香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Department of Mechanical and Aeronautical Engineering

Laboratory 1

Numerical simulation of the steady flow around a NACA-0012 airfoil at $\mathbf{R_e = 2.0 \times 10^6}$

MECH-4830 Introduction to Aerospace CFD

Author: Selim SHERIF
Lecturer : Prof Larry Li
May 10, 2024

Contents

1 Case Setup and Parameters **1**
1.1 Overview . . . 1
1.1.1 Brief Introduction . . . 1
1.1.2 Schematic and Case summary . . . 1
1.2 Governing equations and Underlying Assumptions . . . 2
1.3 CFD Solver Choice . . . 3
1.4 Simulation properties and parameters . . . 3
1.4.1 Fluid and Flow Properties . . . 3
1.4.2 Initial Conditions and boundary Conditions . . . 4

2 Effect of Computational Domain Size **4**

3 Grid Convergence Test and Mesh Selection **8**

4 Effect of Relaxation Factor **12**
4.1 Understanding the Relaxation Factor . . . 12
4.2 Incorporating the Relaxation Factor: Testing and Analysis . . . 12

5 Effect of Numerical Scheme **15**
5.1 Understanding Linear Vs Upwind Vs LinearUpwind Schemes . . . 15
5.2 Application of Schemes and Comparison . . . 16
5.2.1 Linear Scheme . . . 16
5.2.2 Upwind Scheme . . . 17

6 Lift and Drag Curve for $-5^\circ < \alpha < 15^\circ$ **18**

7 Conclusions and Final Verdict **21**

8 Appendix **21**

References

1 Case Setup and Parameters

1.1 Overview

1.1.1 Brief Introduction

The primary goal of this lab experiment is to conduct a numerical simulation using OpenFOAM Version 2012. The focus will be placed on a 2D Airfoil of NACA Type-0012, maintaining a constant Reynolds number throughout the simulation. The main objective is to investigate the fluid flow around the wing at different angles of attack. Additionally, by adjust parameters such as schemes and relaxation factors, the aim is to study the different CFD behaviors and to gain a deeper understanding of Computational Fluid Dynamics principles. Furthermore, the results obtained will compared with reliable real-world data to ensure their correctness, consistency, and validity.

1.1.2 Schematic and Case summary

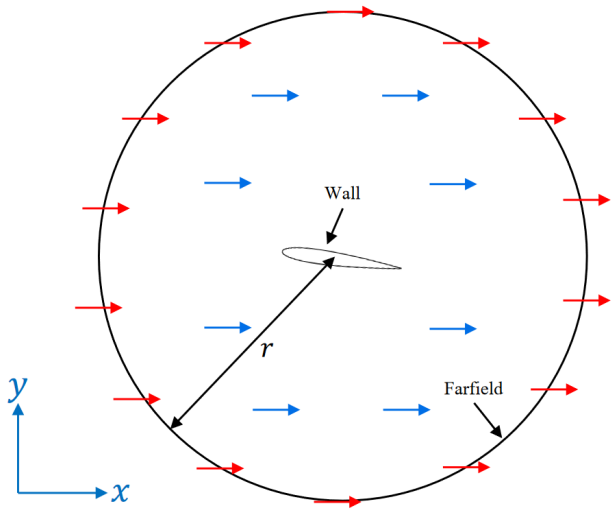


Figure 1: Case Setup

The case setup is relatively straightforward. It involves a circular O-mesh with a radius r, encompassing an airfoil positioned at its center (the origin), appropriately tilted based on the angle of attack. The airfoil is defined by a 'noslip' wall ensuring no relative motion at its surface. Conversely, the circular periphery (farfield) of the circle is designated as an 'InletOutlet' boundary.
Initially, the inner flow is established with a uniform velocity in the x-direction across all grid points. Additionally, the farfield boundary is set with a fixed velocity vector, also in the x-direction, serving as a boundary condition. The Software used in this case to create the 2D airfoil is the Linux Construct2D grid generator [1]

1.2 Governing equations and Underlying Assumptions

The general equation of a generic fluid flow are the following:

Mass conservation: Continuity Equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

Momentum Conservation: Navier-Stokes Equation

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{f}$$

Energy Conservation: First Law of Thermodynamics

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) = -\nabla \cdot (p \mathbf{u}) + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{u}) + \nabla \cdot \mathbf{q} + \rho \mathbf{u} \cdot \mathbf{f} + \mathbf{f} \cdot \mathbf{u}$$

In this specific case the flow is defined around a 2D airfoil flying at constant speed. Therefore, the equations can be greatly simplified with the following assumptions:

- **2D Flow:** All terms with $\frac{\partial}{\partial z}$ are equal to 0.
- **Steady Flow** All terms with $\frac{\partial}{\partial t}$ are equal to 0.
- **Isothermal Conditions** We are neglecting thermal effects. Therefore, no need to use the energy equation.
- **Incompressible flow** This means that $\rho = cst$ and $\nabla \rho = 0$ which further simplifies the equations.

With these assumptions the governing equations reduce to:

Continuity Equation (2D Incompressible, Steady):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

Momentum Equations (2D Incompressible, Steady):

- **x-direction momentum:**

$$\rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) = -\frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

- **y-direction momentum:**

$$\rho \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) = -\frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right)$$

1.3 CFD Solver Choice

Looking at the assumptions and the equations to be solved, the simpleFOAM solver from OpenFOAM seems like a solid choice.
The official description : *simpleFoam is a steady-state solver for incompressible, turbulent flow, using the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm.*
It also solves the following equations in the background: [2]

The solver employs the **SIMPLE** algorithm to solve the continuity equation:

$$\nabla \cdot \mathbf{u} = 0$$

and momentum equation:

$$\nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \mathbf{R} = -\nabla p + \mathbf{S}_u$$

Where:

$\mathbf{u}$	=	Velocity
p	=	Kinematic pressure
$\mathbf{R}$	=	Stress tensor
$\mathbf{S}_u$	=	Momentum source

Very consistent with what has been derived and a very cohesive choice.

1.4 Simulation properties and parameters

1.4.1 Fluid and Flow Properties

The environmental conditions imposed for this simulation are the ambient conditions. Namely, atmospheric pressure and temperature of $T = 15°\text{C}$
This yields the following properties for air:

Table 1: Fluid, Flow and Turbulence Properties

Parameter	**Value**	**Description**
Fluid And Flow Properties		
Density	$\rho = 1.225$ kg/m^3	Air density
Viscosity	$\mu = 1.81 \times 10^{-5}$ Pa·s	Dynamic viscosity
Kinematic Viscosity	$\nu = 1.48 \times 10^{-5}$ m^2/s	Kinematic viscosity
Temperature	$T = 288.15$ K	Flow temperature
Reynolds number	$R_e = 2.0 \times 10^6$	Reynolds number Imposed
Turbulence Model	SpalartAllmaras	Classic SA TM

The turbulence model utilized here is the SA turbulence model which is a fairly good model for airfoil shapes and the imposed Reynolds number is of $R_e = 2.0 \times 10^6$

1.4.2 Initial Conditions and boundary Conditions

Another major important set of features that needs to be customized and choosen in order to start simulating are the boundary and initial conditions. The initial velocity can easily be deduced via the Reynolds number (Re):

$$Re = \frac{\rho \cdot v \cdot L}{\mu}$$

Isolating velocity (v):

$$v = \frac{Re \cdot \mu}{\rho \cdot L}$$

$$v \approx 29.4\,\mathrm{m\,s^{-1}}$$

With this in mind, the following table summarizes all of the boundary and initial conditions.2

