux in the boundary layer.

\( \delta^* = \int_0^\infty \left( 1 - \frac{\langle u \rangle}{U_0} \right) dy \) — the displacement thickness, which is a measure of the loss of the mass flux in the boundary layer.

It is possible to form a non-dimensional quantity, the Reynolds (Re) number, based on the characteristic length and velocity scales of the flow and the kinematic viscosity \( \nu \). Using the length scales defined above, the following Reynolds numbers are formed:

\[
\text{Re}_x = \frac{U_0 x}{\nu}, \quad \text{Re}_{\delta_{99}} = \frac{U_0 \delta_{99}}{\nu}, \quad \text{Re}_{\theta} = \frac{U_0 \theta}{\nu}, \quad \text{Re}_{\delta^*} = \frac{U_0 \delta^*}{\nu}.
\] (2.1)

All of them can be used to fully define the state of the boundary layer.

At the leading edge of the flat plate \((x = 0)\), the boundary layer is laminar and remains as such until a certain critical Reynolds number \((\text{Re}_x \approx 10^6, [75, \text{p. 300}])\) is reached from which transition to turbulence begins. After some distance downstream of that location, the flow is fully transitioned to the turbulent regime. The study of the transition process is outside of the scope of this thesis; therefore, the TBL is further assumed to be fully turbulent.

### 2.1.2 Structure of the Mean Velocity Profile

The left plot in Figure 2.2 shows the mean velocity profile of a ZPG-TBL obtained from a direct numerical simulation (DNS) [83], that is, a numerical solution to the Navier-Stokes equations, resolving all the
turbulent scales. A thin region near the wall where the velocity gradient is large is clearly visible. The origin of this region can be explained in the following way. Away from the wall, the extra mixing introduced by turbulence smooths out the velocity gradient, making the profile flatter. At the same time, at the wall, both the velocity and the turbulent stresses are forced to be zero. This results in the transition to the flatter profile further away from the wall being rapid and steep.

![Figure 2.2](image)

*Figure 2.2.* The mean velocity profile of a TBL at $\text{Re}_\theta = 4060$, taken from DNS data by Schlatter and Örlü [83], shown in outer scaling (left) and inner scaling (right). The region shaded pink covers the inner layer ($y \leq 0.1\delta_{99}$), the region shaded yellow covers the outer layer ($y^+ \geq 50$). The overlap region is shaded orange. The dashed line shows the log-law, see equation (2.4).

In the wall-normal direction, the flow can thus be considered divided into two overlapping regions: the *inner layer* (shaded pink in Figure 2.2), where the flow is independent of $\delta$ and $U_0$, and the *outer layer* (shaded yellow), where the effect of $\nu$ is negligible and turbulent stresses are dominant. The part of the profile that resides in both the inner and the outer layers is called the *overlap* region (shaded orange).

Recall that the wall shear stress, $\tau_w$, is the friction force per unit contact area between the fluid and the wall. The magnitude of this quantity, denoted by $\tau_w$, can be used to define so-called *viscous* scales that are used to describe the flow in the inner layer. A velocity scale can be obtained as $u_\tau = \sqrt{\tau_w/\rho}$ and is referred to as the friction velocity. Further, the viscous length scale, or *wall unit*, can be defined as $\delta_\nu = \nu/u_\tau$. The wall-normal coordinate expressed in wall units is thus $y^+ = y/\delta_\nu$, which is also referred to as the inner coordinate. Correspondingly, $u_\tau$ can be used to normalise the velocity, $u^+ = u/u_\tau$. The mean velocity profile of a ZPG-TBL in inner scaling is shown in the right plot of Figure 2.2. To enlarge the near-wall region, logarithmic scaling is applied to the abscissa.
Having introduced the above quantities, it is possible to define the extents of the inner and outer layers. According to Pope [75, p. 276], the upper bound of the inner layer for high Re-number flows resides at $\approx 0.1\delta$, whereas the beginning of the outer layer is located at $y^+ \approx 50$ [75].

Note that the ratio of the outer and inner length scales define a Reynolds number, $Re_\tau = \delta_{99} u_\tau / \nu$, which can also be used to specify the state of the TBL along with the Re-numbers introduced earlier in (2.1). This implies that, as the Re-number grows, the characteristic size of the turbulent structures in the inner layer ($\sim \delta_\nu$) becomes an increasingly small fraction of $\delta_{99}$.

2.1.3 Laws of the Wall

By definition, the flow in the inner layer can be expressed as a function of $y^+$, whereas a function of $y/\delta$ (the outer coordinate) should be used in the outer layer:

$$\langle u \rangle / u_\tau = \Phi_1(y^+), \quad (2.2)$$
$$\left(U_0 - \langle u \rangle \right) / u_\tau = \Phi_2(y/\delta). \quad (2.3)$$

Here, $\Phi_1$ and $\Phi_2$ are some functions. Based on the fact that, in the overlap region, both relationships should describe the same profile, asymptotic analysis [75] can be used to show that, in the region between $y^+ \approx 30$ and $y/\delta \approx 0.3$, the mean streamwise velocity follows the so-called log-law:

$$\langle u \rangle^+ = \frac{1}{\kappa} \ln y^+ + B. \quad (2.4)$$

Here, $\kappa$ and $B$ are constants. Their exact values are still a matter of scientific investigation, but they are generally considered to be within 5% of $\kappa = 0.41$ and $B = 5.2$ [75, p. 274]. The log-law is shown with a dashed line in the right plot in Figure 2.2. In the wall-normal interval where the law is expected to hold, its agreement with the mean velocity profile is excellent.

Right next to the wall, $y^+ < 5$, the inner-scaled velocity, $\langle u \rangle^+$, can be shown to be equal to $y^+$. In this region, known as the viscous sublayer, the viscous stresses dominate over the turbulent ones. Connecting the viscous sublayer and the log-law is the buffer region, for which the exact relationship between $\langle u \rangle^+$ and $y^+$ is not known.

Based on the above, several analytical expressions for the mean velocity profile in the inner layer have been proposed, referred to as ‘laws of the wall’ in the literature. Two such expressions are used in this thesis, which
are Spalding’s law [96]:

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

(2.5)

where the model constants are the same as in the log-law, and Reichardt’s law [77]:

\[ \langle u \rangle^+ = \frac{1}{\kappa} \ln (1 + \kappa y^+) + C \left( 1 - e^{(-y^+ / B_1)} - \frac{y^+}{B_1} e^{(-y^+ / B_2)} \right), \]  

(2.6)

with the model constant values \( \kappa = 0.41, C = 7.8, B_1 = 11, \) and \( B_2 = 3. \)

The above expressions may appear complicated, but as revealed in Figure 2.3, both simply provide a fit to the mean velocity profile of a ZPG-TBL in the inner layer.

![Figure 2.3](image)

*Figure 2.3. The DNS mean velocity profile of a ZPG-TBL in the inner layer [83] compared to the profiles given by Spalding’s and Reichardt’s laws.*

### 2.2 Turbulent Channel Flow

Channel flow is defined as a flow between two infinite parallel plates driven by a constant mean pressure gradient. A snapshot of the velocity magnitude taken from an LES of channel flow is shown in Figure 2.4. The geometry of the channel is fully defined by the channel height \( h = 2\delta, \) i.e. the distance between the plates.

Due to the infinite length and width of the plates, the velocity statistics are independent of \( x \) and \( z \) and vary only in the wall-normal direction. In other words, the flow is statistically one-dimensional. From a computational point of view, this means that periodic boundary conditions can
be employed in the stream- and spanwise directions. Combined with the simple geometry of the flow, this makes the simulation easy to set up. For this reason, channel flow nearly always serves as one of the initial test cases for novel scale-resolving turbulence modelling approaches (see e.g. [8, 67, 65, 44, 69, 5, 109]).

To characterise the flow, the following velocity scales can be introduced. The bulk velocity is defined as

\[ U_b = \frac{1}{h} \int_0^h \langle u \rangle \, dy. \]  

(2.7)

The centreline velocity, \( U_c \), is defined as \( \langle u \rangle(\delta) \). As a wall-bounded flow, channel flow shares the multi-layered structure of the ZPG-TBL; hence, in the inner layer, the friction velocity \( u_\tau \) is the relevant characteristic velocity.

The velocity scales together with the kinematic viscosity of the flow, \( \nu \), and the length scale defined by the half-height of the channel, \( \delta \), form the following Re-numbers:

\[ \text{Re}_b = \frac{\delta U_b}{\nu}, \quad \text{Re}_c = \frac{U_c \delta}{\nu}, \quad \text{Re}_\tau = \frac{u_\tau \delta}{\nu}. \]  

(2.8)

Each fully defines the flow. Fully developed turbulence is obtained for \( \text{Re}_b > 1800 \), [75].

Figure 2.5 shows the inner-scaled mean streamwise velocity profiles for channel flow and a ZPG-TBL at \( \text{Re}_\tau \approx 1000 \). It is evident that the agreement between the two flows in the inner layer is excellent, the two curves are nearly indistinguishable. This collapse of the profiles in the
inner layer was observed in an abundant number of experimental and computational results, and not only for channel flow and the ZPG-TBL but also for pipe flow [75]. Mathematically, this means that the function $\Phi_1$ in (2.2) is universal.

![Figure 2.5](image)

**Figure 2.5.** Mean velocity profiles taken from the DNS of a ZPG-TBL [83] and channel flow [45] both at $Re_r \approx 1000$.

By contrast, in the outer layer, beyond the overlap region, the agreement between the two flows is poor. This is not surprising since, at the edge of the boundary layer, the turbulence interacts with the free stream, whereas the dynamics are different in the channel flow.
3. Computational Fluid Dynamics

This chapter introduces the equations governing fluid motion, as well as two turbulence modelling approaches: Reynolds-averaged Navier-Stokes (RANS) and large-eddy simulation (LES). A significantly more detailed consideration of the latter is given since LES is the focus of this work. In the last section, finite volume discretisation and the associated numerical methods are discussed. The implications of using these methods for solving the equations governing LES are analysed.

3.1 Equations Governing Fluid Motion

Recall that the motion of a fluid is governed by the Navier-Stokes equations. These equations are presented below for the particular case of a fluid that can be considered incompressible. Specifically, the density of the fluid, $\rho$, is considered constant. Incompressibility is a common assumption when modelling the flow of liquids but can also be employed for gases when the velocity of the flow is significantly lower than the speed of sound.

Under the above assumptions, the Navier-Stokes equations form a system of two conservation laws: for the momentum of the fluid and for its mass. For the former, the governing equation is as follows:

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \tilde p}{\partial x_i} + \frac{\partial}{\partial x_j} \left( 2\nu S_{ij} \right),$$

where

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

Here, $x_i, i = 1,2,3$ are the Cartesian coordinates, $u_i$ are the corresponding components of the velocity field, $\tilde p$ is the pressure, $\nu$ is the kinematic viscosity, and $S_{ij}$ is the rate-of-strain tensor. Summation over repeated indices is implied. For brevity, the modified pressure $p = \tilde p/\rho$ will be referred to as the ‘pressure’ below.

The law of mass conservation in the case of incompressible flow implies that the velocity field must be solenoidal:

$$\frac{\partial u_i}{\partial x_i} = 0.$$
Equations (3.1) and (3.3) must be complemented with initial and boundary conditions. In simulations of turbulent flow, the role of the initial conditions is commonly to introduce perturbations that will facilitate the development of turbulence. In addition, for statistically steady flows, accurate initial conditions can lead to faster convergence. In this work, uniform values of pressure and velocity are used as initial conditions. The perturbations introduced by the numerical schemes used for discretisation (see Section 3.4) were found to be sufficient to trigger the development of turbulence. Moreover, most of the studies here involve a simulation campaign (i.e. the same flow is typically computed more than once). For all the simulations but the first, it is possible to use the solution from a previous simulation as the initial conditions.

The boundary conditions that are appropriate depend on what the boundaries physically represent. For example, in the case of a stationary solid wall, the boundary condition for the velocity components is \( u = 0 \), whereas a zero-gradient condition should be applied for the pressure. At outflow boundaries, the opposite should be used, i.e. \( p = 0 \) and a zero-gradient condition on velocity. As explained earlier, inflow boundaries represent a challenge, and their detailed discussion is postponed to Chapter 4. Finally, periodic boundary conditions can be imposed to simulate a domain that is infinitely large along a certain direction, as discussed previously in the context of channel flow.

Recall that numerically solving (3.1) and (3.3) is referred to as DNS, and it offers the highest degree of accuracy as well as the most complete description of the flow. The cost of DNS for wall-bounded flows can be quantified by estimating how the number of grid cells, \( N \), needed to properly resolve all the turbulent eddies of a ZPG-TBL, scales with its Reynolds number. Such an estimate is given in [12]: \( N \sim \text{Re}^{2.75} \). This implies that, in the foreseeable future, DNS can only be conducted for flows at Re-numbers orders of magnitude lower than typically encountered in industrial applications.

3.2 Reynolds-Averaged Navier-Stokes

In engineering, it is often sufficient to obtain only the average values of the unknowns, i.e. \( \langle u \rangle \) and \( \langle p \rangle \). The equations for these mean quantities can be derived in the following way. Both the pressure and the velocity signals are first decomposed into a mean and fluctuating part: \( u = \langle u \rangle + u' \), \( p = \langle p \rangle + p' \). Inserting this decomposition into equations (3.1) and (3.3) results (after some algebraic manipulations) in the following:
\[
\frac{\partial \langle u_i \rangle}{\partial x_i} = 0, \quad (3.4)
\]
\[
\frac{\partial \langle u_i \rangle}{\partial t} + \frac{\partial \langle u_i \rangle \langle u_j \rangle}{\partial x_j} = -\frac{\partial \langle p \rangle}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \tau_{ij}^{\text{trans}} + 2\nu \langle S_{ij} \rangle \right), \quad (3.5)
\]

where
\[
\tau_{ij}^{\text{trans}} = \langle u_i \rangle \langle u_j \rangle - \langle u_i u_j \rangle \quad (3.6)
\]
is the Reynolds stress tensor,\(^1\) which accounts for the effect of all the turbulent motion on the mean flow. Since \(\tau_{ij}^{\text{trans}}\) is not expressed directly in terms of the mean flow quantities, an alternative method of computing it must be found in order to close the system of equations. Finding such closures constitutes RANS turbulence modelling. A common approach is to employ the Boussinesq approximation, which is the hypothesis that the anisotropic part of the Reynolds stress tensor can be modelled as structurally similar to the viscous stress \([75]\). Expressed mathematically, it means that turbulence can be accounted for by adding a space- and time-dependent turbulent viscosity, \(\nu_t\), to the kinematic viscosity \(\nu\) in the viscous term of the momentum equation. Naturally, this is only part of the solution to the closure problem since now a method to compute \(\nu_t\) must be introduced. The reader interested in a detailed overview of approaches is referred to \([75]\). Here, only the simplest \(\nu_t\) models will be discussed in Chapter 5 in the context of wall modelling.

Since no turbulent fluctuations are resolved in RANS, the grid does not have to be adapted to the size of the turbulent structures present in the flow. However, the form of the mean velocity profile of the TBL (see Section 2.1.2) must be properly resolved, which means that, in the inner layer, the wall-normal cell size is still defined in terms of the inner scale \(\delta_v\). Nevertheless, compared to DNS and LES (see below), RANS is extremely cheap and can be used to compute flows at virtually any Re-number encountered in practice.

