AAE 334: Aerodynamics

HW 9: Nozzle Flow Analysis

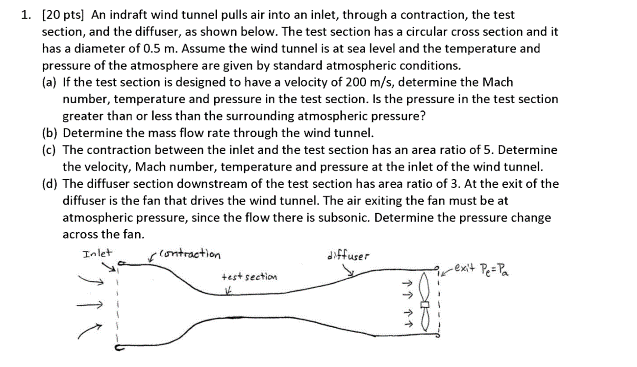
Dr. Blaisdell

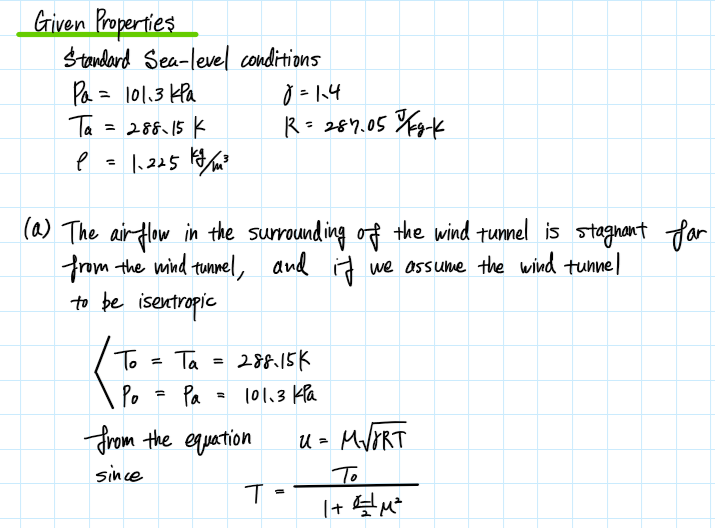
School of Aeronautical and Astronautical

