Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

Generation of tailored boundary layer meshes for non-equilibrium flows

Samuel Gómez González

A thesis presented for the Master's degree in Numerical Methods in Engineering

Thesis Directors:
Matteo Giacomini
Oriol Lehmkuhl

Barcelona
Wednesday 29th June, 2022

Abstract

The intention of the present work is to generate meshes tailored to the boundary layer to improve the flow predictions of wall-modeled large-eddy simulations (WMLES). To this end, the local reconstruction-method was implemented to measure the boundary layer thickness using the computations of a constant-height grid as a baseline.

The NASA Juncture Flow is used as benchmark. WMLES using Alya were conducted for an angle of attack of $AoA = 5^0$ and Reynolds number based on the crank chord $Re_c = 2.4 \cdot 10^6$. The errors in the prediction of mean velocity profiles are characterized at two different locations over the aircraft: the upstream region of the fuselage, and the wing-body juncture close to the leading-edge.

Better prediction of the flow close to the wall are observed with the tailored mesh with a equilibrium wall-function, especially at the upstream location where the equilibrium hypothesis holds.

Contents

Introduction

The aircraft design in the modern aerospace industry needs precise predictive tools to reduce energy consumption. One of the keys to improving energy efficiency is the use of computational fluid dynamics (CFD) for external aerodynamic applications [1], which have improved our understanding and prediction of complex flows. However, today they are not capable of meeting the stringent requirements for accuracy and computational efficiency demanded by the industry [2]. These limitations are largely imposed by the complexity of the turbulence.

Turbulence treatment has been mainly based on Reynolds-averaged Navier-Stokes (RANS) [3] models, from pure RANS solutions to hybrid methods such as Detached Eddy Simulation (DES) [4]. In the latter case, the RANS model is used near the wall, while eddy resolution methodologies are used far from it. The physical limitations of the RANS models have been reduced by expanding and calibrating their coefficients. However, despite their dependence on tunable parameters, they became the main methodology for the industry due to their computational efficiency.

An alternative methodology that is spreading lately is large-eddy simulation (LES) [5]. In LES, small scales of turbulence are modeled using a subgrid model (SGS) while large scales are resolved directly. In addition, computational and mesh requirements can be reduced by modeling the region near the wall (wall modeling) in such a way that only the outer layer region of the boundary layer is resolved [6]. Although WMLES improves on the shortcomings of the RANS-based methods, DES has been more widely accepted for its ease of implementation and the low level of development of the wall models.

In WMLES, even with a perfect wall model, the results would still be inaccurate, since the LES is under-resolved at the first grid point off the wall and thus provides inaccurate information to the wall model [7]. However, there is no requirement for the velocity to be evaluated at the first grid point off the wall, the only requirement is that the velocity is evaluated in the inner part of the boundary layer. Hence, a proper discretization of the boundary layer is critical have a sufficiently accurate solution at the exchange interface.

In the present work, the performance of a tailored boundary layer mesh is tested using WMLES to predict the mean flow quantities over the NASA Juncture Flow Experiment [8]. In order to generate such a mesh an accurate estimation of the boundary layer thickness is necessary, thus state of the art methodologies are reviewed to assess their suitability for general non-equilibrium flows.

The employed benchmark consist of a full-span wing-fuselage body based on truncated DLR-F6 wings, with and without leading-edge horn at the wing root. Most CFD efforts on the NASA Juncture Flow Experiment so far have been focus on RANS or hybrid-RANS solvers [8–10], whwreas this work employs WMLES following the idea in [11].

The remainder of this thesis is organized as follows. In Chapter 1, the basic notions of Computational Fluid Dynamics to understand the content of this work are reviewed. Chapter 2, the methodologies for determining the thickness of the boundary layer are evaluated Later, the local-reconstruction of the inviscid velocity profile is validated (Chapter 3) and applied to the tailored mesh of the NASA Juncture Flow (Chapter 4). Finally, some conclusions are drawn and two appendices (Appendix A and Appendix B) provide some technical details on the employed methodology.

Chapter 1

Basics on CFD

Turbulent flows are characterized by chaotic changes in pressure and flow velocity. Many engineering applications involve turbulent flows, whence their numerical simulation is of great importance to scientists as well as to the engineering community.

However, turbulence has long resisted detailed physical analysis, and the interactions within the turbulence create a very complex phenomenon. Even though many turbulent flows can be easily observed, it is very difficult to give an exact and precise definition of turbulence. Richard Feynman described turbulence as "the most important unsolved problem in classical physics". However, most researchers generally agree on certain characteristics present in turbulent flows: unpredictability, three-dimensionality of fluctuations, diffusivity, and a wide spectrum of spatial and temporal scales.

For low values of the Mach number, fluid flows can be described using the incompressible Navier-Stokes (NS) equations, a set of partial differential equations that describe the motion of viscous flows. They arise from applying Newton's second law to fluid motion, together with the assumption that the stress in the fluid is the sum of a viscous diffusion term (proportional to the velocity gradient) and a pressure term. Its expression for the conservation of momentum and mass for incompressible flow is as follows:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \mathbf{0} \tag{1.1}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \tag{1.2}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor defined as $\boldsymbol{\sigma} = 2\nu\boldsymbol{\nabla}^S\boldsymbol{u} - p\boldsymbol{I}$, with $\boldsymbol{\nabla}^S\boldsymbol{u} = \frac{1}{2}(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}\boldsymbol{u}^T)$, p the kinematic pressure, $\boldsymbol{u}$ the fluid velocity field, and ν the kinematic viscosity.

Turbulent flows can be understood as a cascade of kinetic energy transmitted from larger structures to smaller ones. As the large scales are not able to dissipate energy at the rate they receive it, they break down into smaller and smaller scales until they are able to dissipate the excess energy. The responsible for the breaking process is the convective term while the diffusive term is the one in charge of the dissipation [12].

Although NS is a perfect mathematical model for flow physics, it cannot be solved analytically, except for simple cases. Therefore, they have to be approximated by numerical methods to obtain an acceptable solution and they need an appropriate space-time discretization.

1.1 Direct Numerical Simulations

Direct Numerical Simulation (DNS) is a methodology capable of representing the entire spatial and temporal energy spectrum. However, in most cases, this methodology is unfeasible due to the computational resources needed to solve all the turbulent structures [13].

1.2 Large Eddy Simulations

The Large Eddy Simulation (LES) method is based on solving the spatially filtered NS equations with filter length Δ and only take into account the large flow structures. The lack of dissipation due to unresolved scales smaller than Δ is accounted for by adding a subgrid viscosity ν_{sgs} to the diffusive term. Then, the NS momentum equation (Equation 1.1) for LES becomes roughly

$$\frac{\partial \overline{\boldsymbol{u}}}{\partial t} + (\overline{\boldsymbol{u}} \cdot \boldsymbol{\nabla})\overline{\boldsymbol{u}} = \boldsymbol{\nabla} \cdot \left[2(\nu + \nu_{sgs})\boldsymbol{\nabla}^S \overline{\boldsymbol{u}}\right] - \boldsymbol{\nabla}\overline{p} \tag{1.3}$$

where $\overline{\odot}$ represents the filtered variables. Additionally, a model is needed to evaluate ν_{sgs} [5]. For example, Vremann [14] proposed

$$\nu_{sgs} = c\sqrt{\frac{B_\beta}{\alpha_{ij}\alpha_{ij}}} \tag{1.4}$$

with

$$\alpha_{ij} = \frac{\partial \overline{u}_j}{\partial x_i}, \tag{1.5}$$

$$\beta_{ij} = \Delta_m^2 \alpha_{mi}\alpha_{mj}, \tag{1.6}$$

$$B_\beta = \beta_{11}\beta_{22} - \beta_{12}^2 + \beta_{11}\beta_{33} - \beta_{13}^2 + \beta_{22}\beta_{33} - \beta_{23}^2 \tag{1.7}$$

The model constant c is related to the Smagorinsky constant C_S [15] by $c \approx 2.5C_S^2$.

Nonetheless, this method has an extremely high computational cost for most application cases [6], with a total computational cost of a wall resolved (WR) computation $T_{cc}^{WR} \cong Re_L^{65/21}$ [16]. Re_L being the Reynolds number for the characteristic dimension L. Recalling the definition of the Reynolds Number as the ratio between inertial and viscous forces $Re_L = uL/\nu$, where u is the flow velocity and ν is the kinematic viscosity of the fluid.

1.3 Wall Modeled Large Eddy Simulation

Wall modeled Large Eddy Simulation (WMLES) takes the advantage of the accuracy of the LES methodology while avoiding the unaffordable requirements imposed by the presence of a solid wall, thus reducing computational cost to $T_{cc}^{WM} \cong Re_L^{4/3}$ [16].

It is worth recalling that in order to simulate viscous flows, boundary layer effects need to be accurately captured. To achieve this objective a sufficiently fine mesh is required in the vicinity of physical walls.

Although the boundary layer varies depending on the Reynolds number and the geometry, its physical behavior is generalizable. In fact, under equilibrium conditions, the boundary layer behavior is universal (law of the wall).

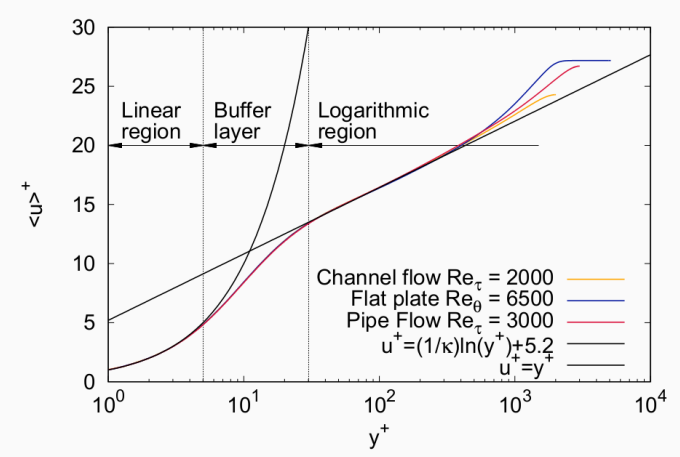


Figure 1.1: The law of the wall obtained through DNS of three different canonic flows: Channel flow at $Re_\tau \approx 2000$ [18] (orange line), a Flat plate turbulent boundary layer at $Re_\theta \approx 6500$ [19] (blue), and a Pipe flow at $Re_\tau \approx 3000$ [20] (red). The expressions of Prandtl and Von Kármán are also shown (extracted from [16]).

The wall closest layer is the viscous sublayer, which features a laminar flow regime. Then, there is a transition region from laminar to turbulent regime, the buffer layer. Finally, the mean streamwise velocity function evolves to a logarithmic profile in the log-law region, which was first identified by Von Kármán in 1930 [17]. The following expression for the mean velocity profile was proposed in [17]

$$u^+ = \frac{1}{\kappa} \ln y^+ + B, \tag{1.8}$$

where κ is the Von Kármán constant whose value is approximately 0.41, and B is another constant of value approximately 5.2. Additionally, quantities have been non-dimensionalized in wall units ($\odot^+$), with the friction velocity $u_\tau = \tau_w/\rho$ (where τ_w is the mean wall shear stress and ρ is the fluid density) and the viscous length $\mathcal{L} = \nu/u_\tau$.

