ABSTRACT

Title of Dissertation: ADAPTIVITY IN WALL-MODELED
LARGE EDDY SIMULATION

Ali Berk Kahraman
Doctor of Philosophy, 2022

Dissertation Directed by: Professor Johan Larsson
Department of Mechanical Engineering

In turbulence-resolving simulations, smaller eddies account for most of the computational cost. This is especially true for a wall-bounded turbulent flow, where a wall-resolved large eddy simulation might use more than 99% of the computing power to resolve the inner 10% of the boundary layer in realistic flows. The solution is to use an approximate model in the inner 10% of the boundary layer where the turbulence is expected to exhibit universal behavior, a technique generally called wall-modeled large eddy simulation.

Wall-modeled large-eddy simulation introduces a modeling interface (or exchange location) separating the wall-modeled layer from the rest of the domain. The current state-of-the-art is to rely on user expertise when choosing where to place this modeling interface, whether this choice is tied to the grid or not. This dissertation presents three post-processing algorithms that determine the exchange location systematically.

Two algorithms are physics-based, derived based on known attributes of the turbulence in attached boundary layers. These algorithms are assessed on a range of flows, including flat

plate boundary layers, the NASA wall-mounted hump, and different shock/boundary-layer interactions. These algorithms in general agree with what an experienced user would suggest, with thinner wall-modeled layers in nonequilibrium flow regions and thicker wall-modeled layers where the boundary layer is closer to equilibrium, but are completely ignorant to the cost of the simulation they are suggesting.

The third algorithm is based on the sensitivity of the wall-model with the predicted wall shear stress and a model of the subsequent computational cost, finding the exchange location that minimizes a combination of the two. This algorithm is tested both a priori and a posteriori using an equilibrium wall model for the flow over a wall-mounted hump, a boundary layer in an adverse pressure gradient, and a shock/boundary-layer interaction. This third algorithm also produces exchange locations that mostly agree with what an experienced user would suggest, with thinner layers where the wall-model sensitivity is high and thicker layers where this sensitivity is low. This suggests that the algorithm should be useful in simulations of realistic and highly complex geometries.

ADAPTIVITY IN WALL-MODELED LARGE EDDY SIMULATION

by

Ali Berk Kahraman

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Advisory Committee:
Professor Johan Larsson, Chair/Advisor
Professor James Baeder
Professor Christoph Brehm
Professor James Duncan
Professor Arnaud Trouvé

Acknowledgements

I would like to start by thanking my mother, Mahmure Kahraman, who has put so much effort into raising me that she deserves credit with every and any piece of achievement I get.

I would definitely like to thank Prof. Larsson, whose guidance has been nothing but exceptional both in terms of technical and professional development. I am sure his level of technical insight and, more importantly, human understanding will be inspiring me for years.

My labmates have made life much more bearable than it would otherwise have been. I would like to extend my thanks to all of them, Siavash, Abishek, Nikhil, John, Vedant, Raihan, Kamrul, Jon and Walter. Especially the friendship and camaraderie of Nikhil was invaluable during the isolative early days of the COVID-19 pandemic.

I would also like to acknowledge the financial support and computing power I have received. This work was supported by the NASA Transformation Tools and Technologies program (grant 80NSSC18M0148) and the NNSA Predictive Science Academic Alliance program (grant DE-NA0003993). The computing power was provided by the University of Maryland HPC facility Deepthought2, Maryland Advanced Research Computing Center, the Department of Energy INCITE program, and the NNSA PSAAP program.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

DNS	Direct Numerical Simulation
EQWM	Equilibrium wall-model
h_{wm}	Wall-model exchange location
LES	Large eddy simulation
RANS	Reynolds Averaged Navier Stokes Equations Simulation
Re	Reynolds Number
WMLES	Wall-modeled large eddy simulation
WM	Wall-model
WRLES	Wall-resolved large eddy simulation

Chapter 1: Introduction

Fluid motion can be laminar or turbulent, depending on whether the fluid moves smoothly or chaotically. Turbulent flow is encountered in many engineering problems, for example a ship sailing in water, a plane moving in air, or a car engine mixing fuel and air in its cylinders. To improve performance in such applications, it is imperative to understand and be able to accurately model turbulent flows.

Turbulence is characterized by chaotic swirling motions called "eddies". These eddies are introduced by instabilities in the governing equation of the fluid flow, the Navier-Stokes equations. Eddies exist in a broadband spectrum from large and energetic eddies to the smallest and least energetic eddies. Only the smallest eddies experience viscous effects that dissipate the kinetic energy into heat. The smaller the viscosity, the smaller the eddies can become before being overcome by viscous effects. The energy transfer from larger to smaller scales is called the energy cascade [12, 13].

The Navier-Stokes equations describe the motion of all eddies, both small and large, so it is possible to simulate turbulence by solving these equations directly. The kind of simulation that captures and resolves all of the eddies is called direct numerical simulation (DNS), and requires a computational cost that increases very rapidly with the Reynolds number.

In many flows, the smaller eddies appear universal. They do not affect the large flow structures, they simply exist to forward the energy from the larger eddies to the smaller ones in the cascade. This observation suggests that we do not have to resolve all of the eddies, and rather we can model smaller ones according to the expected universal dissipation behavior, cutting the cost of the simulation significantly. This approach is called large eddy simulation (LES), signifying that only the larger, energetic and large-flow-structures-altering eddies are resolved. Note that, in a flow free from wall effects, this means that the computational cost is independent of the Reynolds number.

1.1 Wall Turbulence and Its Effect on LES

In a turbulent boundary layer, the structure of turbulence changes near the wall. Away from the wall, say for $y/\delta \gtrsim 0.1$ where y is the wall distance and δ is either the boundary layer thickness or the channel half-width, the turbulence is free from the constrictive effects of the wall and behaves like free shear flow turbulence. In this region, the length scale of the most energetic eddies is approximately independent of wall distance. Near the wall, say for $y/\delta \lesssim 0.1$, the kinematic damping by the wall restricts the size of the energetic eddies to be approximately proportional to the wall distance.

Note that a wall distance of 0.1δ was used here as the divider between the near-wall ("inner") and farther-from-the-wall ("outer") layers, but different suggestions have appeared in the literature [14, 15]. Pope [12] even defines inner layer in terms of viscous units (to be explained in the next paragraph), however in this work we follow the convention to define it in terms of "outer units", i.e., in terms of δ.

It is worth defining the inner scaling, or viscous scaling, at this point. This scaling is based on the wall shear stress, and is useful when describing the effects of viscosity in the near wall region. The inner (or viscous) length scale is defined as

$$\delta_\nu = \frac{\mu_\mathrm{w}}{u_\tau \rho_\mathrm{w}}, \tag{1.1}$$

with any length y nondimensionalized by viscous units such that

$$y^+ = \frac{y}{\delta_\nu}, \tag{1.2}$$

and any velocity u nondimensionalized by viscous units such that

$$u^+ = \frac{u}{u_\tau}, \tag{1.3}$$

where velocity scale is the friction velocity,

$$u_\tau = \sqrt{\tau_\mathrm{w}/\rho}\,, \tag{1.4}$$

with τ_w being the shear stress exerted by the wall (c.f. Pope [12], Tennekes and Lumley [13]). There is no consensus yet on what the velocity scale is away from the wall, but most evidence suggests that it is still given by u_τ.

With the inner scaling now described, it is possible to talk more deeply about the inner layer. It is divided mainly into 3 parts. Typically, $y^+ \lesssim 5$ marks the viscosity dominated region, with the velocity following the relation

$$u^+ = y^+ \tag{1.5}$$

whereas $5 \lesssim y^+ \lesssim 30$ is a transition region labeled as buffer layer between the viscosity and the inviscid mixing effect. Above $y^+ \approx 30$ and below $y/\delta \approx 0.1$, the eddies are governed by inviscid behavior, but with their length and velocity scales dominated equally by the friction velocity and outer units. This region is typically called the logarithmic layer, because the mathematical solution to the mean flow equations gives a logarithmic velocity profile as

$$u^+ = (1/\kappa)\ln(y^+) + B. \tag{1.6}$$

The interested reader is referred to a book on turbulence, e.g. books by Pope [12] or Tennekes and Lumley [13], to learn more about the derivation of these equations.

The significance of the inner layer for turbulence simulations is that, the sizes of the large and flow-affecting eddies and the smaller dissipating eddies approach each other (cf. Pope [12]). This means that the LES needs to resolve these eddies of dissipating scale, the eddies that would be modeled in a wall-free flow. This procedure is called wall-resolved LES (WRLES) in the literature and it means that the Re dependence of the computational cost is significant, with the relation being estimated as $\mathcal{N} \sim Re_\tau^2$ for a WRLES by Larsson et al. [14] between the friction Reynolds number

$$Re_\tau = \frac{\rho u_\tau \delta}{\mu} \tag{1.7}$$

and the number of grid cells $\mathcal{N}$ for a sample box with size of $\{\delta, \delta, \delta\}$. Furthermore, it should be noted more than 99% of these grid points would be spent resolving the inner layer in most engineering situations, e.g. for an airfoil at relatively high Re [14] or for an airliner fuselage simplified to a flat plate at cruise conditions with $Re_x \sim 10^{10}$ [16]. Modern computers are still not powerful enough to make such computations feasible or even possible, e.g. Spalart et al. [17] estimates that WRLES of an aircraft wing could be possible by 2040, if (and this is a big if) Moore's law of computational power still holds by then.

1.2 Modeling of the Wall Turbulence: Wall Models

Since the existence of a wall makes the WRLES impractical for engineering flows due to small but relevant scales of motion very close to the wall, the question arises: can we model this region, just like we modeled the dissipative scales of turbulence to arrive at LES before. The answer is yes, because there exists some level of universality for both the inner and outer layers of the boundary layer, at least for equilibrium flows. The relations for the inner layer of the boundary layer are universal for different Re with respect to wall shear stress, as introduced in the previous section. If it is also wished that the outer layer is modeled, there exists a universality with the "outer" variables such as the free-stream flow velocity and boundary layer thickness (e.g. Musker [18]). Reynolds Averaged Navier-Stokes Simulation (RANS) methods, industry standard methods that model the effect of all eddies, are also typically well calibrated for attached flows, adding to the attainability of wall-modeling.

There are basically two approaches for this: solving the whole or a part of the boundary layer with a different set of RANS equations and linking it to the LES, or simply not resolving the small eddies in the inner layer and instead including their effects in the calculation of the wall-stress with a so-called wall-model. The first approach is generally called Hybrid RANS-LES, whereas the second has the name wall-stress modeling. Note that both approaches can be deemed wall-modeled LES since they both model the wall, however the community tends to use WMLES explicitly for wall-stress modeling, and this is what we will do in this work.

Hybrid RANS/LES methods work by dividing the solution domain into parts of RANS and LES treated regions. Typically they resolve the whole boundary layer while turning to LES only for extremely large eddies or abrupt flow separation. They can be divided by how they decide between LES and RANS regions, by either describing a single governing equation with changing eddy viscosity by some implicit or grid dependent relation (e.g. methods inspired by Spalart et al. [17]) or by user-decided RANS and LES regions with different governing equations (e.g. Quemere and Sagaut [19]). Because this type of simulations are out of the scope of this work, we will not go into more details on them. However, interested reader is referred to the review paper by Spalart [20] and the book by Sagaut et al. [21].

Wall-stress models work by modifying the shear stress applied by the wall in the fluid flow. In the entirety of the domain the LES equations are valid, except that the wall boundary condition is changed with a shear stress τ_{w} which is computed by inserting flow variables above some wall-model height h_{wm} (which is the main focus of this work) into a wall-model. This procedure is demonstrated by Figure 1.1. There are many options for wall-stress models, with many of them focusing on solving for a velocity profile between the h_{wm} and the wall, e.g. assuming a log-law [22], a generalized velocity profile [23], or a 1-D RANS solution equilibrium wall-model (EQWM) [24] to name a few. This kind of wall-modeling is the main focus of this thesis, and will be talked about in more detail in the upcoming chapter.

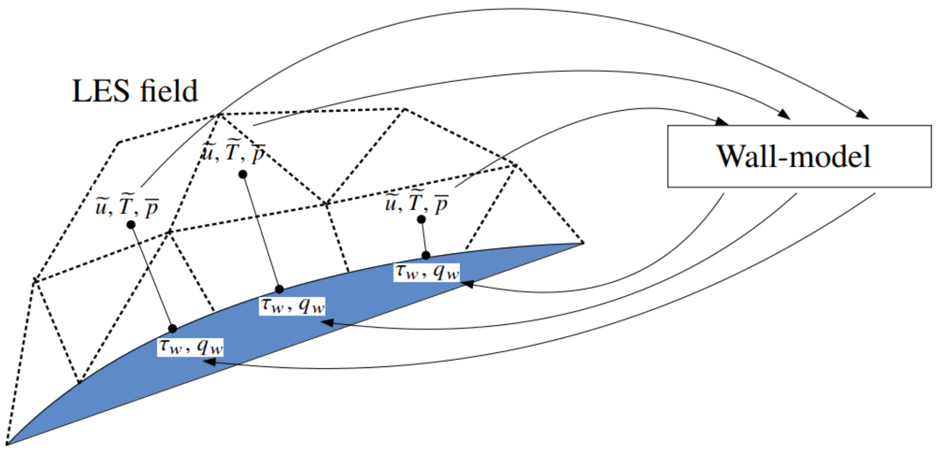
LES field
Wall-model
$\widetilde{u}, \widetilde{T}, \overline{p}$
$\widetilde{u}, \widetilde{T}, \overline{p}$
$\widetilde{u}, \widetilde{T}, \overline{p}$
τ_w, q_w
τ_w, q_w
τ_w, q_w

Figure 1.1: Schematic of the implementation of a wall-model in LES.

1.3 Motivation

All WMLES approaches require, perhaps implicitly, a choice of where to start augmenting the underlying LES method: this might be referred to as the "modeling interface", the "exchange location", or the "top of the wall-modeled layer". We will refer to the quantity h_{wm} as the thickness of this "wall-modeled" layer, and then make the argument that all WMLES approaches necessarily involve a choice for how h_{wm} varies across the wall boundaries. Detached eddy simulation derived approaches make this choice implicitly, based on the computational grid, a model parameter, and possibly the flow solution itself. Some hybrid LES/RANS approaches instead require/allow an explicit specification of h_{wm}; this could be viewed as a weakness (requiring additional user input) or a strength (allowing for convergence by refining the grid while maintaining a fixed interface location, [14]). For wall-stress-models, the nature of the choice for h_{wm} comes down to the implementation in the flow solver: if the "exchange location" is hard-coded to a specific grid point (e.g., the second [25], third [26] or fifth [27]) or location in a high-order element (e.g., the bottom of the second element [28]) the choice of h_{wm} is implicit, made when designing the computational grid; if the exchange process is coded in a more general way, then the choice for h_{wm} is uncoupled from the computational grid and can/must therefore be made explicitly (e.g., Refs. [29, 30]). In this latter case where the exchange location is uncoupled from the grid, h_{wm} could be uniform or variable in space, though most studies in the literature have used a uniform value (including most works by the advisor of this dissertation).

Since h_{wm} influences the modeling in all WMLES approaches, it is self-evident that the accuracy of all WMLES approaches depends on h_{wm}. This fact stands in some contrast to the observation that the current state-of-the-art is to rely on the user to make this choice, however it is made. In addition, the current state-of-the-art advice is to choose $h_{\mathrm{wm}} \approx 0.1\delta$ where δ is the local boundary layer thickness [14]. While this has solid theoretical grounding for equilibrium boundary layers (this height lies in the log-layer and leads to a computational grid that is a good compromise between the requirements for outer layer resolution and minimal log-layer mismatch), there is no general consensus for how to choose h_{wm} in flows with strong non-equilibrium effects (e.g., pressure gradients, corner flows, etc). Moreover, a good non-equilibrium wall-model should tolerate a larger h_{wm} than an equilibrium wall-model. It is thus safe to say that the correct choice of h_{wm} depends on both the flow characteristics and the wall-model being used.

The philosophy of the present work is that the choice of h_{wm} should instead be made by an adaptive algorithm. Our overall hypothesis is that this will improve WMLES by making it a more systematic method, with results that depend less on the user. We primarily see a need for this type of adaptive algorithm in truly complex flows, where the thickness of boundary layers differs by large factors throughout the domain: for example, in a simulation of a complete aircraft. Stated explicitly, our objective with the adaptive algorithm is not to improve accuracy in canonical problems where experienced users can easily choose a nearly optimal h_{wm}, but instead to enable increasingly automated and robust WMLES of complex flows.

1.4 Novelties and Contributions of the Work

Our work resulted in one conference paper and one journal paper as follows:

- “Adaptive Determination of the Wall Modeled Region in WMLES”, AIAA Scitech 2020 Forum [31],
- “Adaptive Determination of the Optimal Exchange Location in Wall-Modeled Large-Eddy Simulation”, AIAA Journal 2022, in press [32].

Our first attempt at developing an algorithm for finding h_{wm} were focused on physics-based ideas related to equilibrium wall-models. Specifically, it introduced two algorithms, one aimed at finding where the constant shear stress region ends (where the equilibrium wall-model is expected to work), the other aimed at finding this location based on information in the complete total stress tensor. This work is presented in chapter 4.

While those original physics-based algorithms were partially successful, they suffered from two important problems: (i) they were insensitive to the actual wall-model being used in the code; and (ii) they were ignorant of the computational cost of solving the resulting WMLES. Thus, the objective of our second attempt was to develop a new algorithm that aims to choose an h_{wm} field while also addressing these issues. The algorithm was derived by framing the problem as an optimization problem that must balance accuracy and cost, and by introducing a systematic way to use the actual wall-model being used in the code to assess how sensitive the accuracy is to the thickness of the wall-modeled layer h_{wm}. This work is presented in chapter 5.

Chapter 2: Relevant Background and Methodology

In this chapter we present the LES equations that were used in this work, a literature review on wall-modeling and the wall-model being used here and its implementation.

Throughout this thesis, Favre or mass-weighted averaging or filtering is defined by $\widetilde{f} = \overline{\rho f}/\overline{f}$ with associated fluctuation $f'' = f - \widetilde{f}$. The reason for this form of averaging or filtering is that it preserves the structure of the compressible Navier-Stokes equations.

2.1 LES and Its Implementation

The LES equations are as follows for the mass, momentum and energy conservations respectively (cf. Garnier et al. [33]):

$$\frac{\partial\overline{\rho}}{\partial t} + \frac{\partial\overline{\rho}\widetilde{u}_j}{\partial x_j} = 0 \tag{2.1}$$

$$\frac{\partial\overline{\rho}\widetilde{u}_i}{\partial t} + \frac{\partial\overline{\rho}\widetilde{u}_i\widetilde{u}_j}{\partial x_j} + \frac{\partial\overline{p}}{\partial x_i} - \frac{\partial\widetilde{\sigma}_{ij}}{\partial x_j} = \frac{\partial\mathcal{T}_{ij}}{\partial x_j} \tag{2.2}$$

$$\frac{\partial\overline{\rho}\widetilde{E}}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{E}+\overline{p})\widetilde{u}_j}{\partial x_j} - \frac{\partial\widetilde{\sigma}_{ij}\widetilde{u}_i}{\partial x_j} + \frac{\partial\widetilde{q}_j}{\partial x_j} = \frac{\partial\mathcal{Q}_j}{\partial x_j} \tag{2.3}$$

where summation over repeating indices on multiplication or division is implied. The energy term is the summation of kinetic and internal energies, represented as

$$\overline{\rho}\widetilde{E} = \frac{\overline{p}}{\gamma - 1} + \frac{1}{2}\overline{\rho}\widetilde{u}_i\widetilde{u}_i, \tag{2.4}$$

whereas the pressure, temperature and density is connected to each other via the ideal gas law,

$$\overline{p} = \overline{\rho}R\widetilde{T}, \tag{2.5}$$

where R is the gas constant.

The term that denotes the viscous stresses in the momentum equations $\widetilde{\sigma}_{ij}$ is modeled as,

$$\widetilde{\sigma}_{ij} \quad = \quad \mu(\widetilde{T})\left(2\widetilde{S}_{ij} - \frac{2}{3}\widetilde{S}_{kk}\delta_{ij}\right), \tag{2.6}$$

where $\mu(\widetilde{T})$ is the temperature-dependent molecular viscosity governed by the power law

$$\mu(\widetilde{T}) = \mu_\infty\left(\frac{\widetilde{T}}{\widetilde{T}_\infty}\right)^{(3/4)}, \tag{2.7}$$

and $\widetilde{S}_{ij}$ is the rate of strain tensor described as

$$\widetilde{S}_{ij} \quad = \quad \frac{1}{2}\left(\frac{\partial\widetilde{u}_i}{\partial x_j} + \frac{\partial\widetilde{u}_j}{\partial x_i}\right). \tag{2.8}$$

The molecular diffusion of heat is represented by $\widetilde{q}_j$ in the energy equation and is modeled by,

$$\widetilde{q}_j = -C_p \frac{\mu}{Pr} \frac{\partial \widetilde{T}}{\partial x_j}, \tag{2.9}$$

where $C_p \mu / Pr$ is the molecular diffusivity calculated with the Prandtl number Pr and the specific heat under constant pressure C_p. The $\widetilde{\sigma}_{ij} \widetilde{u}_i$ is the viscous heating term, which denotes the heat generation by friction in the fluid.

The subgrid scale terms are denoted with the right hand sides of the momentum and energy equations. They are basically identical to the viscous stress term for the momentum equation and the sum of the viscous heating and heat diffusion terms for the energy equation, except that the relevant viscosity and the Pr is different. They are defined as

$$\begin{aligned} \mathcal{T}_{ij} &= 2\mu_{sgs}(\widetilde{S}_{ij} - \frac{2}{3}\widetilde{S}_{kk}\delta_{ij}), & (2.10) \\ \mathcal{Q}_j &= \widetilde{u}_i \mathcal{T}_{ij} + C_p \frac{\mu_{sgs}}{Pr_t} \frac{\partial \widetilde{T}}{\partial x_j}, & (2.11) \end{aligned}$$

where μ_{sgs} is the subgrid scale viscosity and Pr_t is the turbulent Prandtl number.

The subgrid scale model that was consistently used, unless otherwise noted, is the Vreman model [34]. It is described as,

$$\mu_{sgs} = c_{vr} \rho \sqrt{\frac{B_\psi}{\phi_{ij}\phi_{ij}}}, \tag{2.12}$$

where c_{vr} is a user chosen constant dependent on the numerics of the code. The remaining relevant variables are defined as,

$$\phi_{ij} = \frac{\partial \widetilde{u}_j}{\partial x_i}, \tag{2.13}$$

$$\psi_{ij} = \Delta_m^2 \phi_{mi} \phi_{mj}, \tag{2.14}$$

$$B_\psi = \psi_{11}\psi_{22} - \psi_{12}^2 + \psi_{11}\psi_{33} - \psi_{13}^2 + \psi_{22}\psi_{33} - \psi_{23}^2. \tag{2.15}$$

Note that the original Vreman model has a term that excludes the trace of the subgrid scale stresses in equation 2.10, which was omitted here.

The LES equations and the subgrid treatment presented here is only one way of doing this. The interested reader is refered to the comprehensive book by Garnier et al. [33] for a more detailed analysis.

The in-house finite-difference solver *Hybrid* is used to solve these LES equations. It computes the convective fluxes using a solution-adaptative method that switches between a fifth-order WENO scheme with Roe flux splitting near discontinuities and a sixth-order accurate central difference scheme in the split form of Ducros et al. [35] in the rest of the domain. The diffusive terms are computed using a conservative finite-volume-like approach with high-order-like modified wavenumber behavior, and the system is integrated in time using a fourth-order Runge-Kutta method. The inflow turbulence, where needed, is generated using the digital filtering method of Klein et al. [36], and the constant needed by Vreman subrgrid scale model is used as $c_{vr} = 0.03$.

