Analysis of Boundary Layers Using CFD

Finn Cracknell

2nd May, 2025

Contents

1 Introduction

Fluid dynamics is one of the various branches of classical mechanics that deals with the mechanisms underpinning the behaviour of liquids, gases, and plasmas. Despite its presence in an incredibly broad range of our everyday lives, fluid dynamics, in the overwhelming majority of cases, is not understood to an exact level. This has in turn driven professionals from all fields, particularly aerospace engineers, to seek out methods of better understanding and approximating fluid behaviour, especially in complex scenarios.

More specifically, the behaviour of fluid flow within a boundary layer has many vital applications that are at the forefront of aerospace engineering research and design. As a means of conducting this research and observing results in complex fluid phenomena that would otherwise be difficult to reproduce by physical experimentation, computational fluid dynamics (CFD) is employed around the world by leading researchers and designers in the aerospace industry [1].

This investigation explores the behaviour of boundary layer flows in a range of conditions and regimes through the use of CFD, comparing these computational results to well-understood boundary layer theory. To achieve this objective, three separate boundary layer flow cases are considered, including both laminar and turbulent incompressible flow over a flat plate, as well as laminar compressible flow over an infinite cylinder. Using the numerical results, key quantities such as boundary layer thicknesses and friction coefficients are measured for both flow cases over the flat plate and comparison is made with theory. The concept of thermal boundary layers is also explored in the laminar flat plate case across three different isothermal plate temperature conditions, by observing the local temperature profile. Similarly, through the local streamwise velocity distribution normalised by boundary layer thickness, the self-similarity of boundary layer flow across a flat plate is also explored in both the laminar and turbulent cases. Lastly, in the cylindrical case, the behaviour of boundary layers for a bluff body is investigated by analysing the local tangential flow distribution at discrete angular positions along the cylindrical surface, comparing with inviscid theory and the boundary layer behaviour in the previous flat plate cases.

2 Results and Discussion

2.1 Incompressible Laminar Flow Over a Flat Plate

Using SU2, an open-source CFD software developed by Stanford University, discrete values for both the temperature and pressure, as well as horizontal and vertical velocity components (T, P, U and V respectively), were obtained for incompressible laminar flow over a flat plate with $U_\infty = 69.17\ ms^{-1}$, $T_\infty = 297.62\ K$, $\mu = 1.83 \times 10^{-5}\ Nsm^{-2}$, $\rho_\infty = 1.13\ kgm^{-3}$, $c_p = 1004.7\ Jkg^{-1}K^{-1}$, and $Pr = 0.72$. This was first performed with $T_{wall} = 148.81\ K$ and the resultant velocity field was analysed as depicted in Figure 1 below.

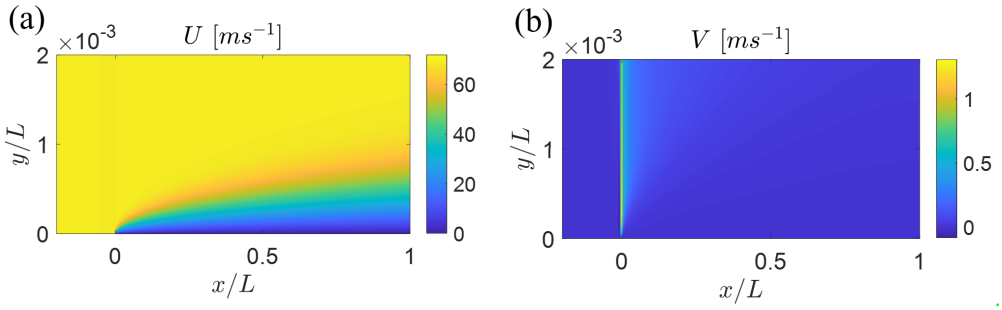

Figure 1: Contour plots of both horizontal (a) and vertical (b) velocity components for incompressible laminar flow over a flat plate, cropped to the region near the boundary.

To make comparison between the simulated boundary layer behaviour and known theory, boundary layer thickness δ, displacement thickness δ^*, momentum thickness θ, local friction coefficient c_f, and total friction coefficient C_f, can be computed numerically using the general equations for these boundary layer quantities [2] (see Appendix A.1) and plotted as a function of streamwise distance x. This comparison can be made as these key quantities in incompressible laminar boundary layer behaviour are underpinned by known theoretical counterparts derived from the Blasius boundary layer solution [2] (see Appendix A.2).

In Figure 2 below, these quantities are plotted alongside their respective Blasius solutions to assess their agreement with expected theoretical values. Due to the course grid sizing, the streamwise velocity distribution was interpolated to finer values of y at each streamwise location to smooth the results displayed.

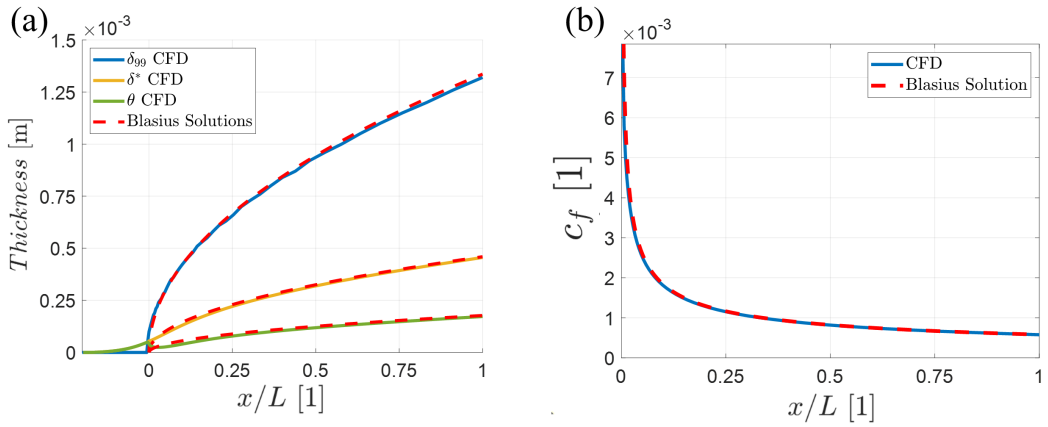

Figure 2: Key boundary layer quantities as computed by the streamwise velocity field resultant from CFD for incompressible laminar flow over a flat plate plotted against streamwise distance alongside their respective Blasius solutions. Boundary layer thickness δ, displacement thickness δ^*, and momentum thickness θ (a). Local friction coefficient c_f (b).

Observing Figure 2 above, it can be seen that for $x/L > 0$, these values develop seemingly exactly as predicted by boundary layer theory. This is reflected by the red dotted line closely matching the distribution of all key boundary layer quantities as a function of streamwise distance from the leading edge of the flat plate.

However for $x < 0$, by closely analysing Figure 2 (a) it can be seen that there exists a region where both δ^* and θ, as calculated by the computationally derived streamwise velocity field, are non-zero. Given there exists no boundary in this region, we expect there to exist no boundary layer thickness as indicated by the Blasius solutions only existing for $x/L > 0$. This anomaly could possibly be explained by numerical dissipation in the CFD code caused by round-off errors that induce tiny perturbations in the free-stream velocity field causing it to slightly deviate from U_∞ for $x/L < 0$. This in turn causes a non-zero δ^* and θ to be calculated as these small deviations accumulate over the integrated streamwise velocity distribution.

As depicted in Figure 2, with the local friction coefficient obtained as a function of x, the total drag coefficient C_f can also be obtained through equation 1.

$$C_f = \frac{1}{L}\int_0^L c_f(x)dx \tag{1}$$

Doing so resulted in a total drag coefficient, calculated from the CFD results, of $C_{f,CFD} = 0.0011$. Comparing this to the Blasius solution of $C_{f,Balsius} \approx \frac{1.328}{\sqrt{Re_L}} = 0.0012$, $C_{f,CFD}$ is seen to agree well

with the approximate theoretical value to a relative percent difference of only -7.91%.

Despite agreement with theory being observed in this particular computationally resolved solution for flow over a flat plate, this behaviour is not likely to continue indefinitely with x in a real-world experimental scenario. This is due to the fact that the Blasius solution and the CFD simulation don't account for disturbances or turbulence that may exist in the flow. Particularly, as the distance in the streamwise direction increases along an infinitely long plate, Re_x also increases. In a real-world scenario, this eventually causes Re_x to cross some critical number, in turn causing a transition into turbulent flow. Under turbulent flow conditions, the key boundary layer quantities explored above no longer follow the previously discussed theoretical relationship. Instead, the theoretical relationship changes as a result of turbulent flow, therefore causing this behaviour to *not* continue indefinitely with x, as further discussed in Section 2.3.