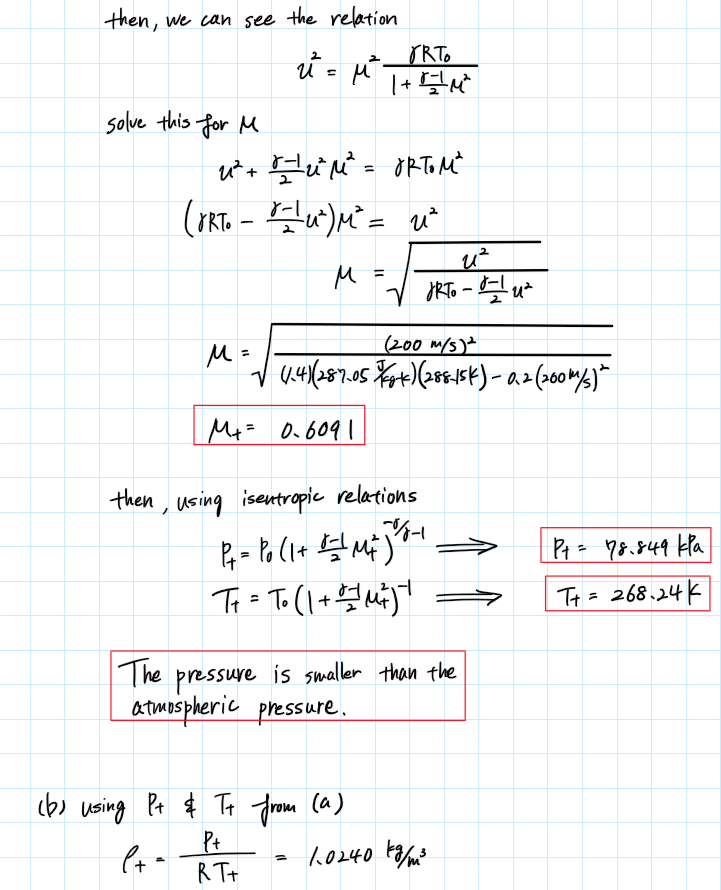
Purdue University

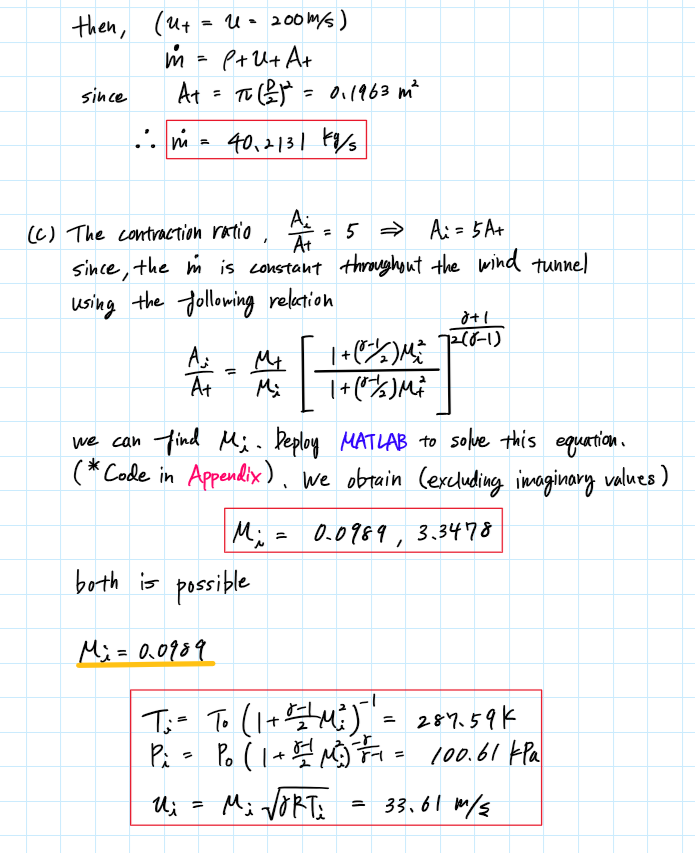
Tomoki Koike

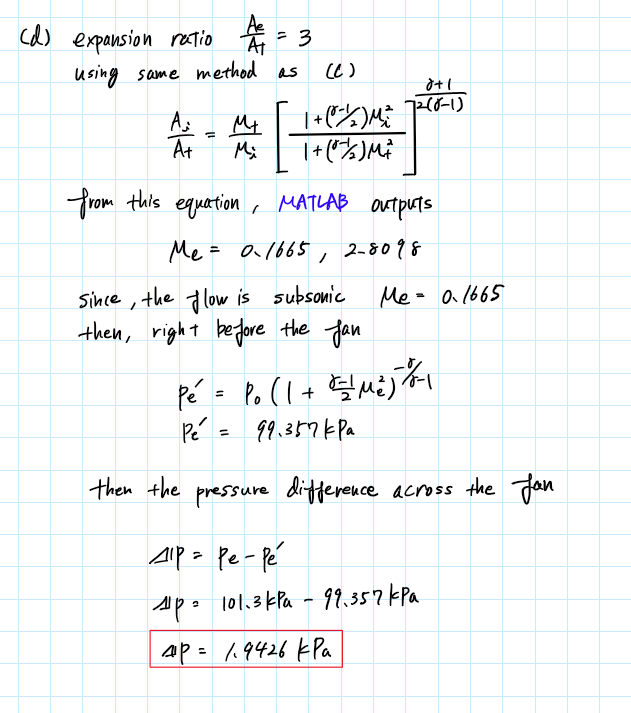
Friday April 10th, 2020

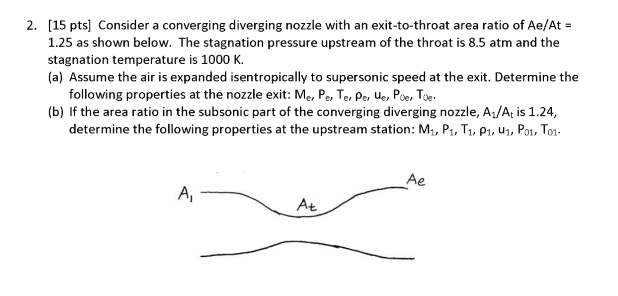


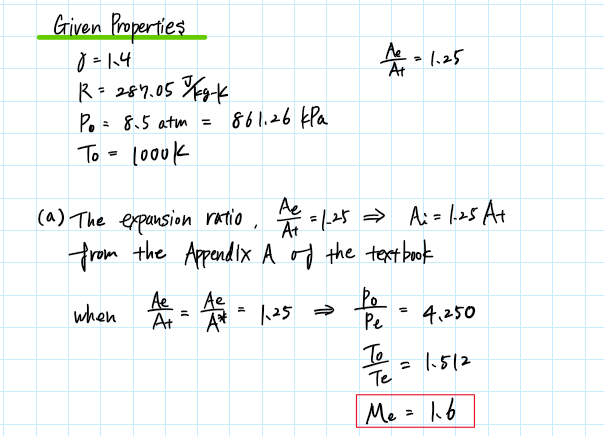


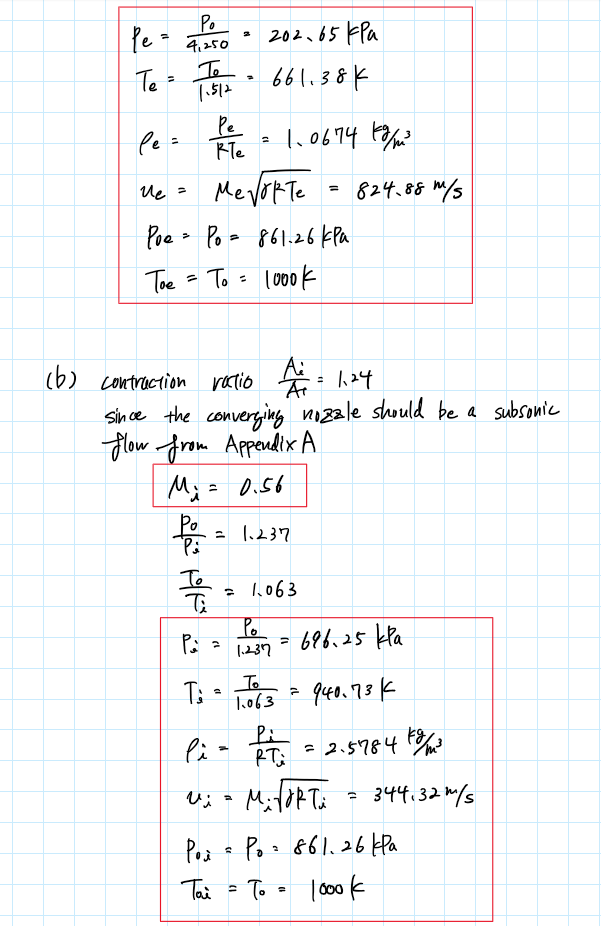


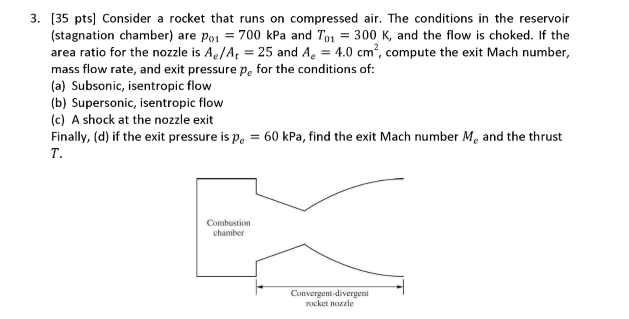


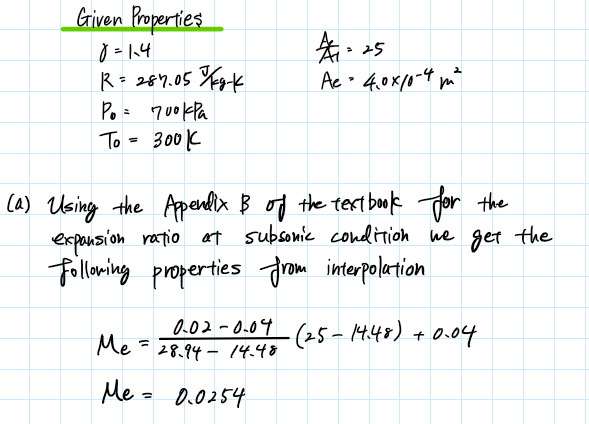


