COMP90072

Template

John Snow , 000 000

Kevin Rassool, 000 001

Semester 1, 3030

Contents

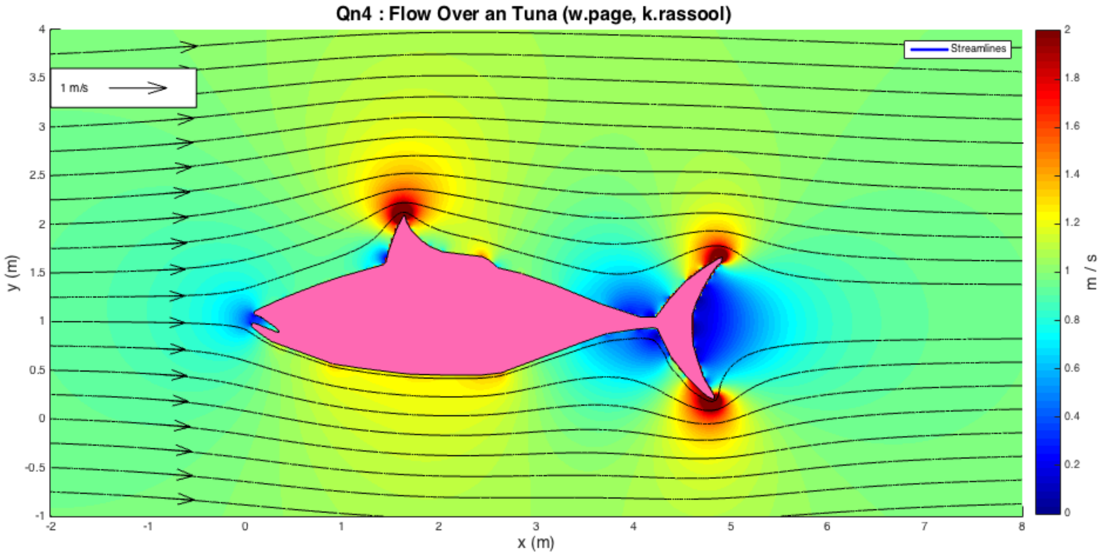

Figure 1: Velocity profile shows that it is, infact, a tuna.

1 Introduction

Hodor. Hodor hodor hodor hodor; hodor HODOR hodor, hodor hodor hodor! Hodor hodor HODOR! Hodor hodor hodor - hodor; hodor hodor hodor! Hodor! Hodor hodor, hodor hodor... Hodor hodor hodor.

2 Door

Hodor. Hodor hodor HODOR! Hodor hodor hodor hodor hodor?! Hodor. Hodor hodor - hodor hodor! Hodor, hodor hodor hodor hodor hodor hodor hodor, hodor, hodor hodor.

1. Hodor. Hodor (see appendix A.1)
2. Hodor. Hodor
3. Hodor. Hodor
4. ***Hodor. Hodor***

```
% make an tuna
function tuna = its_not_a_tuna % Makes a tuna

[NUM,TXT,RAW]=xlsread('tunapoints_excel.xls'); % Read data points, exported from ...
    solidworks
tuna = zeros(length(NUM),4); % Pre-allocate array

for j=1:length(NUM) % For every co-ordinate, add the
    if j==length(NUM) ; tuna(j,:) = [NUM(j,:),NUM(1,:)  ]; % Last endpoint == 1st ...
        startpoint
    else                 tuna(j,:) = [NUM(j,:),NUM(j+1,:)]; % Add an endpoint
    end ; end % Close if and for loops
tuna = tuna/100; % Dimensionalise to match the velocity evaluation meshgrid
```

2.1 Hold the Door.

Hodor. Hodor hodor HODOR! Hodor hodor hodor hodor hodor?! Hodor. Hodor hodor - hodor hodor! Hodor, hodor hodor hodor hodor hodor hodor hodor, hodor, hodor hodor.

