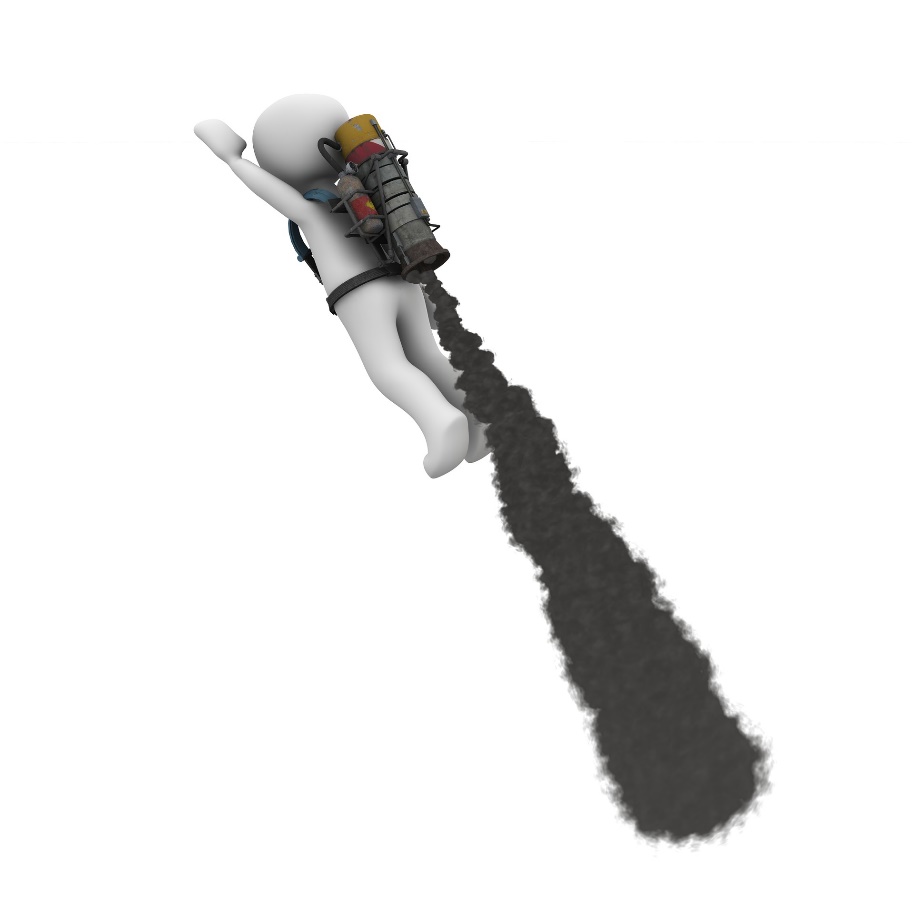
AAE 339: Aerospace Propulsion

HW6: Realistic Turbofan Cycle Analysis

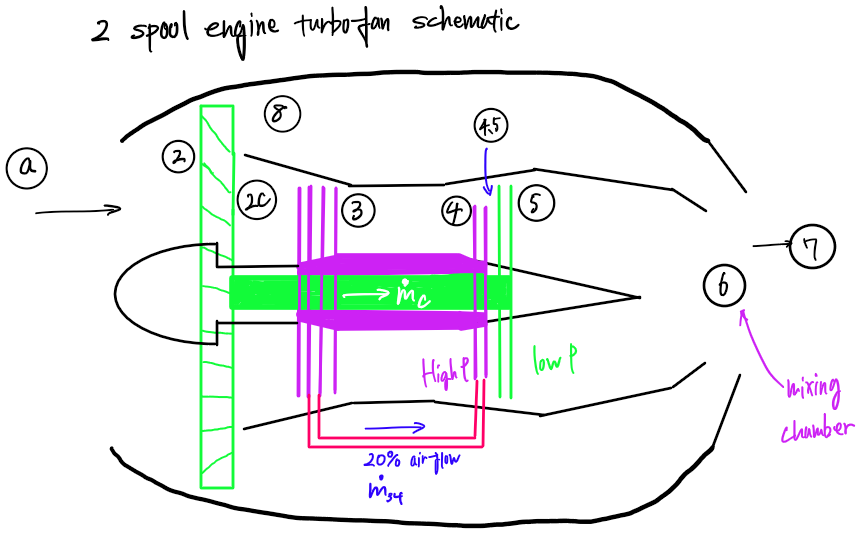
Dr. Anderson

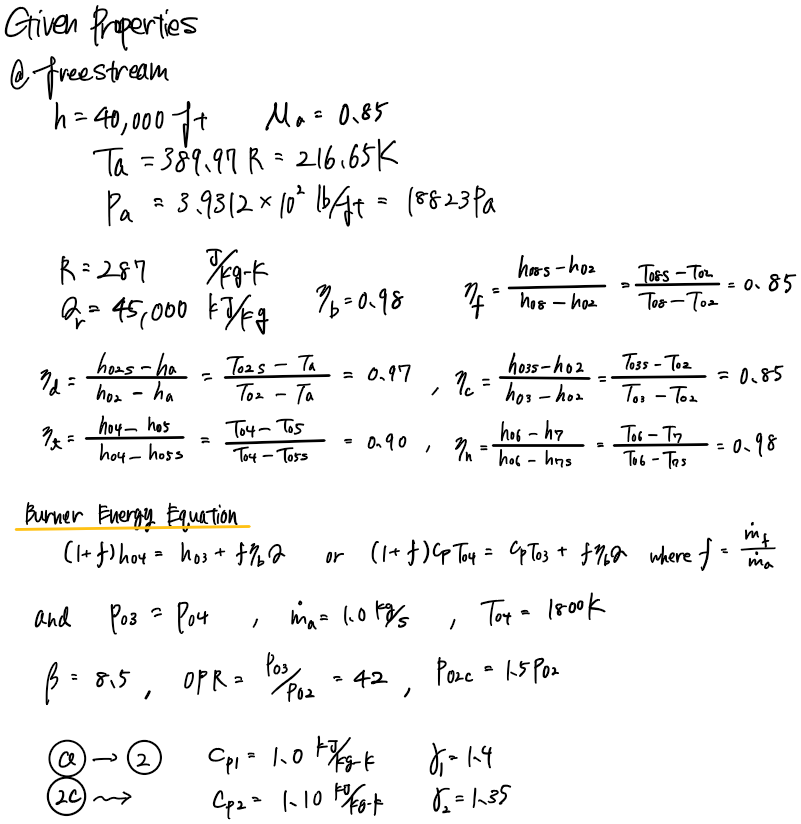
Tomoki Koike