3.3 Large-Eddy Simulation

3.3.1 Scale Separation via Filtering

In the introduction, scale-resolving turbulence modelling approaches were introduced as a more accurate alternative to lower fidelity approaches, such as RANS. Unlike the latter, they provide the means of obtaining the\(^1\)Strictly, \(\nu_t^{\text{trans}}\) is the stress tensor, but here the mass-specific quantity is referred to by the same name, similar to the treatment of pressure. This is common when incompressible flows are considered (see e.g. \([75\), p. 578]).
instantaneous values of the unknowns over the whole time-span covered by the simulation, in principle making it possible to compute statistical moments of arbitrary order and not just the mean. Furthermore, large-eddy simulation was pointed out as the most widely used scale-resolving approach. At the core of the LES approach is a scale separation procedure that distinguishes between turbulent motions that will be directly resolved and those that will be modelled. In particular, the separation is achieved via spatial filtering [82]:

$$\tilde{\phi}(x,t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi(\xi,t) G(x,\xi,\Delta) d^3\xi.$$  \hspace{1cm} (3.7)

Here, $\phi$ is some flow variable, $\tilde{\phi}$ is its resolved part, $G$ is the kernel of the filter, and $\Delta$ is a characteristic length scale also referred to as the filter width. The scales that are filtered out in (3.7) are referred to as subgrid scales (SGS). The definitions of $G$ and $\Delta$ are commonly tightly connected to the discretisation practices employed in the process of solving the LES equations, which are presented below. The discussion of filtering will therefore be continued in Section 3.4, dedicated to numerical methods.

The filtering operation defines a new set of unknowns, namely the filtered velocity $\bar{u}_i$ and the filtered pressure $\bar{p}$. To derive the equations governing these quantities, filtering is applied to (3.1) and (3.3). Assuming that filtering commutes with differentiation, this gives the following result [82]:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \hspace{1cm} (3.8)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left( 2\nu \bar{S}_{ij} \right), \hspace{1cm} (3.9)$$

$$\bar{S}_{ij} = \frac{1}{2} \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right). \hspace{1cm} (3.10)$$

Note that the term $\bar{u}_i \bar{u}_j$ is not readily expressible in terms of the new unknowns. By adding and subtracting $\bar{u}_i \bar{u}_j$ to $\bar{u}_i \bar{u}_j$ the following is obtained:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \bar{\tau}_{ij}^{\text{sgs}} + 2\nu \bar{S}_{ij} \right), \hspace{1cm} (3.11)$$

where

$$\bar{\tau}_{ij}^{\text{sgs}} = \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j \hspace{1cm} (3.12)$$

is the SGS stress tensor. There is a striking similarity between the RANS equations (3.4)-(3.5) and those obtained above, even though they represent completely different turbulence modelling methodologies. Just like
the Reynolds stress tensor in the case of RANS, the SGS stress tensor must be modelled in order to close the system of equations. The principle difference is that $\tau_{ij}^{\text{trans}}$ represents the effect of all the turbulent fluctuations, whereas $\tau_{ij}^{\text{sgs}}$ only stands for those motions that are filtered out and their dynamic coupling to the resolved fluctuations. A brief review of some of the existing approaches for modelling $\tau_{ij}^{\text{sgs}}$ is given in the next section, but already now it can be mentioned that most of them employ the Boussinesq approximation introduced above in the context of RANS.

The filtering of the unknowns used in LES leads to the cost of simulation being significantly less than that of DNS. However, in the case of wall-bounded flows, the LES grid should resolve the dynamically important scales in both the outer and inner layers. This entails that the required number of grid points still grows rapidly with Re. In particular, for a ZPG-TBL over a flat plate, $N \sim Re^{1.85}$ [12, 78, 47].

3.3.2 Modelling the Subgrid Stresses

A rich variety of approaches to modelling $\tau_{ij}^{\text{sgs}}$ has been developed, and a thorough description of many of them can be found in [82]. However, only a limited number of the models proposed in the literature have found their way into the general-purpose software for computational fluid dynamics (CFD). The models that are used in this thesis are described below.

As indicated previously, the most common approach to SGS modelling is to employ the Boussinesq approximation [82], which boils down to adding a space- and time-dependent subgrid viscosity field, $\nu_{\text{sgs}}$, to the kinematic viscosity $\nu$ in the viscous term of equation (3.11). The task is then to find a computable expression for $\nu_{\text{sgs}}$. This is possible under the assumption that a characteristic length scale $l_{\text{sgs}}$ and time scale $t_{\text{sgs}}$ are sufficient to describe the subgrid scales [82]. Then, based on dimensional analysis, it follows that

$$\nu_{\text{sgs}} \sim \frac{\nu^2}{t_{\text{sgs}}} = u_{\text{sgs}} l_{\text{sgs}},$$

(3.13)

where $u_{\text{sgs}}$ is the corresponding velocity scale. A natural choice for $l_{\text{sgs}}$ is the filter width $\Delta$. Defining a time or velocity scale is a much more challenging task. The latter can, however, be represented as the square root of the SGS turbulent kinetic energy, $k_{\text{sgs}} = \tau_{kk}^{\text{sgs}}/2$. To obtain $k_{\text{sgs}}$, solving the following transport equation was proposed independently by several researchers [30, 41, 86, 110, 111, 98]:

$$\frac{\partial k_{\text{sgs}}}{\partial t} + \frac{\partial \overline{u_i k_{\text{sgs}}}}{\partial x_i} = 2\nu_{\text{sgs}} \overline{S_{ij} \overline{S}_{ij}} - C_{e} \frac{k_{\text{sgs}}^{3/2}}{\Delta} + \frac{\partial}{\partial x_i} \left( \nu_{\text{sgs}} \frac{\partial k_{\text{sgs}}}{\partial x_i} \right) + \nu \frac{\partial^2 k_{\text{sgs}}}{\partial x_i \partial x_i},$$

(3.14)
where $C_e = 1.048$ is a model constant. The expression for $\nu_{\text{sgs}}$ is then taken to be:

$$\nu_{\text{sgs}} = C_k \Delta \sqrt{k_{\text{sgs}}}, \quad (3.15)$$

where, $C_k = 0.094$, is another model constant. More details on the derivation of (3.14) and the employed modelling assumptions can be found in [82, p. 128]. An improvement of this model was proposed in [40], based on treating $C_e$ and $C_k$ as a spatially and temporally varying fields instead of assuming them to be constant. A dynamic procedure for determining the local values of $C_e$ and $C_k$ was introduced, based on the work of Germano [26] and Lilly [52]. This dynamic version of the model is referred to as the dynamic one-equation model below.

Another popular approach, known as the Smagorinsky model [92], is to use the magnitude of the filtered rate-of-strain tensor as the inverted time scale, $1/t_{\text{sgs}}$. This leads to:

$$\nu_{\text{sgs}} = (C_s \Delta)^2 \sqrt{2S_{ij}S^*_{ij}}, \quad (3.16)$$

where $C_s \approx 0.18$ is the Smagorinsky constant. A known downside to the Smagorinsky model is that it suffers from incorrect behaviour of $\nu_{\text{sgs}}$ in the limit $y \to 0$, where $y$ is the distance from the wall. Namely, the values of the $\nu_{\text{sgs}}$ are over-predicted. It is noted for completeness that a dynamic version of the Smagorinsky model exists as well. In fact, it is for this model that the dynamic procedure was originally developed [26, 52, 59].

Yet another approach is the wall-adapting local eddy viscosity (WALE) model [64], where the time scale is expressed using both the filtered rate-of-strain tensor $\tilde{S}_{ij}$ and the filtered vorticity tensor $\tilde{\Omega}_{ij} = (\partial \tilde{u}_i / \partial x_j - \partial \tilde{u}_j / \partial x_i)/2$. In this model, $\nu_{\text{sgs}} \sim y^3$ as $y \to 0$, which is the correct behaviour. Let $S^d$ be the traceless symmetric part of the square of the velocity gradient tensor. The values of $\nu_{\text{sgs}}$ are then obtained as follows:

$$\nu_{\text{sgs}} = (C_w \Delta)^2 \frac{(S^d_{ij}S^d_{ij})^{3/2}}{(S_{ij}S_{ij})^{5/2} + (S^d_{ij}S^d_{ij})^{5/4}}. \quad (3.17)$$

A drastically different approach to SGS modelling, referred to as implicit LES (ILES) [27], is to rely on the extra dissipation introduced by the employed numerical methods instead of adding it explicitly via $\nu_{\text{sgs}}$. The study [25] provides a comparison between ILES and the conventional SGS modelling. In particular, flux limiting schemes are shown to give an error term consistent with the structure of conventional SGS models. In the experience of the author (see [60]), when the LES grid resolves all the dynamically important scales, ILES gives results that are at least not worse than those obtained with $\nu_{\text{sgs}}$-based models. For that reason, ILES
is employed in Paper I. The ILES approach is also attractive because removing the need for computing $\nu_{sgs}$ saves computational resources, especially compared to models that require solving a transport equation to compute $\nu_{sgs}$, such as the dynamic one-equation model.

3.4 Numerical Methods

In this section, the numerical methods used to solve (3.8) and (3.11) are discussed. All the simulations that were conducted as part of the work on this thesis were performed using the freely available general-purpose CFD library OpenFOAM. Therefore, the methods presented below describe a part of the capabilities of this software. For a more detailed description of the numerical algorithms behind OpenFOAM and their implementation, the interested reader is referred to [34], [16], and [81].

To discretise the governing equations, OpenFOAM uses the finite volume method (FVM). The computational domain is decomposed into a large number of polyhedral cells, also called control volumes. Collectively, the cells are referred to as the computational grid or mesh. The continuous flow field is represented by the values of the unknowns in the centres of the cells. It can be shown [21] that, for some field $\phi$, the value stored in the centre of a given cell provides a second-order accurate approximation of the field average across the cell volume.

Generally, for a convection-diffusion equation, such as (3.11), a discrete formulation in space is obtained by applying the following procedure for each cell in the mesh:

- Integrating the equation over the volume of the cell.
- Applying the Gauss-Ostrogradsky theorem to the convective and diffusive terms to convert volume integrals into sums of surface integrals over the faces of the cell.
- Approximating the surface integrals using the values of relevant quantities at the face centres of the cell. These values are obtained using an interpolation procedure, see below.

Before proceeding with a more detailed consideration of the last step of the above algorithm, two remarks are made regarding its application to (3.11). One is that the convective term must be linearised. The other is that the pressure gradient term can also be discretised using the Gauss-Ostrogradsky theorem by considering $\partial p/\partial x_i$ as the divergence of a vector field $\partial p/\partial x_i \hat{i}$, where $\hat{i}$ is a unit vector directed along the $i$-th axis [21, p. 233].

To obtain the face-centred values necessary to compute the surface fluxes, interpolation using the stored cell-centred values must be used. The choice of interpolation schemes affects the overall order of accuracy of
the spatial discretisation, as well as numerical stability. To obtain reliable results, maintaining at least second-order accuracy is highly desirable.

A common scheme to use in conjunction with LES is linear interpolation using the values in the centres of the cells sharing a given face (see e.g. [101, 23, 7, 4]). This scheme is second-order accurate but, unfortunately, not bounded when applied to interpolation of convective fluxes. Using it for this purpose, therefore, leads to the introduction of numerical oscillations. This does not cause a problem in the case of traditional LES due to the associated small grid-cell size, but for wall-modelled LES (discussed in Chapter 5), these oscillations can potentially contaminate the solution significantly. Moreover, based on the experience of the author, when an unstructured grid is used, divergence of the entire simulation can be expected. Therefore, besides for linear interpolation, the linear-upwind stabilised transport (LUST) scheme [103, 56] is also considered a candidate for discretising the convective term in (3.11). This scheme computes the face-centred value using a weighted average of the value obtained by linear interpolation (75%), and that obtained using a second-order upwind scheme (25%). The accuracy of the scheme is thus second-order, but the oscillations coming from linear interpolation are reduced due to the numerical dissipation coming from the upwinding. For the diffusive cell-face fluxes, the linear scheme can be used without negative side effects.

To obtain a fully discrete formulation, the obtained semi-discrete form should be numerically integrated in time. To that end, a second-order accurate implicit backward-differencing scheme is used (see [34] for details).

The solution of the governing equations involves two steps that form the PISO\(^2\) pressure-velocity coupling algorithm [32, 21]. First, the linearised momentum equation (3.11) is solved to obtain the velocity field. This field is, however, not necessarily divergence free. The second step is solving a Poisson-type equation for the pressure, which is derived from (3.8). The obtained pressure values are then used to correct the velocity field and enforce incompressibility. Commonly, the second step needs to be repeated multiple times to ensure convergence. In the simulations presented here, three to four repetitions were used depending on the flow case.

Note that, by construction, using the FVM implies that we cannot resolve turbulent scales that are smaller than the local cell size. This introduces a natural length scale that can be used to specify the filtering operation (3.7). Let \( V \) be the (local) volume of the computational cell.

\(^2\)PISO: Pressure implicit with splitting operator.
Then, if $\Delta$ is defined as $V^\frac{1}{3}$ and the filter

$$G = \begin{cases} 
\frac{1}{\Delta^3}, & \mathbf{x} \in V \\
0, & \text{otherwise,}
\end{cases}$$

(3.18)
is employed, the resolved part of any flow variable is equal to its local cell-volume average. As explained above, the latter is in turn approximated by the value stored in the cell centre, meaning that filtering with the kernel (3.18) is performed implicitly by the employed discretisation method.

Coupling the filtering with discretisation is extremely common [82] due to the simplifications introduced by eliminating the need for explicit filtering. Recall, for instance, that the stresses due to the filtered-out turbulent fluctuations are referred to as subgrid, which implies a connection between filtering and the grid. However, such a coupling means that the computational mesh is a part of the definition of the governing equations. Consequently, it becomes impossible to study the properties of LES independently of the used discretisation procedure.

In conclusion, how strongly all the modelling choices for LES affect each other is stressed. The density of the grid defines the size of the turbulent structures that can be resolved and thus what part of the turbulent spectrum the SGS model must account for. The grid size also defines the size of the numerical errors that are produced by the employed discretisation practices. The latter are in turn coupled to the SGS model through the LES quantities that the model uses for computing the SGS stresses (e.g. $\bar{S}_{ij}$). Moreover, the dissipation coming from $\nu_{sgs}$-based SGS models is combined with numerical dissipation, leading to the total amount of dissipation that is introduced becoming excessive. This tight coupling between the modelling parameters means that optimising their values independently from each other is impossible. Instead, a combination of all parameters giving the best predictive accuracy should be sought.
4. Inflow Generation

Recall that, for LES, the problem of inflow generation consists of providing the temporally and spatially varying turbulent motions at an inlet boundary of the domain. These motions should correspond as accurately as possible to the solution of the LES equations (3.11)-(3.8) for the flow entering through the inlet. The first section of this chapter gives an overview of the existing inflow generation methods, focusing on precursor-based approaches. In the second section, a new method for generating an inflow turbulent boundary layer is presented.

4.1 Review of Existing Approaches

It is possible to distinguish two main approaches to generating turbulent inflow. One is based on using another simulation, a precursor. The methods that fall into this category are called precursor simulation methods [99] or recycling methods [106, 54]. The other class is based on using a stochastic process, which commonly involves imposing some a-priori known values of one- and two-point statistics of the generated velocity fields. The methods based on this approach are referred to as synthesis or synthetic methods [99, 106, 33]. In this thesis, only precursor methods are considered. The reader interested in a substantial review of synthesis methods is referred to [99, 106, 33, 82].

