AAE 334: Aerodynamics

HW12: Supersonic, Subsonic, Transonic Linear Theories

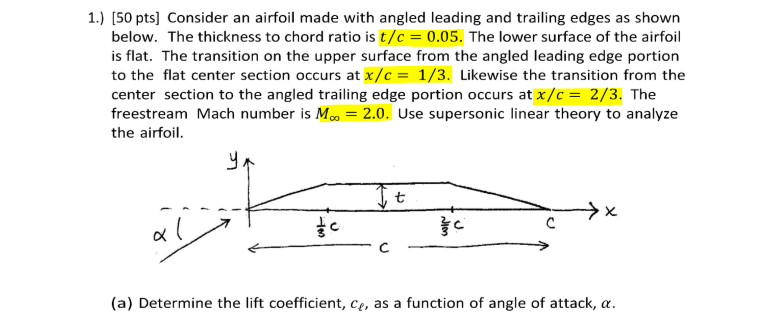
Dr. Blaisdell

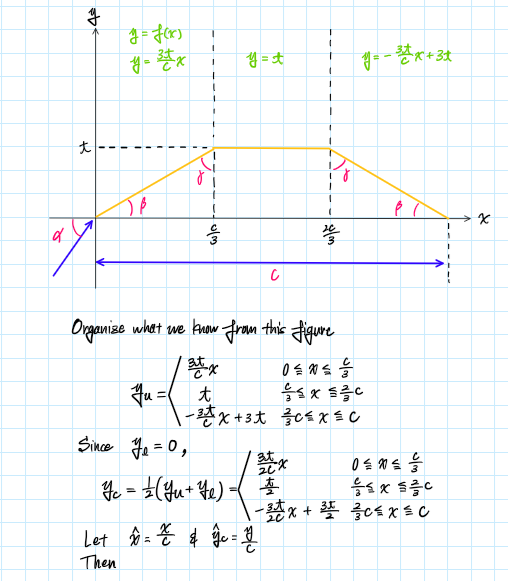
School of Aeronautical and Astronautical

