**ME5309 Aircraft Engines and Rocket Propulsion**

**Assignment 2: Ideal and Non-Ideal Cycle Analysis**

**Part (a)**

Due to the engine is ideal, and with the altitude goes higher, the temperature will follow:

And do it in the MATLAB, the coding will be attached behind.

A graph with a red line and blue line

Description automatically generatedA graph with a red line

Description automatically generated

A graph with a blue line

Description automatically generatedA graph with red and blue lines

Description automatically generated

A graph with a blue line

Description automatically generated

**Part (b)**

For this case, at different altitude the maximum specific thrust happens in different height, so first of all is to fix the height and change the parameter to find out the best specific thrust performance at this altitude, and change the altitude to do the same over and over again. And the result is shown:

A graph with a red line and blue line

Description automatically generated

A graph with a blue line

Description automatically generatedA graph with red and blue lines

Description automatically generated

A graph with a blue line

Description automatically generatedA graph with red line and blue line

Description automatically generated

**Part (c)**

The turbojet is not ideal, so we need to do more calculations, and add them into the code.

A graph with a red line and blue line

Description automatically generatedA graph of a graph with a blue line

Description automatically generated





**Part (d)**

In this part, and have a variance of . So for .

1. For

A graph with different colored lines

Description automatically generated

1. For



From two charts above, we can conclude that the specific thrust is more sensitive to .

**Part (e)**

Fix the height to 10km:



**Appendix**

**Part(a)**

clc;

clear;

clear all;

%known parameters

H\_values = linspace(0,15000,1000);%%define the height

M\_0\_values = [0.85,2.1];

gamma = 1.4;

R = 287;

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

T\_t4 = 1900;

pi\_c = 15;

Qr = 43000000;

%need to be known

T\_0\_values = zeros(size(H\_values));

a\_0\_values = zeros(size(H\_values));

V\_0\_values = zeros(size(H\_values));

t\_r\_values = zeros(size(H\_values));

t\_lambda\_values = zeros(size(H\_values));

pi\_r\_values = zeros(size(H\_values));

t\_c\_values = zeros(size(H\_values));

eff\_c\_values = zeros(size(H\_values));

f\_values = zeros(size(H\_values));

t\_t\_values = zeros(size(H\_values));

pi\_t\_values = zeros(size(H\_values));

P\_t9\_9\_values = zeros(size(H\_values));

M\_9\_values = zeros(size(H\_values));

T\_9\_T\_0\_values = zeros(size(H\_values));

V\_9\_a\_0\_values = zeros(size(H\_values));

thrust\_values = zeros(length(M\_0\_values),length(H\_values));

TSFC\_values = zeros(length(M\_0\_values),length(H\_values));

eff\_p\_values = zeros(length(M\_0\_values),length(H\_values));

eff\_th\_values = zeros(length(M\_0\_values),length(H\_values));

eff\_o\_values = zeros(length(M\_0\_values),length(H\_values));

%definr the T\_0

for i = 1:length(H\_values)

H = H\_values(i);

if H < 11000

T\_0\_values(i) = 288.15-0.0065\*H;

else

T\_0\_values(i) = 216.65;

end

for j=1:length(M\_0\_values)

M\_0 = M\_0\_values(j);

a\_0\_values(i) = sqrt(gamma \* R \* T\_0\_values(i));

V\_0\_values(i) = a\_0\_values(i) \* M\_0;

t\_r\_values(i) = 1+(gamma-1)/2 \* M\_0^2;

pi\_r\_values(i) = t\_r\_values(i)^(gamma/(gamma - 1));

t\_lambda\_values(i) = T\_t4 / T\_0\_values(i);

t\_c\_values(i) = pi\_c^((gamma-1)/gamma);

f\_values(i) = (t\_lambda\_values(i)-t\_r\_values(i) \* t\_c\_values(i)) ./ (Qr/(C\_p \* T\_0\_values(i))-t\_lambda\_values(i));

t\_t\_values(i) = 1-t\_r\_values(i)\*(t\_c\_values(i)-1)/t\_lambda\_values(i);

pi\_t\_values(i) = t\_t\_values(i)^(gamma/(gamma-1));

