AAE 339

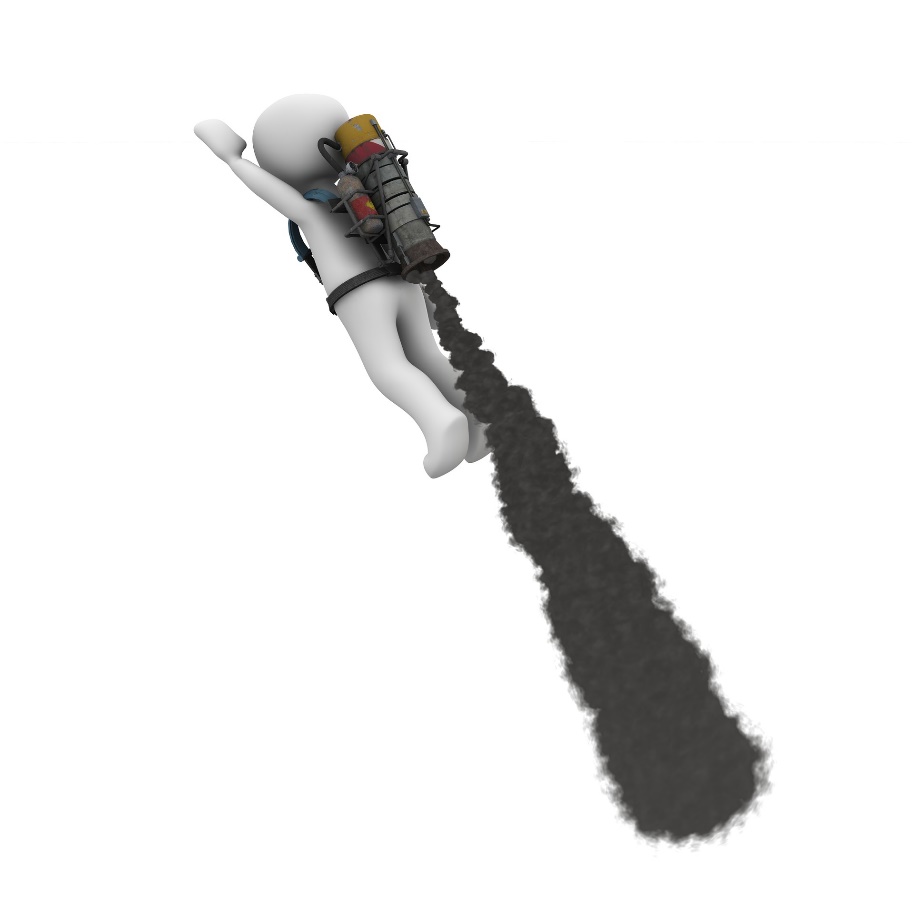
HW7: Compressor Stage Analysis

Dr. Anderson

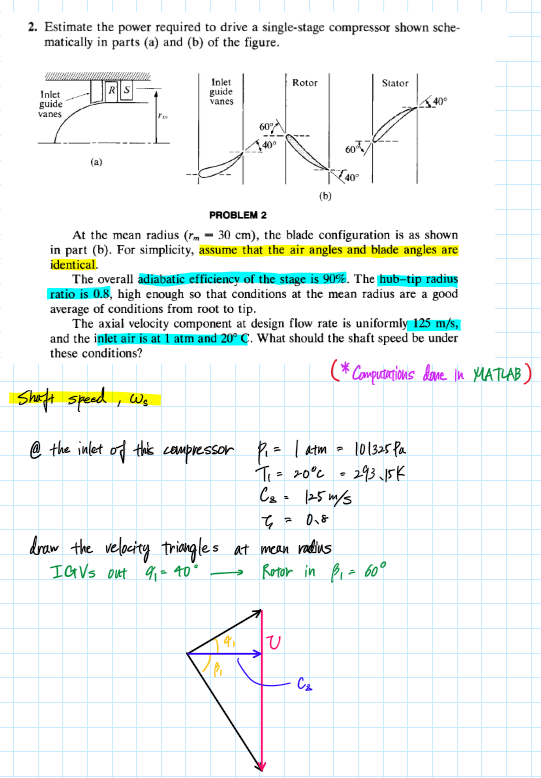
School of Aeronautical and Astronautical Engineering