2.2 Wall Stress Modeling and Related Literature

As it was covered in the introductory chapter, the wall-stress models complement the LES equations with a shear stress boundary condition. The wall-model "reads" the flow variables at the exchange (or interface) location located a distance of h_{wm} away from the wall, and generates the wall shear stress τ_{w}, as Figure 1.1 demonstrates. There are many different wall-models, many of them trying to regenerate the unresolved velocity profile. This subsection will cover some literature of the field, mainly to lead upto the current work.

Some earlier works used the log-law as a wall model [22, 37]. While they did modifications that they deemed necessary since they argued the log-law should only work for averaged flow fields, there were later studies that used the log-law as is [38]. The justification is that the eddies that were not resolved for high Reynolds numbers live on a much faster timescale and smaller lengthscale than the resolved large eddies, the cell filtering operation is basically a quasi-averaging [16].

The shortcoming of the log-law is that the buffer layer and the viscous sublayer is not represented. There exist algebraic relations to cover the whole velocity profile down to the wall, e.g. by Spalding [23], Reichardt [39] and Werner and Wengle [40], to overcome this issue.

In order to avoid the prescribed velocity profiles, it is also possible to use the averaged turbulent boundary layer equations (TBLE)

$$\frac{\partial \overline{\rho}\widetilde{u}_1}{\partial t} + \frac{\partial \overline{\rho}\widetilde{u}_1\widetilde{u}_j}{\partial x_j} + \frac{\partial \overline{p}}{\partial x} = \frac{\partial}{\partial y}\left[(\mu + \mu_t)\,\frac{\partial \widetilde{u}_1}{\partial y}\right] \tag{2.16}$$

directly as a wall-model, which requires a complimentary grid in which the TBLE equations are solved. It was initially used by Balaras et al. [41], and then later improved by many works including Cabot and Moin [42], Wand and Moin[43], Kawai and Larsson [27] and Park and Moin [44]. This approach typically yields good results (e.g. [30]), however the complexity of building a new 3-D RANS mesh around the surface raises the application barrier.

One way to simplify the TBLE is to assume equilibrium flow and neglect the time derivative, the convective and the pressure terms, which leaves only the right hand side of equation 2.16 [24]. This essentially turns the problem into a 1-D diffusion problem, which now needs only a 1-D grid for its solution. This equilibrium assumption does not hold for non-equilibrium flows, but this is not as big of a problem as it may seem. The left hand side of equation 2.16 is mainly responsible for the inviscid flow phenomenon in the outer layer, which is already well-resolved in a WMLES. Also, the terms in the left hand side are expected to cancel each other out in an inviscid regime, thus their effect in the near wall viscous layer can be limited. This hypothesis was actually tested and confirmed by Hickel et al. [45]. In practice, it is also observed that the equailibrium wall-model gives moderately accurate results, e.g. the works of Iyer and Malik [26] and Park [30] on wall-mounted hump geometry of NASA.

The simplicity of either a 1-D ODE solution or an algebraic equation has attracted attention of researchers, trying to figure out ways to incorporate non-equilibrium effects while not compromising this implementational simplicity. Just to name a few, Yang et al. [46] has complemented the log-law equation with a linear dependence on y^+ and suggested utilization of Von Karman integral to solve for the additional term. Dzanic and Oefelein [47] have attempted to model the left hand side of equation 2.16 with a polynomial fit obtained by the LES equation and reported improvements over low Re flows.

Note that there are many more wall-models out there, it is a really rich field. Here only a couple were given that we thought were intresting. A further interested reader is referred to the review papers by Piomelli [48], Larsson et al. [14] and Bose et al. [15] for more in depth discussions.

All of these wall-models require a choice of a wall-model exchange location, h_{wm}, and as pointed out by Larsson et al. [14] there is a need for a systematic way to pick it. While it is true that the wall-model itself has to be accurate, which is most of the literature aimed at, the LES information that it is fed is also equally important.

2.3 The Equilibrium ODE Wall Model

The wall-model used in this work is the equilibrium wall-model as described by Kawai and Larsson [24]. The equations for the wall-parallel momentum and energy, respectively, are as follows,

$$\begin{aligned} \frac{d}{dy}\left[\left(\mu_{\mathrm{wm}}+\mu_{\mathrm{t,wm}}\right)\frac{d\widetilde{u}_{\mathrm{wm}}}{dy}\right] &= 0 && (2.17)\\ \frac{d}{dy}\left[C_p\left(\frac{\mu_{\mathrm{wm}}}{Pr}+\frac{\mu_{\mathrm{t,wm}}}{Pr_t}\right)\frac{d\widetilde{T}_{\mathrm{wm}}}{dy}\right] &= -\frac{d}{dy}\left[\left(\mu_{\mathrm{wm}}+\mu_{\mathrm{t,wm}}\right)\widetilde{u}_{\mathrm{wm}}\frac{d\widetilde{u}_{\mathrm{wm}}}{dy}\right], && (2.18) \end{aligned}$$

where $\mu_{\mathrm{t,wm}}$ is the turbulent eddy viscosity, Pr and Pr_t are molecular and turbulent Prandtl numbers respectively and C_p is the specific heat at constant pressure. Note that these equations are basically 1-D diffusion equations which are obtained from the boundary layer equations by assuming the flow is in equilibrium and the effects along the streamwise direction are negligible.

The eddy viscosity of choice was Johnson-King model [49], described with the following equations [50]:

$$\mu_{\mathrm{t,wm}} = \kappa\overline{\rho}_{\mathrm{wm}}\sqrt{\frac{\tau_{\mathrm{w,wm}}}{\overline{\rho}_{\mathrm{wm}}}}y\left[1-\exp\left(-y^{+}/A^{+}\right)\right] \tag{2.19}$$

$$y^{+} = y\sqrt{\overline{\rho}_{\mathrm{w,wm}}\tau_{\mathrm{w,wm}}}/\mu_{\mathrm{w,wm}} \tag{2.20}$$

where $\overline{\rho}_{\mathrm{w,wm}}$ and $\mu_{\mathrm{w,wm}}$ are respectively the density and molecular viscosity at the wall given by the wall-model solution. This eddy-viscosity model is chosen basically because it is simple yet accurate. While there are other eddy viscosity models in RANS framework that are more accurate for multi-D RANS, they come at the cost of solving one or two additional evolution equations, which are not trivial to simplify down to a boundary layer equation in 1-D. Out of the ones that are widely used by the community and do not require an additional evolution equation, i.e. zero-equation models, Johnson-King appear to bring the best accuracy while staying simple [51].

The boundary conditions to close the ODEs are two Dirichlet type boundary conditions, the no-slip (zero velocity) condition at the wall and the wall parallel velocity from the LES at the given interface location h_{wm}, i.e. $\widetilde{u}_{||,LES} = \widetilde{u}_{h_{\mathrm{wm}}}$. The boundary condition for the energy equation at the interface location is similar to momentum equation, it is the temperature of the LES solution $\widetilde{T}_{LES} = \widetilde{T}_{h_{\mathrm{wm}}}$. The boundary condition at the wall is also used as a Dirichlet condition with the specified wall temperature as in the relevant problem description, but it can also be a Neumann condition for other cases.

The ODE is solved using the finite volume method. The 1-D solution space is divided into finite volumes of increasing size with a stretching ratio sr such that

$$\Delta y_j = sr\Delta y_{j-1}, \tag{2.21}$$

where Δy_j is the size of the 1-D finite volume from the left face $y_{face,j}$ to right face $y_{face,j+1}$. The solution lives on the volume (or cell, interchangable words) centers y_j. The momentum and energy equations are then integrated along this volume

$$\int\limits_{y_{face,j}}^{y_{face,j+1}} \frac{d}{dy}\left[\left(\mu_{\mathrm{wm}} + \mu_{\mathrm{t,wm}}\right)\frac{d\widetilde{u}_{\mathrm{wm}}}{dy}\right] dy = \int\limits_{y_{face,j}}^{y_{face,j+1}} 0dy \tag{2.22}$$

$$\begin{aligned}\int\limits_{y_{face,j}}^{y_{face,j+1}} \frac{d}{dy}\left[c_p\left(\frac{\mu_{\mathrm{wm}}}{Pr} + \frac{\mu_{\mathrm{t,wm}}}{Pr_t}\right)\frac{d\widetilde{T}_{\mathrm{wm}}}{dy}\right] dy = \\ -\int\limits_{y_{face,j}}^{y_{face,j+1}} \frac{d}{dy}\left[\left(\mu_{\mathrm{wm}} + \mu_{\mathrm{t,wm}}\right)\widetilde{u}_{\mathrm{wm}}\,\frac{d\widetilde{u}_{\mathrm{wm}}}{dy}\right] dy\end{aligned} \tag{2.23}$$

whose solutions end up to be

$$\left[\left(\mu_{\mathrm{wm}} + \mu_{\mathrm{t,wm}}\right)\frac{d\widetilde{u}_{\mathrm{wm}}}{dy}\right]\Bigg|_{y_{face,j}}^{y_{face,j+1}} = 0 \tag{2.24}$$

$$\left[c_p\left(\frac{\mu_{\mathrm{wm}}}{Pr} + \frac{\mu_{\mathrm{t,wm}}}{Pr_t}\right)\frac{d\widetilde{T}_{\mathrm{wm}}}{dy}\right]\Bigg|_{y_{face,j}}^{y_{face,j+1}} = -\left[\left(\mu_{\mathrm{wm}} + \mu_{\mathrm{t,wm}}\right)\widetilde{u}_{\mathrm{wm}}\frac{d\widetilde{u}_{\mathrm{wm}}}{dy}\right]\Bigg|_{y_{face,j}}^{y_{face,j+1}}. \tag{2.25}$$

Note that the solution lives on the cell centers, however the derivatives and interpolations are needed on the cell faces to solve the equations. The derivatives are calculated using a first-order finite difference formula of

$$\left.\frac{d\phi}{dy}\right|_{y_{face,j}} = \frac{\phi_{face,j+1} - \phi_{face,j}}{y_{face,j+1} - y_{face,j}}, \tag{2.26}$$

and the interpolation operation is performed as linear interpolation between the two cell centers.

When the discrete set of equations is written for each cell, the resulting linear system of equations for each equation can be represented as

$$A_u\vec{\xi_u} = \vec{b}_u \tag{2.27}$$

$$A_T\vec{\xi_T} = \vec{b}_T \tag{2.28}$$

where the $\vec{\xi_u}$ is the solution vector that contains the velocity $\widetilde{u}_{\mathrm{wm}}$, $\vec{\xi_T}$ is the solution vector that contains the temperature $\widetilde{T}_{\mathrm{wm}}$, A_u is the discretization matrix for the left hand side of the momentum equation, A_T is the discretization matrix for the left hand side of the energy equation and $\vec{b}_u$ and $\vec{b}_T$ are the respective discretized right hand sides of the equations. Note that both A matrices are functions of the temperature through the viscosity-temperature relation, and $\vec{b}_T$ is a function of the velocity. To simplify the solution of the system, the following iterative algorithm is used:

```
Initial linear guess for $\vec{\xi_u}$ and $\vec{\xi_T}$
Compute $\mu_{\mathrm{wm}}$ and $\mu_{\mathrm{t,wm}}$
Solve the system 2.27
Compute $\mu_{\mathrm{wm}}$ and $\mu_{\mathrm{t,wm}}$
Solve the system 2.28
Compute $\mu_{\mathrm{wm}}$ and $\mu_{\mathrm{t,wm}}$
Compute $\tau_{\mathrm{w}}$ and $q_{\mathrm{w}}$
if $\tau_{\mathrm{w}}$ and $q_{\mathrm{w}}$ are converged then
    Return $\tau_{\mathrm{w}}$ and $q_{\mathrm{w}}$
else
```

```
11:        Go back to line 2
12:  end if
```

The solutions of the linear systems in steps 3 and 5 are computed using the tridiagonal matrix algorithm, also known as the Thomas algorithm (cf. Chapra and Canale pp. 286-287 [52]).

Chapter 3: Errors in Wall-Models

There are different sources of error for WMLES. Mainly, the errors can originate from the quality of the information that is fed into the wall-model, which would be the responsibility of the LES solution itself; or from the quality of the wall-model to not be able to accurately model the distance between the wall-model exchange location h_{wm} and the wall. In the first section of this chapter, some relevant literature for the first type of error is discussed, however this work does not put any emphasis on this kind of error because this is an already solved problem. In the second section, the errors that occur due to the wall-model itself are discussed, which is new work.

3.1 Errors due to the LES Grid

The wall-model relies on accurate data coming from the LES solution, Figure 1.1. This fact alone means that the minimum location for the interface location is the first LES solution point off the wall. In fact, this has been the standard practice until very recently. The one thing that plagued almost all of these simulations was however the so called log-layer mismatch.

The log-layer mismatch is the name given to the situation when the log-law is not recovered correctly with a simulation. Typically the slope of the log-law, i.e. κ in equation 1.6, is correct in such simulations, but the so called log-law intercept, i.e. B, has some error. The reported log-law curve can have either positive mismatch (e.g. Kawai and Larsson [24]), or negative mismatch (e.g. Cabot and Moin [42], Lee et al. [53]) depending on the combination of numerices, subgrid-scale model, LES grid etc.

The solution was suggested by Kawai and Larsson [24]. They have suggested that the reason of the log-layer mismatch is that the near-wall eddies cannot be properly resolved in the first off wall point due to simple numerics with the Nyquist criterion, that the eddies of the such sizes would neccesarily be underresolved. Their suggested solution was to put the h_{wm} away from the first off wall grid point and more into the flow field, thus allowing the proper resolution of the eddies of size h_{wm}. They also suggest that for such eddies to be properly resolved, the wall-parallel grid spacings are also important. They thus reason that the grid spacings should be fractions of h_{wm} such that

$$\Delta x_i = K_i h_{\text{wm}} \tag{3.1}$$

where K_i is a numerical model-dependent constant. This relation is used throughout this study when building a grid compatible with any generated h_{wm} field, with the constants $K_1 = 0.8$, $K_2 = 0.3$, $K_3 = 0.8$ where the first constant represents the streamwise, the second the wall-normal and the third the spanwise direction.

3.2 Errors due to Wall Model and h_{wm}

The output of a wall-stress model is the wall shear stress τ_{w} and (for an isothermal wall) the wall heat flux q_{w}. The stand-alone accuracy of a wall-model can therefore be assessed by taking accurate DNS (or wall-resolved LES) data as input and comparing the predicted wall-model

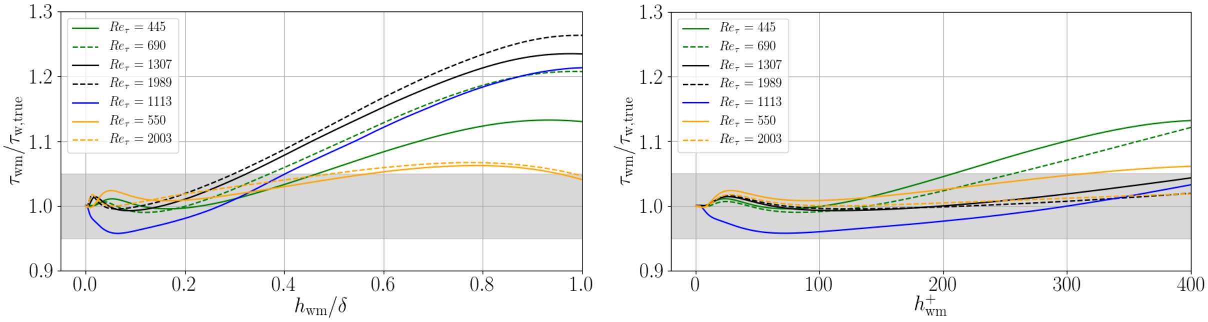

Figure 3.1: *A priori* error in an equilibrium wall-model based on DNS data for different equilibrium flows [1, 2, 3, 4, 5, 6, 7]. Black and green curves are incompressible boundary layer flow, orange curves are incompressible channel flow and the blue curve is a supersonic boundary layer flow. The shaded gray region marks $\pm 5\%$ error compared to the DNS value.

outputs (τ_{w} and q_{w}) to the DNS values. This type of *a priori* assessment provides a lower bound on the error in a WMLES, since errors lower than this in an actual WMLES would necessarily be due to some type of error cancellations.

The first *a priori* test is DNS of different equilibrium flows by del Alamo and Jimenez [1], Hoyas and Jimenez [2], Simens et al. [3], Jimenez et al. [4], Sillero et al. [5, 6] and Pirozzoli et al. [7]. The mean velocity from the DNS is fed into the wall-model at different heights h_{wm}/δ, where δ is the boundary layer thickness or the channel half-width. The resulting relative error is shown in Fig. 3.1. The relative error seems to reach a small but finite value below roughly $h_{\mathrm{wm}}/\delta = 0.2$, however it does not converge directly onto zero. These finite errors are very likely due to the turbulent eddy viscosity not modeling the buffer layer right, as can be confirmed in right figure of 3.1, such that the highest error is observed for roughly $h^+_{\mathrm{wm}} \approx 30$ for the incompressible cases. The compressible case seems to not get it quite right at all, which is again due to the eddy viscosity of the wall model not being calibrated for a compressible flow.

The second *a priori* test is conducted using the dataset published by Zhang et al. [8], with

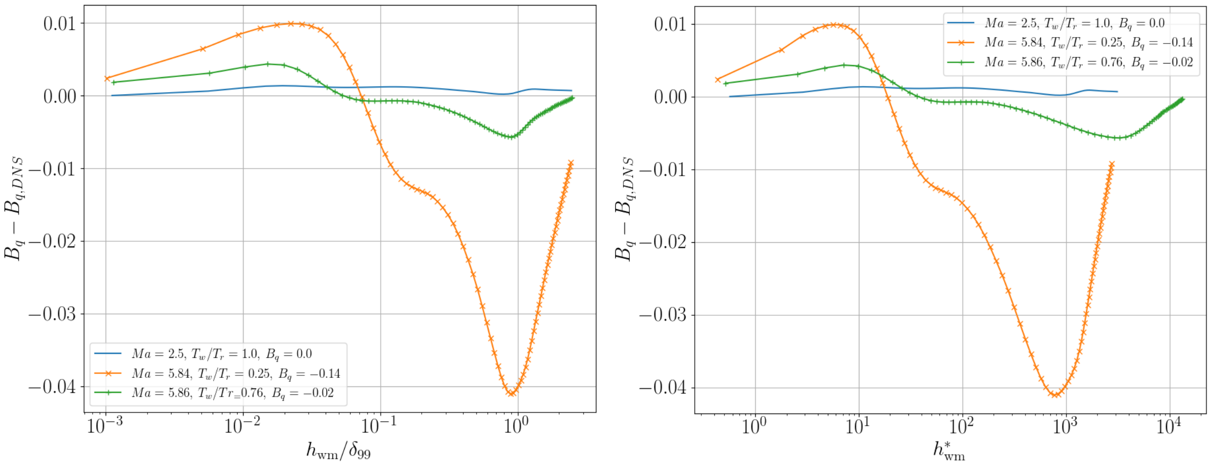

Figure 3.2: *A priori* error in an equilibrium wall-model based on DNS data for different equilibrium flows [8]. The solid curve represents adiabatic wall at $Ma = 2.5$, the "+" marked curve represents a cold wall at $Ma = 5.84$, $T_w/T_r = 0.76$ and the "x" marked curve represents a colder wall at $Ma = 5.84$, $T_w/T_r = 0.25$.

the purpose of determining the error behavior of the wall-model in determining the wall heat flux. The resulting non-dimensional heat transfer $B_q = q_w/(C_p \rho_w u_\tau T_w)$ is plotted in Figure 3.2. It is seen that if the wall is adiabatic, the wall-model cannot really get it wrong and produce a significant wall heat transfer. However, if there is wall heat transfer, the wall-model generates at least 10% error in heat transfer as long as the h_{wm} is not in the buffer layer. In the buffer layer, towards the viscous sublayer, the correct value is gradually obtained. It looks like there is a step error from the buffer layer into the log-layer, and then the error behavior is qualitatively similar to what τ_{w} was for the previous test case.

One caveat should also be added here, the y^+ in the eddy viscosity in equation 2.19 was replaced with $y^* = y\sqrt{\rho \tau_{\mathrm{w}}}/\mu$ where the quantities that are not marked with subscript w are local wall-model quantities. This was done with the knowledge that this type of eddy viscosity gives a more accurate solution for walls with heat transfer.

The last *a priori* test is the NASA wall-mounted hump, for which wall-resolved LES was performed by Uzun and Malik [9]. Ali Uzun has recently performed even higher resolution computations of this case; while that data is yet unpublished, it was provided to us in private communication by Dr. Uzun [10]. The geometry is shown in Fig. 3.3 which also shows the 8 locations where the *a priori* assessment was performed.

As the $h_{\rm wm}$ reaches the wall, the wall-model-predicted $\tau_{\rm wm}$ converges to the true value; however, this occurs only when the $h_{\rm wm}$ reaches the viscous sublayer for some locations, including those experiencing strong acceleration on the forebody of the hump and the locations in the separation bubble and immediately beyond it. This *a priori* test shows that a relative error of 25% or less will necessitate (with the present wall-model) effectively switching to wall-resolved LES on the forebody and in the separation region; this would be a highly conservative approach to WMLES, with high accuracy at a high cost. If one were to tolerate a relative error in the 25-50% range, the method would resort to WRLES only in the separated region.

The reasoning above assumes that the relative error is important, but this is plausibly not true in the separated region where the wall stress is very small.

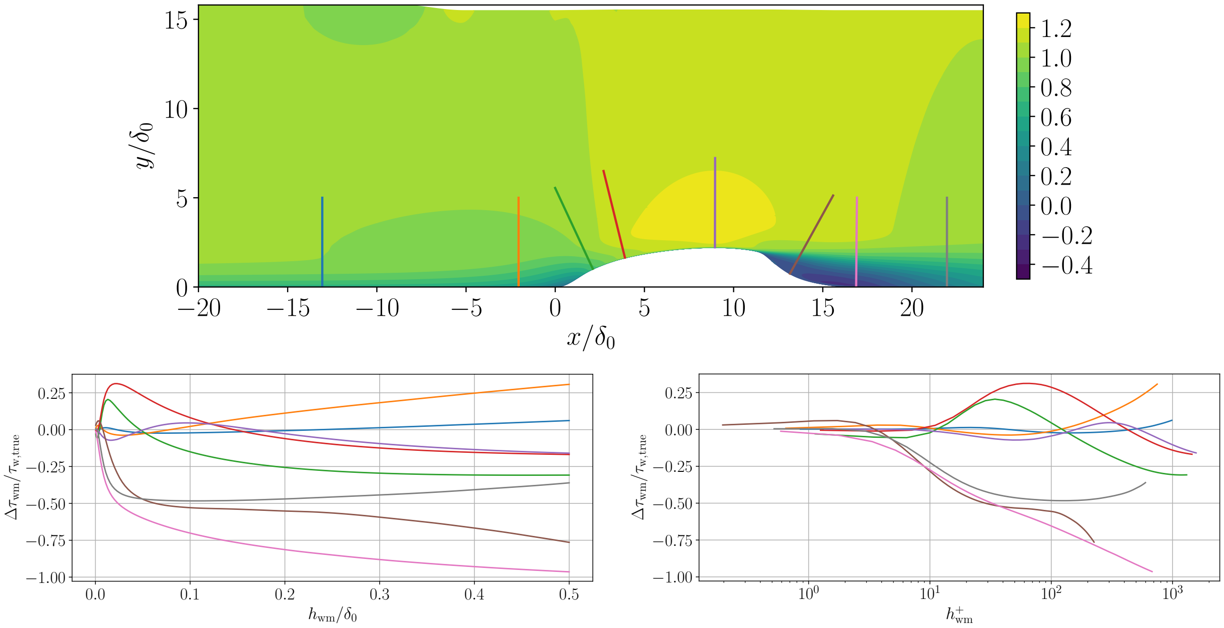

Figure 3.3: *A priori* assessment using wall-resolved LES data by Ali Uzun [9, 10] for the NASA wall-mounted hump. Mean streamwise velocity field (contours) and locations of *a priori* assessment (top). Relative error in the predicted wall stress $\Delta\tau_{\rm wm} = \tau_{\rm wm} - \tau_{\rm w,true}$ when fed WRLES data from different heights $h_{\rm wm}$ in outer scaling (bottom left) and in inner scaling (bottom right).