First, note that the idea of using an auxiliary simulation to generate the inflow only makes sense if the auxiliary simulation itself does not require an inflow generation procedure. A class of flows that falls into this category comprises those with a periodicity in the streamwise direction, for example, fully developed turbulent channel and pipe flows. Hence, it is most straightforward to apply a precursor-based method to a simulation for which it makes sense to prescribe inlet velocity values taken directly from some periodic flow. In that case, the velocity field from the precursor simulation is sampled from a plane perpendicular to the mean flow direction and directly mapped to the inlet of the main simulation. This way of prescribing the inflow is referred to as strong recycling [106]. An example of such a configuration is sketched in Figure 4.1 that illustrates the application of turbulent channel flow for generating inflow for a simulation of a plane asymmetric diffuser.
Despite the conceptual simplicity of strong recycling, several aspects of this procedure are not yet fully understood. One is the relationship between the grids in the precursor and at the inlet of the main simulation. Ideally, the grids should be identical since, in that case, no interpolation is needed to map the sampled data to the inlet grid, and the turbulent inflow only contains structures that this grid can resolve. However, reusing a precursor for multiple simulations using different grids would be highly beneficial. In addition, being able to compute the precursor on a grid that is coarser than the one used at the inflow would decrease the relative cost of the precursor compared to the main simulation. That is important since the computational overhead is one of the main downsides of precursor-based methods compared to synthetic ones.

In [39], the authors analysed the effects of applying a low-pass filter to the velocity fields sampled from a precursor before using them as inflow. This roughly corresponds to using a coarser grid in the precursor. Filtering out eddies of a size smaller than the integral length scale had a negligible effect on the adaptation length, which is a measure of the distance after which the flow recovers to the correct physical behaviour. In [90], instead of using a periodic flow, a precursor simulation of a ZPG-TBL using a coarser grid was used to provide inflow data for a DNS of the same flow. The adaptation length in the DNS was found to be very short.
These studies indicate that using a coarser grid in the precursor simulations is feasible. However, to the author’s best knowledge, no extensive evaluation of this possibility has been published to date.

Another approach to saving computing power and storage size would be to produce a precursor database spanning some limited time frame and then loop through the sampled values several times if inflow for a larger time span is needed. A blending procedure would have to be applied to provide a smooth transition between the velocity values in the beginning and end of the sampled database. In [42], such a procedure is discussed within the context of concatenating results from two independent simulations of homogeneous isotropic turbulence in a periodic box.

Nikitin [66] analysed another problem with utilising a streamwise-periodic precursor. Using turbulent pipe flow as a model problem, the author showed that the periodic structure of the solution to the precursor problem is transferred to the main simulation, thus leading to spurious, non-physical periodicity. This behaviour is shown to be independent of the initial conditions used in the main simulation. This effect can, however, be anticipated to be less prominent when the main simulation is not simply a spatially developing version of the precursor [106].

The discussion is now continued with an overview of precursor-based approaches dedicated to generating an inflow ZPG-TBL. This type of inflow is typical of many canonical turbulent flows but also occurs in applications. As an example, consider the flow over a backward-facing step, which is simulated in Paper VI. The domain of the simulation is shown in Figure 4.2. The flow consists of a ZPG-TBL that enters the domain at the inlet, separates at the step, reattaches further downstream,
and then proceeds to grow towards the outlet. The behaviour of the flow depends significantly on the properties of the inflow TBL (e.g. its thickness in relation to the height of the step). Hence, an accurate inflow generation procedure is a critical component of the simulation setup.

In 1988, Spalart [93] presented a way to simulate a ZPG-TBL at a given Reynolds number using periodic boundary conditions in the streamwise direction. This was made possible by choosing a coordinate system that was ‘fitted’ to account for the growth of the TBL, therefore making the boundary layer thickness $\delta$ and the viscous length scale $\delta_v$, independent of $x$. In particular, the new wall-normal coordinate, $\eta$, was defined as a weighted average of $y^+$ and $y/\delta$.

The key assumptions to be made were that, in the streamwise direction, the growth of $\delta$ and $\delta_v$ is slow and so is the change in the mean and variance values of the velocity. The components of velocity could then be represented as follows:

$$u_i(x, \eta, z, t) = \langle u_i \rangle(x, \eta) + A_i(x, \eta)u_{i,p}(x, \eta, z, t), \quad (4.1)$$

where $A_i(x, \eta)$ are proportional to the standard deviation of the values of the corresponding velocity components, and $u_{i,p}$ represent the fluctuations. By assumption, $\langle u_i \rangle$ and $A_i$ are slow-varying functions of $x$. Contrary to this, the functions $u_{i,p}$ are fast varying. By definition, their mean values are zero, and the standard deviations are independent of $x$. Thus, it is possible to apply streamwise-periodic boundary conditions to $u_{i,p}$.

Expressing the Navier-Stokes and continuity equations in the new coordinate system leads to numerous extra source terms. This makes implementing Spalart’s method in a general-purpose CFD code a tedious job and is probably the reason it did not find widespread use [107, 54]. However, it is an accurate method to generate inflow corresponding to a ZPG-TBL at a particular Re-number.

Wu et al. [107] used the ideas of Spalart [93] presented above to develop a method that is suitable for a conventional simulation of a spatially developing TBL that does not require the introduction of special terms into the governing equations. The key idea of the method is to use the velocity values from a plane located downstream of the inlet to obtain the inflow. The plane is commonly referred to as the recycling plane, and in the review [106], the methods based on this idea are referred to as weak recycling methods.

The authors of [107] proposed that, at each time step, the velocity field at the recycling plane is decomposed according to (4.1). The similarity coordinate $\eta$ is defined as in [93]. The ‘periodic’ part of the signal, $u_{i,p}$, is then copied to the inflow plane. Under the assumption of a slow change of the mean and standard deviation values of $u_i$ with $x$, the derivatives of these statistical quantities with respect to $x$ can approximately be
considered constant. This provides a way to evaluate them at the inlet. For the mean velocity components, it follows that:

\[
\frac{\langle u_i \rangle(x_{\text{mid}}, \eta, t) - \langle u_i \rangle(x_{\text{in}}, \eta, t + \Delta t)}{\langle u_i \rangle(x_{\text{rec}}, \eta, t) - \langle u_i \rangle(x_{\text{in}}, \eta, t + \Delta t)} = \frac{x_{\text{mid}} - x_{\text{in}}}{x_{\text{rec}} - x_{\text{in}}},
\]

(4.2)

where the index ‘mid’ refers to the location half way between the inlet and the recycling plane, the index ‘in’ refers to the inlet, and the index ‘rec’ refers to the recycling plane. A similar relationship can be written for the values of the standard deviations. Together with the periodic part, \(u_{i,p}\), this gives all the components required to reconstruct the signal at the inflow according to (4.1).

The same authors, Lund, Wu, and Squires, later introduced another weak recycling method \([54]\), sometimes referred to as the LWS method in the literature. Here, the velocity signal is instead decomposed using Reynolds decomposition into a mean and fluctuating part. The coordinate \(\eta\) is not employed. Instead, the velocity values themselves are scaled using inner and outer coordinates to produce two separate profiles.

The rescaling for the mean streamwise velocity is inferred from equations (2.2) and (2.3). Since these equations are valid for any \(x\), it directly follows that:

\[
\langle u_i \rangle_{\text{in}}^{\text{inner}}(y_{\text{in}}^+) = \gamma \langle u_i \rangle_{\text{rec}}^{\text{inner}}(y_{\text{in}}^+),
\]

(4.3)

\[
\langle u_i \rangle_{\text{out}}^{\text{outer}}(\eta_{\text{in}}) = \gamma \langle u_i \rangle_{\text{rec}}^{\text{outer}}(\eta_{\text{in}}) + (1 - \gamma) U_0,
\]

(4.4)

where \(\gamma\) is the ratio of friction velocities, \(u_{\tau,\text{in}}/u_{\tau,\text{rec}}\), and \(\eta = y/\delta\) is the outer coordinate. The mean wall-normal velocity values are treated in a more ad-hoc manner:

\[
\langle v \rangle_{\text{in}}^{\text{inner}}(y_{\text{in}}^+) = \langle v \rangle_{\text{rec}}^{\text{inner}}(y_{\text{in}}^+),
\]

(4.5)

\[
\langle v \rangle_{\text{in}}^{\text{outer}}(\eta_{\text{in}}) = \langle v \rangle_{\text{rec}}^{\text{outer}}(\eta_{\text{in}}).
\]

(4.6)

This is motivated by the relative unimportance of the wall-normal component compared to the streamwise. The fluctuations are assumed to scale with \(u_\tau\) throughout the whole TBL:

\[
\langle u'_i \rangle_{\text{in}}^{\text{inner}}(y_{\text{in}}^+) = \gamma \langle u'_i \rangle_{\text{rec}}^{\text{inner}}(y_{\text{in}}^+),
\]

(4.7)

\[
\langle u'_i \rangle_{\text{in}}^{\text{outer}}(\eta_{\text{in}}) = \gamma \langle u'_i \rangle_{\text{rec}}^{\text{outer}}(\eta_{\text{in}}).
\]

(4.8)

The inner and outer profiles are blended using a weighting function.

The LWS weak recycling method is widely used and has been extended to work with compressible flows, environmental flows, and flows with surface roughness (see [106] for references). Nevertheless, weak recycling methods suffer from some inherent downsides. One issue that has received significant attention is the choice of the location for the recycling plane.
The plane should be placed sufficiently far away for the distance to the inlet to be larger than the size of the largest coherent structures that are present in the TBL. Otherwise, there is a risk of introducing non-physical periodic forcing. In [91], the authors placed the recycling plane as far as $850\theta_{in}$ to avoid this problem. To rectify this issue, in [94], a shift of the velocity values in the spanwise direction was added to the rescaling procedure, and in [35], the velocity values were instead mirrored.

The weak coupling mechanism can also give rise to problems in the beginning of the simulation since the non-physical values at the recycling plane are copied to the inlet and therefore remain inside the domain. In [53], this issue was addressed by making the placement of the recycling plane dynamic.

What is also important is that, while the rescaling procedure shrinks the turbulent structures in the wall-normal direction to account for the growth of the TBL, the spanwise lengths remain unchanged. Thus, the turbulent structures are distorted via this ‘compression’ in $y$, and the inflow is no longer a solution to the LES equations (3.8) and (3.11). The distortion is proportional to the distance to the recycling plane.

4.2 Strong Recycling of Channel Flow for TBL Simulations

4.2.1 Definition of the Method

This section describes a new framework for generating inflow conditions for a flow problem with a ZPG-TBL at the inlet, introduced in Paper I. In Chapter 2, the similarities between the TBL and channel flow were discussed. Specifically, it was shown that the mean velocity profiles are close to being identical in the inner layer. This suggests that velocity values taken from a precursor channel flow simulation can be used as an inflow condition for a TBL. However, several issues must be mentioned.

First, the mean wall-normal velocity in a fully developed channel flow is zero, which is not so in a TBL. Possibly, this is not a very serious issue due to the small magnitude of the wall-normal velocity compared to the streamwise component. Recall that a similar argument was made in [54].

Second, while in the inner layer the channel and ZPG-TBL flows agree well, this is no longer the case in the outer layer. This concerns both the mean streamwise velocity and the Reynolds stress tensor. In particular, in channel flow, the Reynolds stresses remain non-zero even in the core of the channel, whereas in a ZPG-TBL, they decay to zero as the flow approaches the free stream.
The above discrepancies can be expected to lead to the presence of an adaption region. A detailed analysis of its length is provided in Paper I, and a summary is given in Section 4.2.3 below.

Figure 4.3. Schematic picture illustrating the use of a channel flow precursor simulation to generate inflow conditions for the main simulation. The precursor simulation is set up to match the desired momentum thickness of the main simulation, as described in Section 4.2.2.

The proposed procedure is illustrated schematically in Figure 4.3. The parameters of the channel flow are determined from those of the main simulation. It is assumed that the inlet is rectangular,

\[ 0 < y < b_m, \quad 0 < z < h_m, \]

where \( b_m \) denotes the width of the inlet, and \( h_m \) denotes its height. The inflow TBL is assumed to be attached to the wall located at \( y = 0 \). The velocity components are prescribed at the inlet using values obtained from the precursor simulation without any additional manipulation.

As discussed in Chapter 2, the TBL is fully characterised by one of the Re-numbers listed in (2.1). Here, it is assumed that \( \text{Re}_\theta \) is provided and thus the momentum thickness of the TBL, \( \theta_{in} \), is known, but the proposed framework can easily be reformulated for the case in which \( \text{Re}_{\delta^*}, \text{Re}_{\delta_{yy}} \), or \( \text{Re}_\tau \) is defined instead.

As discussed in Chapter 2, the TBL is fully characterized by one of the Re-numbers listed in (2.1). Here, it is assumed that \( \text{Re}_\theta \) is provided, and thus the momentum thickness of the TBL, \( \theta_{in} \), is known, but the proposed framework can easily be reformulated for the case when \( \text{Re}_{\delta^*}, \text{Re}_{\delta_{yy}} \) or \( \text{Re}_\tau \) is defined instead.

The full list of parameters specifying the inflow is:

\[ (b_m, h_m, U_0, \theta_{in}, \nu). \]

Given their values, the task is to formulate the most appropriate channel flow precursor simulation. The proposition is to match \( \text{Re}_\theta \), so that it is identical for the precursor simulation and the inflow TBL. Momentum thickness is not commonly used as a characteristic length scale for channel
flow, but it can nevertheless be computed. The upper limit of the integral defining $\theta$ should be set to the channel half-height $\delta$, which roughly corresponds to the boundary layer thickness. The closest analogue to the free-stream velocity $U_0$ is then the centreline velocity $U_c$. Hence, for channel flow the momentum thickness-based Re-number is defined as follows:

$$
\text{Re}_\theta = \frac{U_c}{\nu} \int_0^\delta \left( 1 - \frac{\langle u \rangle}{U_c} \right) dy. \quad (4.9)
$$

To make the setup of the precursor easier, a method of determining $\text{Re}_b$ given $\text{Re}_\theta$ is provided below. The full list of precursor parameters that need to be determined therefore consists of the three quantities constituting $\text{Re}_b$ and the length and width of the computational domain:

$$(l_p, b_p, \delta, U_b, \nu_p).$$

4.2.2 Determination of the Precursor Parameters

A procedure in four steps to determine the parameters $(l_p, b_p, \delta, U_b, \nu_p)$ for the precursor simulation from the parameters $(b_m, h_m, U_0, \theta_m, \nu)$ of the main simulation is now proposed.

**Step 1** The width of the domain, the kinematic viscosity, and the momentum thickness are matched to those of the main simulation, i.e. $b_p = b_m$, $\nu_p = \nu$, and $\theta_p = \theta_m$.

**Step 2** The bulk velocity for the precursor simulation is determined as follows. The centreline velocity of the precursor is matched to the free-stream velocity of the main simulation, i.e. $U_c = U_0$. The bulk velocity is then determined by the following relation:

$$
\frac{U_c}{U_b} = f_1(\text{Re}_\theta), \quad (4.10)
$$

for channel flow. To obtain a computable semi-empirical expression, the ansatz, $f_1(\text{Re}_\theta) \approx 1 + \alpha_1 \text{Re}_\theta^{\beta_1}$, is fitted to the DNS data [45], which includes results for channel flow at five different Re-numbers. The resulting coefficient values are given in Table 4.1, and the fit to the data is illustrated in the top plot in Figure 4.4.

**Step 3** The length scale of the precursor simulation is defined by $\delta$, which is determined from the relation:

$$
\frac{\delta}{\theta} = f_2(\text{Re}_\theta), \quad (4.11)
$$

for channel flow. Similar to Step 2, an ansatz, $f_2 \approx \gamma_2 + \alpha_2 \text{Re}_\theta^{\beta_2}$, is fitted to the DNS data of [45]. See Table 4.1 and the bottom plot in Figure 4.4 for the values of the coefficients and an illustration of the fit, respectively.
Figure 4.4. Illustration of the functions $f_1$ and $f_2$. The full lines are given by the semi-empirical expressions with coefficients from Table 4.1. The DNS values are taken from [45].