2 Effect of Computational Domain Size

In CFD simulations, the computational domain size significantly influences the accuracy and efficiency of the results. When the computational size is too small, boundary conditions can disproportionately affect the flow, leading to inaccuracies in the simulation. Conversely, if the computational domain is excessively large then a lot of computational power is being used but is not useful, rendering the simulation inefficient. Therefore, balancing these factors is crucial to achieving reliable and efficient CFD results.
To address the challenge, a proposal is made to conduct an experiment investigating the influence of mesh size on simulation outcomes, utilizing the NACA-0012 airfoil geometry. The objective is to determine the optimal balance between computational accuracy and

Table 2: Initial and Boundary Conditions

Parameter	**Value**	**Description**
Initial Conditions		
internalField		
Initial Velocity	Uniform $U_{\mathrm{i}}(0) = (29.4, 0, 0)$ m/s	Uniform unidirectional initial internal velocity field
Initial Pressure	Uniform $P_{\mathrm{i}}(0) = 0$ Pa	Zero initial relative internal pressure
Boundary Conditions		
Farfield		
Boundary Type	Patch	
Velocity BC	InletOutlet $U_{\mathrm{ff}} = U_{\mathrm{i}}(t)$	Inlet & outlet velocity same as local internal velocity
Pressure BC	InletOutlet $P_{\mathrm{ff}} = P_{\mathrm{i}}(t)$	Inlet & outlet pressure same as local internal pressure
Walls		
Boundary Type	Wall	No fluid can go through airfoil
Pressure BC	zeroGradient	Zero pressure gradient normal to airfoil
Velocity BC	noSlip	Zero velocity at airfoil boundary
frontandBack		
Boundary Type	Empty	Equations not to be solved across front and back planes

efficiency by altering the computational domain size. Multiple simulations with varying computational domains will be conducted, and the resulting data will be analyzed to deduce the ideal computational domain size.
In this experiment, the radius parameter, which is the primary factor affecting the domain size in the grid, will be varied. Specifically, the following radius sizes will be explored:

Table 3: Mesh Radii Sizes

Radii Sizes (m)						
2	5	10	15	20	30	50

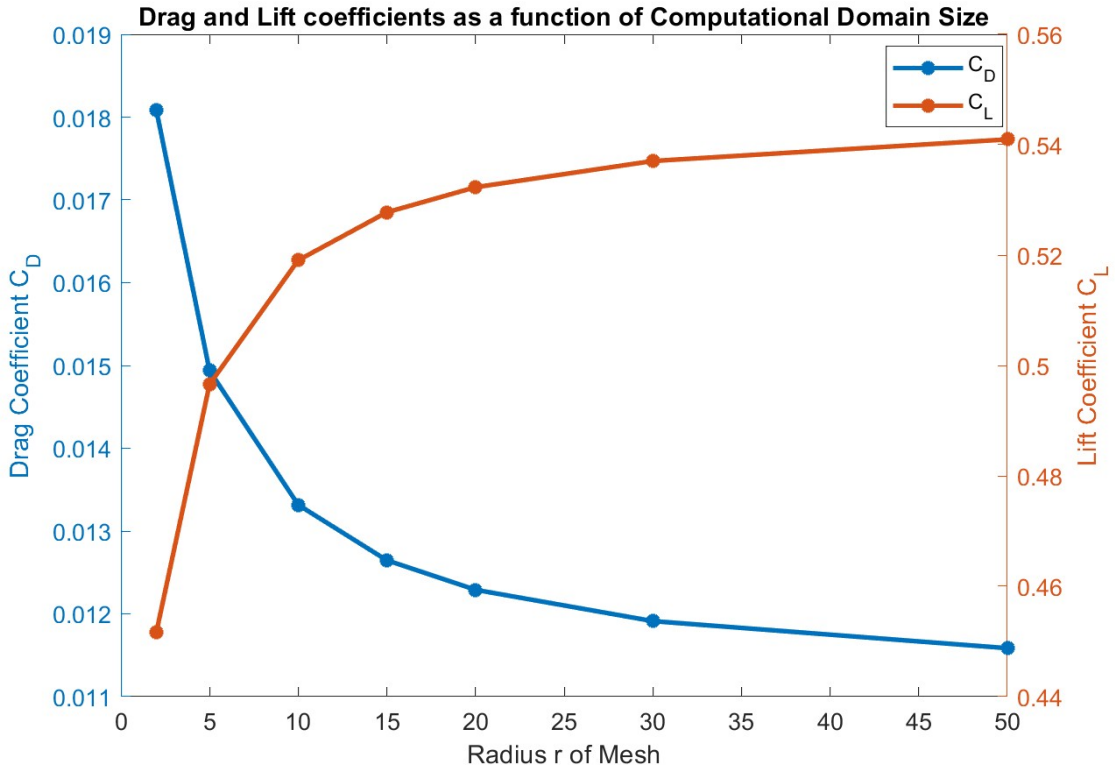
It's essential to maintain consistent simulation parameters across all cases with respective radii sizes. The following parameters stay constant throughout the experiments conducted in this section:

Table 4: Constant simulation Parameters

AOA	**# of Cells**	**# of Iter.**	**Chord Length (m)**
$\alpha = 5°$	25000	2000	1

The decision to employ a non-zero angle of attack is deliberate and strategic. Extensive testing has revealed that a zero angle of attack overlooks several intricate flow phenomena occurring near the boundary. Introducing a non-zero angle of attack creates asymmetry, thereby generating more dynamic flow patterns. Notably, the non-zero angle of attack significantly energizes the boundary layer, enhancing the capability to capture detailed information about the flow dynamics. Furthermore, it enables a comprehensive analysis of both lift and drag forces, whereas a zero angle of attack restricts the assessment solely to drag. Moreover, in conducting both computational domain size tests and grid convergence tests, the objective is to capture as much interaction and dynamic behavior as possible. This ensures that the grid and computational domain are independent of the result, allowing for robust and reliable simulations.

With these values and concepts in mind, along with the simulation parameters outlined in the preceding section, conducting the simulations is straightforward. The outcomes are as follows 1112:



Visually, it is evident that the optimal mesh size falls within the range of $r = 10$ m to $r = 25$ m. However, a more rigorous approach to determining this optimal range is through the analysis of relative errors. For example:

$$E = \frac{|C_d(i) - C_d(i+1)|}{(r(i) - r(i+1)) * C_d(i+1)}$$

This represents a normalized relative error of the drag coefficient. If we de the operation on all the values obtained, the resulting data can be significantly interpreted :

Table 5: Relative Errors of Coefficients for Different Radii

Comparison	**Relative Error (%) for Drag Coefficient**	**Relative Error (%) for Lift Coefficient**
$r = 2$ vs $r = 5$	7.0172 %	3.0152 %
$r = 5$ vs $r = 10$	2.4346 %	0.8658 %
$r = 10$ vs $r = 15$	1.0599 %	0.3293 %
$r = 15$ vs $r = 20$	0.5791 %	0.1707 %
$r = 20$ vs $r = 30$	0.3176 %	0.0884 %
$r = 30$ vs $r = 50$	0.1410 %	0.0362 %

Upon analyzing both the graphical representations and the numerical data, it becomes evident that the lift coefficient stabilizes relatively quickly, particularly around $r = 10$, maintaining a consistent value of approximately 0.54. In contrast, the convergence of the drag coefficient takes longer, eventually settling around a stable value of 0.012 at approximately $r = 15$. This observation aligns with the trends seen in the relative errors, which notably decrease to just over 1% for the drag and 0.33% at $r = 10$, continuing to decrease thereafter to 0.56% and 0.17% at $r = 15$.
Considering these insights, a mesh radius of $r = 20$ emerges as an optimal choice. This selection strikes a balanced compromise between accuracy and computational efficiency. It is also a conservative and prudent choice given the already favorable results observed at $r = 15$. Opting for this radius ensures that the mesh size is sufficiently large to succumb to boundary inconsistencies, with a relative error estimate of a tiny 0.3%. Moreover, it offers an efficient use of computational resources by avoiding an excessively large computational domain.

