AAE 334: Aerodynamics

HW 10: Shockwave Interactions

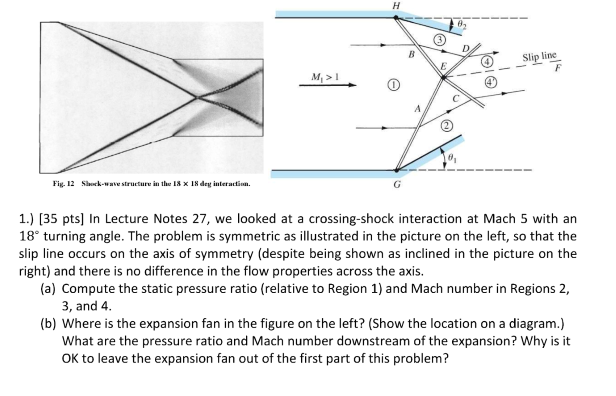
Dr. Blaisdell

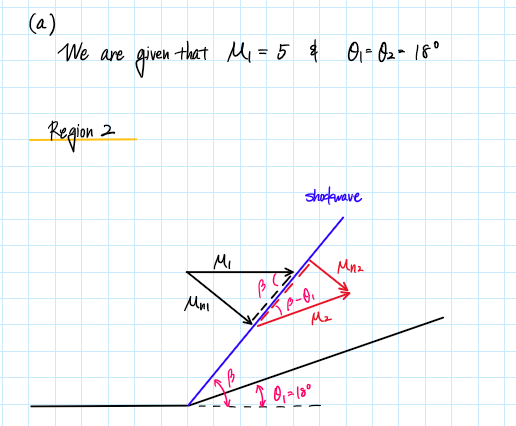
School of Aeronautical and Astronautical

