

Aerodynamics of Wings and Bodies

Homework 1

Lior Isakov (200928372)

Eden Shazar (204378608)

Fresh start

```
clc
close all
clear variables
```

Parameters

```
% user input
airfoil.xx = 12;      % last two NACA airfoil digits
xs_offset = 1e-3;    % forward offset of start of doublet line from origin
xf_offset = 1e-3;    % backward offset of end of doublet line from cord end
airfoil.N = 100;     % number of doublet line sections
U_oo = 100;          % horizontal uniform flow speed

% calculated parameters
airfoil.t = airfoil.xx/100; % max thickness as fraction of cord
doublet.xs = xs_offset;    % start of doublet line
doublet.xf = 1 - xf_offset; % end of doublet line
```

Wing

```
% doublet line definition
doublet.c_star = doublet.xf - doublet.xs; % doublet line length
theta_j = linspace(0, pi, airfoil.N+1); % cosine sectioning
doublet.tj = doublet.c_star/2 * (1 - cos(theta_j)) + doublet.xs;
doublet.xi = ((doublet.tj(1:end-1) + ...
               doublet.tj(2:end))/2).'; % middle points of sections

% NACA 00xx geometry
airfoil.Y = @(x) 5 * airfoil.t * (0.2969 * sqrt(x) - ...
                                0.1260 * x - ...
                                0.3516 * x.^2 + ...
                                0.2843 * x.^3 - ...
                                0.1015 * x.^4);
```

Solver

```

                                % solve A*mj = b
A = 1./airfoil.Y(doublet.xi) .* ...
    (atan((doublet.tj(2:end) - doublet.xi)./airfoil.Y(doublet.xi)) - ...
    atan((doublet.tj(1:end-1) - doublet.xi)./airfoil.Y(doublet.xi)));
b = ones(airfoil.N, 1);
doublet.mj = (A\b).';

```

Aerodynamic equations

```

% flow components
u_eq = @(x, y) U_oo * (1 + ...
    sum(doublet.mj.*((doublet.tj(2:end) - x) ./ ...
    (((doublet.tj(2:end) - x).^2 + y^2) - ...
    (doublet.tj(1:end-1) - x) ./ ...
    (((doublet.tj(1:end-1) - x).^2 + y^2))));

v_eq = @(x, y) - U_oo * ...
    (sum(doublet.mj.*(y./((doublet.tj(2:end) - x).^2 + y^2) - ...
    y./((doublet.tj(1:end-1) - x).^2 + y^2))));

% pressure coefficient
Cp_eq = @(u, v) 1 - (u.^2 + v.^2)./U_oo^2;

```

Results

```

% plot parameters
lw_l = 1.2;    % large line width
lw_s = 0.9;    % small line width
fs = 15;      % font size
ms = 7;       % marker size
load('colors.mat')

% comparison with Abbot & von Doenhoff
load('NACA_0012_data.mat', 'table_data')
for i = 1:length(table_data.x)
    u_comparison(i) = u_eq(table_data.x(i), airfoil.Y(table_data.x(i)));
    v_comparison(i) = v_eq(table_data.x(i), airfoil.Y(table_data.x(i)));
end
Cp_comparison = Cp_eq(u_comparison, v_comparison);
V = sqrt(u_comparison.^2 + v_comparison.^2);
V_comparison = V/U_oo;
Cp_error = abs(1 - Cp_comparison - table_data.one_minus_Cp) ./ ...
    table_data.one_minus_Cp;
V_error = abs(V_comparison - table_data.V)./table_data.V;
T = table(table_data.x.', u_comparison.', v_comparison.', V.', Cp_comparison.', ...
    table_data.one_minus_Cp.', (1 - Cp_comparison).', Cp_error.', ...
    table_data.V.', V_comparison.', V_error.', ...
    'VariableNames', {'x', 'u', 'v', 'V', 'Cp', ...
    'one_minus_Cp_Abbot', 'one_minus_Cp_calculation', 'one_minus_Cp_error', ...
    'normalized_V_Abbott', 'normalized_V_calculation', 'normalized_V_error'});
writetable(T, 'Abbot & von Doenhoff comparison.xlsx')

