Jet Processing

Question 1.

a.

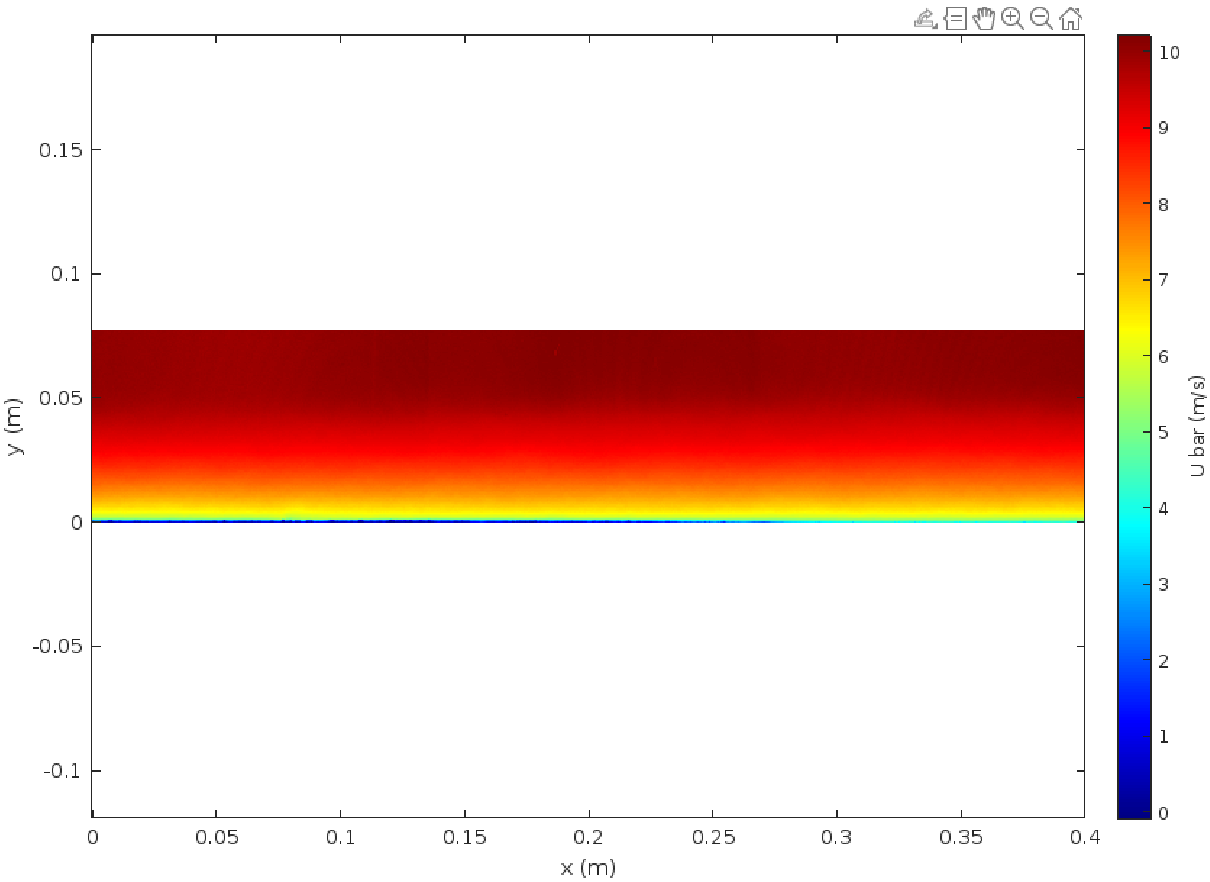

Figure 1: Plot of temporally averaged x-velocity, ubar

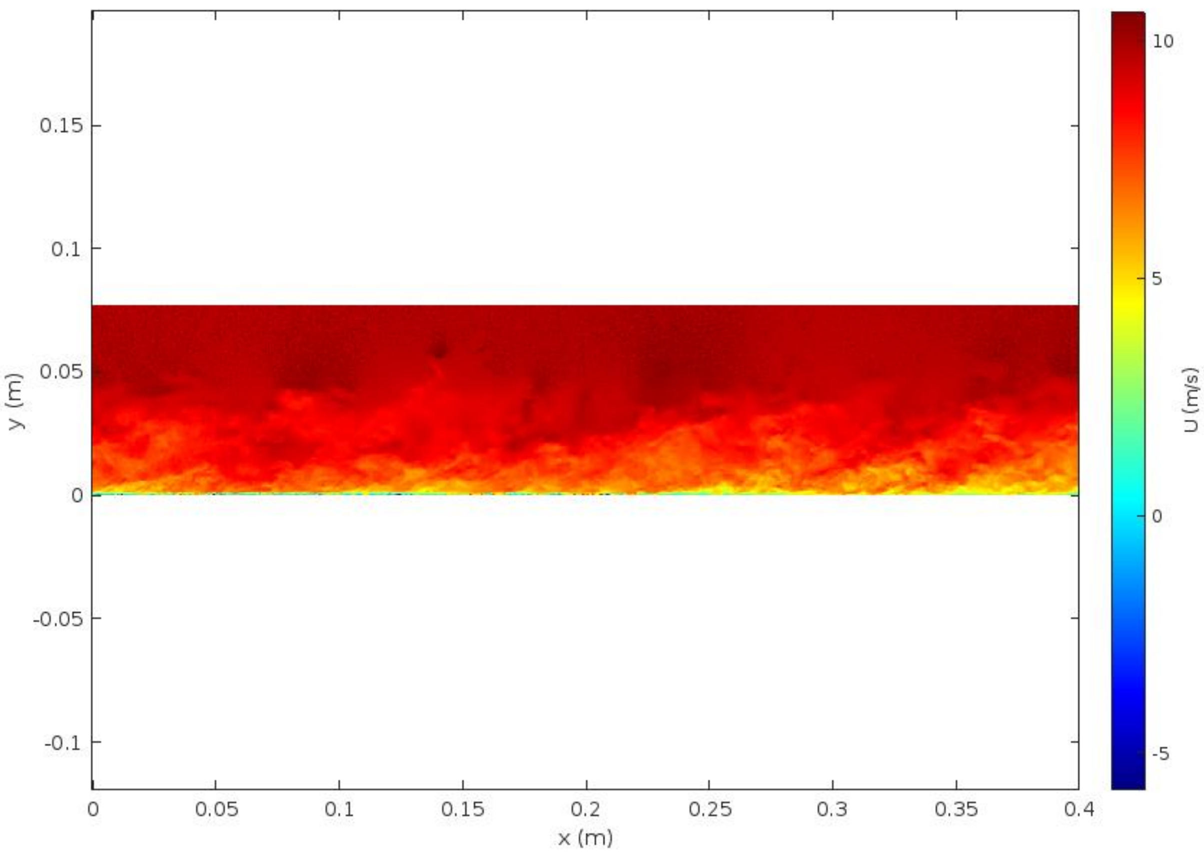

Figure 2: Plot of instantaneous x-velocity

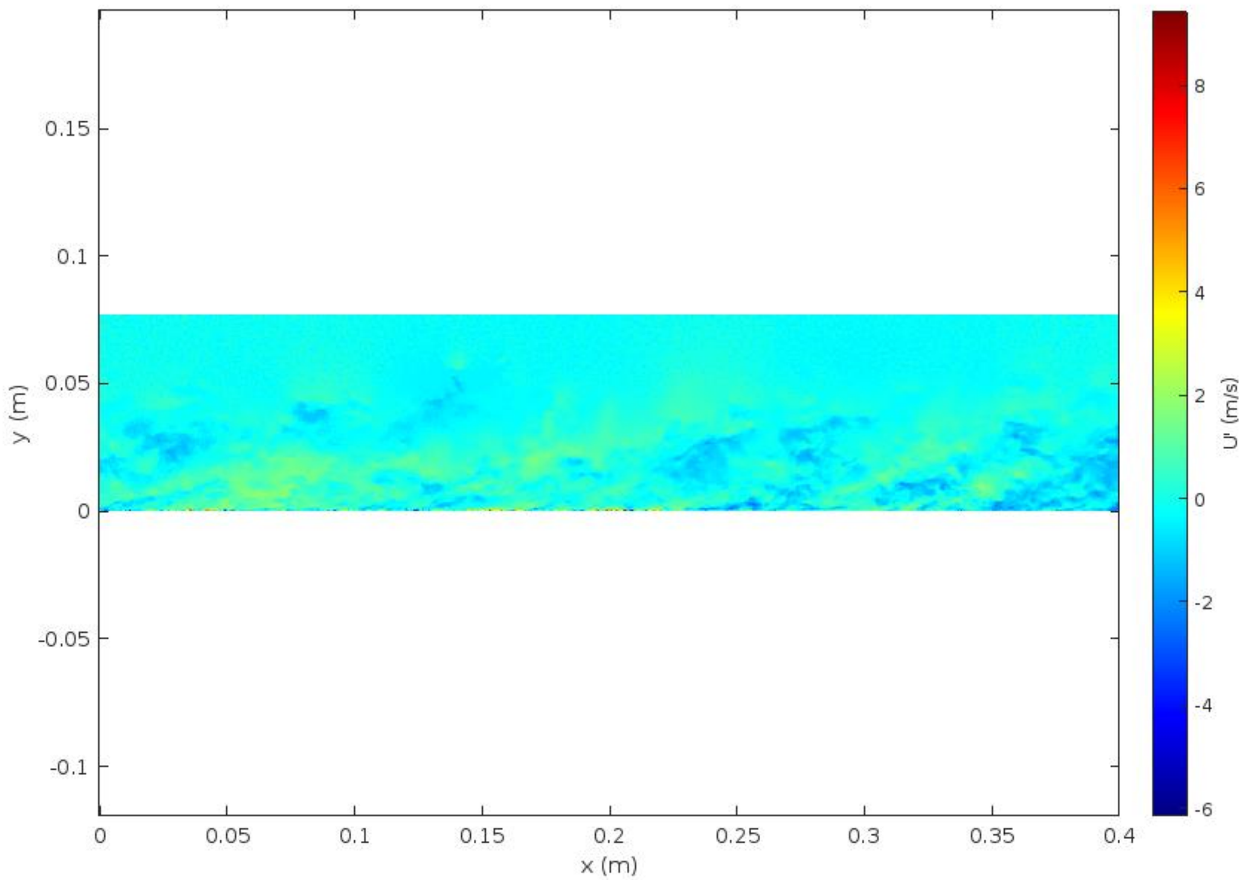

Figure 3: Plot of instantaneous fluctuating x-velocity

b.

The freestream velocity was calculated as the maximum of the temporally-averaged x-velocity as $U_\infty = 10.22\,\frac{\mathrm{m}}{\mathrm{s}}$. The boundary layer thickness was calculated by iterating over all the x-indices, finding the minimum height where $\bar{u}(x, y) > 0.99U_\infty$, and averaging those heights. The computed value is $\delta_{99} = 0.0634\mathrm{m}$. The displacement and momentum thickness were calculated as, respectively:

$$\delta^* = \int_0^\delta \left(1 - \frac{u}{U_\infty}\right) dy = 0.0093\mathrm{m}$$

$$\theta^* = \int_0^\delta \frac{u}{U_\infty}\left(1 - \frac{u}{U_\infty}\right) dy = 0.0069\mathrm{m}$$

Where both displacement and momentum thickness were calculated at each x-station and averaged to arrive at a final value.

c.

