AAE 334: Aerodynamics

HW11

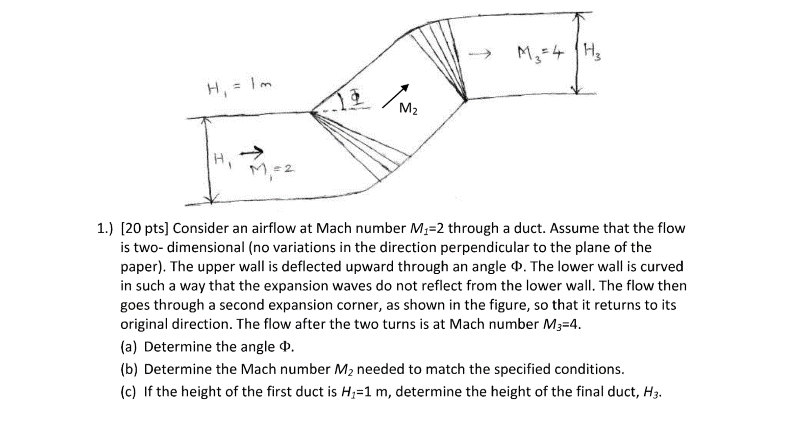
Dr. Blaisdell

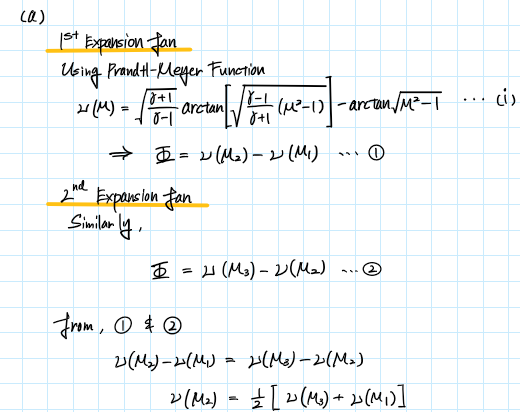
School of Aeronautical and Astronautical