In Figure 1.1, the law of the wall, $u^+ = f(y^+)$, obtained through DNS, is compared for three different equilibrium flows and it can be seen that the numerical results match the analytical expressions proposed by Prandtl and Von Kármán. Hence, WM allows to reduce the computational cost of simulations by providing a good approximation to model the inner layer instead of solving it.

One option is to employ a wall shear stress method in which an LES model solves for the entire domain, while an external model provides a shear stress to the LES domain as boundary condition.

The simplest approach is to use a function that evaluates the shear stress depending on the solution at a point and its distance from the wall, known as wall functions, namely

$$\tau_w = f_w(u, y) \tag{1.9}$$

In this context, Owen et al. [21] showed that in order to properly predict τ_w, the streamwise velocity u, should be taken at $y = 0.125\delta$, δ being the boundary layer thickness. Additionally, the mesh near the wall must be sufficiently fine so the flow is well resolved at this point.

In the next chapter different strategies to estimate the thickness δ will be reviewed

Chapter 2

Estimating the boundary layer thickness

In this chapter, different methods to compute the boundary layer thickness are reviewed, their limitations are discussed and their accuracies are compared, based on the review by Griffin et al. [22].

2.1 Zero-pressure-gradient boundary layers

Flows with no pressure gradient, or ZPGBL, approach the free flow velocity U_∞ as we move away from the wall. Then, in this type of flow, we can define the boundary layer thickness δ_n as the distance to the wall y where a percentage n of the U_∞ is reached, i.e.

$$\left.\frac{U}{U_\infty}\right|_{y=\delta_n} = \frac{n}{100} \tag{2.1}$$

where $n = 99$ is generally accepted as the definition of the boundary layer thickness.

However, this definition cannot be extended to canonical cases with pressure gradient or complex geometries.

2.2 Mean-vorticity based

These methods are based on the assumption that free-stream flow is irrotational. Therefore, they identify δ as the edge where the vorticity vanish. Within the boundary layer approximation, we can base the definition of velocity on vorticity (generalized velocity, Lighthill [23]) as

$$\widetilde{U} = \int_0^y -\Omega_z dy, \tag{2.2}$$

where Ω_z denotes the mean vorticity component in the spanwise direction.

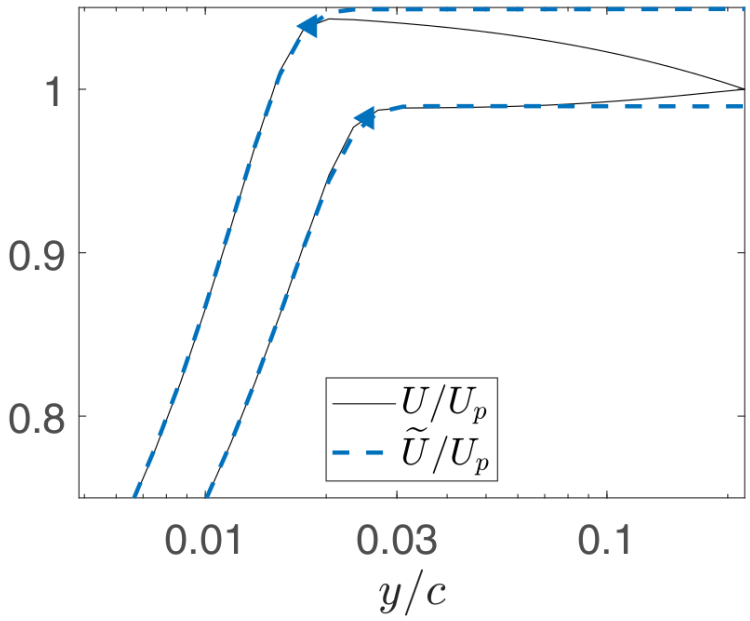


Figure 2.1: Mean streamwise velocity (black solid lines) and the generalized velocity (blue dashed lines) versus the wall-normal distance normalized by the airfoil chord c. Profiles from the suction side of a NACA 4412 airfoil at the angle of attack $(AoA) = 5^0$ and the Reynolds number based on the chord length $Re_c = 10^6$ at the streamwise location $x/c = 0.72$ (upper) and a NACA 0012 airfoil at $AoA = 0^0$ and $Re_c = 4 \cdot 10^5$ at the streamwise location $x/c = 0.90$ (lower). Blue triangles indicate the locations $y = \delta_{99}$ (extracted from [22]).

2.2.1 The $\widetilde{U}_\infty$ method

This method was proposed by Coleman et al. [24]. It determines δ similarly to a ZPGBL, using the generalized velocity $\widetilde{U}$ (Equation 2.2), namely

$$\left.\frac{\widetilde{U}}{\widetilde{U}_\infty}\right|_{y=\delta_n} = \frac{n}{100}. \tag{2.3}$$

In Figure 2.1, we observe that the mean streamwise velocity U continues to vary as $y \to y_p$ ($\odot_p$ denotes the furthest point from the wall in the wall-normal direction; it is an arbitrary length scale), whereas the generalized velocity achieves a constant asymptotic value as the flow becomes irrotational where the boundary layer edge is identified.

2.2.2 The $-y\Omega_z$ threshold method

Another approach was proposed by Uzun and Malik in [25], where a threshold $-y\Omega_z$ was established as a function of the value δ_n to be determined, that is,

$$\left.\frac{-y\Omega_z}{\max(-y\Omega_z)}\right|_{y=\delta_n} = C_\Omega. \tag{2.4}$$

For δ_{99}, $C_\Omega = 0.02$ is estimated empirically [25]. In Figure 2.2, $-\Omega_z$ is monotonically decreasing while $-y\Omega_z$ features multiple peaks and achieves its maximum in

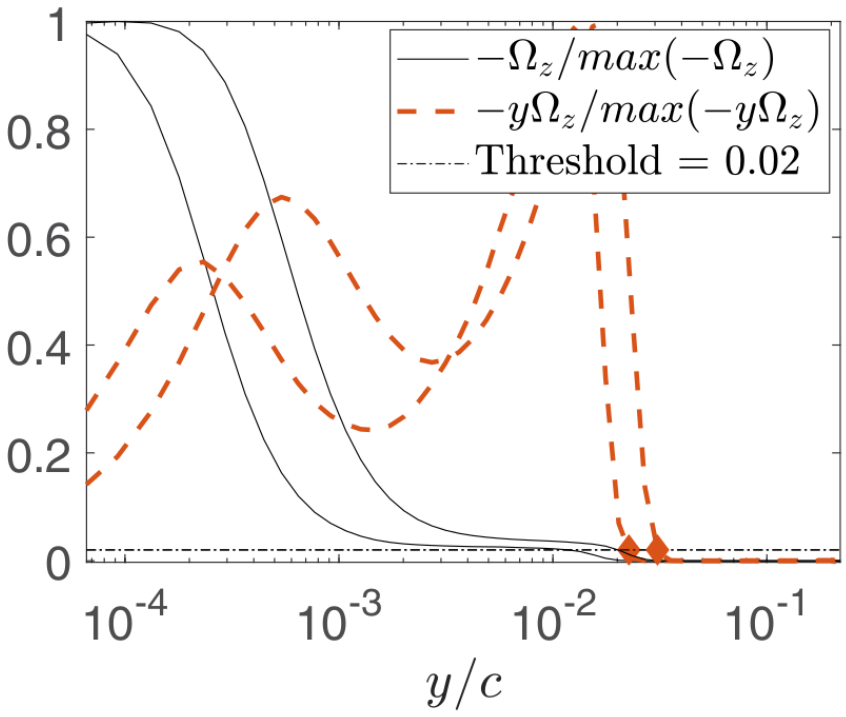


Figure 2.2: Distributions of the mean z vorticity Ω_z (black solid lines) and the boundary layer sensor $-y\Omega_z$ (orange dashed lines) versus the wall-normal distance normalized by the airfoil chord c. Profiles from the suction side of a NACA 4412 airfoil at $AoA = 5^0$ and $Re_c = 10^6$ at the streamwise location $x/c = 0.72$ (left) and a NACA 0012 airfoil at $AoA = 0^0$ and $Re_c = 4 \cdot 10^5$ at the streamwise location $x/c = 0.90$ (right). Orange diamonds indicate the locations $y = \delta_{99}$ (extracted from [22]).

the outer portion of the boundary layer. Since the $-y\Omega_z$ profile crosses the threshold C_Ω multiple times, the boundary layer thickness can be defined as the furthest point from the wall for which Equation 2.4 is satisfied.

A major disadvantage of methods based on mean vorticity is that an initial assumption about the upper limit of integration is required because a semi-infinite domain of integration is generally not available. Therefore, it is necessary to iteratively recompute it over a larger domain until it convergence is achieved.

Additionally, the consistency of mean vorticity-based methods with the classical definition of δ for ZPGBL flat plates depends on $\partial V/\partial x << \partial U/\partial y$. This condition is satisfied when the boundary layer thickness is much smaller than the distance to the leading edge of the flat plate and consequently $Re_x = U_e x/\nu$ is high. Therefore, these methods are not valid for low Reynolds numbers.

Another drawback is that they need the $\partial V/\partial x$ data, which is not usually available.

Finally, the method proposed by Uzun and Malik is also sensitive to the choice of an empirical threshold.

2.3 Mean-shear threshold

For these methods, δ is defined as the wall-normal distance y, at which the mean shear $\partial U/\partial y$ drops below an arbitrary threshold. If the threshold is set to zero, the method is referenced to as max method. Moreover, for its general use, the mean shear must be properly non-dimensionalized.

On the one hand, for the laminar case, it holds

$$\left.\frac{\partial(U/U_\infty)}{\partial(y/\delta)}\right|_{y=\delta_n} = C_l, \tag{2.5}$$

where a suitable non-dimensionalization has been considered ans C_l is the laminar threshold.

On the other hand, for the turbulent case, quantities are non-dimensionalized in wall units with the friction velocity u_τ and the viscous length $\mathcal{L}$. Consider the boundary layer description by Coles's wall-wake law [26]

$$U^+ = \frac{1}{\kappa}\ln y^+ + B + \frac{2\Pi}{\kappa}\sin^2\left(\frac{\pi y}{2\delta}\right), \tag{2.6}$$

the following mixed non-dimensionalization was proposed

$$\left.\frac{\partial U^+}{\partial y/\delta}\right|_{y=\delta_n} = C_t, \tag{2.7}$$

$C_t \approx 1/\kappa$ being the turbulent threshold.