2.2 This is including vector graphics

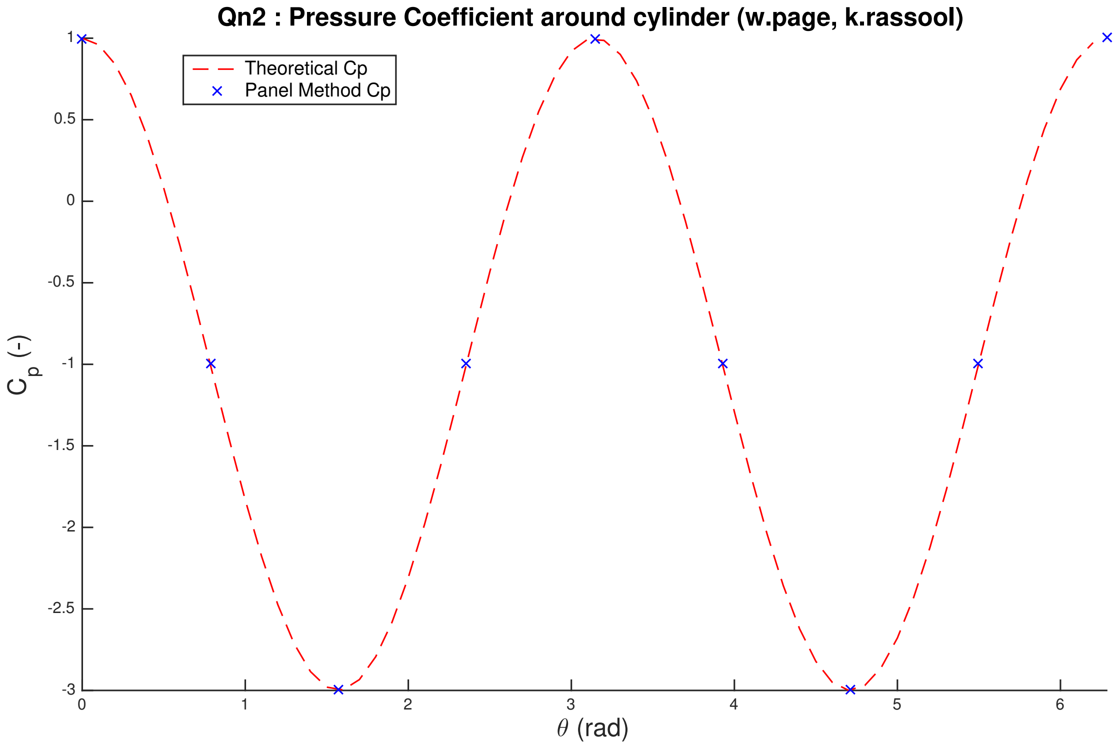

Figure 2: Pressure co-efficient around approximated cylinder

3 Spoilers

Hodor, HODOR hodor, hodor hodor? Hodor, hodor, hodor. Hodor hodor, hodor. Hodor hodor - hodor hodor! Hodor hodor... Hodor hodor hodor hodor HODOR hodor, hodor hodor?! Hodor! Hodor hodor, hodor; hodor hodor, hodor, hodor hodor. Hodor! Hodor hodor, hodor... Hodor hodor hodor, hodor. Hodor hodor - hodor. Hodor. Hodor! Hodor hodor, HODOR hodor, hodor hodor - hodor?!

4 Appendices

A Matlab Functions

A.1 Question 1

Main

```matlab
% William Page (587000) - Kevin Rassool (540773) ;
% Semester 2 2015 - University of Melbourne        ; Started:     21/4/17
% MCEN90018 - Advanced Fluid Dynamics              ; Last Edited: 29/4/17
% Assignment 2 : Panel Methods - 'n' Panel Cylinder
%
% Estimates the flow field around an 'n' panel cylinder
%% Clear MATLAB environment, set format
clc , clear , close all %, format bank

%% Create the panels and find the influsence co-efficients
U_inf  = 1 ;
n_pan  = 64 ; % Number of panels to use
panels = n_panel_circle(n_pan) ;  % Define the number of approximation panels
I=(zeros(n_pan,n_pan)) ; Phi_i=zeros(n_pan,1) ; % Initialise influence

% Calculate influence
for m=1:n_pan; % Loop throught each panel
    Xi=[panels(m,1),panels(m,3)]; % end?points of panel j in x and y
    Yi=[panels(m,2),panels(m,4)];

    Phi_i(m)=atan2((Yi(2) -Yi(1)),(Xi(2) - Xi(1))); % phi_i (eqn 24)

    for k=1:n_pan ; % Calculate the influence coeff on every other panel
        Xj=[panels(k,1),panels(k,3)]; % Midpoints of panel i in x and y
        Yj=[panels(k,2),panels(k,4)];

        I(m,k)=panel_source_strength_1_0(Xi, Yi, Xj, Yj); % Find coeff
    end
end
I(eye(size(I))~=0) = 0.5;  % Where i==j hard code 0.5 strength (using logicals)

V_inf_i = -U_inf*sin(2*pi-Phi_i) % find V_inf, flowing from left to right
q       = I\V_inf_i                      % Solve for source strength densities (q)

%% Find veloctities
tic ; mesh_res = 0.02 ; % Meshgrid density (resolution for results)
[xp, yp] = meshgrid( -3:mesh_res:3 , -2.5:mesh_res:2.5 );
[u_hat,v_hat] = deal(zeros(size(xp))) ; % Initialise cartesian velocity directions

% This next loop runs through each of the panels and sums the velocity
% contribution at each point in space as a result of the panels.

for n=1:n_pan ; % For each point in space, calculate the induced velocity from panels
    Xj=[panels(n,1),panels(n,3)];
    Yj=[panels(n,2),panels(n,4)];

    [u,v] = flow_field_cyl_1_0( Xj , Yj , q(n) , xp , yp );

    u_hat=u_hat + u;
    v_hat=v_hat + v;
end
u_hat_inf = u_hat + U_inf;
time_pattern = toc