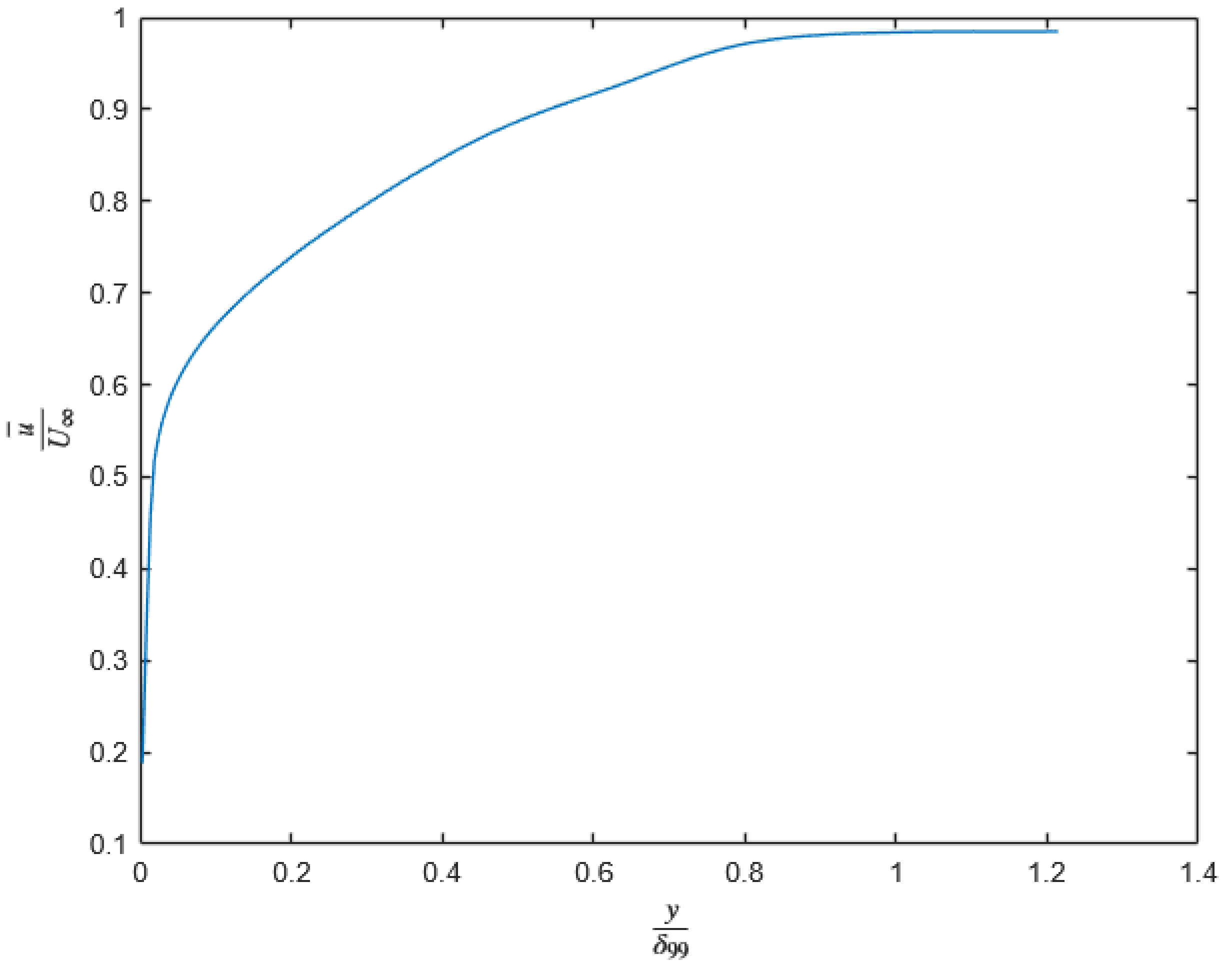

Figure 4: Wall-normal profile of mean velocity

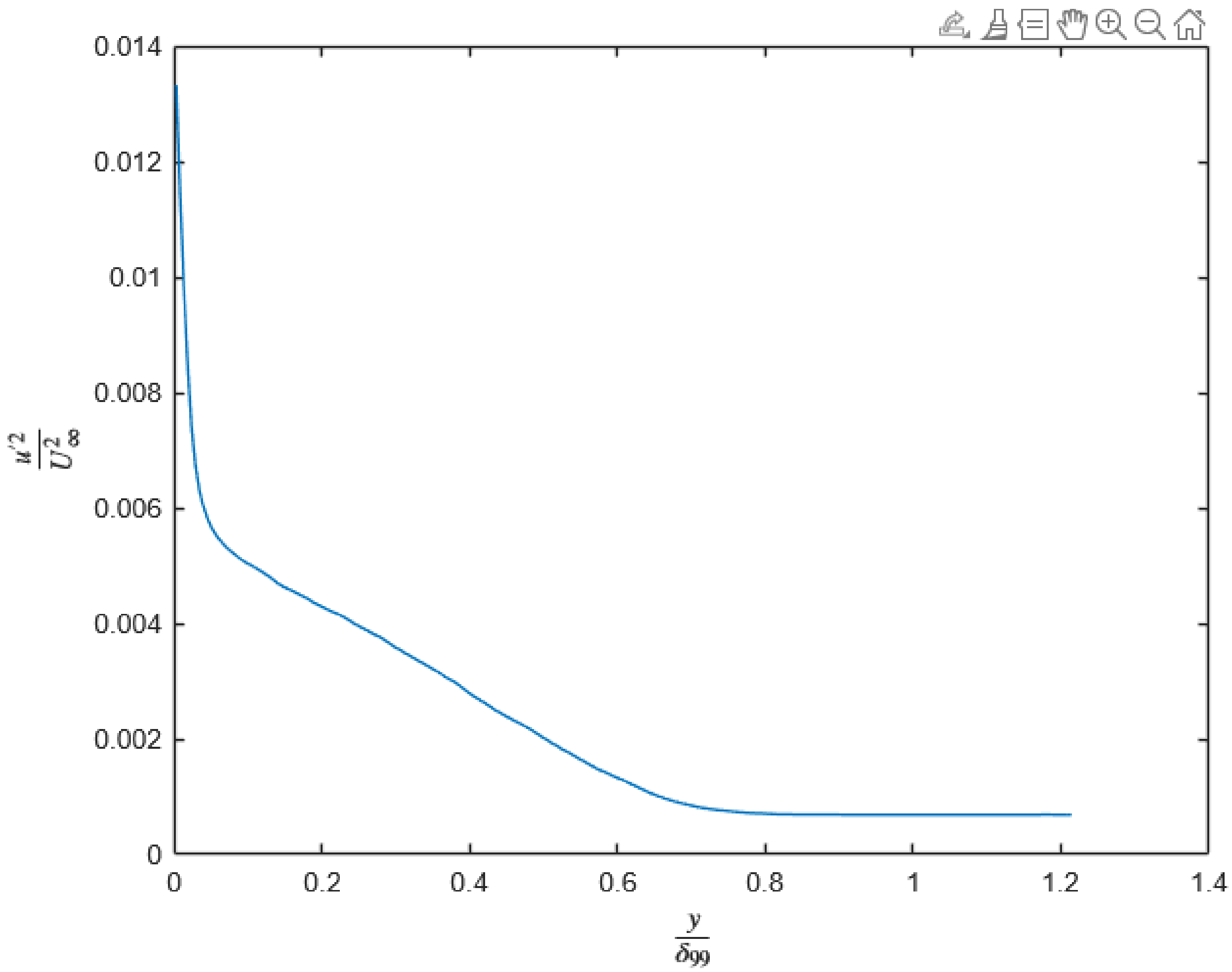

Figure 5: Wall-normal fluctuating velocity

The wall-normal average velocity profile starts near zero. It starts at a non-zero value since the first measurement is not conducted on the wall, but near the wall. The velocity is only zero at the wall. The gradient is initially very steep, up to y/delta~0.02. The gradient then levels out and recovers to near the freestream value as y/delta→1. With a turbulent boundary layer, we expect high levels of momentum mixing to lower the gradients near the boundary layer edge, and increase the gradients near the wall. This behavior is observed.

The wall-normal fluctuating velocity profile appears to start at a peak. At the wall, the fluctuating velocity is 0. Like the mean velocity profile, the first measurement is off the wall, so a zero fluctuating velocity is not observed. The initial values are very high, and the peak appears to happen right next to the wall. Plotting on a log-scaled horizontal axis might reveal different near-wall behavior. Fluctuations decrease farther from the wall, but they do not decrease to zero because of non-zero levels of freestream turbulence.