Theoretically, the mean shear can be broken down into a viscous part and an inviscid part due to the curvature of the geometry. In practice, these methods do not provide a precise way to distinguish such contributions, so thresholds based on a equilibrium wall model are not generalizable. Therefore, these methods will only be successful if the inviscid part is small and the threshold is greater than it. In other words, for a case with a monotonically increasing velocity profile, there is no guarantee that a given threshold will be crossed. Additionally, since both non-dimensionalizations depend on δ, δ has to be computed iteratively and there is no guarantee that it will converge. Furtermore, there is no valid definition for laminar-turbulent transitions or vice versa, so these methods are not applicable to transitional flows.

2.4 Diagnostic plot

The method proposed by Vinuesa et al. [27] is based on the diagnostic plot of streamwise turbulence intensity $\sqrt{\overline{u'^2}}/U_e$ as a function of the streamwise velocity U/U_e. For its extension to non-equilibrium flows, we have to take into account $\sqrt{\overline{u'^2}}/(U_e\sqrt{H})$ (Drózdz et al. [28]), where $H = \delta^*/\theta$ is the shape factor of the boundary layer (δ^* and θ being the displacement and momentum thickness respectively). Vinuesa et al. [27] defined the boundary layer edge δ_{99} as the normal distance to the wall at which $\sqrt{\overline{u'^2}}/(U_e\sqrt{H}) = 0.02$, after observing asymptotic behavior of the diagnostic plot. However, this method is characterized by a number of drawbacks:

(i) Since the criterion for calculating δ depends on U_e, it is an iterative method that requires a suitable first guess of δ.

(ii) Although, streamwise turbulence intensity and streamwise velocity statistics are usually available in experimental and numerical databases. In Large Eddy Simulations (LES) the turbulence intensity is strongly dependent on the quality of the mesh resolution and for Reynolds-averaged Navier-Stokes (RANS) is modeled. Hence the accuracy of this method will be influenced by them.

(iii) It depends on an empirical threshold and have to be adapted for $n \neq 99$.

(iv) Since it is based on turbulence, it cannot be used for laminar or transitional flows.

2.5 Local-reconstruction

The local-reconstruction method definition (Griffin et al. [22]) for the boundary-layer thickness is given by

$$\left.\frac{U}{U_I}\right|_{y=\delta_n} = \frac{n}{100}, \tag{2.8}$$

where U_I is local reconstruction of the inviscid mean streamwise velocity. The calculation of U_I begins with the definition of the square velocity magnitude

$$U_m^2 = U_I^2 + V^2, \tag{2.9}$$

where the mean wall-normal velocity V is computed in the 2D local reference frame. Moreover, the stagnation pressure is given by

$$P_0 = P + \frac{1}{2}\rho U_m^2, \tag{2.10}$$

where P is the static pressure. Then, Bernoulli's equation can be applied globally rather than just along streamlines by assuming an irrotational inviscid flow that has $V[y]$ and $P[y]$ profiles equivalent to those of the viscous flow. Hence, the inviscid mean streamwise velocity is described as

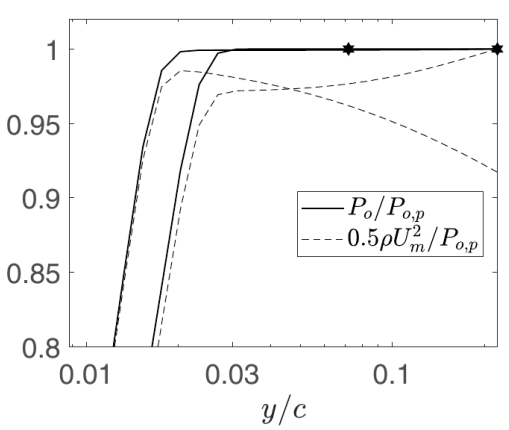
$$U_I = \pm\sqrt{\frac{2}{\rho}(P_{0,ref} - P[y]) - V[y]}, \tag{2.11}$$

where $P_{0,ref} = \max(P_0)$ denotes the total pressure achieved at a wall-normal distance in the outer flow and the sign of U_I should match that of $U[y = \delta]$.

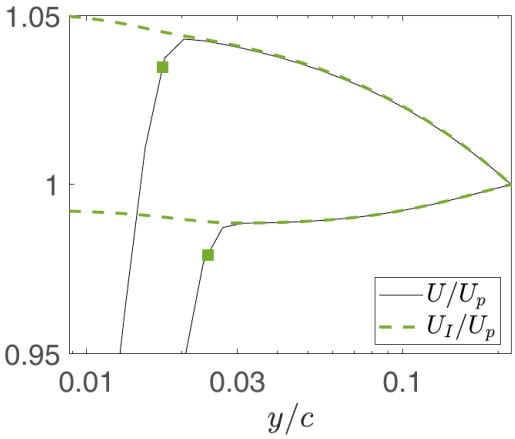
In Figure 2.3a, the dynamic pressure varies continuously whereas the stagnation pressure converges to a constant outside the boundary layer. In Figure 2.3b the inviscid solution $U_I[y]$ agrees with the viscous solution $U[y]$ outside the boundary layer.

2.6 Qualitative assessment

The flow over the NACA 4412 airfoil at $AoA = 5^0$ and $Re_c = 10^6$ is considered to evaluate the estimates of δ_{99} provided by the five methods discussed above, at different stations.



(a) Distributions of the stagnation pressure P_0 and the dynamic pressure versus the normalized wall-normal coordinate y/c.

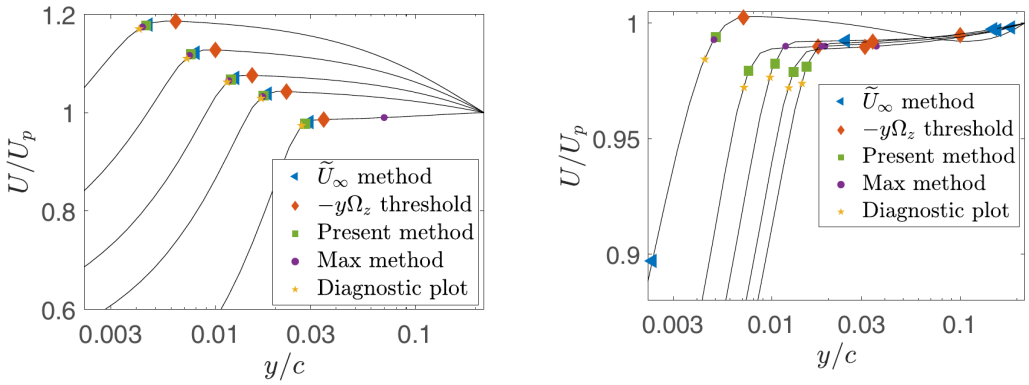


(b) Distributions of the velocity profile U and the locally reconstructed inviscid velocity profile U_I versus the normalized wall-normal coordinate y/c.

Figure 2.3: Profiles from the suction side of a NACA 4412 airfoil at $(AoA) = 5^0$ and $Re_c = 10^6$ at the streamwise location $x/c = 0.72$ (left) and a NACA 0012 airfoil at $AoA = 0^0$ and $Re_c = 4 \cdot 10^5$ at the streamwise location $x/c = 0.90$ (right). The maximum stagnation pressure (a) and the boundary-layer edge (b) are indicated with the black hexagram and the green square, respectively (extracted from [22]).

On the suction side (see Figure 2.4a), the U_∞ method, the diagnostic diagram method and the local reconstruction method are provide consistent and accurate results. The results of the max method (i.e., mean-shear threshold method with zero threshold) are correct, except for the most downstream station, where the velocity profile increases monotonically. The $-y\Omega_z$ threshold method overestimates the boundary layer thickness, suggesting that the threshold proposed by Uzun and Malik [25] is not suitable for this case and it cannot be generalized.

In Figure 2.4b, the results for the flow on the pressure side are displayed. In this case, the results of the diagnostic plot method and the local-reconstruction method are able to consistently identify the boundary layer thickness at various stations. In contrast, the U_∞ method underestimates the edge of the boundary layer at the most upstream station and overpredicts it at the rest. The $-y\Omega_z$ threshold method also overestimates δ. Finally, as expected, the max method is not reliable, since several of the profiles are monotonically increasing.



(a) Suction side

(b) Pressure side

Figure 2.4: Distributions of the mean streamwise velocity (black solid lines) versus the wall-normal coordinate y/c for various profiles from a NACA 4412 at $AoA = 5^0$ and $Re_c = 10^6$. These wall-normal profiles originate from the airfoil surface at the streamwise stations $x/c = 0.20, 0.37, 0.55, 0.72, 0.90$ (from top left to bottom right). The estimates of the boundary-layer edges (and the corresponding edge velocities) are plotted with symbols as indicated in the legend (extracted from [22]).

2.7 Comparison of the methods

A summary of the limitations of the previously discussed methods is presened in Table 2.1.

Method	$\widetilde{U}_\infty$	$-y\Omega_z$	Mean shear	Diagnostic plot	Local-reconstruction
Integral Based	✓			✓	
Computes derivatives	✓	✓	✓	✓	
Empirical data		✓	✓	✓	
Iterative	✓		✓	✓	
High Re	✓	✓		✓	
Quiet free stream				✓	
Data apart from $U[y]$	Ω_z	Ω_z		$\sqrt{\overline{u'^2}}$	$V[y]$ $P[y]$
Total	5	4	3	7	1

Table 2.1: Undesirable features

The local-reconstruction method limits the number of such undesirable characteristics, making it the most robust and generalizable among the discussed options. The major drawback of this approach is the necessity of velocity and pressure profiles, besides the mean streamwise velocity $U[y]$. Nonetheless, similar shortcomings are experienced also by other methods, e.g., the mean vorticity Ω_z or the turbulent intensity $\sqrt{\overline{u'^2}}$. In addition, it is worth noticing that the required data on $V[y]$ and $P[y]$ are, in general, easily accessible.

Chapter 3

Numerical validation of the Local-reconstruction method

In order to generate a tailored mesh for the boundary layer, we first determine its thickness δ from a previous simulation employing the local-reconstruction method. In this chapter, a validation of this strategy and an assessment of its computational performance are presented.

3.1 δ determination

First of all, the workflow that is followed to calculate the thickness of the boundary layer of a point p_B at the boundary is explained:

1. Create a discretized line of arbitrary length passong through p_B in the normal direction to the boundary.
2. Interpolate the variables (P_0, p, V, and U) needed to determine the inviscid velocity U_I and the percentage of the mean streamwise velocity n, at the points of the discretized line p_{line} (for the details of the interpolation the reader is addressed to Appendix B).
3. Compute $P_{0,ref}$.
4. Compute $U_{I_{line}}$ at each p_{line} using (2.11).
5. Determine n_{line} at each p_{line} using (2.8).

At this point, we have a discretization of n as a function of the distance to the wall y (Figure 3.1). In order to determine δ, we could either take the value closest to the wall where $n > 99$ or interpolate from points just above and below $n = 99$. However, if the flow is not well resolved this can lead to false positives and, consequently, to an incorrect δ prediction. Instead, the points whose $y \leq y_{95}$, where y_{95} is the closest point to the wall where $n \geq 95$, are taken and fitted to a third-order curve with the method of least squares. Finally, from this fitted curve $y_{99} = \delta$ is extrapolated.