Furthering the analysis of these results, vertical distance from the plate y can be normalised by the boundary layer thickness δ (δ_{99}) and plotted against streamwise velocity for several points along the plate to observe the self-similar nature of the boundary layer development, done so in Figure 3 below.

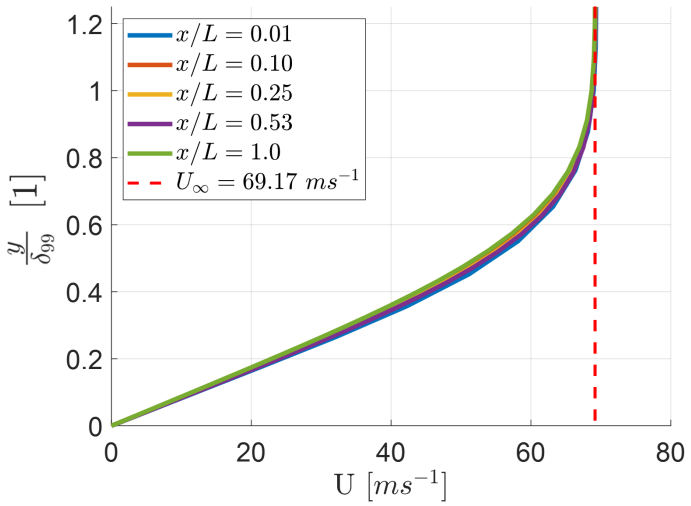

Figure 3: Vertical distance from plate y normalised by boundary layer thickness δ (δ_{99}) plotted against streamwise velocity U alongside $U_\infty = 69.17\ ms^{-1}$ for the laminar flat plate case.

Here Figure 3 depicts the streamwise velocity distributions with y normalised by the boundary layer thickness at 5 points along the plate. As shown for all points along x/L in Figure 3, it is seen that the streamwise velocity starts from zero and settles at the free-stream velocity U_∞ in approximately the exact same manner when y is normalised in this way. Given this collapse onto the same is curve is observed, we can therefore say that the streamwise velocity profile for this flow case is self-similar.

Having now considered the boundary layer behaviour for stream wise velocity, using the temperature field T that is also outputted by the CFD routine, the possibility of boundary-layer-like behaviour in temperature can be investigated. Hence, as an initial investigation, the temperature distribution was plotted as a function of y at $x/L = 0.53$ to observe possible boundary-layer-like behaviour. As a means of further investigating the effect of plate temperature on this temperature distribution, this was performed for the original case of $T_{wall} = 148.81\ K$, as well as $T_{wall} = T_\infty$ and $T_{wall} = 350\ K$, summarised in Figure 4 below.

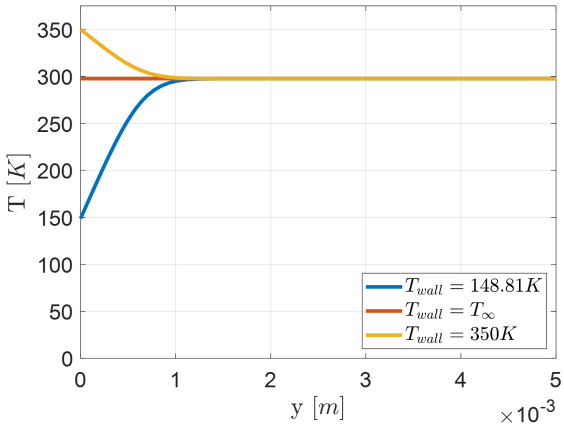

Figure 4: Temperature profile near the boundary at $x/L = 0.53$, $T_\infty = 297.62K$, the cases of $T_{wall} = 148.81\ K$, $T_{wall} = T_\infty$, and $T_{wall} = 350\ K$.

Figure 4 clearly shows a uniform temperature distribution for the case of $T_{wall} = T_\infty$ as no heat transfer occurs between the fluid and the wall given they are at the same temperature, therefore causing a uniform temperature field. However, for the cases where $T_{wall} = 148.81\ K$ and $T_{wall} = 350\ K$, the temperature is clearly shown to begin at the boundary temperature for $y = 0$, changing in their respective directions and stabilising to the free-stream temperature T_∞ at relatively large values of y. This is exactly analogous to the behaviour of the local stream wise velocity distribution that gives rise to boundary-layer-like behaviour in streamwise velocity, hence boundary-layer-like behaviour is also observed for temperature in the cases where $T_{wall} = 148.81\ K$ and $T_{wall} = 350\ K$. However, this behaviour is not observed for $T_{wall} = T_\infty$ as the temperature distribution is uniform for all x and y.

As an extension and further example for this analysis, it is possible to characterise the boundary-layer-like behaviour of temperature by first defining the non-dimensional variable $\tilde{\theta}$.

$$\tilde{\theta} = 1 - \frac{T(y) - T_\infty}{T_{wall} - T_\infty} = \frac{T(y) - T_{wall}}{T_\infty - T_{wall}} = f'(\eta), \quad \eta = y\sqrt{\frac{U_\infty}{2\nu x}} \tag{2}$$

Using this new non-dimensional variable, we can define a temperature boundary layer thickness variable analogous to δ_{99} by finding the value of y for which $\tilde{\theta} >= 0.99$. Figure 5 explores the implications of this, performing similar analysis for temperature boundary layer development in the streamwise direction and similarly identifying self-similarity in the temperature distribution.

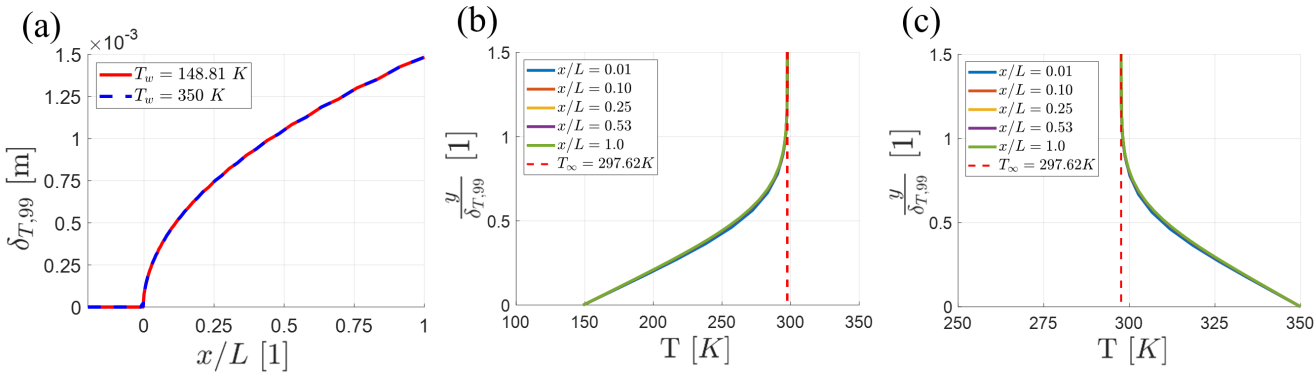

Figure 5: $\delta_{T,99}$ plotted against x/L for both $T_{wall} = 148.81\ K$ and $T_{wall} = 350\ K$ (a). Vertical distance y normalised by $\delta_{T,99}$ plotted against T for various locations along the plate for both $T_{wall} = 148.81\ K$ (b) and $T_{wall} = 350\ K$ (c).

As shown in Figure 5 above, $\delta_{T,99}$ has seemingly the same evolution in the streamwise direction for both $T_w = 148.81\ K$ and $T_w = 350\ K$. This expected and due to the fact that, as given in Eq. 2, $\tilde{\theta}$ is equal to $f'(\eta)$ where η is constant in the two cases given the same free-stream conditions and hence the values of y for which $\tilde{\theta} >= 0.99$, are the same.

Also depicted is the self-similarity in the temperature distribution as normalised by this temperature boundary layer thickness. Paralleling the self-similarity in the streamwise velocity, the plots of y normalised by the temperature boundary layer thickness against temperature are seen to overlap at all values of x/L shown. Mirroring the previous result, this shows an inherent self-similarity that is also present in the boundary layer temperature distribution.

2.2 Incompressible Turbulent Flow Over a Flat Plate

Through SU2's Reynolds Averaged Navier-Stokes turbulent model, discrete values for the horizontal and vertical velocity components were again obtained, this time for incompressible, turbulent flow over a flat plate. This was performed with $U_\infty = 69.44\ ms^{-1}$, $\mu = 1.85 \times 10^{-5}\ Nsm^{-2}$, $\rho_\infty = 1.33$. Using the Eqs. 6 to 10 in Appendix A.1, the same key quantities were computed using the outputted velocity field.