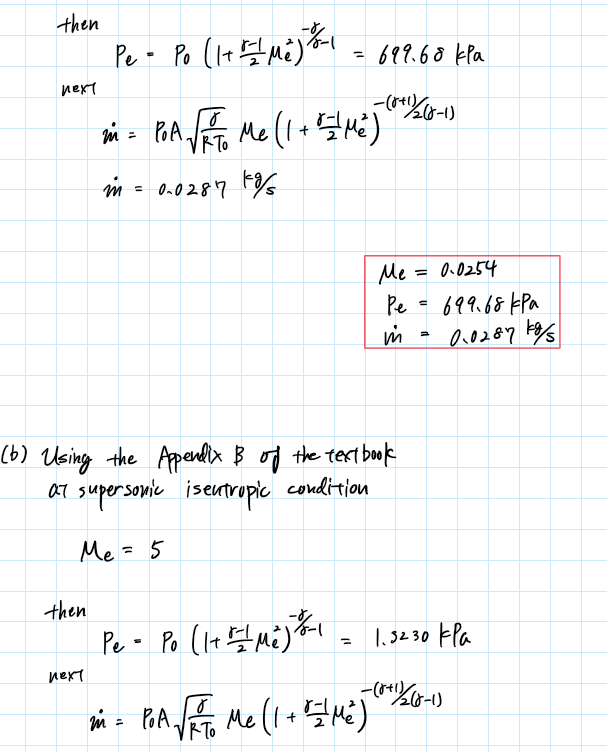


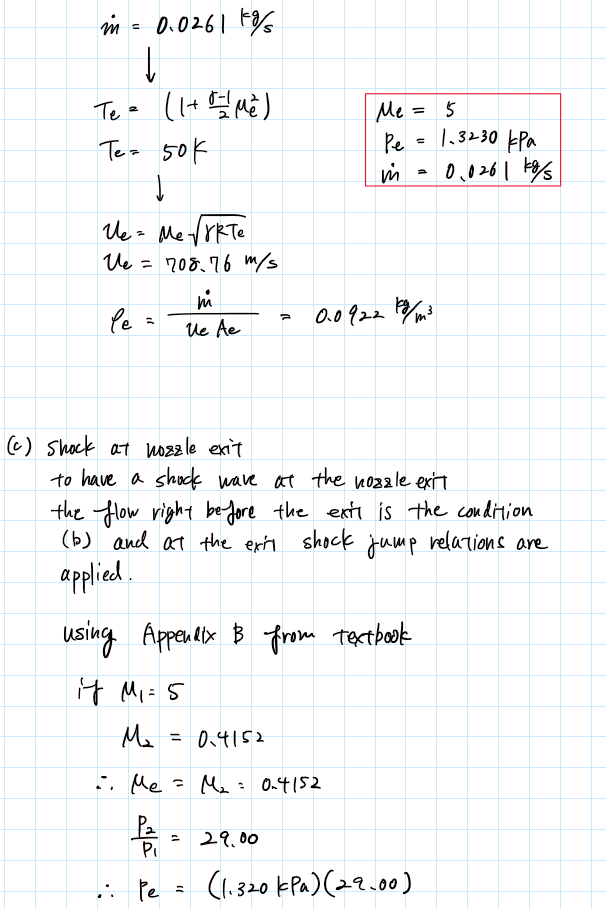


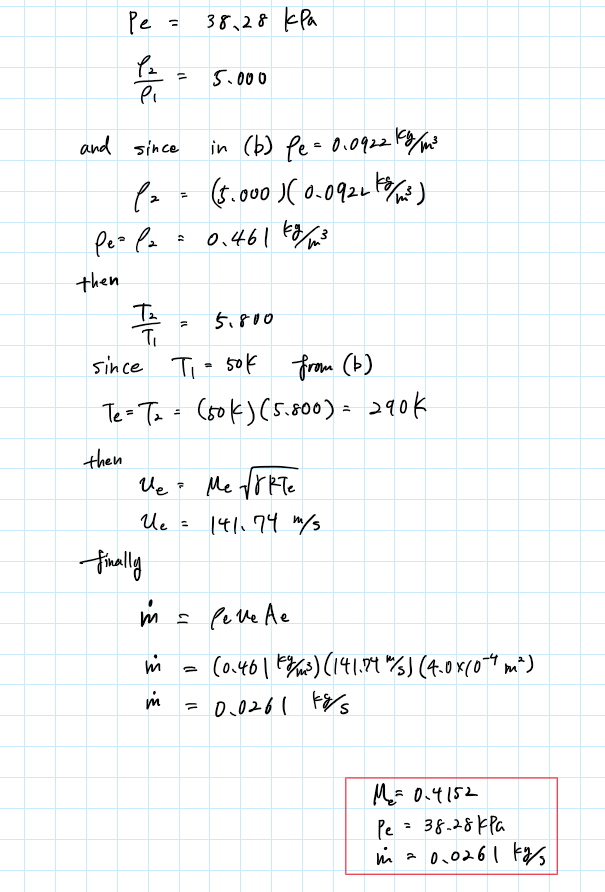


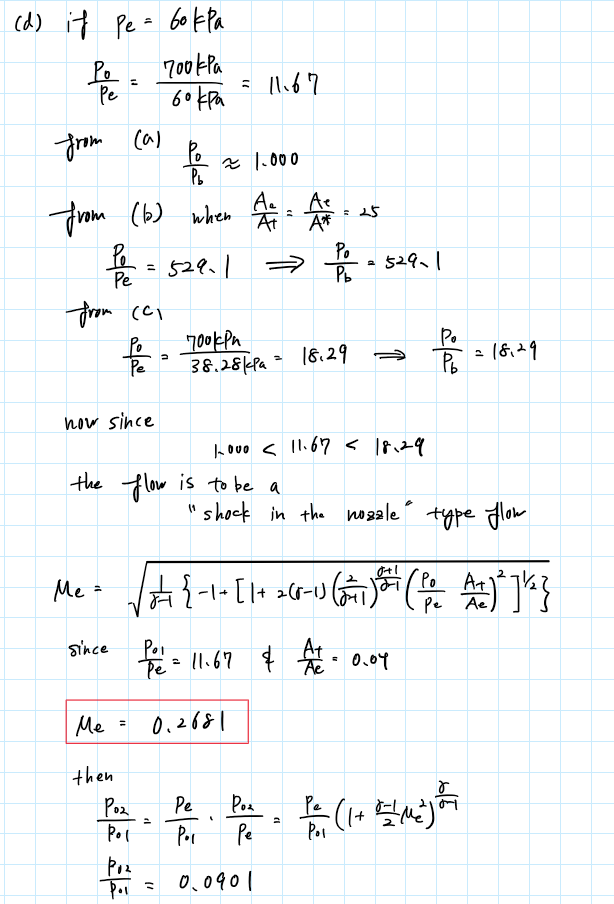


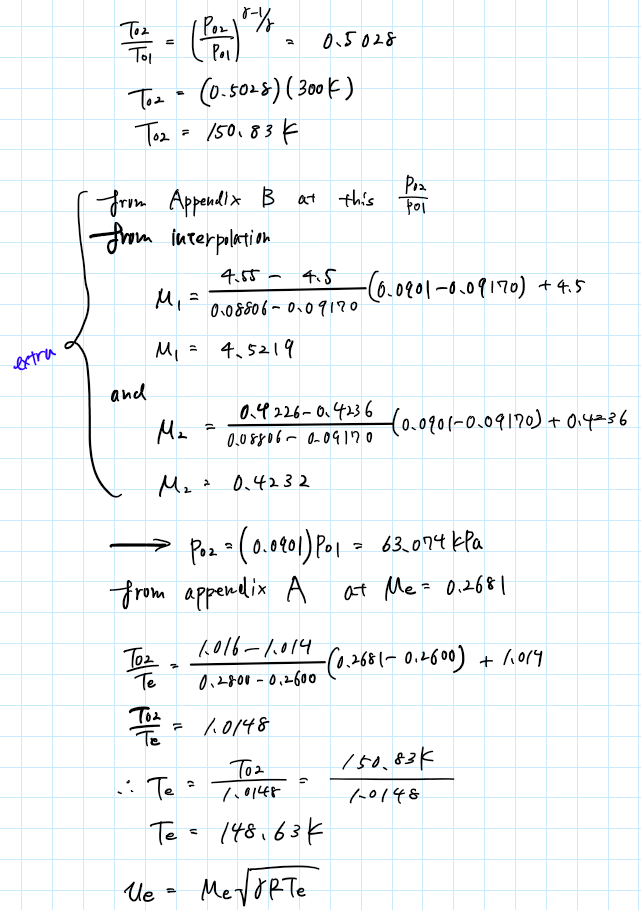


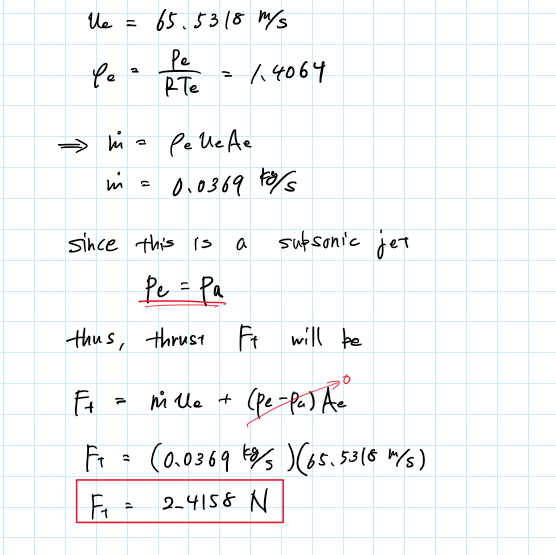


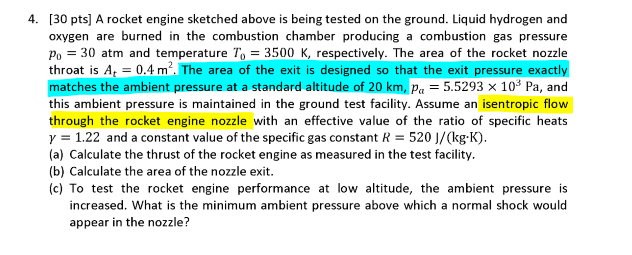


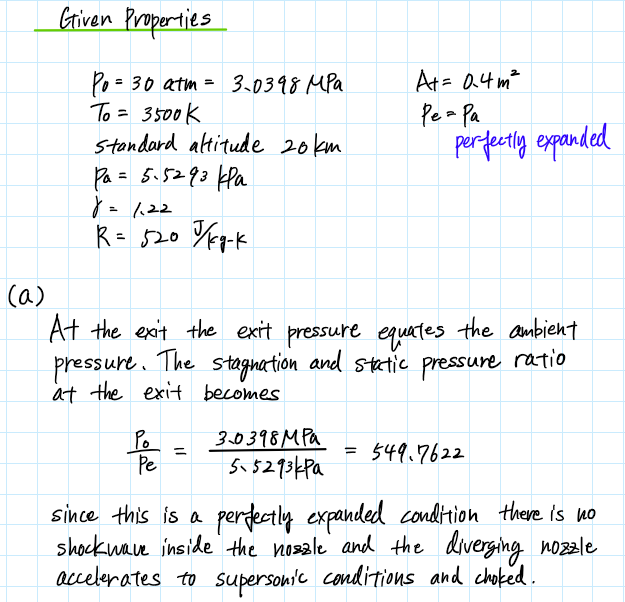


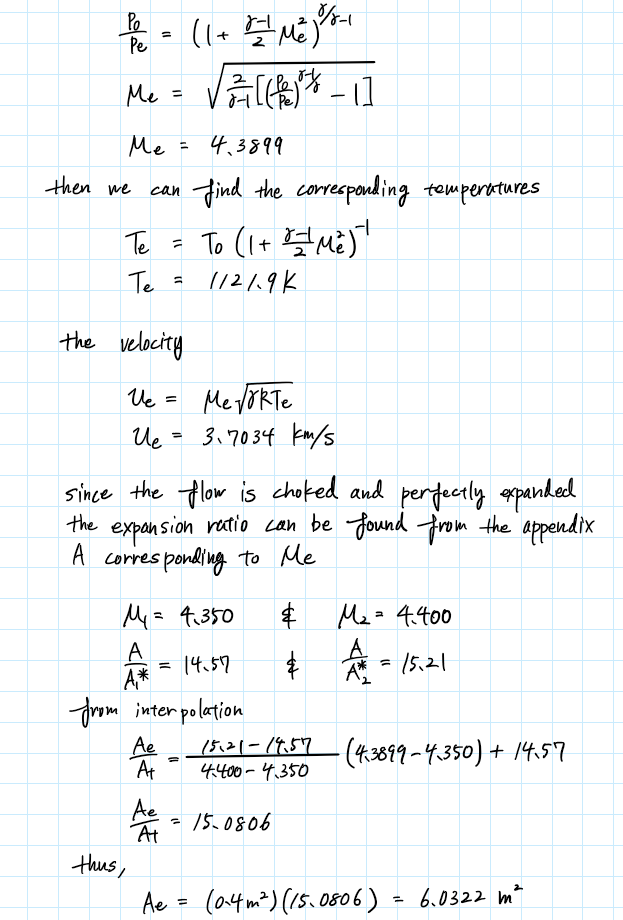