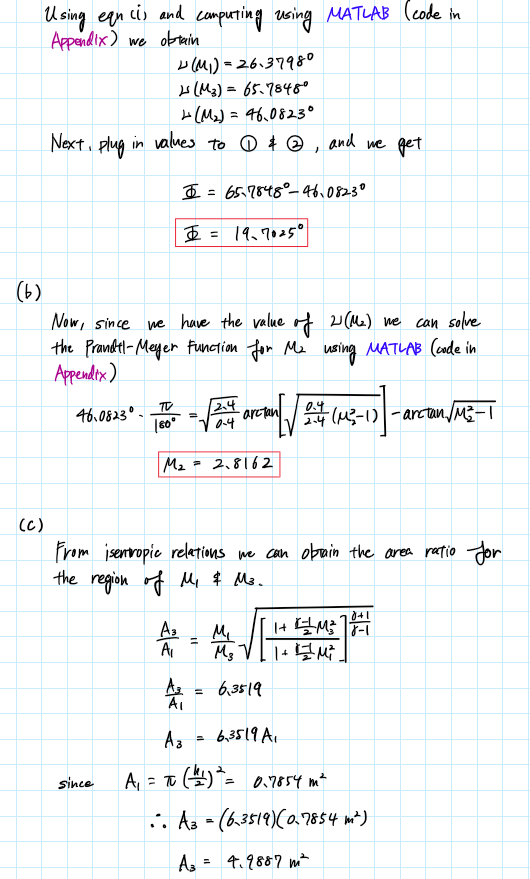
Purdue University

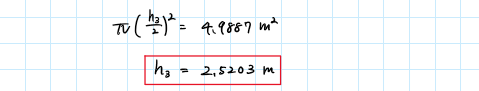
Tomoki Koike

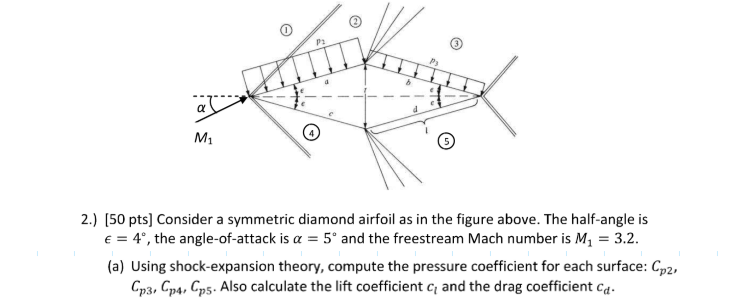
Friday April 24th, 2020

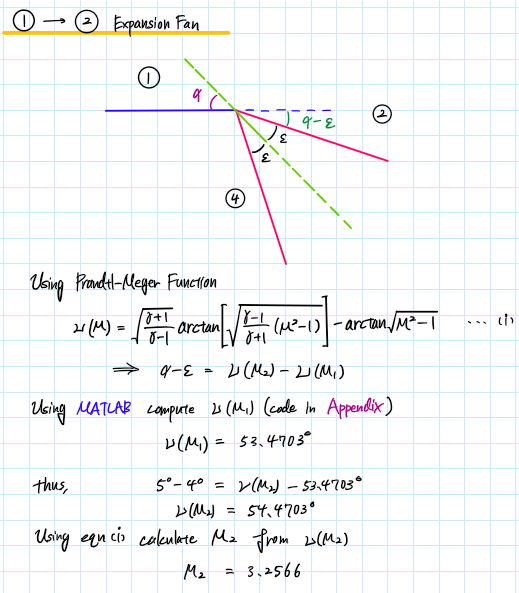


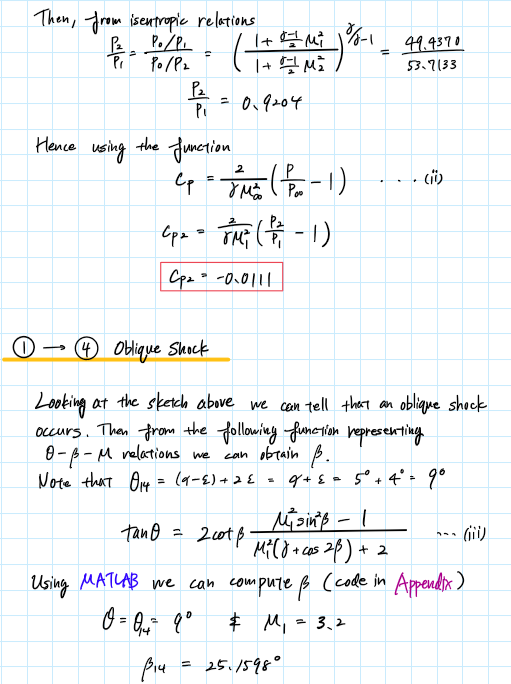