P\_t9\_9\_values(i) = pi\_c \* pi\_r\_values(i) \* pi\_t\_values(i);

M\_9\_values(i) = sqrt(2/(gamma-1)\*(P\_t9\_9\_values(i)^((gamma-1)/gamma)-1));

T\_9\_T\_0\_values(i) = t\_lambda\_values(i) \* t\_t\_values(i) / (P\_t9\_9\_values(i)^((gamma-1)/gamma));

V\_9\_a\_0\_values(i) = M\_9\_values(i)\*sqrt(T\_9\_T\_0\_values(i));

thrust\_values(j,i) = a\_0\_values(i)\* (V\_9\_a\_0\_values(i)- M\_0);

TSFC\_values(j,i) = f\_values(i)/thrust\_values(j,i);

eff\_p\_values(j,i) = 2 \* V\_0\_values(i) \* thrust\_values(j,i) / (a\_0\_values(i)^2 \* (V\_9\_a\_0\_values(i)^2 - M\_0^2));

eff\_th\_values(j,i) = (a\_0\_values(i)^2 \* (V\_9\_a\_0\_values(i)^2 - M\_0^2)) / (2 \* f\_values(i) \* Qr);

eff\_o\_values(j,i) = eff\_p\_values(j,i) \* eff\_th\_values(j,i);

end

end

figure;

plot(H\_values,thrust\_values(1,:),'r','LineWidth',2);

title('Variation of Thrust with Height');

xlabel('Height');

ylabel('Thrust');

hold on

plot(H\_values,thrust\_values(2,:),'b','LineWidth',2);

legend('M=0.85','M=2.1');

grid on;

figure;

plot(H\_values,TSFC\_values(1,:),'r','LineWidth',2);

title('Variation of TSFC with Height');

xlabel('Height');

ylabel('TSFC');

hold on

plot(H\_values,TSFC\_values(2,:),'b','LineWidth',2);

legend('M=0.85','M=2.1');

grid on;

figure;

plot(H\_values,eff\_p\_values(1,:),'r','LineWidth',2);

title('Variation of efficiency of propulsion with Height');

xlabel('Height');

ylabel('efficiency of propulsion');

hold on

plot(H\_values,eff\_p\_values(2,:),'b','LineWidth',2);

legend('M=0.85','M=2.1');

grid on;

figure;

plot(H\_values,eff\_th\_values(1,:),'r','LineWidth',2);

title('Variation of efficiency of thermal energy with Height');

xlabel('Height');

ylabel('efficiency of th');

hold on

plot(H\_values,eff\_th\_values(2,:),'b','LineWidth',2);

legend('M=0.85','M=2.1');

grid on;

figure;

plot(H\_values,eff\_o\_values(1,:),'r','LineWidth',2);

title('Variation of total efficiency with Height');

xlabel('Height');

ylabel('total efficiency');

hold on

plot(H\_values,eff\_o\_values(2,:),'b','LineWidth',2);

legend('M=0.85','M=2.1');

grid on;

**Part(b)**

clc;

clear;

clear all;

%known parameters

H\_values = linspace(0,15000,1000);%%define the height

M\_0\_values = [0.85 2.1];

gamma = 1.4;

R = 287;

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

T\_t4 = 1900;

Qr = 43000000;

%need to be known

T\_0\_values = zeros(size(H\_values));

a\_0\_values = zeros(size(H\_values));

V\_0\_values = zeros(size(H\_values));

t\_r\_values = zeros(size(H\_values));

t\_lambda\_values = zeros(size(H\_values));

pi\_r\_values = zeros(size(H\_values));

t\_c\_values = zeros(size(H\_values));

pi\_c\_values = zeros(length(M\_0\_values),length(H\_values));

f\_values = zeros(size(H\_values));

t\_t\_values = zeros(size(H\_values));

pi\_t\_values = zeros(size(H\_values));

P\_t9\_9\_values = zeros(size(H\_values));

M\_9\_values = zeros(size(H\_values));

T\_9\_T\_0\_values = zeros(size(H\_values));

V\_9\_a\_0\_values = zeros(size(H\_values));

thrust\_values = zeros(length(M\_0\_values),length(H\_values));