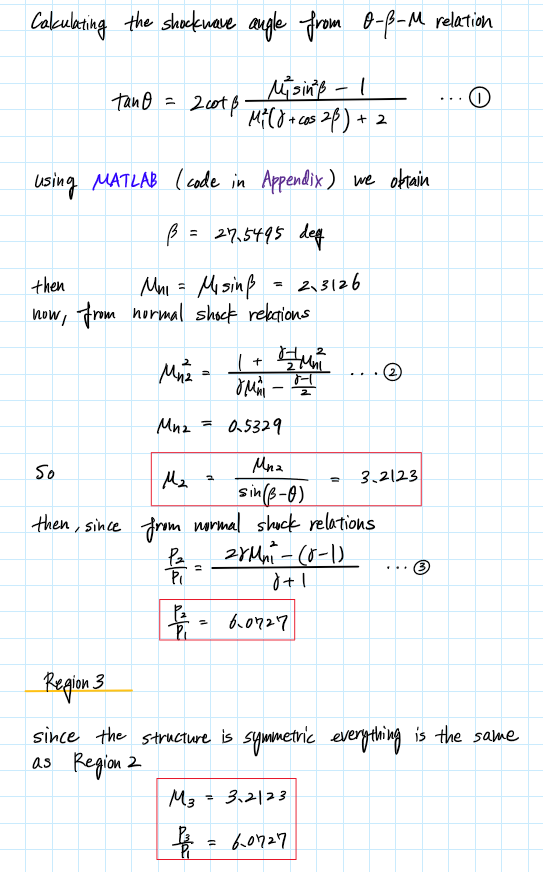
Purdue University

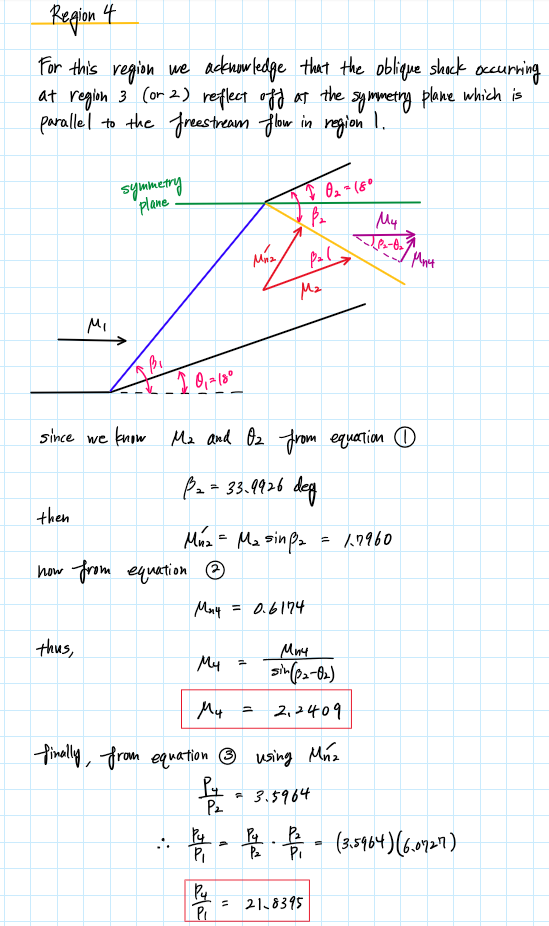
Tomoki Koike

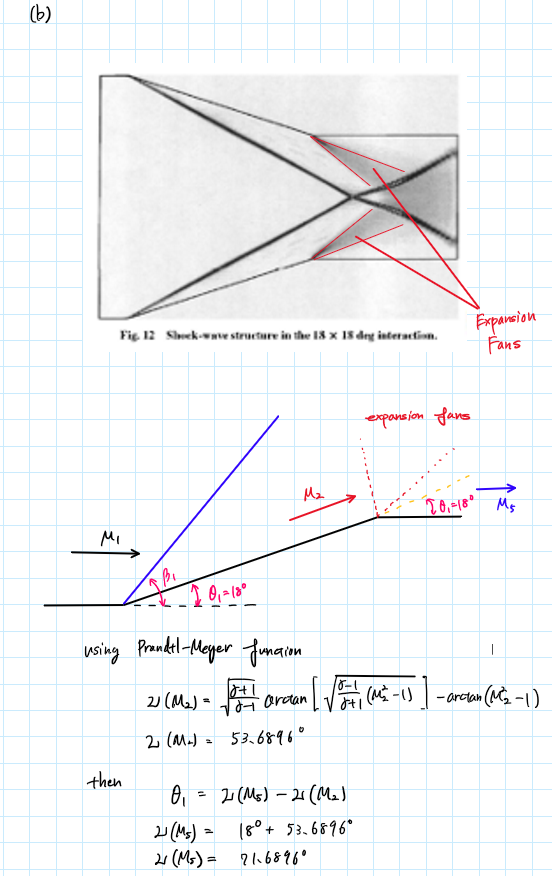
Friday April 17th, 2020

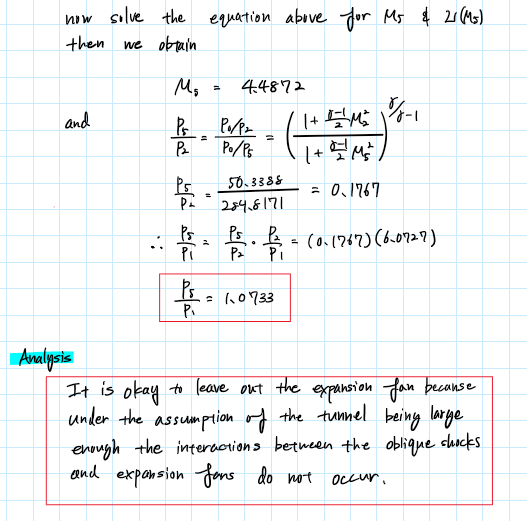


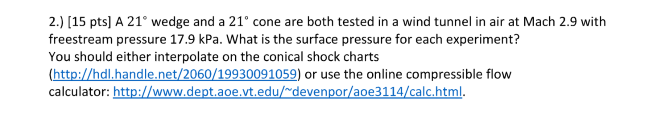


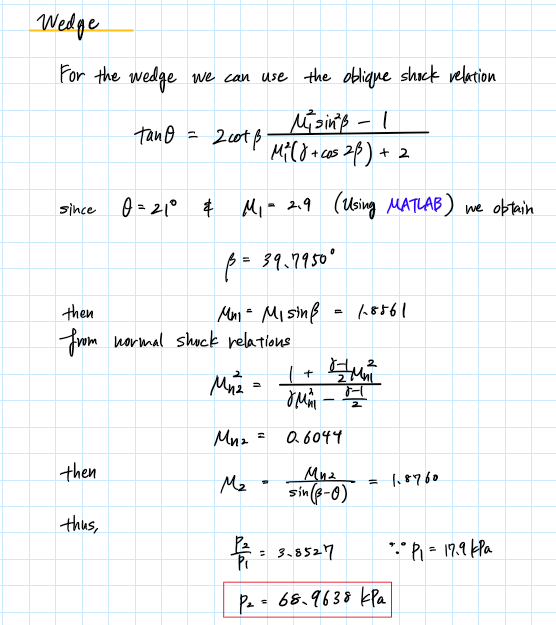


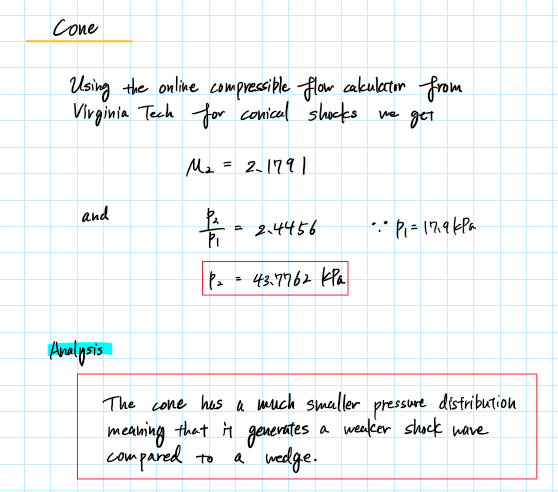


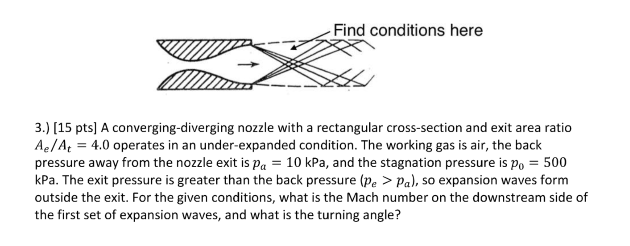


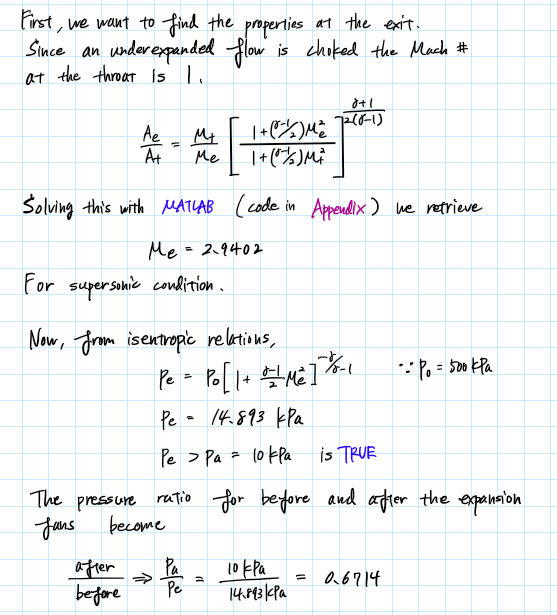


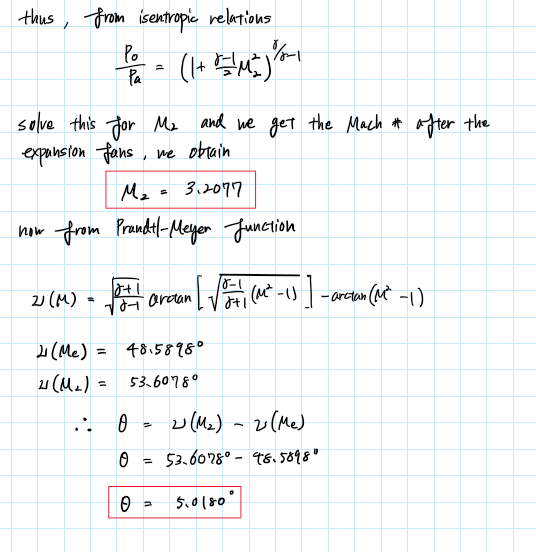


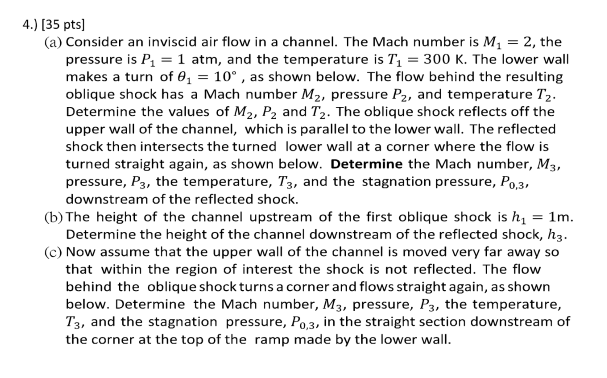


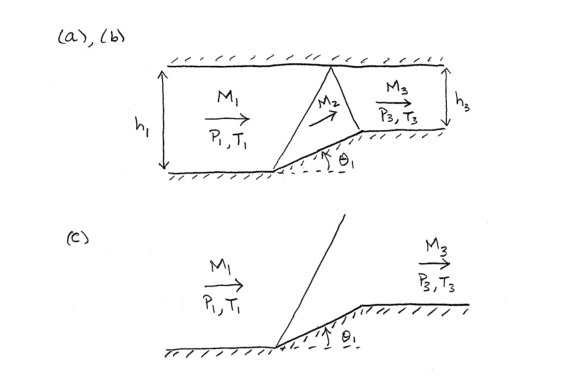