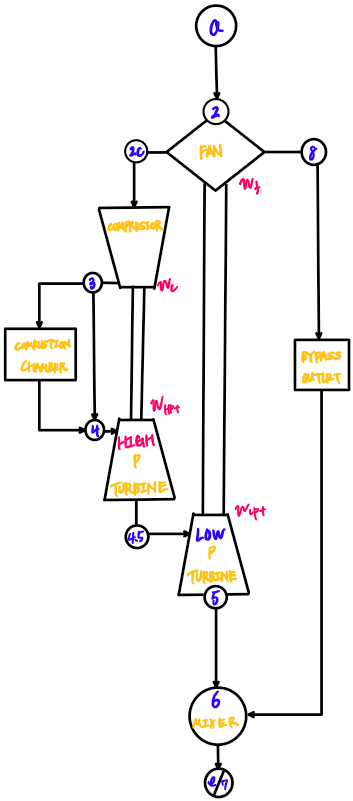
Thursday March 5, 2020





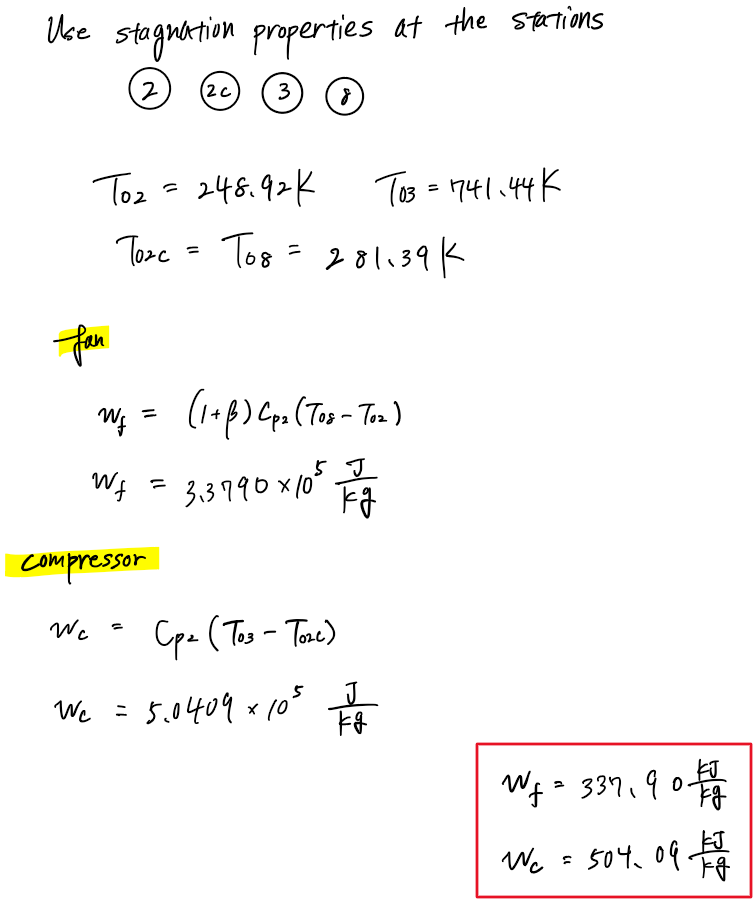




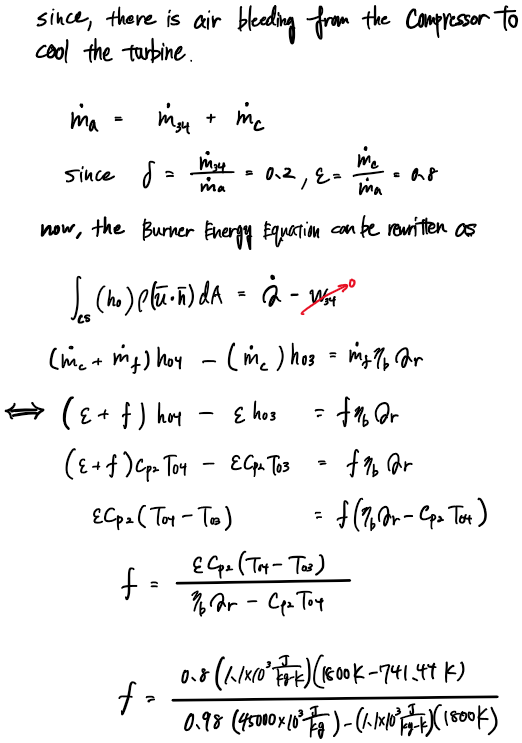


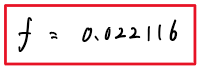
To solve (b) we use answers from part (d)