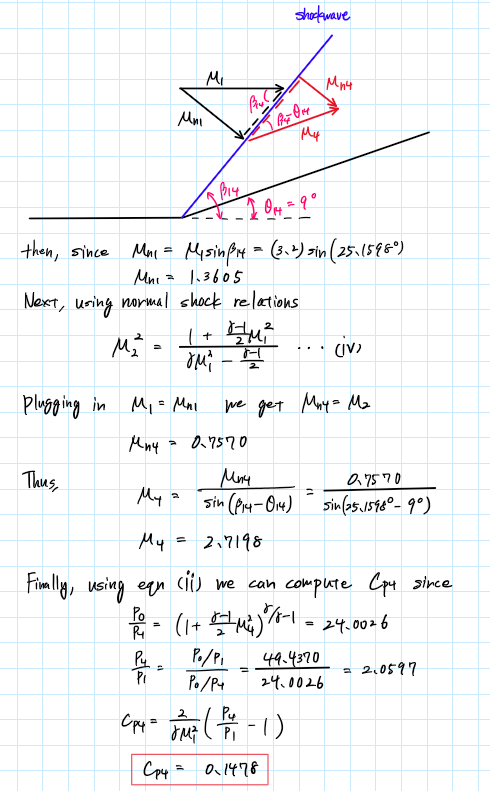


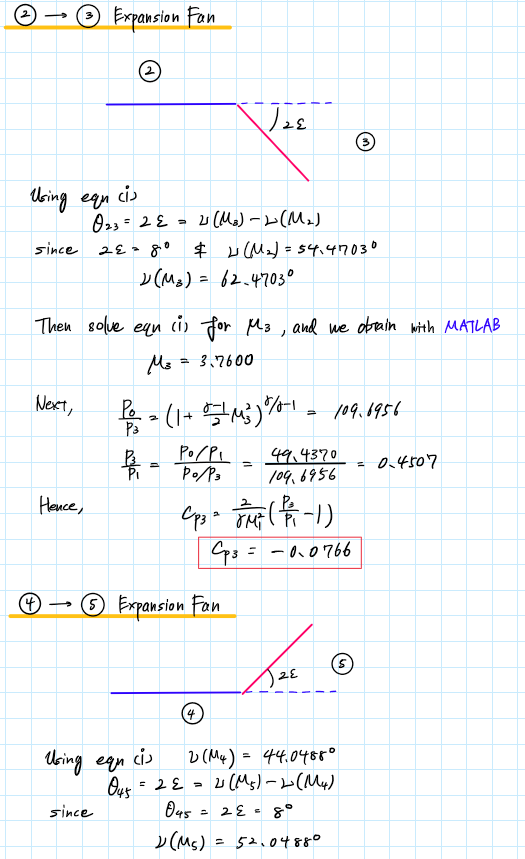


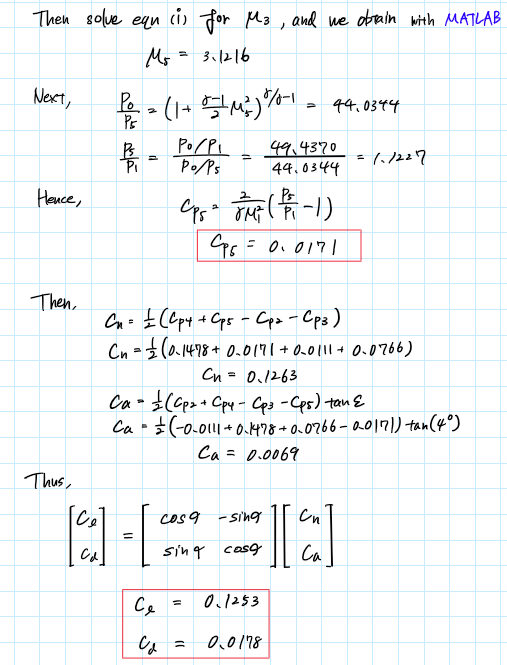


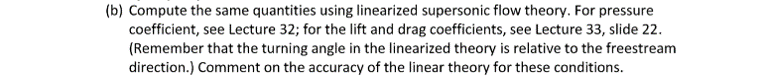












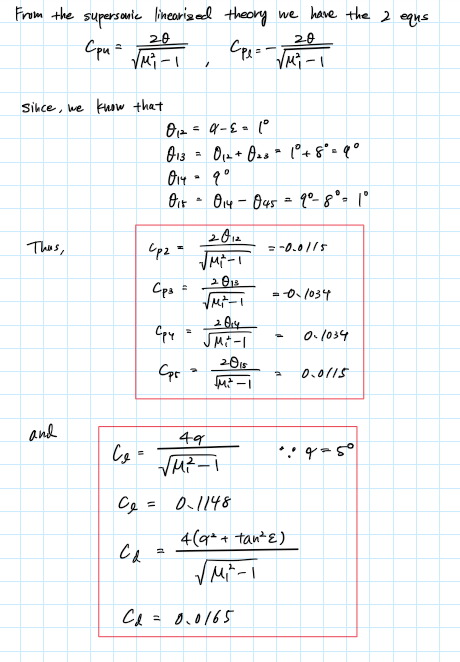
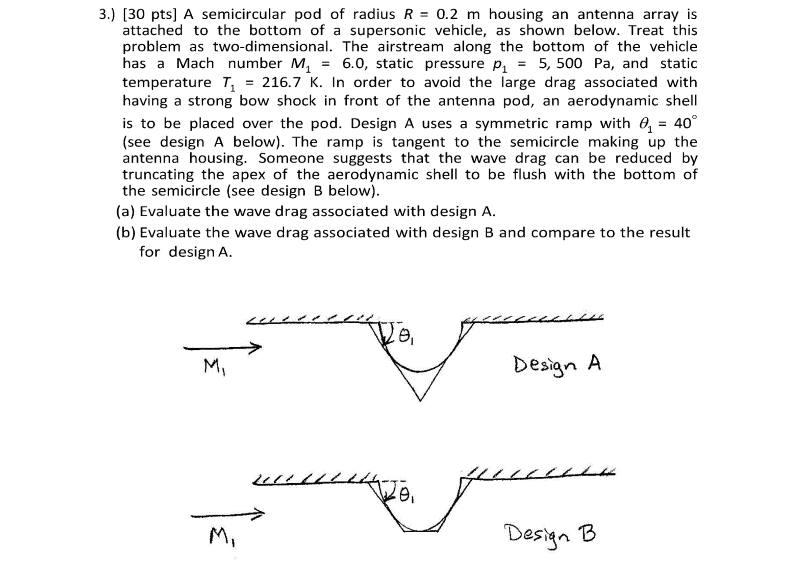