3 Grid Convergence Test and Mesh Selection

The grid size is a pivotal parameter in Computational Fluid Dynamics (CFD) simulations that strikes a delicate balance between computational power and test accuracy. This parameter determines the granularity of the mesh, i.e., how finely we divide our computational domain into small grid cells. The finer the grid, the more accurate the simulation becomes as it captures intricate flow details. However, finer grids also demand significantly more computational resources. Conversely, coarser grids are computationally less demanding but may sacrifice accuracy by oversimplifying flow behavior.
In the forthcoming study, the investigation will focus on assessing the influence of grid size on both the accuracy and computational cost of the simulation. Specifically, six distinct grids will be analyzed, each progressively finer than the last. These incremental variations will facilitate a comprehensive evaluation of the trade-offs between computational efficiency and simulation accuracy. To ensure consistency in the geometry, adjustments to

the JMAX and NSRF parameters in the Construct2D program, responsible for generating the meshes, will be made. These parameter changes will be proportionally adjusted to maintain identical geometric characteristics across all grid configurations. According to the Construct2D official documentation[3]:

- **NSRF:** *Number of points placed on the airfoil surface. For the O-grid topology, this is the same as the number of points in the i-direction.*
- **JMAX:***Number of points in the j-direction (the normal direction to the airfoil surface). More points reduce the error in the numerical solution, but it also increases the computation time both for the grid generator and the CFD solver.*

This implies that if JMAX is increased by a factor of α, NSRF must also be increased by the same factor. Consequently, the following options for mesh parameters arise:

Table 6: Grid Convergence Test : Grid Parameters

Grid Number	JMAX	NSRF	Number of Grid Cells	Grid Size Description
1	125	50	6000	Very Coarse
2	250	100	25000	Default
3	375	150	56000	Medium
4	500	200	100000	Medium 2
5	625	250	155500	Fine
6	750	300	225000	Very Fine

Using the same parameters as Table 4 the simulations provide the following insights (see table and 2 plots below,and for more details 1112):

Table 7: Errors for Each Grid

Grid	Relative Error (%)		Absolute Error (%)
	Lift	Drag	Lift
1	4.4517	5.5183	5.5183
2	0.6600	2.4169	2.4169
3	0.2153	1.7572	1.7572
4	0.0944	1.4425	1.4425
5	0.0412	1.3330	1.3330
6			1.192

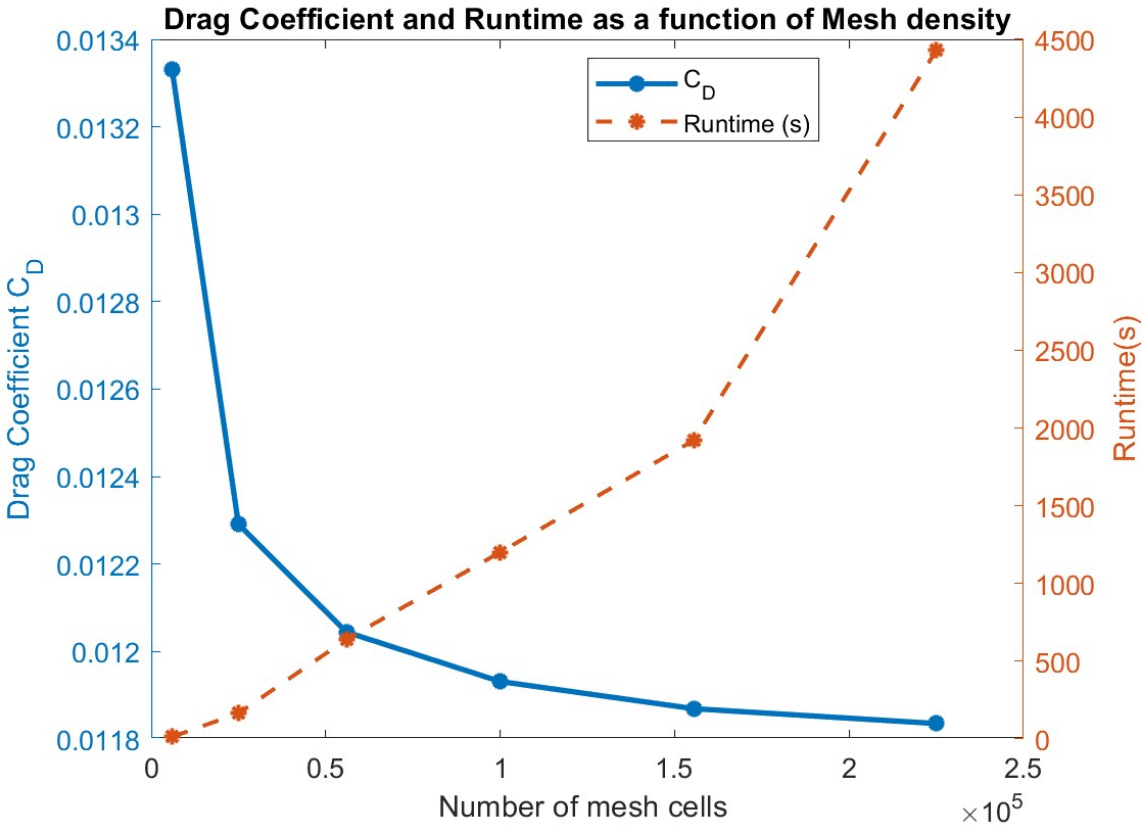


Figure 2

A couple of things to point out here:

- The relative errors are calculated using the same formula as in the previous section, whereby normalization is done by the difference between the various grid sizes. However, due to the significant magnitude of the grid sizes, a multiplication factor of 10000 is employed in this specific case to yield interpretable results, as shown in Table 7.

- The absolute error here is calculated relative to the real value of C_L. This real value is sourced from a NASA study on a NACA-0012 airfoil [4]. It is noteworthy that there is no explicit value for $\alpha = 5°$. Only a value for $\alpha = 10°$ is provided. Therefore, it is assumed that at 10° the lift curve remains in the linear regime and is not close to stall. Given the symmetric nature of the airfoil, it is known that at 0° AOA, the lift is zero. Consequently, the real value of $C_{L_{exp_5}}$ is directly interpolated at 5° by