Chapter 4: Physics Based Algorithms

This chapter describes our work as published in the conference proceedings of AIAA Scitech Forum 2020 [31]. We introduce two algorithms based on our understanding of the physics of equilibrium wall-model, one trying to find where the equilibrium assumption is not valid off the wall, the other trying to pinpoint 0.1δ using the total stress tensor.

4.1 Algorithms for Finding the Wall-Model Height

The algorithms are defined on a wall-normal line starting at a wall point assuming that data (velocity, shear stress, etc.) exists along that line. The h_{wm} they generate is then used in the next simulation.

Before starting the algorithms, some definitions are needed. The total stress tensor is defined as

$$\tau_{ij} = 2\mu\hat{S}_{ij} - \overline{\rho}\widetilde{u_i''u_j''}\,, \tag{4.1}$$

where

$$\hat{S}_{ij} = \frac{1}{2}\left(\frac{\partial\widetilde{u}_i}{\partial x_j} + \frac{\partial\widetilde{u}_j}{\partial x_i} - \delta_{ij}\frac{2}{3}\frac{\partial\widetilde{u}_k}{\partial x_k}\right) \tag{4.2}$$

is the deviatoric part of the rate-of-strain tensor. Note that we ignore the contribution to the total stress of any explicit subgrid model.

The wall-parallel shear stress of the streamwise velocity is denoted by τ_{pn}, which is obtained by a projection operation.

4.1.1 Algorithm 1, Based on the Total Shear Stress

The underlying assumption of the law of the wall is the existence of a constant shear stress layer, where the total shear stress is almost equal to the wall shear stress τ_w. In a Cartesian coordinate system,

$$\tau_{12} = \mu \frac{\partial \widetilde{u}}{\partial y} - \overline{\rho}\widetilde{u'v'} \approx \tau_w \,. \tag{4.3}$$

Since equilibrium wall stress models are derived in a way that effectively assumes a constant total shear stress, it is intuitively appealing to assume that the deviation from this state is related to the ideal choice of h_{wm}. We then define Algorithm 1 as

$$h_{\mathrm{wm}} = \alpha(\beta) \min\left\{ y : \, |\tau_{pn}(y) - \tau_w| / |\tau_w| \leq \beta \right\} \,, \tag{4.4}$$

where $\beta \in (0, 1)$ is a tolerance parameter to be set by the user and $\alpha(\beta)$ is a correction factor to be calibrated using reference data. The basic idea behind the algorithm and the inclusion of the $\alpha(\beta)$ correction factor is sketched in Fig. 4.1, which also shows sample τ_{pn}^{+} profiles from DNS. Small values of β (say, $\beta \approx 0.1$) will target the end of the constant stress layer directly, while larger values of β (say, $\beta \approx 0.5$) will target the middle of the boundary layer. The role of the correction factor $\alpha(\beta)$ is to make the algorithm produce an estimated h_{wm} regardless of which part of the boundary layer is targeted.

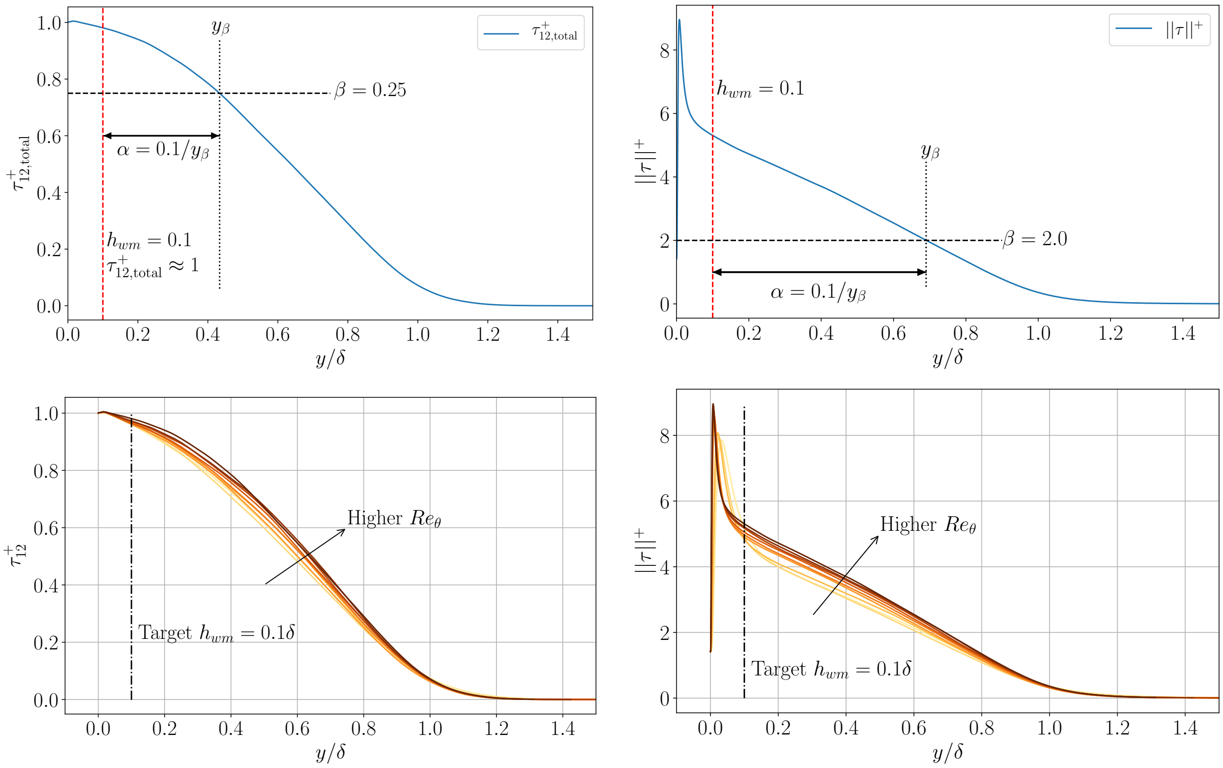

Figure 4.1: Schematic of how the algorithms work (top row) and the boundary layer DNS data used for calibration (bottom row). Algorithm 1 based on the total shear stress (left column) and Algorithm 2 based on the Frobenius norm of the total stress tensor (right column). The DNS data covers the range $Re_\theta = 1100 - 6500$ [3, 4, 5, 6].

4.1.2 Algorithm 2, Based on the Norm of the Total Stress Tensor

Algorithm 2 relies on the Frobenius norm of the stress tensor $||\tau|| = \sqrt{\tau_{ij}\tau_{ij}}$, which is also shown in Fig. 4.1 for a range of Reynolds numbers. The idea is to include the full information in the stress tensor rather than just the shear stress component. The algorithm is defined as

$$h_{\mathrm{wm}} = \alpha(\beta) \min \left\{ y : \, ||\tau|| < \beta|\tau_w| \, \text{and} \, y > y_{\|\tau\|_{max}} \right\} \, , \tag{4.5}$$

where $y_{\|\tau\|_{max}}$ is the wall normal location where the maximum $||\tau||$ is reached. The reason for the latter condition is that $||\tau||$ peaks in the viscous buffer layer; by including the latter condition we ensure that the algorithm targets locations farther out in the boundary layer. An alternative method could be to instead require that $\partial\|\tau\|/\partial y < 0$.

Note that small values of β target the outer edge of the boundary layer while larger values target locations closer to the wall.

4.1.3 Calibration of the $\alpha(\beta)$ Correction Factor

The correction factor $\alpha(\beta)$ is required to make the algorithms predict h_{wm} for different tolerance levels β. The basic process for the algorithms is shown in Fig. 4.1. For a user-specified β value, the corresponding wall distance y_β is found, which is then multiplied by the $\alpha(\beta)$ correction factor to produce the final predicted h_{wm}. The correction factors are calibrated using τ^+_{pn} and $\|\tau\|^+$ profiles from boundary layer DNS data by Simens et al. [3], Jimenez et al. [4], and Sillero et al. [5, 6]; with the target of $h_{\mathrm{wm}}/\delta \approx 0.1$ which is commonly recommended in the literature [cf. 14]. Note that this target is consistent with the *a priori* test in Fig. 3.1 which suggests

Algorithm 1			Algorithm 2		
β	α	$\approx y_\beta/\delta$	β	α	$\approx y_\beta/\delta$
0.05	0.676	0.148	0.25	0.096	1.042
0.125	0.389	0.257	0.5	0.105	0.952
0.25	0.252	0.397	1.0	0.119	0.840
0.375	0.196	0.510	1.5	0.134	0.746
0.5	0.163	0.613	2.0	0.153	0.654
0.675	0.133	0.752	3.0	0.215	0.465
0.75	0.123	0.813	4.0	0.357	0.280

Table 4.1: Selected α, β pairs and the approximate height y_β/δ that these β would target.

that $h_{\mathrm{wm}}/\delta \lesssim 0.2$ produces sufficiently low errors. The DNS data ranges from $Re_\theta \approx 1100$ to $Re_\theta \approx 6500$, with α calibrated to produce predicted h_{wm}/δ that are centered around the target value of 0.1 for this range of Reynolds numbers.

The results of the calibration are shown in Fig. 4.2, with selected values listed in Table 4.1. The uncertainty in h_{wm} is about $\pm 20\%$ for small β values (that target the end of the constant stress layer directly). The uncertainty decreases for larger β values that target the shear stress profile farther out in the boundary layer. This greater robustness must, of course, be balanced against the larger degree of non-universality in the outer parts of the boundary layer in non-equilibrium flows.

With the ranges of β considered here, Algorithm 2 targets locations closer to the edge of the boundary layer which results in lower uncertainty; however, Algorithm 2 produces about the same uncertainty when targeting similar locations in the boundary layer.

4.2 *A priori* Assessment of the Algorithms

We next assess the algorithms using DNS or LES data for four different flows. The algorithms are used, with the given DNS/LES data, to compute the h_{wm} distribution in space. We then

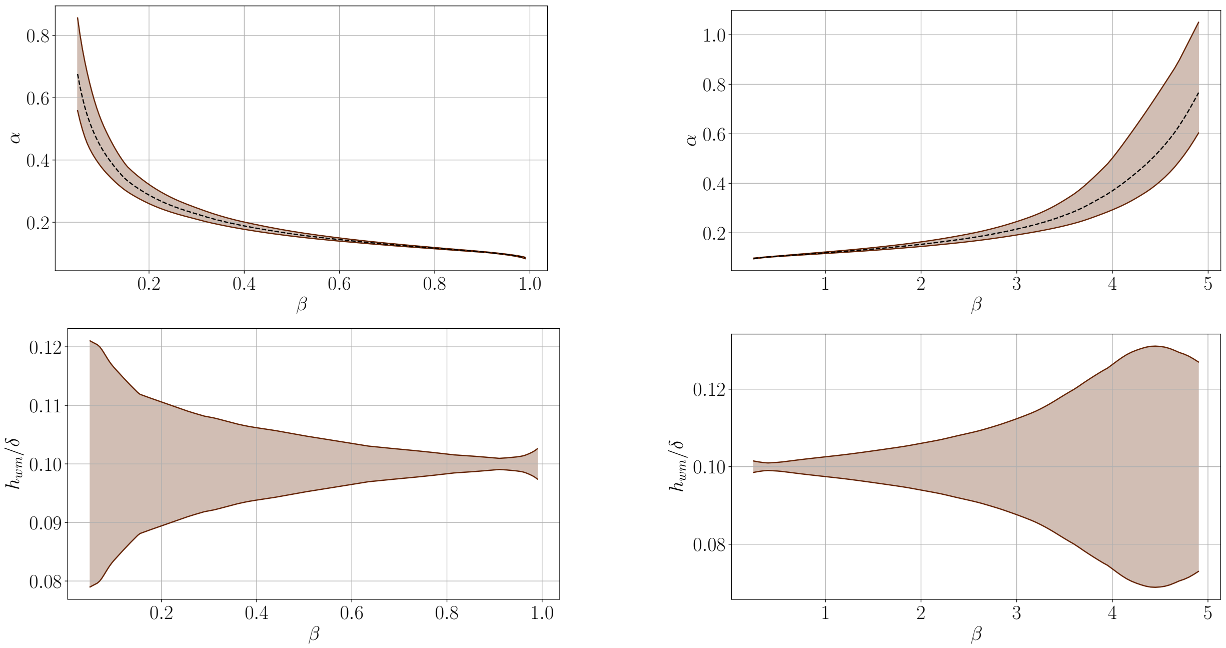

Figure 4.2: Calibration of the $\alpha(\beta)$ correction factor for algorithms 1 (left column) and 2 (right column). Showing the calibrated $\alpha(\beta)$ curves (top) and the resulting predicted range of h_{wm} for the range of Reynolds numbers in the boundary layer DNS data.

extract the averaged LES solution at that h_{wm} height and use this as input to compute the wall stress predicted by the wall-model $\tau_{w,\text{wm}}$. This is then compared to the true τ_w from the existing LES.

Each algorithm is used with three different values of β, targeting different heights in an equilibrium boundary layer as documented in Table 4.1. Algorithm 1 is used with $\beta = 0.25$ (targeting $y/\delta \approx 0.4$), $\beta = 0.5$ ($y/\delta \approx 0.6$), and $\beta = 0.75$ ($y/\delta \approx 0.8$). Algorithm 2 is used with $\beta = 2.0$ ($y/\delta \approx 0.7$), $\beta = 1.0$ ($y/\delta \approx 0.8$), and $\beta = 0.5$ ($y/\delta \approx 1.0$). We thus note that the cases of Algorithm 1 with $\beta = 0.75$ and Algorithm 2 with $\beta = 1.0$ can be directly compared since they target the same location in an equilibrium boundary layer. We also note that we used slightly different, but not enough to affect the qualitative results, α values than we suggest in Table 4.1.

4.2.1 DNS of Shock/Boundary-Layer Interaction (SBLI)

The first test case is the DNS data of Volpiani et al. [54] for a Mach 2.3 flow with an oblique shock impinging on a turbulent boundary layer over an adiabatic wall. The incoming boundary layer is at $Re_\theta \approx 1000$ and $Re_{\delta 2} \approx 650$, which is really too low for wall-modeled LES to make much sense; it is used here regardless as a first assessment of the algorithms.

The results of applying the algorithms are shown in Fig. 4.3. Both algorithms, for all values of β, predict $h_{\text{wm}} \approx 0.1\delta$ in the incoming boundary layer, as expected given the calibration. The inflow turbulence is generated using the digital filtering method and is thus quite unphysical for the first 5-10 δ_0. Algorithm 2 is robust to this issue, as is Algorithm 1 with $\beta = 0.75$. Algorithm 1 with the lower values of β, however, shows a clear sensitivity to this unphysical turbulence. We note that the robust algorithms all target $y/\delta \gtrsim 0.7$ while the sensitive ones target $y/\delta \lesssim 0.6$,

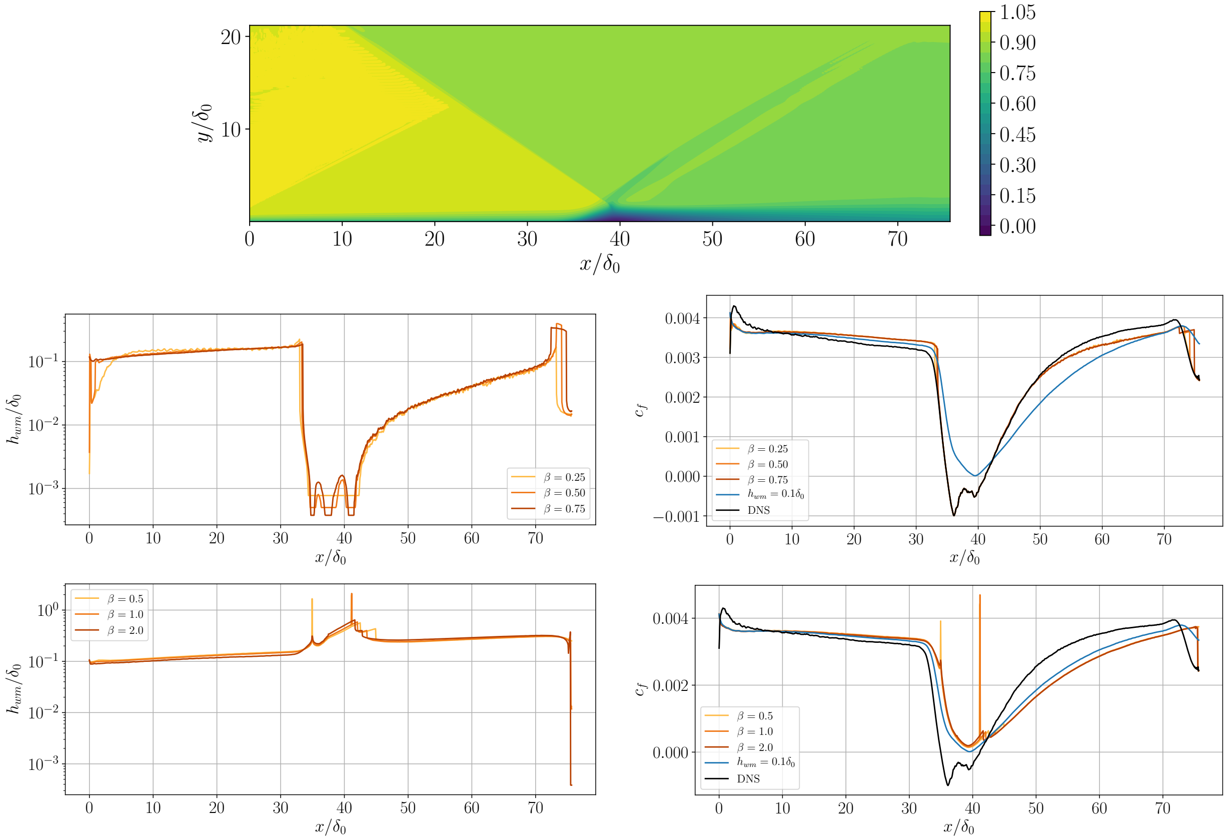

Figure 4.3: *A priori* results for the shock/boundary-layer interaction problem in section 4.2.1, showing the predicted h_{wm} (left column) and the resulting predicted wall stress using that h_{wm} (right column). Shown for Algorithm 1 (middle row) and 2 (bottom row). Compared to the DNS truth and the results of a wall-model with uniform $h_{\mathrm{wm}} = 0.1\delta_0$.

hence it is unclear whether this is an effect of the algorithms or what location in the boundary layer they are targeting.

The algorithms behave completely differently from each other as the separation point is approached around $x/\delta_0 \approx 35$. Algorithm 1 predicts a very small h_{wm} that is consistent with WRLES while Algorithm 2 produces increased h_{wm} compared to the incoming boundary layer. This difference is clearly due to the different natures of the algorithms. Algorithm 2 senses the overall turbulence level which is high in the separation bubble which thus leads to an increased h_{wm}.

In terms of the predicted $\tau_{w,\mathrm{wm}}$, the differences with DNS near the inflow are not meaningful since the DNS is affected by the unphysical inflow turbulence as well. The wall-model-predicted wall stress is consistently higher than the DNS value in the incoming boundary layer. This is likely caused by the low Reynolds number of this flow, which is known to produce a higher log-law intercept (and thus a lower wall stress) than the classic one. The wall-model is designed to mimic high-Re wall turbulence and should therefore overpredict the wall stress.

Algorithm 1 defaults to a WRLES in the separation bubble and hence produces a perfectly predicted $\tau_{w,\mathrm{wm}}$. Algorithm 2 produces a quite erroneous $\tau_{w,\mathrm{wm}}$ throughout the separation bubble and into the recovering post-bubble boundary layer, although there seems to be a recovery towards better accuracy as the boundary layer re-equilibrates. The profiles beyond $x/\delta_0 = 70$ are in the sponge layer of the DNS and are therefore meaningless.

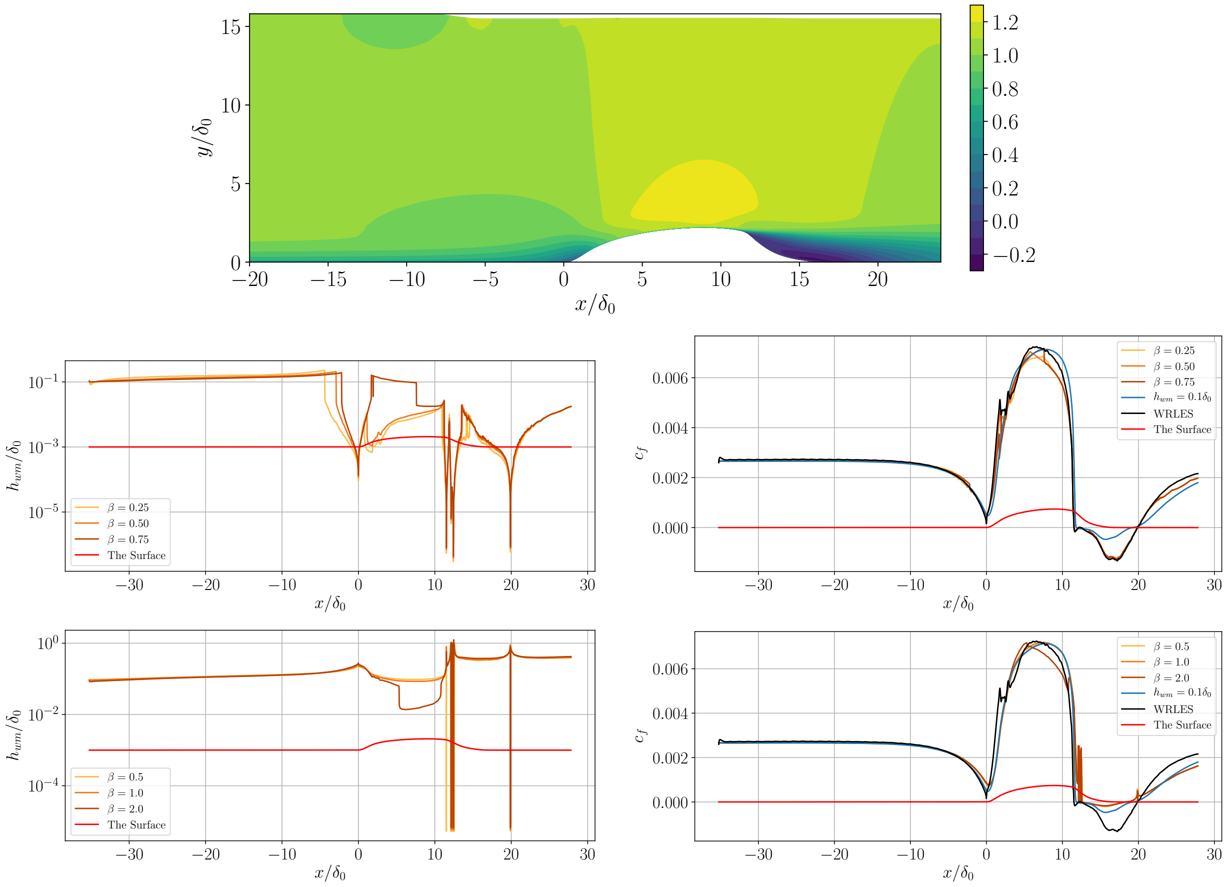

Figure 4.4: *A priori* results for the wall-mounted hump case in section 4.2.2, with the predicted h_{wm} (left column) and the resulting friction coefficient using that h_{wm} (right column). Shown for Algorithm 1 (middle row) and 2 (bottom row). Compared to the WRLES by Uzun [10] and the results of a wall-model with uniform $h_{\mathrm{wm}} = 0.1\delta_0$.