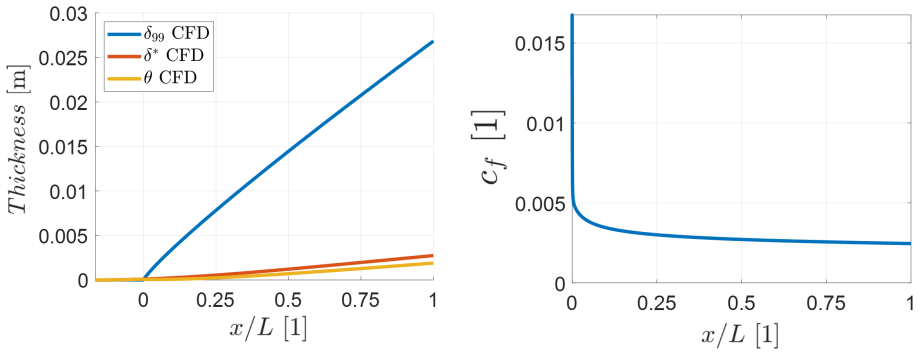

Figure 6: Key boundary layer quantities as computed by the streamwise velocity field resultant from CFD for incompressible turbulent flow over a flat plate plotted against streamwise distance. (a) Boundary layer thickness δ, displacement thickness δ^*, and momentum thickness θ. (b) Local friction coefficient c_f.

Noting the relative size and order of magnitude in both Figure 6 (a) and (b), it is seen that the values of δ_{99}, δ^*, θ, and c_f are all much greater in the turbulent case when comparing with the laminar results displayed in Figure 2. This increase in boundary layer thicknesses is due to an increased mixing of high-momentum fluid in the free-stream with low-momentum fluid near the boundary that occurs in turbulent flow, causing the boundary layer to grow faster and hence become thicker. This mixing in turn induces steeper velocity gradients and hence greater shear stress at the wall, increasing both c_f and C_f. Therefore the increase of these key quantities in the turbulent case aligns with theoretical expectations.

Calculating the drag coefficient C_f by Eq. 10, results in a value of $C_{f,turbulent} = 0.0029$ which is significantly larger than $C_{f,laminar} = 0.0011$ calculated in Section 2.1. This is in agreement with the expected increase in skin friction for the turbulent case due to the reasons discussed above.

As done in Figure 3 of Section 2.1, it is possible investigate whether or not this turbulent velocity boundary layer exhibits self-similar behaviour by following the same methods. As such, the turbulent streamwise velocity distributions with y normalised by the turbulent boundary thickness δ for 6 points along the plate are depicted in Figure 7 below.

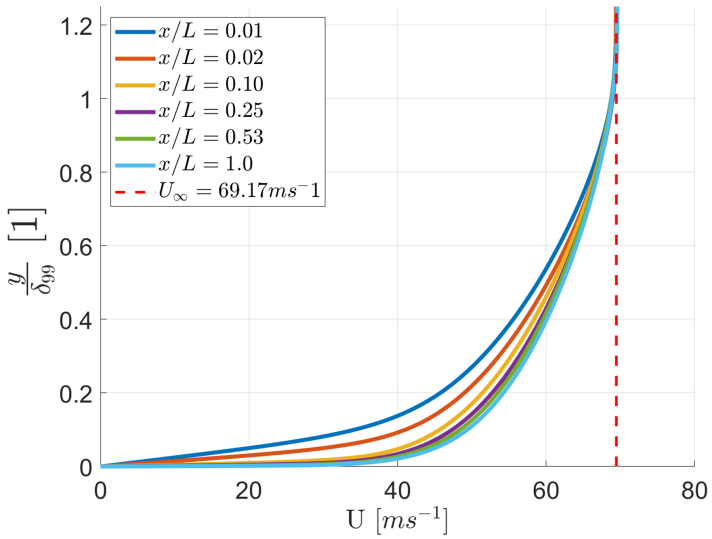

Figure 7: Vertical distance from plate y normalised by boundary layer thickness δ (δ_{99}) plotted against streamwise velocity U alongside $U_\infty = 69.17\ ms^{-1}$ for the turbulent flat plate case.

Through Figure 7, a clear difference is seen between the normalised streamwise velocity distribution in the turbulent case as compared to the laminar case. Here, it cannot be said that the normalised distribution maintains its shape at all positions in x as the distribution is clearly shown transform between small and large values of x. Therefore, as each normalised streamwise velocity distribution is not globally seen to collapse onto the same curve, it cannot be said that this turbulent streamwise velocity boundary layer exhibits self-similar behaviour.

2.3 Compressible Laminar Flow Over an Infinite Cylinder

Once more through the use of SU2, discrete values for the pressure as well as horizontal and vertical velocity components were obtained for compressible, laminar flow over an infinite cylinder across a polar grid to investigate the viscous flow dynamics of a bluff body.

Plotting these resultant values of pressure along side both horizontal and vertical velocity, allows a rough overview of the flow behaviour to be visualised and is hence done in Figure 8 below.

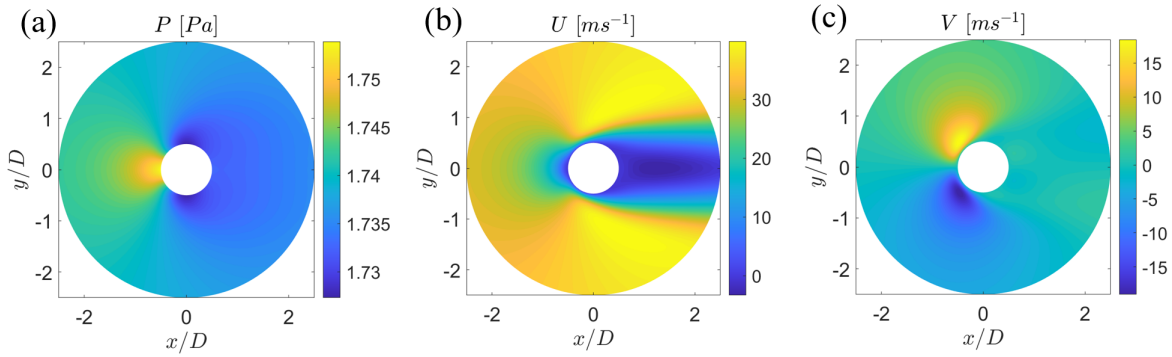

Figure 8: The pressure field (a), horizontal velocity field (b), and vertical velocity field (c) outputted by the CFD routine for laminar flow over an infinite cylinder.

Comparing the results of Figure 8 to the case of inviscid potential flow around the same body, clear distinctions in the flow behaviour can be made due to the presence of viscosity. Figure 8 (a) shows a

small high pressure region near the leading face of the cylinder with a large low pressure region seen near the trailing face, giving rise to a wake structure. This opposes the pressure distribution expected for inviscid flow that is symmetric about the vertical axis and has no presence of a wake. Figure 8 (b) further emphasises this presence of a wake as the horizontal velocity is seen to be amongst its lowest values in this trailing region. In contrast, the inviscid case predicts recovery to a purely horizontal velocity downstream of the cylinder as no momentum is lost due to viscous effects. Lastly, in the inviscid case, any presence of a vertical velocity component is entirely due to the streamline curvature around the cylinder with no separation occurring. Given this, we would expect to see vertical velocity components that indicate the flows movement around the cylinder which is seen in Figure 8 (c) near the leading face with positive and negative regions indicating the flows movement, away from the centre line. However, on the contrary to inviscid flow, opposing regions of vertical velocity indicating the flows movement back towards the centre line near the trailing face of the cylinder is not seen in the viscous case, indicating flow separation as the flow no longer conforms to the cylindrical contours. Instead, the vertical velocity distribution near the trailing face is seen to be near zero, indicating a purely horizontal flow velocity continuing in the direction of the wake and therefore separating itself from the cylinder's trailing face.

In order to analyse this flow that is attached to the cylinder's surface, it is far more convenient to consider the radial and tangential velocity components (U_r and U_θ respectively). The new velocity field was obtained by applying the below coordinate transformation, calculating r and θ at each grid point (x, y).

$$U_r = U \cdot cos(\theta) + V \cdot sin(\theta) \tag{3}$$

$$U_\theta = V \cdot cos(\theta) - U \cdot sin(\theta) \tag{4}$$

As such the radial and tangential velocity fields can be visualised as done in Figure 9 below.