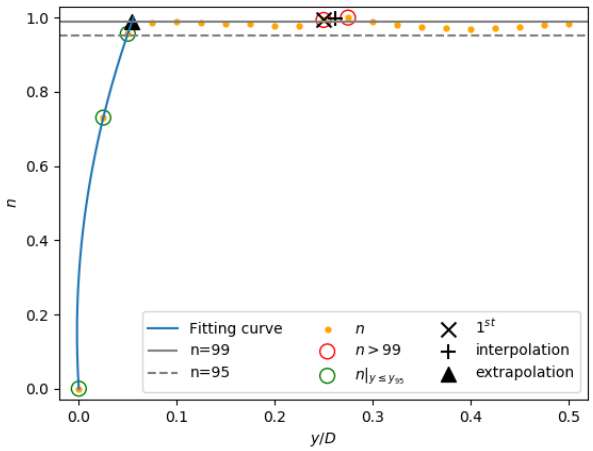


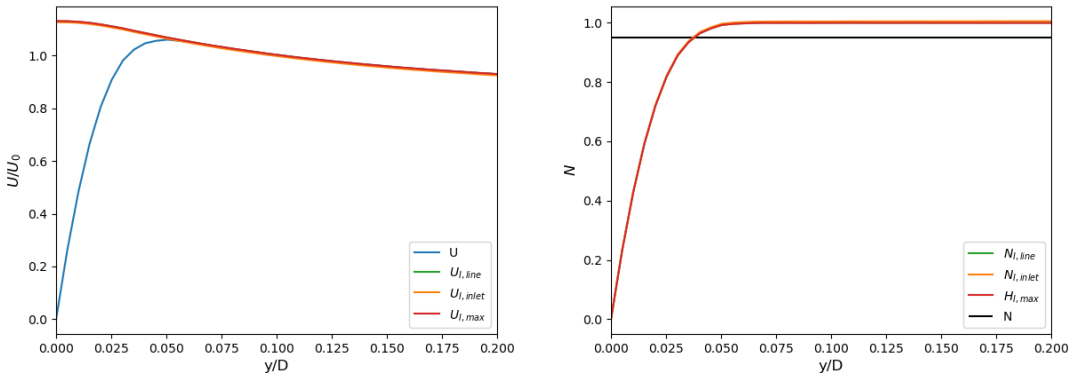
Figure 3.1: Distribution n as a function of the wall-normal distance normalized by the sphere diameter y/D. Profile of the flow around a sphere at the Reynolds number based on the diameter length $Re_D = 10^4$ at the $\boldsymbol{p}_B = (-0.215, -0.25, 0.375)$ location. Symbols indicate diferent ways of determine δ.

3.2 Reference total pressure $P_{0,ref}$

One of the key ingredients of this methodology is the computation of $P_{0,ref}$. Recall that $P_{0,ref} = \max(P_0)$, that is the total pressure achieved in the outer flow. The following strategies are available to determine $P_{0,ref}$:

- The maximum total pressure in a line of arbitrary length in the normal direction of the boundary point $P_{0,ref} = \max(P_{0,line})$.
- The maximum stagnation pressure in the whole domain $P_{0,ref} = \max(P_{0,global})$.
- The total pressure related to inlet boundary condition $P_{0,ref} = \max(P_{0,inlet})$.

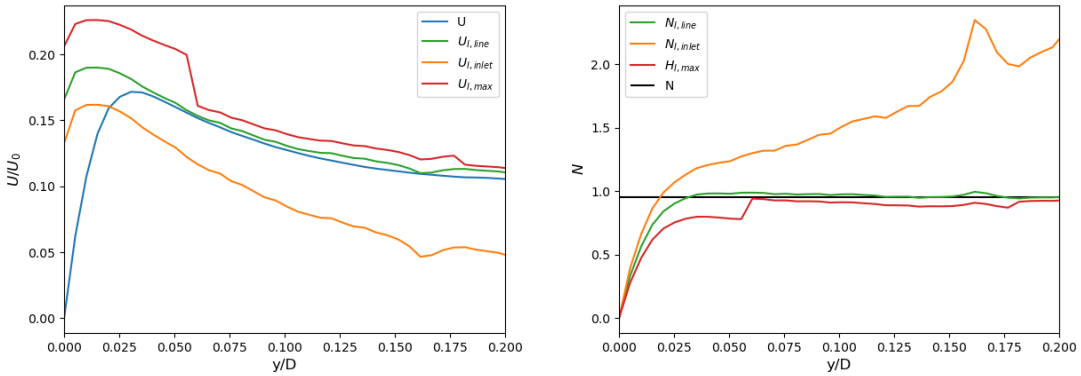
The flow around a sphere at the Reynolds number based on the diameter length $Re_D = 10^5$ is used to assess the above options to determine the reference total pressure.



(a) U and U_I versus y/D (b) n versus y/D

Figure 3.2: Comparison of the inviscid reconstructed flow for different $P_{0,ref}$ at location $\boldsymbol{x} = (-0.27, -0.39, 0.15)$

On the one hand, in Figure 3.2a we observe that the local reconstruction of the inviscid mean streamwise velocity U_I does not depend on the choice of $P_{0,ref}$ because the point lies in an area where the boundary layer is fully developed. Consequently, the percentage of the mean streamwise velocity n is not influenced by it either (Figure 3.2b).



(a) U and U_I versus y/D (b) n versus y/D

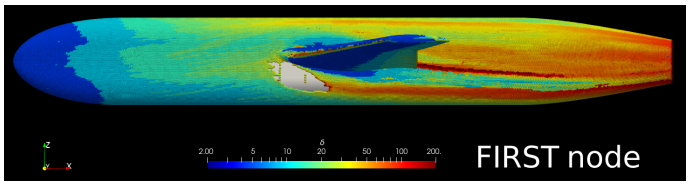
Figure 3.3: Comparison of the inviscid reconstructed flow for different $P_{0,ref}$ at location $\boldsymbol{x} = (0.49, 0.04, -0.06)$

On the other hand, Figure 3.3a, evaluated at a point located close to the stagnation region of the sphere, shows major discrepancies in the reconstruction of U_I and n. This disagreement is a consequence of an overprediction of $P_{0,ref}$ when $P_{0,ref} = \max(P_{0,global})$ and an underprediction when $P_{0,ref} = \max(P_{0,inlet})$. On the contrary, $P_{0,ref} = \max(P_{0,line})$ is more robust and provides a good prediction of the U_I. Hence, $P_{0,ref} = \max(P_{0,line})$ is the optimal choice and the one used in this work.

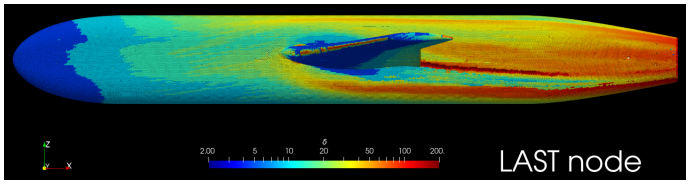
3.3 2D local reference frame

Apart from an appropriate estimation of $P_{0,ref}$, the definition of the 2D local frame of reference is required to determine $V[y]$, $p[y]$ and $U[y]$. In order to get $V[y]$ and $p[y]$, we have to rotate the velocity and pressure field in order to have the Y-component aligned with the normal of the boundary node from which we want to determine δ. However, for U we have to rotate the velocity field until the tangential component of the velocity is aligned with the X axis of the local reference system. Nonetheless, it is unclear with respect to which point it has to be rotated.

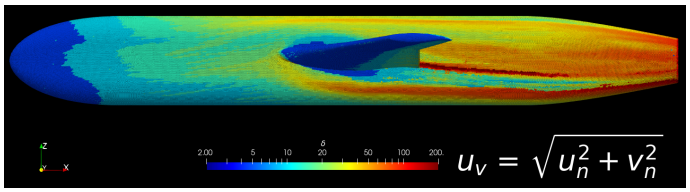
A priori, one would think that using the first out-wall node would be correct since is the approach that is normally used in wall modeling. However, Figure 3.4a shows a region of the fuselage near the leading edge and under the wing where the model is unable to predict δ. Alternatively, using the last point as reference for the rotation (Figure 3.4b), the previous region is correctly predicted. But, although it is an improvement over the previous estimation, it is a risky approach as the last point is arbitrarily chosen and could be close to another wall which would introduce a significant error. Additionally, these two options overpredict δ at the leading edge as well as fail to determine it at the wingtip.



(a) Rotation from first node



(b) Rotation from last node



(c) Magnitude of the tangential components of velocity

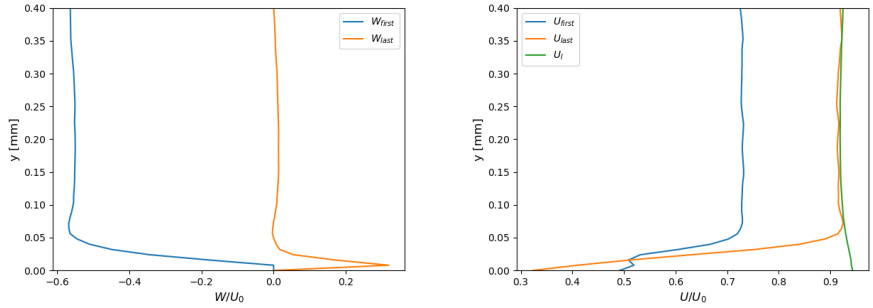
Figure 3.4: Boundary layer thickness δ over the surface of the NASA Juncture Flow geometry

In order to understand the reason why we are not able to predict δ in the region mentioned before, we plot the velocity profiles for both tangential components ($U[y]$ and $W[y]$) using the first and the last node as the reference for the rotation. As we can see in Figure 3.5a, $W \neq 0$, except in the point taken as base for the rotation, which calls into question the two-dimensionality of the boundary layer flow and shows that the flow is rotating around the normal axis inside it.

If we inspect in detail the evolution of W we see that in the outer region ($y/c \gtrapprox 0.07$) tends to a constant $W_{first} \approx -0.58$ and $W_{last} \approx 0$. The behavior inside the boundary layer is different in the two cases: for the first node, from zero on the wall, the velocity increases until it reaches its maximum at the edge of the boundary layer; for the last, from zero on the wall it goes to a maximum at the first out-wall node and then decreases back to zero at the edge.

On the one hand, all this adds up to a deficit in the magnitude of U_{first}, which is why no δ is found since it is very far from U_I (Figure 3.5b). On the other hand, W_{last} is practically zero at the edge, just the point we are looking for. However, near the wall there is a deficit of U_{last} (Figure 3.5b) which makes it inaccurate in thin boundary layers as the ones in the leading edge and wingtip.