d.

The equation can be rearranged for x to give:

$$\frac{\delta}{x} = 0.375\left(\frac{xU_\infty}{\nu}\right)^{-0.2}$$

$$x = \left(\frac{\delta}{0.375}\right)^{1.25}\left(\frac{U_\infty}{\nu}\right)^{0.25}$$

This gives a downstream distance for x of $x \approx 3.14\text{m}$.

e.

By assuming that the downstream distance is at the initial x-station of the data, a theoretical ratio between the boundary-layer thickness at the start and end of the data can be calculated:

$$\frac{\delta_{\text{start}}}{\delta_{\text{end}}} = 1.101$$

This indicates an approximate variation of ~10% in boundary layer thickness.

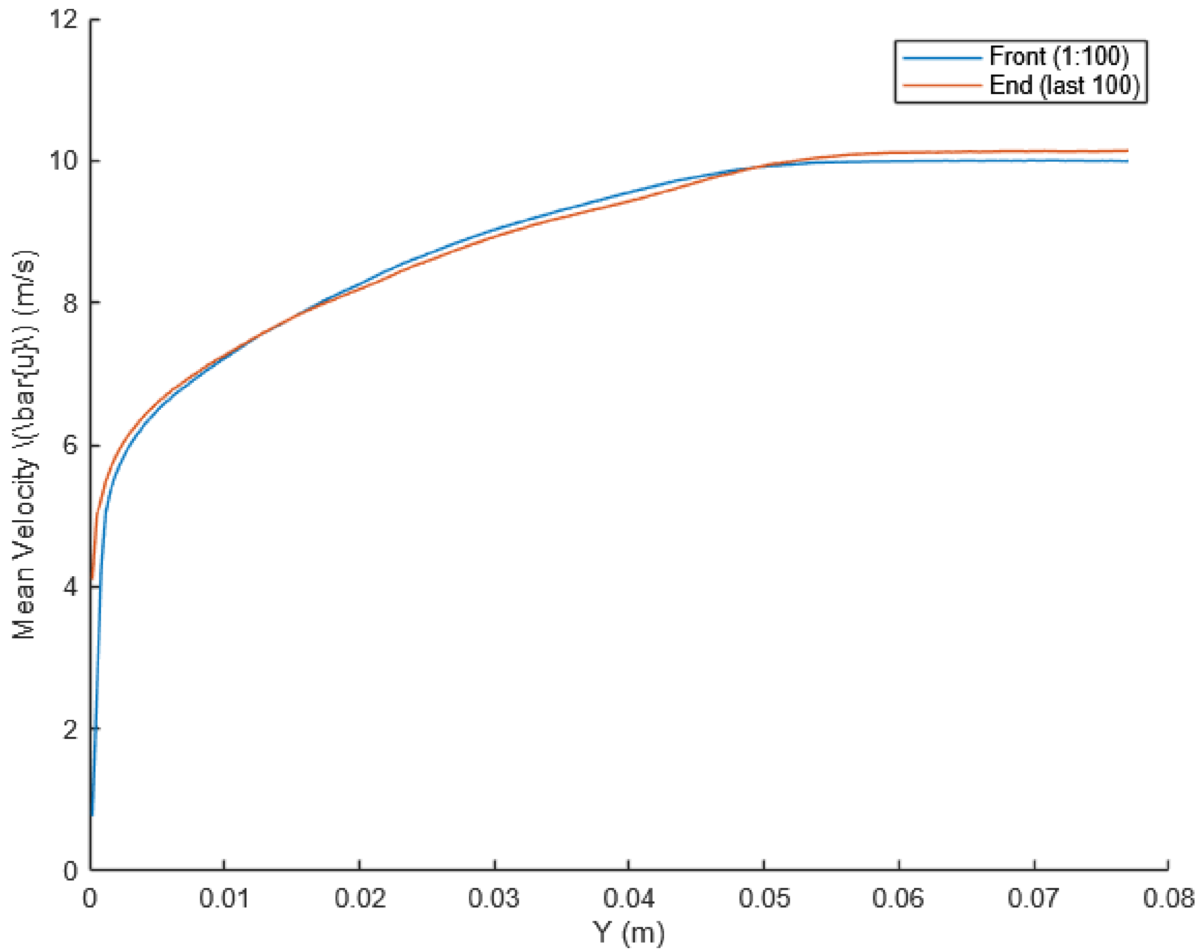

Figure 6: Profiles of wall-normal mean velocity at the start and end of data

The plot of the first and last sections of streamwise-averaged mean velocity support the above assertion. In the range of velocities close to the freestream velocity (8-10m/s), the boundary layer segment near the end reaches the velocity at a higher y-value. This supports the conclusion of a thicker boundary layer. The near-wall gradients appear to be steeper, which might be due to higher turbulent intensity, or measurement artifacts.