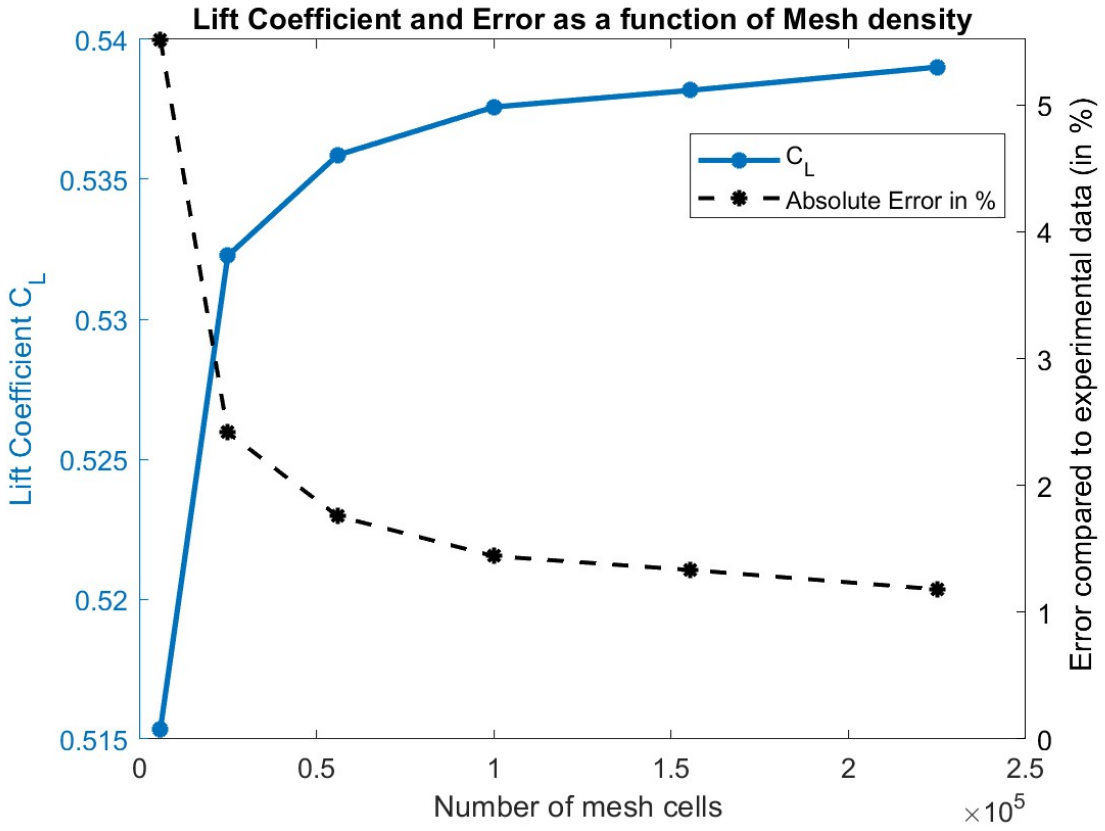


Figure 3

dividing the available value at ten degrees by 2:

$$C_{L_{exp_5}} = \frac{C_{L_{exp_{10}}}}{2} = \frac{1.0909}{2} = 0.5455.$$

-

- Finally, the absolute error of the drag coefficient was not calculated in this case. This decision stems from the substantial deviation of our drag error from the experimental data; thus, interpreting the absolute error would provide limited insight. Nevertheless, analyzing the relative drag error within the context of the convergence test remains pertinent. Various factors may contribute to the inaccuracies in drag results, including the turbulence model employed, surface roughness, numerical schemes, Mach number considerations, among others.

As for the choice of grid, the medium grid (grid #3) emerges as a robust option. Upon

inspection of the plots (Figures 2 and 3), several crucial insights come to light. Firstly, in the context of the drag polar, grid #3 precisely intersects the computation runtime and the drag curve. While this intersection may be susceptible to axis manipulation, it nonetheless serves as a valuable guidepost for gauging optimal mesh density. Secondly, focusing on the lift curve, grid #3 reveals a significant observation: at a mesh density of 55000, the curve starts to level off into a plateau phase. This plateau indicates increasingly diminishing returns with further increases in mesh density. Hence, grid #3 strikes a favorable balance between accuracy and computational efficiency and will be used for subsequent simulations.

4 Effect of Relaxation Factor

4.1 Understanding the Relaxation Factor

The relaxation factor (ω) in simulations is a coefficient used to adjust the update rate of solution variables, aiming to improve convergence and stability. It's often employed in iterative methods like the Jacobi or Gauss-Seidel iterations to accelerate convergence or ensure stability. The equation incorporating the relaxation factor is:

$$x^{(k+1)} = (1 - \omega) \cdot x^{(k)} + \omega \cdot x^*$$

where:

- $x^{(k+1)}$ is the updated solution at iteration $k + 1$,
- $x^{(k)}$ is the previous solution at iteration k,
- x^* is the computed solution from the iterative method, and
- ω is the relaxation factor, typically chosen between 0 and 1.

Adjusting ω affects the trade-off between convergence speed and stability: higher values may accelerate convergence but risk instability, while lower values improve stability but slow convergence.
It is essential to address a nuanced distinction here. The equation referenced earlier pertains to an explicit relaxation factor implementation. However, the approach used throughout the lab involves an implicit relaxation, where the relaxation factor is integrated directly into the fundamental equations. This differs from the explicit method, where the relaxation factor is applied after the iteration is solved. Generally, the implicit form is considered more stable than the explicit one.

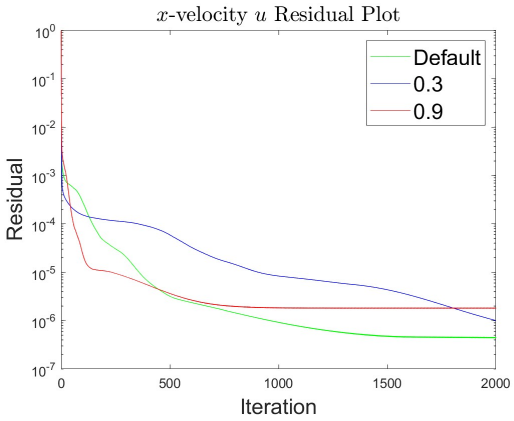
4.2 Incorporating the Relaxation Factor: Testing and Analysis

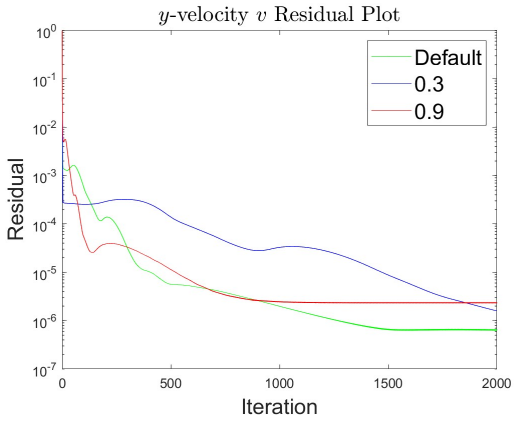
To delve deeper into the influence of the relaxation factor, a comparative analysis is proposed. Three distinct scenarios will be examined: the default case, one with all parameters set to 0.3, and another with all parameters set to 0.9.

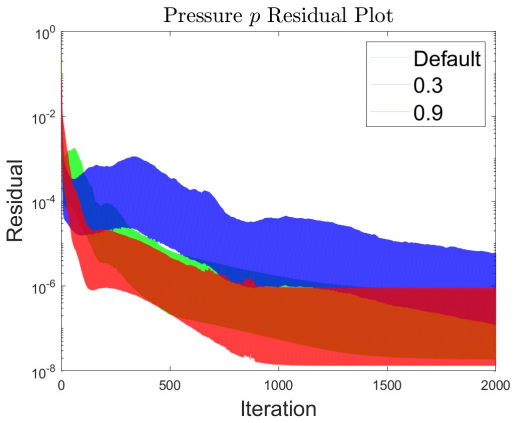
Table 8: Relaxation factors for different cases

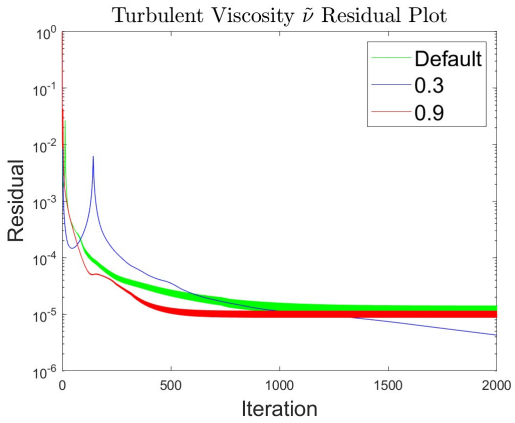
Variable	**Case 1 (Default)**	**Case 2**	**Case 3**
U	0.7	0.3	0.9
p	0.3	0.3	0.9
$\tilde{v}$	0.9	0.3	0.9

All tests will be conducted utilizing the mesh determined to be optimal in preceding sections. As per definition, the primary aspect affected by the relaxation factor is the residuals. Below, the plots depicting the individual residuals for the aforementioned three cases are presented.