Therefore, the last option is to rotate the tangential components of the velocity for each of the points on the line. Which can be computationally expensive depending on the number of points on the line. However, rotation per point is equivalent to calculating the magnitude of these components per point, which is much simpler and cheaper. This is the approach that has been followed in Figure 3.4c where it can be seen that the prediction of the boundary layer thickness in the leading edge and the wingtip is adequate.



(a) Secondary mean tangential velocity (W) versus y/mac

(b) Main mean tangential velocity (U) versus y/mac

Figure 3.5: Tangential velocity profiles of the flow around the Juncture flow at the angle of attack $(AoA) = 5^0$ and the Reynolds number $Re_c = 2.4 \cdot 10^6$ based on the crack chord length c at location $\boldsymbol{X}(2120.38, -232.15, -160.34)$ [mm]

3.4 Parallelization performance

Once clarified how to determine the data that is needed in a robust way, we can focus on the parallelization of the method. Until now we had assumed, for simplicity, that we wanted to obtain the boundary layer thickness at a point on the domain boundaries of a sequential numerical simulation, without partitioning. However, to reduce the time to solution, CFD runs are parallelised, which throws up a series of questions that have to be answer:

(i) how to identify the points of a discretized line generated from a node in a subdomain that belong to other subdomains, and how to do it for several lines,

(ii) how to reconstruct the line once the interpolation in those subdomains have been completed,

(iii) how to do it efficiently.

The easy way to solve (i) is to share all generated points with all subdomains. However, this approach is memory intensive as we will end up with

$$n_{points} = n_{bn} n_{sub} \tag{3.1}$$

per subdomain, with n_{bn}, and n_{sub} the total number of boundary nodes and the number of subdomains respectively. A workaround following this basis is to solve N portions, reducing the memory requireents by a factor N. After interpolating the points, they are distributed so that $n_{lines} = n_{bp}/(N n_{Sub})$ is the number of discretized lines per subdomain. This way the calculation of δ will be balanced (ii).

One drawback of the previously described approach is that when the points are gathered they are stored in a sequential way (see Figure 3.6), i.e., first the points from subdomain 1, then subdomain 2 up to domain N. Hence, when partitioning will be generating points from a specific range of subdomains and, consequently, only them an their neighbours will be working. If we take the partition from figure Figure 3.6, only the processes 1, 2 and 3 will be working whilst 4 and 5 will be

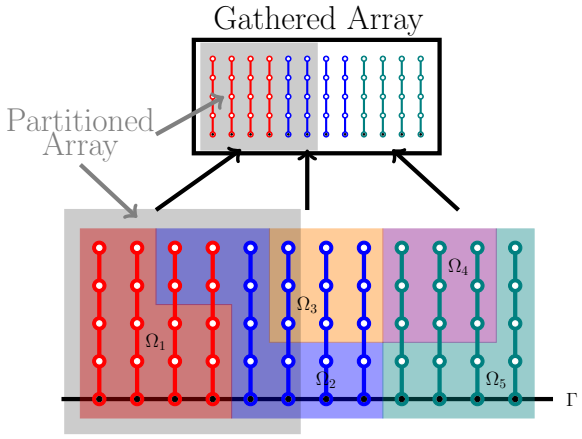
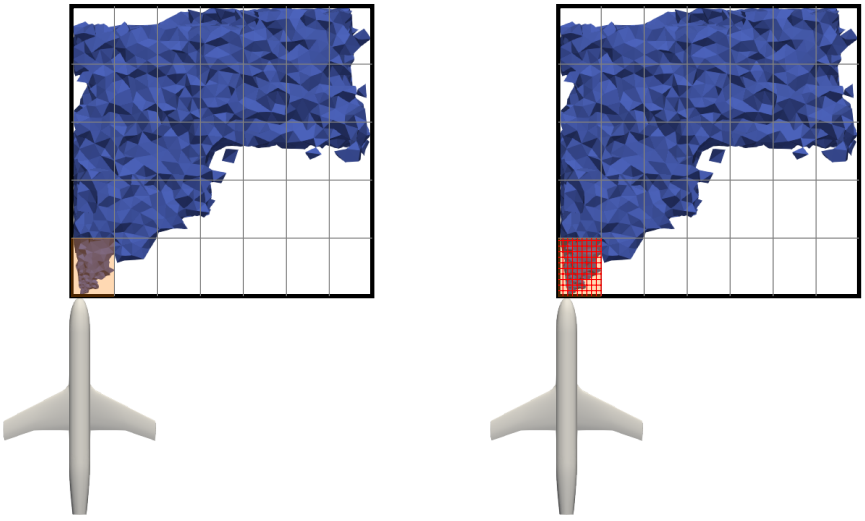


Figure 3.6: Partitioned points array

waiting. An improvement to the performance of the partitioned method is to mix the boundary points so that as many subdomains as possible work in each partition.

In Figure 3.7a, we have an example of the discretization of a subdomain for the search of elements (for details see Appendix A), where a box contains the elements close to the wall. This causes the search algorithm to slow down since the points that we generate will fall within it and there will be many elements to search for. Therefore, in cases where the box contains more than 200 elements, it is divided into subboxes using the 20th percentile of the size of the elements contained in the box (Figure 3.7b).

The ideal alternative to the partitioning method would be to send to each subdomain only the points which lie within the subdomain. This option cannot be carried out since the shape of the subdomain should be known in advance. However, we can determine a bounding box oriented with the principal axes. Then, each subdomain determines which of its points are inside each bounding box and sends it to the corresponding subdomain.



(a) non-optimal box (b) subboxes

Figure 3.7: Subdomain discrezation for element search

Case	Sphere		Juncture Flow	
n_{bn}	151,395		199,421	
	Time [min]	Speed up	Time [min]	Speed up
Partitioned	10.5	**-**	50	**-**
Subboxes	9.8	**1.07**	30.5	**1.64**
Shuffle	4.8	**2.19**	25.2	**1.98**
Subboxes & shuffle	4.9	**2.14**	15.6	**3.20**
Bounding boxes	3	**3.5**	13	**3.84**

Table 3.1: Performance of the δ calculation

In Table 3.1, the performance of each method commented before is shown. *Partitioned*, stands for the partition method. *Subboxes*, indicates that partition method includes the subboxes improvement. *Shuffle*, denote the boundary nodes have been mixed. Finally, *Bounding boxes*, refers to the method that distributes the points based on the bounding boxes of each subdomain. On the one hand, we can see how the shuffle option is the one that improves the partition method the most, although the subboxes also help. On the other hand, it is clear that the bounding boxes is the one that performs better.

Chapter 4

Application to a full aircraft configuration

The NASA Juncture Flow geometry [8] is based on the wing DLR-F6 and a leading-edge horn to mitigate the effect of the horseshoe vortex over the wing-fuselage juncture (Figure 4.1). The model wingspan is nominally 3397.2 mm, the fuselage length is 4839.2 mm, and the crank chord (chord length at the Yehudi break) is $c = 557.1$ mm.

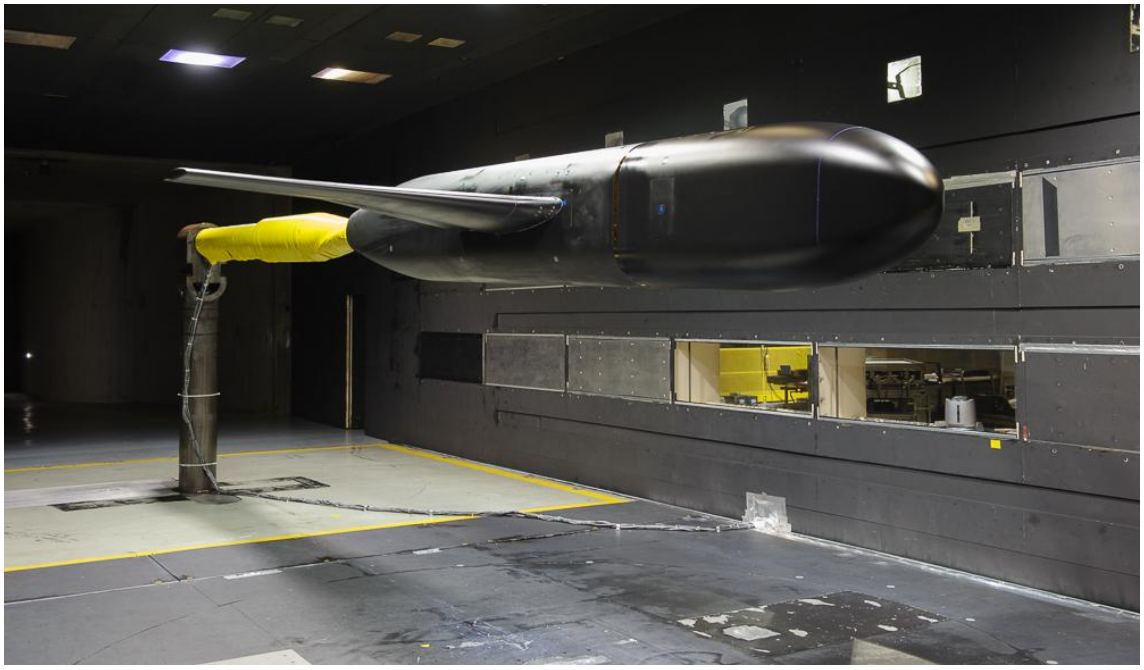


Figure 4.1: Wing-fuselage junction model installed in the 14- by 22-Foot Subsonic Wind Tunnel.

In the experiment, the model was tripped near the front of the fuselage and on the upper and lower surfaces of both wings. In our case, preliminary computations showed that tripping was also necessary to trigger the transition to turbulence over the wings. Hence, the geometry of the wing was modified by displacing in the z direction a line of surface mesh points close to the leading edge by 1 mm along the suction side and pressure side of the wing (Figure 4.2).

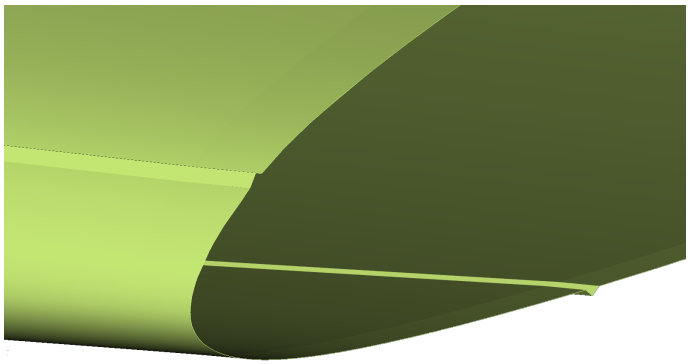


Figure 4.2: Wing tripping to trigger the transition to turbulence

4.1 Problem definition

The flow around the NASA Juncture Flow with a leading-edge horn is approximated with WMLES at $Re_c = 2.4$ million based on the crank chord length $c = 557.1$ mm and $AoA = 5^0$. The case is solved in a spherical domain of diameter $D_{farfield} = 5{\cdot}10^4$. A bulk velocity is imposed as the inflow boundary condition on the upstream half of the spherical domain and null traction as outflow condition on the downstream half. At the walls, we impose slip boundary conditions with the shear stress provided by the wall model as described in Section 4.2.