TSFC\_values = zeros(length(M\_0\_values),length(H\_values));

eff\_p\_values = zeros(length(M\_0\_values),length(H\_values));

eff\_th\_values = zeros(length(M\_0\_values),length(H\_values));

eff\_o\_values = zeros(length(M\_0\_values),length(H\_values));

%definr the T\_0

for j = 1:length(M\_0\_values) % 遍历不同的 M\_0

M\_0 = M\_0\_values(j); % 当前 M\_0

for i = 1:length(H\_values)

H = H\_values(i);

if H < 11000

T\_0\_values(i) = 288.15-0.0065\*H;

else

T\_0\_values(i) = 216.65;

end

a\_0\_values(i) = sqrt(gamma \* R .\* T\_0\_values(i));

V\_0\_values(i) = a\_0\_values(i) \* M\_0;

t\_r\_values(i) = 1+(gamma-1)/2 \* M\_0^2;

pi\_r\_values(i) = t\_r\_values(i)^(gamma/(gamma - 1));

t\_lambda\_values(i) = T\_t4 / T\_0\_values(i);

t\_c\_values(i) = sqrt(t\_lambda\_values(i)) / t\_r\_values(i);

pi\_c\_values(j,i) = t\_c\_values(i)^(gamma/(gamma-1));

f\_values(i) = (t\_lambda\_values(i)-t\_r\_values(i) \* t\_c\_values(i)) ./ (Qr/(C\_p \* T\_0\_values(i))-t\_lambda\_values(i));

t\_t\_values(i) = 1-t\_r\_values(i)\*(t\_c\_values(i)-1)/t\_lambda\_values(i);

pi\_t\_values(i) = t\_t\_values(i)^(gamma/(gamma-1));

P\_t9\_9\_values(i) = pi\_c\_values(j,i) \* pi\_r\_values(i) \* pi\_t\_values(i);

M\_9\_values(i) = sqrt(2/(gamma-1)\*(P\_t9\_9\_values(i)^((gamma-1)/gamma)-1));

T\_9\_T\_0\_values(i) = t\_lambda\_values(i) \* t\_t\_values(i) / (P\_t9\_9\_values(i)^((gamma-1)/gamma));

V\_9\_a\_0\_values(i) = M\_9\_values(i)\*sqrt(T\_9\_T\_0\_values(i));

thrust\_values(j,i) = a\_0\_values(i)\* (V\_9\_a\_0\_values(i)- M\_0);

TSFC\_values(j,i) = f\_values(i)/thrust\_values(j,i);

eff\_p\_values(j,i) = 2 \* V\_0\_values(i) \* thrust\_values(j,i) / (a\_0\_values(i)^2 \* (V\_9\_a\_0\_values(i)^2 - M\_0^2));

eff\_th\_values(j,i) = (a\_0\_values(i)^2 \* (V\_9\_a\_0\_values(i)^2 - M\_0^2)) / (2 \* f\_values(i) \* Qr);

eff\_o\_values(j,i) = eff\_p\_values(j,i) \* eff\_th\_values(j,i);

end

end

figure;

plot(H\_values, pi\_c\_values(1,:), 'b', 'LineWidth', 2);

hold on;

title('Optimal \pi\_c for Maximum Thrust vs Altitude');

xlabel('Altitude (m)');

ylabel('Optimal \pi\_c');

plot(H\_values, pi\_c\_values(2,:), 'r', 'LineWidth', 2);

legend('M\_0 = 0.85', 'M\_0 = 2.1');

grid on;

hold off

figure;

plot(H\_values,thrust\_values(1,:),'r','LineWidth',2);

title('Variation of Thrust with Height');

xlabel('Height');

ylabel('Thrust');

hold on

plot(H\_values,thrust\_values(2,:),'b','LineWidth',2);

legend('M=0.85','M=2.1');

grid on;

figure;

plot(H\_values,TSFC\_values(1,:),'r','LineWidth',2);

title('Variation of TSFC with Height');

xlabel('Height');

ylabel('TSFC');

hold on

plot(H\_values,TSFC\_values(2,:),'b','LineWidth',2);

legend('M=0.85','M=2.1');

grid on;

figure;