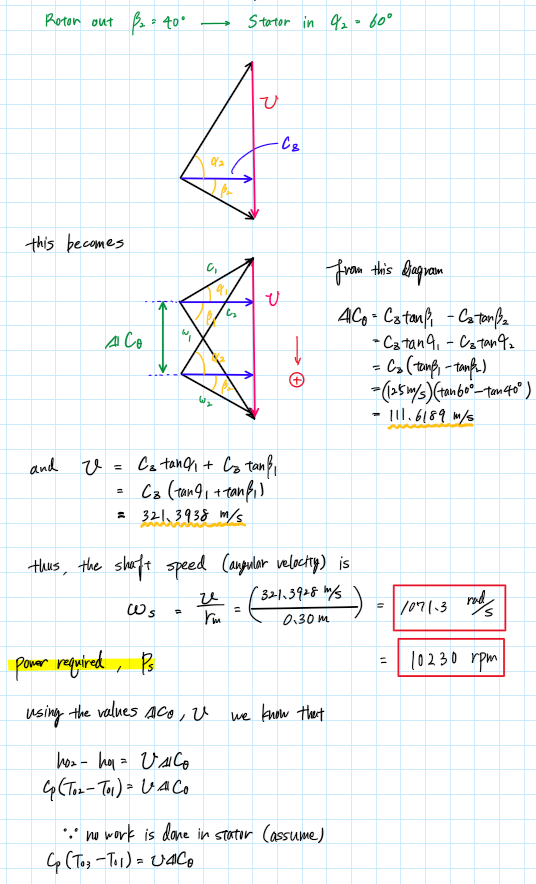
Tomoki Koike

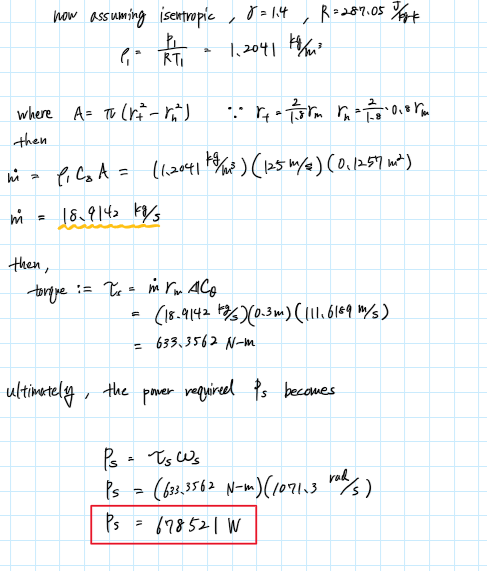
Thursday March 12, 2020



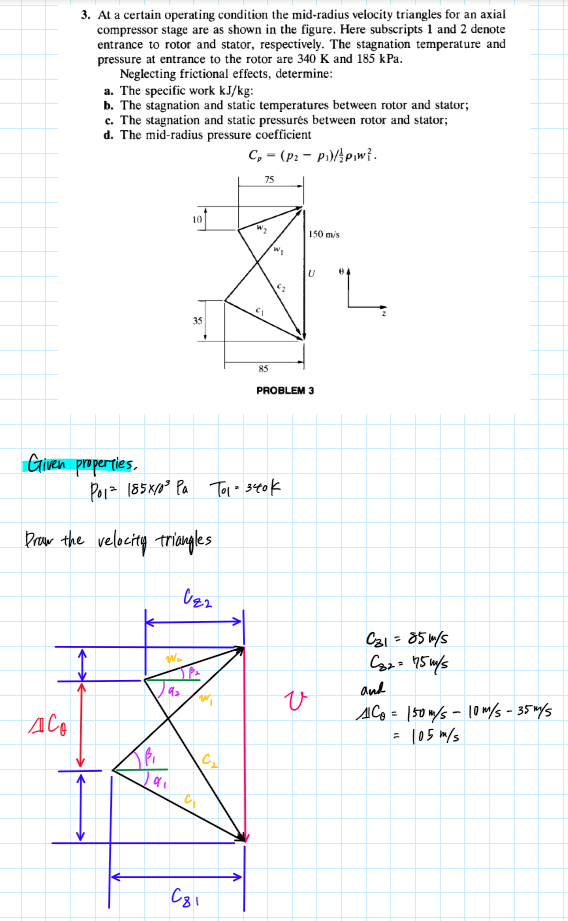
Problem 7.2

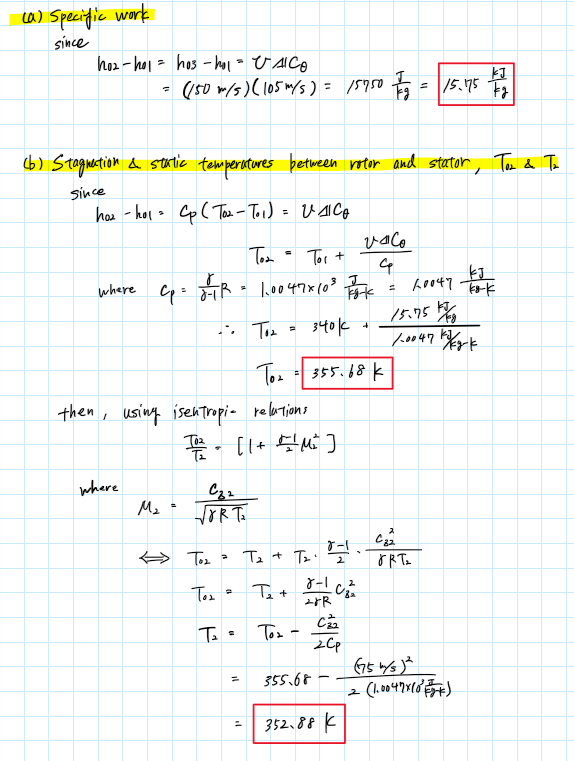


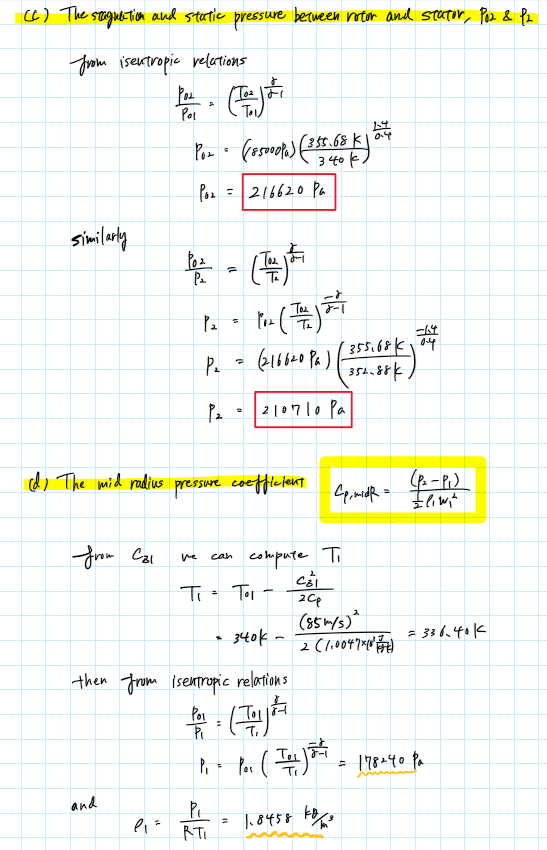


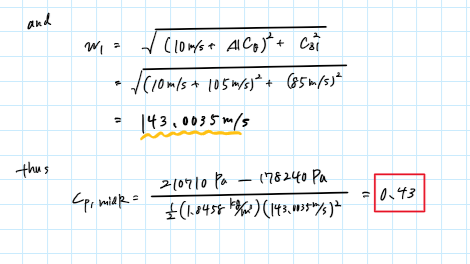


Problem 7.3

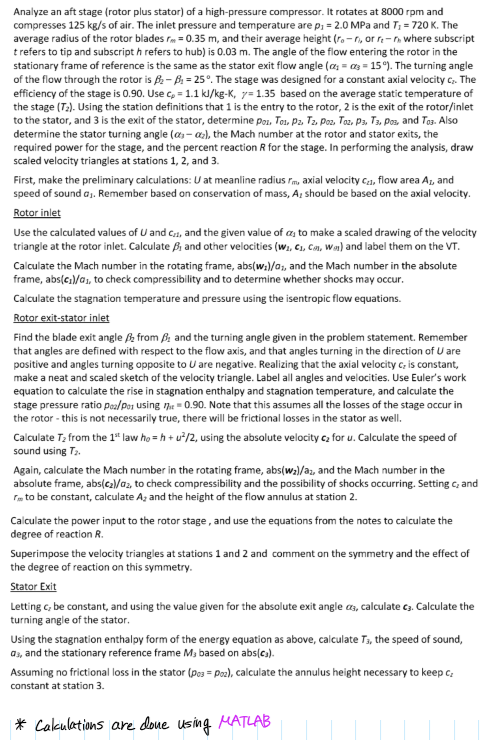