4.2 Numerical method

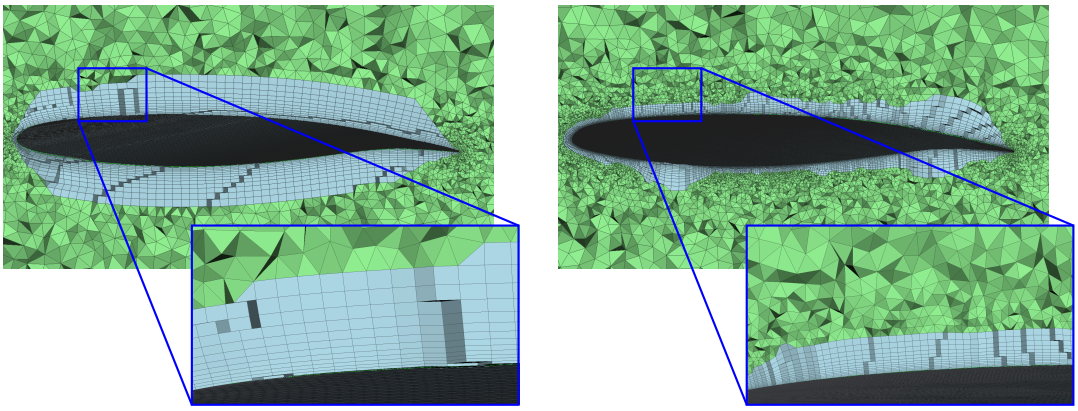
In order to solve the LES equations, the code Alya is used [29]. Alya is a low dissipation finite elements code, second order accurate in space [30]. The convective term is discretized using a Galerkin finite element scheme which preserves linear and angular momentum together with kinetic energy at a discrete level [31]. The pressure-velocity coupling is solved, using a fractional-step method [32], whereas equations are integrated in time using a semi-implicit approach, or IMEX, with a Crank-Nicholson method for the diffusive term and a third-order Runge-Kutta for the convective one, combined with an eigenvalue-based time-step estimator [33]. The Vreman subgrid-scale model [14] is used for turbulence closure with $c = 0.1$ (Equation 1.4).

A wall model is employed to reduce the restrictive spatial resolution requirements to resolve the small scales in the near-wall. The no slip boundary condition on the walls is replaced by a wall stress boundary condition obtained from the algebraic equilibrium wall model of Reichardt's law of the wall [34]. The velocity evaluation point for the wall function is located at a distance $h = 0.5N_{ExLoc} * \Delta_n$, where N_{ExLoc} is the exchange location multiplier.

4.3 Computational meshes

Two strategies were followed in order generate the unstructured hybrid grids:

(a) **Constant-height mesh** (Baseline). The grid size in the neighborhood of the plane surface is $\Delta_{t_1} \approx \Delta_{t_2} \approx \Delta_t \approx 10\Delta_n$, with Δ_t the tangential components,



(a) Baseline (b) Adaptive

Figure 4.3: Mesh topologies

Δ_{t_1} and Δ_{t_2}, and Δ_n the component normal to the surface. Additionally, the number of layers in the normal direction is specified such that the isotropic element, $\Delta_t \approx \Delta_n$, is reached with a growth ratio of 1.2. In addition, a refinement zone Δ_{wake} is defined around the wings in order to properly capture the wake development. Finally, a resolution of the farfield grid, $\Delta_{farfield} > \Delta_{wake}$, is set. Figure 4.3a illustrates the spacial discretization for $\Delta_n = 1$ mm, $\Delta_{wake} = 100$ mm, and $\Delta_{farfield} = 200$ mm.

The advantage of this method is its simplicity, however, it does not take into account the development of the boundary layer, which implies that the neighborhood of the leading edges are underresolved and as we move downstream we reach a point where the resolution of the mesh is excessive. Grid strategy (b) aims to lessen this problem.

(b) **Tailored boundary layer mesh** (Adaptive). In general aspects this strategy is very similar to strategy (a), however, the definition of Δ_n is no longer a constant but rather a function of the boundary layer thickness $\Delta_n = f(\delta)$. Consequently, the mesh of the vicinity of the walls varies depending on the flow (Figure 4.3b). In our application case, in order to have a sufficiently well-resolved solution at the exchange location point, $\Delta_n \approx 0.125\delta/3$ (Figure 4.4) has been chosen, where δ has been calculated employing the local-reconstruction method from a

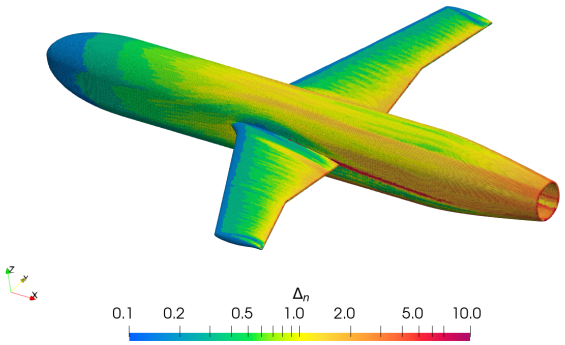
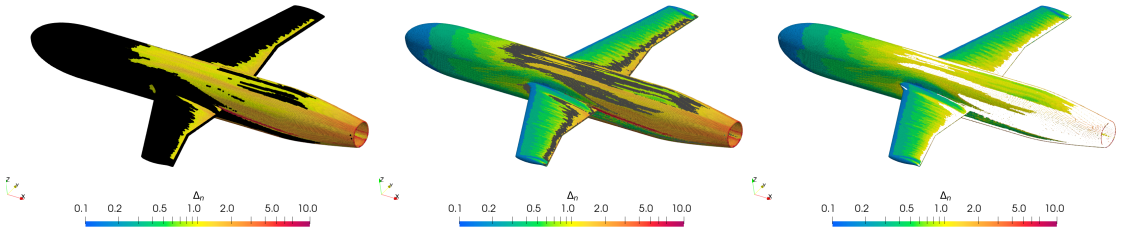


Figure 4.4: Grid resolution in the normal direction Δ_n



(a) Grid resolution regions underestimated (black)

(b) Grid resolution regions correct (gray)

(c) Grid resolution regions overestimated (white)

Figure 4.5: Resolution Comparison: Adaptive vs. Baseline

previous simulation using mesh strategy (a).

Figure 4.4 shows how the resolution required in the boundary layer should vary from 0.1 mm, at the nose and leading edges of the wings, to 10 mm, at the end of the fuselage. Moreover, Figure 4.5b highlights that only on a small region close to the trailing edges and the top of the fuselage were correctly discretized in the baseline mesh. On the one hand, in Figure 4.5c we can see that the trailing edges and the rear of the fuselage are overdiscretized, which implies a waste of nodes that could have been distributed more intelligently. On the other hand, the mesh on the front of the fuselage and in most of the wings is oversized, so it would have to be reduced, which would entail two counterpoints: a mesh with more unknowns and a smaller time step due to the explicit integration of the convective term. Depending on the characteristics of the flow in this area, the time step could be reduced in the same proportion as the mesh.

4.4 Results

In order to assess the performance of the tailored boundary layer mesh (adaptive) with respect to the constant-height mesh (baseline), the computational cost and the streamwise velocity profiles in different stations of the fuselage are compared.

4.4.1 Computational cost

Table 4.1 summarizes the number of nodes in the mesh, the number of CPUs used for the simulation, the mean time step (Δt), the approximated CPU cost per time step ($t_{\Delta CPU}$), the physical time needed to obtain a stationary solution (t_{ss}) and the cost of the simulation (T_{cc}).

As the number of unknowns triples, the computational time per time step increases accordingly as we use the same number of CPUs for both cases. Additionally, the time step

$$\Delta t = \mathcal{C}_{conv} \left| \frac{\Delta_i}{u_i} \right|_{min} \tag{4.1}$$

is reduced, to satisfy the convective CFL $\mathcal{C}_{conv} \leq 1$, since Δ_n is smaller in the adaptive mesh, to accommodate the thickness of the boundary layer at the leading edge. Furthermore, as smaller turbulent scales in the near-wall are being resolved,

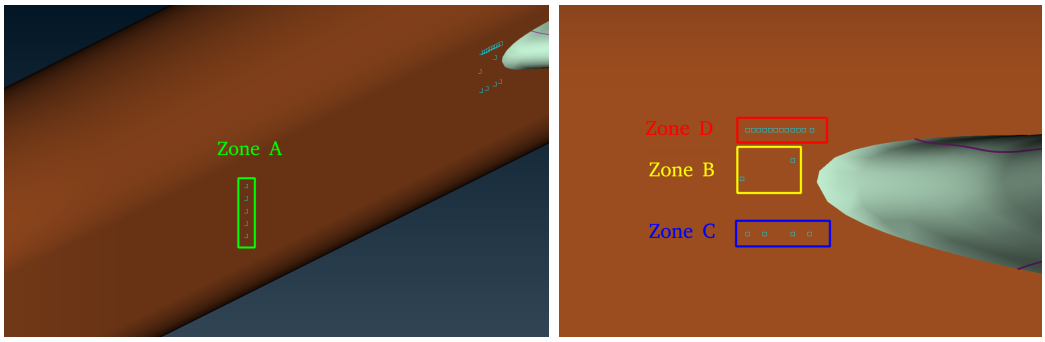
Mesh	# nodes	# CPUs	Δt[T.U's][1]	$t_{\Delta CPU}$[s]	t_{ss} [T.U's]	t_{cc} [CPUh]
Baseline	8.23 M	1,058	$4.462 \cdot 10^{-4}$	0.692	20	10 K
Adaptive	25.2 M	1,058	$7.831 \cdot 10^{-5}$	1.805	70	508 K

Table 4.1: Computational details of the simulations

the physical time to reach a steady state is increased. Therefore, the computational cost is massively increased because we have more unknowns, we take longer to solve them, we move forward in time more slowly and we need more time to reach the steady state.

4.4.2 Velocity profiles

Figure 4.6a and Figure 4.6b show four different zones where experimental measurements of the velocity profiles were taken at various stations (cyan squares). These measurements are taken as ground truth to assess the precision of the two meshes. Zone B is discarded since it only includes two points. In addition, the influence of the exchange location on the mesh will be evaluated by varying the multiplier, N_{ExLoc}, from 6 ($\approx 0.125\delta$) to 8 ($\approx 0.166\delta$).



(a) Upstream region of the fuselage (b) Wing-body juncture leading edge

Figure 4.6: Velocity profile stations at the plane fuselage

The absolute and relative $\mathcal{L}_2$ errors

$$\Delta\varepsilon_{\mathcal{L}_2} = \sum_{i=1}^{n_{points}} \sqrt{(u_i^{num} - u_i^{exp})^2} \tag{4.2}$$

$$\varepsilon_{\mathcal{L}_2} = \frac{\sum_{i=1}^{n_{points}} \sqrt{(u_i^{num} - u_i^{exp})^2}}{\sum_{i=1}^{n_{points}} \sqrt{(u_i^{exp})^2}} \tag{4.3}$$

are summarized in the Table 4.2, where u^{num} and u_i^{exp} are the numerical and experimental solution respectively.