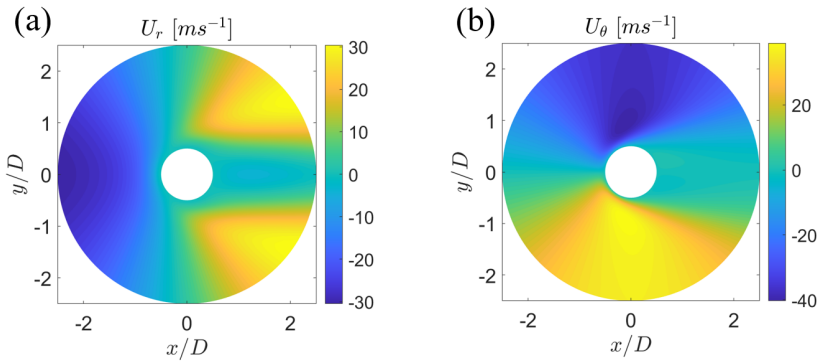

Figure 9: The radial velocity field (a) and tangential velocity field (b) calculated from the outputted horizontal and vertical components for flow over an infinite cylinder. The polar coordinate frame is centred about the cylinder with θ measured from the positive x-axis in the anti-clockwise direction as viewed in this figure.

In order to study the behaviour of boundary layers in this particular flow case, we must analyse the velocity component that moves tangent to the surface of the cylinder. The tangential velocity U_θ is therefore the most appropriate choice given boundary layers, at the most fundamental level, develop along a surface in the streamwise direction and thus in the direction of U_θ. In contrast, U_r is, by definition, everywhere normal to the cylinder's surface and therefore will not capture the development of a boundary layer along the cylinder's surface.

Subsequently, in order to more closely study the development of the boundary layer around this cylinder, the tangential velocity distribution as a function of r ($U_\theta(r)$) was plotted in Figure 10 below for eight uniformly spaced angular positions on the cylinder's surface for $\theta \in [\pi, 2\pi]$.

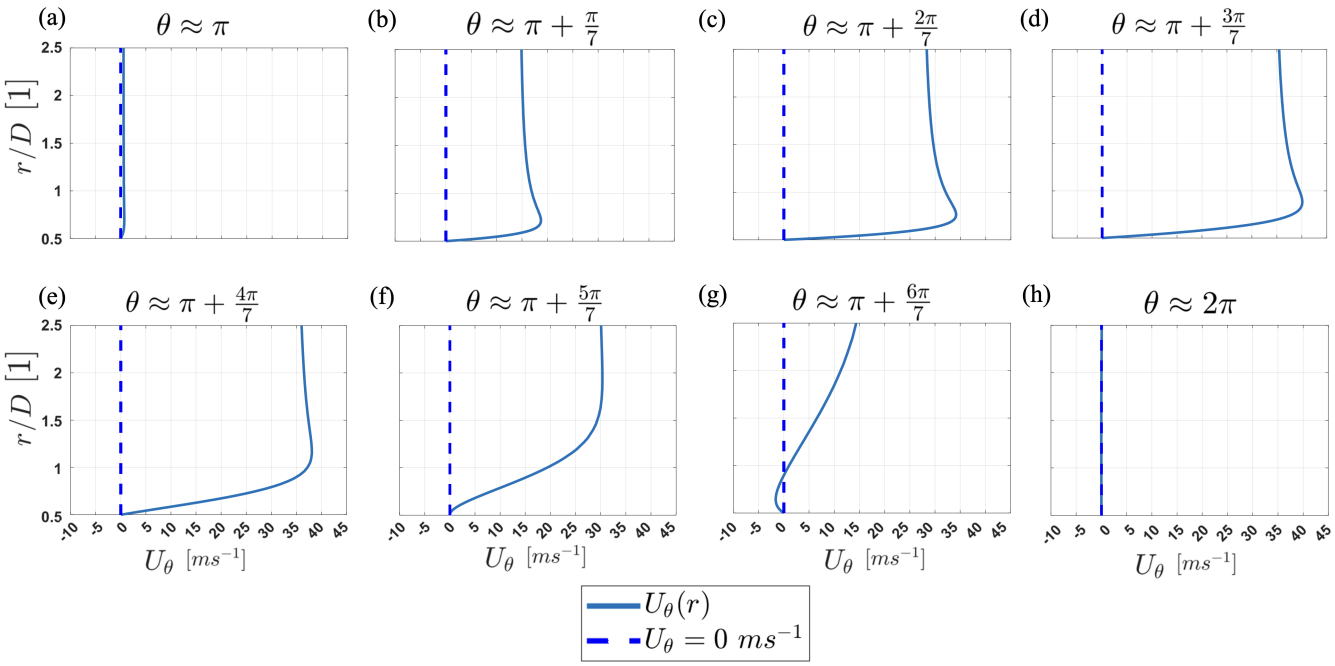

Figure 10: The tangential velocity distribution as a function of radial distance plotted for eight uniformly spaced angular positions on the cylinder's surface.

Here Figure 10 visually depicts the clear formation and evolution of a boundary layer for the tangential velocity around the cylinder in sub-figures (a) through (h). The formation of this boundary layer above the cylindrical surface in fact shares many common features with that of the flat plate flow case. Figure 10 shows the existence of a no-slip boundary condition at the wall with $U_\theta = 0$ at $r = D/2$ for all values of θ, aligning with the same boundary condition for the flat plate case. Furthermore, as θ increases, not only is an increase in boundary layer thickness observed, but also an increasingly steep velocity gradient at the wall and hence increasing shear stress is also seen. These features are consistent with the growth of both boundary layer thickness and shear stress present in a boundary layer above a flat plate, aligning with the expected increase in these features with increasing streamwise distance x, of which θ is analogous to in the cylindrical case. Moreover, U_θ is seen to plateau for larger values of r/D, suggesting a similar transition to free-stream velocity as that for the flate plate case.

Despite these similarities, there exist key differences in the velocity profile that differentiate the cylindrical case from flow over a flat plate. In particular, it is seen that this 'free-stream' velocity at which U_θ plateaus, is seen to vary greatly across different values of θ. This differs to the everywhere uniform free-stream velocity that is present in the case of flow over a flat plate. Most strikingly is the existence of reversed and therefore separated flow in the cylindrical case shown in Figure 10 (g), that was previously hinted at in Figure 8 (b) and (c). This is in stark comparison to ideal laminar flow over a flat plate for which at all points, U is not reversed and therefore is anywhere ≥ 0.

Although these differences are present, there are several possible reasons for their existence. As alluded to by Figure 8 (a), the existence of pressure gradients across the cylinder may be a cause for these distinctions in the velocity distribution between the two cases. In this case, a favourable pressure gradient may act to quicken or impede the free-stream tangential flow, possibly explaining the cause of variation in the velocity at which U_θ stabilises to at large values of r/D. Furthermore, this pressure gradient is likely the reason for flow separation occurring in Figure 10 (g), decelerating the flow to the point of negative magnitude. This is in direct contrast to the near complete absence of pressure variation and therefore pressure gradients in the flat plate case, as illustrated in Figure 11 below.

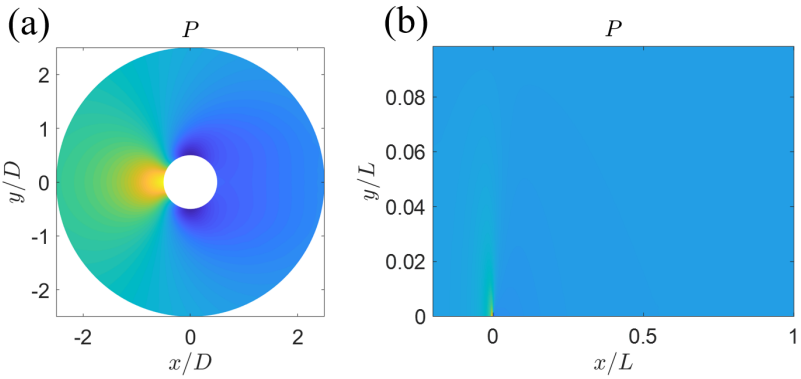

Figure 11: The pressure field contours for both laminar flow over a cylinder (a) and laminar flow across a flat plate (b).

To further explore the apparent flow separation that exists in the cylindrical flow case, the exact point at which separation occurs can was identified by evaluating the tangential velocity gradient at the wall and pinpointing the angular position θ_{sep} where this value equals zero.

$$At \;\; \theta_{sep} : \left.\frac{\partial U_\theta}{\partial r}\right|_{r=D/2} = 0 \tag{5}$$

Doing so saw that separation occurred at both $\theta_{sep} = 309.35°$ and $\theta_{sep} = 50.65°$, where θ is measured anticlockwise from the positive x-axis as viewed in Figure 11 (a) above. Cross-referencing this result with Figure 8 (b) sees agreement with the previous suggestion that separation is occurring on the trailing face of the cylinder. To better understand the cause of separation and attempt to justify the previous reasoning behind its occurrence, it is useful to plot the pressure distribution at the cylinder wall as a function of θ, alongside these values of θ_{sep}, to investigate the possible effects of present pressure gradients. This is illustrated in Figure 12 below.