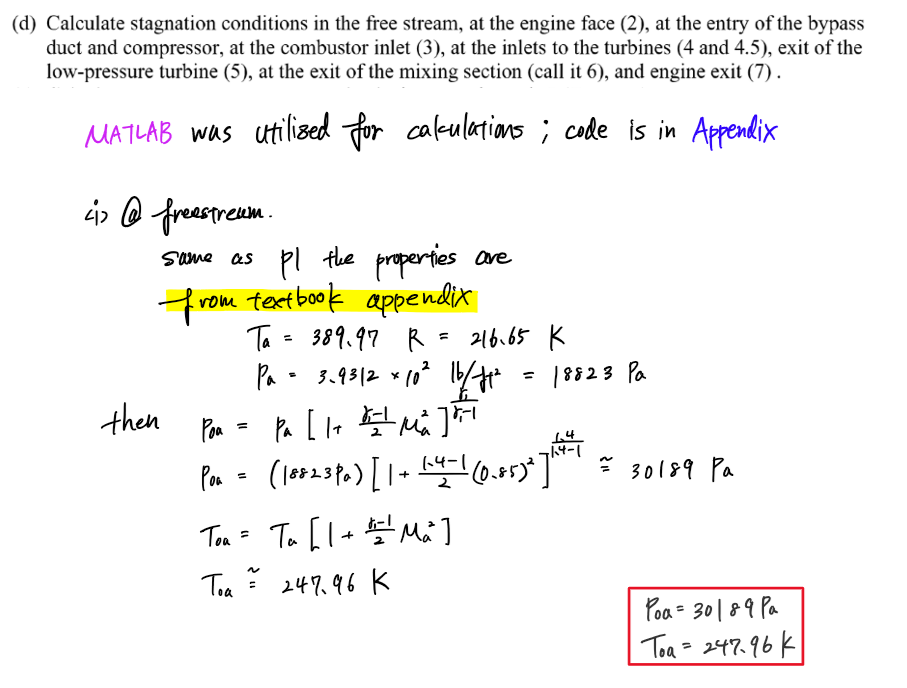


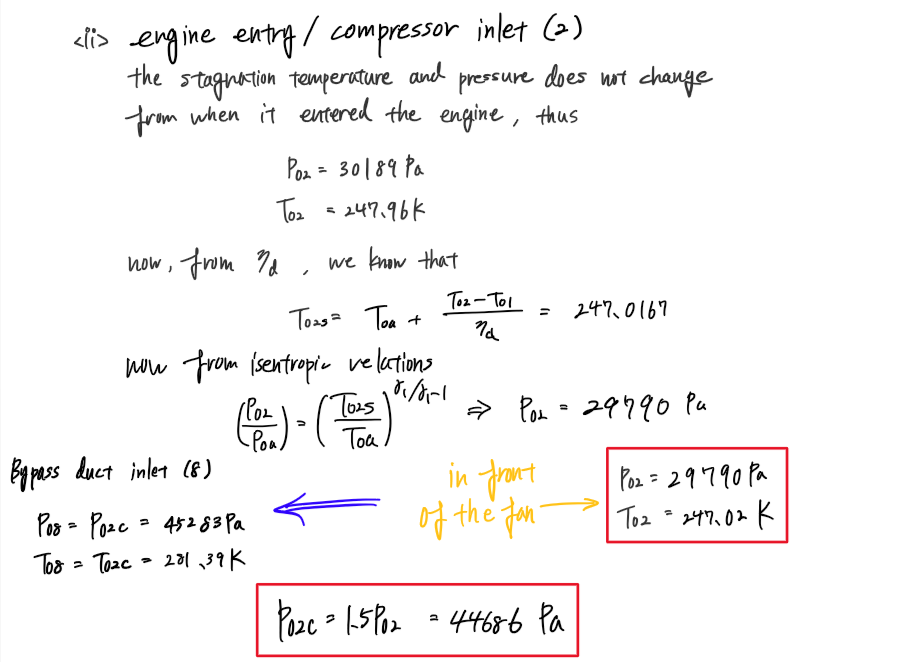


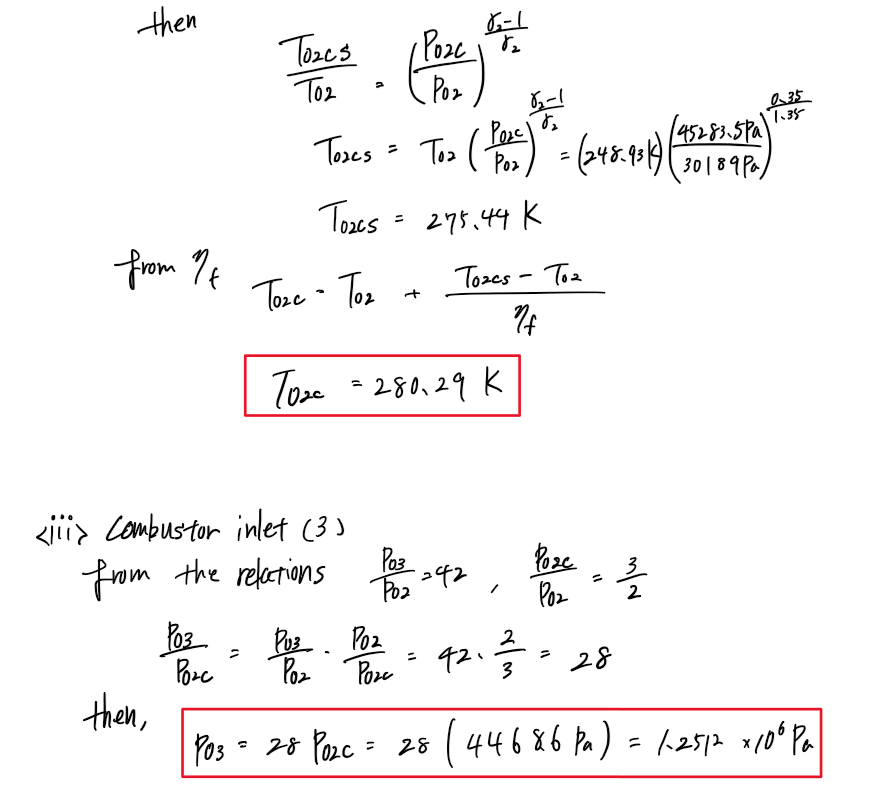


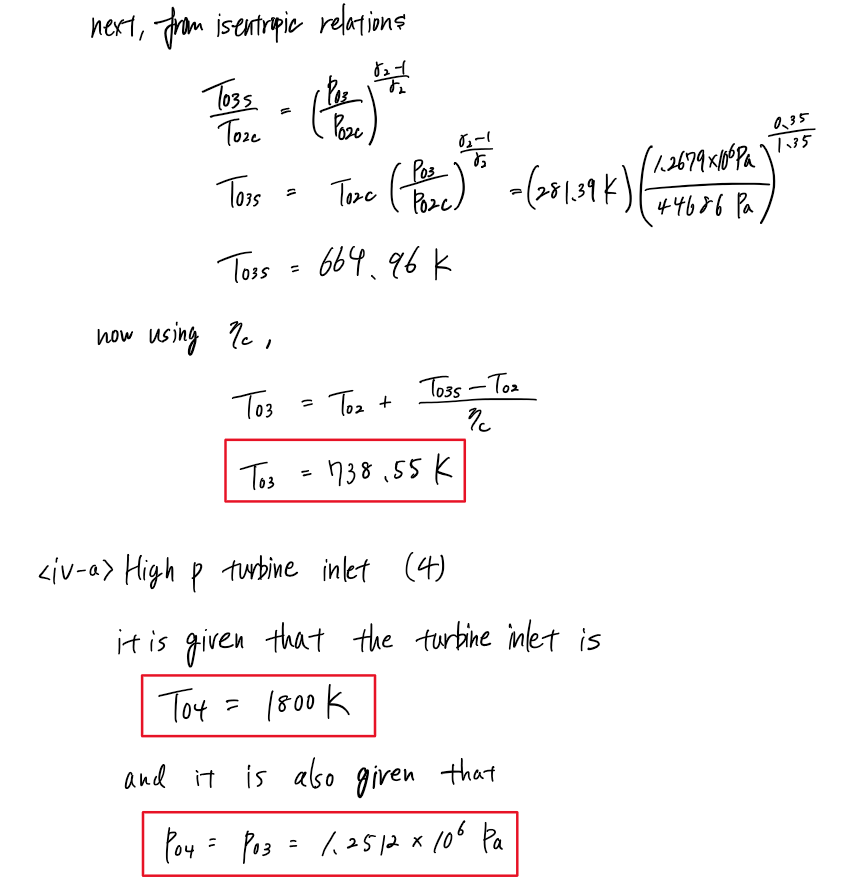


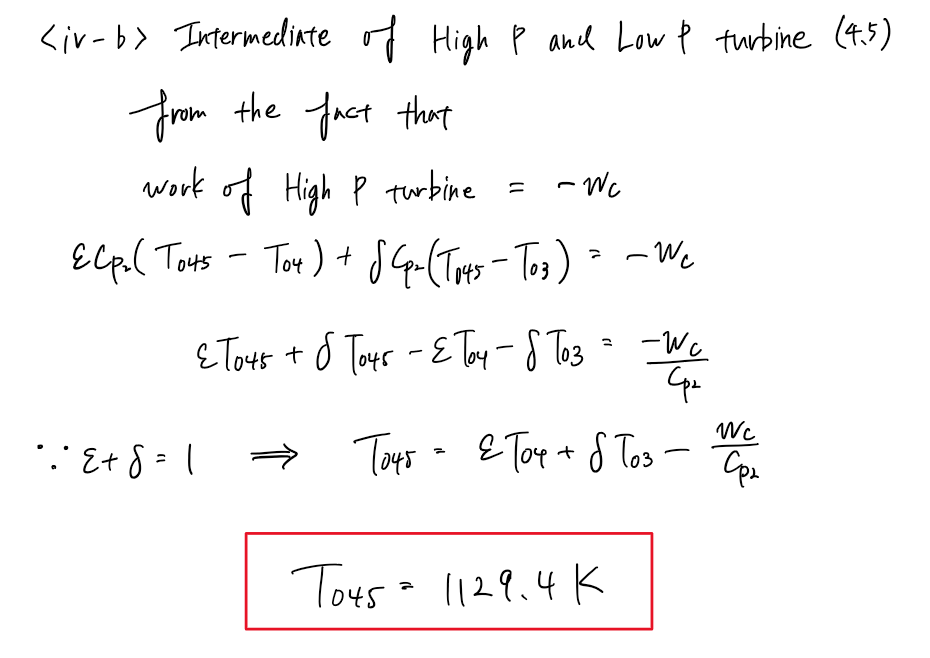


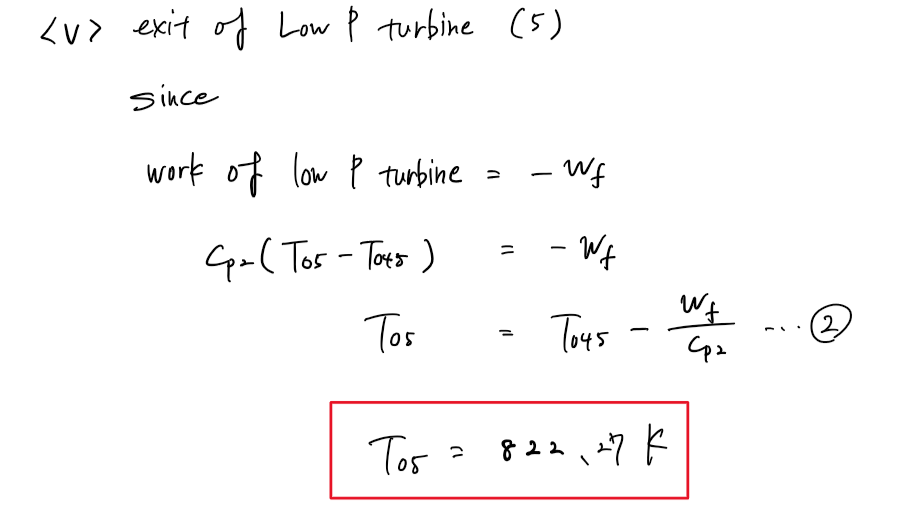


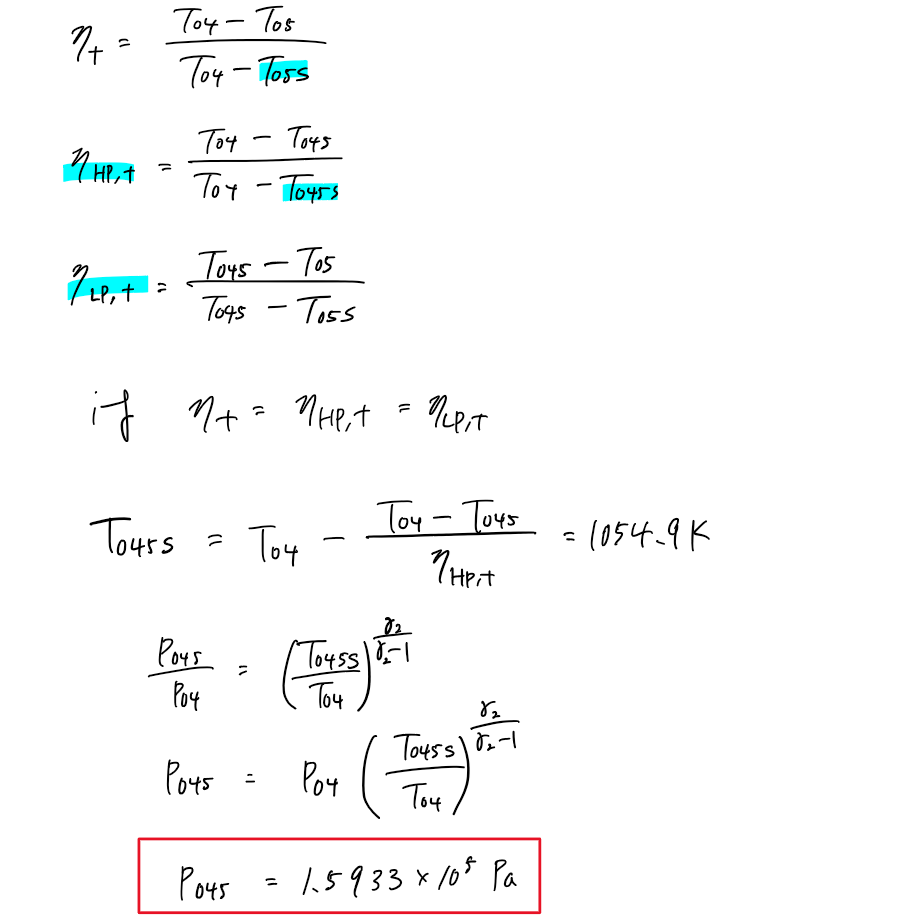


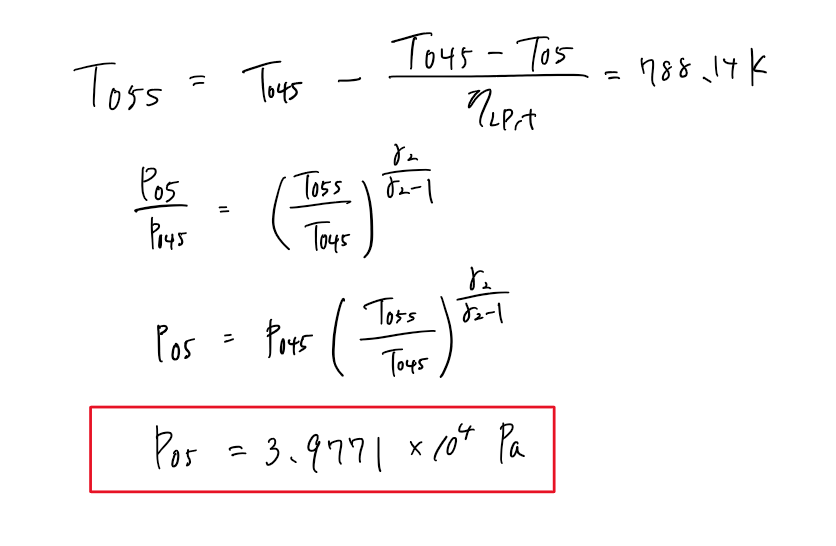


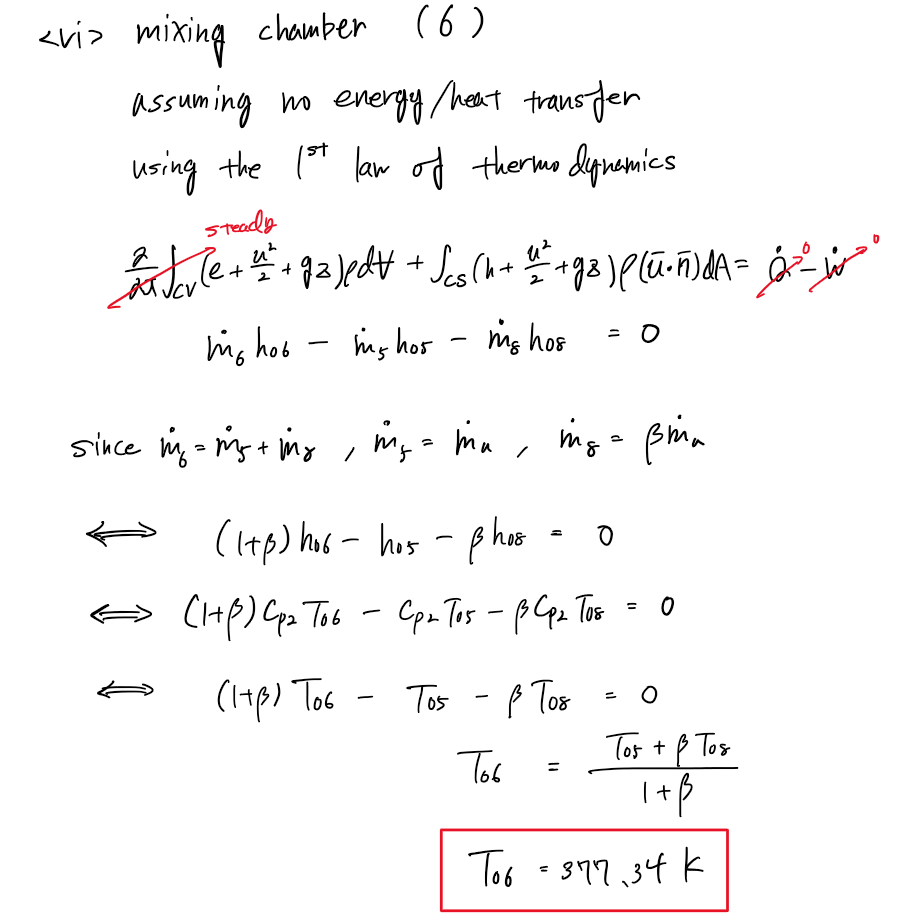


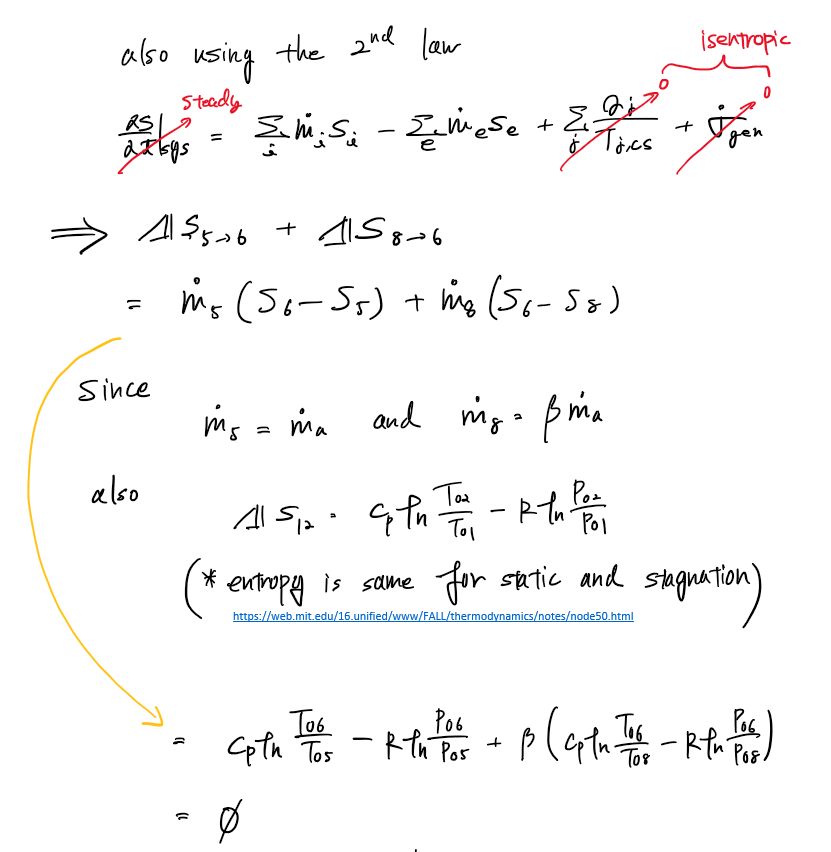


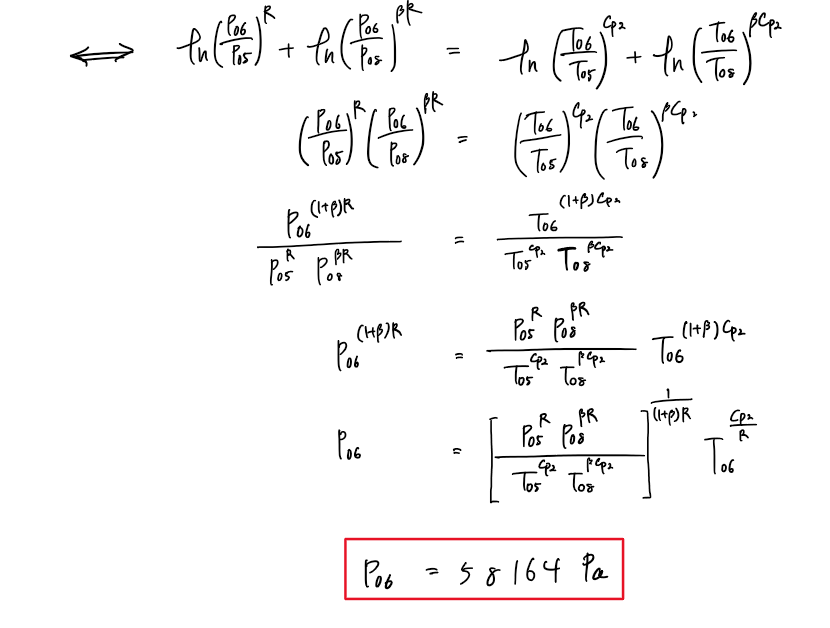


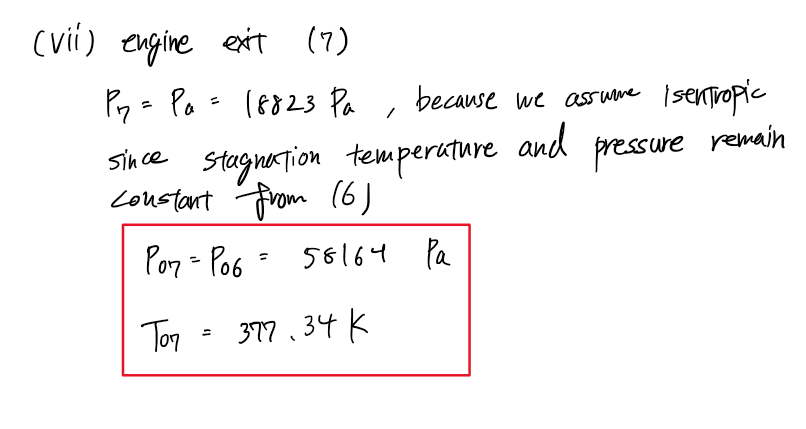


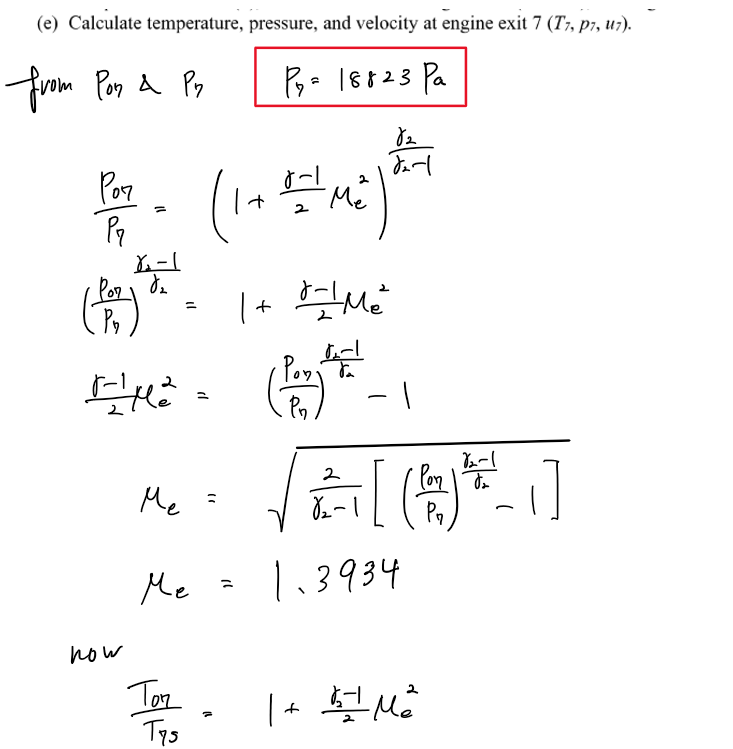


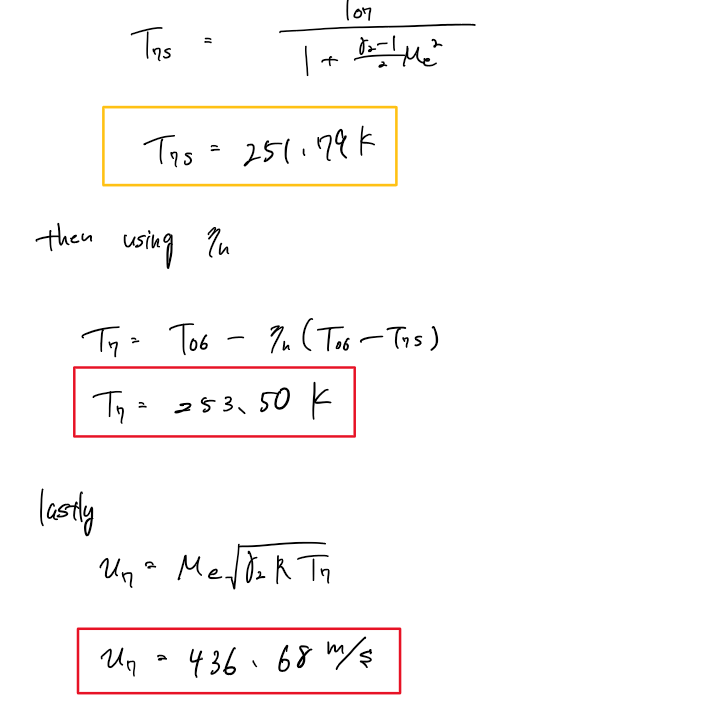


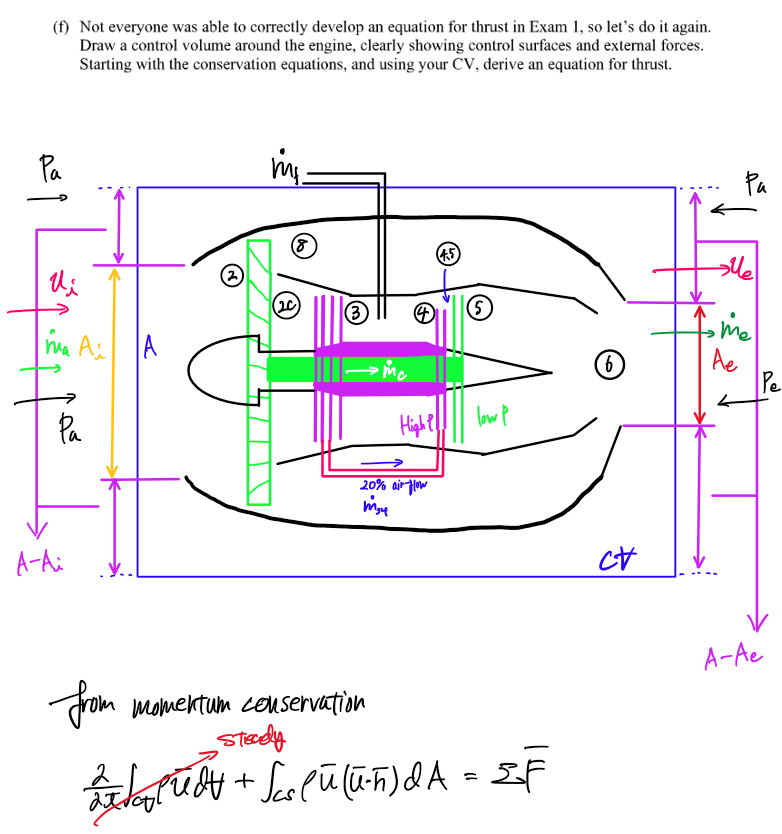


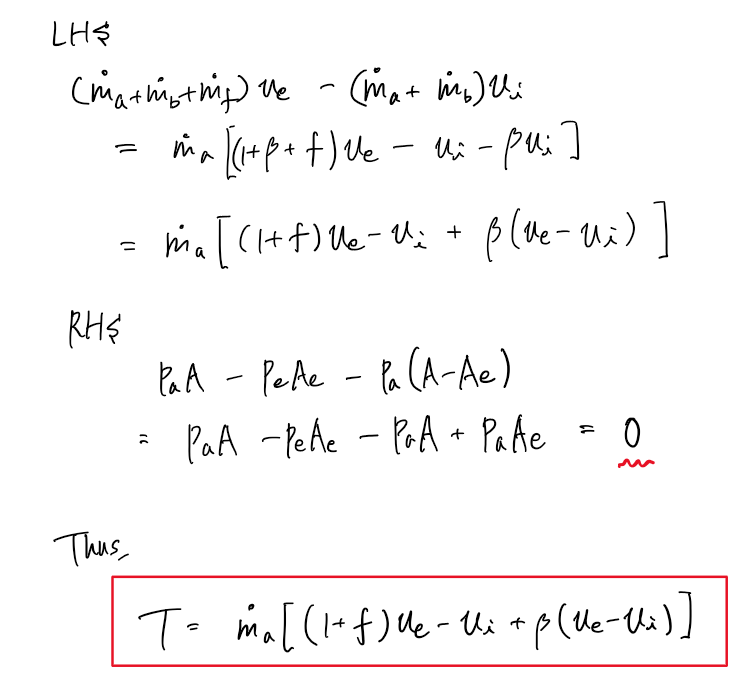


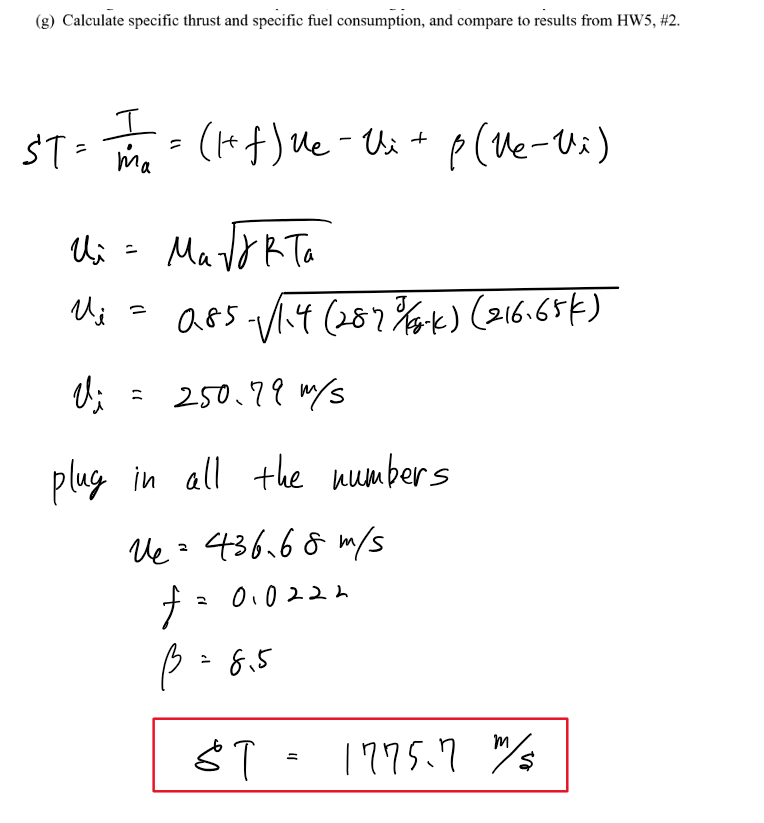


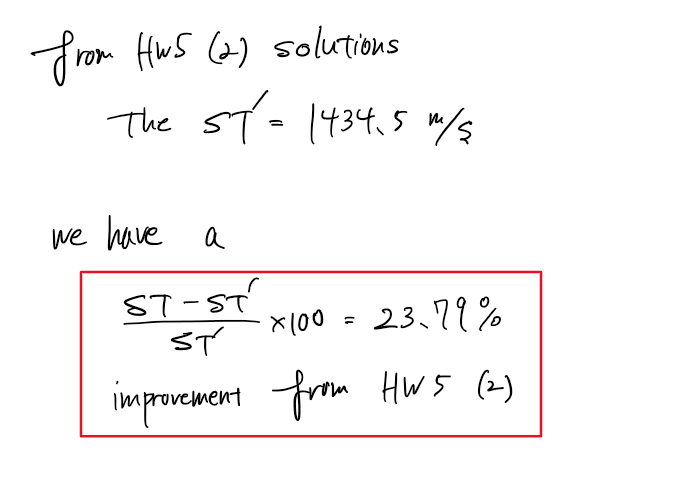