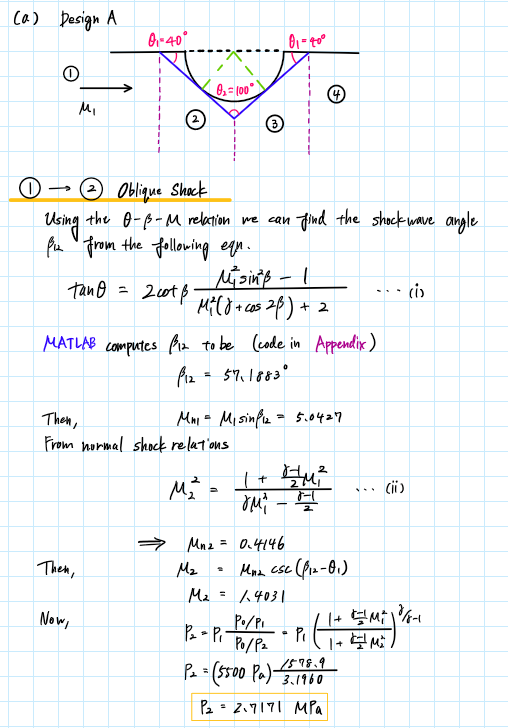
Table of Comparison

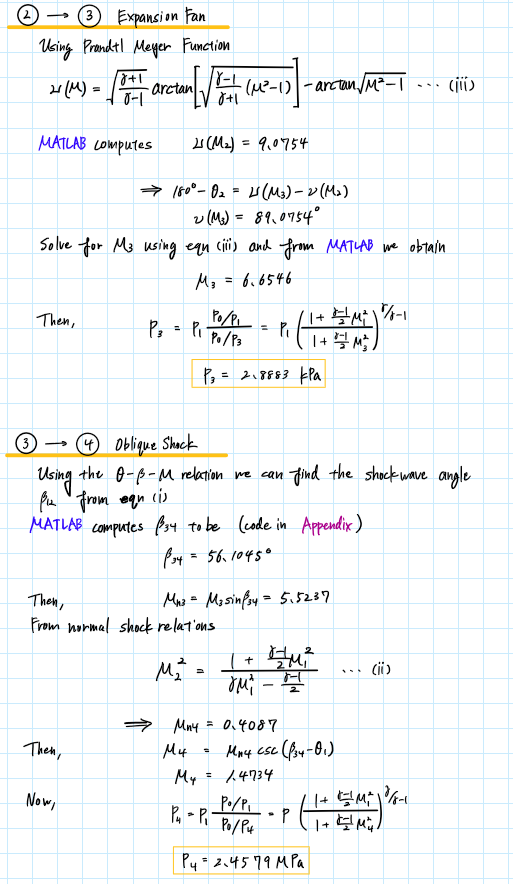
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Angle Relative to Freestream [deg] | Shockwave and Expansion Fan Analysis (I) | Supersonic Linearized Theory (II) | Difference  (I) – (II) |
|  | -1 | -0.0111 | -0.0115 | 0.0004 |
|  | -9 | -0.0766 | -0.1034 | 0.0267 |
|  | 9 | 0.1478 | 0.1034 | 0.0445 |
|  | -1 | 0.0171 | 0.0115 | 0.0056 |
|  | NaN | 0.1253 | 0.1148 | 0.0104 |
|  | NaN | 0.0178 | 0.0165 | 0.0014 |