```

```

% plotted results
x_plot = linspace(1e-3, 1, 400);
x_surf = linspace(-0.5, 1.5, 400);
y_surf = linspace(-1, 1, 400);
[X_grid, Y_grid] = meshgrid(x_surf, y_surf);
X_grid = X_grid.';
Y_grid = Y_grid.';
[u, v] = deal(zeros(size(X_grid)));
[u_plus, v_plus] = deal(zeros(size(x_plot)));
for i = 1:length(x_surf)
    for j = 1:length(y_surf)
        if x_surf(i) <= 0 || x_surf(i) >= 1 || ...
            y_surf(j) >= airfoil.Y(x_surf(i)) || y_surf(j) <= - airfoil.Y(x_surf(i))
            u(i,j) = u_eq(x_surf(i), y_surf(j));
            v(i,j) = v_eq(x_surf(i), y_surf(j));
        else
            [u(i,j), v(i,j)] = deal(NaN);
        end
    end
end

end
Cp = Cp_eq(u, v);
for i = 1:length(x_plot)
    u_plus(i) = u_eq(x_plot(i), airfoil.Y(x_plot(i)));
    v_plus(i) = v_eq(x_plot(i), airfoil.Y(x_plot(i)));
end
Cp_plus = Cp_eq(u_plus, v_plus);

% FIGURE 1: airfoil geometry and doublet line
figure
hold on
plots = plot_NACA_00xx(airfoil, doublet, true);
title(sprintf('NACA %02d$ airfoil geometry ', airfoil.xx), ...
    'and defined doublet line', 'FontSize', fs)
xlabel('$\frac{x}{c}$', 'FontSize', fs)
ylabel('$\frac{y}{c}$', 'FontSize', fs)
legend([plots.geometry, plots.doublet], ...
    sprintf('NACA %02d$ geometry', airfoil.xx), ...
    'Doublet line sections', ...
    'Location', 'Northeast');
axis equal
grid on
hold off

% FIGURES 2-3: u flow component
figure
hold on
contour(X_grid, Y_grid, u/U_oo, 300, 'LineWidth', lw_s)
plot_NACA_00xx(airfoil, doublet);
title('Flow component: $\frac{u}{U_{\infty}}$', 'FontSize', fs)
xlabel('$\frac{x}{c}$', 'FontSize', fs)
ylabel('$\frac{y}{c}$', 'FontSize', fs)

```

```

view([0, 0, -1])
colorbar
axis image
hold off

figure
hold on
plot_NACA_00xx(airfoil, doublet);
plot(x_plot, u_plus/U_oo, 'LineWidth', lw_l, 'Color', colors.red)
title('Horizontal flow component along upper surface', 'FontSize', fs)
xlabel('$\frac{x}{c}$', 'FontSize', fs)
ylabel('$\frac{u}{U_{\infty}}$', 'FontSize', fs)
axis image
axis manual
y_limits = ylim;
ylim([y_limits(1), 1.4]);
grid on
hold off

% FIGURES 4-5: v flow component
figure
hold on
contour(X_grid, Y_grid, v/U_oo, 300, 'LineWidth', lw_s)
plot_NACA_00xx(airfoil, doublet);
title('Flow component: $\frac{v}{U_{\infty}}$', 'FontSize', fs)
xlabel('$\frac{x}{c}$', 'FontSize', fs)
ylabel('$\frac{y}{c}$', 'FontSize', fs)
view([0, 0, -1])
colorbar
axis image
hold off

figure
hold on
plot_NACA_00xx(airfoil, doublet);
plot(x_plot, v_plus/U_oo, 'LineWidth', lw_l, 'Color', colors.red)
title('Vertical flow component along upper surface', 'FontSize', fs)
xlabel('$\frac{x}{c}$', 'FontSize', fs)
ylabel('$\frac{v}{U_{\infty}}$', 'FontSize', fs)
axis image
axis manual
y_limits = ylim;
ylim([y_limits(1), 0.8]);
grid on
hold off

% FIGURES 5-6: pressure coefficient
figure
hold on
% surf(X_grid, Y_grid, Cp, 'LineStyle', 'none', 'FaceColor', 'interp')
contour(X_grid, Y_grid, Cp, 300, 'LineWidth', lw_s)
plot_NACA_00xx(airfoil, doublet);
title('Pressure coefficient', 'FontSize', fs)