A couple of remarks here:

- The general tendency is that the higher the relaxation factor, the faster the residuals converge. Across all four residuals, the red curve consistently outpaces the others in the initial convergence phase. Conversely, the blue curve tends to converge much slower, sometimes not even reaching convergence within 2000 iterations.
- Another general tendency is that higher relaxation factors lead to higher final converged states. In other words, higher relaxation factors result in less precise final solutions due to higher final residuals. Conversely, lower relaxation factors lead to lower final residuals, indicating more stable and precise converged results.
- On a more specific basis, it's interesting to observe the overlap between the red and green curves in the Turbulent Viscosity plot. This overlap occurs because both cases 1 and 3 have a turbulent relaxation factor of 0.9. However, in the case of the relaxation factor at 0.3, the turbulent viscosity does not oscillate like in the overlapping cases, possibly due to the lack of convergence in the 0.3 case or for another unknown reason.
- In contrast, for the Pressure plot, the red and green curves overlap despite having different relaxation factors. The red curve has a relaxation factor of 0.9, while the blue curve has a factor of 0.3. This suggests that pressure convergence depends not only on pressure relaxation factors but also on the other relaxation factors (Maybe because of the way the SIMPLE Algorithm auto-corrects using a pressure correction equation)[5].
- The results for the two velocity plots are consistent and show that higher relaxation factors lead to faster convergence but result in diminished precision, while lower relaxation factors lead to slower but more stable convergence and precision.

After continuing the simulation for the non-converged Case 2 we get the following table showcasing the different results :

	Less Relaxed (2)	**Default (1)**	**More Relaxed (3)**
$\mathbf{C_D}$	0.0121	0.0120	0.0121
$\mathbf{C_L}$	0.5348	0.5359	0.5359
Runtime(s)	1012	642	352
# of Iter.	4172	2000	1000

Table 9: Relaxation Factor Comparison

The crucial thing to acknowledge here is that computational time and the number of iterations decrease as relaxation factor settings become more relaxed, leading to faster

convergence of solutions. Interestingly, despite observing a notable increase in the final residual for the relaxed case in our study, the final forces, particularly for lift and drag coefficients, exhibited minimal change. This outcome might be attributed to various factors, but it's plausible that the relaxation factors weren't relaxed or constrained sufficiently to induce significant alterations in the results.

5 Effect of Numerical Scheme

5.1 Understanding Linear Vs Upwind Vs LinearUpwind Schemes

In this section, a comparison will be made between the performance and efficiency of the linear upwind model utilized in previous simulations and the linear model. To provide context and insight into this analysis, a brief summary is presented based on the background information outlined in references [6] and [7].

- **Linear Scheme**: A second-order, central difference scheme suitable for isotropic meshes that has limited numerical dissipation.

  This scheme works as follows:

  $$\phi_f = r\phi_P + (1-r)\phi_N \tag{1}$$

  where $r = \frac{fN/P}{N}$, i.e., the ratio of the distance between the center of cell N and the face shared by cells N and P divided by the distance between cells P and N.

- **Linear Upwind Scheme**: A second-order, upwind-biased, unbounded scheme (less so than linear, though), which employs an explicit correction based on the local cell gradient.

  The scheme is implemented as:

  $$\phi_f = \phi_P + r \cdot (\nabla\phi)_P \tag{2}$$

  where r denotes the vector connecting the center of the upwind cell P with the center of the face f (where the value of ϕ is being evaluated) and $(\nabla\phi)_P$ is the gradient of ϕ at the center of the upwind cell P.

- **Upwind Scheme**: A first-order, bounded scheme where the face value is set according to the upstream cell value. Assumes isotropic cell values with a value representing the average. Implemented as:

  $$\phi_f = (\nabla\phi)_P$$

  where $(\nabla\phi)_P$ is the gradient of ϕ at the center of the upwind cell P.

In summary, the linear model straightforwardly interpolates values from the owning cell and its neighboring cells, without incorporating additional information. Conversely, the upwind scheme relies solely on information from neighboring cells without any interpolation or averaging. This additional information is crucial for accurately capturing the divergence term in the Navier-Stokes equations. The linear upwind scheme combines the strengths of both approaches, utilizing neighboring cell information while maintaining stability akin to the linear model. Furthermore, the linear upwind model is partially bounded, making it slightly less stable than the upwind model but more precise than the linear model. Nevertheless, this enhancement comes at the cost of increased computational power, as is often the case in numerical simulations.

5.2 Application of Schemes and Comparison

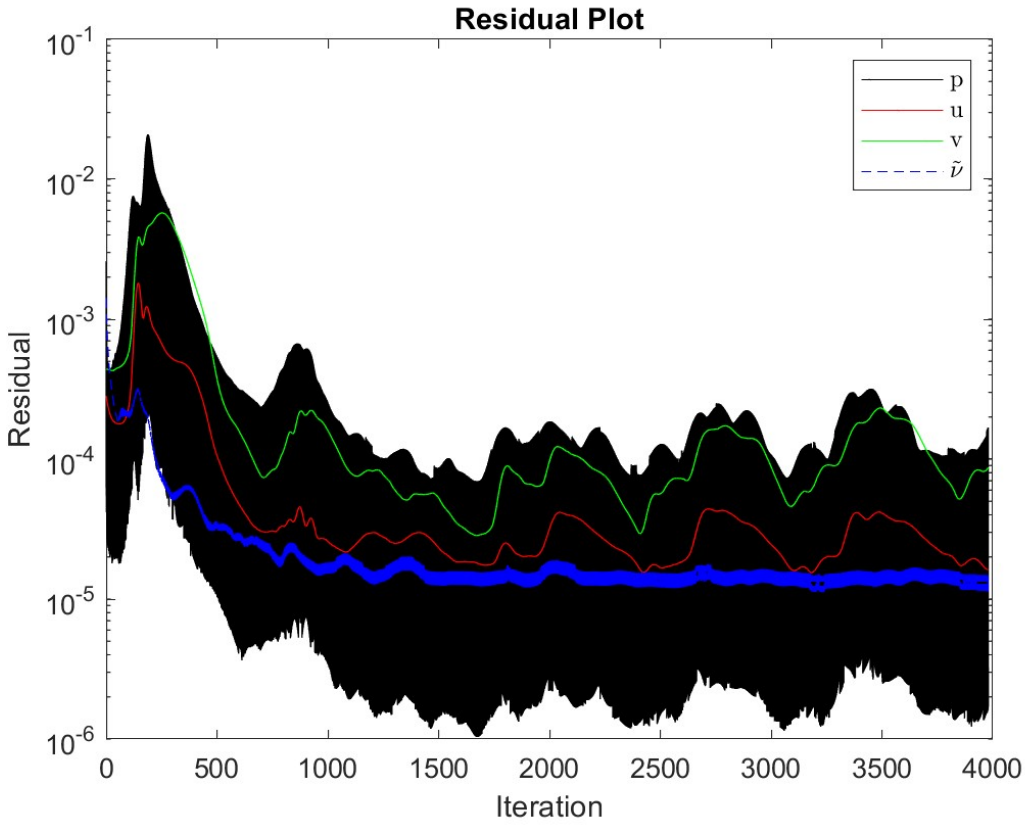
With the identical setup employing the optimal mesh and the default relaxation factors, the only variation introduced is the substitution of the Linear Upwind scheme with a Linear scheme and the Upwind scheme in the velocity divergence term. The resulting outcomes are as follows:

Table 10: Comparison of Linear and Linear Upwind Results

	Linear	Linear Upwind	Upwind
y^+ (min)	0.074603	0.0481707	0.0309183
y^+ (max)	1.03157	1.00873	0.965832
y^+ (average)	0.409275	0.404578	0.387977
u residual	1.60417×10^{-6}	4.1000×10^{-7}	3.36232×10^{-7}
v residual	4.37774×10^{-6}	4.7354×10^{-7}	6.45708×10^{-7}
p residual	1.38782×10^{-6}	6.6279×10^{-7}	2.94318×10^{-7}
$\tilde{v}$ residual	1.47634×10^{-6}	1.1741×10^{-5}	9.00085×10^{-6}
Iterations	4000	1600	2000
Runtime (s)	1548	642	621
C_d	0.0114	0.0123	0.0243
C_l	0.5464	0.5323	0.5031
Converged?	Not Yet	Yes	Yes

5.2.1 Linear Scheme

Delving deeper into the details for the different schemes, the following figure is the residual plot for the linear scheme:



The plot highlights the instability inherent in the linear scheme when contrasted with the Linear Upwind scheme utilized in the preceding sections. A pronounced wave of instability permeates all four residuals, indicative of a less robust simulation. Referring back to 10, although the computational cost for the linear scheme significantly increased, and the final residuals weren't nearly as small or as converged as those of the Linear Upwind model, it's important to note that there was a slight increase in precision, with the lift value of the linear model being nearly the exact same as the experimental lift of $C_L = 0.5455$. Additionally, the drag coefficient significantly decreased and is again closer relative to the Linear Upwind model to the experimental result of 0.007.
To conclude on the linear scheme, the linear model is more precise, but it uses more computational power and is more vulnerable in terms of stability.

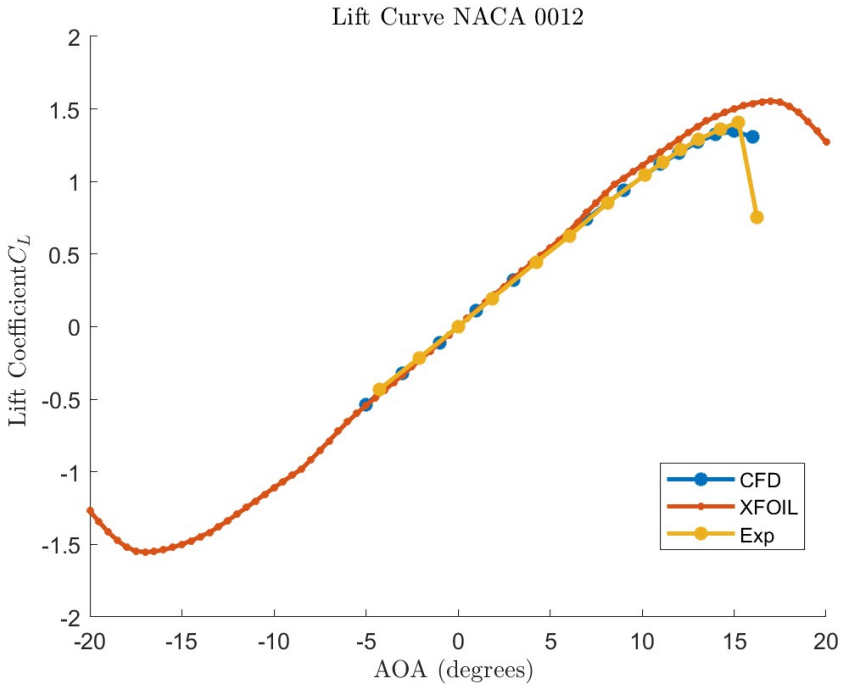
5.2.2 Upwind Scheme

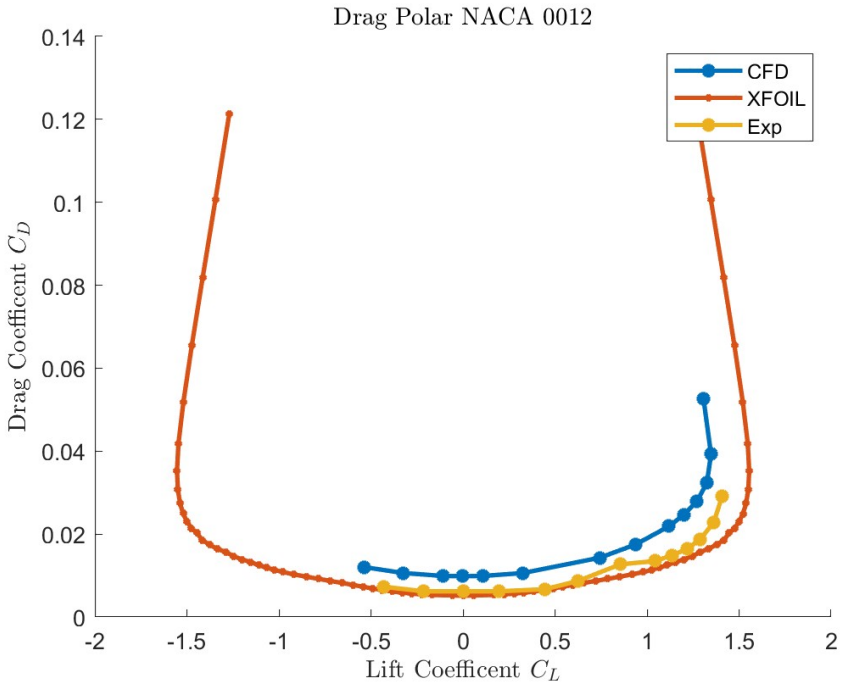
The upwind scheme, while notably simplistic, inherently sacrifices accuracy as its primary trade-off. This compromise is evident in the comparatively lower accuracy of both lift

and drag coefficients, with the latter nearly doubling in comparison to the linear upwind scheme. Nevertheless, due to its first-order nature, the upwind scheme offers significant computational efficiency, resulting in faster simulation times per iteration. However, it is crucial to highlight that despite the reduced computational time per iteration, the linear scheme required a higher total number of iterations to converge. Consequently, this effectively nullifies the computational advantage initially gained with the upwind scheme. The key takeaway is that while the first-order scheme is less computationally demanding, it is also far less accurate compared to second-order schemes.

6 Lift and Drag Curve for $-5^\circ < \alpha < 15^\circ$

Based on the comprehensive information gathered from the preceding sections, the case exhibits stability and consistency, enabling the prediction of the lift curve and drag polar of the NACA-0012 Airfoil with confidence. Additionally, the computational fluid dynamics (CFD) results will be juxtaposed with those obtained from Xfoil in XFLR 5, alongside reputable experimental data [8]. This comparison will be depicted in the following plots.





The analysis presents a notable coherence in the lift curve shape across all three cases studied, yet reveals a significant incongruity in the predicted drag values compared to actual wing testing. This discordance likely stems from various factors inherent in computational fluid dynamics (CFD) simulations, including model selection, numerical schemes, viscosity, turbulence parameters, and wall treatments. Notably, while experimental results share the same Reynolds number as CFD simulations, they diverge in Mach number, potentially contributing to the observed disparity despite the flow's compressibility.