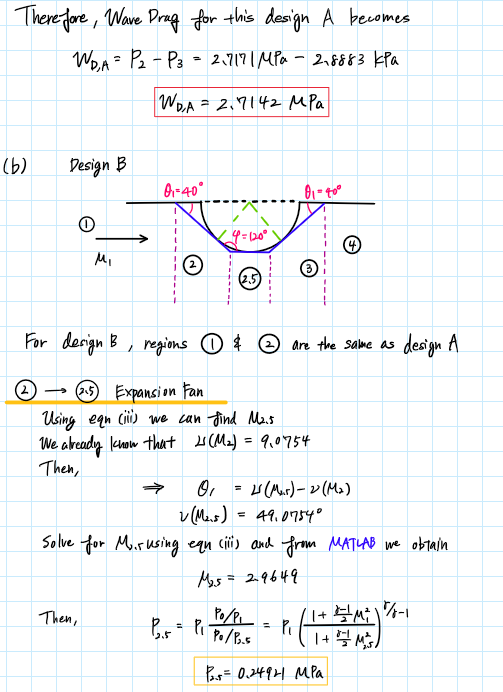
Analysis

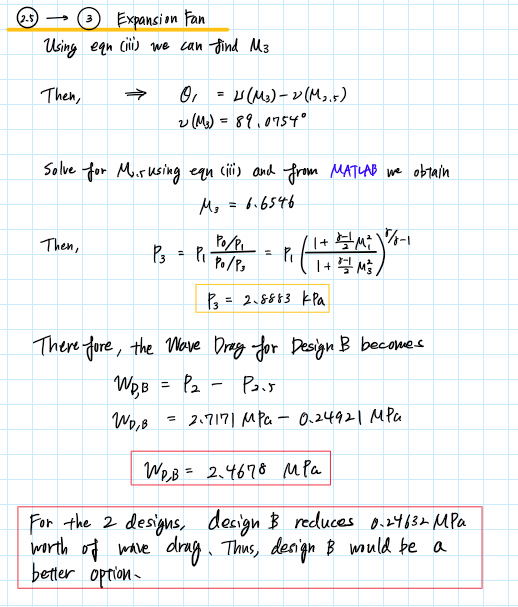
* From the tabulated results we can see that for small angles the error of the pressure coefficients computed using the supersonic linearized theory is small.
* The drag coefficient has a smaller error compared to the lift coefficient. This is probably due to the large deviations of the pressure coefficients with large angles relative to the freestream.











Appendix

## **AAE 334 HW11**

clear all; close all; clc;

% Global constants

gamma = 1.4;

R = 287.05;

### P1 (a)

% Given properties

M1 = 2;

M3 = 4;

h1 = 1; % [m]

% Apply Prantl-Meyer

nu\_M1 = Prandtl\_Meyer\_Expansion(M1,gamma);

nu\_M3 = Prandtl\_Meyer\_Expansion(M3,gamma);

nu\_M2 = 0.5\*(nu\_M1 + nu\_M3);

Phi = nu\_M3 - nu\_M2;

### (b)

% Find M2

M2 = calc\_M\_from\_PrantlMeyer(nu\_M2,gamma);

### (c)

% Obtain the area ratio from isentropic relations

A3\_A1 = areaRatio\_from\_isentropic\_relation(M1,M3,gamma);

A1 = pi\*(h1/2)^2;

A3 = A3\_A1\*A1;

h3 = 2\*sqrt(A3/pi);

### P2 (a)

% Given properties

alpha = 5; % [deg]

epsilon = 4; % [deg]

M1 = 3.2;

gamma = 1.4;

% 1 -> 2

nu\_M1 = Prandtl\_Meyer\_Expansion(M1,gamma);

theta12 = alpha - epsilon;

nu\_M2 = theta12 + nu\_M1;