Question 2.

a.

Under the condition $\bar{u}^+ = y^+$, the friction velocity can be expressed as $u_\tau = \sqrt{\frac{\overline{u}\nu}{y}}$. This results in a value of $u_\tau = 0.3747\,\frac{\mathrm{m}}{\mathrm{s}}$ and a corresponding value of $y^+ = 5.13$. This relation is valid in the viscous sublayer, for $y^+ < 5$. Therefore, this approximation is likely good. However, since the y-plus value was calculated using the friction velocity which assumed it was in the VSL, it might be inaccurate.

b.

The corresponding Reynolds number is $Re_\tau = 1627$.

c.

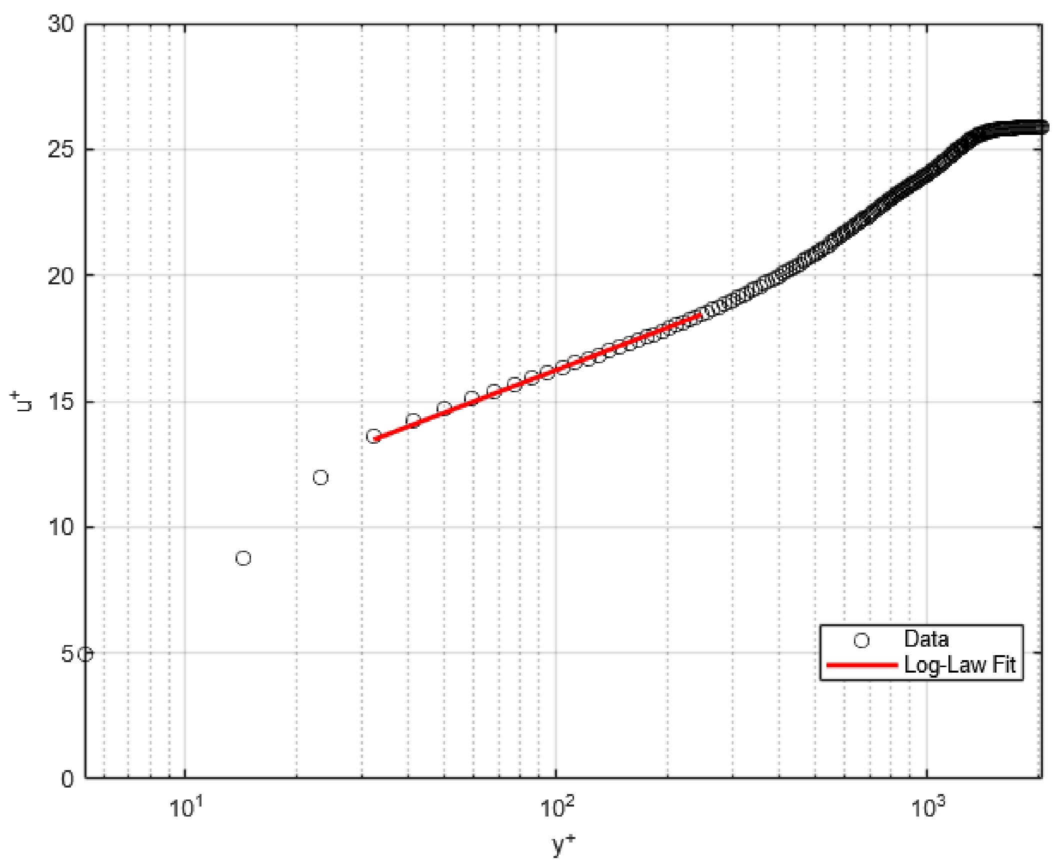

Figure 7: Fitted profile for u+

The above fit was computed using Matlab's fminsearch subroutine. The resultant friction velocity and Reynolds number are $u_\tau = 0.3893\,\frac{\mathrm{m}}{\mathrm{s}}$ and $Re_\tau = 1691$.

d.

Combining equations 3 and 4 gives an expression for the friction velocity in terms of the downstream Reynolds number:

$$\frac{2\rho u_\tau^2}{\rho U_\infty^2} \approx 0.02Re_x^{-0.133}$$

$$u_\tau \approx U_\infty\sqrt{0.01Re_x^{-0.133}}$$

Using this relation, the friction velocity is calculated as $u_{\tau,\mathrm{start}} = 0.3870\,\frac{\mathrm{m}}{\mathrm{s}}$ and $u_{\tau,\mathrm{end}} = 0.3840\,\frac{\mathrm{m}}{\mathrm{s}}$ at the start and end of the domain. The start is assumed to be at the downstream distance calculated in Question 1. This matches the calculations from parts a and c very well, considering that it is based off an experimental correlation. The correlated friction velocity is within $0.01\,\frac{\mathrm{m}}{\mathrm{s}}$ of the measured and fitted velocities.

Over the extent of the domain, the friction velocity is expected to decrease by $0.003\,\frac{\mathrm{m}}{\mathrm{s}}$, which is less than 1% of the total friction velocity. The x-dependence is very weak in the equation for the correlated friction velocity, which explains the insensitivity to downstream distance.

e.

With a laminar boundary layer, the expected friction coefficient is $c_f = 0.00038$, compared to the turbulent boundary layer where $c_f = 0.0029$. The friction coefficient is 7.56 times higher for a turbulent boundary layer.