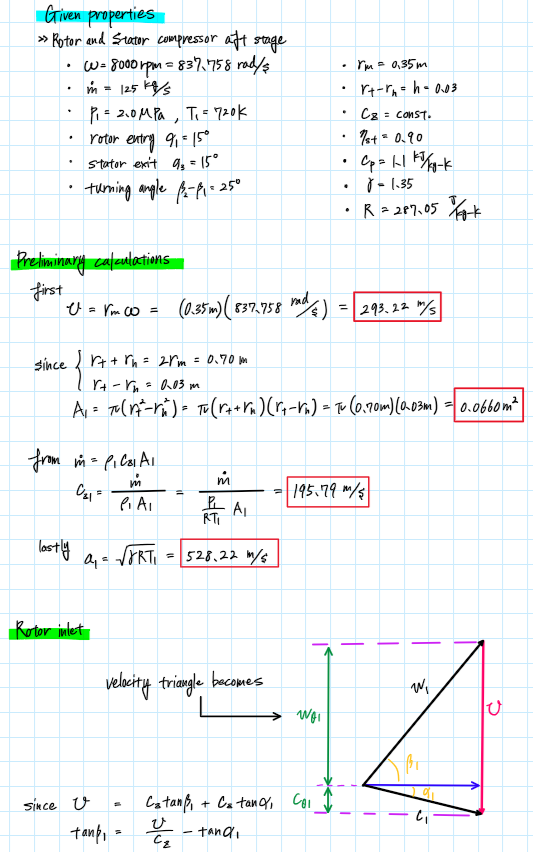


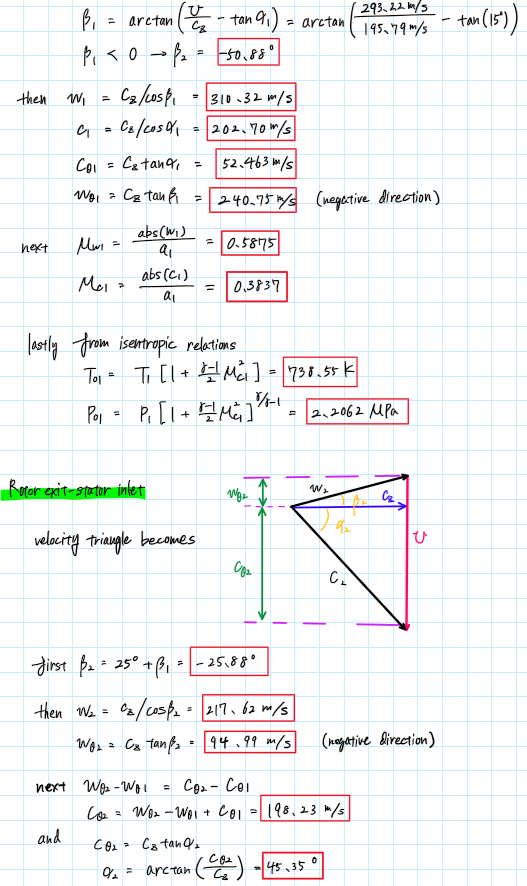




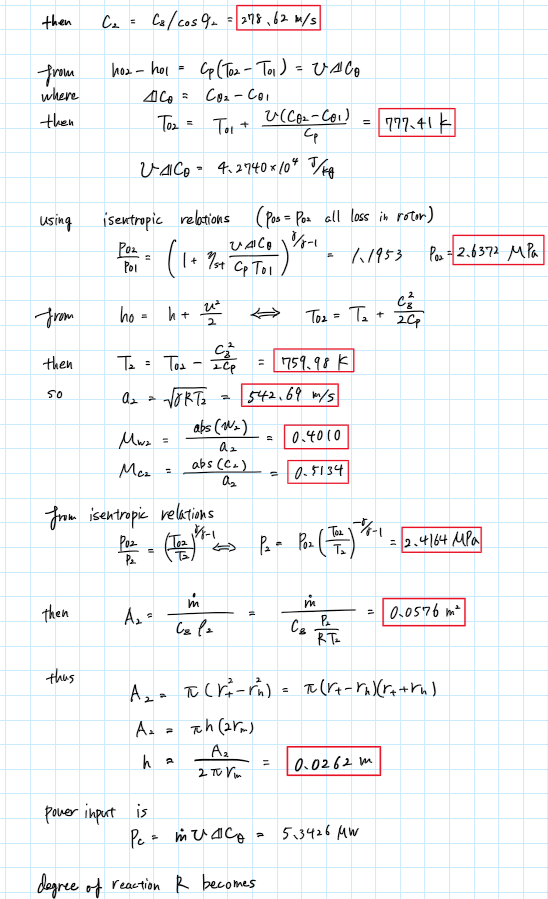
Problem 3

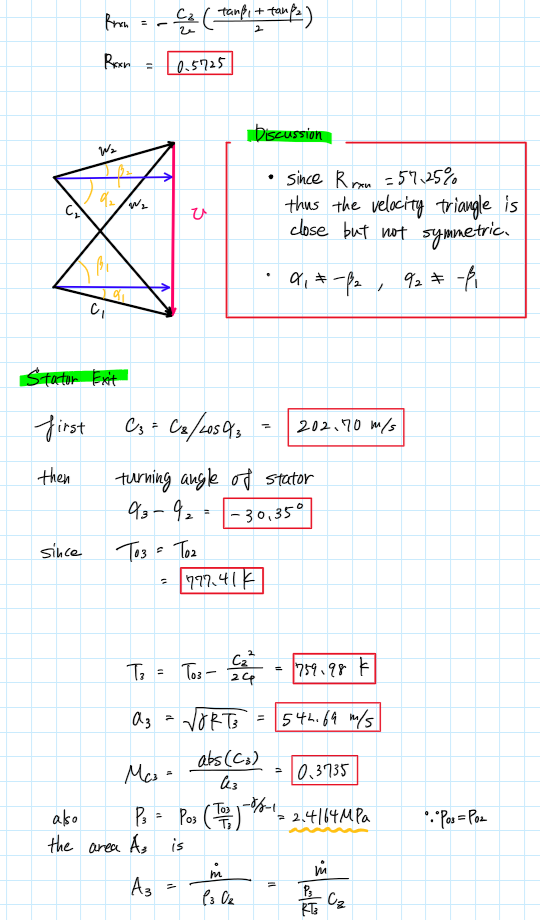




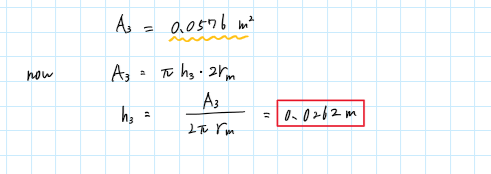












Appendix

**AAE 339 HW 7 MATLAB CODE**

close all; clear all; clc;

**p1.**

% Defining given properties

P1 = 101325; % [Pa]