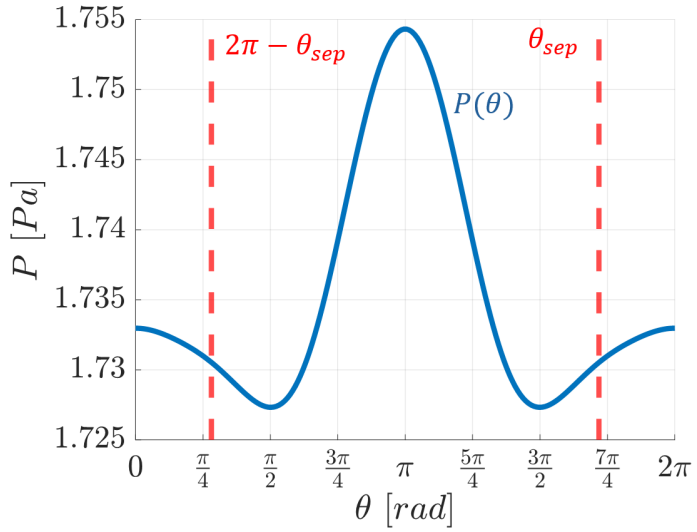

Figure 12: The pressure distribution at the cylinder wall plotted as a function of angular position θ.

As shown in Figure 12 above, separation is seen to occur in regions where pressure is increasing in the direction of flow, therefore indicating a positive adverse pressure gradient. Note that this is the

case due to the definition of θ causing the sign of U_θ to be negative on the top $(+y)$ side of the cylinder and positive on the bottom $(-y)$ of the cylinder. As previously theorised and now confirmed, this adverse pressure gradient slows the boundary layer flow near the wall, reducing the velocity gradient and hence wall shear stress. Once this wall shear stress reaches zero, the flow is reversed and hence flow separation occurs.

3 Conclusions

To conclude, this investigation explored the use of computational fluid dynamics as a means of studying the development of boundary layers in three distinct cases: laminar and turbulent incompressible flow over a flat plate, and compressible laminar flow over a cylinder.

In the case of incompressible, laminar flow across a flat plate, key quantities including boundary layer thickness δ, displacement thickness δ^*, momentum thickness θ, and local friction coefficient c_f were evaluated using the resultant CFD data and plotted as a function of streamwise distance x. In doing so, all computed quantities and their evolution was seen to show excellent agreement with what is predicted by the Blasius boundary layer solution. The total drag coefficient was calculated to be $C_{f,CFD} = 0.0011$, closely matching the theoretical value of $C_{f,Blasius} = 0.0012$. Furthermore, the streamwise velocity profile was seen to exhibit self-similarity when the distance orthogonal to the wall was normalised by δ, with the distributions collapsing onto the same curve across different positions of x. Similarly, boundary-layer-like behaviour was also observed in the temperature field for the $T_{wall} = 148.81\ K$ and $T_{wall} = 350\ K$ cases, however not observed for the $T_{wall} = T_\infty$ case. This was further confirmed by evaluating the temperature boundary layer thickness $\delta_{T,99}$, which showed similar growth with x compared to δ. Self-similarity of the temperature profile in both the $T_{wall} = 148.81\ K$ and $T_{wall} = 350\ K$ cases was also shown to exist.

For the case of incompressible turbulent flow across a flat plate, the same key boundary layer quantities were also evaluated. It was seen that the values of δ, δ^*, θ, and c_f were significantly larger than the laminar case, consistent with what is predicted by turbulent boundary layer theory. Moreover, the drag coefficient $C_f = 0.0029$ was found to be more than double that found in the laminar case, reflecting higher wall shear stress due to turbulent mixing and further aligning with theoretical expectations. Using the same methods described for the laminar flat plate case, self-similarity was not observed in the turbulent case, indicating the turbulent boundary layer structure to not settle into a universal form.

Finally, for the case of laminar compressible flow over an infinite cylinder, differences between the viscous simulated flow and the inviscid case were identified, including the presence of a wake and most notably, the evidence of flow separation. To more closely analyse this, the development of the tangential velocity profile was evaluated at several angular positions, giving rise to boundary layer behaviour in tangential velocity being exhibited. This tangential velocity boundary layer was seen to mirror the behaviour of a flat plat boundary layer with key differences including variable 'free-stream' velocity and separation being observed. The angular position of this separation was identified by finding where the tangential velocity gradient at the wall was equal to zero, resulting in separation being found at $\theta_{sep} = 309.35°$ and $\theta_{sep} = 50.65°$. Pressure gradients were presented as a possible reason for this separation and subsequently confirmed by plotting the pressure distribution at the cylinder wall as a function of angular position θ. Observing the pressure gradient at $\theta_{sep} = 309.35°$ and $\theta_{sep} = 50.65°$ confirmed that separation occurred in regions with adverse pressure gradients that opposed the direction of the flow, confirming reasoning and aligning with theoretical expectations.

4 References

[1] T. J. Chung, *Computational fluid dynamics* (Cambridge University Press, 2010).

[2] H. Schlichting and K. Gersten, *Boundary-layer theory* (Springer Nature, 2017).

Appendices

A Relevant Equations

A.1 General Boundary Layer Equations (Numerical Approach)

$$For\ each\ x:\ U(y=\delta) = 0.99 \cdot U_\infty \tag{6}$$

$$\delta^* = \int 1 - \frac{U(y)}{U_\infty} dy \tag{7}$$

$$\theta = \int \frac{U(y)}{U_\infty}(1 - \frac{U(y)}{U_\infty})dy \tag{8}$$

$$c_f = \frac{2\mu}{\rho_\infty U_\infty^2} \cdot \left.\frac{\partial U}{\partial y}\right|_{y=0} \tag{9}$$