Question 3.

a.

The wake strength is found to be $\Pi = 0.6493$.

b.

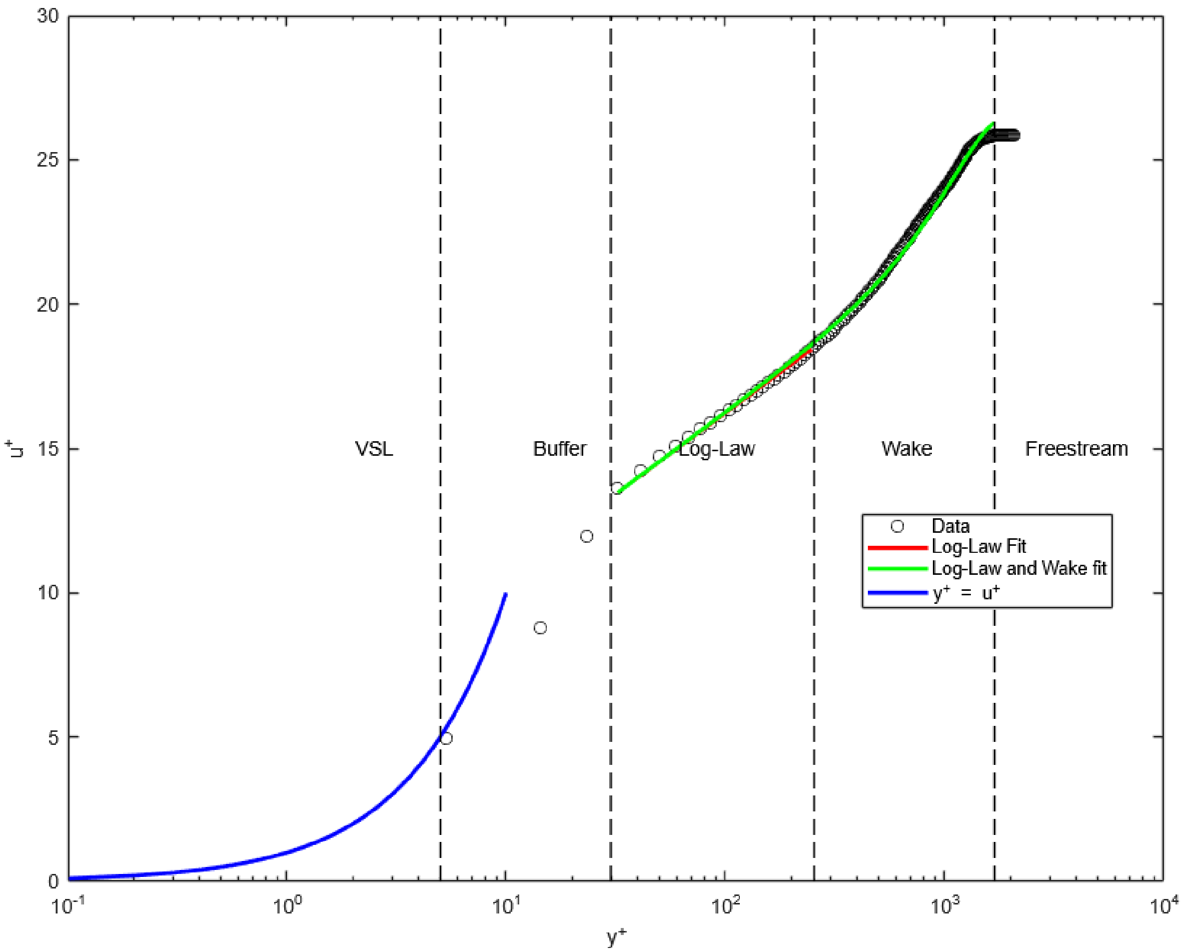

Figure 8: Viscous sublayer, log-law, and combined wake/log-law fits on velocity profile

The approximations match the velocity profile well in the regions they are derived for. In the VSL, the first datapoint is at $y^+ = 5$, and the fit nearly perfectly coincides with the first datapoint. In the log-law region, both the log-law fit and the combined log-law and wake fit capture the behavior of the velocity profile. Towards the end of the wake region, the wake fit does not level out with the experimental velocity profile, and so it overpredicts the velocity near the end of the wake.

c.