T1 = 293.15; % [K]

c\_z = 125; % [m/s]

r\_m = 0.3; % [m]

zeta = 0.8;

alpha1 = 40; % [deg]

alpha2 = 60; % [deg]

beta1 = 60; % [deg]

beta2 = 40; % [deg]

gamma = 1.4;

R = 287.05;

delta\_c\_theta = c\_z\*(tand(beta1) - tand(beta2))

U = c\_z\*(tand(alpha1) + tand(beta1))

w\_s = U/r\_m

w\_c = U\*delta\_c\_theta

rho1 = P1/R/T1

% Calculate m\_dot

r\_t = 2/1.8\*r\_m;

r\_h = r\_t\*0.8;

A = pi\*(r\_t^2 - r\_h^2)

m\_dot = rho1\*A\*c\_z

tau\_s = m\_dot\*r\_m\*delta\_c\_theta

Pow\_s = tau\_s\*w\_s

**p2.**

clear all; close all; clc;

% <a>

P01 = 185000; % [Pa]

T01 = 340; % [K]

delta\_c\_theta = 105; % [m/s]

U = 150; % [m/s]

c\_z1 = 85; % [m/s]

c\_z2 = 75; % [m/s]

Pow\_s = U\*delta\_c\_theta

% <b>

gamma = 1.4;