4.2.2 WRLES of the NASA Wall-Mounted Hump

The second test case is a NASA turbulence validation case, the wall-mounted hump at low Mach number. The unpublished WRLES data of Ali Uzun is used, who was very kind to share it with us. A published version of the case can be found in Uzun and Malik [9].

The results of the application of the algorithms, with the mean flow field and geometry, are shown in Fig. 4.4. Same as in the previous test case, both algorithms predict $h_{\mathrm{wm}} \approx 0.1\delta$

in the incoming boundary layer. As the first non-equilibrium effects right before the hump are felt, Algorithm 1 predicts smaller h_{wm} while Algorithm 2 predicts larger. On the hump, both algorithms show differences generated by different β values. Algorithm 1 predicts larger h_{wm} for larger β values over most of the hump. This is the result of higher β bypassing the non-equilibrium effect created due to the favorable pressure gradient on the hump, and catching the outer boundary layer. The smaller β values, however, catch this pressure gradient effect and predict finer h_{wm} as a result. The situation is similar for Algorithm 2, where the highest β value (which targets the lowest y/δ in the equilibrium boundary layer) catches the effect of the pressure gradient and generates a lower h_{wm} while the lower β values do not get triggered by it.

The separated region also shows a difference between the algorithms. Algorithm 1 tends to have a lower h_{wm}, due to catching the strong non-equilibrium effect by the reversed flow. Algorithm 2 on the other hand, sees the turbulent activity all through the separation bubble and predicts an h_{wm} above it all, avoiding that region. The reattached region shows slow increase of the h_{wm} by Algorithm 1 as the flow re-equilibriates, where Algorithm 2 keeps predicting an h_{wm} above the whole turbulent layer in the not yet re-equilibriated flow.

The friction coefficient c_f (defined here as $c_f = \tau_{\mathrm{w}}/[0.5\rho_\infty \widetilde{u}_\infty^2]$) predictions by both of the algorithms for the incoming boundary layer are accurate. At the beginning of the hump and on the hump the algorithms do not show large deviations from the wall-resolved τ_{w}. In the separation bubble, Algorithm 1 shows very accurate predictions while Algorithm 2 underpredicts c_f, even worse than the reference $h_{\mathrm{wm}} = 0.1\delta_0$ case. The reattachment shows slight deviation for Algorithm 1, where Algorithm 2 keeps under-predicting.

4.2.3 WRLES of Transonic Axisymmetric Bump

The third test case is the Bachalo-Johnson problem, an axisymmetric bump at transonic flow conditions. This case has two distinctions from the wall-mounted hump. First, the wall is not planar, it is axisymmetric. Second, the separation is caused by a shock due to the accelerated flow on the smooth body, and not by the rapid change of geometry. The WRLES data of Uzun and Malik [11] is used. Figure 4.5 shows the mean flow field and geometry. Note the shock at $x/\delta_0 \approx 30$.

The results of the algorithms are shown in Fig. 4.5. They look qualitatively very similar to the wall-mounted hump, given the similar geometries. As before, both algorithms predict $h_{\mathrm{wm}} \approx 0.1\delta$ in the incoming boundary layer. Algorithm 1 responds to the first non-equilibrium effects of the bump by reducing the h_{wm}, where Algorithm 2 responds by slightly increasing it. On the hump, the lower β of Algorithm 1 bypasses the favorable pressure gradient effect for a small portion, but then catches it like the rest of the β values. Algorithm 2 catches the non-equilibrium by the pressure gradient with its higher two β values. The separation sees defaulting to the WRLES by both of the algorithms, however with different mechanisms. Algorithm 1 defaults because the non-equilibrium effect is too strong, but Algorithm 2 defaults because it fails to find a proper h_{wm} with $\tau_w \approx 0$, in which case the algorithm forces $h_{\mathrm{wm}} = 0$. In the separated region, Algorithm 1 predicts lower h_{wm} than Algorithm 2, as it did for the wall-mounted hump case. Note that there is a secondary separation bubble at the “corner” of the end of the bump, where the algorithms default to WRLES again. In the reattached region, just like in the wall-mounted hump case, Algorithm 1 slowly increases the h_{wm} as the flow reaches equilibrium again, while Algorithm 2 waits for it at a relatively high h_{wm}.

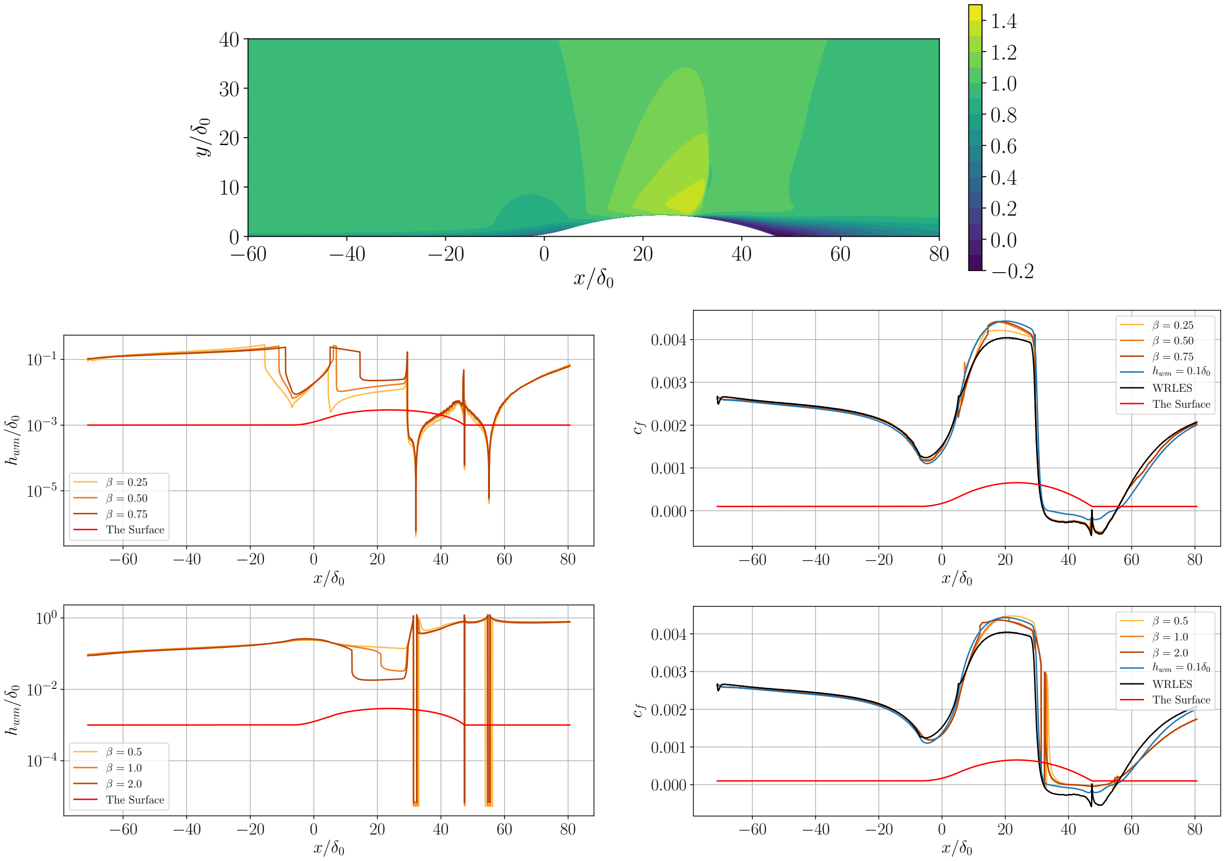

Figure 4.5: *A priori* results for the axisymmetric bump case in section 4.2.3 with the predicted h_{wm} (left column) and the resulting friction coefficient using that h_{wm} (right column). Shown for Algorithm 1 (middle row) and 2 (bottom row). Compared to the WRLES by Uzun and Malik [11] and the results of a wall-model with uniform $h_{\text{wm}} = 0.1\delta_0$.

The c_f predictions of both of the algorithms are accurate in the incoming boundary layer. At the beginning of the hump, both algorithms show slight deviations from the WRLES values, but not by large amounts. However, on the hump, both algorithms overpredict the c_f. After the hump, in the separated and reattachment region, Algorithm 1 predicts very accurate c_f, almost always staying with the WRLES. Algorithm 2 is not as successful as it underpredicts the c_f even worse than the constant $h_{\text{wm}} = 0.1\delta_0$ case.

4.2.4 WMLES of Subsonic Equilibrium Boundary Layer

The last *a priori* test case is the WMLES of a zero pressure gradient boundary layer on a flat plate. All of the previous cases were wall-resolved, so the algorithms had complete information all the way down to the wall. We now assess the algorithms without that "perfect" wall information, noting that being used in WMLES is the actual reason they are developed for.

The case is run with the in-house code `Hybrid` as defined in Chapter 2. The domain has dimensions $L_x = 70\delta_0$, $L_y = 10\delta_0$, $L_z = 9\delta_0$ with the grid resolutions $\Delta x = 0.117\delta_0$, $\Delta y_w = 0.03\delta_0$, $\Delta z = 0.1\delta_0$ and $h_{\text{wm}} = 0.16\delta_0$. We chose this h_{wm} because this would be $h_{\text{wm}} \approx 0.1\delta$ when the flow becomes physically meaningful turbulence around the middle of the domain, and with this grid it would satisfy the requirements of Kawai and Larsson [24] as presented in Chapter 3. The momentum thickness based Reynolds number is $Re_\theta = \rho_\infty \widetilde{u}_\infty \theta / \mu_\infty \approx 3500$ in the middle of the domain, and $M_\infty = 0.8$.

We first verify our results by comparing the Van Driest transformed velocity and the Reynolds stresses with the DNS of Sillero et al. [5, 6], shown in Fig. 4.6. The mean velocity matches the log-law, and the Reynolds stresses show good agreement with the DNS in the LES region of the

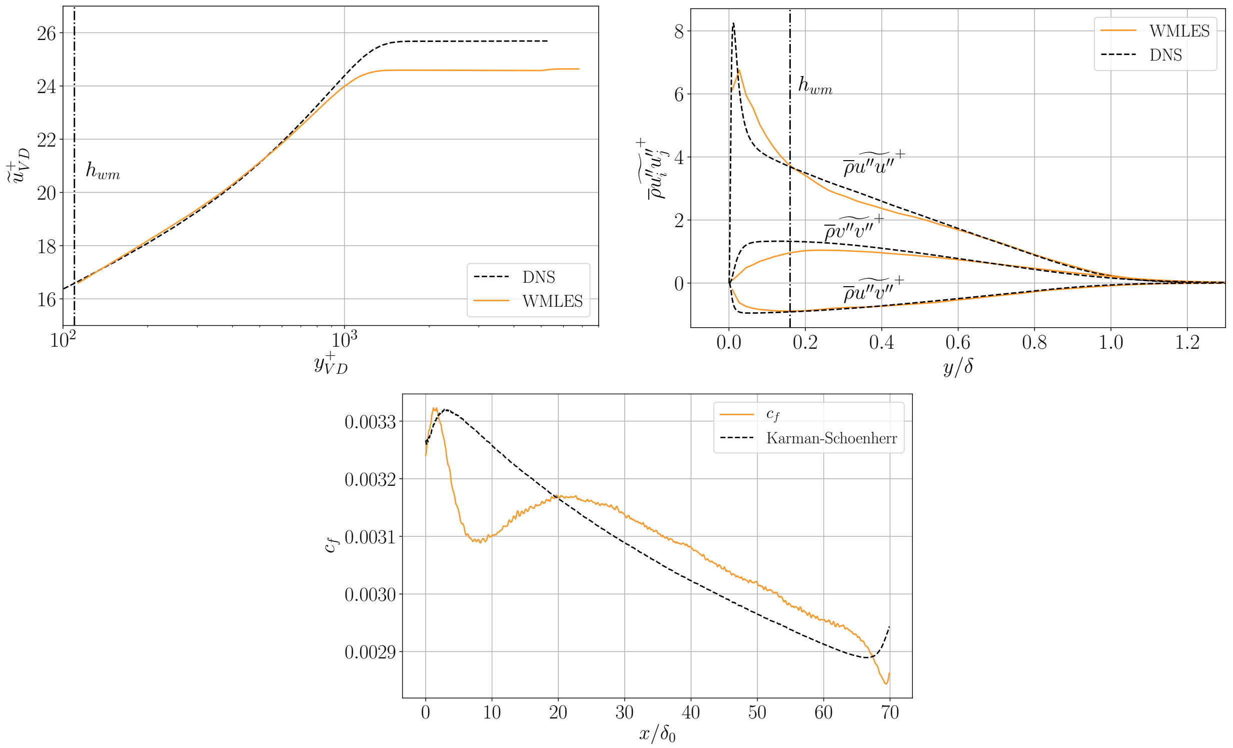

Figure 4.6: WMLES of equilibrium boundary layer with uniform h_{wm}, used for *a priori* assessment. Top row showing the Van Driest transformed mean velocity (top left) and Reynolds stresses (top right), both compared to the DNS of Sillero et al. [5, 6]. Bottom showing the local skin friction coefficient compared to the empirical Karman-Schoenherr formula.

flow. The skin friction coefficient becomes almost parallel to the Karman-Schoenherr relation around $x/\delta_0 \approx 35$, with a small mismatch. Given that the Karman-Schoenherr relation is empirical in nature, and there exists other relations resulting in slightly different curves, we conclude that the flow is physically meaningful for $x/\delta_0 \gtrapprox 35$.

We then apply our algorithms, with results shown in Fig. 4.7. The main behavior we expect is that they follow 0.1δ lines roughly, as they are calibrated to do so, and they achieve that. Note that the flow very near the inlet is unphysical, causing the algorithms to behave inaccurately. We observe some wiggles in the h_{wm} prediction for Algorithm 1, particularly with lower β thresholds.

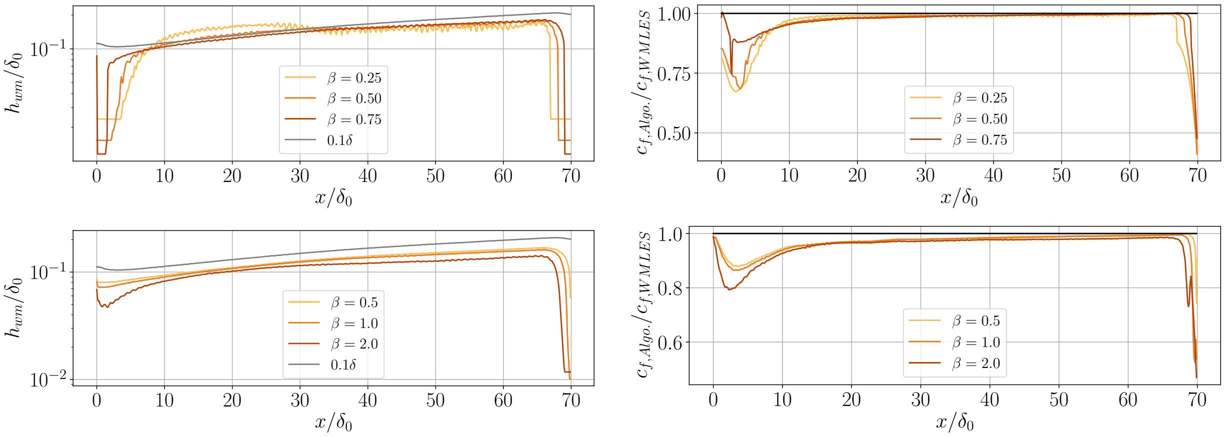

Figure 4.7: *A priori* results for the WMLES of equilibrium boundary layer case in section 4.2.4 with the predicted h_{wm} (left column) and the resulting friction coefficient using that h_{wm} compared to the WMLES (right column). Shown for Algorithm 1 (top row) and 2 (bottom row).

Also, the lowest β seems to be most affected by the incoming unphysical turbulence, with effects being significant until $x/\delta_0 \approx 30$. Algorithm 2 does not suffer from wiggles like Algorithm 1 does, likely due to the averaging of the individual Reynolds stresses together. The end of the domain has a sponge layer starting at $x/\delta_0 = 65$, so the predictions in that region are again unphysical. The τ_w estimations also look good, with the physically correct region being close to the "truth" value of the WMLES, especially so for the larger h_{wm} thicknesses of Algorithm 1.

4.3 *A posteriori* Assessment

We finally perform an *a posteriori* test on the equilibrium boundary layer, by starting from a clearly erroneous h_{wm} field and testing whether the algorithms can push the simulation towards the correct one. For each simulation, once the h_{wm} distribution was determined, the grids were designed to satisfy the requirements of Kawai and Larsson [24] as presented in Chapter 3.

Table 4.2: The grids of the adaptive WMLES runs of equilibrium boundary layers produced by Algorithm 1 with $\beta = 0.25$.

Run	$\Delta x/\delta_0$	$\Delta y_w/\delta_0$	$\Delta z/\delta_0$
0	2.8	0.714	1.5
1	0.496	0.185	0.474
2	0.131	0.049	0.13

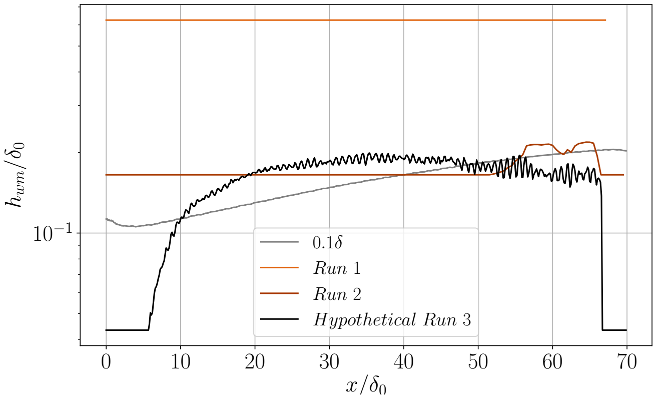

Figure 4.8: The predicted next h_{wm} distributions of each of the adaptive runs of equilibrium flat plate using Algorithm 1 with $\beta = 0.25$.

The first run, labeled “run 0”, is done with an extremely coarse grid with grid spacing $\Delta x = 2.8\delta_0$, $\Delta y = 0.7\delta_0$, $\Delta z = 1.5\delta_0$ and $h_{\text{wm}} = 4.0\delta_0$. This is nowhere near sufficient, but we wish to assess the algorithms’ abilities to suggest refinement where needed. Also, this type of situation can easily happen in realistic (complex) geometries.

We start by presenting the results for Algorithm 1 with $\beta = 0.25$, which produces the h_{wm} fields for each run in Fig. 4.8 and the associated grid spacings in Table 4.2. The predicted h_{wm} fields are noisy, and are therefore spatially filtered before being used in the subsequent run. In addition, the h_{wm} is extrapolated in a constant way throughout the unphysical regions $x/\delta_0 < 30$ and $x/\delta_0 > 60$.

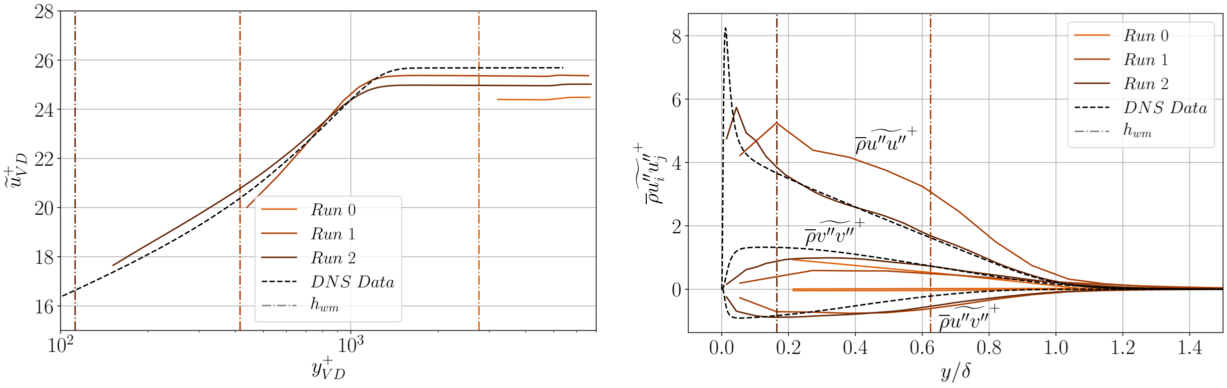

Figure 4.9: *A posteriori* assessment of the adaptive WMLES runs of the equilibrium boundary layer with Algorithm 1 and $\beta = 0.25$, compared to the DNS of Sillero et al. [5, 6]. Van Driest transformed velocity (left) and Reynolds stresses (right).

The predicted h_{wm} is basically the same after runs 1 and 2 (i.e., for run 2 and the hypothetical run 3), and hence the process was stopped after run 2. This illustrates the abilities of Algorithm 1: it is able to predict refinement when needed, and also senses when the "converged" h_{wm} is reached.

The mean velocity and Reynolds stresses from each run are shown in Fig. 4.9. Note that the h_{wm} is too high to allow a log-law to form and almost any turbulent fluctuations to develop in run 0. Then velocity shows good convergence starting from run 1, and the Reynolds stresses reach a good state for run 2. There is a slight log-layer mismatch at the run 2, but not bad enough to declare the algorithm a failure.

Next we present results for Algorithm 2 with $\beta = 1.0$. The grids used are shown in Table 4.3, and the h_{wm} in Fig. 4.10. The results, shown in Fig. 4.11, show similar characteristics to Algorithm 1. The shared run 0 is too coarse to generate much turbulence, and the velocity starts showing good results from run 1 on, and Reynolds stresses on run 2.

Table 4.3: The grids of the adaptive WMLES runs of equilibrium boundary layers produced by Algorithm 2 with $\beta = 1.0$.

Run	$\Delta x/\delta_0$	$\Delta y_w/\delta_0$	$\Delta z/\delta_0$
0	2.8	0.714	1.5
1	0.219	0.082	0.214
2	0.113	0.042	0.111

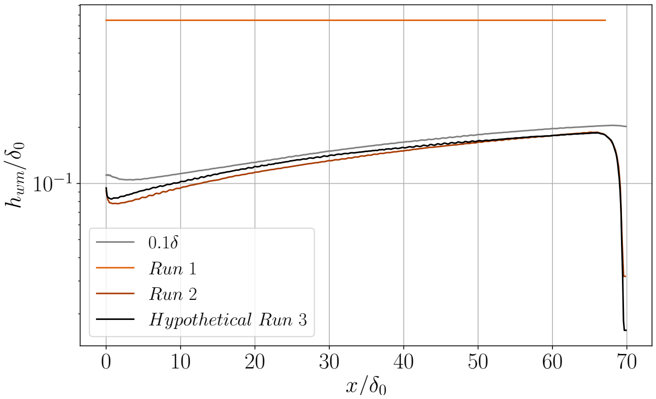

Figure 4.10: The predicted next h_{wm} for each of the adaptive runs of equilibrium flat plate using Algorithm 2 with $\beta = 1.0$.