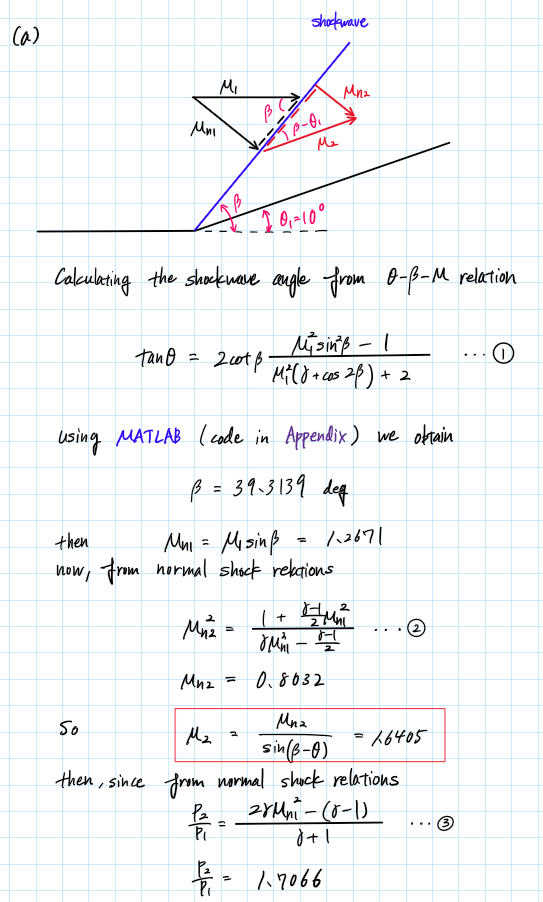


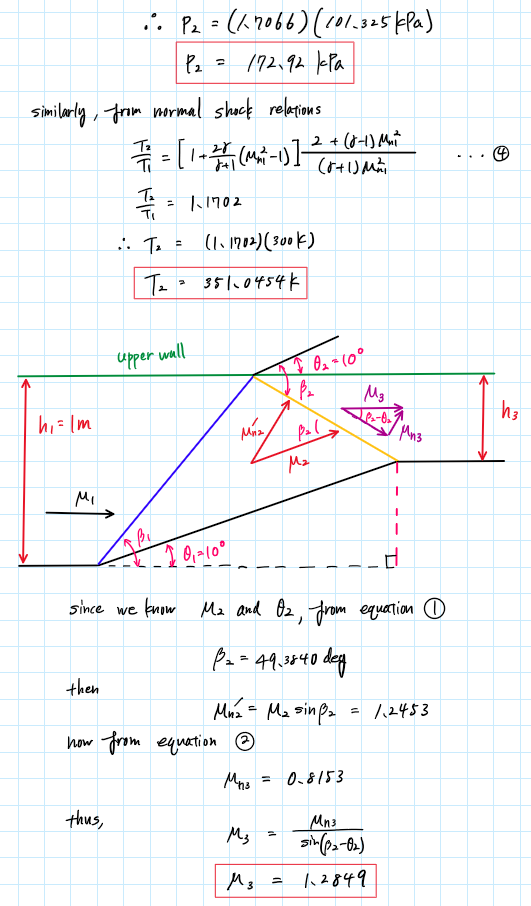


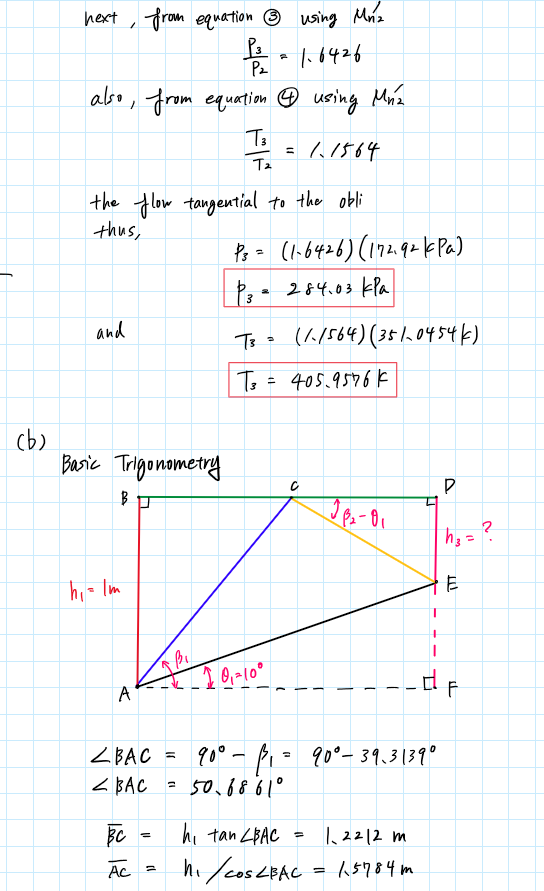


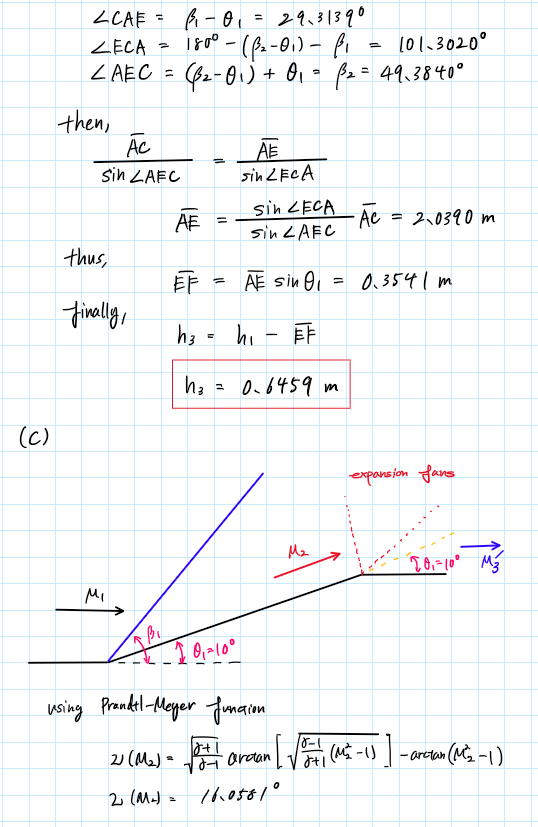


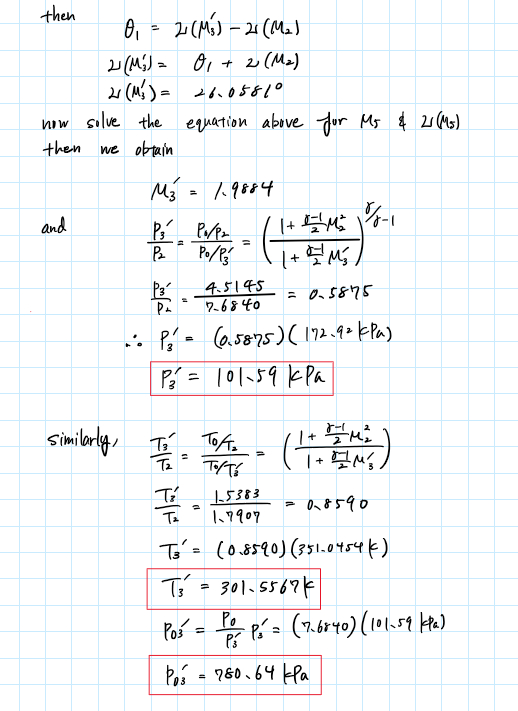












Appendix

## **AAE 334 HW10**

clear all; close all; clc;

### P1

% (a)

% Given Properties

M1 = 5;

theta = 18; % [deg]

gamma = 1.4;

% Region 2

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

Mn1 = M1\*sind(beta);

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

M2 = Mn2/sind(beta-theta);