Appendix

AAE 339 HW6 MATLAB CODE

close all; clear all; clc

% Given properties

Ma = 0.85;

gamma1 = 1.4;

Cp1 = 1.0\*10^3; % [J/kg/K]

gamma2 = 1.35;

Cp2 = 1.10\*10^3; % [J/kg/K]

R = 287; % [J/kg/K]

Qr = 45000\*10^3; % [J/kg]

eta\_d = 0.97;

eta\_c = 0.85;

eta\_f = eta\_c;

eta\_t = 0.90;

eta\_n = 0.98;

eta\_b = 0.98;

beta = 8.5;

Po3\_Po2\_ratio = 42;

Po2c\_Po2\_ratio = 1.5;

To4 = 1800; % [K]

% freestream (a)

Ta = 216.65; % [K]

Pa = 18823; % [Pa]

Poa = p\_from\_M\_and\_gamma(Pa, Ma, gamma1, "stagnation")

Toa = T\_from\_M\_and\_gamma(Ta, Ma, gamma1, "stagnation")

% engine entry (2)

To2 = Toa

To2s = T\_from\_adiabatic\_eff\_stagnation(Ta, To2, eta\_d, "diffuser")

Po2 = P\_from\_isentropic\_relation(Poa, To2s, Toa, gamma1)