$$C_f = \frac{1}{L}\int_0^L c_f(x)dx \tag{10}$$

A.2 Blasius Solution for Boundary Layer Flow

$$\delta = \frac{5 \cdot x}{\sqrt{Re_x}} \tag{11}$$

$$\delta^* = \frac{1.72 \cdot x}{\sqrt{Re_x}} \tag{12}$$

$$\theta = \frac{0.664 \cdot x}{\sqrt{Re_x}} \tag{13}$$

$$c_f = \frac{0.664}{\sqrt{Re_x}} \tag{14}$$

$$C_f = \frac{1.328}{\sqrt{Re_x}} \tag{15}$$

B MATLAB Code

B.1 Incompressible Laminar Flow Over a Flat Plate

```
% Wrapper for SU2

set(0,'DefaultLineLineWidth',2,...
    'DefaultLineMarkerSize',10,...
    'DefaultAxesFontSize',18,...
    'DefaultTextInterpreter','latex',...
    'DefaultLegendInterpreter','latex',...
    'DefaultLegendFontSize',16,...
    'DefaultFigureColor',[1 1 1])

clear all

% Run SU2 on terminal, if you have already produced a .csv, you can comment
% this if statement out to use the rest of the wrapper without running the
% CFD again.
% if ispc
%     system('SU2_CFD.exe lam_flatplate.cfg');
```

```
% else
%     system('./SU2_CFD lam_flatplate.cfg');
% end

%%

% Run importfile.m to extract data from the .csv file

% temperature case 1: T_wall = 148.81 K
[~, ~, Pressure1, Velocity_x1, Velocity_y1, Temperature1,...
    Pressure_Coefficient1, Density1, Laminar_Viscosity1,...
    Skin_Friction_Coefficient_x1, Skin_Friction_Coefficient_y1,...
    Heat_Flux1, Y_Plus1] = importfile('T148_restart_flow.csv');

% temperature case 2: T_wall = T_free-stream
[~, ~, Pressure2, Velocity_x2, Velocity_y2, Temperature2,...
    Pressure_Coefficient2, Density2, Laminar_Viscosity2,...
    Skin_Friction_Coefficient_x2, Skin_Friction_Coefficient_y2,...
    Heat_Flux2, Y_Plus2] = importfile('TFS_restart_flow.csv');

% temperature case 3: T_wall = 350 K
[x, y, Pressure3, Velocity_x3, Velocity_y3, Temperature3,...
    Pressure_Coefficient3, Density3, Laminar_Viscosity3,...
    Skin_Friction_Coefficient_x3, Skin_Friction_Coefficient_y3,...
    Heat_Flux3, Y_Plus3] = importfile('T350_restart_flow.csv');


% Find number of grid points
aux=x==x(1);
nx=length(find(aux==true)); %Number of x grid points
ny=length(x)/nx; %Number of y grid points

% Reshape large column vectors from .csv into an nX x nY grid
xgrid=reshape(x,nx,ny);
ygrid=reshape(y,nx,ny);

U1=reshape(Velocity_x1,nx,ny);
V1=reshape(Velocity_y1,nx,ny);
T1=reshape(Temperature1,nx,ny);

U2=reshape(Velocity_x2,nx,ny);
V2=reshape(Velocity_y2,nx,ny);
T2=reshape(Temperature2,nx,ny);

U3=reshape(Velocity_x3,nx,ny);
V3=reshape(Velocity_y3,nx,ny);
T3=reshape(Temperature3,nx,ny);

P1=reshape(Pressure1,nx,ny);

xline=xgrid(1,:);
yline=ygrid(:,1);

figure
```

```
contourf(xgrid/max(xline),ygrid,U1.*69.1687,128,'Linestyle','none')
xlabel('$x/L$')
ylabel('$y/L$')
title('$U\ [ms^{-1}]$')
ylim([0, 0.002])
colorbar

figure
contourf(xgrid/max(xline),ygrid/max(xline),V1.*69.1687,128,...
    'Linestyle','none')
xlabel('$x/L$')
ylabel('$y/L$')
title('$V\ [ms^{-1}]$')
ylim([0, 0.002])
colorbar

%%

u_inf = 69.1687;
mu = 1.83463*10^-5;
rho_inf = 1.13235;
cp = 1004.703;
Pr = 0.72;

d_99_list = [];
d_star_list = [];
theta_list = [];
cf_list = [];

y_index = length(yline):-1:1;

x_index = 1:1:length(xline);

x_vals_fine = min(xline):0.00001:max(xline);
y_vals_fine = min(yline):0.00001:max(yline);

y_index = 1:1:length(y_vals_fine);

for x = x_index

    u = interp1(yline, U1(:,x), y_vals_fine);

    d_star_list = [d_star_list trapz(y_vals_fine, (1 - u))];

    theta_list = [theta_list trapz(y_vals_fine, (u) .* (1 - u))];

    cf_list = [cf_list (2 * mu * ((U1(64,x)*u_inf-U1(65,x)*u_inf)...
        /(yline(64)-yline(65))))/(rho_inf*u_inf^2) ];

    for y = y_index
        if u(y) >= 0.99
            d_99_list = [d_99_list y_vals_fine(y)];
            break
        end
```

```
    end
end

plotwidth = 3;

figure
hold on
p1 = plot(xline/max(xline), d_99_list, 'DisplayName',...
    '$\delta_{99}$ CFD', 'LineWidth', plotwidth);
p2 = plot(x_vals_fine/max(xline), 5.*x_vals_fine./...
    real(sqrt((u_inf.*rho_inf.*x_vals_fine)./(mu))), 'r--', 'LineWidth',...
    plotwidth);

p3 = plot(xline/max(xline), d_star_list, 'DisplayName',...
    '$\delta^*$ CFD', 'LineWidth', plotwidth);
p4 = plot(x_vals_fine/max(xline),...
    1.72.*x_vals_fine./real(sqrt((u_inf.*rho_inf.*x_vals_fine)./(mu))),...
    'r--', 'LineWidth', plotwidth);

p5 = plot(xline/max(xline), theta_list, 'DisplayName',...
    '$\theta$ CFD', 'LineWidth', plotwidth);
p6 = plot(x_vals_fine/max(xline), 0.664.*...
    x_vals_fine./real(sqrt((u_inf.*rho_inf.*x_vals_fine)./(mu))),...
    'r--', 'DisplayName', 'Blasius Solutions', 'LineWidth', plotwidth);

xlabel("$x/L$ [1]")
ylabel("$Thickness$ [m]")
xlim([min(xline)/max(xline), max(xline)/max(xline)+0.0001])
ylim([0, 1.5*10^-3])
xticks(0:0.25:1)
yticks(0:0.25*10^-3:1.5*10^-3)
legend([p1, p3, p5, p6], 'Location','northwest')
grid on

figure
plot(xline/max(xline), cf_list, 'LineWidth', plotwidth)
xlabel("$x/L$ [1]")
ylabel("$c_f$ [1]")
xlim([0, max(xline)/max(xline)])
ylim([0, max(cf_list(22:65))])
hold on
% plot c_f from Blasius solution
plot(x_vals_fine/max(xline),...
    0.664./real(sqrt((u_inf.*rho_inf.*x_vals_fine)./(mu))),...
    'r--', 'LineWidth', plotwidth+0.5)
xticks(0:0.25:1)
legend(["CFD", "Blasius Solution"])
grid on

% computing C_D from CFD
C_D_CFD = (1/max(xline)) * trapz(xline(22:65), cf_list(22:65));
disp(C_D_CFD)

disp('per unit length:')
```

```
disp(C_D_CFD/max(xline))

% computing C_D from Blasius
C_D_Blasius = 1.328./sqrt((u_inf.*rho_inf.*max(xline))./(mu));
disp(C_D_Blasius)

% relative error between C_D values
rel_err_percentage = (C_D_CFD - C_D_Blasius)*100/C_D_Blasius;
disp(rel_err_percentage)

%% normalising y by boundary layer thickness for the plots of local
% streamwise velocity distributions to observe self-similarity in the flow

figure
hold on
for x = x_index
    if x == 25 || x == 40 || x == 50 || x == 58 || x == 65
        u = U1(:,x);
        plot(U1(:,x).*u_inf, yline./d_99_list(x), 'LineWidth', plotwidth)
    end
end

plot(zeros(length(U1(:,1))) + u_inf, yline./d_99_list(x), 'r--')
ylim([0, 1.25])
xlim([0, 80])
yticks(0:0.2:1.25)
ylabel("$\frac{y}{\delta_{99}}$ [1]")
xlabel("U [$ms^{-1}$]")
legend(["$x/L = 0.01$", "$x/L = 0.10$", "$x/L = 0.25$", "$x/L = 0.53$",...
    "$x/L = 1.0$", "$U_\infty=69.17\ ms^{-1}$"], 'Location', 'northwest')
grid on

%% temperature plots for T_wall = T_inf and T_wall = 350 K

T_inf = 297.62;
T_wall_1 = 148.81;
T_wall_2 = T_inf;
T_wall_3 = 350;

figure
plot(yline, T1(:,58)*T_inf, 'LineWidth', plotwidth)
hold on
plot(yline, T2(:,58)*T_inf, 'LineWidth', plotwidth)
plot(yline, T3(:,58)*T_inf, 'LineWidth', plotwidth)
yticks(0:50:375)
xlim([0, 0.005])
ylim([0,375])
xlabel("y [$m$]")
ylabel("T [$K$]")
legend(["$T_{wall}=148.81K$", "$T_{wall}=T_{\infty}$",...
    "$T_{wall}=350K$"], 'location', 'southeast')
grid on

%%
```

```
% get thermal energy thickness derived from the thermal boundary layer
% integral equation
% the thermal energy thickness is a direct and well studied result of the boundary
% layer-like behaviour of temperature in the case of flow over a flat plate

delta_T1_list = [];
delta_T3_list = [];

y_vals_fine = 0:0.000001:max(yline);

y_index = 1:1:length(y_vals_fine);
x_index = 1:1:length(xline);

T1_99_list = [];
T3_99_list = [];

for x = x_index

    t1 = interp1(yline, T1(:,x), y_vals_fine);
    t3 = interp1(yline, T3(:,x), y_vals_fine);
    for y = y_index
        if (t1(y)*T_inf - T_wall_1)/(T_inf - T_wall_1) >= 0.99
            T1_99_list = [T1_99_list y_vals_fine(y)];
            break
        end
    end

    for y = y_index
        if (t3(y)*T_inf - T_wall_3)/(T_inf - T_wall_3) >= 0.99
            T3_99_list = [T3_99_list y_vals_fine(y)];
            break
        end
    end
end


% T1_temp = (T1(:,60)*T_inf - T_wall_1)/(T_inf - T_wall_1);
% T3_temp = (T3(:,60)*T_inf - T_wall_3)/(T_inf - T_wall_3);
%
% figure
% hold on
%
% title('yeah')
% plot(yline, T1_temp)
% plot(yline, T3_temp, '--')
%

figure
hold on
grid on

plot(xline/max(xline), T1_99_list, 'r', 'LineWidth', plotwidth)
```

```
plot(xline/max(xline), T3_99_list, 'b--', 'LineWidth', plotwidth)
xlabel("$x/L$ [1]")
ylabel("$\delta_{T,99}$ [m]")
xlim([min(xline)/max(xline)-0.0001, max(xline)/max(xline)])
ylim([0, 0.0015])
xticks(0:0.25:1)
yticks(0:0.25*10^-3:1.5*10^-3)
legend(["$T_w=148.81\ K$", "$T_w=350\ K$"], 'location', 'northwest')

%% normalising y by boundary layer thickness for the plots of local
% streamwise velocity distributions to observe self-similarity in the flow

% TODO: self-similar temperature boundary layer

figure
hold on
for x = x_index
    if x == 25 || x == 40 || x == 50 || x == 58 || x == 65
        plot(T1(:,x).*T_inf, yline./T1_99_list(x), 'LineWidth', plotwidth)
    end
end

plot(zeros(length(T1(:,1)))+T_inf, yline./T1_99_list(x),'r--')
ylim([0, 1.5])
xlim([100, 350])
ylabel("$\frac{y}{\delta_{T,99}}$ [1]")
xlabel("T [$K$]")
legend(["$x/L = 0.01$", "$x/L = 0.10$", "$x/L = 0.25$", "$x/L = 0.53$",...
    "$x/L = 1.0$", "$T_\infty=297.62\ K$"], 'Location', 'northwest')
grid on

figure
hold on
for x = x_index
    if x == 25 || x == 40 || x == 50 || x == 58 || x == 65
        plot(T3(:,x).*T_inf, yline./T3_99_list(x), 'LineWidth', plotwidth)
    end
end

plot(zeros(length(T3(:,1)))+T_inf, yline./T3_99_list(x),'r--')
ylim([0, 1.5])
xlim([250, 350])
xticks(250:25:350)
ylabel("$\frac{y}{\delta_{T,99}}$ [1]")
xlabel("T [$K$]")
legend(["$x/L = 0.01$", "$x/L = 0.10$", "$x/L = 0.25$", "$x/L = 0.53$",...
    "$x/L = 1.0$", "$T_\infty=297.62\ K$"], 'Location', 'northwest')
grid on

%% pressure field for comparison with infinite cylinder

figure
contourf(xgrid./max(xline),ygrid./max(xline),P1,128,'Linestyle','none')
xlabel('$x/L$')
```

```
ylabel('$y/L$')
title('$P$')
% colorbar
```