plot(H\_values,eff\_p\_values(1,:),'r','LineWidth',2);

title('Variation of efficiency of propulsion with Height');

xlabel('Height');

ylabel('efficiency of propulsion');

hold on

plot(H\_values,eff\_p\_values(2,:),'b','LineWidth',2);

legend('M=0.85','M=2.1');

grid on;

figure;

plot(H\_values,eff\_th\_values(1,:),'r','LineWidth',2);

title('Variation of efficiency of thermal energy with Height');

xlabel('Height');

ylabel('/eta\_t\_h');

hold on

plot(H\_values,eff\_th\_values(2,:),'b','LineWidth',1.5);

legend('M=0.85','M=2.1');

grid on;

figure;

plot(H\_values,eff\_o\_values(1,:),'r','LineWidth',2);

title('Variation of total efficiency with Height');

xlabel('Height');

ylabel('total efficiency');

hold on

plot(H\_values,eff\_o\_values(2,:),'b','LineWidth',2);

legend('M=0.85','M=2.1');

grid on;

**Part(c)**

%%

clc;

clear;

clear all;

%known parameters

H\_values = linspace(0,15000,1000);%%define the height

M\_0\_values = [0.85 2.1];

T\_t4 = 1900;

pi\_c = 15;

Qr = 43000000;

pi\_d = 0.98;

e\_c = 0.85;

pi\_b = 0.95;

eta\_b = 0.98;

e\_t = 0.85;

eta\_m = 0.98;

pi\_n = 0.97;

gamma\_c = 1.4;

C\_pc = 1004;

gamma\_t = 1.33;

C\_pt = 1156;

R\_c = (gamma\_c-1)/gamma\_c \* C\_pc;

R\_t = (gamma\_t-1)/gamma\_t \* C\_pt;

%need to be known

T\_0\_values = zeros(size(H\_values));

a\_0\_values = zeros(size(H\_values));

V\_0\_values = zeros(size(H\_values));

t\_r\_values = zeros(size(H\_values));

t\_lambda\_values = zeros(size(H\_values));

pi\_r\_values = zeros(size(H\_values));

t\_c\_values = zeros(size(H\_values));

eff\_c\_values = zeros(size(H\_values));

f\_values = zeros(size(H\_values));

t\_t\_values = zeros(size(H\_values));

pi\_t\_values = zeros(size(H\_values));

eta\_t\_values = zeros(size(H\_values));

P\_t9\_9\_values = zeros(size(H\_values));

M\_9\_values = zeros(size(H\_values));

T\_9\_T\_0\_values = zeros(size(H\_values));

V\_9\_a\_0\_values = zeros(size(H\_values));

thrust\_values = zeros(size(H\_values));

TSFC\_values = zeros(size(H\_values));

eff\_p\_values = zeros(size(H\_values));

eff\_th\_values = zeros(size(H\_values));

eff\_o\_values = zeros(size(H\_values));

%definr the T\_0

for j = 1:length(M\_0\_values) % different M\_0

M\_0 = M\_0\_values(j);

for i = 1:length(H\_values)

H = H\_values(i);

if H < 11000

T\_0\_values(i) = 288.15-0.0065\*H;

else

T\_0\_values(i) = 216.65;

end

a\_0\_values(i) = sqrt(gamma\_c \* R\_c \* T\_0\_values(i));

V\_0\_values(i) = a\_0\_values(i) \* M\_0;

t\_r\_values(i) = 1+(gamma\_c-1)/2 \* M\_0^2;

pi\_r\_values(i) = t\_r\_values(i)^(gamma\_c/(gamma\_c - 1));

t\_lambda\_values(i) = T\_t4 \* C\_pt / (T\_0\_values(i) \* C\_pc);

t\_c\_values(i) = pi\_c^((gamma\_c - 1)/(gamma\_c \* e\_c));

eff\_c\_values(i) = (pi\_c^((gamma\_c - 1)/gamma\_c)-1)/(t\_c\_values(i) - 1);

f\_values(i) = (t\_lambda\_values(i)-t\_r\_values(i) \* t\_c\_values(i)) / ((eta\_b \* Qr)/(C\_pc \* T\_0\_values(i))-t\_lambda\_values(i));

t\_t\_values(i) = 1 - (t\_r\_values(i)\*(t\_c\_values(i)-1)/t\_lambda\_values(i))/(eta\_m \* (1 + f\_values(i)));

pi\_t\_values(i) = t\_t\_values(i)^(gamma\_t /((gamma\_t-1)\*e\_t));

eta\_t\_values(i) = (1-t\_t\_values(i)) / (1 - t\_t\_values(i)^(-e\_t));