Table 4.1. Coefficients in the semi-empirical expression for $f_1$ and $f_2$. The coefficient values are obtained by a least-squares fit of the two ansatzes to the DNS data from [45].

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\alpha_i$</th>
<th>$\beta_i$</th>
<th>$\gamma_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3427</td>
<td>0.1287</td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>$2.603 \cdot 10^{-4}$</td>
<td>0.9834</td>
<td>11.28</td>
</tr>
</tbody>
</table>
Step 4 The length of the channel flow domain is determined to be proportional to the channel height. The analysis of two-point correlations shows that the value, $l_p = 8 \delta$, can be considered sufficiently large.

Since the precursor simulation is set up based on the requirements of the main simulation, it is natural to also do this with the grid generation. It is thus recommended that the grid on the inflow patch matches the grid on the sampling plane of the precursor simulation and that the same time step is used in the two simulations.

4.2.3 Adaption Length

To evaluate the performance of the method, it has been applied to an LES of a ZPG-TBL flow over a flat plate with $Re_\theta \approx 830$ at the inlet. Two additional simulations of the same flow have been conducted. One used no inflow generation procedure at all, which required increasing the streamwise extent of the domain to incorporate the region upstream of $Re_\theta \approx 830$. Another simulation used the rescaling procedure defined by the LWS weak recycling method [54] to generate the inflow.

The analysis of the integral quantities of the TBL as well as statistical moments of its velocity field showed that the adaption length exhibited by the proposed method is on par with that produced by the weak recycling [54]. Its size is estimated to be $\approx 270\theta_{in}-370\theta_{in}$. Furthermore, the accuracy of the results downstream of the adaption region were similar to those of the simulation with no inflow generation procedure employed.

The above allows the conclusion that the proposed inflow generation procedure can be successfully applied to flows with an inflow ZPG-TBL and that the accuracy can be expected to be on par with other state-of-the-art approaches. A practical advantage of the method is that it requires minimal effort to implement if the employed solver supports surface sampling and reading boundary data from the disk. This is the case for OpenFOAM and most other general-purpose solvers. Compared to approaches based on weak recycling, the proposed method is more robust because there is also no backward-coupling between the flow field in the main simulation and the generated inflow values.
5. Wall Modelling

Recall that the number of grid points needed for a wall-resolved LES of a turbulent boundary layer scales as $\text{Re}^{1.85}$, making simulation of high Reynolds number flows infeasible. The reason behind the unfavourable scaling was pointed out to be the need to resolve the inner layer of the TBL, where the turbulent structures are of size $\sim \delta_v$. This motivates the approach known as wall-modelled LES (WMLES), where special modelling is introduced in order to account for the dynamics of the inner layer, thus alleviating the need to resolve it with the grid. It can be shown, [12, 79, 47], that this leads to a linear scaling of the required number of grid points with the Reynolds number, $N \sim \text{Re}$, significantly expanding the range of Reynolds numbers that can be simulated.

A large number of WMLES approaches have been developed, reviews of which can be found in the literature [72, 43, 6]. Two major classes can be distinguished: wall-stress modelling and hybrid LES/RANS [43]. The fundamental difference between them lies in the definition of the extent of the LES domain. In wall-stress modelling, it covers the whole near-wall region, including the inner layer. The job of the wall model is thus to provide the correct boundary condition for the LES equations at the wall. Note that the no-slip condition cannot be used due to the coarseness of the WMLES grid [5]. It will be shown below that the appropriate physical quantity to be prescribed is the filtered wall shear stress, $\tau_w$. To predict its local value, wall-stress models typically employ the LES solution sampled from a single point located at some distance $h$ from the wall, see the left plot in Figure 5.1. This input is fed into the equations governing the model, which are commonly derived from RANS.

![Figure 5.1. Schematic representation of two wall modelling approaches: wall-stress modelling (left) and hybrid LES/RANS (right).](image)

Hybrid methods have historically been developed [95] for lower fidelity simulations, in which the entire TBL, including the outer layer, is not resolved by the LES grid. Unlike wall-stress modelling, the computational
domain is divided into two regions in which different sets of equations govern the solution. Away from walls, LES modelling (equations (3.8)-(3.11)) is applied, whereas in the near-wall region the RANS equations (3.4)-(3.5) are solved instead, see the right plot in Figure 5.1. Depending on the employed method, the location of the interface between the two regions is either defined explicitly [15, 100] or by controlling the density of the grid [95, 94, 88]. By placing the interface in the overlap region, hybrid approaches can be used for WMLES [67, 88]. A fundamental difficulty is properly coupling the RANS and LES solutions, for which different treatments have been proposed in the literature [14, 13, 88].

There are benefits and disadvantages to both wall-stress modelling and hybrid LES/RANS approaches, and pursuing further development of both is, therefore, fruitful for the advancement of WMLES. In this work, the object of investigation is wall-stress modelling, and in the sections that follow various aspects of this methodology are discussed in detail. In particular, in Section 5.1 wall-stress modelling is fully defined within the here-employed numerical framework. In Section 5.2, this is followed by a discussion of the properties of the filtered wall shear stress. As indicated previously, the local value of this quantity is what wall-stress models aim to predict. The existing approaches for doing this are then presented in Section 5.3. How the accuracy of the wall modelling depends on $h$ is investigated in Section 5.4. The influence of modelling parameters other than the wall-stress model itself is considered in Section 5.5. Section 5.6 presents a methodology for unstructured mesh generation suitable for WMLES. Finally, Sections 5.7 and 5.8 describe two novel wall-stress modelling techniques.

5.1 Definition of Wall-Stress Modelling

In this section, wall-stress modelling is mathematically defined within the here-employed discretisation framework, which is the finite volume method. For convenience, a local Cartesian coordinate system is introduced, with the axes oriented such that the wall lies in the $x_1$-$x_3$ plane, and $x_2$ points in the wall-normal direction into the fluid domain. Moreover, the $x_1$ axis is aligned with the mean wall-parallel component of velocity.

The effect of wall friction enters the LES momentum equation (3.11) via the term $\partial \tau_{ij} / \partial x_j$, where $\tau_{ij} = \tau_{ij}^{sgs} + 2\nu \hat{S}_{ij}$. To derive the discrete form of this term, the procedure defined in Section 3.4 can be used. Consider a finite volume cell with a face of size $S_w$ adjacent to the wall. Integrating $\partial \tau_{ij} / \partial x_j$ over the volume of the cell and then applying the Gauss-Ostrogradsky theorem results in an integral over the cell surface, $\int \tau_{ij} n_j dS$, where $n$ is the surface normal. This integral can be further
decomposed into a sum of integrals over the cell faces. The contribution to the sum from the face at the wall is obtained as follows:

\[
\int_{S_w} \tau_{ij} n_j dS = - \int_{S_w} \tau_{i2} dS \approx \tau_{w,i2} S_w. \quad (i = 1, 3) \tag{5.1}
\]

Here, \( \tau_{w,12} \) and \( \tau_{w,32} \) are the two wall-parallel components of the filtered wall shear stress vector \( \tau_w \), the magnitude of which is denoted as \( \tau_w \). In a wall-resolved simulation, the correct value of \( \tau_w \) is obtained directly by applying the no-slip condition to the filtered velocity.

By contrast, in WMLES, the employed grid is too coarse to resolve the wall-normal velocity gradient. Hence, \( \tau_w \) has to be supplied by a wall model instead. This is achieved by applying the following three-step procedure, see also the left plot in Figure 5.1.

**Step 1** The values of \( \bar{u}, \bar{p}, \) or quantities derived from them (e.g. the pressure gradient) are sampled from a cell centre in the LES domain, located at some wall-normal distance from the wall, \( h \). The sampled values serve as input to the wall model.

**Step 2** From the sampled input values, the local value of the magnitude of the filtered wall shear stress, \( \tau_w \), is computed using the wall model.

**Step 3** The computed \( \tau_w \) is enforced at the given face centre by prescribing appropriate values for \( \tau_{w,12} \) and \( \tau_{w,32} \).

At each time step of the simulation, this algorithm is applied at each wall face. The first two steps are given attention in the sections below. Here, the discussion continues with considering Step 3. Employing the no-slip condition for \( \bar{u}_i \) and the Boussinesq approximation for SGS modelling, the standard finite volume approximation of \( \tau_{i2} \) at the wall gives the following relations:

\[
\tau_{w,i2} = (\nu + \nu_{sgs}) f \frac{\bar{u}_i P}{\Delta x_2}, \quad (i = 1, 3) \tag{5.2}
\]

The sub-script \( P \) implies evaluation in the centre of the wall-adjacent cell, the sub-script \( f \) evaluation in the centre of the wall face, and \( \Delta x_2 \) is the wall-normal distance between these two points. The correct \( \tau_w \) is then enforced if the following value of \( \nu_{sgs} \) is set at the wall face:

\[
\nu_{sgs} = \frac{\tau_w}{\left[ (\bar{u}_{1,P}/\Delta x_2)^2 + (\bar{u}_{3,P}/\Delta x_2)^2 \right]^{1/2} - \nu}. \tag{5.3}
\]

Note that here the no-slip condition for \( \bar{u}_i \) is a part of the wall modelling procedure, and not a physical boundary condition, as it is in the case of a wall-resolved simulation. Furthermore, an implicit assumption is made that the local \( \tau_w \) is aligned with the wall-parallel velocity in the centre of the wall-adjacent cell.
5.2 Relationship Between $\tau_w$ and $\overline{\tau}_w$

According to the above, the ideal wall-stress model is capable of computing the exact values of $\overline{\tau}_w$. Prior to reviewing the approaches for predicting this quantity, the relationship between $\overline{\tau}_w$ and its unfiltered counterpart, $\tau_w$, is analysed.

By definition, $\overline{\tau}_w$ is an approximation of the average of the corresponding unfiltered quantity over the surface of the wall face. Hence, the difference between $\overline{\tau}_w$ and $\tau_w$ is mainly in the fluctuations that they exhibit. In particular, $\overline{\tau}_w$ fluctuates less, due to part of the fluctuations being filtered out. It will be shown below that accurately predicting the fluctuations, $\overline{\tau}_w'$, is significantly more challenging than just predicting the mean of $\overline{\tau}_w$. Therefore, it is interesting to analyse under what circumstances $\overline{\tau}_w'$ can be considered negligible.

In WMLES, the aim is to resolve the TBLs outer layer, therefore, the density of the grid is appropriately defined in terms of the outer length scale, $\delta$. A detailed discussion of the grid resolution will be given in Section 5.5. Here, assume that the size of the wall face is $(\delta/20)^2$. Note that as the Reynolds number of the flow, $Re_\tau = \delta/\delta_\nu$, grows, the size of the wall face in wall units will increase. This implies that averaging $\tau_w$ across the surface of the face will filter out an increasingly large portion of its fluctuations. The wall-parallel extent of the turbulent structures located in the direct vicinity of the wall can be estimated to be $\approx 750\delta_\nu \times 100\delta_\nu$ [11, 75]. Based on that, the wall face large enough to make the filtering average out all of $\tau_w'$ has been estimated [74] to be of size $\approx 1500\delta_\nu \times 700\delta_\nu$. For simplicity, assume that a square face of size $(1000\delta_\nu)^2$ is sufficient as well. Given the size of the face in outer scaling, $(\delta/20)^2$, it follows that $Re_\tau = 20000$. A general estimate for the Re-number at which the wall model can consider $\overline{\tau}_w'$ to be negligible is thus $Re_\tau \sim 10^4$.

In relation to the above, it is important to note that a large number of WMLES studies consider flows at $Re_\tau \leq 2000$ (e.g. [101, 20, 71, 69, 68]). The reason is often the absence of reliable reference data for flows at higher Re-numbers, in particular for more complicated configurations, such as flows with separation. It is therefore interesting to investigate what the relationship between $\overline{\tau}_w$ and $\tau_w$ is at a relatively low Re-number. This has been done in Paper IV where a wall-resolved LES of channel flow at $Re_\tau = 1000$ has been used to obtain a time-series of $\tau_w$ and the corresponding values of $\overline{\tau}_w$, computed by averaging the unfiltered quantity over a surface of size $(\delta/20)^2$ in the course of the simulation.

Figure 5.2 shows a scatter plot of $\tau_w^+ = \tau_w/(\overline{\tau}_w)$ versus $\overline{\tau}_w^+ = \overline{\tau}_w/(\overline{\tau}_w)$ produced using the obtained data. As expected, the mean values of the two signals agree well, the relative difference being $\approx 0.5\%$. The correlation coefficient between them is 0.794, and generally only the extreme
values of $\tau_w$ get filtered out. This implies that the wall modelling should be capable of predicting a significant part of the inner layer dynamics. Nevertheless, the difference in the root-mean-square values of the fluctuations is significant. For the unfiltered quantity, $\tau_w^{\text{rms}} / \langle \tau_w \rangle = 0.418$, which is in line with experimental results [1]. The corresponding value for $\overline{\tau}_w$ is 0.288.

In summary, it is stressed that $\tau_w$ and $\overline{\tau}_w$ are different quantities and the relationship between them is highly dependent on the Re-number of the flow. This statement may be considered trivial, but, unfortunately, it is common in the WMLES literature to refer to both $\tau_w$ and $\overline{\tau}_w$ as (the magnitude of) the wall shear stress, which can be misleading unless it is clear that the discussion pertains to the mean values of these quantities. Even at relatively low Re-numbers, any estimates for higher-order statistics of the unfiltered wall shear stress cannot be used to judge the quality of the wall model’s prediction of $\overline{\tau}_w$, as is sometimes done [68]. Finally, only at high Re-numbers (Re $\sim 10^4$ and above) can the fluctuations of $\tau_w$ be considered averaged out by the spatial filtering. Conclusions regarding the accuracy of wall models based on WMLES of lower Re-number flows should be considered conservative.

In conclusion, it is important to note that the above analysis regarding the dependency of $\overline{\tau}_w'$ on the Re-number is based on the classical theory of wall-bounded turbulence that considers the inner layer to be an autonomous dynamical system [75]. This theory has been challenged in recent studies [57, 58, 9] indicating that the processes in the outer layer have a significant effect on $\tau_w$. Further development of this theory is necessary to fully understand its implications for wall modelling, but it
is plausible that even at high Re-numbers the ideal wall model should be able to take the fluctuations of $\overline{\tau'_w}$ into account.

5.3 Methods for Computing $\overline{\tau_w}$

Methods for predicting $\overline{\tau_w}$ given the solution to the LES equations sampled at some distance $h$ from the wall are now reviewed. The existing approaches can be classified based on the type of the underlying equations. Three wall-stress model classes are thus distinguished: algebraic, ordinary differential equation (ODE)-based, and partial differential equation (PDE)-based.

5.3.1 Algebraic Models

Being the most simple, algebraic models were the first to be developed. The study of Schumann [86], dated 1975, considers WMLES of fully developed turbulent channel flow. Given the magnitude of the mean pressure gradient driving the flow, it is possible to compute the mean of $\tau_w$ a-priori. Let $u$ denote the magnitude of the wall-parallel component of the velocity vector. The value of $\langle u \rangle$ at the sampling point can be obtained using a law of the wall, such as the log-law. Based on this, Schumann proposed computing the streamwise component of the filtered wall shear stress as

$$\overline{\tau_{w,12}} = \frac{\overline{u'_1|h}}{\langle u \rangle_h} \langle \tau_w \rangle,$$  \hspace{1cm} (5.4)

where $\cdot|h$ is used to denote the value at the sampling point, which was located at the grid point closest to the wall. For the spanwise component no wall model was applied since both $\langle u_3 \rangle$ and $\langle \tau_{w,32} \rangle$ are zero. Instead, it was computed directly by evaluating the wall-normal gradient of the spanwise velocity using a first-order finite difference stencil. Since the mean of $\overline{\tau_{w,12}}$ is assumed to be known a-priori, (5.4) is essentially a model for its fluctuations, $\overline{\tau'_{w,12}}$. Notably, it is assumed that $\overline{u'_1|h}$ and $\overline{\tau'_{w,12}}$ are in phase.