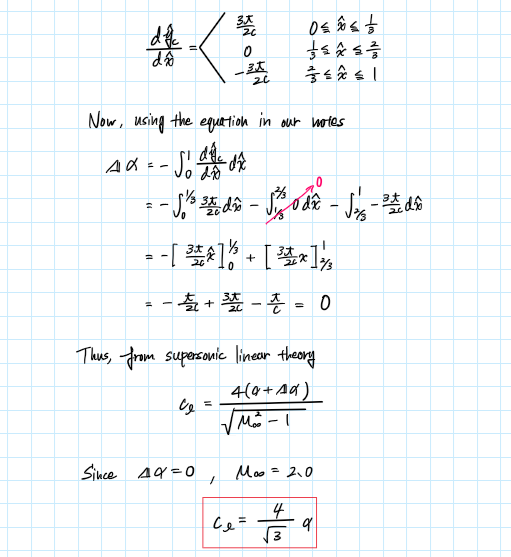
Purdue University

Tomoki Koike

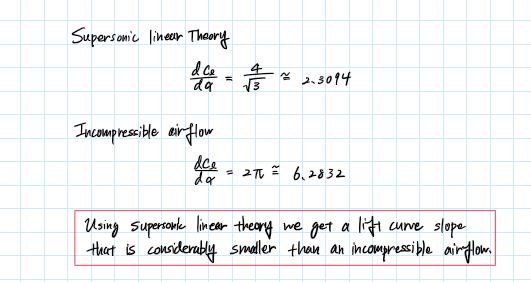
Friday May 1st, 2020



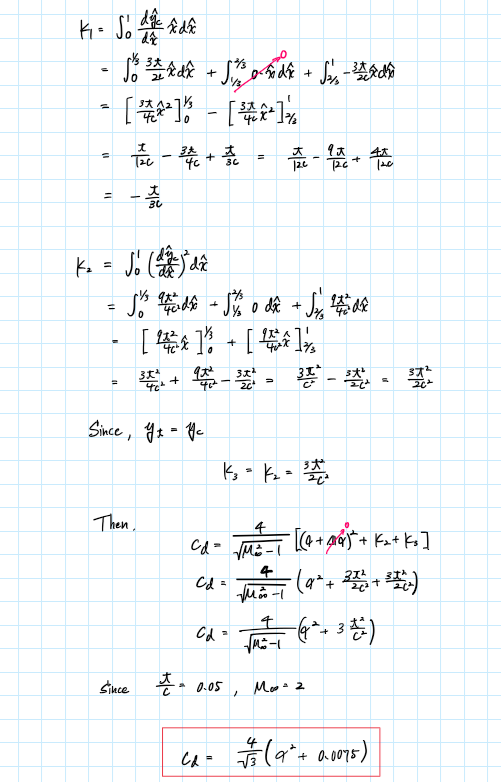




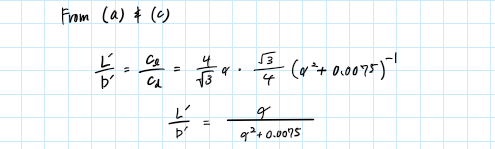










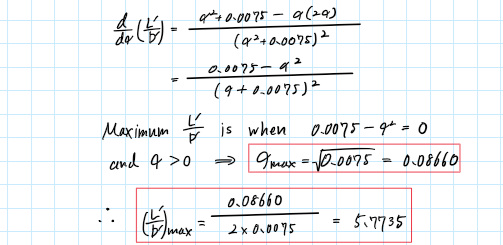


Plotting (using MATLAB)

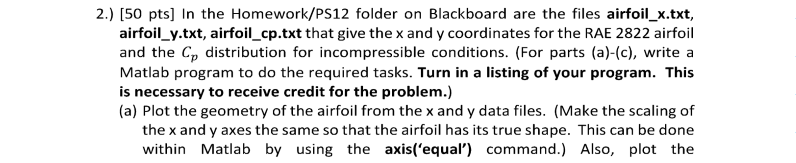
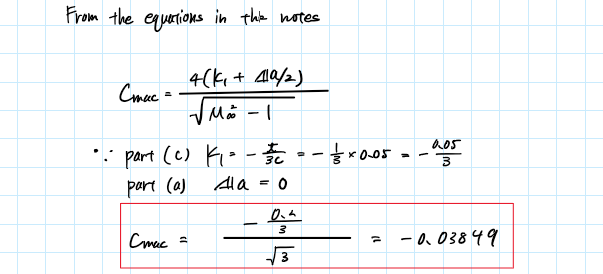
A close up of a map

Description automatically generated

Maximum value









Plots

A close up of text on a white background

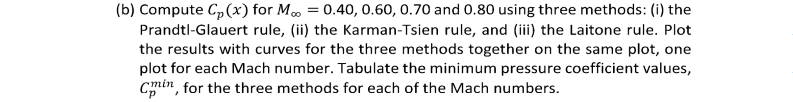
Description automatically generated

A close up of a map

Description automatically generated

Minimum

Using MATLAB, we can find the minimum incompressible pressure coefficient.



Plots

A close up of a map

Description automatically generatedA close up of a map

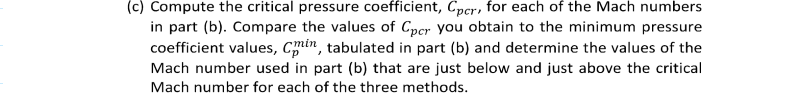
Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Minimum Pressure Coefficients

|  |  |  |  |
| --- | --- | --- | --- |
| Mach Number | Prandtl-Glauert | Karman-Tsien | Laitone |
| 0.40 | -0.709980307 | -0.731664128 | -0.758489723 |
| 0.60 | -0.813384625 | -0.88540185 | -1.011909502 |
| 0.70 | -0.911173033 | -1.047605489 | -1.387353085 |
| 0.80 | -1.084512833 | -1.38490153 | -3.120387693 |



|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| M = 0.40 | M = 0.60 | M = 0.70 | M = 0.80 |
| -3.66201724484731 | -1.29434359045528 | -0.779065964559632 | -0.434640479155229 |

Analysis

* The critical pressure coefficient becomes larger as the Mach number increase (the magnitude decreases).
* At M = 0.40, the critical pressure coefficient is smaller than all 3 of the minimum pressure coefficients computed with the 3 rules
* At M = 0.60, the critical pressure coefficient is smaller than all 3 methods
* At, M = 0.70, the critical pressure coefficient is now larger than all 3 methods
* At, M = 0.80, the critical pressure coefficient is larger than all 3 methods like at M = 0.60