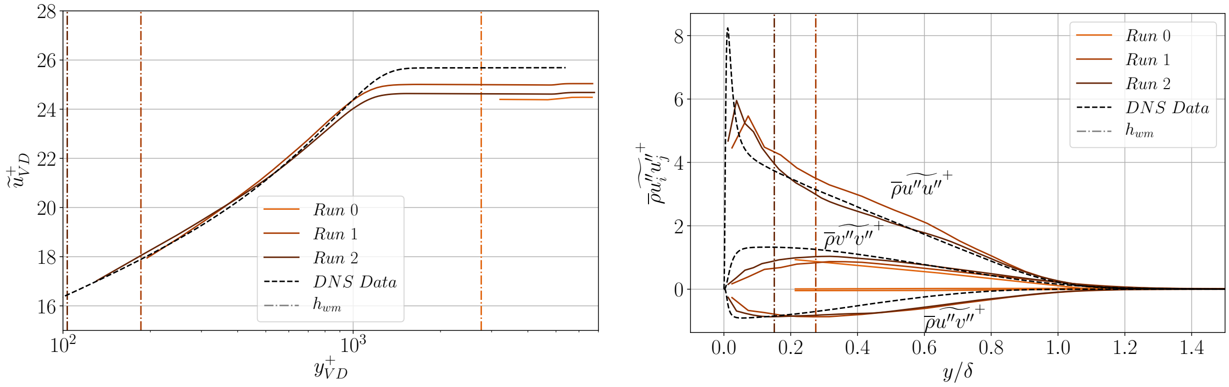

Figure 4.11: *A posteriori* assessment of the adaptive WMLES runs of the equilibrium boundary layer with Algorithm 2 and $\beta = 1.0$, compared to the DNS of Sillero et al. [5, 6]. Van Driest transformed velocity (left) and Reynolds stresses (right).

Chapter 5: Optimization Based Algorithms

This chapter describes our work that is published in AIAA Journal [32]. Our algortihms in the previous chapter were based on understanding of equilibrium wall-models and they ignored the cost that would be required to run a WMLES with the suggested h_{wm}. We attempt to fix both issues in this chapter, where we first describe a way to quantify the error (or rather sensitivity) due to h_{wm} without assuming any specific wall-model. Then, we use an optimization algorithm that equally distributes this error in the domain while adhering to the limits of a user-chosen cost of the next simulation.

5.1 Proposed Methodology

As in the previous chapter, we envision an iterative process where one chooses an initial h_{wm}, builds a suitable computational grid, runs a WMLES, and then uses the results from this initial WMLES run to generate a new h_{wm} field, and so on. Therefore, this h_{wm}-adaptation algorithm is a post-processing operation, operating independently from the flow solver.

The algorithm is based on estimating an upper bound to the global error incurred with a candidate h_{wm} field and then finding the h_{wm} field that minimizes this error bound for a given computational cost. This is accomplished by defining the global objective functional

$$\mathcal{J}(h_{\mathrm{wm}}) = \mathcal{E}(h_{\mathrm{wm}}) + \lambda\mathcal{C}(h_{\mathrm{wm}})\,, \tag{5.1}$$

where $\mathcal{E}$ is the estimated global error bound and $\mathcal{C}$ is the estimated computational cost of solving a WMLES, with λ being a Lagrange multiplier. The problem is then to define the functionals $\mathcal{E}$ and $\mathcal{C}$.

5.1.1 Error Functional $\mathcal{E}(h_{\mathrm{wm}})$

The error in the estimated wall shear stress τ_{w} at a point on the surface is $E_{\mathrm{model}}(h_{\mathrm{wm}}) = \tau_{\mathrm{wm}}(h_{\mathrm{wm}}) - \tau_{\mathrm{w,true}}$, where $\tau_{\mathrm{w,true}}$ is the true value of wall shear stress and $\tau_{\mathrm{wm}}(h_{\mathrm{wm}})$ is the wall shear stress predicted by the wall-model with input data taken at height h_{wm}. This error is shown in Fig. 3.1 for a standard equilibrium wall-model (the one by Kawai and Larsson[24]) with input data taken from DNS of different channels and boundary layers in multiple different studies [1, 2, 3, 4, 5, 6, 7]. Ignoring the small wiggles near the wall (to be discussed below), the figure suggests two desirable conditions for a model of this error: that the error should increase as we move away from the wall, and that it should have a nearly zero first derivative in and below the log-layer (when applied to equilibrium boundary layers). We thus use the simple second-order polynomial

$$\tau_{\mathrm{wm}}(h_{\mathrm{wm}}) - \tau_{\mathrm{w,true}} \approx E_{\mathrm{model}}(h_{\mathrm{wm}}) = Ch_{\mathrm{wm}}^2\,, \tag{5.2}$$

as our error model, where C is a spatially varying coefficient to be determined. With this error model, the absolute value of the pointwise error in τ_{w} can then be integrated over the wall to produce an upper bound on the error in the integrated frictional drag, which we take as our error functional; i.e.,

$$\mathcal{E}(h_{\mathrm{wm}}) = \frac{1}{D_{\mathrm{ref}}} \iint |E_{\mathrm{model}}(h_{\mathrm{wm}})| \, dS = \frac{1}{D_{\mathrm{ref}}} \iint |C| \, h_{\mathrm{wm}}^2 dS \,, \tag{5.3}$$

with D_{ref} as a reference drag for non-dimensionalization reasons.

The remaining question is the determination of the coefficient field C. We propose the use of existing WMLES data from a previous simulation for this, thus creating an iterative, adaptive, method. The idea is to use flow data along wall-normal lines as input to the wall-model to create multiple different estimates $\tau_{\mathrm{wm}}(h_{\mathrm{wm}})$ for multiple different h_{wm} values, and then find C from a least-squares solution to Eqn. (5.2). In practice, we extract mean flow data and the wall-distance h_{wm} from multiple grid points in the existing grid, and then feed this into the wall-model equations. For every chosen input location, we then get a resulting predicted τ_{wm} for the corresponding wall-distance h_{wm}. We then solve Eqn. (5.2) using least squares for the unknown C and $\tau_{\mathrm{w,true}}$.

It is critically important to include the right data points in this method, to ensure a reasonable estimate of C. We use data from grid points located at wall distances between $y_{\mathrm{data,min}}$ and $y_{\mathrm{data,max}}$, and choose those limits as follows. Data from the first few grid points should not be used since the WMLES is unavoidably under-resolved there. We thus define $y_{\mathrm{data,min}}$ as the smallest height above the wall for which the prior WMLES is expected to be sufficiently well resolved; following the discussion around the sufficient grid spacing of Chapter 3, the general recommendation is therefore $y_{\mathrm{data,min}} = \max\{\Delta x/K_1, \Delta y/K_2, \Delta z/K_3\}$. In the present work, we simply used the h_{wm} from the prior simulation since that used a grid designed to satisfy these criteria.

The choice for the upper bound $y_{\text{data,max}}$ is non-trivial due to inherent imperfections in the wall-model. In an ideal world, an equilibrium wall-model should be accurate for exchange locations $h_{\text{wm}}/\delta \lesssim 0.1-0.2$ in an equilibrium boundary layer; in the context of the present error model, this would imply that the least-squares fit should produce small values of C for $y_{\text{data,max}}/\delta \approx 0.1-0.2$. Figure 3.1, however, shows that this would not be the case, and in fact the least-squares procedure would not robustly return $C \approx 0$ unless $y_{\text{data,max}}$ was near the viscous sublayer. The reason is that the wall-model has imperfections in the buffer layer which creates the small positive bumps near the wall in the figure. This creates a problem with the current algorithm: if computing C from sufficiently resolved flow data, the error model would represent the error in the buffer layer and thus push the h_{wm}-adaptation algorithm towards a wall-resolved LES. This is clearly not desired: we rather want the error model to represent the larger features of the profiles in Fig. 3.1, i.e., the error for $0.1 \lesssim h_{\text{wm}}/\delta \lesssim 0.6$ or so. This can be accomplished by choosing $y_{\text{data,max}}$ sufficiently large, in which case the least-squares procedure will attempt to fit the error model (5.2) to the larger features in Fig. 3.1. We therefore define

$$y_{\text{data,max}} = 1.5 \max\left\{ Y : \left| \frac{\tau_{\text{wm}}(y) - \tau_{\text{wm,med}}(y_{\text{data,min}}, y)}{\tau_{\text{wm,med}}(y_{\text{data,min}}, y)} \right| \leq \epsilon \,, \forall y \leq Y \right\} \,, \tag{5.4}$$

where ϵ is an error tolerance and $\tau_{\text{wm,med}}(y_{\text{data,min}}, y)$ is the median value of the predicted τ_{wm} for input data taken from locations between $y_{\text{data,min}}$ and y. The concept is illustrated in Fig. 5.1. For underresolved data (very coarse prior WMLES), the procedure results in using few data points in the least-squares process, and (for the data in the figure) a large C. For overresolved data (e.g., if the prior WMLES was excessively fine in some region), the procedure results in using a large

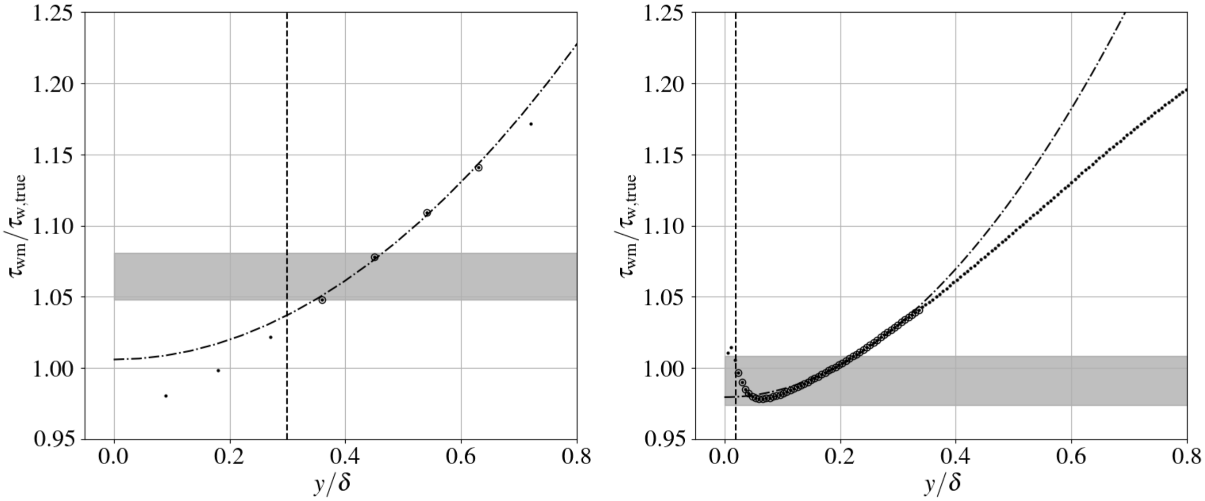

Figure 5.1: Illustration of how to choose which data points to include in the least-squares problem for determining C, for underresolved (left) and overresolved (right) prior WMLES. Black dots ($\cdot$) show each data point, black circles ($\circ$) show points included in the least-squares fit, the dash-dotted line ($-\cdot$) shows the curve fit, and the dashed line ($--$) marks the lower bound $y_{\mathrm{data,min}}$ below which the prior WMLES is inaccurate and should not be trusted. The gray regions show the acceptable region defined by $\epsilon = 0.015$. Data from unpublished work by Ali Uzun [10].

number of data points and thus a much smaller resulting C than if only the first few points were used.

The error tolerance ϵ should be taken as representative of the unavoidable errors in the buffer layer when applied to equilibrium flows. For the incompressible cases in Fig. 3.1, $\epsilon \approx 0.03$ is reasonable. The single supersonic case in the figure requires a larger error tolerance, reflective of the additional errors in compressible wall-models. The interested reader should refer to Iyer and Malik [51] who performed a comprehensive assessment of wall-modeling errors in compressible flows with adiabatic and non-adiabatic walls.

5.1.2 Cost Functional $\mathcal{C}(h_{\mathrm{wm}})$

The total cost of an LES can be modeled as

$$\mathrm{Cost} = \mathcal{N}_{\mathrm{time\,steps}}\mathcal{N}_{\mathrm{cells}} \approx \iiint\limits_{\mathrm{V}} \frac{\boldsymbol{T}}{\Delta x \Delta y \Delta z \Delta t} dV \,, \tag{5.5}$$

where $\mathcal{N}_{\mathrm{time\,steps}}$ is the number of time steps, $\mathcal{N}_{\mathrm{cells}}$ is the number of cells, $\Delta x, \Delta y, \Delta z$ are the grid-spacings, Δt is the time step, V is the domain of the simulation and $\boldsymbol{T}$ is the total simulation time. In order to derive a cost functional for Eqn. (5.1), we need to relate the grid-spacings and time step to the h_{wm} field.

The grid has to be sufficiently fine to resolve the outer layer turbulence, and the turbulence around the "exchange location" as discussed in chapter 3, which of these is the limiting factor depends on the value of h_{wm}/δ. We formulate our cost functional for the limit of small h_{wm}/δ, i.e., to be valid at the end of the adaptation sequence, by taking the grid-spacings as $\Delta x = K_1 h_{\mathrm{wm}}$, $\Delta z = K_3 h_{\mathrm{wm}}$ and $\Delta y = K_2$, with values for these constants given in the discussion in chapter 3. These grid-spacings are meaningful only between the wall and the modeling interface; the grid in the remaining domain is determined by LES resolution requirements independently of h_{wm}. This implies that our cost functional $\mathcal{C}(h_{\mathrm{wm}})$ should only be sensitive to the computational cost of the near-wall region. In addition, the connection between Δy and h_{wm} implies that the number of cells in the wall-normal direction remains constant even as h_{wm} changes. Our cost functional should therefore scale as

$$\mathcal{C} \sim \iint \frac{\boldsymbol{T}}{K_1 K_3 h_{\mathrm{wm}}^2 \Delta t} dS \,.$$

which is a surface integral over all walls.

The time step is limited by the smallest cell in an explicit time stepping scheme. For the purpose of roughly determining the cost of a new simulation with a different h_{wm}, we suggest the simplistic model $\boldsymbol{T}/\Delta t \sim (L_{\mathrm{ref}}/h_{\mathrm{wm}})^{\alpha}$ with L_{ref} as some reference length. A value of $\alpha = 0$ implies that h_{wm} does not affect the time step, a value of $\alpha = 1$ implies that each cell uses a local time step. Neither is plausible in practical scenarios (particularly the latter); we use it here merely to assess the effect of (approximately) accounting for the time step. Our final cost functional, omitting all constants, is therefore

$$\mathcal{C}(h_{\mathrm{wm}}) = L_{\mathrm{ref}}^{\alpha} \iint \frac{1}{h_{\mathrm{wm}}^{(\alpha+2)}} dS \,. \tag{5.6}$$

5.1.3 Solution to the Optimization Problem

Inserting the error and cost functionals from Eqns. (5.3) and (5.6) into (5.1) yields the final minimization problem

$$\mathcal{J}(h_{\mathrm{wm}}) = \iint \left(\frac{|C| \, h_{\mathrm{wm}}^2}{D_{\mathrm{ref}}} + \lambda \frac{L_{\mathrm{ref}}^{\alpha}}{h_{\mathrm{wm}}^{(\alpha+2)}} \right) dS \,,$$

with the integrals extending over the walls. The solution to this optimization problem is given by the Euler-Lagrange equation

$$\frac{\partial}{\partial h_{\mathrm{wm}}} \left(\frac{|C| \, h_{\mathrm{wm}}^2}{D_{\mathrm{ref}}} + \lambda \frac{L_{\mathrm{ref}}^{\alpha}}{h_{\mathrm{wm}}^{(\alpha+2)}} \right) = 0 \,,$$

which produces the optimal h_{wm} field as

$$h_{\mathrm{wm}} = \left(\underbrace{\lambda D_{\mathrm{ref}} L_{\mathrm{ref}}^{\alpha}}_{\hat{\lambda}} \frac{2+\alpha}{2\,|C|} \right)^{\dfrac{1}{4+\alpha}} . \tag{5.7}$$

In practice, this is solved by finding the $|C|$ field over the wall, and then solving a root-finding problem for the Lagrange multiplier λ until the cost functional takes on the desired value (equivalently, for $\alpha = 0$, until the number of cells on the wall is as desired). In addition, due to averaging errors and noise in the data, some form of low-pass filtering applied to either $|C|$ or the resulting h_{wm} is usually needed. We have used a Gaussian filter with width $\sim \mathcal{O}(\delta)$ or smaller in the present study. We finally note that the reference values for drag and length are actually not needed: it is simpler to solve directly for the dimensional Lagrange multiplier $\hat{\lambda}$.

5.1.4 Simulation Process and Computational Cost

The algorithm described here is entirely a post-processing operation, done off-line using averaged flow data from a prior WMLES run. The use of averaged flow data makes for a simpler method and incurs an error of at most 2-3% [50].

The algorithm thus starts with averaged flow data from a prior WMLES run. The value of h_{wm} used in that prior simulation is irrelevant to the algorithm. Step 1 is then to find C for every grid point (or cell face) on the wall, which is done by solving multiple wall-modeling problems where data is taken from different wall-distances. This is the most expensive part of the algorithm, with a cost roughly similar to that of a single time step of the WMLES solver. Step 2 is find the new h_{wm} field from the optimization problem. This is done by guessing an initial value of the Lagrange multiplier λ, finding h_{wm} from Eqn. (5.7), and computing the resulting cost from Eqn. (5.6). The value of λ is then adjusted using a root-finding technique (bisection method in this work) until the cost is as desired (generally quadrupled cost compared to the prior run in this work).

Increasing the desired cost of each WMLES run ensures that the added cost of running multiple simulations is limited: e.g., quadrupling the cost at each iteration ($\approx$ 40% finer grid in all directions assuming the time step also changes) means that the cumulative cost of all coarser grids is only 1/3 of the cost on the finest grid. In addition, running on multiple grids allows for grid sensitivity assessment.

5.2 *A priori* Analysis: the NASA Wall-Mounted Hump

The algorithm is tested first using the WRLES data of Uzun [10] for the flow over a wall-mounted hump [55]. This data set is the same unpublished dataset as the one in Chapter 4, it has higher grid resolution than the published dataset [9]. The geometry is shown in Fig. 3.3. The incoming boundary layer is fully developed (essentially self-similar, having lost memory of the inflow condition) at $Re_\tau \approx 2000$ and Mach 0.1. A secondary, much thinner, boundary layer forms from the beginning of the hump, which makes the decision of where to place the wall-modeling exchange location challenging.

The *a priori* test is done by choosing a uniform initial h_{wm} that is excessively large, in fact twice larger than the incoming boundary layer thickness. We then use WRLES data from above this initial h_{wm} height only and apply the algorithm; this is meant to mimic a situation where a prior WMLES had been run with this particular h_{wm} distribution. The algorithm produces a new h_{wm} distribution, this h_{wm} is filtered along streamwise direction with a Gaussian filter with a width of $\sim 0.25\delta_0$ for the incoming boundary layer and the reattached region, and with a width of $\sim 0.1\delta_0$ on the hump, and the process is repeated. At each step, we choose a desired increase in the computational cost (as measured by the estimated number of faces on the wall) and find the corresponding Lagrange multiplier λ by root-finding.

The whole process is computationally very cheap. We coded it in $Python$ and ran it on a laptop computer with a third generation Intel i7 CPU, and the whole process of finding the next h_{wm} field took between 63 and 114 seconds. These times are for the *a priori* analysis using the WRLES data with roughly 6000 grid points on the wall; the cost is proportionally smaller in actual WMLES cases with fewer wall points. One could also solve only at some wall points and use interpolation.

5.2.1 Basic Assessment

The algorithm is first analyzed without accounting for the computational cost of the time step (i.e., taking $\alpha = 0$) and with the computational cost being increased by 4 in each iteration. The resulting h_{wm} and τ_{wm} fields are shown in Fig. 5.2. The gradual increase in the cost is clearly seen at all locations, as with each iteration h_{wm} gets smaller. Recall that we expect the equilibrium wall-model to work for $h_{\mathrm{wm}} \approx 0.1\delta$ in the incoming boundary layer, but the algorithm keeps refining it well past that point. While this at first appears less than ideal, we will argue below that it is actually a positive thing.

Since we started with an excessively large h_{wm}, it takes a few iterations of the algorithm before the h_{wm} is well within the boundary layer and thus that WMLES starts to make sense.

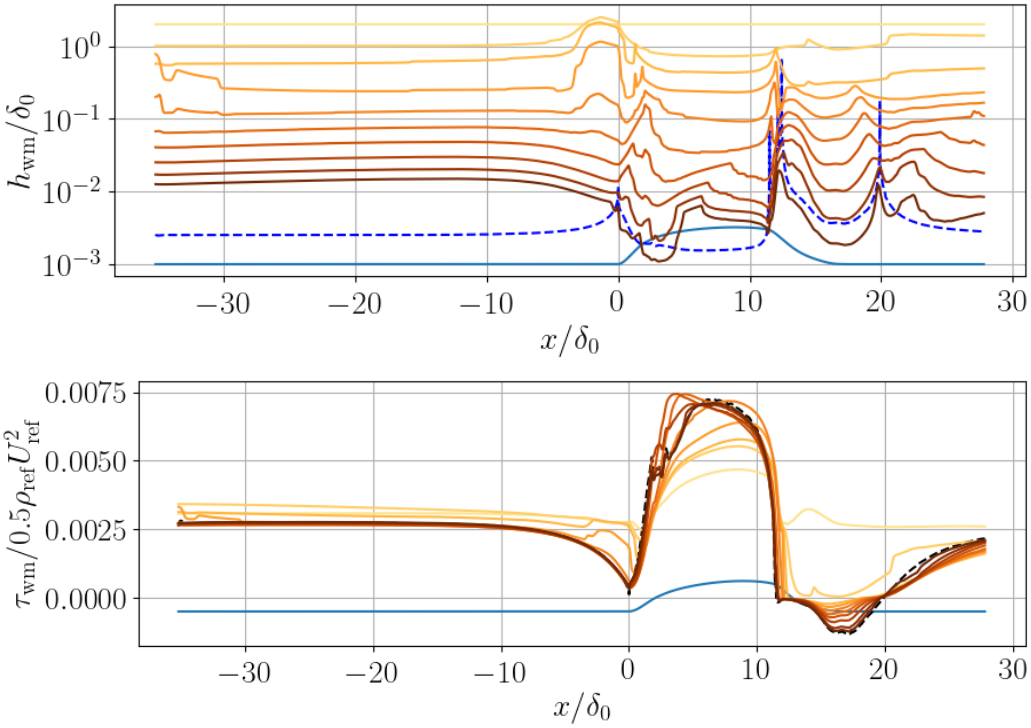

Figure 5.2: *A priori* results for the NASA wall-mounted hump using WRLES data from Uzun [10], showing a sequence of iterations of the algorithm going from light to dark colors. Each iteration corresponds to approximately 4 times higher requested computational cost. In the top figure, the dashed line is $y^+ = 5$. In the bottom figure, the dashed line is the WRLES truth.

From that point on, h_{wm} is always significantly smaller over the hump than in the incoming boundary layer, by factors ranging from 2-4. This is (in a qualitative sense) the expected result because the non-equilibrium flow around the hump should give rise to higher errors. Above the hump are two confluent boundary layers, one from the incoming flow with a thickness $\approx \delta_0$ and one much thinner secondary boundary layer with a thickness $\approx 0.1\delta_0$. The identified h_{wm} is therefore 2-4 times thinner over the hump in absolute terms but thicker in relative terms, when comparing to the secondary boundary layer. The h_{wm} then rises around the separation point since the magnitude of the error in the wall shear stress is smaller there. A similar rise occurs the reattachment point, for the same reason.