M2 = calc\_M\_from\_PrantlMeyer(nu\_M2,gamma);

P0\_P1 = isentropic\_relation\_P\_ratio(M1,gamma);

P0\_P2 = isentropic\_relation\_P\_ratio(M2,gamma);

P2\_P1 = P0\_P1/P0\_P2;

Cp2 = calc\_pressure\_coeff(P2\_P1,M1,gamma);

% 1 -> 4

theta14 = alpha + epsilon;

beta14 = theta\_beta\_M\_relation(theta14,M1,gamma);

Mn1 = M1\*sind(beta14);

Mn4 = normalShock\_jump\_M(Mn1,gamma);

M4 = Mn4/sind(beta14-theta14);

P0\_P4 = isentropic\_relation\_P\_ratio(M4,gamma);

P4\_P1 = P0\_P1/P0\_P4;

Cp4 = calc\_pressure\_coeff(P4\_P1,M1,gamma);

% 2 -> 3

theta23 = 2\*epsilon;

nu\_M3 = theta23 + nu\_M2;

M3 = calc\_M\_from\_PrantlMeyer(nu\_M3,gamma);

P0\_P3 = isentropic\_relation\_P\_ratio(M3,gamma);

P3\_P1 = P0\_P1/P0\_P3;

Cp3 = calc\_pressure\_coeff(P3\_P1,M1,gamma);

% 4 -> 5

theta45 = 2\*epsilon;

nu\_M4 = Prandtl\_Meyer\_Expansion(M4,gamma);

nu\_M5 = theta45 + nu\_M4;

M5 = calc\_M\_from\_PrantlMeyer(nu\_M5,gamma);

P0\_P5 = isentropic\_relation\_P\_ratio(M5,gamma);

P5\_P1 = P0\_P1/P0\_P5;

Cp5 = calc\_pressure\_coeff(P5\_P1,M1,gamma);

% Lift and drag coefficients

Cn = 0.5\*(Cp4 + Cp5 - Cp2 - Cp3);

Ca = 0.5\*(Cp2 + Cp4 - Cp3 - Cp5)\*tand(epsilon);

DCM = [cosd(alpha) -sind(alpha);

sind(alpha) cosd(alpha)];

res = DCM\*[Cn; Ca];

Cl = res(1);

Cd = res(2);

### (b)

Cp2\_ssl = supersonic\_linear\_theory\_Cp(-theta12,M1,"upper");

theta13 = theta12 + theta23;

Cp3\_ssl = supersonic\_linear\_theory\_Cp(-theta13,M1,"upper");

Cp4\_ssl = supersonic\_linear\_theory\_Cp(-theta14,M1,"lower");

theta15 = theta14 - theta45;

Cp5\_ssl = supersonic\_linear\_theory\_Cp(-theta15,M1,"lower");

Cl\_ssl = 4\*deg2rad(alpha)/sqrt(M1^2 - 1);

Cd\_ssl = 4\*((deg2rad(alpha))^2 + (tand(epsilon))^2)/sqrt(M1^2 - 1);

% Percent Errors

C1 = [Cp2 Cp3 Cp4 Cp5 Cl Cd];

C2 = [Cp2\_ssl Cp3\_ssl Cp4\_ssl Cp5\_ssl Cl\_ssl Cd\_ssl];

diff = C1 - C2;

### P3 (a)

% Given properties

M1 = 6;

T1 = 216.7; % [K]

P1 = 5500; % [Pa]

theta1 = 40; % [deg]

gamma = 1.4;

theta2 = 180 - 2\*theta1;

% 1 -> 2

beta12 = theta\_beta\_M\_relation(theta1,M1,gamma);

Mn1 = M1\*sind(beta12);

Mn2 = normalShock\_jump\_M(Mn1,gamma);

M2 = Mn2\*cscd(beta12 - theta1);

P0\_P1 = isentropic\_relation\_P\_ratio(M1,gamma);

P0\_P2 = isentropic\_relation\_P\_ratio(M2,gamma);

P2 = P1\*P0\_P1/P0\_P2;