B.2 Incompressible Turbulent Flow Over a Flat Plate

```
% Wrapper for SU2

set(0,'DefaultLineLineWidth',2,...
    'DefaultLineMarkerSize',10,...
    'DefaultAxesFontSize',18,...
    'DefaultTextInterpreter','latex',...
    'DefaultLegendInterpreter','latex',...
    'DefaultLegendFontSize',16,...
    'DefaultFigureColor',[1 1 1])

clear all

% Run SU2 on terminal, if you have already produced a .csv, you can comment
% this if statement out to use the rest of the wrapper without running the
% CFD again.
% if ispc
%     system('SU2_CFD.exe turb_flatplate.cfg');
% else
%     system('./SU2_CFD turb_flatplate.cfg');
% end

%%

% Run importfile.m to extract data from the .csv file
[x, y, Pressure, Velocity_x, Velocity_y, Temperature,...
    Pressure_Coefficient, Density, Laminar_Viscosity,...
    Skin_Friction_Coefficient_x, Skin_Friction_Coefficient_y,...
    Heat_Flux, Y_Plus] = importfile('restart_flow.csv');

%Find number of grid points
aux=x==x(1);
ny=length(find(aux==true)); %Number of x grid points
nx=length(x)/ny; %Number of y grid points

% Reshape large column vectors from .csv into an nX x nY grid
xgrid=reshape(x,ny,nx);
ygrid=reshape(y,ny,nx);

U=reshape(Velocity_x,ny,nx);
V=reshape(Velocity_y,ny,nx);
T=reshape(Temperature,ny,nx);

xline=xgrid(1,:);
yline=ygrid(:,1);

figure
contourf(xgrid/max(xline),ygrid/max(xline),U,128,'Linestyle','none')
xlabel('$x/L$')
```

```
ylabel('$y/L$')
title('$U$')
colorbar

figure
contourf(xgrid/max(xline),ygrid/max(xline),V,128,'Linestyle','none')
xlabel('$x/L$')
ylabel('$y/L$')
title('$V$')
colorbar

%%

u_inf = 69.4448;
mu = 1.84592*10^-05;
rho_inf = 1.32905;

d_99_list = [];
d_star_list = [];
theta_list = [];
cf_list = [];

y_index = 1:1:length(yline);

x_index = 1:1:length(xline);

x_vals_fine = min(xline):0.00001:max(xline);
y_vals_fine = min(yline):0.00001:max(yline);

y_index = 1:1:length(y_vals_fine);

for x = x_index

    u = interp1(yline, U(:,x), y_vals_fine);

    d_star_list = [d_star_list trapz(y_vals_fine, (1 - u))];

    theta_list = [theta_list trapz(y_vals_fine, (u) .* (1 - u))];

    cf_list = [cf_list (2 * mu * ((u(2)*u_inf-u(1)*u_inf)/...
        (y_vals_fine(2)-y_vals_fine(1))))/(rho_inf*u_inf^2) ];


    for y = y_index
        if u(y) >= 0.99
            d_99_list = [d_99_list y_vals_fine(y)];
            break
        end
    end
end

plotwidth = 3;

figure
```

```
hold on
grid on

p1 = plot(xline/max(xline), d_99_list, 'DisplayName',...
    '$\delta_{99}$ CFD', 'LineWidth', plotwidth);
p3 = plot(xline/max(xline), d_star_list, 'DisplayName',...
    '$\delta^*$ CFD', 'LineWidth', plotwidth);

p5 = plot(xline/max(xline), theta_list, 'DisplayName',...
    '$\theta$ CFD', 'LineWidth', plotwidth);

xlabel("$x/L$ [1]")
ylabel("$Thickness$ [m]")
xlim([min(xline)/max(xline), max(xline)/max(xline)+0.0001])
ylim([0, 0.03])
xticks(0:0.25:1)
yticks(0:0.005:0.04)
legend([p1, p3, p5], 'Location','northwest')


figure
plot(xline/max(xline), cf_list, 'LineWidth', plotwidth)
xlabel("$x/L$ [1]")
ylabel("$c_f$ [1]")
xlim([0, max(xline)/max(xline)])
ylim([0, max(cf_list)])
hold on

xticks(0:0.25:1)

grid on

% computing C_D from CFD
C_D_CFD = (1/max(xline)) * trapz(xline(xline > 0), cf_list(xline > 0));
disp(C_D_CFD)

% TODO: get theoretical value for turbulent plate
CD_theory = (1/max(xline))* trapz(x_vals_fine(x_vals_fine > 0), 0.027 ...
    ./ real(((u_inf .* rho_inf .* x_vals_fine(x_vals_fine > 0)) ...
    ./ mu).^(1/7)));
disp("Theory")
disp(CD_theory)

disp('Per unit length:')
disp(C_D_CFD/max(xline))

% TODO: Debug the CFD and theoretical C_D values above

% TODO: relative error between C_D values


%% normalising streamwise velocity by boundary layer thickness

figure
```

```
hold on
for x = x_index
    if x == 117 || x == 135 || x == 255 || x == 374 || x == 457 ||...
            x == 545
        u = U(:,x);
        plot(U(:,x).*u_inf, yline./d_99_list(x), 'LineWidth', plotwidth)
    end
end

plot(zeros(length(U(:,1))) + u_inf, yline./d_99_list(x), 'r--')
yticks(0:0.2:1.5)
ylim([0, 1.25])
xlim([0, 80])
ylabel("$\frac{y}{\delta_{99}}$ [1]")
xlabel("U [$ms^{-1}$]")
legend(["$x/L = 0.01$", "$x/L = 0.02$", "$x/L = 0.10$", "$x/L = 0.25$",...
    "$x/L = 0.53$", "$x/L = 1.0$", "$U_\infty=69.17 ms^-1$"],...
    'Location', 'northwest')
grid on
```