Critical Mach Numbers

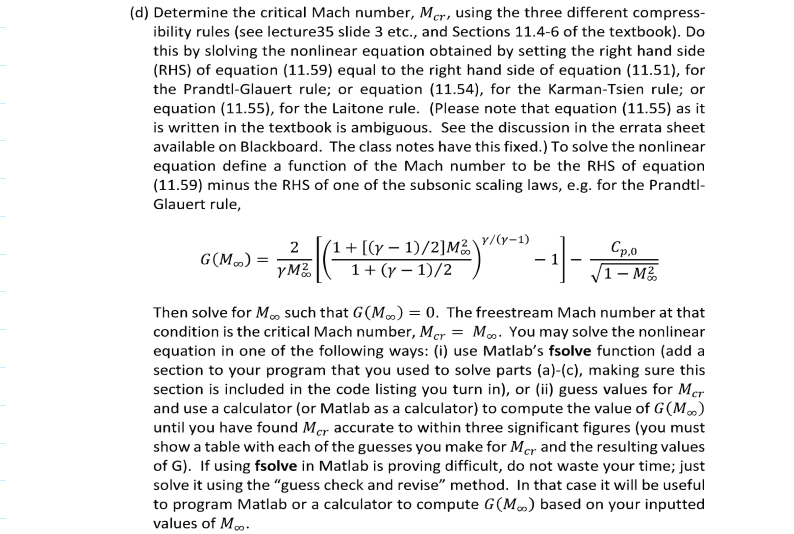
The following is the tabulated data of the absolute difference between the minimum pressure coefficient for all 3 rules at each Mach number and the critical pressure coefficient for each Mach number. For example,

Prandtl-Glauert:

|  |  |  |  |
| --- | --- | --- | --- |
| Mach Number | Prandtl-Glauert | Karman-Tsien | Laitone |
| 0.40 | 2.952036938 | 2.930353117 | 2.903527522 |
| 0.60 | 0.480958965 | 0.40894174 | 0.282434088 |
| 0.70 | 0.132107068 | 0.268539524 | 0.608287121 |
| 0.80 | 0.649872354 | 0.950261051 | 2.685747214 |

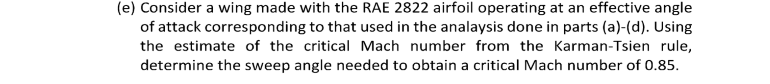
The light-blue highlight represents the smallest absolute difference, meaning that the minimum pressure coefficient is the closest to the critical pressure coefficient for the corresponding Mach number. Thus, the rough prediction of the critical Mach number for the 3 different methods can be deduced as the following.

|  |  |  |
| --- | --- | --- |
| Rough Prediction of the Critical Mach Number | | |
| Prandtl-Glauert | Karman-Tsien | Laitone |
| 0.70 | 0.70 | 0.60 |



Using the command patternsearch() which finds the global minimum of the absolute difference of the equations of Prandtl-Glauert, Karman-Tsien, or Laitone and the critical pressure coefficient equation. The following are the tabulated results.

|  |  |  |
| --- | --- | --- |
| Actual Critical Mach Number | | |
| Prandtl-Glauert | Karman-Tsien | Laitone |
| 0.676035594940186 | 0.658168983459473 | 0.632974100112915 |



To find the sweep angle we use the following equation.

Where = Mach number normal to the wing, = Freestream Mach number

Since, we want the swept wing to resist a freestream Mach number of 0.85 while the Mach number of the airflow normal to the wing is at the critical Mach number for the case of the Karman-Tsien rule which we computed in the previous problem.

Thus, if

and

Using MATLAB, we can compute the sweep angle to be

39.257o

Appendix

MATLAB CODE

## **AAE334 HW12 P1**

clear all; close all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE334\matlab\outputs\HW12';

set(groot, 'defaulttextinterpreter',"latex");

set(groot, 'defaultAxesTickLabelInterpreter',"latex");

set(groot, 'defaultLegendInterpreter',"latex");

% Plotting

a = deg2rad(-10:0.0001:25);

Lp\_Dp = a./(a.^2 + 0.0075);

fig = figure("Renderer","painters");

plot(a,Lp\_Dp);

title({'$\frac{L^{\prime}}{D^{\prime}}$ Over Angle of Attack $\alpha$', ['' ...