The result in the region immediately after reattachment is particularly interesting. In the early and middle iterations, the h_{wm} in this region is approximately equal to the value in the incoming boundary layer, despite the flow quite clearly being out of equilibrium. The reason can be found in Fig. 3.3 (gray line), which shows how the error is essentially flat for $0.05 < h_{\mathrm{wm}}/\delta_0 < 0.4$, which tricks the algorithm into thinking that it has converged to a correct h_{wm}. This behavior is cured in the later iterations, where the error model starts to pick up the true behavior and the algorithm then produces lower h_{wm} than the incoming boundary layer.

These observations about the region immediately after reattachment make us conclude that the tendency of the algorithm to produce ever lower h_{wm} values as the requested computational cost is increased is ultimately a feature, not a bug. While one could modify the algorithm to "stop" at $h_{\mathrm{wm}} \approx 0.1\delta$, this would be appropriate only for fully developed equilibrium regions. The behavior in the region behind reattachment shows how difficult it can be to distinguish an equilibrium region from a still developing one based only on data located far from the wall (i.e., assuming that the data comes from a prior WMLES).

We next analyze how the algorithm allocates computational resources to the different parts of the domain. Figure 5.3 shows the cost density, defined as $\iint_{x,\text{region}} (1/h_{\text{wm}}^2) dS/(L_z L_{x,\text{region}})$, for the following regions of the domain: the incoming boundary layer (x/δ_0 from -30 to -10), before and over the hump (x/δ_0 from -10 to 11.5), and the separation and reattachment region (x/δ_0 from 11.5 to the end). The cost for each region is normalized by the cost for the incoming boundary layer. It takes about 4 iterations (starting from the right in the figure) before the algorithm starts distinguishing between the flow regions, after which the algorithm starts allocating ever more of the cost to the regions of strongly non-equilibrium flow. Note that the change occurs around the point where $h_{\text{wm}} \approx 0.1\delta_0$ in the incoming boundary layer, which is where WMLES experience suggests that the wall-model becomes highly accurate in that region. Also note that the algorithm starts allocating more resources to the region around the hump before it starts increasing the resources to the separation region. At the last iteration, the algorithm allocates more resources to the separation region than the hump. Going back to Fig. 5.2, note that the h_{wm} has reached into the viscous sublayer at the front of the hump, which is consistent with a reduction in the addition of more resources.

The basic goal of a wall-model is to estimate the wall shear stress, so to this end the predicted skin friction for the sequence of iterations is shown in Fig. 5.2. The results are obtained by feeding WRLES data at the given h_{wm} to the wall-model, similar to what was done above. It can be seen that by each iteration, the τ_{wm} in the incoming boundary layer gets closer and closer to the WRLES τ_{w}, and reaches it around iteration 4. On the hump, the τ_{wm} starts from underpredicted values, becomes correct halfway through the sequence, but then keeps on rising to go overpredicted, only to come back to the correct value at the end. This happens only because there is a sweet spot for that specific region of the flow that gets the τ_{w} right by pure luck. True conver-

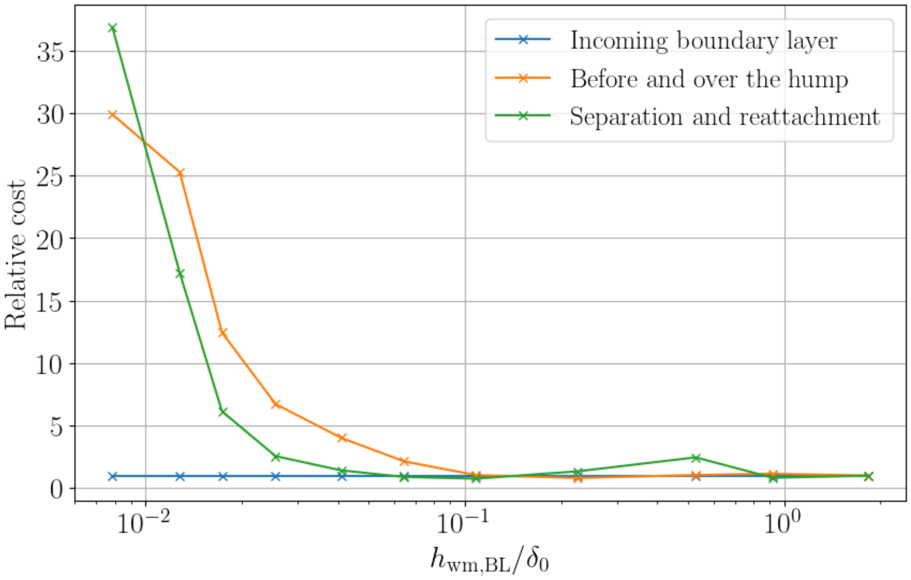

Figure 5.3: *A priori* results for the NASA wall-mounted hump using WRLES data from Uzun [10], showing the approximate relative computational cost for different regions, plotted versus h_{wm} in the incoming boundary layer. The relative cost is defined as $\mathcal{C}(\mathrm{region})/\mathcal{C}(\mathrm{incoming\,BL})$, where the cost of each region is $\mathcal{C}(\mathrm{region}) = \iint_{x,\mathrm{region}}(1/h_{\mathrm{wm}}^2)dS/(L_z L_{x,\mathrm{region}})$. Each iteration corresponds to approximately 4 times higher requested computational cost.

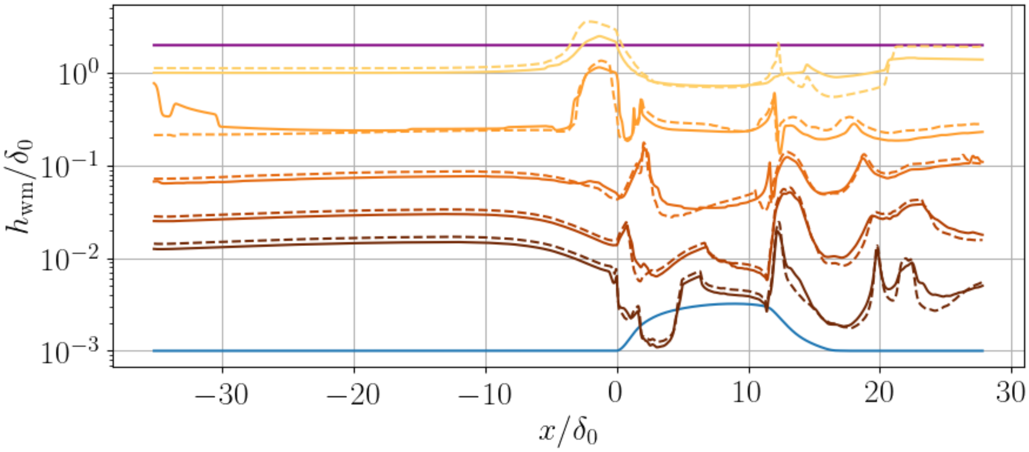

Figure 5.4: *A priori* results for the NASA wall-mounted hump using WRLES data from Uzun [10], showing the h_{wm} profiles created by increasing the cost by factors of 4 (solid, showing every other iteration) and 2 (dashed, showing every fourth iteration) in each iteration. Purple is the initial h_{wm}, afterwards darker color means later iterations.

gence is achieved beyond that point. The convergence is somewhat monotonic in the separated and reattaching regions.

5.2.2 Path Independence

We next demonstrate the path independence of the algorithm. Since the algorithm requires the use of imperfect WMLES data from the previous iteration, it may be path dependent. The tests above used the algorithm with 4 times increase in the computational cost between iterations. We now apply the algorithm starting from the same initial (and very large) h_{wm}, but increasing the cost by a factor of 2 between iterations. The results are shown in Fig. 5.4. There are minor differences in the results, but the algorithm looks path independent in general.

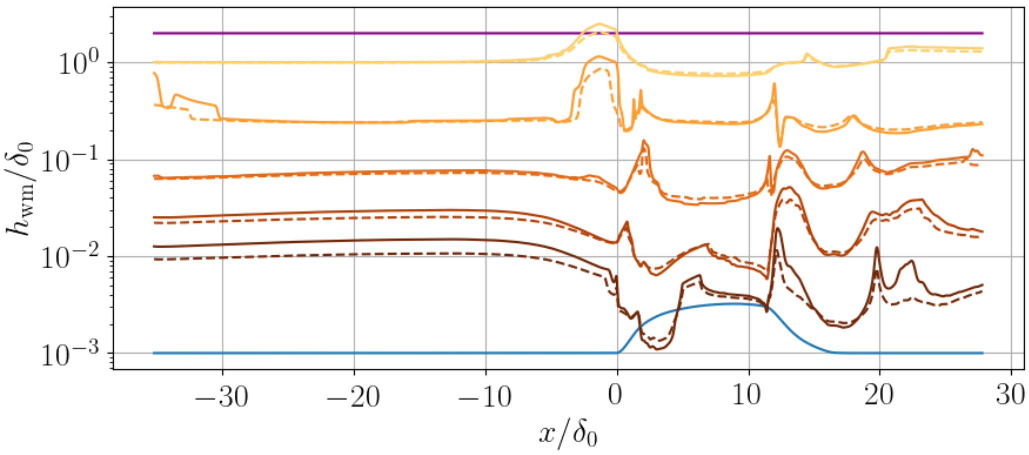

Figure 5.5: *A priori* results for the NASA wall-mounted hump using WRLES data from Uzun [10], showing the h_{wm} profiles created with $\alpha = 0$ (solid) and $\alpha = 1$ (dashed). Purple is the initial h_{wm}, afterwards darker color means further iterations.

5.2.3 Accounting for the Computational Time Step

The prior tests were run with $\alpha = 0$, meaning that the contribution of the time step to the estimated cost of the WMLES was neglected. We next test the algorithm with $\alpha = 1$, which would be the case for a perfect code where each cell moves with a perfectly local time step. The test is run by increasing the estimated number of wall-adjacent cells by a factor of 4 in each iteration in order to enable a direct comparison with the results for $\alpha = 0$. The results are shown in Fig. 5.5. There is not much difference between the two cases, except that $\alpha = 1$ leads to slightly less variability across the different flow regions due to the smaller exponent in Eqn. (5.7).

Neither value of α is realistic in a compressible code with explicit time stepping. Rather than viewing one of these cases as the correct choice, we instead use the difference in the resulting h_{wm} profiles as an estimate of how the results would change were one to accurately account for the time step.

5.2.4 The Effect of the Error Tolerance

We finally investigate the effect of the error tolerance ϵ in Eqn. (5.4) when deciding which data points to perform the curve fit over. Tolerances of ϵ =0.005, 0.025, and 0.05 were used, with results shown in Fig. 5.6.

There are no significant differences between the curves for the earlier iterations (larger h_{wm}), however, the smallest tolerance produces a different resulting h_{wm} in the final iteration (largest cost, lowest h_{wm}). Specifically, the h_{wm} in the incoming boundary layer is close to half the value for the other tolerances. This is the result of the error tolerance being too low, thus failing to account for the inherent small errors of the wall-model in the buffer layer, originally discussed in subsection 5.1.1.

Another point to note is that we had to start the process for the largest error tolerance $\epsilon = 0.05$ at a lower initial h_{wm} (higher desired cost) to avoid degenerate results in the incoming boundary layer. In the flow region right before the hump, the τ_{w} predicted by the wall-model is within ± 0.05 for all $h_{\mathrm{wm}} \geq 2\delta_0$, and thus the algorithm returns $C = 0$ in that region (if $\epsilon = 0.05$). The problem was avoided when starting the test with an initial $h_{\mathrm{wm}} = 0.5\delta_0$. The lesson is that one should avoid choosing ϵ too large and, more importantly, one must ensure (perhaps *a posteriori*) that the identified exchange location is within the boundary layer.

5.3 *A posteriori* Analysis: Adverse Pressure Gradient Boundary Layer

The first *a posteriori* test case is a boundary layer growing under an adverse pressure gradient. The flow is sketched in Fig. 5.7. The adverse pressure gradient is created by a diverging inviscid wall which is implemented using a simple immersed boundary method. The flow set-up

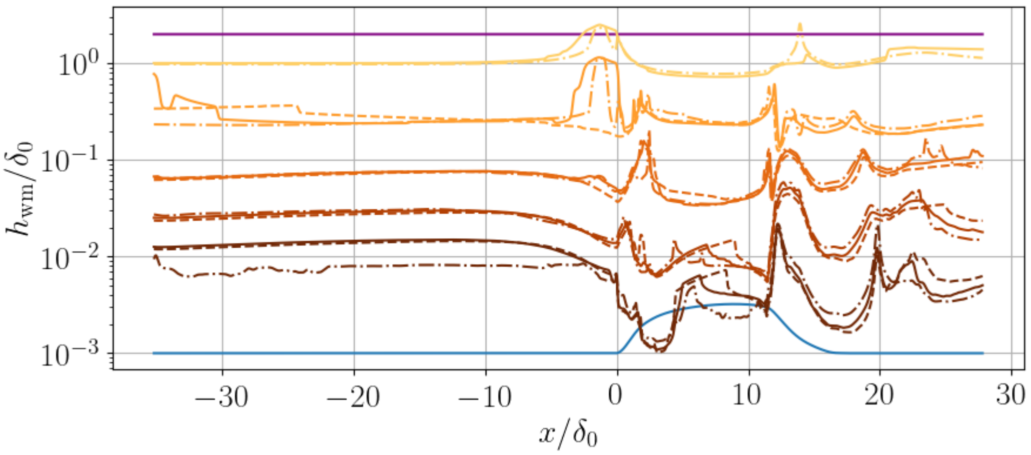

Figure 5.6: *A priori* results for the NASA wall-mounted hump using WRLES data from Uzun [10], showing the h_{wm} profiles created with $\epsilon = 0.025$ (solid), $\epsilon = 0.005$ (dash-dotted) and $\epsilon = 0.05$ (dashed). Purple is the initial h_{wm} for solid and dashed lines, afterwards darker color means further iterations.

and the shape of the upper wall is inspired by the work of Skare and Krogstad [56], but does not reproduce it perfectly. The shape of the diverging inviscid upper wall used here is defined by

$$\frac{y_{\mathrm{upper}}}{\delta_{\mathrm{ref}}} = \left(6.608^4 + \left[\sum_{n=0}^{4} a_n \left(\frac{x}{\delta_{\mathrm{ref}}}\right)^n\right]^4\right)^{1/4} \qquad \begin{aligned} &a_0 = -0.504\,,\ a_1 = 0.518\,,\ a_2 = -8.80\cdot 10^{-3}\,,\\ &a_3 = 8.53\cdot 10^{-5}\,,\ a_4 = -3.47\cdot 10^{-7}\,, \end{aligned} \tag{5.8}$$

where δ_{ref} is the boundary layer thickness at $x/\delta_{\mathrm{ref}} \approx 8$. The resulting flow has a Clauser parameter that increases from 0 to about 20 by $x/\delta_{\mathrm{ref}} \approx 40$ and then stays within 20-30 up to $x/\delta_{\mathrm{ref}} \approx 80$. Some values are listed in Table 5.1. The Clauser parameter is slightly higher than in the experiment, likely due to the inviscid rather than viscous upper wall.

The in-house finite-difference solver *Hybrid* is used, whose characteristics were explained in chapter 2.

The very high Reynolds number makes DNS or WRLES unaffordable, and we therefore in-

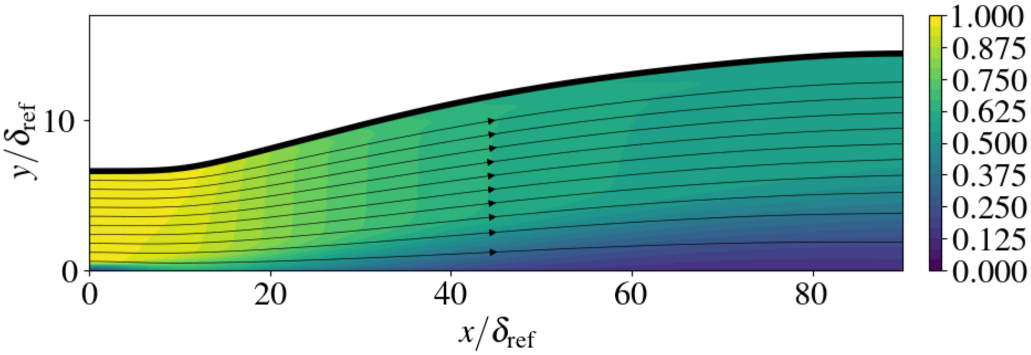

Figure 5.7: Contours of the mean streamwise velocity and streamlines for the adverse pressure gradient case.

stead use a well resolved WMLES as our reference case. This case uses a uniform $h_{\mathrm{wm}} = 0.09\delta_{\mathrm{ref}}$ and a grid with $\Delta x = \Delta z = 0.06\delta_{\mathrm{ref}}$ and $\Delta y_{\mathrm{wall}} = 0.03\delta_{\mathrm{ref}}$. The wall-normal grid spacing Δy is stretched such that it supports at least 20 points within the boundary layer everywhere in the domain. Since the boundary layer is growing monotonically and since the streamwise pressure gradient is moderate, this WMLES should therefore be rather accurate. The spanwise domain width is $20\delta_{\mathrm{ref}}$. The boundary layer thickness δ is computed using a simplified version of the method by Vinuesa et al. [57] based on the local turbulence intensity, with δ defined implicitly as

$$\left(\frac{\widetilde{u''u''}(\delta) + \widetilde{v''v''}(\delta) + \widetilde{w''w''}(\delta)}{3}\right)^{1/2} \frac{1}{U(\delta)} = 0.04\,.$$

The local edge velocity is then defined as $U_{\mathrm{e}} = U(\delta)/0.99$, and the global reference velocity U_{ref} is taken as the value of U_{e} at $x/\delta_{\mathrm{ref}} \approx 8$. Note that the reference quantities δ_{ref} and U_{ref} do not carry any inherent physical meaning, as the incoming boundary layer is not fully realistic at that location due to the imperfect synthetic turbulence at the inflow. The Mach number at the reference location is 0.28. Some key quantities describing the boundary layer state are given in

x/δ_{ref}	$y_{\text{upper}}/\delta_{\text{ref}}$	$\delta/\delta_{\text{ref}}$	$\delta^*/\delta_{\text{ref}}$	$\theta/\delta_{\text{ref}}$	Re_θ	β
30	9.67	2.45	0.6	0.392	29500	10.5
40	11.1	3.26	0.998	0.577	39400	18.2
50	12.2	4.08	1.45	0.751	48300	23.3
60	13.1	4.8	1.87	0.898	55500	26.7
70	13.7	5.45	2.23	1.02	61600	26.8
80	14.2	5.97	2.47	1.13	66400	20.8

Table 5.1: Key quantities at different streamwise locations in the adverse pressure gradient case. Note that $Re_\theta = \rho_\infty U_e \theta / \mu_\infty$ and that the Clauser parameter $\beta = \delta^*/\tau_w \, \partial p/\partial x$.

Table 5.1.

The initial h_{wm} field is taken as uniform at $2\delta_{\text{ref}}$; this is outside the boundary layer for $x/\delta_{\text{ref}} \lesssim 25$ and greater than $\delta/3$ throughout the domain, and thus constitutes a highly underresolved situation that tests the robustness of the algorithm. We run a WMLES, collect statistics, and then apply the h_{wm}-algorithm with a desired cost increase of a factor of 4 to find the next h_{wm} profile. This h_{wm} profile is then low-pass filtered (using a Gaussian filter with width $10\Delta x$) to remove noise.

Once the h_{wm} profile is fully defined, a computational grid that supports this h_{wm} is generated. While fully unstructured grids are required to take full advantage of the variable h_{wm} profile, we use structured Cartesian grids here to simplify the interpretation of the results. These grids are generated to satisfy the standard resolution criteria [14, 50] of $\Delta x, \Delta z \leq 0.8 h_{\text{wm}}$ and $\Delta y_w \leq 0.3 h_{\text{wm}}$. The grid is stretched in the streamwise direction with a maximum stretching ratio of 0.5%, and gently stretched in the vertical direction. The Δx variation in the domain is shown in Fig. 5.8.

The resulting sequence of h_{wm} profiles is shown in Fig. 5.8 along with the resulting wall shear stress. We first note that the local h_{wm}/δ ratio stays above 0.2 for the first two iterations,

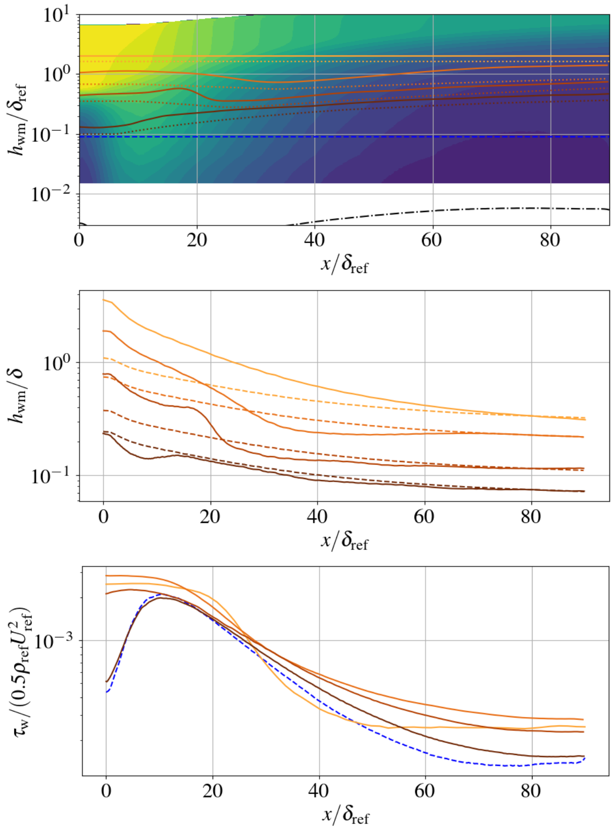

Figure 5.8: Sequence of adapted h_{wm} (top figure, solid lines) and Δx (top, dotted) for the adverse pressure gradient case, with darker colors meaning later iterations, with h_{wm} of the reference WMLES being shown as a dashed blue line and $y^+ = 5$ being shown as a dash-dotted black line. Also showing the local h_{wm}/δ ratio (middle, with $h_{\mathrm{wm}} \propto \sqrt{\delta}$ in dashed lines) and the skin friction (bottom, with the reference WMLES as dashed blue line).

and is above 1 in parts of the domain for the first three iterations. Therefore, the fourth (and final) iteration is the first to represent a meaningful WMLES. This last iteration produces a τ_{w} that is rather close to the reference, albeit somewhat overpredicted towards the outflow. Note that the increase in τ_{w} near the inflow is partly caused by the acceleration there and partly caused by the imperfect synthetic inflow turbulence. The mean velocity, Reynolds stresses and turbulent energy spectrum are shown in Fig. 5.9, with all of them showing convergence to the reference profiles.

The energy spectrum in the middle of the boundary layer converges by filling out at the smaller scales, as one would expect in LES.