% compressor inlet (2c)

Po2c = Po2c\_Po2\_ratio\*Po2

To2cs = T\_from\_isentropic\_relation(To2, Po2c, Po2, gamma2)

To2c = T\_from\_adiabatic\_eff\_static(To2, To2cs, eta\_f, "fan")

% combustor inlet (3)

Po3\_Po2c\_ratio = Po3\_Po2\_ratio/Po2c\_Po2\_ratio

Po3 = Po3\_Po2c\_ratio\*Po2c

To3s = T\_from\_isentropic\_relation(To2c, Po3, Po2c, gamma2)

To3 = T\_from\_adiabatic\_eff\_static(To2, To3s, eta\_c, "compressor")

% combustor exit/High pressure turbine inlet (4)

To4 = 1800 % [K]

Po4 = Po3

% bypass duct inlet (8)

Po8 = Po2c

To8 = To2c

% calculate the work done by fan and compressor

Wf = (1 + beta) \* Cp2 \* (To8 - To2)

Wc = Cp2 \* (To3 - To2c)

% calculate the fuel-air-ratio

delta = 0.2;

epsilon = 0.8;

f = epsilon\*Cp2\*(To4 - To3)/(eta\_b\*Qr - Cp2\*To4)

disp(vpa(f, 7))

% intermediate of high pressure and low pressure turbines (4.5)

To45 = epsilon\*To4 + delta\*To3 - Wc/Cp2

eta\_HP\_t = eta\_t;

To45s = To4 - (To4 - To45)/eta\_HP\_t

Po45 = P\_from\_isentropic\_relation(Po4, To45s, To4, gamma2)

% exit of the low pressure turbine (5)

To5 = To45 - Wf/Cp2

eta\_LP\_t = eta\_t;

To5s = To45 - (To45 - To5)/eta\_LP\_t

Po5 = P\_from\_isentropic\_relation(Po45, To5s, To45, gamma2)

% mixing chamber (6)

Cp2\_alt = Cp2/1000;

R\_alt = R/1000;

To6 = (To5 + beta\*To8)/(1 + beta)

a1 = Po5^R\_alt;

a2 = Po8^(beta\*R\_alt);

a3 = To5^(Cp2\_alt);

a4 = To8^(beta\*Cp2\_alt);

b1 = To6^(Cp2\_alt/R\_alt);

A1 = (a1\*a2/a3/a4)^(1/(1 + beta)/R\_alt);

Po6 = A1\*b1