The assumption of prior knowledge of the mean wall shear stress is impractical. In fact, predicting the distribution of $\langle \tau_w \rangle$ can often be one of the major purposes of a simulation. In 1987, Grötzbach [28] developed a more flexible approach, which is at the core of most contemporary algebraic wall models. Consider a law of the wall, for example, the log law (2.4). Moving all terms to the left-hand-side, it can be rewritten as

$$\langle u \rangle / \sqrt{\langle \tau_w \rangle} - \frac{1}{\kappa} \ln \left( x_2 \sqrt{\langle \tau_w \rangle} / \nu \right) - B = 0.$$  \hspace{1cm} (5.5)
If an approximation of $\langle u \rangle$ can be obtained from the LES solution at the sampling point ($x_2 = h$), the above expression constitutes a non-linear algebraic equation for the unknown mean wall shear stress $\langle \tau_w \rangle$. The equation may be solved using an iterative procedure, e.g. the Newton-Raphson method. It is stressed that while the log-law is used to exemplify the approach, and was indeed used by Grötzbach, any law of the wall can be used, including Spalding’s (2.5) and Reichardt’s (2.6).

A law of the wall-based algebraic wall model can be applied in several ways. One is to use it to compute $\langle \tau_w \rangle$ and employ a separate model to compute the fluctuations. This is the original proposition of Grötzbach, who employed Schumann’s model (5.4) to get $\tau_w'$. Combining an algebraic wall model for the $\langle \tau_w \rangle$ with a separate model for the fluctuations (based on the inner-outer layer interaction theory [57, 58, 9]) was also done by Sidebottom et al. in [89]. To approximate $\langle u \rangle|_h$, time averaging can be used. For flows with statistically homogeneous directions, spatial averaging can be used as well.

Another alternative is to simply consider $\tau_w'$ to be negligible and impose $\langle \tau_w \rangle$ at the boundary in the course of the whole simulation. This was done in [44], resulting in a good agreement with reference data for the mean velocity profiles in WMLES of turbulent channel flow and ZPG-TBL flow. It is concluded that ‘the fluctuating component of the wall shear stress is not an essential ingredient for the success of WMLES.’ This is further supported by the above-mentioned study by Sidebottom et al. [89], where the explicit model for $\tau_w'$ was shown to have a negligible effect on the predicted $\langle \tau_w \rangle$. However, it may be risky to draw general conclusions regarding the significance of $\tau_w'$ based solely on WMLES of the simplest wall-bounded flows. Also, the prediction of the fluctuations of $\tau_w$ (its higher-order statistics) may be of interest as such, even if it is not necessary to obtain an accurate mean.

The third possible way to use the law of the wall is to consider it to be approximately valid for $\tau_w$ and the magnitude of the wall-parallel component of the instantaneous LES velocity, denoted $\bar{u}$. This is, arguably, the most widespread way of applying algebraic wall-stress modelling and also how it is used in Papers III, IV, and VI of this thesis. Note that, in this case, $\bar{u}$ and $\tau_w$ are again assumed to be in phase, similar to Schumann’s model (5.4).

The capability of a law of the wall to predict $\tau_w'$ is analysed in Paper IV for the case of turbulent channel flow at $Re_\tau = 1000$. A wall-resolved LES of this flow was performed, and reference $\tau_w$ values were computed using surface averaging, as already discussed in Section 5.2. Reference values of $\bar{u}$ were obtained in a similar manner, by averaging the wall-resolved velocity field over a cubic volume. Moreover, several time-series of $\bar{u}$ were sampled, corresponding to different values of $h$. 

48
Each reference \( \bar{u} \) signal was then used as input to an algebraic wall model based on Spalding’s law (2.5) to obtain predictions of \( \bar{\tau}_w \) that could then be compared to corresponding reference values sampled from the wall-resolved simulation. This way of analysing the accuracy of a wall model will be referred to as an a-priori analysis below. This is opposed to actually performing a WMLES using the studied model.

To quantify the accuracy of the predictions of \( \bar{\tau}_w' \), the correlation coefficient between the reference and the predicted values of the friction velocity \( \bar{u}_r \) was computed. The result is shown in Figure 5.3. For the smallest considered value of \( h/\delta = 0.025 \), the obtained correlation coefficient is \( \approx 0.62 \). Farther from the wall, it quickly decreases and at \( h/\delta \approx 0.3 \), which is the upper bound of the validity of the log-law, its value is \( < 0.1 \). Analysis also shows (see Paper IV) that the standard deviation of \( \bar{\tau}_w \) is consistently under-predicted by the wall model, with accuracy also worsening with \( h/\delta \). These results are overall disappointing but not entirely unexpected. In fact, the obtained correlation coefficient values are similar to those between \( \bar{\tau}_w \) and the velocity values \( \bar{u} \) themselves.

![Correlation coefficient between the reference \( \bar{u}_r \) signal and that produced by an algebraic model based on Spalding’s law, \( \bar{u}_r^* \). Reference \( \bar{u} \) sampled at different \( h/\delta \) used as input to the wall model.](image)

Figure 5.3. Correlation coefficient between the reference \( \bar{u}_r \) signal and that produced by an algebraic model based on Spalding’s law, \( \bar{u}_r^* \). Reference \( \bar{u} \) sampled at different \( h/\delta \) used as input to the wall model.

Nevertheless, it can be claimed that, using a small \( h/\delta \), an approximation of \( \bar{\tau}_w \) can be obtained by considering a law of the wall to be valid for the instantaneous LES quantities. Recall also that in Section 5.2 it was shown that, for the considered case of channel flow at \( \text{Re}_\tau = 1000 \), the \( \bar{\tau}_w \) signal still contains the footprint of the inner-layer dynamics. At higher Re-numbers, the problem of low correlation between \( \bar{u} \) and \( \bar{\tau}_w \) is likely to remain, but perhaps the level of fluctuations will be overestimated in-
stead. Temporally filtering the input velocity signal, as proposed in [109, 108], can, in that case, be speculated to improve results.

Algebraic wall models have been applied to a large variety of flows. This includes canonical cases, such as turbulent channel flow (see e.g. [86, 28, 101, 44, 108, 22]), the flow over periodic hills [101, 20], the ZPG-TBL flow over a flat plate [44], and the flow over a backward-facing step (BFS) [8]. More applied flow configurations have also been considered, for example, a flow around an airplane [46], around a car [2], and around a ship hull in model scale [49]. This list is certainly not exhausting.

The overall consensus regarding the accuracy of the wall modelling is that, as long as the state of the TBL is consistent with the employed law of the wall, an accurate distribution of $\langle \tau_w \rangle$ can be obtained. When this is not the case, the performance is worse, although the level of the degradation in accuracy appears to be heavily case-dependant. To make algebraic models applicable to a wider range of flows, laws of the wall incorporating the strength of the pressure gradient have been developed, see [87, 24]. As of right now, the performance of wall-stress models based on these laws has not been extensively investigated.

In this thesis (Papers III, VI), an algebraic wall-stress model was used in WMLES of three of the above-mentioned canonical flows: turbulent channel flow, the flow over a BFS, and flat-plate ZPG-TBL flow. The results from some of these simulations are considered further below, but with respect to the discussion above it is noted already here that for the flow over a BFS good accuracy of $\langle \tau_w \rangle$ predictions was obtained in the recirculation region, where the law of the wall used by the model is not valid.

5.3.2 PDE-based Models

A PDE-based wall-stress model can be derived by considering the RANS equations (3.5) near the wall. Analysis shows [84] that the wall-parallel diffusion terms can be disregarded in this region, which leads to the so-called turbulent boundary layer equations (TBLE) [8, 102, 73, 72]:

$$\frac{\partial}{\partial x_2} \left[ (\nu + \nu_t) \frac{\partial \langle u_i \rangle}{\partial x_2} \right] = F_i,$$

(5.6)

where

$$F_i = \frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_i} + \frac{\partial \langle u_i \rangle}{\partial t} + \frac{\partial}{\partial x_j} \langle u_i \rangle \langle u_j \rangle.$$

(5.7)

Here, $i = 1, 3$, i.e. the wall-parallel directions, whereas $u_2$ can be found from the continuity equation. In the formulation above, the Boussinesq
assumption concerning the form of the Reynolds stress tensor has been made.

The TBLE can be used as a wall-stress model. For that, they should be solved on a separate, ‘embedded’, three-dimensional grid spanning the wall-normal distance between the wall and $h$. In line with the general algorithm for wall-stress modelling (see Section 5.1), the LES velocity values at $y = h$ serve as a boundary condition, and the filtered wall shear stress obtained from solving the TBLE is enforced at the wall boundary of the LES domain. Notably, the pressure gradient distribution can also be obtained from the LES, making the common assumption that the wall-normal variation of pressure in the TBL is negligible [75]. This means that the Poisson equation for pressure does not have to be solved, greatly reducing the computational cost of the wall modelling.

The TBLE have to be complemented by a RANS model for computing $\nu_t$. In principle, any of the multiple existing closures could be applied. In the majority of studies (e.g. [8, 102, 37]) a simple one-dimensional turbulence model is employed. Several $\nu_t$ models of this type are presented below in the context of ODE-based wall-stress models. A notable exception is the work of Diurno et al. [18], where the Spalart-Allmaras model was used instead.

In spite of leading to potentially the most accurate type of wall-stress models, the TBLE-approach is not without drawbacks. One issue is the complexity of implementing this wall model in the setting of a general-purpose CFD solver. Coupling the solution of two sets of PDEs in a parallel, unstructured setting is not simple. A detailed discussion of this issue can be found in [70], which is fully dedicated to the analysis of the associated difficulties and algorithms to resolve them. Another practical issue is that of the generation of the embedded grid. Automatic generation is difficult to implement for arbitrary geometries, and leaving the task to the user significantly complicates the setup process of the simulation case.

More fundamental issues exist as well. One is the fact that the cost of WMLES becomes no longer independent of Re$_\tau$ because the embedded RANS grid has to be wall-resolved in the wall-normal direction. In [69, 70], the authors report that the ratio of sizes of the embedded and LES grids are 0.2-0.5, depending on the flow case. The additional cost of wall modelling is reported to be 100-150% of the LES. The wall model based on the TBLE is thus by far the most computationally expensive among those discussed in this thesis. The cost benefits compared to wall-resolved LES are still very significant, however.

Furthermore, the discussion regarding modelling the fluctuations of $\overline{\tau}_{uw}$, which was made in the context of algebraic models is relevant here as well. When an unsteady velocity signal is used as the input to the TBLE-model, resolved turbulent fluctuations of velocity appear in the solution on the
embedded grid, which further leads to fluctuations in the predicted $\tau_w$. Having resolved turbulence leads to difficulties because $\nu_t$ models are designed to account for the effect of the whole spectrum of turbulent scales. As a result, the modelled Reynolds stresses they introduce are too large. Significant effort has been put into devising a procedure that would modify the computed Re-stresses based on the amount of resolved fluctuations present, see [8, 102, 37, 69]. However, to the best of the author’s knowledge, no studies focusing on the accuracy of $\tau'_w$ produced by PDE models have been published.

The most important question is whether using a PDE-based model leads to a significant improvement in accuracy compared to simpler approaches. Unfortunately, so far few studies have addressed this question. In [102], flow past the trailing edge of an airfoil is considered, and the TBLE-based model is shown to yield the most accurate results. In [68], this model also gave the most accurate predictions, the considered case being the flow over the NASA wall-mounted hump. Thus, PDE-based wall-stress modelling is potentially the most accurate approach, but further studies are needed to verify that.

In this thesis, PDE-based models are not used. To a large extent, this is due to the associated practical issues (implementation, grid generation). The insignificant amount of evidence regarding the superiority of these models compared to simpler approaches is another factor.

5.3.3 ODE-based Models

An alternative to directly solving the TBLE is modelling the terms in (5.7) using the LES solution at the sampling point. In that case, the modelled $F_i$ is a constant source term, and (5.6) takes the form of an ODE, which can be solved for each wall face. In fact, it becomes possible to integrate (5.6) analytically, leading to:

$$\langle \tau_{w,i} \rangle = \frac{\langle u_i \rangle_h - F_i I_1}{I_2}, \quad (i = 1, 3)$$

(5.8)

where

$$I_1 = \int_0^h \frac{x_2}{\nu + \nu_t} \, dx_2, \quad I_2 = \int_0^h \frac{dx_2}{\nu + \nu_t}. \quad (5.9)$$

The magnitude of the stress is then obtained as follows:

$$\langle \tau_w \rangle = \left( \langle u_i \rangle_h \langle u_i \rangle_h + F_i F_i I_1^2 - 2\langle u_i \rangle_h F_i I_1 \right)^{1/2} / |I_2|. \quad (5.10)$$

Getting the value of $\langle \tau_w \rangle$ thus amounts to numerically computing the integrals $I_1$ and $I_2$, which requires a model for $\nu_t$. Commonly, approaches based on the mixing length model coupled with the van Driest damping
function near the wall [19] are used. For example, in [8, 102, 37] the following \( \nu_t \) model is employed:

\[
\nu_t = \nu \kappa x_2^+ \left( 1 - \exp\left( -x_2^+ / A \right) \right)^2, \tag{5.11}
\]

where the values used for the constants are \( \kappa = 0.41 \) and \( A = 17.8 \). Note that (5.11) depends on the value of \( \langle u_\tau \rangle \). Fixed point iteration using (5.10) and (5.11) until convergence is therefore needed. In [3], a structurally similar \( \nu_t \) model is used, but incorporating the magnitude of the rate-of-strain tensor. A model designed to work better with separating flows was developed by Duprat et al. [20]. It is based on a mixed velocity scale introduced in [55]: \( u_{\tau p} = (u^2_\tau + u^2_p)^{1/2} \), where \( u_p = |\nu/\rho(\partial p/\partial x_1)|^{1/3} \). The non-dimensional parameter \( \alpha = u_p^2 / u_{\tau p}^2 \) is defined to measure the relative strength of the shear stress and the streamwise pressure gradient. Note that, unlike \( u_\tau, u_{\tau p} \) does not become zero at separation and reattachment points. Defining \( x_2^* = x_2 \langle u_{\tau p} \rangle / \nu \), the expression for the turbulent viscosity is

\[
\nu_t = \nu \kappa x_2^* \left[ \alpha + x_2^* (1 - \alpha)^{3/2} \right]^{\beta} \left[ 1 - \exp\left( -\frac{x_2^*}{1 + A\alpha^3} \right) \right]^2, \tag{5.12}
\]

where \( A = 17 \) and \( \beta = 0.78 \) are model constants.

Within this ODE-model framework, the key question that needs to be addressed is the modelling of \( F \). Ideally, the model should represent the combined effect of convective, transient, and pressure gradient terms. Commonly, however, only some of the terms are taken into account. The simplest choice, \( F = 0 \), leads to an equilibrium ODE model that is essentially equivalent to using a law of the wall. Results from applying equilibrium ODE-based models can be found in [102, 38, 68], and the strengths and drawbacks of the approach are similar to those of algebraic models. This has also been observed in Paper VI of this thesis, where an equilibrium ODE-based model and an algebraic model based on Spalding’s law (2.5) resulted in very similar predictions of \( \langle \tau_w \rangle \) for the flow over a BFS.