The perhaps most interesting observation is that the h_{wm}/δ ratio keeps decreasing throughout the domain, which seemingly contradicts the current understanding/assumption that the correct exchange location should be a fixed fraction of the boundary layer thickness in self-similar boundary layers (the present case is close to self-similar, see Skare and Krogstad [56]). The discrepancy between the results of the h_{wm}-algorithm and the common thinking is caused by the fact that the latter disregards the computational cost. To see this, assume that the relative error in the wall-model scales with the h_{wm}/δ ratio; our assumed error model then yields $E(h_{\mathrm{wm}}) \sim \tau_{\mathrm{w}}(h_{\mathrm{wm}}/\delta)^2$ and thus $|C| \sim \tau_{\mathrm{w}}/\delta^2$. The solution to the optimization problem (Eqn. 5.7 with $\alpha = 0$ here) is then $h_{\mathrm{wm}} \sim \tau_{\mathrm{w}}^{-1/4}\sqrt{\delta}$. If τ_{w} changes slowly (like in a zero-pressure-gradient boundary layer), the algorithm will then produce an exchange location that moves away from the wall as the boundary layer grows, but less rapidly than the boundary layer itself. In the present case the wall shear stress does vary significantly, but the general trend that h_{wm} grows at a rate slower than the boundary layer itself remains true. This can be seen in Fig. 5.8, where the h_{wm} field in the final iteration stays quite close to the predicted $\sqrt{\delta}$ behavior.

We note that the $h_{\mathrm{wm}} \sim \tau_{\mathrm{w}}^{-1/4}\sqrt{\delta}$ behavior predicted here is for the specific error model used here, and thus should not be viewed as an absolute truth; the results should be interpreted in the more qualitative sense of suggesting that the optimal exchange location must account for both the error and the computational cost. The optimization balances the marginal changes in the error and cost, and this marginal balance is altered when the boundary layer is growing in a way that produces a smaller h_{wm}/δ for a thicker boundary layer; simplistically, reducing h_{wm}/δ costs little in a thicker boundary layer.

5.4 *A posteriori* Analysis: Shock/Boundary-Layer Interaction

The penultimate test case is a shock/boundary-layer interaction where an oblique shock wave impinges upon a Mach 2.3 turbulent boundary layer, creating a separation bubble and a strongly non-equilibrium flow both in the interaction region and in the recovering boundary layer behind the interaction. A representative flow visualization is shown in Fig. 5.10. The separation bubble is quite long, about 7 times the incoming boundary layer thickness δ_{ref}, but also very shallow (the highest point of flow reversal occurs at $y/\delta_{\text{ref}} \approx 0.3$; for most of the bubble this height is much lower) making it challenging to accurately capture with WMLES. The wall is adiabatic and the incoming boundary layer is fully developed (i.e., having essentially lost memory of the inflow condition) at $Re_{\delta 2} \approx 2200$.

The reference data was computed by WRLES using the same *Hybrid* flow solver as described above. A computational domain of size $L_x \times L_y \times L_z = 107\delta_{\text{in}} \times 40\delta_{\text{in}} \times 8\delta_{\text{in}}$ was used for the reference WRLES, where δ_{in} is the boundary layer thickness at the inflow. The WMLES is computed in a slightly smaller domain of size $L_x \times L_y \times L_z = 90\delta_{\text{in}} \times 40\delta_{\text{in}} \times 6\delta_{\text{in}}$ since we are less interested in the flow development far downstream in the present WMLES computations. The inviscid impingement location of the incoming shock is $x_{\text{imp}} = 50\delta_{\text{in}}$. The reference boundary layer properties are taken from $x_{\text{ref}} = 34\delta_{\text{in}}$, where the boundary layer thickness $\delta_{\text{ref}} \approx 1.65\delta_{\text{in}}$. The incoming shock angle is $35.0°$, which corresponds to a flow deflection angle of $\phi = 10.4°$.

The grid-spacing for the reference WRLES is uniform in x with $\Delta x = \Delta z \approx 0.013\delta_{\text{ref}}$. At the reference location, this corresponds to $\Delta x^+ = \Delta z^+ \approx 9$, $y_1^+ \approx 0.6$, and $\Delta y_e/\delta_{\text{ref}} \approx 0.017$. Behind the shock, the values in inner scaling change to $\Delta x^+ = \Delta z^+ \approx 13$ and $y_1^+ \approx 0.9$ due to the decrease of the viscous length scale across the shock. The grid is thus close to DNS-level, at least before the shock, and hence this WRLES is computed without any explicit subgrid model. A more complete description including assessment of grid convergence can be found in Larsson et al. [58].

The h_{wm}-adaptation algorithm is tested in a similar way as for the previous test case, by starting with a uniform h_{wm} that is unrealistically large, $\approx 2\delta_{\mathrm{ref}}$, placing the wall-model exchange location outside the boundary layer and above the separation bubble. At each iteration we then seek the h_{wm} field that produces approximately four times higher computational cost. Each resulting h_{wm} profile is low-pass filtered to remove noise. In addition, we set the h_{wm} to be uniform for $(x - x_{\mathrm{imp}}) < -17\delta_{\mathrm{ref}}$ and $(x - x_{\mathrm{imp}}) > 18\delta_{\mathrm{ref}}$ to reduce the effect of imperfect boundary conditions on the assessment of the algorithm (this is not necessary, it is done here only to make the assessment maximally clear). Once the h_{wm} field is known, either from the initial guess or after solving the adaptation problem, a grid is generated that satisfies $\Delta x, \Delta z \leq 0.8h_{\mathrm{wm}}$ and $\Delta y_{\mathrm{wall}} \leq 0.3h_{\mathrm{wm}}$.

The sequence of h_{wm}, streamwise grid-spacings Δx, and resulting skin friction coefficient c_{f} are shown in Fig. 5.11. Note that the spanwise grid-spacing Δz is equal to the smallest Δx in the domain and that the wall-normal Δy_{wall} is equal to 0.3 times the smallest h_{wm} in the domain, at each iteration. The initial wall shear stress is qualitatively completely wrong, as expected with the exchange location outside the boundary layer. The algorithm initially produces an almost uniform reduction of h_{wm} due to the inability to distinguish between different flow regions. From the fourth iteration (third adaptation), the algorithm produces a markedly smaller h_{wm} around the separation bubble. The refinement is particularly focused around $(x - x_{\mathrm{imp}})/\delta_{\mathrm{ref}} \approx -6$ which is near the separation location; in fact, by the sixth iteration the h_{wm}-algorithm effectively recommends wall-resolved LES around the separation location. The algorithm generally produces a smaller h_{wm} in the downstream flow compared to the upstream, despite the larger boundary layer thickness downstream of the interaction. The reason is that the boundary layer has not re-equilibrated yet, and thus the wall-model is less accurate in the downstream region. Iterations 4 and 5 produce a spurious peak in h_{wm} around $(x - x_{\mathrm{imp}})/\delta_{\mathrm{ref}} \approx -12$.

The predicted τ_{w} is somewhat inaccurate for all iterations. Part of the reason is that this shock/boundary-layer interaction case has a very shallow ($y^+ \lesssim 5$) "pre-cursor" separation bubble of length $\sim \delta_{\mathrm{ref}}$ before the "real" separation [58], which is impossible to capture with WMLES: the fact that the final iteration almost captures the correct point of decreasing τ_{w} is due to the algorithm having produced a h_{wm} consistent with WRLES in this region.

Note that the τ_{w} gets slightly inaccurate for the reattached region for the finest iteration. This does not signify that the algorithm is pushing the solution to a less accurate state, rather it shows that the previous accurate results were due to pure luck with regards to the h_{wm}.

The flow properties along wall-normal lines can be seen in the left column of the Figure 5.12, and they tell a clearer story. The final iteration produces a large improvement in accuracy, which is due to the grid reaching a critical level of resolution (to be discussed further below). However, note that we are trying to simulate a situation where the user does not know the exact grid resolution requirements for a complicated flow, so this convergence actually shows the need for coupled grid-adaptation with this h_{wm}-adaptation algorithm.

5.4.1 Decoupling the Effects of the Grid from the Exchange Location

The sudden convergence of the results in the final iteration raises an important question: were the inaccurate results on the previous iterations caused by an insufficient h_{wm} or an insufficient grid? Recall that the grid must be sufficiently fine both with respect to the turbulence around the exchange location (e.g., $\Delta x/h_{\mathrm{wm}}, \Delta z/h_{\mathrm{wm}} \lesssim 0.8$ and $\Delta y/h_{\mathrm{wm}} \lesssim 0.3$, although these specific numbers were developed for equilibrium flow) and in the outer part of the boundary layer more generally (e.g., $\Delta x/\delta, \Delta z/\delta \lesssim 0.05 - 0.08$ and $\Delta y/\delta \lesssim 0.01 - 0.04$) [14, 50]; only the former of these criteria was enforced when generating the grids in the present study (to mimic an applied situation where one may not know the boundary layer thickness in advance).

To investigate this issue, a second batch of WMLES cases was computed for the shock/boundary-layer interaction problem. This second batch has the same h_{wm} profiles as before, but with every iteration using the finest grid: this therefore isolates the effect of changing h_{wm} from the effect of changing the grid. The results are shown in Fig. 5.12 (right column). It is quite clear that the results with the fixed fine grid converge much more smoothly and rapidly to the WRLES results, with almost converged results being reached as early as the third-to-last iteration. This clearly proves that the results in the previous section were caused by inadequacy of the grid, not the h_{wm} field by itself. This does not mean that there is no need to be careful in finding the right h_{wm}, but rather that there is a strong need to adapt both the grid and the h_{wm} in a combined manner. In addition, the fact that the present algorithm is able to produce converged results, despite the simplistic connection between the h_{wm} and the grid, is a clear strong point.

5.5 Application to an Airfoil

Lastly, we test the algorithm on a NACA 64A-110 airfoil at 0° angle of attack. This case was run by our collaboration partners at University of Stuttgart using their discontinous Galerkin code FLEXI [59], and was gracefully shared with us by Marcel Blind [60]. The data is currently unpublished. The chord Reynolds number is $Re_c = 936,000$ and the freestream Mach number is $Ma = 0.72$ which means the flow is compressible, but it is not fast enough to create a shock wave on the airfoil. There is both WRLES and WMLES data on the flow. The WMLES was run with $h_{\mathrm{wm}} \approx 0.1\delta$ using the local boundary layer thickness δ obtained by the WRLES, which is plotted as the solid line in the top plot in Figure 5.13.

We have run the algorithm on the WMLES data, asking what would be the suggested h_{wm} for the same cost, unlike the previous tests in this chapter. The resulting h_{wm} field is represented in the top plot of the Figure 5.13 with a dash-dotted line. The first thing that strikes the eye is that the cost does not seem the same between the base case and the suggested new h_{wm}. This is however an illusion, and a good one proving the special attention the leading edge needs. The portion where the base h_{wm} is smaller than the suggested h_{wm} at the leading edge contributes much of the cost for the base simulation, through the cost being scaled by $1/h_{\mathrm{wm}}^2$ which can create large jumps in small numbers, that it can balance the cost between h_{wm}s for the remaining large length of the chord. In fact, this phenomenon is so strong at the leading edge region, where δ approaches 0 at the leading edge, the current test was only applied to the $x/c \gtrsim 0.002$, to overcome the leading edge generating infinite cost to the new suggested simulation.

The results after the leading edge region are more or less expected. The algorithm picks up where the flow is tripped into turbulence at $x = 0.05c$ and suggests a different h_{wm} there, although larger. This suggests that the τ_{w} estimation at the tripping region is not sensitive to the h_{wm} choice. Afterwards, the suggested h_{wm} stays almost constant where the previous $h_{\mathrm{wm}} \approx 0.1\delta$ keeps growing. There are two effects here, the boundary layer growing which should push the h_{wm} away from the wall, and the flow having stronger non-equilibrium effects which should push the equilibrium wall-model towards the wall. These effects seem to perfectly balance each other for this case, leading to an almost constant h_{wm} throughout.

The suggested h_{wm} field has been fed the averaged WMLES flow field data to assess the *a priori* τ_{w} prediction, and the results are seen in the bottom plot of Figure 5.13. The suggested h_{wm} does not seem to make too much of an improvement compared to the base h_{wm}, meaning that the wall-model is not sensitive to the h_{wm} location too much. Note that this could also be deduced by the suggested h_{wm} being constant for the larger portion of the domain. There exists a difference between the WRLES and the WMLES estimates for $0.1 \lesssim x/c \lesssim 0.6$, showing the simulation is not converged from the viewpoint of the wall-model for that region. For the remainder of the airfoil $x/c \gtrsim 0.6$, different h_{wm} fields generate accurate results compared to the WRLES, showing that this is not a case of getting the τ_{w} right by luck, which was the case for some portions of the wall-mounted hump at some iterations as previously explained.

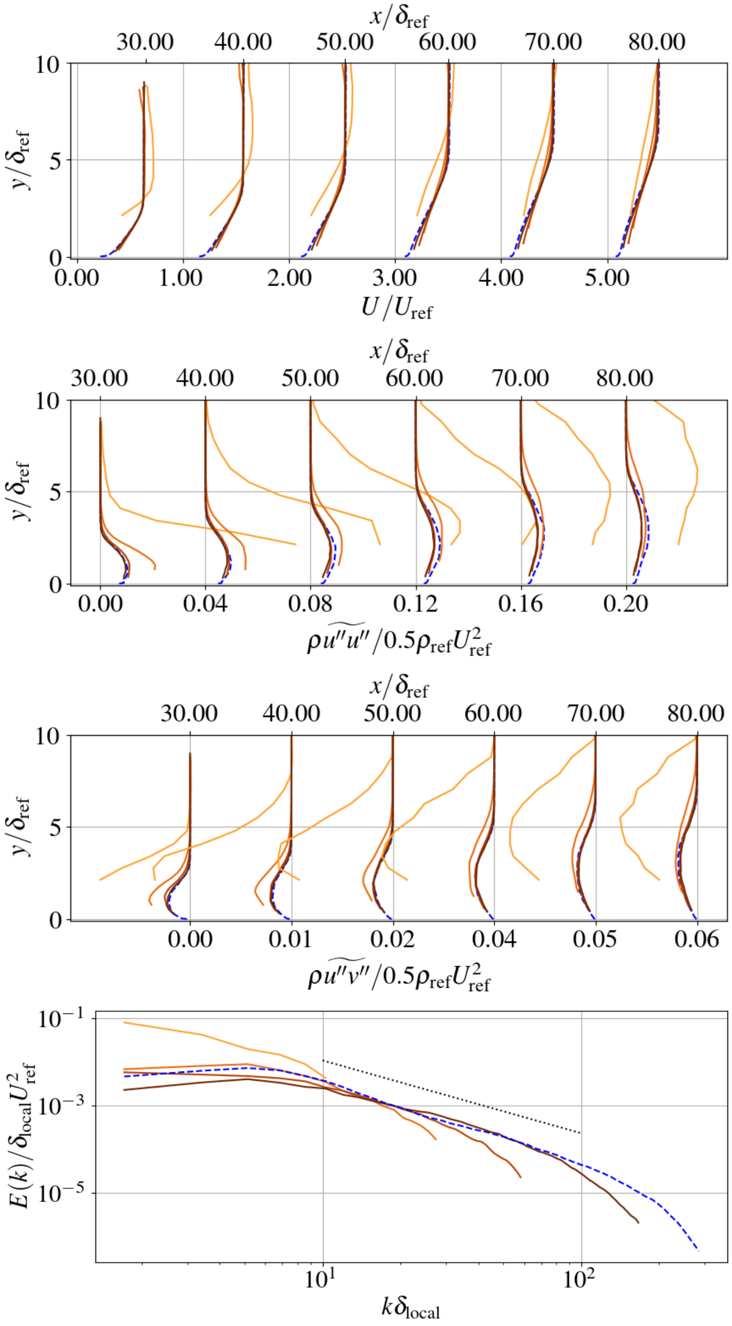

Figure 5.9: *A posteriori* results for the adverse pressure gradient case at select locations. Showing the (in order from top to bottom) mean velocity, streamwise Reynolds stress, Reynolds shear stress, and kinetic energy spectrum in the spanwise direction in the middle of the boundary layer at $x/\delta_{\mathrm{ref}} = 70$. Darker color means further iterations, each iteration has roughly $4\times$ higher cost than the previous one. The blue dashed line is the base WMLES, and for the energy spectrum the dotted line is the $-5/3$ slope.

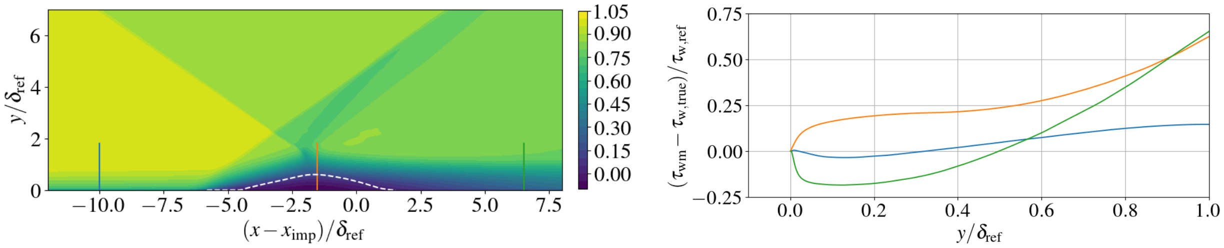

Figure 5.10: Streamwise mean velocity for the shock/boundary-layer interaction case (left) showing the separation bubble (dashed white line) and three sample locations (colored lines), and the wall-model error variation at the three sample locations (right).

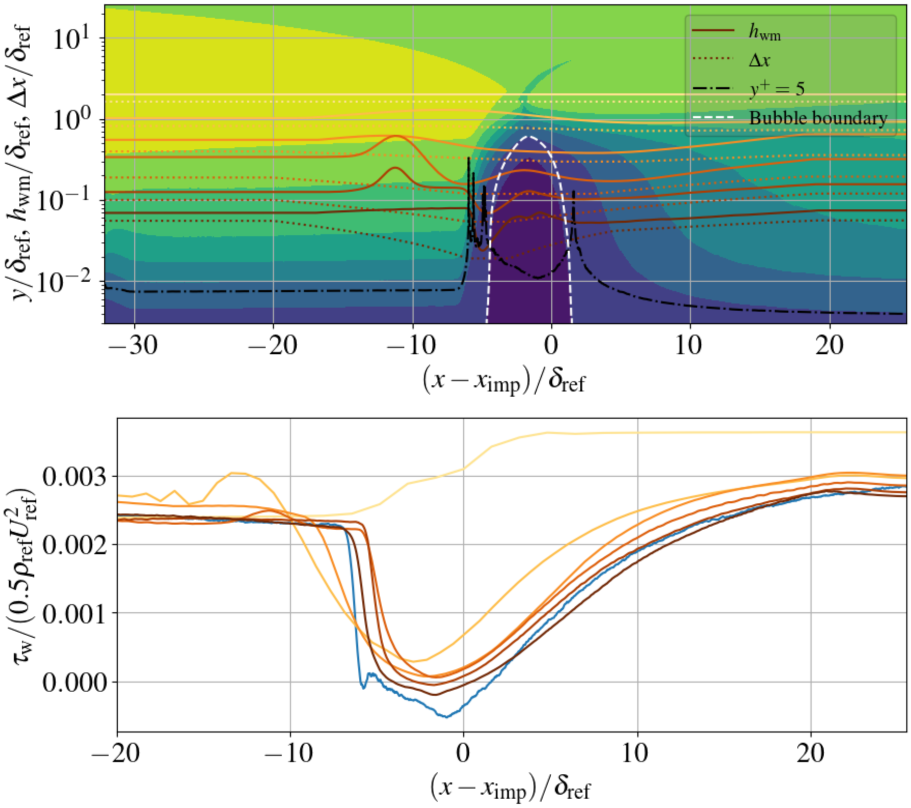

Figure 5.11: Results for the shock/boundary-layer interaction problem, showing the sequence of h_{wm} profiles (top figure, solid lines), Δx grid-spacing (top, dotted) and resulting skin friction coefficient (bottom). Darker colors mean later iterations. The blue line is WRLES, the black dash-dotted line is $y^+ = 5$, and the white dashed line is the separation bubble. Each iteration has roughly $4\times$ higher cost than the previous one.

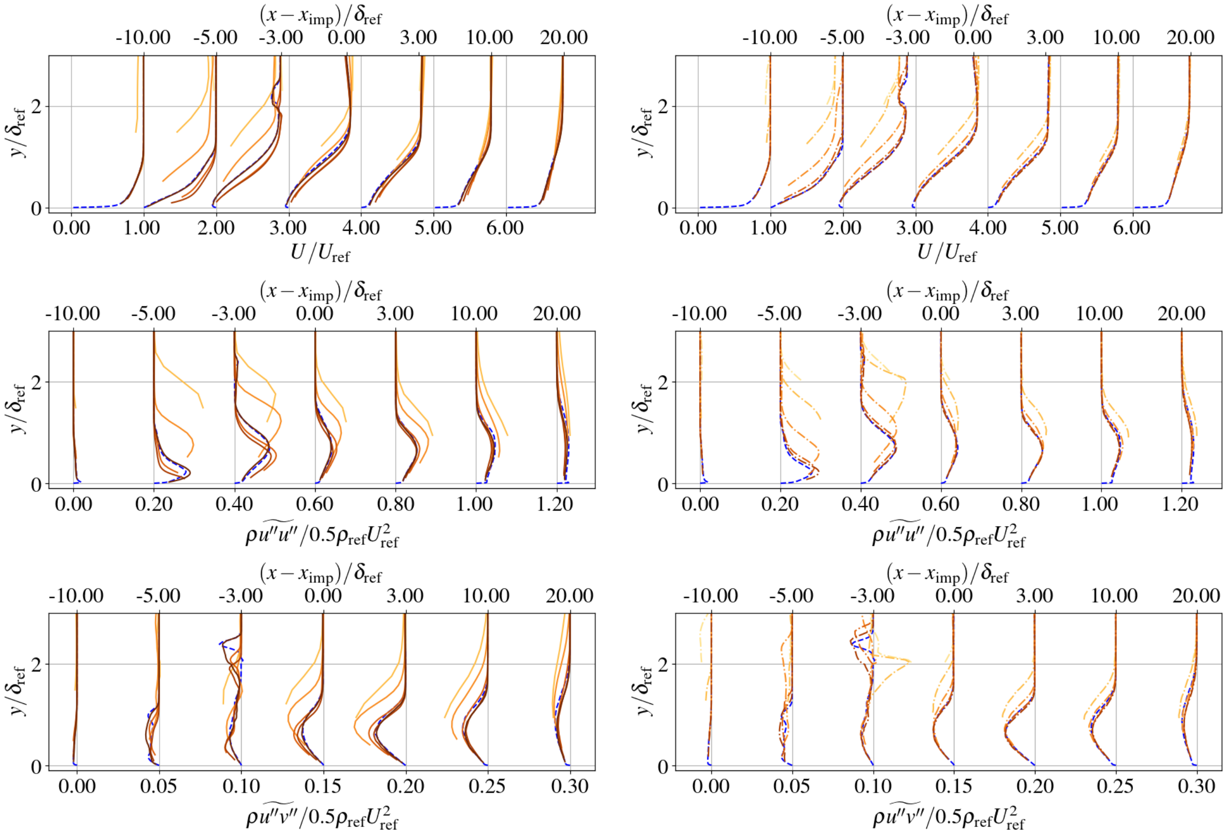

Figure 5.12: Results for the shock/boundary-layer interaction problem, showing the mean velocity (top), the streamwise Reynolds stress (middle), and the Reynolds shear stress (bottom). Left column: base sequence, with the grid refined at every iteration. Right column: sequence with a fine grid for all h_{wm}. Darker colors mean later iterations. The dashed blue line is the WRLES. Each iteration has roughly $4\times$ higher cost than the previous.