B.3 Compressible Laminar Flow Over an Infinite Cylinder

```
% Wrapper for SU2

set(0,'DefaultLineLineWidth',2,...
    'DefaultLineMarkerSize',10,...
    'DefaultAxesFontSize',18,...
    'DefaultTextInterpreter','latex',...
    'DefaultLegendInterpreter','latex',...
    'DefaultLegendFontSize',16,...
    'DefaultFigureColor',[1 1 1])

clear all

% Run SU2 on terminal, if you have already produced a .csv, you can comment
% this if statement out to use the rest of the wrapper without running the
% CFD again.
% if ispc
%     system('SU2_CFD.exe lam_cylinder.cfg');
% else
%     system('./SU2_CFD lam_cylinder.cfg');
% end

%%

% Run importfile.m to extract data from the .csv file
[x, y, Pressure, Velocity_x, Velocity_y, Temperature, Pressure_Coefficient, ...
    Density, Laminar_Viscosity, Skin_Friction_Coefficient_x,...
    Skin_Friction_Coefficient_y, Heat_Flux, Y_Plus] =...
    importfile('restart_flow.csv');
```

```
% fix variable assignment as the supplied code imports the wrong variables
Velocity_x_True = Skin_Friction_Coefficient_y;
Velocity_y_True = Heat_Flux;
Pressure_True = Pressure_Coefficient;



%%

% interpolation into polar mesh
nr=200;
nt=200;
r=linspace(0,2,nr)+0.5;
theta=linspace(0,2*pi,nt);

[RR,TT]=meshgrid(r,theta);

yi=RR(:).*sin(TT(:));
xi=RR(:).*cos(TT(:));

xi=reshape(xi,nr,nt);
yi=reshape(yi,nr,nt);

P=griddata(x-0.5,y,Pressure_True,xi,yi);
P=reshape(P,nr,nt);

U=griddata(x-0.5,y,Velocity_x_True,xi,yi);
U=reshape(U,nr,nt);

V=griddata(x-0.5,y,Velocity_y_True,xi,yi);
V=reshape(V,nr,nt);

test=griddata(x-0.5,y,Skin_Friction_Coefficient_y,xi,yi);
test=reshape(test,nr,nt);

test2=griddata(x-0.5,y,Heat_Flux,xi,yi);
test2=reshape(test2,nr,nt);

%% Plot resultant pressure and x, y velocity fields

figure
contourf(xi,yi,P,64,'Linestyle','none')
axis equal
xlabel('$x/D$')
ylabel('$y/D$')
title('$P\ [Pa]$')
colorbar

figure
contourf(xi,yi,U,64,'Linestyle','none')
axis equal
xlabel('$x/D$')
ylabel('$y/D$')
```

```
title('$U\ [ms^{-1}]$')
colorbar

figure
contourf(xi,yi,V,64,'Linestyle','none')
axis equal
xlabel('$x/D$')
ylabel('$y/D$')
title('$V\ [ms^{-1}]$')
colorbar

%% Computing and plotting the radial and tangential components of velocity

plotwidth = 3;

% u_r = u cos(theta) + v sin(theta)
% u_theta = - u sin(theta) + v cos(theta)

x_index = 1:1:200;
y_index = 1:1:200;

U_r = zeros(size(xi));
U_theta = zeros(size(xi));

for x = x_index

    for y = y_index

        U_temp = U(y,x);

        V_temp = V(y,x);

        x_pos = xi(y,x);

        y_pos = yi(y,x);

        theta_pos = atan2(y_pos, x_pos);

        U_r_temp = U_temp*cos(theta_pos) + V_temp*sin(theta_pos);

        U_theta_temp = - U_temp*sin(theta_pos) + V_temp*cos(theta_pos);

        r_pos = sqrt(y_pos^2 + x_pos^2);

        U_r(y,x) = U_r_temp;

        U_theta(y,x) = U_theta_temp;

    end

end

figure
contourf(xi,yi,U_r,64,'Linestyle','none')
```

```
axis equal
xlabel('$x/D$')
ylabel('$y/D$')
title('$U_r\ [ms^{-1}]$')
colorbar

figure
contourf(xi,yi,U_theta,64,'Linestyle','none')
axis equal
xlabel('$x/D$')
ylabel('$y/D$')
title('$U_{\theta}\ [ms^{-1}]$')
colorbar

%% Plotting U_theta for several values of theta
U_inf = U(100, 200);

% figure
% hold on
%
% % theta ~= pi
% plot(U_theta(125,:), r, 'LineWidth', plotwidth)
%
% % theta ~= pi + pi/7
% plot(U_theta(150,:), r, 'LineWidth', plotwidth)
%
% % theta ~= pi + 2pi/7
% plot(U_theta(175,:), r, 'LineWidth', plotwidth)
%
% % theta ~= pi + 3pi/7
% plot(U_theta(190,:), r, 'LineWidth', plotwidth)

% xlabel("$U_{\theta}$")
% ylabel("$r$")

theta_index = 1:1:length(theta);
plot_number = 0;

for theta_i = theta_index
    if theta_i == 101 || theta_i == 114 || theta_i == 129 ||...
            theta_i == 143 || theta_i == 157 || theta_i == 171||...
            theta_i == 186 || theta_i == 200

        if plot_number == 0
            title_var = "$\theta \approx \pi$";
        elseif plot_number == 1
            title_var = "$\theta \approx \pi + \frac{\pi}{7}$";
        elseif plot_number == 2
            title_var = "$\theta \approx \pi + \frac{2\pi}{7}$";
        elseif plot_number == 3
            title_var = "$\theta \approx \pi + \frac{3\pi}{7}$";
        elseif plot_number == 4
            title_var = "$\theta \approx \pi + \frac{4\pi}{7}$";
        elseif plot_number == 5
```

```
            title_var = "$\theta \approx \pi + \frac{5\pi}{7}$";
        elseif plot_number == 6
            title_var = "$\theta \approx \pi + \frac{6\pi}{7}$";
        elseif plot_number == 7
            title_var = "$\theta \approx 2\pi$";
        end

        figure
        % subplot(2, 4, plot_number+1)
        plot(U_theta(theta_i,:), r, 'LineWidth', plotwidth)
        hold on
        grid on
        plot(zeros(length(U_theta(theta_i,:))), r, 'b--', 'LineWidth',...
            plotwidth)
        % legend(["$U_\theta(r)$", "$U_\theta = 0$ $ms^{-1}$"])
        xlabel("$U_\theta$")
        ylabel("$r/D$")
        xlim([-10, 45])
        xticks(-10:5:45)
        yticks(0.5:0.5:2.5)
        set(get(gca, 'XAxis'), 'FontWeight', 'bold');
        set(get(gca, 'YAxis'), 'FontWeight', 'bold');

        title(title_var, 'FontWeight', 'bold')

        plot_number = plot_number + 1;

    end
end


%% pressure field for comparison with laminar flat plate
figure
contourf(xi,yi,P,64,'Linestyle','none')
axis equal
xlabel('$x/D$')
ylabel('$y/D$')
title('$P$')
% colorbar

% presence of adverse pressure gradient in cylinder flow vs now pressure
% gradient in laminar flat plate

% free-stream velocity in laminar flat plate is anywhere parallel to the
% plate boundary, whereas in the case of the cylinder this is not the
% case

%% Finding the value of theta where flow separation occurs

theta_index = 101:1:length(theta);

for theta_val = theta_index

    U_theta_temp =  U(theta_val, :);
```

```
    % find normal derivative of the tangential velocity at cylinder wall
    % via finite difference

    norm_deri = (U_theta_temp(2) - U_theta_temp(1))/(r(2) - r(1));

    disp(norm_deri)

    if norm_deri <= 0

        fprintf("Value of theta at which flow separation occurs: " + ...
            "theta = %.8f\n", theta(theta_val))

        disp(theta_val)

        break

    end

end

figure
plot(U_theta(171,:), r)
xlabel("$U_{\theta}$")
ylabel("$r$")

figure
plot(U_theta(172,:), r)
xlabel("$U_{\theta}$")
ylabel("$r$")

%% Plotting pressure distribution at the cylinder wall as a function of theta
figure
grid on
hold on
plot(theta, P(:,1), 'LineWidth', plotwidth)
xline(theta(172), 'r--', 'LineWidth', plotwidth)
xline(2*pi - theta(172), 'r--', 'LineWidth', plotwidth)
xlabel('$\theta\ [rad]$', 'Interpreter', 'latex')
ylabel('$P\ [Pa]$', 'Interpreter', 'latex')
ylim([1.725,1.755])
yticks(1.725:0.005:1.755)
xticks(0:pi/4:2*pi)
xticklabels({...
    '$0$', ...
    '$\frac{\pi}{4}$', ...
    '$\frac{\pi}{2}$', ...
    '$\frac{3\pi}{4}$', ...
    '$\pi$', ...
    '$\frac{5\pi}{4}$', ...
    '$\frac{3\pi}{2}$', ...
    '$\frac{7\pi}{4}$', ...
    '$2\pi$'...
})
```

```
% legend(["$P(\theta)$", "$\theta = \theta_{sep}$ (flow separation angles)"])
% Set LaTeX interpreter for tick labels
ax = gca;
ax.TickLabelInterpreter = 'latex';

% favourable pressure gradient for theta near theta = pi (negative
% gradient), then adverse pressure gradient (positive gradient)
```