Choosing

\[
F_i = \left. \frac{\partial \langle p \rangle}{\partial x_i} \right|_h \tag{5.13}
\]

has received significant attention due to the role of the adverse pressure gradient in the process of flow separation.

In [20], this treatment of \( F \), coupled with the \( \nu_t \) model given by (5.12), was shown to give improved results for periodic flow over a hill compared to using an algebraic model based on Spalding’s law. In [102] and [10], however, the results obtained when only considering the pressure gradient
term were unsatisfactory. In Paper VI, this was also the case when the flow over a BFS was considered: an equilibrium ODE model performed better than those attempting to account for the pressure gradient. In [43], the authors argue that one must either consider the effect of all the terms in (5.7) or ignore all of them since e.g. the convective and pressure gradient terms are often of the same magnitude, but of opposite sign. The only attempt to model the convective terms appears to be the work of Hickel et al. [29]. Two models were proposed, and an a-priori analysis of the accuracy was performed, using data from a wall-resolved LES of an adverse-pressure-gradient TBL and of shock/boundary layer interaction. The model that performed well in the tests considered the magnitude of the convective term to perfectly balance the pressure gradient at \( x_2 = h \) and decrease linearly towards the wall. Note that this means that \( F \) is a function of \( x_2 \), and not constant, as assumed in the derivation of (5.10). No results from WMLES employing the proposed models seem to have been reported, however. In summary, the correct treatment of \( F \) in ODE-type models remains an open question.

5.4 Influence of the Distance to the Sampling Point

Independently of the employed wall-stress modelling approach, an important parameter is the choice of the distance to the sampling point, \( h \). This section is dedicated to the discussion of the influence of \( h \) on the predictive accuracy of the wall modelling. The analysis primarily focuses on algebraic wall-stress models applied to equilibrium TBLs, but many conclusions should remain valid for other flows and approaches as well.

For a TBL exhibiting a canonical mean velocity profile (see Section 2.1.2), the choice of \( h \) is guided by the following considerations. First, the sampling point should be located in the outer layer \((x_2^+ > 50)\) because that is the region that is properly resolved by the grid. Also, the point should lie in the region where the equation (law) governing the wall model is valid. The extent of this region depends on the model but the end of the logarithmic layer \((x_2/\delta \approx 0.3)\) can be considered to be the upper bound. Within these limits, the choice of \( h \) should be governed by the accuracy of the velocity signal, with the aim to place the sampling point at a location where the accuracy is best.

Kawai and Larsson [38] argue that the above principle excludes the distance to the centre of the wall-adjacent cell as a candidate for \( h \). The motivation is that in this cell the turbulent structures are necessarily under-resolved and, therefore, the accuracy of the velocity signal is low. In their work, this theoretical argument is supported by WMLES of a ZPG-TBL. It is demonstrated that moving \( h \) to a location further away from the wall indeed leads to an improved prediction of \( \langle \tau_w \rangle \). More recent
studies have reported similar results [44, 22]. Recall, however, that the correlation between \( \bar{u} \) and \( \bar{\tau}_w \) decays rapidly with \( h \). Therefore, to capture the fluctuations of \( \bar{\tau}_w \), using the centre of the wall-adjacent cell for wall model input may be optimal. Getting an accurate mean value is, however, commonly a higher priority.

Recall that, in Paper IV, an a-priori analysis of the accuracy of the algebraic wall model based on Spalding’s law is presented, with the reference data collected from a wall-resolved simulation of channel flow at \( \text{Re}_r = 1000 \). Figure 5.4 shows how the obtained relative error in the model’s prediction of \( \langle \bar{\tau}_w \rangle \), denoted \( \epsilon[\langle \bar{\tau}_w \rangle] \), depends on \( h/\delta \). The error is computed with respect to DNS data [45]. The accuracy is clearly lowest when the input \( \bar{u} \) are obtained by averaging the wall-resolved velocity across a wall-adjacent volume (left-most point in the plot, \( h/\delta = 0.025 \)). However, this result cannot be explained using the argument of Kawai and Larsson [38], which is that the poor prediction is due to the inaccuracy of the input velocity signal. The accuracy of the reference \( \bar{u} \) values is excellent and does not change across the whole range of the considered \( h \)-values, see Paper IV.

\[ \begin{align*}
\epsilon[\langle \bar{\tau}_w \rangle] &= \frac{\frac{\bar{\tau}_w}{\langle \bar{\tau}_w \rangle} - 1}{\langle \bar{\tau}_w \rangle} \\
&= \frac{\langle \bar{\tau}_w \rangle - \frac{\bar{\tau}_w}{\langle \bar{\tau}_w \rangle}}{\langle \bar{\tau}_w \rangle}
\end{align*} \]

\[ \text{Figure 5.4. The dependency of } \epsilon[\langle \bar{\tau}_w \rangle]\% \text{ on } h/\delta \text{ obtained when reference values of } \bar{u} \text{ are used as input to a wall model based on Spalding’s law.} \]

Another possible explanation is that the observed error pattern is a consequence of the choice of the law of the wall. Indeed, the accuracy with which the law approximates the true mean velocity profile varies with \( h/\delta \), and these systematic errors are reflected in the \( \langle \bar{\tau}_w \rangle \) predictions. While this explains the oscillating behaviour of the error at large \( h/\delta \), the deviation of Spalding’s law from the DNS data at the smallest considered \( h/\delta \) is not significant enough to explain the poor performance. Also, a
very similar error pattern was observed when Reichardt’s law (2.6) was employed instead of Spalding’s, see Paper IV.

Two valid explanations for why sampling the velocity signal from a point close to the wall is sub-optimal are now provided. One is based on the physical properties of this signal. In particular, it is noted that \( \frac{\overline{u_{\text{rms}}}}{\langle u \rangle} \), which measures the strength of the fluctuations of \( \bar{u} \) relative to its mean value, grows rapidly towards the wall. This means that as \( h/\delta \) becomes smaller the predictions of \( \tau_\text{w} \) made by the wall model are to an increasingly larger extent determined by the fluctuations. Since the law of the wall is only valid for the mean, this leads to a deterioration in the model’s accuracy. A way to mitigate this error is to apply time averaging to the sampled \( \bar{u} \).

The second source of error is that, in the employed numerical framework, \( \bar{u} \) represents the average value of \( u \) across the WMLES cells. The law of the wall, on the other hand, expects the input to be the (mean) pointwise value of \( u \) at \( x_2 = h \). The cell-averaged and pointwise values agree well at larger \( h \), but in the cell adjacent to the wall this is not the case. This is clearly illustrated in Figure 5.5, showing the outer-scaled mean velocity profile for channel flow at \( \text{Re}_\tau = 1000 \) taken from DNS data [45] and the averaged values of \( \langle u \rangle/U_\delta \) across hypothetical WMLES cells of size \( \delta/20 \) in the wall-normal direction.

![Figure 5.5. The outer-scaled mean velocity profile of channel flow at Re_\( \tau \) = 1000, DNS data [45]. Dashed lines show the boundaries of hypothetical WMLES cells, taken to be of size \( \delta/20 \). Dotted lines show the location of the centres of the cells. Red horizontal lines show the average values of \( \langle u \rangle/U_\delta \) across the corresponding WMLES cells.](image)

The error contributions discussed above add on top of any error associated with the values of \( \bar{u} \). To account for the influence of the latter, a systematic study involving WMLES of the same flow but using different
values of $h$ is necessary. Such a study is presented in Paper VI. In particular, turbulent channel flow at $\text{Re}_r = 5200$ is simulated and an algebraic wall model based on Spalding’s law is employed. The domain is meshed with cubic cells, and different densities of the mesh are considered. To define the density, the number of cells used to discretise the channel half-height, $n/\delta$, is used. Further, two different choices of the interpolation scheme for the convective fluxes discussed in Section 3.4 are considered: linear and LUST. This study has been extended further to include different choices for the SGS model and a much wider range of choices for $h/\delta$. These new results, subject to publication in [80] in the near future, are presented below.

![Graph](image)

**Figure 5.6.** The relative error in $\langle \tau_w \rangle$ as a function of $h/\delta$. Blue lines show results obtained with the linear scheme, and the red lines those obtained with the LUST scheme.

The dependency of the error in $\langle \tau_w \rangle$ on $h/\delta$, obtained for all the considered combinations of modelling parameters, is presented in Figure 5.6. A general trend is that the accuracy of the prediction increases up to $h/\delta \approx 0.2$. Analysis reveals a clear correlation between the error in $\langle \tau_w \rangle$ and the corresponding error in $\langle \bar{u} \rangle$ at the sampling point. This supports the conclusion of Kawai and Larsson [38] that a larger value for $h$ is preferable. However, the data shows that the particular choice of the wall-adjacent cell is not necessarily always the worst. When the Smagorinsky model is used together with the linear scheme, the lowest accuracy is obtained when sampling from the second consecutive off-the-wall cell instead. This implies that $h$ should be determined in connection with the other parameters of the WMLES.

In conclusion, the identified sources of error in algebraic wall-stress modelling and their dependency on $h/\delta$ are summarised.
1. Error due to inaccuracy in the sampled values of the filtered velocity. The size of the error is determined mainly by the setup parameters of the WMLES, such as the density of the grid, the SGS model, and the employed numerical schemes. Generally, the velocity signal is more accurate at $h/\delta > 0.1$

2. The error due to the inaccuracy of the law of the wall with respect to the true mean velocity profile. This error can be very large if the state of the TBL is not consistent with the law employed. When the law is generally accurate, small variations in the accuracy of $\langle \tau_w \rangle$ predictions with $h/\delta$ can be expected.

3. The error caused by the difference between $u_\|_h$ and the corresponding value of the unfiltered velocity at the same point. This error is only significant when $h$ is set to the distance to the centre of the wall-adjacent cell.

4. The error due to the prevalence of the fluctuating component in the sampled velocity signal. The size of the error increases non-linearly towards the wall, with rapid growth for $h/\delta < 0.05$.

5.5 Influence of Other Simulation Parameters

Above, it was shown that the way the error in $\langle \tau_w \rangle$ changes with $h/\delta$ depends significantly on the employed SGS model, numerical schemes, and density of the computational grid. The influence of these modelling parameters on both $\langle \tau_w \rangle$ and the mean velocity profile is further explored here. A small part of the results presented below is included in Paper VI. The remainder has not been previously published and are to be included in a manuscript that is currently in preparation [80].

As in the study of the influence of $h/\delta$, the flow case used to investigate the effect of the modelling parameters is turbulent channel flow at $Re_x = 5200$. This flow is cheap to compute, allowing to conduct a systematic simulation campaign. Also, the errors in the quantities of interest can be calculated precisely based on available DNS data [45]. Another benefit is that, given accurate input from the LES solution, an algebraic wall-stress model can be expected to perform well, resulting in a minimal systematic error in $\tau_w$ due to wall modelling. The particular choice of the law of the wall is not important for this study. Here, Spalding’s law is employed.

To mesh the domain, cubic cells are used and the density of the mesh is thus again defined as $n/\delta$. In the literature, the recommended value for this quantity ranges from 10 to 30 [11, 95, 43]. Several values within this interval are tested here.

For SGS modelling, the three choices based on the Boussinesq approximation discussed in Section 3.3.2 are investigated, that is, the Smagorinsky model, the WALE model, the dynamic one-equation model. This is
motivated by the widespread use of these approaches, and their availability in OpenFOAM.

As discussed in Section 3.4, for the diffusive and pressure gradient terms the linear scheme can be used to get an accurate approximation of the values of the relevant quantities at the cell faces. The treatment of convective fluxes, on the other hand, represents a challenge. Here, the linear scheme and LUST schemes are considered for this purpose.

The observed trends in the influence of the considered simulation parameters on the predictive accuracy are now summarised. As indicated above, the analysis is confined to the errors in two quantities: the mean value of the wall shear stress, $\langle \tau_w \rangle$, and the mean streamwise velocity profile, $\langle u \rangle$. For an analysis of second-order moments of velocity, the reader is referred to Paper VI.

Figure 5.7 shows the relative error in $\langle \tau_w \rangle$ as a function of $n/\delta$. Blue lines show the results obtained with the linear scheme, and the red lines those obtained with the LUST scheme. Results for all three considered SGS models are presented. Additionally, for the WALE and Smagorinsky models, results obtained using two different values of $h/\delta$, 0.125 and 0.25, are reported. A clearly observed outcome is that using the LUST scheme generally results in a lower magnitude of the error. In particular, for $n/\delta > 20$ the observed error is less than 3%. The spread in the error size across different SGS models is also smaller when LUST is used, meaning it is more robust. These observations imply that the dissipation that the LUST scheme introduces into the simulation improves the predictive accuracy. The positive effect of dissipation is further supported by the fact that the best results using the linear scheme are achieved using the Smagorinsky model, which is the most dissipative SGS model of the three considered.

Figure 5.8 shows the corresponding results for the error in $\langle \tilde{u} \rangle$. Each error value is computed as the largest pointwise error observed in the respective $\langle \tilde{u} \rangle$ profile in the region $y/\delta > 0.2$. Closer to the wall, the accuracy of $\langle \tilde{u} \rangle$ is generally lower. The figure reveals that employing the LUST scheme leads to smaller errors, better consistency with respect to the choice of the SGS model, and a decrease in the magnitude of the errors as the mesh gets refined. On the other hand, the only advantage of the linear scheme is that it performs better on the coarsest grids.

Based on the above results, using the LUST scheme together with a dense grid, e.g. $n/\delta = 30$, can be recommended as best-practice guidelines for WMLES. Regarding SGS modelling, none of the considered options performed significantly better than the other. This implies that using the most computationally expensive approach, which is the dynamic one-equation model, cannot be motivated. Between WALE and Smagorinsky, the former can, perhaps, be considered preferable since it was shown to work well in WMLES of other flow configurations, see e.g. [101]. The
Figure 5.7. The relative error in $\langle \tau_w \rangle$ as a function of $n/\delta$. Blue lines show results obtained with the linear scheme, and the red lines those obtained with the LUST scheme.

Figure 5.8. The maximum relative error in $\langle \bar{u} \rangle$ as a function of $n/\delta$. Only velocity values at $y/\delta > 0.2$ are considered. Blue lines show results obtained with the linear scheme, and the red lines those obtained with the LUST scheme.
above recommendations should be complemented with one for the value of $h$. Results shown in Figure 5.6 above indicate that, for the considered combination of modelling parameters, good accuracy for $\langle \tau_w \rangle$ is achieved as long as velocity is not sampled from the wall-adjacent cell. Since using a smaller $h$ can improve the prediction of the fluctuations of $\tau_w$, sampling from the second consecutive off-the-wall cell represents a good compromise. The complete set of recommended simulations parameters is thus: $n/\delta = 30$, the WALE model for SGS modelling, the LUST scheme for convective flux interpolation, and $h$ set to the distance to the second consecutive off-the-wall cell.

An important consideration is whether the guidelines derived here based on simulations of channel flow are applicable to WMLES of other flows. In Papers III and VI, it is demonstrated to be the case for, respectively, WMLES of a ZPG-TBL and of the flow over a backward-facing step. Considering more flow configurations is a direction of future work.