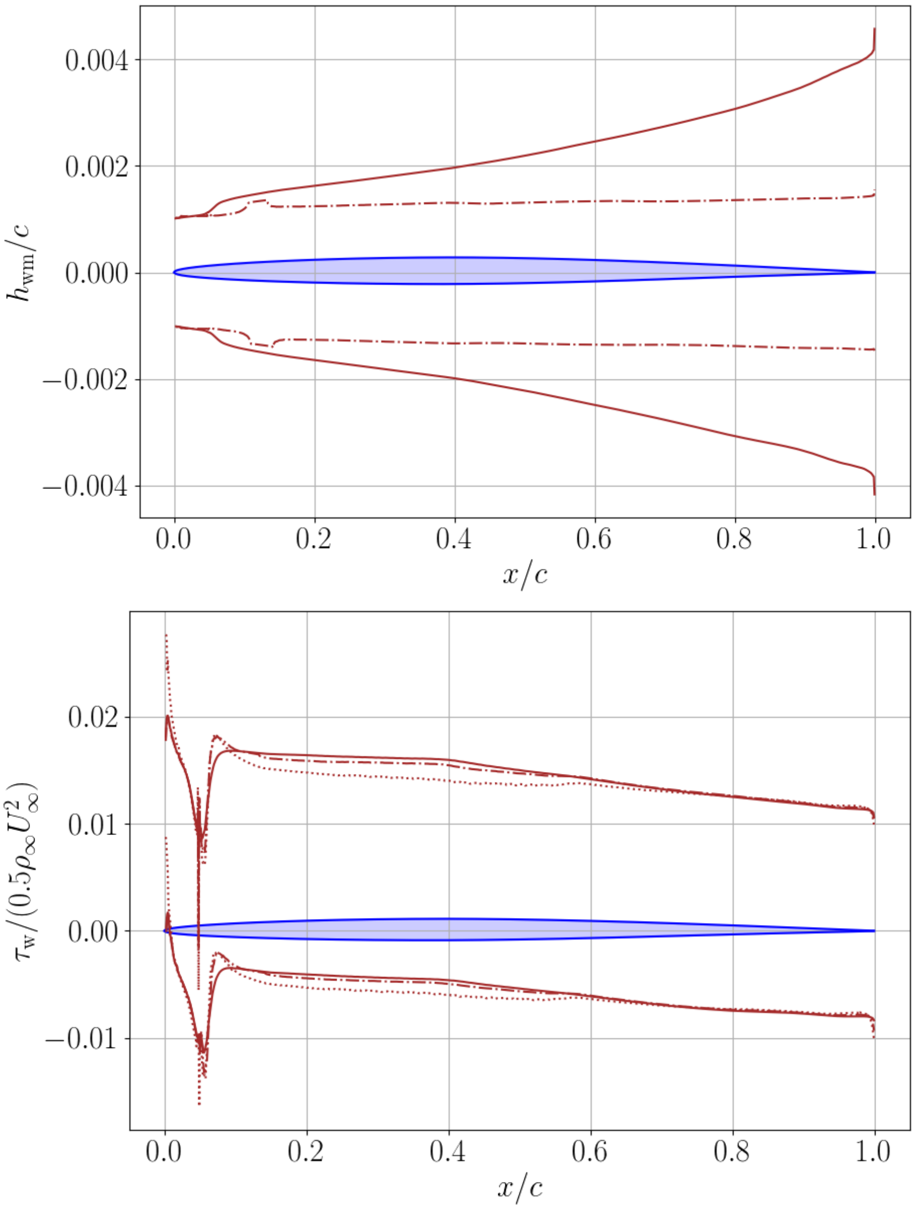

Figure 5.13: Result of the application to the NACA 64A-110 airfoil. The h_{wm} profiles are seen on the top, with straight lines 0.1δ and dash-dotted line suggested h_{wm} with similar cost. The τ_{w} profiles are seen on the bottom, shifted by ± 0.1 for suction and pressure sides respectively, straight line representing the WMLES, dash-dotted line the *a priori* application of the EQWM to the flow field, and dotted line the WRLES results.

Chapter 6: Conclusions and Future Work

The objective of this thesis is to advance the state-of-the-art in wall-modeled large eddy simulations (WMLES). We approach the question of WMLES from a different angle compared to most prior studies, asking how we can use a prior solution to systematically (using an algorithm) determine the "best" thickness of the wall-modeled layer. This is important since all WMLES approaches require some user choice in specifying the distance h_{wm} away from the wall over which the wall-model is applied (the "exchange location" or "modeling interface"). This choice may be implicit, e.g., if tied to the computational grid, but is nevertheless a choice that is made based on user expertise.

We develop and explore three algorithms, two based on ideas from flow physics and the wall-model characteristics, and one based on mathematical concepts to estimate the sensitivity of the solution to the chosen h_{wm}. Each one is entirely a postprocessing operation, performed using averaged data from an existing LES run. The algorithms are therefore meant to be applied in an iterative manner: an initial WMLES run with an initial guess of h_{wm} , followed by iterative improvements until the results are deemed converged. All three are relatively successful, being able to identify that a smaller portion of the boundary layer should be modeled in regions of strong acceleration or deceleration.

The two physics-based algorithms are intended to determine a good h_{wm} for the equilibrium wall-model only, with Algorithm 1 targeting where the equilibrium assumption becomes invalid and Algorithm 2 targeting where turbulent stresses vanish. Both algorithms are calibrated to produce $h_{\mathrm{wm}} \approx 0.1\delta$ in equilibrium boundary layers. The algoritms were tested a priori on DNS and WRLES of different flow cases, seeing that both of the algorithms show satisfactory results for the attached regions of the flow, but Algorithm 2 somewhat failing to predict a correct τ_{w} for the separated regions. These algorithms were also tested by performing WMLES of an equilibrium boundary layer where we started from a way too large h_{wm} and a way too coarse initial grid. Both algorithms were able to produce an essentially converged h_{wm} in 3 iterations.

The third algorithm was based on mathematical concepts, and was developed because the physics-based algorithms are ignorant to the cost of a new simulation they suggest, and they were specific to the equilibrium wall-model. This algorithm had three components: (i) an error model relating the error in the wall stress to the local h_{wm}; (ii) a way to find the parameter in the error model from existing simulation data; and (iii) an optimization formulation that defines the optimal h_{wm} field. It was also a post-processing algorithms meaning several LES runs were needed. For this algorithm, each iteration is done with an increasing computational cost, analogously to simulations on a sequence of finer and finer grids. By increasing the computational cost in each iteration, the cumulative cost of all prior iterations becomes small: e.g., if the cost is doubled in each iteration, the cumulative cost of all prior iterations is no more than twice the cost of the final run; if the cost is quadrupled in each iteration, the cumulative cost is at most 4/3 the cost of the final run. The added cost of computing multiple cases in a sequence is therefore small.

This algorithm is also tested with an equilibrium wall model. It is found to produce reduced h_{wm} values in regions of nonequilibrium flow and larger h_{wm} in regions where the wall model is expected to be more accurate, similar to its physics-based counterparts. All algorithms (except physics-based Algorithm 2 only at separated regions) arguably produces a similar h_{wm} field to what an experienced user would choose for the wall-mounted hump and shock/boundary-layer interaction problems used here. The ability to replicate an experienced user is viewed as a success for these relatively simple flow problems, suggesting that the algorithms will become very useful in more complicated flows. Especially, the fact that the optimization-based algorithm produces converging results even when given highly inaccurate initial guesses is encouraging for realistic applications.

One important observation is the relevance of the grid in the simulation process, as presented in the case of the optimization-based algorithm. The shock/boundary-layer interaction case showed clearly that the solutions converge only when both the h_{wm} and the grid are sufficiently fine, and thus the present h_{wm} algorithm cannot, by itself, be sufficient. This was a finding contrary to our understanding and estimation at the beginning of this work, where we expected that since the grid is refined with h_{wm} to feed the wall-model sufficiently good data, the h_{wm} refinement itself would be enough. Clearly, the errors purely due to the grid quality affected the simulations and proved the need to pay attention to the grid and use additional, independent judgement when creating it.

These algorithms should help make the wall-modeling in LES less user-dependent, helping standardize the practice, and saving human time and energy that may be needed to perfect other parts of the simulation. In the further future, a better LES practice would be one that is almost fully automated in every aspect, with minimal (however still finite) input required by the experience of a user, and these algorithms can be one pillar of that almost fully automated practice.

6.1 Future Work

The present work is the first in the literature to explore ideas about adaptively finding the ideal exchange location in WMLES, and therefore there is much future work to be done.

First, the math-based algorithm should be tested with a non-equilibrium wall-model, to verify that this would produce larger h_{wm} values in regions of non-equilibrium flow. This is one of the hypothesized strong points of the algorithm, but it is yet to be tested.

Second, an improved but more specialized math-based algorithm can be created if the first physics-based algorithm (based on the total shear stress) is implemented as the error estimator for the optimization based algorithm. This would likely be able to tell where an equilibrium boundary layer is, a feature that the math-based algorithm lacks in its current form on purpose. While this algorithm would be exclusive to equilibrium wall-models, it would still be valuable to the community as they are widely used.

Third, the error estimation should also be developed for the wall heat transfer in the optimization based algorithm. The focus of this work has been on adiabatic flows and their wall shear stress τ_{w}, however flows with wall heat transfer would equally benefit from such an h_{wm} determination algorithm. The question will also inevitably arise, how to combine the error models of the momentum and the energy equations, which one takes precedence, if any of them indeed does.

Fourth, the algorithms should also be applied to more complex flow fields, to assess their utility in enabling WMLES for realistic problems. While the problems that were solved here had complicated flow features, they are still too simplistic compared to real-life flow scenarios, e.g. a complete aircraft.

Finally, the algorithms need to be merged with a grid determination algorithm, possibly with the one developed in the same lab by Toosi and Larsson [61]. As discussed, the last section of the optimization based algorithm proved that the grid is equally as important, if not even more important, when getting the results right, and the h_{wm} and grid adaptation really belong together under one big umbrella of adaptation in LES.

Bibliography

[1] J. C. del Alamo and J. Jimenez. Spectra of the very large anisotropic scales in turbulent channels. *Physics of Fluids*, 15(6):L41–L43, 2003. doi: https://doi.org/10.1063/1.1570830.

[2] S. Hoyas and J. Jimenez. Scaling of the velocity fluctuations in turbulent channels up to $Re_\tau = 2003$. *Physics of Fluids*, 18(1):011702, 2006. doi: https://doi.org/10.1063/1.2162185.

[3] Mark P. Simens, Javier Jiménez, Sergio Hoyas, and Yoshinori Mizuno. A high-resolution code for turbulent boundary layers. *Journal of Computational Physics*, 228(11):4218 – 4231, 2009. doi: https://doi.org/10.1016/j.jcp.2009.02.031.

[4] JAVIER Jimenez, SERGIO Hoyas, MARK P. Simens, and Yoshinori Mizuno. Turbulent boundary layers and channels at moderate reynolds numbers. *Journal of Fluid Mechanics*, 657:335–360, August 2010. doi: https://doi.org/10.1017/S0022112010001370.

[5] J. A. Sillero, J. Jimenez, and R. D. Moser. One-point statistics for turbulent wall-bounded flows at Reynolds numbers up to $\delta^+ \approx 2000$. *Physics of Fluids*, 25(10):105102, 2013. doi: https://doi.org/10.1063/1.4823831.

[6] Juan A. Sillero, Javier Jiménez, and Robert D. Moser. Two-point statistics for turbulent

boundary layers and channels at reynolds numbers up to $\delta^+ \approx 2000$. *Physics of Fluids*, 26 (10):105109, 2014. doi: https://doi.org/10.1063/1.4899259.

[7] S. Pirozzoli, F. Grasso, and T. B. Gatski. Direct numerical simulation and analysis of a spatially evolving supersonic turbulent boundary layer at *M*=2.25. *Physics of Fluids*, 16(3): 530–545, 2004. doi: https://doi.org/10.1063/1.1637604.

[8] Chao Zhang, Lian Duan, and Meelan M. Choudhari. Direct numerical simulation database for supersonic and hypersonic turbulent boundary layers. *AIAA Journal*, 56(11):4297–4311, 2018. doi: 10.2514/1.J057296. URL `https://doi.org/10.2514/1.J057296`.

[9] A. Uzun and M. R. Malik. Large-eddy simulation of flow over a wall-mounted hump with separation and reattachment. *AIAA Journal*, 56(2):715–730, 2018. doi: https://doi.org/10.2514/1.J056397.

[10] A. Uzun. Private communication, 2019.

[11] Ali Uzun and Mujeeb R. Malik. Wall-resolved large-eddy simulations of transonic shock-induced flow separation. *AIAA Journal*, 57(5):1955–1972, 2019.

[12] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

[13] H. Tennekes and J. L. Lumley. *A First Course in Turbulence*. MIT Press, 1972.

[14] J. Larsson, S. Kawai, J. Bodart, and I. Bermejo-Moreno. Large eddy simulation with modeled wall-stress: recent progress and future directions. *Mechanical Engineering Reviews*, 3 (1):15–00418, 2016. doi: https://doi.org/10.1299/mer.15-00418.

[15] S. T. Bose and G. I. Park. Wall-modeled large-eddy simulation for complex turbulent flows. *Annual Review of Fluid Mechanics*, 50(1):535–561, 2018. doi: https://doi.org/10.1146/annurev-fluid-122316-045241.

[16] U. Piomelli and E. Balaras. Wall-layer models for large-eddy simulations. *Annual Review of Fluid Mechanics*, 34(1):349–374, 2002. doi: https://doi.org/10.1146/annurev.fluid.34.082901.144919.

[17] P. R. Spalart, W.-H. Jou, M. Strelets, and S. R. Allmaras. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. In C. Liu and Z. Liu, editors, *Advances in DNS/LES*, Greyden, Columbus, OH, 1997.

[18] A. J. Musker. Explicit expression for the smooth wall velocity distribution in a turbulent boundary layer. *AIAA Journal*, 17(6):655–657, 1979. doi: 10.2514/3.61193. URL `https://doi.org/10.2514/3.61193`.

[19] P. Quéméré and P. Sagaut. Zonal multi-domain rans/les simulations of turbulent flows. *International Journal for Numerical Methods in Fluids*, 40(7):903–925, 2002. doi: https://doi.org/10.1002/fld.381. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.381`.

[20] P. R. Spalart. Detached-eddy simulation. *Annual Review of Fluid Mechanics*, 41(1):181–202, 2009. doi: https://doi.org/10.1146/annurev.fluid.010908.165130.

[21] P. Sagaut and S. Decka nd M. Terracol. *Multiscale and Multiresolution Approaches in Turbulence*. IMPERIAL COLLEGE PRESS, 2006. doi: 10.1142/p447. URL `https://www.worldscientific.com/doi/abs/10.1142/p447`.

[22] J. W. Deardorff. A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics*, 41:453–480, 1970.

[23] D. B. Spalding. A Single Formula for the "Law of the Wall". *Journal of Applied Mechanics*, 28(3):455–458, 09 1961. ISSN 0021-8936. doi: 10.1115/1.3641728. URL `https://doi.org/10.1115/1.3641728`.

[24] S. Kawai and J. Larsson. Wall-modeling in large eddy simulation: length scales, grid resolution and accuracy. *Physics of Fluids*, 24(1):015105, 2012. doi: https://doi.org/10.1063/1.3678331.

[25] A. S. Ghate, G. K. Kenway, G.-D. Stich, O. M. Browne, J. A. Housman, and C. C. Kiris. Transonic lift and drag predictions using wall-modeled large eddy simulations. In *AIAA Scitech 2021 Forum*, 2021. doi: https://doi.org/10.2514/6.2021-1439. 2021-1439.

[26] P. S. Iyer and M. R. Malik. Wall-modeled large eddy simulation of flow over a wall-mounted hump. In *46th AIAA Fluid Dynamics Conference*, 2016. doi: https://doi.org/10.2514/6.2016-3186. 2016-3186.

[27] S. Kawai and J. Larsson. Dynamic non-equilibrium wall-modeling for large eddy simulation at high Reynolds numbers. *Physics of Fluids*, 25(1):015105, 2013. doi: https://doi.org/10.1063/1.4775363.

[28] A. Frere, C. C. de Wiart, K. Hillewaert, P. Chatelain, and G. Winckelmans. Application of wall-models to discontinuous Galerkin LES. *Physics of Fluids*, 29(8):085111, 2017. doi: https://doi.org/10.1063/1.4998977.

[29] I. Bermejo-Moreno, L. Campo, J. Larsson, J. Bodart, D. Helmer, and J. Eaton. Confinement effects in shock wave/turbulent boundary layer interactions through wall-modeled large-eddy simulations. *Journal of Fluid Mechanics*, 758:5–62, November 2014. doi: https://doi.org/10.1017/jfm.2014.505.

[30] G. I. Park. Wall-modeled large-eddy simulation of a high Reynolds number separating and reattaching flow. *AIAA Journal*, 55(11):3709–3721, 2017. doi: https://doi.org/10.2514/1.J055745.

[31] Ali Berk Kahraman and Johan Larsson. Adaptive determination of the wall modeled region in wmles. In *AIAA Scitech 2020 Forum*, 2020. doi: https://doi.org/10.2514/6.2020-1074.

[32] Ali Berk Kahraman and Johan Larsson. Adaptive determination of the optimal exchange location in wall-modeled large-eddy simulation. *AIAA Journal*, 0(0):1–12, 0. doi: 10.2514/1.J061347. URL `https://doi.org/10.2514/1.J061347`.

[33] Eric Garnier, Nikolaus Adams, and Pierre Sagaut. *Large eddy simulation for compressible flows*. Springer Science & Business Media, 2009.

[34] A. W. Vreman. An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids*, 16(10):3670–3681, 2004. doi: https://doi.org/10.1063/1.1785131.

[35] F. Ducros, F. Laporte, T. Souleres, V. Guinot, P. Moinat, and B. Caruelle. High-order fluxes for conservative skew-symmetric-like schemes in structured meshes: application to compressible flows. *Journal of Computational Physics*, 161(1):114–139, 2000. doi: https://doi.org/10.1006/jcph.2000.6492.

[36] M. Klein, A. Sadiki, and J. Janicka. A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations. *Journal of Computational Physics*, 186(2):652–665, 2003. doi: https://doi.org/10.1016/S0021-9991(03)00090-1.

[37] U. Schumann. Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli. *Journal of Computational Physics*, 18:376–404, 1975.

[38] P. J. Mason and N. S. Callen. On the magnitude of the subgrid-scale eddy coefficient in large-eddy simulation of turbulent channel flow. *Journal of Fluid Mechanics*, 162:439–462, 1986.

[39] H. Reichardt. Vollständige darstellung der turbulenten geschwindigkeitsverteilung in glatten leitungen. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 31(7):208–219, 1951. doi: https://doi.org/10.1002/zamm.19510310704. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/zamm.19510310704`.

[40] H. Werner and H. Wengle. Large-eddy simulation of turbulent flow over and around a cube in a plate channel. In *Turbulent Shear Flows*, 1991.

[41] E. Balaras, C. Benocci, and U. Piomelli. Two-layer approximate boundary conditions for large-eddy simulations. *AIAA Journal*, 34(6):1111–1119, 1996.

[42] W. Cabot and P. Moin. Approximate wall boundary conditions in the large-eddy simulation of high Reynolds number flow. *Flow Turb. Comb.*, 63:269–291, 1999.

[43] M. Wang and P. Moin. Dynamic wall modeling for large-eddy simulation of complex turbulent flows. *Physics of Fluids*, 14(7):2043–2051, 2002.

[44] George Ilhwan Park and Parviz Moin. An improved dynamic non-equilibrium wall-model for large eddy simulation. *Physics of Fluids*, 26(1):015108, 2014. doi: 10.1063/1.4861069. URL `https://doi.org/10.1063/1.4861069`.

[45] S. Hickel, E. Touber, J. Bodart, and J. Larsson. A parametrized non-equilibrium wall-model for large-eddy simulations. In *Proceedings of the Summer Program*, pages 127–136. Center for Turbulence Research, 2012.

[46] X. I. A. Yang, J. Sadique, R. Mittal, and C. Meneveau. Integral wall model for large eddy simulations of wall-bounded turbulent flows. *Physics of Fluids*, 27:025112, 2015.

[47] Tarik Dzanic and Joseph C. Oefelein. Non-equilibrium wall modeling for large eddy simulation of stalled iced airfoils. In *AIAA Scitech 2020 Forum*, 2020. doi: https://doi.org/10.2514/6.2020-1076.

[48] U. Piomelli. Wall-layer models for large-eddy simulations. *Progress in Aerospace Sciences*, 44(6):437–446, 2008. doi: https://doi.org/10.1016/j.paerosci.2008.06.001.

[49] D. A. Johnson and L. S. King. A mathematically simple turbulence closure model for attached and separated turbulent boundary layers. *AIAA Journal*, 23(11):1684–1692, 1985. doi: 10.2514/3.9152. URL `https://doi.org/10.2514/3.9152`.

[50] Wall-modeled large eddy simulation resource. `http://wmles.umd.edu/instantaneous-vs-averaged-input`, retrieved January 20, 2022.

[51] Prahladh S. Iyer and Mujeeb R. Malik. Analysis of the equilibrium wall model for high-speed turbulent flows. *Physical Review Fluids*, 4(7):1–26, 2019. doi: https://doi.org/10.1103/PhysRevFluids.4.074604.

[52] S.C. Chapra and R.P. Canale. *Numerical methods for engineers*. McGraw-Hill, 2nd ed. edition, 1988.

[53] J. Lee, M. Cho, and H. Choi. Large eddy simulations of turbulent channel and boundary layer flows at high Reynolds number with mean wall shear stress boundary condition. *Physics of Fluids*, 25:110808, 2013.

[54] Pedro S. Volpiani, Matteo Bernardini, and Johan Larsson. Effects of a nonadiabatic wall on supersonic shock/boundary-layer interactions. *Physical Review Fluids*, 3:083401, Aug 2018.

[55] D. Greenblatt, K. B. Paschal, C.-S. Yao, J. Harris, N. W. Schaeffler, and A. E. Washburn. Experimental investigation of separation control Part I: baseline and steady suction. *AIAA Journal*, 44(12):2820–2830, 2006. doi: https://doi.org/10.2514/1.13817.

[56] P. E. Skare and P.-A. Krogstad. A turbulent equilibrium boundary layer near separation. *Journal of Fluid Mechanics*, 272:319–348, August 1994. doi: https://doi.org/10.1017/S0022112094004489.

[57] R. Vinuesa, A. Bobke, R. Örlü, and P. Schlatter. On determining characteristic length scales in pressure-gradient turbulent boundary layers. *Physics of Fluids*, 28(5):055101, 2016. ISSN 10897666. doi: https://doi.org/10.1063/1.4947532.

[58] J. Larsson, V. Kumar, N. Oberoi, M. Di Renzo, and S. Pirozzoli. Large eddy simulations of idealized shock/boundary-layer interactions with crossflow. *AIAA Journal*, 2022. doi: https://arc.aiaa.org/doi/10.2514/1.J061060. (In press).

[59] Florian Hindenlang, Gregor J. Gassner, Christoph Altmann, Andrea Beck, Marc Staudenmaier, and Claus-Dieter Munz. Explicit discontinuous galerkin methods for unsteady problems. *Computers & Fluids*, 61:86–93, 2012. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2012.03.006. URL `https://www.sciencedirect.com/science/article/pii/S004579301200093X`. ”High Fidelity Flow Simulations” Onera Scientific Day.

[60] M. Blind. Private communication, 2022.

[61] S. Toosi and J. Larsson. Towards systematic grid selection in LES: identifying the optimal spatial resolution by minimizing the solution sensitivity. *Computers & Fluids*, 201:104488, April 2020. doi: https://doi.org/10.1016/j.compfluid.2020.104488.