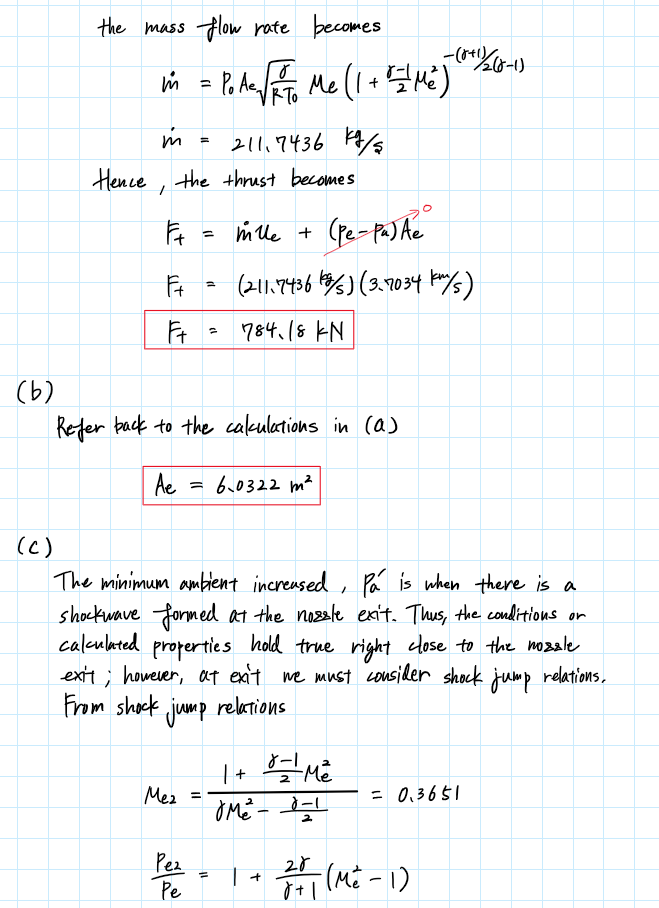


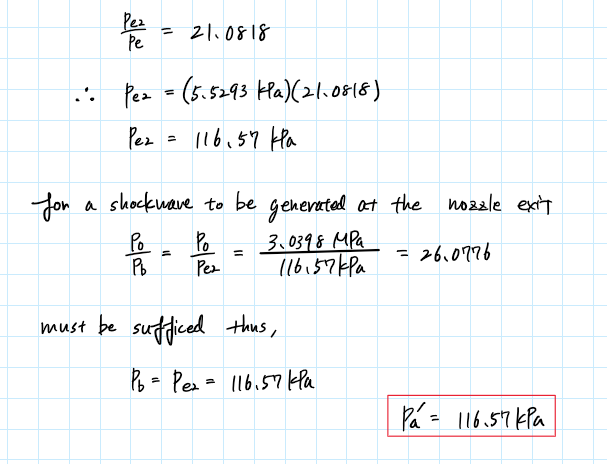












Appendix

## **AAE 334 HW9**

clear all; close all; clc;

### 1-a

% Given properties

Pa = 101.3e3;

Ta = 288.15;

rho\_a = 1.225;

gamma = 1.4;

R = 287.05;

D = 0.5;

A\_t = pi\*(D/2)^2;

% Mach number

P0 = Pa; T0 = Ta;

u\_t = 200;

den = u\_t^2;