5.6 Simulation on Unstructured Grids

The absolute majority of studies using LES, both wall-resolved and wall-modeled, employ structured hexahedral grids. This ensures absence or minimal influence of the errors associated with non-orthogonality and skewness of the cells. It also allows to use unbounded numerical interpolation schemes for convective fluxes, such as the linear scheme, without the simulation diverging due to the introduced numerical oscillations. However, when complex-geometry flows are considered, which is typical of industrial applications, creating a structured hexahedral grid is either extremely time-consuming or altogether impossible. Therefore, studying the accuracy of WMLES on unstructured grids and using a mildly diffusive interpolation scheme for the convective fluxes is highly relevant. In the previous section, it was shown that a good candidate for the interpolation scheme is LUST.

In Paper III, an unstructured meshing methodology for WMLES is proposed. The aim is that the resulting quality of the mesh in the TBL region is high, and that its density remains constant with respect to the local boundary layer thickness $\delta$. The proposed procedure consists of the following four steps.

1. An estimate for the distribution of $\delta$ should be obtained. For simple flows, analytical estimates exist, see e.g. [79] for the case of a ZPG-TBL. For more complicated flows, a precursor RANS simulation can be conducted to obtain the estimate instead.

2. The surface of the wall is discretised with an unstructured mesh in such a way that the average distance between the face centres remains a constant fraction of the estimated $\delta$. 

61
Figure 5.9. The two unstructured meshes used in the simulations: paved (left) and polyhedral (right). The surface mesh, the extruded layers, and the coarse mesh above the TBL are visualised. For the paved mesh, the AMI is also shown.

3. The generated surface mesh is extruded in the wall-normal direction. Several cell layers are built, with the upper boundary of the extrusion coinciding with the local value of $\delta$ and the thickness of each layer equal to the average distance between the faces of the surface mesh at this location. This step concludes the meshing of the TBL.

4. The part of the domain away from the walls is meshed.

In Paper III, this procedure was applied to generating meshes for WMLES of a ZPG-TBL. Two unstructured surface meshes were constructed, one quad-dominant (paved) and one consisting of polygons, see Figure 5.9. The local average distance between the face-centres was set to $\approx 0.064\delta$. This corresponds to $n/\delta \approx 15.5$ on a structured cubic grid. A higher value was not chosen in order to keep the cost of the simulation in line with the available computational resources. The extruded mesh-layers are also shown in Figure 5.9 as well as a small part of the mesh in the free-stream region. In the case of the paved mesh, the latter consists of structured hexahedrals, which are coupled to the TBL mesh via an arbitrary mesh interface (AMI). For the polyhedral mesh, polyhedra could be used in the free-stream region as well.

A WMLES using both meshes has been performed, using an algebraic wall model based on Spalding’s law (2.5). Furthermore, two values for $h$ were considered: the distance to the centre of the wall-adjacent cell and to that of the third consecutive off-the-wall cell. Note that, as a consequence of the employed meshing strategy, $h$ is always a constant fraction of $\delta$.

Figure 5.10 exhibits the distributions of the skin friction coefficient, $c_f$, obtained in the four WMLES. A power-law estimate [79] as well as DNS data [83] are used as reference. It is observed that sampling from the wall-adjacent cell leads to an under-prediction of the skin friction, in line with the discussion in Section 5.4. Both simulations employing sampling
from the third consecutive off-the-wall cell produce accurate results, only slightly under-predicting the DNS data.

More flow quantities are analysed in Paper III, and the overall conclusion from the study is that the proposed meshing strategy leads to accurate results when the modelling parameters of the WMLES are chosen according to the guidelines given in Section 5.5. Both generated meshes, paved and polyhedral, led to a similar level of predictive accuracy. The polyhedral mesh gave slightly better results for $c_f$, but for $\langle \bar{u} \rangle$, a somewhat better agreement is achieved with the paved mesh. In [49], the same mesh generation strategy was applied to construct a polyhedral mesh for WMLES of the flow around a bulk carrier’s hull in model scale, demonstrating the applicability of the proposed approach to meshing complex-geometry domains. Application to other industrially relevant flows is a promising direction for future work.

5.7 Integrated Formulation of Algebraic Wall Models

The chapter is concluded by two sections describing novel ways of improving the accuracy of algebraic wall-stress modelling proposed in Paper IV. The first proposal aims at reducing the sensitivity of the models’ accuracy on $h/\delta$. Recall that a significant contribution to this dependency is the discrepancy between the (correct) value of $\langle \bar{u} \rangle$ at the sampling point and the corresponding pointwise value $\langle u \rangle$, see Figure 5.5. Let $h_1$ and $h_2$ define the wall-normal bounds of the cell used for sampling the velocity
signal. The centre of the cell is then located at \( h = (h_1 + h_2)/2 \). The law of the wall provides an expression connecting \( \langle \tau_w \rangle \) and \( \langle u \rangle \). To mitigate the error, a relationship between \( \langle \tau_w \rangle \) and

\[
\langle \bar{u} \rangle \approx \frac{1}{h_2 - h_1} \int_{h_1}^{h_2} \langle u \rangle \, dx_2
\]

is sought instead. Assume that the law of the wall is expressed as \( \langle u \rangle^+ = F(\langle \tau_w \rangle, x_2^+, q) \), where \( q \) are model constants. Both the log-law (2.4) and Reichardt’s law (2.6) fall into this class. Additionally, assume that \( F(\langle \tau_w \rangle, x_2^+, q) \) is integrable with respect to \( x_2 \). Then, the relationship between \( \langle \bar{u} \rangle \) and \( \langle \tau_w \rangle \) can be directly obtained as:

\[
\langle \bar{u} \rangle = \frac{\langle u_\tau \rangle}{h_2 - h_1} \int_{h_1}^{h_2} F(\langle \tau_w \rangle, x_2^+, q) \, dx_2.
\]  

(5.14)

The above is referred to as the integrated formulation of an algebraic wall model based on \( F(\langle \tau_w \rangle, x_2^+, q) \). This approach has been introduced in prior work [105, 101], but formulating it for a sampling cell located at an arbitrary distance \( h \) from the wall represents a novelty. A further generalisation of the integrated formulation can be achieved by incorporating integration over multiple consecutive off-the-wall cells. Then, \( h_1 \) is set to the lower bound of the lower-most cell and \( h_2 \) to the upper bound of the upper-most cell. The value of \( \bar{u} \) is computed as the wall-normal average of the WLES velocity across all the considered cells.

In Papers IV and VI, using the integrated formulation of the model based on Reichardt’s law is shown to reduce the error in \( \langle \bar{\tau}_w \rangle \) in the case when \( \bar{u} \) is sampled from the wall-adjacent cell. When sampling from a single cell at a higher \( h/\delta \), using the integrated formulation has little effect on the results. Integrating over multiple cells has only been tested in an a-priori setting. It is demonstrated that this leads to smoothing out of the errors due to the inaccuracies of the law at specific \( h/\delta \).

### 5.8 Dynamically Adjusting Wall Model Constants

The second method introduced in Paper IV aims at improving the accuracy of the models’ predictions of the fluctuating component of \( \bar{\tau}_w \). To that end, an algorithm for dynamically adjusting the constants of the wall model, e.g. \( \kappa \) and \( B \) in the case of Spalding’s law, is proposed. In this setting, the law of the wall is used as an ansatz for the form of the instantaneous filtered velocity profile, and the associated constants are instead treated as unknown coefficients that can, in principle, assume any value. To determine their value at each time step, a fit to the velocity values sampled from several consecutive off-the-wall cells is performed.
The procedure to determine $\tau_w$ can be summarised in the following steps.

1. Velocity values $\tilde{u}_i$, $i = 1 \ldots n$ are sampled from $n$ consecutive cells with cell centres at distances $h_i/\delta$ away from the wall.

2. Given an initial guess for the filtered friction velocity, $\tilde{u}_w^*$, the inner-scaled quantities $\tilde{u}_i^* = \tilde{u}_i/\tilde{u}_w^*$, $h_i^* = h_i\tilde{u}_w^*/\nu$ are computed. A good candidate for the initial guess is the value of $\tilde{u}_w$ at the previous time step.

3. The values of the model coefficients are determined by performing a least-squares fit of the employed law of the wall to the sampled data, $\tilde{u}_i^*$ and $h_i^*$.

4. Given the values of the coefficients, the law is used to determine $\tau_w$ in the same way as in a classical algebraic wall model. To that end, any of the sampled velocity values can be used.

In Paper IV, it is demonstrated that given reference velocity signals $\tilde{u}_i$, the above algorithm produces values of $\tilde{u}_w$, which are highly correlated with reference data. The correlation coefficient is $\approx 0.91$ and it is independent of which $u_i$ is used to predict $\tau_w$ at step 4 of the algorithm.

Recall that, using a standard algebraic wall model, the highest correlation coefficient that was obtained was $\approx 0.62$, using the velocity signal from a distance close to the wall, $h/\delta = 0.025$, see Figure 5.3. Implementation of the proposed method for use in WMLES and further investigation of its predictive accuracy is a direction of future work.
6. Developed Software

This chapter discusses three software solutions, all of which were developed in the course of the work on this thesis to facilitate the scientific investigations described in the chapters above. One is a Python package for inflow generation methods, used here in Paper I. Another is a Python package for post-processing simulation results, employed in Papers I, III, and VI. The third software is an OpenFOAM-based C++ library implementing algebraic and ODE-based wall-stress modelling approaches. It is used in the simulations presented in Papers III and VI. All three codes are made available to the public under an open-source licence to facilitate their use by other research groups and CFD engineers. The following sections briefly describe the functionality and design of each software.

6.1 A Python Package for Inflow Generation

A large variety of methods for inflow generation have been proposed in the literature, and a review of those based on using a precursor simulation has been given in Chapter 4. Naturally, it is important to assess which of the proposed approaches performs best. Several works have investigated this [39, 99, 76, 17, 36], but each study typically considered a relatively small number of methods. One reason for this is that the implementation of the methods is usually done within the framework of a specific CFD solver. This makes the task of testing a large set of approaches tedious, since it involves a large amount of programming. Therefore, it would be highly beneficial to develop a stand-alone inflow generation library, that would allow each implemented method to be used with most CFD software solutions.

The key question is how the inflow generated by the library is to be incorporated into the simulation by a given solver. The most obvious approach is to save the inflow data to disk. Indeed, most modern CFD solvers support reading boundary data from one or multiple files. The task of making all the library’s inflow generation methods available in such a solver would thus boil down to implementing the ability to save the fields in a file format that the solver supports.

Naturally, saving the inflow data represents an overhead, both in terms of storage and the runtime of the solver because reading from a disk is a slow operation. However, storage is becoming increasingly inexpensive,
and a significant part of the runtime overhead can be mitigated using file formats that allow the solver to read in the data in parallel.

Another downside to the proposed framework is that it is impossible to implement methods that somehow interact with the flow field inside the computational domain. This includes approaches based on weak recycling methods, such as the LWS method [54]. Another example is the method developed in [97] which relies on introducing extra body forces into the momentum equation. Nevertheless, the number of approaches that can potentially be included is vast.

An implementation of a library for inflow generation based on the design principle discussed above is presented in Paper II. The library is distributed in the form of a Python package called eddylicious. A simple code structure is adopted, with the functionality of the package divided into three modules: readers, writers, and generators. As the names suggest, the former two contain functions for input/output (I/O) of inflow data, and the latter includes the inflow generation methods.

Currently, the package includes the type of functionality that was necessary for conducting the studies on inflow generation presented in Paper I. The existing I/O routines allow eddylicious to read fields sampled in an OpenFOAM simulation (a precursor) and write the generated fields in file formats that OpenFOAM solvers can read in. The rescaling procedure defined by the LWS method [54] is implemented as a generator. As mentioned, weak recycling methods cannot be directly incorporated into eddylicious because the fields are generated prior to the execution of the simulation. However, the rescaling can be considered a general way of transforming a given database of velocity fields sampled from a precursor TBL or channel flow simulation to match certain desired integral characteristics. Another included generator simply interpolates sampled precursor data onto the inlet grid. Additionally, several auxiliary scripts are present, e.g. for converting saved data from one file format to another, computing the time-averaged statistics of the generated inflow, etc.

A significant effort has been put into providing detailed documentation for all the available functionality and its implementation.\footnote{See http://eddylicious.readthedocs.io.} This includes an extensive user guide, detailed doc-strings for every function, and a tutorial that can be used to become familiar with using the package.

Further development of eddylicious is envisioned to be a collaborative effort. Individuals with experience in a certain generation method or using a particular solver are most welcome to contribute to the package with associated developments.
6.2 A Python Package for Post-Processing of CFD Simulations

To analyse the results of a simulation, it is necessary to post-process it, i.e. extract and visualise the relevant data. A large variety of tools providing associated functionality exist, including specialised software. The primary focus of the existing solutions is providing an interactive environment for visualising and manipulating complex three-dimensional data. In Paper V, a complementary post-processing tool is introduced, focusing instead on the following features:

- The possibility to quickly reproduce the exact same post-processing for a large number of cases. This is crucial when results from a large simulation campaign are considered. Having to reapply the same analysis manually is tedious and error prone.
- The functionality to produce customised publication-quality plots. This is necessary when the produced visualisations are intended to be included in a published work, which is common in an academic setting.
- Visualisation of two- and one-dimensional data. While the majority of the CFD simulations are three-dimensional, the data that are visualised in figures are predominantly two- or one-dimensional, commonly extracted as cut-planes and profiles along a line.

The introduced tool is a Python package called turbulucid. It combines data manipulation functionality from the library VTK [85] and the plotting routines available in the Python package matplotlib [31] to provide a set of classes and functions for post-processing two-dimensional CFD datasets. Using them, the analysis can be scripted in Python, making it easy to examine a large set of simulations in a structured way or quickly apply an existing analysis to a new case. Additionally, users have the possibility to utilise any of the numerous packages available in the Python ecosystem in the post-processing.

An important feature is that no assumption is made regarding the topology of the computational mesh the data resides on, meaning that any mesh consisting of polygonal cells can be used. The data are assumed to be stored in a format for unstructured meshes defined by VTK, but the package is designed to be easily extendible to read data in other formats (see Paper V for a discussion of the associated API). Scalar, vector, and tensorial quantities are supported.

For a complete list of provided plotting and data extraction routines and details on how the package should be used, the reader is referred to Paper V and to the documentation available online.² Apart from this thesis, examples of applying turbulucid to post-processing of simulations of various flows can be found in [79, 48, 50]. In Paper VI, the package is

²https://timofeymukha.github.io/turbulucid/
used for post-processing simulations of flow over a backward-facing step. Two figures produced in the course of analysing the behaviour of this flow are used here to demonstrate some of the functionality of turbulucid.

In Figure 6.2, the distributions of the mean streamwise flow velocity \( \langle u \rangle \) and turbulent kinetic energy, \( k = \langle u'_i u'_i \rangle / 2 \), are shown. The quantities are normalised with the velocity of the free stream, \( U_0 \), and the coordinate axes with the height of the step, \( H \). To produce these plots, the function plot_field() provided by turbulucid was used. The white lines in the figure show the profiles of the same quantities at selected streamwise locations. To extract the values of the unknowns along vertical lines, the function profile_along_line() is used, and to produce the line plots, the matplotlib plot() function is employed. Finally, to show the zero levels of the profiles, the matplotlib function vlines() is used. This example shows how plotting and data extraction routines available in turbulucid can be used in tandem with the existing functionality of matplotlib.

\[
\langle \bar{u} \rangle / U_0 \quad \text{and} \quad 20k / U_0^2
\]

Figure 6.1. The distribution of \( \langle \bar{u} \rangle / U_0 \) (top) and \( 20k / U_0^2 \) (bottom). Profiles of these quantities at selected streamwise locations are shown with white lines. Dashed white lines are used to indicate the zero levels of the profiles.