% 2 -> 3

nu\_M2 = Prandtl\_Meyer\_Expansion(M2,gamma);

nu\_M3 = 180 - theta2 + nu\_M2;

M3 = calc\_M\_from\_PrantlMeyer(nu\_M3,gamma);

P0\_P3 = isentropic\_relation\_P\_ratio(M3,gamma);

P3 = P1\*P0\_P1/P0\_P3;

% 3 -> 4

beta34 = theta\_beta\_M\_relation(theta1,M3,gamma);

Mn3 = M3\*sind(beta34);

Mn4 = normalShock\_jump\_M(Mn3,gamma);

M4 = Mn4\*cscd(beta34 - theta1);

P0\_P4 = isentropic\_relation\_P\_ratio(M4,gamma);

P4 = P1\*P0\_P1/P0\_P4;

% Wave drag

WD\_A = P2 - P3;

### (b)

phi = 180 - theta1;

% 2 -> 2.5

nu\_M25 = theta1 + nu\_M2;

M25 = calc\_M\_from\_PrantlMeyer(nu\_M25,gamma);

P0\_P25 = isentropic\_relation\_P\_ratio(M25,gamma);

P25 = P1\*P0\_P1/P0\_P25;

% 2.5 - > 3

nu\_M3 = theta1 + nu\_M25;

M3 = calc\_M\_from\_PrantlMeyer(nu\_M3,gamma);

P0\_P3 = isentropic\_relation\_P\_ratio(M3,gamma);

P3 = P1\*P0\_P1/P0\_P3;

% Wave Drag

WD\_B = P2 - P25;

% Compare

WD\_diff = WD\_A - WD\_B

### Functions

function M = calc\_M\_from\_PrantlMeyer(nu,gamma)

M = sym('M');

assume(M,["real","positive"]);

a1 = sqrt((gamma + 1)/(gamma - 1));

a2 = atand(a1^(-1)\*sqrt(M^2 - 1));

a3 = atand(sqrt(M^2 - 1));

eqn = nu == a1\*a2 - a3;

M = double(vpasolve(eqn,M));

if M < 0

M = -M;

end

end

function A2\_A1 = areaRatio\_from\_isentropic\_relation(M1,M2,gamma)

% Calculate the Mach number at the inlet

a1 = 1 + (gamma - 1)/2\*M2^2;

a2 = 1 + (gamma - 1)/2\*M1^2;

a3 = (gamma + 1)/2/(gamma - 1);

A2\_A1 = M1/M2 \* (a1/a2)^(a3);

end

function P\_rat = isentropic\_relation\_P\_ratio(M,gamma)

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

end

function Cp = calc\_pressure\_coeff(P\_rat,M,gamma)

Cp = 2/gamma/M^2\*(P\_rat - 1);

end

function Cp = supersonic\_linear\_theory\_Cp(theta,M\_inf,type)

theta = deg2rad(theta);

if type == "upper"

Cp = 2\*theta/sqrt(M\_inf^2 - 1);

elseif type == "lower"

Cp = -2\*theta/sqrt(M\_inf^2 - 1);

end

end

function nu = Prandtl\_Meyer\_Expansion(M,gamma)

%{

Function: Prandtl\_Meyer\_Expansion

Author: Tomoki Koike

Description: This function calculates the Prandtl-Meyer function results for

a given flow with given Mach number to find the

expansion fan relations

>>Inputs

M1: Mach number before expansion fan

gamma: specific hear ratio

Outputs<<

nu: Prandtl-Meyer function result [deg]

%}

a1 = sqrt((gamma + 1)/(gamma - 1));

a2 = atand(a1^(-1)\*sqrt(M^2 - 1));

a3 = atand(sqrt(M^2 - 1));

nu = a1\*a2 - a3;

end

function M2 = normalShock\_jump\_M(M1,gamma)

%{

Function: normalShock\_jump\_M

Author: Tomoki Koike

Description: This function calculates the Mach number jump after a normal shockwave.

>>Inputs

M1: Mach number

gamma: specific hear ratio

Outputs<<

M2: Mach number after shock

%}

a1 = (gamma - 1)/2;

M2 = sqrt((1 + a1\*M1^2)/(gamma\*M1^2 - a1));

end