```

```
%% Solve the streamlines

% Set up simulation conditions
t0   = 0    ; % Initial time
tf   = 10   ; % Final time
h    = 0.01 ; % Step size

y_range = (-2:.25:2).'; % Range over which to seen line for flow definition
ic0  = [ -3*ones(length(y_range),1) , y_range ]; % % Initial condition matrix
xs = ic0(:,1) ; ys = ic0(:,2) ; % Initial conditions in solver format

% Calculate streamlines in same fashion as fluids
tic ; [xr, yr] = approx_streamline2(xs, ys, tf-t0, h, @flow_general , q , panels, ...
    U_inf);
time_streams = toc

%% Plot results and make pretty, re-run-able after all solutions found
close all ;

% Find Endpoints of panels in x and y
Xi = [panels(:,1),panels(:,3)] ;  Yi = [panels(:,2),panels(:,4)] ;

% Plot approximated cylinder with velocity field
figure ; hold on ; plot(Xi, Yi, 'b-', 'LineWidth', 2.5) ; % Plot cylinder
pcolor(xp, yp, real(sqrt(u_hat_inf.^2+v_hat.^2))) ; shading interp ; colormap jet
fill(panels(:,1),panels(:,2),[255 105 180]./256) ; % HOT PINK cylinder

% Create stream-lines
plot(xr.', yr.', 'k') ;
% Plot streamline direction and magnitude
quivers(xr(:,100), yr(:,100), (xr(:,101)-xr(:,100))./h, (yr(:,101)-yr(:,100))./h , ...
    0.5 , 2 , 'm/s' , 'k')

% Label plot and add features accordingly
axis equal; units = colorbar; xlabel('x (m)'); xlabel(units,'m / s');
axis([-3 3 -2.5 2.5]); ylabel('y (m)'); caxis([0 2]); legend('Streamlines');
title('Qn1 : Flow over and 8 Panel Cylinder (w.page, k.rassool) ') ;
```

N Panel Cylinder Generator

```
% William Page (587000) - Kevin Rassool (xxxxxx) ;
% Semester 2 2015 - University of Melbourne        ; Started:     21/4/17
% MCEN90018 - Advanced Fluid Dynamics              ; Last Edited: 29/4/17
% Assignment 2 : Panel Methods - 'n' Panel Cylinder
%
% Creates an equivilant circle using 'n' panels

function all_panels = n_panel_circle(n)

t = linspace(0,2*pi,n+1);
x = cos(t+pi/n);
y = sin(t+pi/n);
% x = cos(t+pi);
% y = sin(t+pi);

panels= [x.',y.']
new_order = circshift(flip(1:n),floor(n/2)+1,2);
new_panels = panels(new_order,:);
all_panels = zeros(n,4);

% Create end-points
for i=1:n
    if i==n % at the end, replace with the first ones
        all_panels(i,:) = [new_panels(i,:),new_panels(1,:)];
```

```
    else
        all_panels(i,:) = [new_panels(i,:),new_panels(i+1,:)];
    end
end
%
% figure ; hold on ; axis([-1 1 -1 1])
% for i=1:n
%    Xj = [all_panels(i,1),all_panels(i,3)]
%    Yj = [all_panels(i,2),all_panels(i,4)]
%    plot(Xj,Yj,'-b','LineWidth', 2.5)
%    axis([-1 1 -1 1])
%    pause
% end

end
```