Figure 6.2 demonstrates the capability of turbulucid at producing vector plots. In the figure, the arrows show the direction of the instantaneous velocity field in a cut-plane perpendicular to the mean flow direction and located shortly downstream of the step. To produce the vector plot, the function plot_vectors() is used.

In conclusion, it is noted that, although turbulucid is tailored to the specific needs of the research community, it can also be useful to CFD engineers when there is a need to produce a publication-quality plot or perform an easily reproducible scripted analysis of a simulation campaign.
Figure 6.2. The distribution of normalised pressure values across a cut-plane perpendicular to the mean streamwise direction of the flow, located at a distance $H$ downstream of the step. Arrows show the direction of the instantaneous filtered velocity field.

6.3 A Library for WMLES based on OpenFOAM Technology

To conduct the WMLES reported in this thesis, it was necessary to implement the considered wall-stress modelling approaches. To that end, an OpenFOAM-based C++ library was developed. A full description of this software, first introduced in [62], can be found in Paper VI. Here, a brief summary of its features and design is given.

The code of the library is tightly integrated into the class structure defined by OpenFOAM. Recall that, here, wall-stress models enforce the correct value of $\tau_w$ by adding an appropriate amount of subgrid viscosity $\nu_{sgs}$ at the wall (see Section 5.1). Accordingly, the library represents each model as a Dirichlet-type boundary condition class for this field. This leads to their implementation being segregated from that of the OpenFOAM’s incompressible LES solvers of OpenFOAM, implying that any such solver can be enhanced with wall modelling without modifying its source code.

The included wall-stress models are of two types: algebraic and ODE-based. In particular, algebraic models based on Spalding’s [96], Reichardt’s [77], and Werner and Wengle’s [105] laws of the wall are present. The integrated formulations of the models based on the latter two laws are also included. For the ODE-based models, two treatments of the source term $F$ are implemented: $F = 0$ and $F_i = \partial p/\partial x_i|_h$. Furthermore, two options are available for the choice of the associated $\nu_t$ model: based on equations (5.11) and (5.12), respectively.

An important property of the library is that the user is given the possibility to adjust every parameter governing the wall modelling. Most
importantly, an arbitrary choice of $h$ is allowed, and an individual $h$-value can be assigned to each wall face. Other modifiable parameters include the model constants associated with each wall-stress modelling approach, the convergence criteria for iterative solvers, the density of the one-dimensional grid used in ODE-based wall models, and others. The desired settings are specified by the user in a configuration dictionary located in the file providing initial and boundary conditions for the $\nu_{sgs}$ field.

The implementation of the above functionality is distributed across several class hierarchies (see Paper VI for a detailed description). This modular structure allows to easily introduce new developments for all the aspects of wall modelling, typically by creating a new derived class in the associated hierarchy and providing an implementation for one or two abstract virtual functions.

To the best of the author’s knowledge, the described library is the most feature-rich open-source software for wall-stress modelling currently available. Apart from Papers III and VI of this thesis, it has been used in several other works [63, 49, 80]. The fact that the provided functionality can be used with general-purpose LES solvers is expected to contribute to a wider adoption of WMLES for simulation of industrially relevant flows.
7. Conclusions

This focus of this thesis is the development of modelling techniques for LES of wall-bounded flows, with the goal of making such simulations more accurate and efficient. The two particular topics that are considered are generation of turbulent inflow and wall-stress modelling.

The main contribution to the former is the precursor-based inflow generation method introduced in Paper I. This method is robust, easy to implement, and its accuracy is on par with other approaches proposed in the literature. In the future, it would be interesting to explore possibilities for reducing the cost of the precursor simulation. Using wall modelling to that end appears to be an attractive alternative.

Concerning WMLES, this thesis both introduces novel wall-stress modelling approaches and comprehensively evaluates the predictive accuracy of methods that are currently in widespread use. How the accuracy of WMLES depends on simulation parameters other than the choice of the wall model is also extensively investigated, in Paper VI. As a result, best-practice guidelines for the choices of the parameters are developed. In particular, it is shown that using a mildly dissipative numerical scheme for convective flux interpolation, a grid with density $n/\delta = 30$, the WALE model for SGS modelling, and sampling from the second consecutive off-wall cell leads to excellent accuracy in the predictions of $\langle \tau_w \rangle$ and $\langle \bar{u} \rangle$ (in the outer layer). In Paper III, it is demonstrated that these guidelines are applicable to simulations on unstructured meshes, and a mesh construction strategy suitable for WMLES of complex-geometry flows is proposed.

The new wall-stress models are presented in Paper IV. One development is the integrated formulation of algebraic wall models, formulated for an arbitrary choice of $h$. It is shown that the integrated formulation allows the removal of a significant contribution to the error in $\langle \tau_w \rangle$ when the wall model input is sampled from the wall-adjacent cell. The approach is further generalised to sampling from multiple consecutive cells, leading to a stronger reduction in the variation of the error with $h$. The second proposed method also employs multi-cell sampling. Here, the novelty lies in considering the model constants of the law of the wall to instead be dynamically adjustable coefficients that can be obtained using a least-squares fit to the data sampled from the cells. In an a-priori study, this is demonstrated to lead to a significant improvement in the predictions of the fluctuations of $\tau_w$. Further evaluation of the benefits offered by these
new developments using systematic WMLES of canonical turbulent flows is a direction of future work.

Another contribution of this work is the open-source software that is described in Papers II, V, and VI. All three codes provide functionality that make it significantly easier to reproduce the simulation results presented in this thesis. In particular, the Python package eddylicious, presented in Paper II, provides routines for working with inflow data generated by precursor OpenFOAM simulations. It also defines a convenient framework for implementing other inflow generation methods, aiming to contribute to their dissemination among users of different CFD codes. Another Python package, presented in Paper V and called turbulucid, can be used for post-processing unstructured two-dimensional datasets, in particular, for producing publication-quality visualisations. Finally, the software presented in Paper VI is a comprehensive library of algebraic and ODE-based wall-stress models that can be used with the LES solvers provided by OpenFOAM. All three software solutions can be further developed to include more functionality. To that end, third-party contributions are welcomed.
8. Summary of Papers

**Paper I** This paper addresses the problem of generating an inflow TBL and introduces a robust and easy to implement method for doing that based on a precursor channel flow simulation (see Section 4.2 above). For validation, LES of a ZPG-TBL at $Re_	heta = 830-2400$ is performed and the accuracy of the proposed approach is compared to using a rescaling procedure defined in [54] as well as not employing an inflow generation procedure (i.e. including the transition of the TBL in the simulation). Analysis of the skin friction coefficient, the mean velocity profiles, and the Re-stress tensor reveals that using channel flow leads to a level of accuracy similar to the benchmark methods.

**Contributions:** The author of this thesis performed and post-processed all the simulations discussed in the paper. The manuscript was written in collaboration with the second author.

**Paper II** The Python package *eddylicious* for inflow generation methods described in Section 6.1 is presented here. In particular, the functionality of the package, its architecture, and potential use cases are considered.

**Contributions:** The author of this thesis designed and implemented the Python package discussed in the paper, wrote its documentation, and tested its functionality. The manuscript was written in collaboration with the second author.

**Paper III** This work demonstrates that the WMLES methodology developed in this thesis is applicable to simulations on unstructured meshes. An unstructured meshing strategy for WMLES, which is discussed in Section 5.6, is proposed and validated using WMLES of a ZPG-TBL. A paved and polyhedral mesh is considered. For both, good agreement with the reference data is observed for the skin friction coefficient, mean streamwise velocity, and the Reynolds stresses. The results from the above WMLES simulations are made available to the public.

**Contributions:** The author of this thesis implemented the wall modelling functionality, which is employed in the paper. He also performed the post-processing of the simulation results and wrote the manuscript, the latter in collaboration with the third author.

**Paper IV** This paper explores the relationship between $\tau_w$ and $\tau_{w}$ for turbulent channel flow at $Re_\tau = 1000$, which was discussed here
in Section 5.2. An a-priori analysis of the performance of algebraic models based on Spalding’s and Reichardt’s laws of the wall is also given. The dependency of the accuracy on $h/\delta$, $n/\delta$, and the model constants is reported. The integrated formulation of an algebraic model based on a law of the wall is defined and generalised to incorporate the LES solution at multiple sampling points (see Section 5.7). The integrated formulation is shown to lead to a reduced dependency of the error in $\langle \tau_w \rangle$ on $h/\delta$. Finally, an idea is described to dynamically adjust wall model constants to increase the accuracy of the predicted fluctuations of $\tau_w$ (see Section 5.8). The employed datasets for a-priori study of WMLES are made publicly available.

**Contributions:** The author of this thesis developed and tested the integrated formulation of algebraic wall models. He also participated in writing the manuscript as well as setting up the simulation cases used to produce the reference $\bar{u}$ and $\tau_w$ values.

**Paper V** This article introduces turbulucid, a Python package for post-processing two-dimensional datasets coming from CFD simulations (see Section 6.2). The functionality of the package is summarised and exemplified using a dataset from an LES of the flow over a backward-facing step.

**Contributions:** The author is the sole contributor to this work.

**Paper VI** This work presents the OpenFOAM-based library for WMLES, discussed here in Section 6.3. To put the implemented features into context, an overview of existing wall-stress modelling approaches, similar to that in Section 5.3, is also given. This is followed by a complete description of the library’s functionality and design. The developed software is applied to WMLES of turbulent channel flow and the flow over a backward-facing step. For both flows, simulation campaigns exploring the influence of various modelling parameters on the predictive accuracy are performed. Results from an extension of the channel flow study have been presented in Sections 5.4 and 5.5 and are used to provide best-practice guidelines for the choice of the considered modelling parameters. An important conclusion from the backward-facing step study is that these guidelines remain valid for this more complex flow configuration. In particular, it was possible to achieve very good agreement with the reference experimental data for both $\langle u \rangle$ and $\langle \tau_w \rangle$. The results of all performed WMLES are made available online.

**Contributions:** The author of this thesis is the leading developer of the library with support from the other authors. He also wrote the manuscript in collaboration with the other authors and performed most of the numerical experiments.
9. Sammanfattning på svenska


Matematiskt beskrivs en fluids tillstånd (så som hastighet, tryck och eventuellt andra egenskaper) av Navier-Stokes ekvationer. För inkompressibel strömning har ekvationerna följande form [75],

\[
\begin{align*}
\frac{\partial u_i}{\partial x_i} &= 0, \\
\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} &= -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}.
\end{align*}
\]

(9.1)

(9.2)

Här är \( x_i, i = 1, 2, 3 \), kartesiska koordinater, \( u_i \) motsvarande hastighetskomponenter, \( p \) är trycket och \( \nu \) den kinematiska viscositeten. Einsteins konvention för summering över upprepade index används.

alla flesta lösare som idag används för industriella beräkningar är base-rade på just denna metod. Därmed är det fördelaktigt att utvärdera nya modelleringstekniker i samma numeriska uppsättning. Lösaren som här används för att utöra alla numeriska experiment ingår i OpenFOAM, vilket är ett allmäntillgängligt programpaket för (framtills för allt) simuleringar av strömning [104].

Fysikaliskt kännetecknas turbulent flöde av en sammansättning av ett stort antal virvlar av varierande storlek. De största motsvarar problemets geometriska dimensioner medan de minsta virvorna generellt är flera storleksordningar mindre [75]. Följden blir att numerisk lösning av (9.1)-(9.2) oftast är för dyr, då även de små rörelserna måste representeras i den diskreta formuleringen, det vill säga bli upplöst. Detta har lett till utveckling av så kallade turbulensmodeller. Turbulensmodellering handlar om att utifrån (9.1)-(9.2) ta fram ett annat ekvationssystem där nya ekvationstermer står för (en del av) de turbulenta fluktuationernas effekt på de rörelser som är upplöst. Den turbulensmodell som används här heter ”large-eddy simulation” (LES). I LES strävar man efter att modellera mindre energifulla virvlar medan stora, kraftfulla virvlar lösas fortfarande upp [82].


WMLES och i synnerhet den ovan beskrivna väggmönstrenklassen har med framgång används för att beräkna ett stort antal olika flöden. Trots detta krävs ytterligare utveckling av denna metodik och validering av dess noggrannhet. Förutom väggmönstren spelar valet av andra mode-leringsalternativ, så som den matematiska modellen för de icke upplöst turbulenta fluktuationerna, en stor roll. Det är viktigt att bygga förståelse för hur de olika alternativen påverkar noggrannheten. Detta undersöks i Artiklar III och VI och som resultat tas rekommendationer fram för de valbara parametraerna. I Artikel IV och VI undersöks även väggmönstren i sig och två nya typer av väggmönster föreslås. Ett av förslagen
gör att modellernas noggrannhet blir mindre beroende av valet av $h$. Det andra förslaget är att använda LES-lösningen från mer än en punkt för att kunna beräkna väggskjuvspänningens fluktuationer med högre noggrannhet.


10. Acknowledgements

Firstly, I would like to express my sincere gratitude to my main scientific advisor Mattias Lievenhahl for his continuous support and guidance. My second advisor, Gunilla Kreiss, is also gratefully acknowledged for her active engagement in the project and all the fruitful scientific discussions that we had. A huge thank you goes to my friend, colleague, and partner in crime Saleh Rezaeiravesh. This thesis has benefited enormously from his brilliant scientific ideas.

It would have also been much harder to complete this thesis if not for the friendly, productive and supportive atmosphere at the department where I had the honour of conducting this work. Therefore, I would like to sincerely thank all my colleagues for contributing to this wonderful environment. A special thank you goes to the other doctoral students. So many of you have become my friends, with whom I shared both the fun and the challenging moments of the last five years. Among the senior staff, I would like to express my sincere gratitude to Lina von Sydow and Michael Thuné for their commitment to making the department a better working place, in particular for doctoral students. The administrative staff of the department is also gratefully acknowledged for their helpfulness, patience, and professional skills in fighting all kinds of bureaucrats.

I am also grateful to everyone who has participated in the weekly “German lunches”. Admittedly, my German has improved only moderately but, thanks to Astrid Raidl, all of us have certainly eaten a lot of cake! In relation to feeding, I would also like to thank Jonatan Werpers for introducing me to League of Legends. The glorious times we have had playing together have been a huge help in distracting me from work-related problems. Finally, I would like to thank all of my awesome office-mates and the following particularly amazing individuals: Simon Sticko, Slobodan Milovanovic, and Anastasia Kruchina. The four of us walked the path of the graduate student side by side and emerged victorious.

I would also like to acknowledge the support I have received from several organisations. The Swedish National Infrastructure for Computing has made it possible for me to perform simulations demanding hundreds of computational cores. The Swedish Research Council has provided financial support (Grant No 621-2012-3721). C. F. Liljewalchs stipendie-stiftelse, the Knut and Alice Wallenberg Foundation, and Åforsk have provided travel funds that allowed me to present my research at several international conferences.
In conclusion, I would like to thank my family for all the inspiration, patience, and love they have given me. If not for Carmen, my best friend and loving partner, this thesis would never have materialised. I will never be able to thank her enough for all the support she has given me throughout these years.
Bibliography


DOI: 10.1023/A:1009995426001.

DOI: 10.1063/1.1389286.

DOI: 10.1063/1.2781596.

DOI: 10.1063/1.870414.

DOI: 10.2514/1.J055745.

DOI: 10.1063/1.4861069.

DOI: 10.1016/j.jcp.2015.11.010.


DOI: 10.1016/j.paerosci.2008.06.001.

DOI: 10.1016/S0142-727X(03)00048-1.


DOI: 10.1002/zamm.19510310704.


DOI: 10.1017/S0022112010003113.


A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology. (Prior to January, 2005, the series was published under the title “Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology”.)