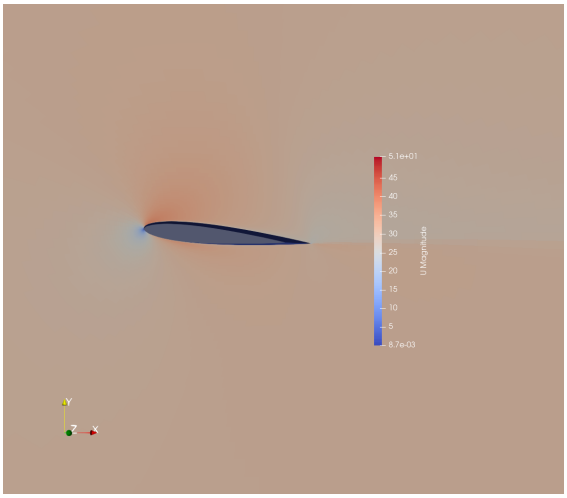
Interestingly, the alignment between findings from Xfoil and NASA experiments suggests a closer correspondence with real-world observations compared to CFD predictions, hinting at potential inaccuracies or oversights in the latter's assumptions. Despite these discrepancies, the consistent shape of the curves across all cases yields a satisfactory and coherent outcome.

In the stall region behavior, slight differences are observed among the three cases, indicating divergence between numerical models and real-world results as assumptions regarding wall boundary conditions become invalid. This is further substantiated by significant increases in y+ values beyond an angle of attack of 9° (see 13):

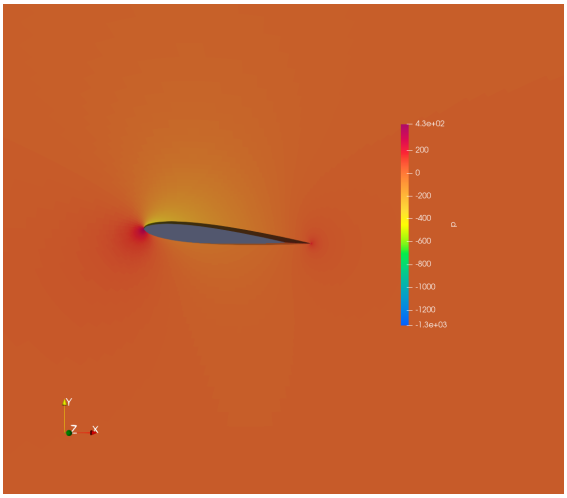
$\alpha = 7°$	$\alpha = 9°$	$\alpha = 11°$	$\alpha = 12°$	$\alpha = 13°$	$\alpha = 14°$	$\alpha = 15°$	$\alpha = 16°$
1.14597	1.30413	1.45288	1.52591	1.61582	1.65687	1.70658	1.74461

After the airfoil begins to stall, the validity of the model slowly becomes less and less

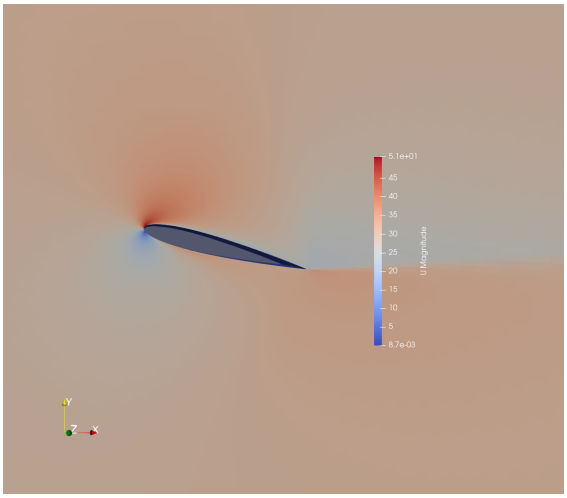
accurate. Here are a couple of images for reference to visually observe how the flow is behaving:



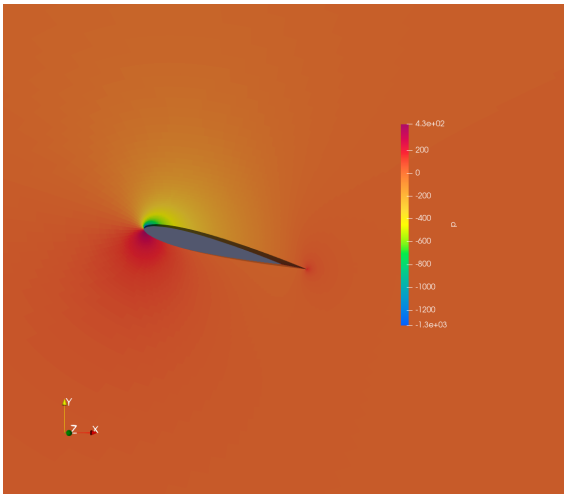
$\alpha = 5°$: U Field (Attached flow)



$\alpha = 5°$: P Field (Attached flow)



$\alpha = 14°$: U Field (Detached Stalled flow)



$\alpha = 14°$: P Field (Detached Stalled flow)

7 Conclusions and Final Verdict

This CFD lab underscores a fundamental truth: trust in CFD results must be earned through relentless iteration and validation. The iterative process is essential, requiring continual refinement and adjustment to account for the myriad variables at play. From initial assumptions to model complexities, CFD is a domain of intricate interplay, which is why CFD is a prominent research and development field
The key lesson learned is the necessity of vigilance when employing CFD. Every parameter warrants scrutiny, ensuring internal coherence and external validity. Most crucially, validation through comparison with credible sources or physical wind tunnel tests is indispensable. By adhering to these principles and being wary we can make sure that the CFD model makes sense and that it can be re-employed for other applications.

8 Appendix

Table 11: Summary of Lift and Drag coefficients

Radius (m)	**Drag Coefficient (C_D)**	**Lift Coefficient (C_L)**
Computational Domain Size		
2	0.0181	0.4517
5	0.0149	0.4966
10	0.0133	0.5191
15	0.0126	0.5278
20	0.0123	0.5323
30	0.0119	0.5370
50	0.0116	0.5410
Grid Convergence Test		
Very Coarse	0.0133	0.5154
Default	0.0123	0.5323
Medium	**0.0120**	**0.5360**
Medium 2	0.0119	0.5376
Fine	0.0119	0.5384
Very Fine	0.0118	0.5390
Relaxation Factor Test		
0.3	0.0125	0.5359
0.9	0.0121	0.5359
Schemes		
Linear	0.0114	0.5464
Upwind	0.0243	0.5031
Lift curve and Drag Polar		
$\alpha = -5$	0.0120	-0.5359
$\alpha = -3$	0.0106	-0.3236
$\alpha = -1$	0.0099	-0.1082
$\alpha = 0$	0.0098	0.0000
$\alpha = 1$	0.0099	0.1082
$\alpha = 3$	0.0106	0.3235
$\alpha = 7$	0.0143	0.7427
$\alpha = 9$	0.0175	0.9390
$\alpha = 11$	0.0219	1.1192
$\alpha = 12$	0.0246	1.1992
$\alpha = 13$	0.0279	1.2715
$\alpha = 14$	0.0324	1.3254
$\alpha = 15$	0.0393	1.3477
$\alpha = 16$	0.0526	1.3070