num = gamma\*R\*T0 - (gamma - 1)/2\*u\_t^2;

M\_t = sqrt(den/num);

% Pressure

P\_t = p\_from\_M\_and\_gamma(P0,M\_t,gamma,"static")

% Temperature

T\_t = T\_from\_M\_and\_gamma(T0,M\_t,gamma,"static")

### 1-b

% Mass flow rate

rho\_t = P\_t/T\_t/R;

m\_dot = rho\_t\*u\_t\*A\_t;

### 1-c

A\_i = 5\*A\_t;

% Calculate the Mach number at the inlet

syms M\_i

assume(M\_i,["real","positive"])

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

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

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

eqn = A\_i/A\_t == M\_t/M\_i \* (a1/a2)^(a3);

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

M\_i = M\_i(M\_i == real(M\_i));

M\_i\_sub = min(M\_i);

M\_i\_sup = max(M\_i);

% If inlet subsonic

T\_i\_sub = T\_from\_M\_and\_gamma(T0,M\_i\_sub,gamma,"static");

P\_i\_sub = p\_from\_M\_and\_gamma(P0,M\_i\_sub,gamma,"static");

u\_i\_sub = M\_i\_sub\*sqrt(gamma\*R\*T\_i\_sub);

% If inlet is supersonic

T\_i\_sup = T\_from\_M\_and\_gamma(T0,M\_i\_sup,gamma,"static");

P\_i\_sup = p\_from\_M\_and\_gamma(P0,M\_i\_sup,gamma,"static");

u\_i\_sup = M\_i\_sup\*sqrt(gamma\*R\*T\_i\_sup);