'with Supersonic Linear Theory, Koike']});

xlabel('Angle of Attack, $\alpha$ [deg]')

ylabel('$\frac{L^{\prime}}{D^{\prime}}$')

grid on; grid minor; box on;

saveas(fig,fullfile(fdir,'lift2dragRatio.png'));

% Maximum value

Lp\_Dp\_max = max(Lp\_Dp)

## **AAE334 HW12 P2**

clear all; close all; clc;

fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE334\matlab\outputs\HW12';

set(groot, 'defaulttextinterpreter',"latex");

set(groot, 'defaultAxesTickLabelInterpreter',"latex");

set(groot, 'defaultLegendInterpreter',"latex");

% Global constants

gamma = 1.4;

### a)

% Import data

xdata = read\_txt\_file("inputs\hw12\airfoil\_x.txt");

ydata = read\_txt\_file("inputs\hw12\airfoil\_y.txt");

Cpdata = read\_txt\_file("inputs\hw12\airfoil\_cp.txt");

% Plot the airfoil geometry

fig = figure("Renderer","painters");

plot(xdata,ydata,'-b');

title({'Airfoil Geometry, Koike'});

xlabel('x direction');

ylabel('y direction');

axis equal; grid on; grid minor; box on;

saveas(fig,fullfile(fdir,"airfoil.png"));

% Plot the incompressible pressure coefficient distribution

fig = figure("Renderer","painters");

plot(xdata,Cpdata,'-g');

title("Incompressible Pressure Coefficient Distribution, Koike");

xlabel('x direction')

ylabel("Pressure Coefficient, $C\_{po}$")

grid on; grid minor; box on; xlim([-0.02,1.02]);

set(gca,'ydir','reverse');

saveas(fig,fullfile(fdir,"pressure\_coeff.png"));

% Find the minimum Cpo

Cpo\_min = min(Cpdata);

### b)

% Computing the correct pressure coefficients based on subsonic condtions using

% Prandtl-Glauert, Karman-Tsien, and Latione formulas

% Mach numbers to investigate

M\_b = [0.40 0.60 0.70 0.80];

% Preallocating an array to store all results

Cp\_PG = zeros([numel(Cpdata),numel(M\_b)]);

Cp\_KT = zeros([numel(Cpdata),numel(M\_b)]);

Cp\_La = zeros([numel(Cpdata),numel(M\_b)]);

% Loop to conduct calculations

for i = 1:length(M\_b)

M\_i = M\_b(i);

Cp\_PG(:,i) = Prandtl\_Glauert(Cpdata,M\_i);

Cp\_KT(:,i) = Karman\_Tsien(Cpdata,M\_i);

Cp\_La(:,i) = Laitone(Cpdata,M\_i,gamma);

% Plotting

fig = figure(i);

plot(xdata,Cp\_PG(:,i))

txt1 = "Pressure Coefficient with 3 Rules of Improved Compressiblity";

txt2 = sprintf("Corrections When M = %.3f, Koike",M\_i);

title({txt1, txt2})

xlabel('x direction')

ylabel('pressure coefficient, $C\_{p}$')

hold on

plot(xdata,Cp\_KT(:,i))

plot(xdata,Cp\_La(:,i))

hold off; grid on; grid minor; box on; xlim([-0.02,1.02]);

legend('Prandtl-Glauert','Karman-Tsien',"Laitone","Location",'southeast')

set(gca,'ydir','reverse');

file\_txt = sprintf("3rulesPlot\_M%.3f.png",M\_i);

saveas(fig,fullfile(fdir,file\_txt));

end

% Minimum pressure coefficients

min\_Cp\_PG = min(Cp\_PG);

min\_Cp\_KT = min(Cp\_KT);

min\_Cp\_La = min(Cp\_La);

arr = [min\_Cp\_PG.' min\_Cp\_KT.' min\_Cp\_La.'];

T = array2table(arr,"VariableNames",{'Prandtl-Glauert','Karman-Tsien','Laitone'});

writetable(T,fullfile(fdir,'3rules\_min\_Cp.xlsx'),"WriteMode","overwritesheet");

### c)

Cpcr = critical\_Cp(M\_b,gamma);

arr\_p = abs(T{:,:} - Cpcr.');

T\_p = array2table(arr\_p,"VariableNames",{'Prandtl-Glauert','Karman-Tsien','Laitone'});

writetable(T\_p,fullfile(fdir,'3rules\_diff\_Cp.xlsx'),"WriteMode","overwritesheet");

### d)

% Compute the Critical Mach Number using Optimization - PatternSearch

% Prandtl-Glauert

M0 = 0.2;

lb = 0.01; ub = 0.95;