P2\_P1 = normalShock\_jump\_P\_static(Mn1,gamma)

% Region 4

beta2 = theta\_beta\_M\_relation(theta,M2,gamma);

Mn2\_new = M2\*sind(beta2);

Mn4 = normalShock\_jump\_M(Mn2\_new,gamma);

M4 = Mn4/sind(beta2 - theta);

P4\_P2 = normalShock\_jump\_P\_static(Mn2\_new,gamma);

P4\_P1 = P4\_P2\*P2\_P1;

% (b) Expansion fan

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

nu\_M5 = theta + nu\_M2;

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

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

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

P5\_P2 = P0\_P2/P0\_P5;

P5\_P1 = P5\_P2\*P2\_P1;

### P2

% Given Properties

M1 = 2.9;

theta = 21; % [deg]

gamma = 1.4;

P1 = 17.9; % [kPa]

beta = theta\_beta\_M\_relation(theta,M1,gamma)

Mn1 = M1\*sind(beta)

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

M2 = Mn2/sind(beta-theta)

P2\_P1 = normalShock\_jump\_P\_static(Mn1,gamma)

P2 = P2\_P1\*P1

### P3

% Exit condtions

Ae\_At = 4.0;

At = 1;

Ae = Ae\_At\*At;

Mt = 1;

gamma = 1.4;

P0 = 500e3; % [Pa]

Pa = 10e3; % [Pa]

[Me\_sub, Me\_sup] = M\_for\_area\_ratio(At,Ae,Mt,gamma);

Me = Me\_sup;

Pe = p\_from\_M\_and\_gamma(P0,Me,gamma,"static");

Pa\_Pe = Pa/Pe;

M2 = M\_from\_P\_ratio(P0,Pa,gamma)

% Expansion angles

nu\_Me = Prandtl\_Meyer\_Expansion(Me,gamma)

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

theta = nu\_M2 - nu\_Me

### P4

% Defining the given properties

M1 = 2;

P1 = 101325; % [Pa]

theta1 = 10; % [deg]

T1 = 300;

T0 = 1; % dummy

gamma = 1.4;

% Region 2

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

Mn1 = M1\*sind(beta12)

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

M2 = Mn2/sind(beta12-theta1)