Case **CFL MAX**	**Residual Values**				**# of Iter.**	**Run-time (s)**	**yplus max**
	Pressure p	x **velocity** u	y **velocity** v	**Turb Viscosity** $\tilde{\nu}$			
Computational Domain Size							
$r = 2$	1.5767e-05	5.1688e-06	4.9784e-06	1.4304e-04	2000	231	
$r = 5$	5.1316e-06	3.1378e-06	4.1593e-06	8.1785e-05	2000	157	
$r = 10$	1.9363e-06	1.9887e-06	3.1151e-06	5.2102e-05	2000	174	
$r = 15$	1.3664e-06	1.4139e-06	2.2915e-06	3.9350e-05	2000	232	
$r = 20$	1.5950e-06	1.1540e-06	1.9187e-06	3.1762e-05	2000	219	
$r = 30$	8.9700e-07	8.5901e-07	1.4693e-06	2.3689e-05	2000	173	
$r = 50$	4.8295e-07	5.6464e-07	9.8552e-07	1.4984e-05	2000	225	
Grid Convergence Test							
V.Coarse	6.9527e-06	7.4575e-06	2.4454e-05	1.5360e-04	997	14	
Default	1.6093e-06	1.1660e-06	2.0003e-06	3.1777e-05	1500	165	
Medium	**4.1000e-07**	**4.7354e-07**	**6.6279e-07**	**1.1741e-05**	**1600**	**642**	
Medium 2	6.6605e-07	2.1527e-07	2.4346e-07	5.7221e-06	4500	1201	
Fine	1.6316e-07	1.5282e-07	1.5393e-07	3.5209e-06	4000	1921	
V.Fine	1.0052e-06	1.0460e-07	1.0938e-07	2.3280e-06	8000	4429	
Relaxation factor							
More Relaxed	9.68536e-07	1.71232e-07	2.40034e-06	8.68503e-06	1000	352	
Less Relaxed	8.12077e-09	1.08601e-07	9.96214e-07	4.23823e-07	4172	1012	
Schemes							
Linear	8.5297e-05	1.6301e-05	8.7270e-05	1.47955e-05	4000	1548	
Upwind	2.94318e-07	3.36232e-07	6.45708e-07	9.00085e-06	2000	621	
Lift Curve and Drag Polar							
$\alpha = -5$	4.3547e-07	4.4377e-07	6.7307e-07	1.1613e-05	2000	322	
$\alpha = -3$	1.3015e-06	4.3388e-07	7.2963e-07	1.2732e-05	2000	262	
$\alpha = -1$	2.1881e-07	4.1439e-07	1.2710e-06	1.3438e-05	2000	261	
$\alpha = 0$	2.0887e-07	4.1243e-07	2.8647e-06	1.3529e-05	2000	240	
$\alpha = 1$	5.1932e-07	4.0348e-07	1.2128e-06	1.2647e-05	2000	357	
$\alpha = 3$	5.6985e-06	4.2765e-07	1.0765e-06	1.2701e-05	2000	296	
$\alpha = 7$	9.2053e-07	5.0047e-07	9.6399e-07	8.9728e-06	2000	394	
$\alpha = 9$	4.0919e-06	7.5564e-07	2.5739e-06	6.5077e-06	2000	622	
$\alpha = 11$	2.2605e-06	1.2054e-06	3.8963e-06	5.9794e-06	2000	409	
$\alpha = 12$	1.5216e-06	1.5565e-06	5.3161e-06	6.5709e-06	2000	432	
$\alpha = 13$	1.8117e-06	1.7601e-06	6.4512e-06	8.2437e-06	2000	664	
$\alpha = 14$	2.1270e-06	2.2530e-06	9.1471e-06	9.1856e-06	2000	536	
$\alpha = 15$	2.5108e-06	2.7300e-06	1.2621e-05	1.5121e-05	2000	636	
$\alpha = 16$	3.3925e-06	2.9411e-06	1.6263e-05	1.9170e-05	2000	540	

Table 12: Summary of Residuals, Convergence, and Runtime

Table 13: Summary of y^+ Values

Case	Minimum	Maximum	Average
Computational Domain Size			
2	0.034055	0.945288	0.414198
5	0.0525165	0.968047	0.408982
10	0.0584415	0.98308	0.408195
15	0.0612063	0.989108	0.408167
20	0.0632229	0.992356	0.408019
30	0.0653283	0.99579	0.407915
50	0.0668632	0.998704	0.407786
Grid Convergence Test			
V.Coarse	0.106534	0.9519	0.409088
Medium	**0.0481707**	**1.00873**	**0.404578**
Medium 2	0.10747	1.00444	0.403705
Fine	0.126245	1.01107	0.402268
V.Fine	0.143077	1.0068	0.40038
Relaxation Factor Test			
0.9	0.0573	1.00761	0.404179
0.3	0.0583984	1.00934	0.410387
Schemes			
Linear	0.0583984	1.00934	0.410387
Upwind	0.0309183	0.965832	0.387977
Angle of Attack			
$\alpha = -5$	0.0481499	1.00869	0.404635
$\alpha = -3$	0.0629445	0.862142	0.408478
$\alpha = -1$	0.0755886	0.733391	0.410319
$\alpha = 0$	0.0818424	0.672485	0.410886
$\alpha = 1$	0.0754984	0.733374	0.410237
$\alpha = 3$	0.0628615	0.862112	0.408511
$\alpha = 7$	0.030749	1.14597	0.398519
$\alpha = 9$	0.0359059	1.30413	0.388503
$\alpha = 11$	0.0281864	1.45288	0.371748
$\alpha = 12$	0.00711118	1.52591	0.362794
$\alpha = 13$	0.00716707	1.61582	0.360087
$\alpha = 14$	0.00778364	1.65687	0.35854
$\alpha = 15$	0.0191871	1.70658	0.360186
$\alpha = 16$	0.0222841	1.74461	0.363116

References

[1] montagdp. (2018) Construct2d computational fluid dynamics structured grid creator for 2d airfoils. SourceForge. [Online]. Available: https://sourceforge.net/projects/construct2d/

[2] OpenFOAM. (2018) Openfoam: User guide v2112. OpenCFD Ltd. [Online]. Available: https://www.openfoam.com/documentation/guides/latest/doc/guide-applications-solvers-incompressible-simpleFoam.html

[3] montagdp. (2018) Construct2d readme official user manual. SourceForge. [Online]. Available: https://sourceforge.net/projects/construct2d/files/README/download

[4] C. Rumsey. (2021) Turbulence modeling resource : 2d naca 0012 airfoil validation case sa model results. NASA, Langley Research Center. [Online]. Available: https://turbmodels.larc.nasa.gov/naca0012_val_sa.html

[5] t. f. e. Wikipedia. (2022) Simple algorithm. Wikimedia Foundation, Inc. [Online]. Available: https://en.wikipedia.org/wiki/SIMPLE_algorithm

[6] C. Greenshields. (2023) Openfoam v11 user guide - 4.5 numerical schemes. CFD Direct. [Online]. Available: https://doc.cfd.direct/openfoam/user-guide-v11/fvschemes

[7] A. T. B. M. G. Maragkosa, S. Vermab. (2021) Evaluation of openfoam's discretization schemes used for the convective terms in the context of fire simulations. a-Department of Structural Engineering and Building Materials, Ghent University, St. Pietersnieuwstraat 41, B-9000 Ghent, Belgium b- Department of Fire Protection Engineering, University of Maryland, College Park, MD 20742, USA. [Online]. Available: https://backoffice.biblio.ugent.be/download/8735958/8735963

[8] C. L. Ladson. (1988) 19880019495 : Effects of independent variation of mach and reynolds numbers on the low-speed aerodynamic characteristics of the naca 0012 airfoil section. NASA, Langley Research Center. [Online]. Available: https://turbmodels.larc.nasa.gov/naca0012_val_sa.html