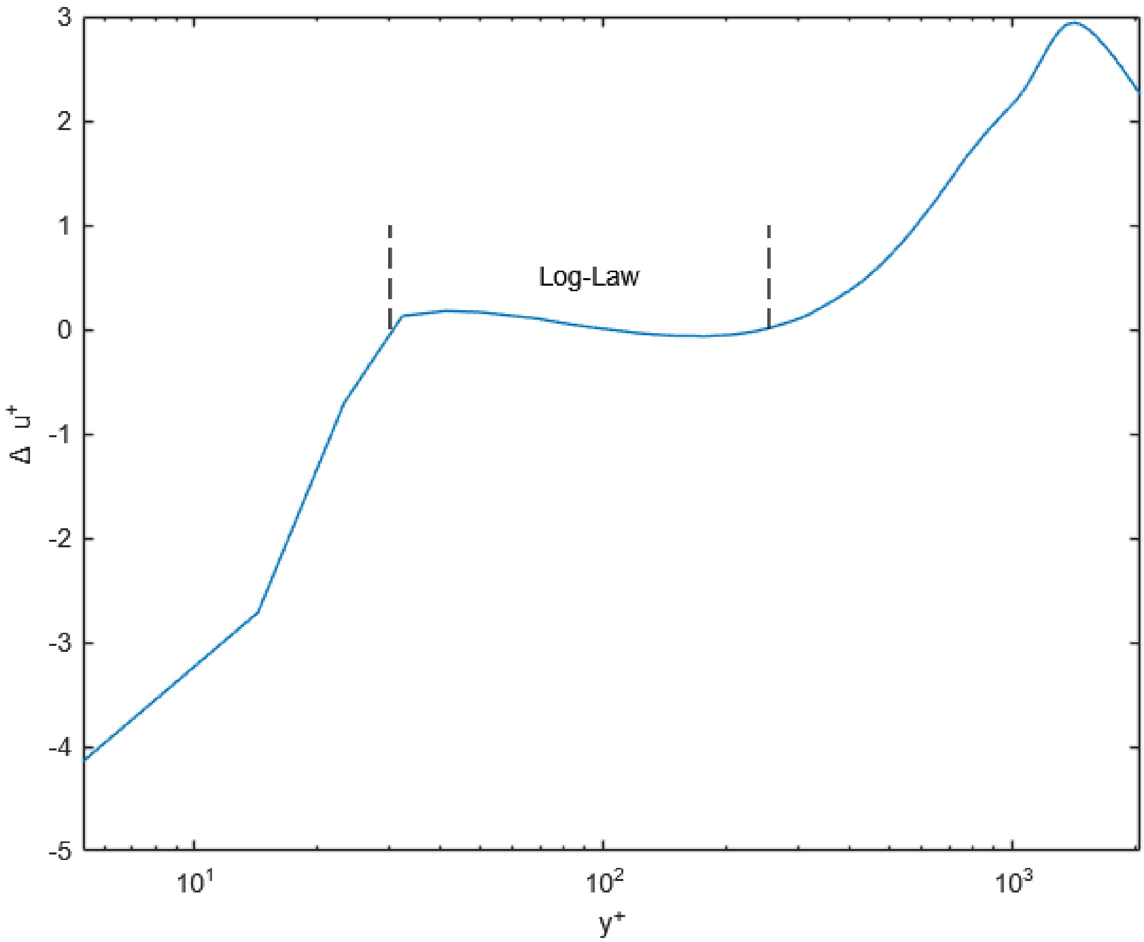

Figure 9: Difference between measured and fitted y plus values.

In the log-law region, the difference between fitted and measured y-plus is very small, on the order of $\Delta y^+ \sim 0.1$. The agreement is good evidence in support of Prandtl's hypothesis that in the log-law region, neither the inner nor outer variables have a large impact on the velocity profile.