R = 287.05;

c\_p = gamma/(gamma - 1)\*R;

T02 = T01 + Pow\_s/c\_p

T2 = T02 - c\_z2^2/c\_p/2

% <c>

P02 = P\_from\_isentropic\_relation(P01, T02, T01, gamma, "1")

P2 = P\_from\_isentropic\_relation(P02, T02, T2, gamma, "2")

% <d>

T1 = T01 - c\_z1^2/2/c\_p

P1 = P\_from\_isentropic\_relation(P01, T01, T1, gamma, "2")

rho1 = P1/R/T1

w1 = sqrt((10 + delta\_c\_theta)^2 + c\_z1^2)

C\_P\_midR = (P2 - P1)/2/rho1/w1^2

**p3.**

clear all; close all; clc;

omega = 8000\*2\*pi/60; % [rad/s]

m\_dot = 125; % [kg/s]

P1 = 2e6; % [Pa]

T1 = 720; % [K]

alpha1 = deg2rad(15);

alpha3 = alpha1;

turn\_ang = deg2rad(25);

r\_m = 0.35; % [m]

h = 0.03; % r\_t - r\_h

eta = 0.90;

cp = 1.1e3; % [kg/kg/K]

gamma = 1.35;

R = 287.05;

PRELIMINARY CALCULATIONS

U = r\_m\*omega

A1 = pi\*2\*r\_m\*h

c\_z1 = m\_dot/(P1/R/T1)/A1

a1 = sqrt(gamma\*R\*T1)

ROTOR INLET

c\_z = c\_z1

beta1 = -atan(U/c\_z - tan(alpha1))

beta1\_deg = rad2deg(beta1)

w1 = c\_z/cos(beta1)

c1 = c\_z/cos(alpha1)

c\_theta1 = c\_z\*tan(alpha1)

w\_theta1 = c\_z\*tan(beta1)

M\_w1 = abs(w1)/a1

M\_c1 = abs(c1)/a1

T01 = T\_from\_M\_and\_gamma(T1,M\_c1,gamma,"stagnation")

P01 = p\_from\_M\_and\_gamma(P1,M\_c1,gamma,"stagnation")

Rotor exit-stator inlet

beta2 = beta1 + turn\_ang;

beta2\_deg = rad2deg(beta2)

w2 = c\_z/cos(beta2)

w\_theta2 = c\_z\*tan(beta2)

c\_theta2 = w\_theta2 - w\_theta1 + c\_theta1

alpha2 = atan(c\_theta2/c\_z)

alpha2\_deg = rad2deg(alpha2)

c2 = c\_z/cos(alpha2)

T02 = T01 + U\*(c\_theta2-c\_theta1)/cp

rise = U\*(c\_theta2-c\_theta1)

P02\_P01 = (1 + eta\*rise/cp/T01)^(gamma/(gamma - 1))

P02 = P02\_P01\*P01

T2 = T02 - c\_z^2/2/cp

a2 = sqrt(gamma\*R\*T2)

M\_w2 = abs(w2)/a2

M\_c2 = abs(c2)/a2

P2 = P02\*(T02/T2)^(-gamma/(gamma - 1))

A2 = m\_dot/c\_z/(P2/R/T2)

h2 = A2/2/pi/r\_m

Pc = m\_dot\*U\*(c\_theta2-c\_theta1)

deg\_rxn = -c\_z/U\*(tan(beta1) + tan(beta2))/2

Stator exit

c3 = c\_z/cos(alpha3)

turning\_stator = alpha3 - alpha2

turning\_stator\_deg = rad2deg(turning\_stator)

T03 = T02

T3 = T03 - c\_z^2/2/cp

a3 = sqrt(gamma\*R\*T3)

M\_c3 = abs(c3)/a3

P03 = P02

P3 = P03\*(T03/T3)^(-gamma/(gamma - 1))

A3 = m\_dot/(P3/R/T3)/c\_z

h3 = A3/2/pi/r\_m

function P2 = P\_from\_isentropic\_relation(P1, T2, T1, gamma, type)

if type == "1"

P2 = P1 \* (T2 / T1)^(gamma / (gamma - 1));

elseif type == "2"

P2 = P1 \* (T2 / T1)^(-gamma / (gamma - 1));

else

disp("You can only enter 1 or 2 for type.")

end

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