On the one hand, clearly, zone A is better predicted by the adaptive (‡), with $\varepsilon_{\mathcal{L}_2} \lessapprox 3\%$, where the assumption of flow in equilibrium is more representative. On

[1]Time Units (T.U.): time that takes the flow at the inlet velocity to travel the crank chord length.

Id	Zone	X	Z	Baseline (§)		Adaptive (†)		Adaptive (‡)[2]		Best
				$\Delta\varepsilon_{\mathcal{L}_2}$	$\varepsilon_{\mathcal{L}_2}[\%]$	$\Delta\varepsilon_{\mathcal{L}_2}$	$\varepsilon_{\mathcal{L}_2}[\%]$	$\Delta\varepsilon_{\mathcal{L}_2}$	$\varepsilon_{\mathcal{L}_2}[\%]$	
1	A	1168.4	0	3.07	3.37	4.61	5.05	2.78	3.04	‡
2	A	1168.4	-30	5.88	6.47	3.77	4.15	2.75	3.03	‡
3	A	1168.4	30	5.09	5.56	3.52	3.85	2.25	2.46	‡
4	A	1168.4	-60	3.77	4.16	2.56	2.83	2.56	2.82	‡
5	A	1168.4	60	4.93	5.38	4.51	4.93	2.20	2.40	‡
6	C	1864.2	6.2	3.13	4.27	4.51	6.14	3.95	5.39	§
7	C	1879.2	6.2	2.78	3.96	3.12	4.43	4.88	6.93	§
8	C	1904.2	6.2	3.61	5.67	4.31	6.76	6.03	9.48	§
9	C	1919.2	6.2	5.28	9.11	6.32	10.90	7.35	12.68	§
10	D	1864.2	98.05	2.63	3.16	2.45	2.95	2.98	3.58	†
11	D	1869.2	98.05	2.79	3.37	2.40	2.90	2.98	3.60	†
12	D	1874.2	98.05	3.08	3.72	2.63	3.18	2.86	3.45	†
13	D	1879.2	98.05	3.19	3.87	2.82	3.42	2.93	3.56	†
14	D	1884.2	98.05	3.33	4.05	3.03	3.69	3.00	3.66	‡
15	D	1889.2	98.05	3.43	4.18	3.31	4.03	3.01	3.67	‡
16	D	1894.2	98.05	3.48	4.25	3.16	3.87	3.21	3.93	†
17	D	1899.2	98.05	3.43	4.20	3.07	3.76	3.22	3.95	†
18	D	1904.2	98.05	3.54	4.33	3.11	3.80	3.52	4.31	†
19	D	1909.2	98.05	3.62	4.46	2.84	3.50	3.79	4.67	†
20	D	1914.2	98.05	3.71	4.57	2.80	3.45	4.09	5.03	†
21	D	1921.2	98.05	4.06	4.91	3.15	3.86	4.44	5.44	†

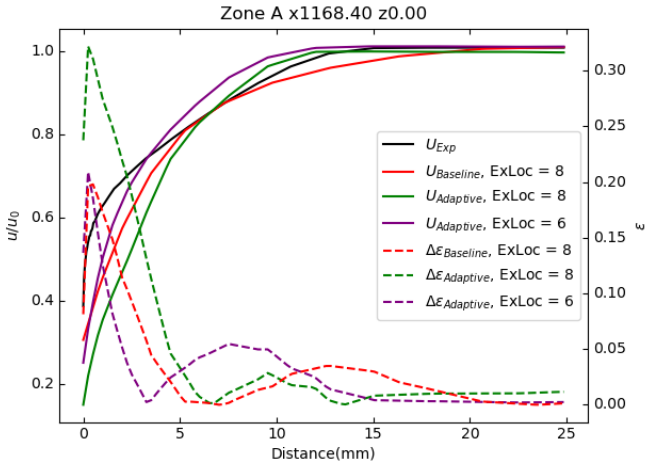
Table 4.2: $\mathcal{L}_2$ error at the different stations

the other hand, zones C and D are better approximated by the baseline (§) and the adaptive (†), respectively. Since the flow goes from bottom to top due to the AoA, it has been influenced by the pressure gradient generated by the rounding of the fuselage. Hence, the errors are larger in these zones with respect to zone A because the equilibrium hypothesis is no longer valid. As we move away from the rounding, the influence of the pressure gradient vanishes and the equilibrium hypothesis becomes acceptable again, which is why the adaptive meshes predict a better solution, (†) slightly better. It is worth noting that as we get closer to the leading edge the approximations get worse as other pressure gradients come into play.

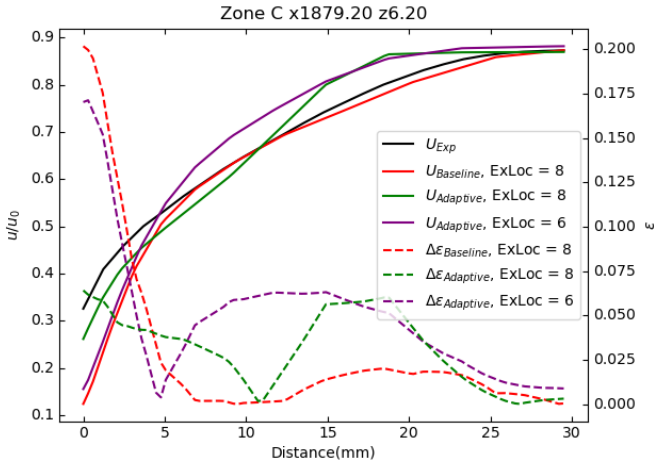
In Figure 4.7 one representative station of each zone is selected, where we can observe that:

- Adaptive (‡) predicts consistently a better solution close to the wall and over-predicts the flow in the transition region between the inner and outer flow.
- Adaptive (†) offers a better estimate in the transitional region than adaptive (‡).
- The three approaches are practically equivalent in the outer flow.

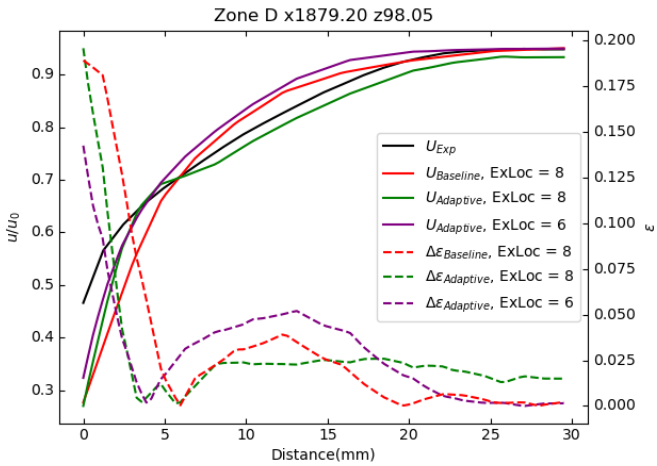
[2] $N_{ExLoc} = 6$, for the other cases $N_{ExLoc} = 8$



(a) Id. 1 (Zone A)



(b) Id. 7 (Zone C)



(c) Id. 13 (Zone D)

Figure 4.7: Velocity and $\varepsilon_{\mathcal{L}_2}$ profiles at different stations

Conclusions

After evaluating several methods to determine the thickness of the boundary layer, it was concluded that the most robust and general method was the local-reconstruction of the inviscid streamwise velocity method. Apart from requiring less statistics for its implementation, it offers a better solution, qualitatively speaking, besides being applicable independently to laminar, turbulent and transitional flows. This method was successfully implemented after validation of its robustness and efficiency.

During the validation, the method showed sensitivity to the choice of the maximum stagnation pressure as well as to the transformation of the 2D reference system. It offers increased robustness in the case that $P_{0,ref}$ is evaluated in each discrete normal line, as well as when the streamwise velocity is computed as the magnitude of the tangential components. Additionally, points below 95% of the boundary layer edge velocity are fit to a curve from which δ is extrapolated. In this way, overprediction is avoided when the resolution of the statistics is too coarse.

As the last point of validation, parallel computing efficiency was assessed. The best performance was obtained when the points were communicated to each subdomain based on their bounding boxes.

The main conclusion we can draw from the results of the NASA Juncture Flow WMLES approximations is that wall modeling works well in regions where the flow resembles a boundary layer in equilibrium. However, solution quality degrades in the presence of pressure gradients and wing-body junctions. Therefore, a non-equilibrium wall model must be explored for flows with these characteristics.

Additionally, the velocity evaluation point has an impact on the solution. Hence, an adaptive exchange location method based on the boundary layer thickness should be investigated as an alternative to the multiplier one.

The above conclusions are based on the results of the velocity profiles. Their impact on predicting skin friction is uncertain due to the lack of experimental measurements, although they are expected to propagate to wall stress.

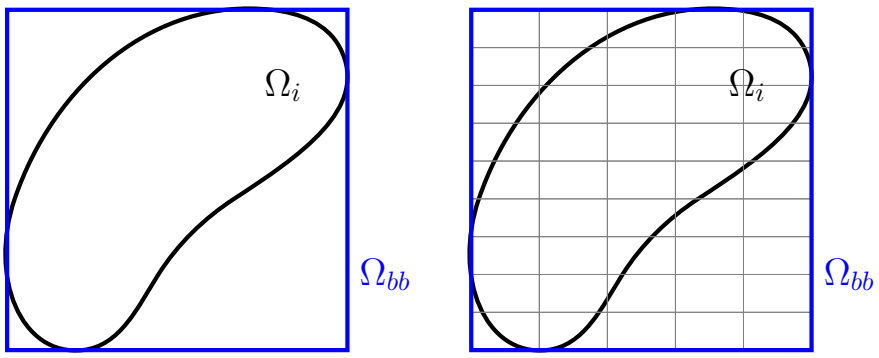
Finally, the slow convergence of the tailored mesh solution significantly increases the computational cost. One option to reduce the computational cost is to increase the mesh size in the laminar area of the leading edge. Error will be introduced because the flow will be underresolved but the turbulent wall law assumption is incorrect as well. Therefore, the time step can be increased without violating the CFL condition.