```

```

xlabel('$\frac{x}{c}$', 'FontSize', fs)
ylabel('$\frac{y}{c}$', 'FontSize', fs)
view([0, 0, -1])
colorbar
axis image
hold off

figure
hold on
plot_NACA_00xx(airfoil, doublet);
plot(x_plot, 1 - Cp_plus, 'LineWidth', lw_1, 'Color', colors.red)
title('Pressure coefficient along upper surface', 'FontSize', fs)
xlabel('$\frac{x}{c}$', 'FontSize', fs)
ylabel('$\frac{u^2+v^2}{U_{\infty}^2}=1-C_p$', 'FontSize', fs)
axis image
axis manual
y_limits = ylim;
ylim([y_limits(1), 1.5]);
grid on
hold off

% FIGURE 7: stream lines
figure
hold on
plot_NACA_00xx(airfoil, doublet);

starty = linspace(min(min(Y_grid)), max(max(Y_grid)), 50);
startx = min(min(X_grid))*ones(size(starty));
streamline(X_grid.', Y_grid.', u.', v.', startx, starty)

title('Stream lines', 'FontSize', fs)
xlabel('$\frac{x}{c}$', 'FontSize', fs)
ylabel('$\frac{y}{c}$', 'FontSize', fs)
xlim([min(min(X_grid)), max(max(X_grid))]);
ylim([min(min(Y_grid)), max(max(Y_grid))]);
axis image
grid on
hold off

% FIGURE 8: error plots
figure
hold on
plot_NACA_00xx(airfoil, doublet);
title('Absolute error compared to Abbot \& Doenhoff', 'FontSize', fs)
xlabel('$\frac{x}{c}$', 'FontSize', fs)
ylabel('Error $[\%]$', 'FontSize', fs)
h(1) = plot(table_data.x, 100*V_error, ...
            'LineWidth', lw_1, 'Color', colors.blue);
h(2) = plot(table_data.x, 100*Cp_error, ...
            'LineWidth', lw_1, 'Color', colors.red);
legend(h, ...
       '$\frac{\sqrt{u^2+v^2}}{U_{\infty}}$', ...
       '$\frac{u^2+v^2}{U_{\infty}^2}=1-C_p$', ...)

```

```

        'Location', 'NorthEast')
axis image
y_limits = ylim;
ylim([y_limits(1), 1.5])
grid on
hold off

```

```

function plots = plot_NACA_00xx(airfoil, doublet, plot_doublet_line)
% plot parameters
lw = 1.2;    % line width
ms = 7;      % marker size
load('colors.mat')

if nargin == 2
    plot_doublet_line = false;
end

geometry_x = linspace(0, 1);
geometry_y = airfoil.Y(geometry_x);

patch([geometry_x, fliplr(geometry_x)], ...
      [geometry_y, -fliplr(geometry_y)], ...
      colors.blue, 'FaceAlpha', 0.2)
if plot_doublet_line
    plots.doublet = plot(doublet.tj, zeros(length(doublet.tj), 1), '- .', ...
                        'LineWidth', lw, 'MarkerSize', ms, 'Color', colors.red);
end
plots.geometry = plot([geometry_x, fliplr(geometry_x)], ...
                     [geometry_y, -fliplr(geometry_y)], ...
                     'LineWidth', lw, 'Color', colors.blue);
end

```