P2\_P1 = normalShock\_jump\_P\_static(Mn1,gamma)

P2 = P1\*P2\_P1

T2\_T1 = normalShock\_jump\_T\_static(Mn1,gamma)

T2 = T2\_T1\*T1

% Region 3'

theta2 = theta1;

beta23 = theta\_beta\_M\_relation(theta2,M2,gamma)

Mn2p = M2\*sind(beta23)

Mn3 = normalShock\_jump\_M(Mn2p,gamma)

M3 = Mn3/sind(beta23 - theta2)

P3\_P2 = normalShock\_jump\_P\_static(Mn2p,gamma)

T3\_T2 = normalShock\_jump\_T\_static(Mn2p,gamma)

P3 = P3\_P2\*P2

T3 = T3\_T2\*T2

P03 = p\_from\_M\_and\_gamma(P3,M3,gamma,"stagnation")

% (b)

h1 = 1;

ang\_BAC = 90 - beta12;

BC = h1\*tand(ang\_BAC);

AC = h1/cosd(ang\_BAC);

ang\_CAE = beta12 - theta1;

ang\_ECA = 180 - (beta23 - theta1) - beta12;

ang\_AEC = beta23;

AE = sind(ang\_ECA)/sind(ang\_AEC)\*AC;

EF = AE\*sind(theta1);

h3 = h1 - EF;

% (c) Expansion fan

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

nu\_M3p = theta1 + nu\_M2

M3p = calc\_M\_from\_PrantlMeyer(nu\_M3p,gamma)

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

P0\_P3p = isentropic\_relation\_P\_ratio(M3p,gamma)

P3p\_P2 = P0\_P2/P0\_P3p

P3p = P3p\_P2\*P2

T0\_T2 = isentropic\_relation\_T\_ratio(M2,gamma)

T0\_T3p = isentropic\_relation\_T\_ratio(M3p,gamma)

T3p\_T2 = T0\_T2/T0\_T3p

T3p = T3p\_T2\*T2

P03p = P0\_P3p\*P3p

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

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

end

function T\_rat = isentropic\_relation\_T\_ratio(M,gamma)

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

end

function [M2\_sub, M2\_sup] = M\_for\_area\_ratio(A1,A2,M1,gamma)

% Calculate the Mach number at the inlet

M2 = sym('M2');

assume(M2,["real","positive"])

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

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

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

eqn = A2/A1 == M1/M2 \* (a1/a2)^(a3);

M2 = double(vpasolve(eqn,M2));

M2 = M2(M2 == real(M2));

M2\_sub = min(M2);

M2\_sup = max(M2);

end

function M = M\_from\_P\_ratio(P0,P,gamma)

a1 = 2/(gamma - 1);

a2 = (P0/P)^((gamma - 1)/gamma);

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

end

function T\_rat = normalShock\_jump\_T\_static(M1,gamma)

%{

Function: normalShock\_jump\_T\_stati

Author: Tomoki Koike

Description: This function calculates the static pressure ratios

before and after a normal shockwave.

>>Inputs

M1: Mach number before shockwave

gamma: specific hear ratio

Outputs<<

T\_rat: static temperature ratio

%}

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

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

a3 = (gamma + 1)\*M1^2;

T\_rat = (a1)\*(a2)/(a3);

end

function P\_rat = normalShock\_jump\_P\_static(M1,gamma)

%{

Function: normalShock\_jump\_P\_static

Author: Tomoki Koike

Description: This function calculates the static pressure ratios

before and after a normal shockwave.

>>Inputs

M1: Mach number before shockwave

gamma: specific hear ratio

Outputs<<

P\_rat: static pressure ratio

%}

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

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

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 p2 = p\_from\_M\_and\_gamma(p1, M, gamma, type)

if type == "stagnation"

p2 = p1 \* (1 + (gamma - 1) / 2 \* M^2)^(gamma/(gamma - 1));

elseif type == "static"

p2 = p1 / (1 + (gamma - 1) / 2 \* M^2)^(gamma/(gamma - 1));

else

disp("Error. Incorrect type. Type can only be 'stagnation' or 'static'.")

end

end