Appendix A

Element search algorithm

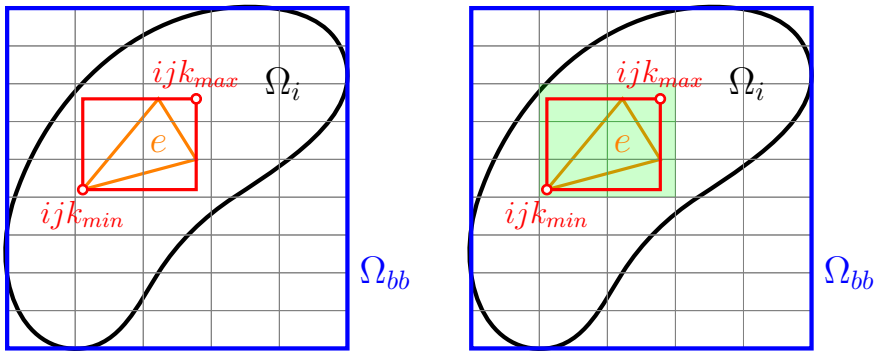
In order to interpolate at a point p the values of a field, the element host must be identified. We could check element by element if the point is contained but obviously this would be highly inefficient. Instead, the bounding box of each subdomain is partitioned into orthogonal boxes, each of which is filled with the elements that have a minimum portion within it. So, knowing the length of the partition in each direction (x,y,z), we can obtain an index (i,j,k) of the point that will give us the box that contains the candidate elements to host the point. In this way, the search time is drastically reduced because the box includes a subset of all the elements. The algorithm is the following:

1. Create a bounding box Ω_{bb} of the subdomain Ω_i (Figure A.1a).
2. Partition the bounding box N_i times in each direction (Figure A.1b).
3. Loop over elements:
   (a) Generate a bounding box, e_{bb}, for element e.
   (b) Determine the index ijk of the maximum and minimum e_{bb} nodes coordinates (Figure A.2a).
   (c) Include e in the boxes from ijk_{min} to ijk_{max} (Figure A.2b).
4. Loop over points:



(a) Bounding box (b) Subdomain partitioning

Figure A.1: Generation of the orthogonal discretization of the subdomain



(a) Element bounding box and indices (red)

(b) Boxes (green) containing the element e

Figure A.2: Boxes filling

(a) Calculate the index p_{ijk} of the point p.

(b) Loop over elements in box b_{ijk}:

i. Split elements into tetrahedrons, pyramids (2), wedges (3), and hexahedrons (5).

ii. Loop over tetrahedrons:

A. Transform global coordinates of point $p(x)$ to the natural coordinates of the tetrahedron $p(\xi)$.

B. Check if $p(\xi)$ is inside the tetrahedron.

$$p(\xi_i) \leq 1 \tag{A.1}$$

$$\sum_{i=1}^{n_{sd}} p(\xi_i) \leq 1 \tag{A.2}$$

with n_{sd} the number spatial dimensions.

Appendix B

Interpolation

The field value $u(p)$ is estimated from the node values $u(\boldsymbol{n}^{e_p})$ of the element containing the data point p, e_p (host element), with the shape functions, N_i,

$$u(p) = \sum_{i=1}^{n_{nodes}^{e_p}} N_i(\xi_p) u(n_i^{e_p}) \tag{B.1}$$

Then, we have to calculate the local coordinates of a point, which is a non-trivial operation, requiring two steps. First, finding the host element (Appendix A). Second, calculating the local coordinates of the point (inverse mapping), which involves solving a multi-dimension non-linear system of equations. Hence, to evaluate the shape functions coefficients it is necessary to compute $\xi(p)$ from $x(p)$ the inverse of the mapping algorithm.

$$x(p) = \sum_{i=1}^{n_{nodes}^{e_p}} N_i(\xi_p) x(n_i^{e_p}) \tag{B.2}$$

By means of Newton-Raphson iterations with residual

$$r(\xi) = x(p) - \sum_{i=1}^{n_{nodes}^{e_p}} N_i(\xi_p) x(n_i^{e_p}), \tag{B.3}$$

given an initial guess ξ_p^0 , set $k = 0$ and iterate:

1. Compute the Jacobian matrix

$$\boldsymbol{J}(\xi_p^k)_{ij} = \frac{\partial r_i(\xi_p^k)}{\partial \xi_j} = -\sum_{i=1}^{n_{nodes}^{e_p}} \frac{\partial N_i}{\partial \xi_j}(\xi_p^k) x(n_i^{e_p}), \tag{B.4}$$

   and the residual $r^k(\xi_p^k)$ using the last computed solution ξ_p^k

2. Compute the new variation $\Delta\xi_p^{k+1}$ of the solution solving

$$\boldsymbol{J}(\xi_p^k)\Delta\xi_p^{k+1} = -r^k(\xi_p^k) \tag{B.5}$$

3. Update the solution

$$\xi_p^{k+1} = \xi_p^k + \Delta\xi_p^{k+1} \tag{B.6}$$

4. If not converged go to 1.

5. Evaluate Equation B.1 using ξ_p^{k+1}.

Bibliography

[1] Jeffrey P Slotnick, Abdollah Khodadoust, Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, and Dimitri J Mavriplis. Cfd vision 2030 study: a path to revolutionary computational aerosciences. Technical report, 2014.

[2] Adam M Clark, Jeffrey P Slotnick, Nigel J Taylor, and Christopher L Rumsey. Requirements and challenges for cfd validation within the high-lift common research model ecosystem. In *AIAA AVIATION 2020 FORUM*, page 2772, 2020.

[3] Giancarlo Alfonsi. Reynolds-averaged navier–stokes equations for turbulence modeling. *Applied Mechanics Reviews*, 62(4), 2009.

[4] Philippe R Spalart. Detached-eddy simulation. *Annual review of fluid mechanics*, 41(1):181–202, 2009.

[5] Yang Zhiyin. Large-eddy simulation: Past, present and the future. *Chinese journal of Aeronautics*, 28(1):11–24, 2015.

[6] Ugo Piomelli and Elias Balaras. Wall-layer models for large-eddy simulations. *Annual review of fluid mechanics*, 34(1):349–374, 2002.

[7] Soshi Kawai and Johan Larsson. Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy. *Physics of Fluids*, 24(1):015105, 2012.

[8] Christopher L. Rumsey, Jan Carlson, and Nashat Ahmad. Fun3d juncture flow computations compared with experimental data. In *AIAA Scitech 2019 Forum*, page 0079, 2019.

[9] Henry Lee, Thomas H Pulliam, Christopher L Rumsey, and Jan-Renee Carlson. Simulations of the nasa langley 14-by 22-foot subsonic tunnel for the juncture flow experiment. Technical report, 2018.

[10] Henry C Lee and Thomas H Pulliam. Overflow juncture flow computations compared with experimental data. In *AIAA Scitech 2019 Forum*, page 0080, 2019.

[11] Adrián Lozano-Durán, Sanjeeb T Bose, and Parviz Moin. Performance of wall-modeled les for external aerodynamics in the nasa juncture flow. *arXiv preprint arXiv:2101.00331*, 2021.

[12] Roel Verstappen. When does eddy viscosity damp subfilter scales sufficiently? *Journal of Scientific Computing*, 49(1):94–110, 2011.

[13] Stephen B Pope and Stephen B Pope. *Turbulent flows*. Cambridge university press, 2000.

[14] AW Vreman. An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of fluids*, 16(10):3670–3681, 2004.

[15] Massimo Germano, Ugo Piomelli, Parviz Moin, and William H Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, 1991.

[16] Joan Calafell Sandiumenge. Efficient wall modeling for large eddy simulations of general non-equilibrium wall-bounded flows. 2019.

[17] Th Karman. von: Mechanische ähnlichkeit und turbulenz, nachr. ges. wiss. göttingen, math.-phys. kl.(1930) 58–76'. *Proc. 3. Int. Cong. Appl. Mech*, 322, 1930.

[18] Sergio Hoyas and Javier Jiménez. Scaling of the velocity fluctuations in turbulent channels up to re τ= 2003. *Physics of fluids*, 18(1):011702, 2006.

[19] Juan A Sillero, Javier Jiménez, and Robert D Moser. One-point statistics for turbulent wall-bounded flows at reynolds numbers up to δ+ 2000. *Physics of Fluids*, 25(10):105102, 2013.

[20] Junsun Ahn, Jae Hwa Lee, Jin Lee, Ji-hoon Kang, and Hyung Jin Sung. Direct numerical simulation of a 30r long turbulent pipe flow at re τ= 3008. *Physics of Fluids*, 27(6):065110, 2015.

[21] Herbert Owen, Georgios Chrysokentis, Matias Avila, Daniel Mira, Guillaume Houzeaux, Ricard Borrell, Juan Carlos Cajas, and Oriol Lehmkuhl. Wall-modeled large-eddy simulation in a finite element framework. *International Journal for Numerical Methods in Fluids*, 92(1):20–37, 2020.

[22] Kevin Patrick Griffin, Lin Fu, and Parviz Moin. General method for determining the boundary layer thickness in nonequilibrium flows. *Physical Review Fluids*, 6(2):024608, 2021.

[23] Michael James Lighthill. Introduction: Boundary layer theory. In Louis Rosenhead, editor, *Laminar Boundary Layers*, chapter II, page 687. Oxford University Press, 1963.

[24] GN Coleman, CL Rumsey, and PR Spalart. Numerical study of turbulent separation bubbles with varying pressure gradient and reynolds number. *Journal of Fluid Mechanics*, 847:28–70, 2018.

[25] Ali Uzun and Mujeeb R Malik. Simulation of a turbulent flow subjected to favorable and adverse pressure gradients. *Theoretical and Computational Fluid Dynamics*, 35(3):293–329, 2021.

[26] Donald Coles. The law of the wake in the turbulent boundary layer. *Journal of Fluid Mechanics*, 1(2):191–226, 1956.

[27] Ricardo Vinuesa, Alexandra Bobke, Ramis Örlü, and Philipp Schlatter. On determining characteristic length scales in pressure-gradient turbulent boundary layers. *Physics of fluids*, 28(5):055101, 2016.

[28] Artur Dróżdż, Witold Elsner, and Stanisław Drobniak. Scaling of streamwise reynolds stress for turbulent boundary layers with pressure gradient. *European Journal of Mechanics-B/Fluids*, 49:137–145, 2015.

[29] Mariano Vázquez, Guillaume Houzeaux, Seid Koric, Antoni Artigues, Jazmin Aguado-Sierra, Ruth Arís, Daniel Mira, Hadrien Calmet, Fernando Cucchietti, Herbert Owen, et al. Alya: Multiphysics engineering simulation toward exascale. *Journal of computational science*, 14:15–27, 2016.

[30] Oriol Lehmkuhl, Guillaume Houzeaux, Herbert Owen, Giorgios Chrysokentis, and Ivette Rodríguez. A low-dissipation finite element scheme for scale resolving simulations of turbulent flows. *Journal of Computational Physics*, 390:51–65, 2019.

[31] Sergey Charnyi, Timo Heister, Maxim A Olshanskii, and Leo G Rebholz. On conservation laws of navier–stokes galerkin discretizations. *Journal of Computational Physics*, 337:289–308, 2017.

[32] Ramón Codina. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics*, 170(1):112–140, 2001.

[33] FX Trias and O Lehmkuhl. A self-adaptive strategy for the time integration of navier-stokes equations. *Numerical Heat Transfer, Part B: Fundamentals*, 60(2):116–134, 2011.

[34] Hans 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.