A = []; b = [];

Aeq = []; beq = [];

objfunc = @(M) opt\_Prandtl\_Glauert\_Mcr(M,Cpo\_min,gamma);

[Mcr\_PG, fval\_PG] = patternsearch(objfunc,M0,A,b,Aeq,beq,lb,ub);

% Karman-Tsien

objfunc = @(M) opt\_Karman\_Tsien\_Mcr(M,Cpo\_min,gamma);

[Mcr\_KT, fval\_KT] = patternsearch(objfunc,M0,A,b,Aeq,beq,lb,ub);

% Laitone

objfunc = @(M) opt\_Laitone\_Mcr(M,Cpo\_min,gamma);

[Mcr\_La, fval\_La] = patternsearch(objfunc,M0,A,b,Aeq,beq,lb,ub);

### d)

Mn = 0.85;

sigma = acosd(Mcr\_KT/Mn);

### FUNCTIONS

function data = read\_txt\_file(file\_str)

% This function reads a txt.file and imports the data as an array

afile = fopen(file\_str,'r');

formatSpec = '%f';

data = fscanf(afile,formatSpec);

fclose(afile);

end

function Cp = Prandtl\_Glauert(Cpo,M)

% Function that calculates the subsonic linear theory using

% Prandtl-Glauert formula

Cp = Cpo./sqrt(1 - M.^2);

end

function Cp = Karman\_Tsien(Cpo,M)

% Function that calculates the subsonic linear theory using

% Karman-Tsien formula

a1 = sqrt(1 - M.^2);

a2 = M^2./(1 + sqrt(1 - M.^2));

Cp = Cpo./(a1 + a2\*Cpo/2);

end

function Cp = Laitone(Cpo,M,gamma)

% Function that calculates the subsonic linear theory using Latione

% formula

a1 = sqrt(1 - M.^2);

a2 = (1 + (gamma - 1)/2.\*M.^2);

a3 = M^2.\*a2/2./sqrt(1 - M.^2);

Cp = Cpo./(a1 + a3.\*Cpo);

end

function Cpcr = critical\_Cp(M,gamma)

% This function computes the critical pressure coefficient for a given

% Mach number

a1 = 2/gamma./M.^2;

a2 = 1 + (gamma - 1)/2.\*M.^2;

a3 = (gamma + 1)/2;

a4 = gamma/(gamma - 1);

Cpcr = a1.\*((a2./a3).^a4 - 1);

end

function G\_M = opt\_Prandtl\_Glauert\_Mcr(M,Cpo\_min,gamma)

% Expression of critical pressure coefficient function

A1 = 2/gamma/M^2;

A2 = 1 + (gamma - 1)/2\*M^2;

A3 = (gamma + 1)/2;

A4 = gamma/(gamma - 1);

Cpcr = A1\*((A2/A3)^A4 - 1);

% Prandtl-Glauert

PG = Cpo\_min/sqrt(1 - M^2);

% Difference between the two equations

G\_M = abs(Cpcr - PG);

end

function G\_M = opt\_Karman\_Tsien\_Mcr(M,Cpo\_min,gamma)

% Expression of critical pressure coefficient function

A1 = 2/gamma/M^2;

A2 = 1 + (gamma - 1)/2\*M^2;

A3 = (gamma + 1)/2;

A4 = gamma/(gamma - 1);

Cpcr = A1\*((A2/A3)^A4 - 1);

% Karman-Tsien

a1 = sqrt(1 - M.^2);

a2 = M^2./(1 + sqrt(1 - M.^2));

KT = Cpo\_min./(a1 + a2\*Cpo\_min/2);

% Difference between the two equations

G\_M = abs(Cpcr - KT);

end

function G\_M = opt\_Laitone\_Mcr(M,Cpo\_min,gamma)

% Expression of critical pressure coefficient function

A1 = 2/gamma/M^2;

A2 = 1 + (gamma - 1)/2\*M^2;

A3 = (gamma + 1)/2;

A4 = gamma/(gamma - 1);

Cpcr = A1\*((A2/A3)^A4 - 1);

% Laitone

a1 = sqrt(1 - M.^2);

a2 = (1 + (gamma - 1)/2.\*M.^2);

a3 = M^2.\*a2/2./sqrt(1 - M.^2);

La = Cpo\_min./(a1 + a3.\*Cpo\_min);

% Difference between the two equations

G\_M = abs(Cpcr - La);

end