% Calculating properties at exit (7)

P7 = Pa

Po7 = Po6

To7 = To6

a1 = (Po7/Pa)^((gamma2 - 1)/gamma2) - 1;

Me = sqrt(2/(gamma2 - 1)\*a1)

T7s = To7/(1 + (gamma2 - 1)/2\*Me^2)

T7 = T\_from\_adiabatic\_eff\_static(To6, T7s, eta\_n, "nozzle")

u7 = Me \* sqrt(gamma2 \* R \* T7)

% calculate velocity at engine entry

ua = Ma\*sqrt(gamma1\*R\*Ta)

% Specfic thrust

ST = (1 + f)\*u7 - ua + beta\*(u7 - ua)

imp = (ST - 1434.5)/1434.5\*100

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 T2 = T\_from\_isentropic\_relation(T1, P2, P1, gamma)

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

end

function To2 = T\_from\_adiabatic\_eff\_static(To1, To2s, eta, type)

if type == "diffuser" || type == "compressor" || type == "fan"

To2 = To1 + (To2s - To1) / eta;

elseif type == "nozzle" || type == "turbine"

To2 = To1 - eta \* (To1 - To2s);

else

disp("Error, incorrect type. Accepted types are 'diffuser,' " + ...

"'compressor,' 'nozzle,' 'turbine,' or 'fan.'");

end

end

function To2s = T\_from\_adiabatic\_eff\_stagnation(To1, To2, eta, type)

if type == "diffuser" || type == "compressor" || type == "fan"

To2s = To1 + (To2 - To1) \* eta;

elseif type == "nozzle" || type == "turbine"

To2s = To1 - (To1 - To2) / eta;

else

disp("Error, incorrect type. Accepted types are 'diffuser,' " + ...

"'compressor,' 'nozzle,' 'turbine,' or 'fan.'");

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

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

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

end