### 1-d

A\_e = 3\*A\_t;

% Calculate the Mach number at the inlet

syms M\_e

assume(M\_e,["real","positive"])

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

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

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

eqn = A\_e/A\_t == M\_t/M\_e \* (a1/a2)^(a3);

M\_e = double(vpasolve(eqn,M\_e))

M\_e = M\_e(M\_e == real(M\_e))

M\_e\_sub = min(M\_e)

M\_e\_sup = max(M\_e)

Pe\_b = p\_from\_M\_and\_gamma(P0,M\_e\_sub,gamma,"static")

delta\_P = Pa - Pe\_b

### 3-a

% Given properties

gamma = 1.4;

R = 287.05;

P0 = 700e3;

T0 = 300;

epsilon = 25;

Ae = 4e-4;

At = Ae/25;

Me = two\_point\_interpolate(25,14.48,28.94,0.04,0.02)

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

m\_dot = mDot\_from\_M(Me,P0,T0,Ae,gamma,R)

### 3-b

Me = 5;

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

m\_dot = mDot\_from\_M(Me,P0,T0,Ae,gamma,R)

Te = T\_from\_M\_and\_gamma(T0,Me,gamma,"static")

ue = Me\*sqrt(gamma\*R\*Te)

rho\_e = m\_dot/ue/Ae

### 3-d

Pe = 60e3;

Me = shockInNozzle\_M(P0,Pe,At,Ae,gamma)

P02\_P01 = Pe/P0\*(1 + (gamma - 1)/2\*Me^2)^(gamma/(gamma - 1))

M1 = two\_point\_interpolate(P02\_P01,0.0917,0.08806,4.5,4.55)

M2 = two\_point\_interpolate(P02\_P01,0.0917,0.08806,0.4236,0.4226)

P02 = P02\_P01\*P0

T02\_T01 = (P02\_P01)^((gamma - 1)/gamma)

T02 = T02\_T01\*T0

T02\_Te = two\_point\_interpolate(Me,0.26,0.28,1.014,1.016)

Te = T02/T02\_Te

ue = Me\*sqrt(gamma\*R\*Te)

rho\_e = Pe/R/Te

m\_dot = rho\_e\*ue\*Ae

Ft = m\_dot\*ue

### 4-a

% Given properties

P0 = 3.0398e6;

T0 = 3500;

Pa = 5.5293e3;

gamma = 1.22;

R = 520;

At = 0.4;

Pe = Pa;

P0\_Pe = P0/Pe

Me = M\_from\_P\_ratio(P0,Pe,gamma)

Te = T\_from\_M\_and\_gamma(T0,Me,gamma,"static")

ue = Me\*sqrt(gamma\*R\*Te)

Ae\_At = two\_point\_interpolate(Me,4.35,4.4,14.57,15.21)

Ae = Ae\_At\*At

m\_dot = mDot\_from\_M(Me,P0,T0,Ae,gamma,R)

Ft = m\_dot\*ue + (Pe - Pa)\*Ae

### 4-c

% Shock jump relations

Me2 = shock\_jump\_M(Me,gamma)

Pe2\_Pe = 1 + 2\*gamma/(gamma + 1)\*(Me^2 - 1)

Pe2 = Pe2\_Pe\*Pe

P0\_Pb = P0/Pe2

### Function

function m\_dot = mDot\_from\_M(M,P0,T0,A,gamma,R)

a1 = P0\*A;

a2 = sqrt(gamma/R/T0);

a3 = (1 + (gamma - 1)/2\*M^2);

a4 = -(gamma + 1)/2/(gamma - 1);

m\_dot = a1\*a2\*M\*a3^(a4);

end

function M = shockInNozzle\_M(P0,Pe,At,Ae,gamma)

a1 = (gamma - 1);

a2 = (gamma + 1);

a3 = 2\*a1\*(2/a2)^(a2/a1);

a4 = (P0/Pe \* At/Ae)^2;

a5 = 1 + a3\*a4;

M = sqrt(1/a1 \* (-1 + (a5)^(0.5)));

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 M2 = shock\_jump\_M(M1,gamma)

a1 = (gamma - 1)/2;

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

end

function T2 = T\_from\_M\_and\_gamma(T1, M, gamma, type)

if type == "stagnation"

T2 = T1 \* (1 + (gamma - 1) / 2 \* M^2);

elseif type == "static"

T2 = T1 / (1 + (gamma - 1) / 2 \* M^2);

else

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

end

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