Code

```
if ~exist('U', 'var')
    load('data_full.mat');
end
close all;

u_bar = mean(U,3);
u_prime = U-u_bar;
t_0 = 150;

%% Calculate part b %%

% u_inf = mean(u_bar(225:end,:), 'all');
u_inf = max(u_bar, [], 'all');
first_exceed_indices = arrayfun(@(col) find(u_bar(:, col) > 0.99 * u_inf, 1),
1:size(u_bar, 2), 'UniformOutput', false);
y_values = cell2mat(cellfun(@(idx) Y(idx),
first_exceed_indices(~cellfun('isempty', first_exceed_indices)),
'UniformOutput', false));
delta_99 = mean(y_values);

delta_displacement = 0;
count = 0;
for i = 1:size(X,2)
    index_max = first_exceed_indices{i};
    if isempty(index_max) % Skip if empty
        continue;
    end
    sum = 0;
    count = count+1;
    for j = 2:index_max
        sum = sum + (Y(j) - Y(j-1)) * (1 - u_bar(j,i) / u_inf);
    end
    delta_displacement = delta_displacement + sum;
end
delta_displacement = delta_displacement / count;

delta_momentum = 0;
count = 0;
for i = 1:size(X,2)
    index_max = first_exceed_indices{i};
    if isempty(index_max) % Skip if empty
        continue;
    end
    sum = 0;
    count = count+1;
    for j = 2:index_max
        sum = sum + (Y(j) - Y(j-1)) * (1 - u_bar(j,i) / u_inf) *
(u_bar(j,i)/u_inf);
    end
    delta_momentum = delta_momentum + sum;
end
delta_momentum = delta_momentum / count;

turbulence_intensity = 100 * mean(abs(u_prime), 'all')/u_inf;
```

```
%% Calculate part c %%
wall_norm_mean = mean(u_bar, 2);
wall_norm_fluctuating = mean(u_prime.^2, [3,2]);

%% Calculate part d %%
x_approx = (delta_99/0.375)^1.25 * (u_inf/nu)^0.25;

%% Calculate part e
delta_ratio_start_end = (x_approx/(x_approx+X(end)))^0.8;
%% Plot part a %%%%
figure
imagesc(X, Y, u_bar)
set(gca, 'ydir', 'normal')
axis equal
xlabel('x (m)')
ylabel('y (m)')
colormap(jet)
cb = colorbar;
% clim([9 10]);
ylabel(cb, "U bar (m/s)")

figure
imagesc(X, Y, U(:,:,t_0))
set(gca, 'ydir', 'normal')
axis equal
xlabel('x (m)')
ylabel('y (m)')
colormap(jet)
cb = colorbar;
ylabel(cb, "U (m/s)")

figure
imagesc(X, Y, u_prime(:,:,t_0))
set(gca, 'ydir', 'normal')
axis equal
xlabel('x (m)')
ylabel('y (m)')
colormap(jet)
cb = colorbar;
ylabel(cb, "U' (m/s)")

% figure
% imagesc(X, Y, mean(u_prime.^2,3))
% set(gca, 'ydir', 'normal')
% axis equal
% xlabel('x (m)')
% ylabel('y (m)')
% colormap(jet)
% cb = colorbar;
% ylabel(cb, "U' averaged (m/s)")

%% Plot part C
figure
plot(Y/delta_99, wall_norm_mean/u_inf);
```

```
xlabel('$$\frac{y}{\delta_{99}}$$', 'Interpreter', 'latex')
ylabel('$$\frac{\bar{u}}{U_\infty}$$', 'Interpreter', 'latex')

figure
plot(Y/delta_99, wall_norm_fluctuating/(u_inf.^2));
xlabel('$$\frac{y}{\delta_{99}}$$', 'Interpreter', 'latex')
ylabel('$$\frac{u^{\prime 2}}{U_\infty^2}$$', 'Interpreter', 'latex')

%% Plot part e
figure
hold on
plot(Y, mean(u_bar(:,1:100),2), 'DisplayName', 'Front (1:100)')
plot(Y, mean(u_bar(:,end-100:end),2), 'DisplayName', 'End (last 100)')
hold off

xlabel('Y (m)')
ylabel('Mean Velocity \(\bar{u}\) (m/s)')
legend

close all;

%% Calculate Part A
u_avg = mean(u_bar, 2);

u_tau_a = sqrt(u_avg(1)*nu/Y(1));
y_plus_a = Y(1)*u_tau_a/nu;

%% Calculate Part B
Re_tau_a = u_tau_a*delta_99/nu;

%% Calculate Part C
kappa = 0.41; A=5;
y_plus_initial = (Y * u_tau_a) / nu;
start_idx = find(y_plus_initial > 30, 1);
end_idx   = find(y_plus_initial > 0.15*Re_tau_a, 1) - 1;

%Fitting code from chatgpt
model = @(utau) norm( ...
    (u_avg(start_idx:end_idx) / utau) ...  % u+ formulation
    - (1/kappa) * log( (Y(start_idx:end_idx) * utau) / nu ) - A ...
);
utau_guess = 1.0;
u_tau_best = fminsearch(model, utau_guess);
Re_tau_best = (u_tau_best * delta_99) / nu;

%% Calculate Part D
Re_x_start = x_approx*u_inf/nu;
Re_x_end = (x_approx+X(end))*u_inf/nu;
u_tau_start = u_inf*sqrt(0.01*Re_x_start^-0.133);
u_tau_end = u_inf*sqrt(0.01*Re_x_end^-0.133);

%% Calculate Part E
cf_laminar = 0.664 * (Re_x_start*0.8+Re_x_end*0.5)^-0.5;
cf_turbulent = 2*u_tau_best^2/u_inf^2;
turb_lam_ratio = cf_turbulent/cf_laminar;
```

```
%% Plot Part C
y_plus = (Y * u_tau_best) / nu;
u_plus = u_avg / u_tau_best;
start_idx = find(y_plus > 30, 1);
end_idx   = find(y_plus > 0.15*Re_tau_best, 1) - 1;

figure;
plot(y_plus, u_plus, 'ko'); hold on;
y_fit = linspace(y_plus(start_idx), y_plus(end_idx), 100);
u_fit = (1/kappa) * log(y_fit) + A;
plot(y_fit, u_fit, 'r-', 'LineWidth', 2);

set(gca, 'XScale','log');
grid on;
xlabel('y^+');
ylabel('u^+');
legend('Data','Log-Law Fit','Location','best');

% clearvars -except u_avg u_tau_best Re_tau_best X Y delta_99 u_inf y_plus
u_plus kappa A;
close all;

%% Calculate Part A
start_idx = find(y_plus > 30, 1);
end_idx   = find(y_plus > Re_tau_best, 1) - 1;
model = @(ws) norm( ...
    u_plus(start_idx:end_idx) ...
    - (1/kappa) * log(y_plus(start_idx:end_idx)) - A ...
    - (2*ws/kappa) .* (sin((pi/2) * (Y(start_idx:end_idx) / delta_99))).^2
...
);
ws_guess = 1.0;
ws = fminsearch(model, ws_guess);

%% Plot Part B

figure;
plot(y_plus, u_plus, 'ko'); hold on;
plot(y_fit, u_fit, 'r-', 'LineWidth', 2);
y_fit_2 = linspace(y_plus(start_idx), y_plus(end_idx), 100);
u_fit_2 = (1/kappa) * log(y_fit_2) + A + 2*ws/kappa*sin(pi/2 *
y_fit_2*nu/u_tau_best/delta_99).^2;
plot(y_fit_2, u_fit_2, 'g-', 'LineWidth', 2);
y_fit_3 = linspace(0,10,100);
u_fit_3 = y_fit_3;
plot(y_fit_3, u_fit_3, 'b-', 'LineWidth', 2);

% Determine the vertical boundaries for the regions
vsl_end    = 5;
buffer_end = 30;
loglaw_end = 0.15 * Re_tau_best;
wake_end   = Re_tau_best;

% Get current y-axis limits for placing the lines and labels
yl = ylim;
```

```
% Draw vertical dashed lines at each boundary
line([vsl_end vsl_end], yl, 'Color', 'k', 'LineStyle', '--');
line([buffer_end buffer_end], yl, 'Color', 'k', 'LineStyle', '--');
line([loglaw_end loglaw_end], yl, 'Color', 'k', 'LineStyle', '--');
line([wake_end wake_end], yl, 'Color', 'k', 'LineStyle', '--');

% Place text labels at the center of each region (vertically centered)
mid_y = mean(yl);
text( vsl_end/2, mid_y, 'VSL', 'HorizontalAlignment', 'center', 'FontSize',
10 );
text( (vsl_end+buffer_end)/2, mid_y, 'Buffer', 'HorizontalAlignment',
'center', 'FontSize', 10 );
text( (buffer_end+loglaw_end)/2-50, mid_y, 'Log-Law', 'HorizontalAlignment',
'center', 'FontSize', 10 );
text( (loglaw_end+wake_end)/2-300, mid_y, 'Wake', 'HorizontalAlignment',
'center', 'FontSize', 10 );
text( (wake_end+max(y_plus))/2+2150, mid_y, 'Freestream',
'HorizontalAlignment', 'center', 'FontSize', 10 );

% Set axis properties and labels
set(gca, 'XScale', 'log');
xlabel('y^+');
ylabel('u^+');
legend('Data','Log-Law Fit','Log-Law and Wake fit','y^+ =
u^+','Location','best');
hold off;

%% Plot Part C
figure;
yl = ylim;
mid_y = mean(yl);
plot(y_plus, u_plus - (1/kappa*log(y_plus)+A))
text( (buffer_end+loglaw_end)/2-50, mid_y, 'Log-Law', 'HorizontalAlignment',
'center', 'FontSize', 10 );
line([buffer_end buffer_end], yl, 'Color', 'k', 'LineStyle', '--');
line([loglaw_end loglaw_end], yl, 'Color', 'k', 'LineStyle', '--');
set(gca, 'XScale', 'log');
xlabel('y^+');
ylabel('\Delta u^+');
```