P\_t9\_9\_values(i) = pi\_c \* pi\_r\_values(i) \* pi\_t\_values(i) \* pi\_n \* pi\_b \* pi\_d;

M\_9\_values(i) = sqrt(2/(gamma\_t - 1)\*(P\_t9\_9\_values(i)^((gamma\_t - 1)/gamma\_t)-1));

T\_9\_T\_0\_values(i) = t\_lambda\_values(i) \* t\_t\_values(i) / P\_t9\_9\_values(i)^((gamma\_t-1)/gamma\_t)\*C\_pc/C\_pt;

V\_9\_a\_0\_values(i) = M\_9\_values(i)\*sqrt(T\_9\_T\_0\_values(i) \* R\_t/R\_c \* gamma\_t/gamma\_c);

thrust\_values(j,i) = a\_0\_values(i)\* ((1 + f\_values(i)) \* V\_9\_a\_0\_values(i)- M\_0);

TSFC\_values(j,i) = f\_values(i)/thrust\_values(j,i);

eff\_p\_values(j,i) = 2 \* V\_0\_values(i) \* thrust\_values(j,i) / (a\_0\_values(i)^2 \* ((1 + f\_values(i)) \* V\_9\_a\_0\_values(i)^2 - M\_0^2));

eff\_th\_values(j,i) = a\_0\_values(i)^2 \* ((1 + f\_values(i)) \* (V\_9\_a\_0\_values(i)^2 - M\_0^2)) / (2 \* f\_values(i) \* Qr);

eff\_o\_values(j,i) = eff\_p\_values(j,i) \* eff\_th\_values(j,i);

end

end

figure;

plot(H\_values, thrust\_values(1,:), 'r', 'LineWidth', 2);

hold on;

plot(H\_values, thrust\_values(2,:), 'b', 'LineWidth', 2);

title('Thrust vs Altitude');

xlabel('Altitude (m)');

ylabel('Specific Thrust');

legend('M\_0 = 0.85', 'M\_0 = 2.1');

grid on;

hold off

figure;

plot(H\_values, TSFC\_values(1,:), 'r', 'LineWidth', 2);

hold on;

plot(H\_values, TSFC\_values(2,:), 'b', 'LineWidth', 2);

title('TSFC vs Altitude');

xlabel('Altitude (m)');

ylabel('TSFC');

legend('M\_0 = 0.85', 'M\_0 = 2.1');

grid on;

hold off

figure;

plot(H\_values, eff\_p\_values(1,:), 'r', 'LineWidth', 2);

hold on;

plot(H\_values, eff\_p\_values(2,:), 'b', 'LineWidth', 2);

title('\eta\_p vs Altitude');

xlabel('Altitude (m)');

ylabel('\eta\_p');

legend('M\_0 = 0.85', 'M\_0 = 2.1');

grid on;

hold off

figure;

plot(H\_values, eff\_th\_values(1,:), 'r', 'LineWidth', 2);

hold on;

plot(H\_values, eff\_th\_values(2,:), 'b', 'LineWidth', 2);

title('\eta\_t\_h vs Altitude');

xlabel('Altitude (m)');

ylabel('\eta\_t\_h');

legend('M\_0 = 0.85', 'M\_0 = 2.1');

grid on;

hold off

figure;

plot(H\_values, eff\_o\_values(1,:), 'r', 'LineWidth', 2);

hold on;

plot(H\_values, eff\_o\_values(2,:), 'b', 'LineWidth', 2);

title('\eta\_o vs Altitude');

xlabel('Altitude (m)');

ylabel('\eta\_o');

legend('M\_0 = 0.85', 'M\_0 = 2.1');

grid on;

hold off

**part (d)**

%%

clc;

clear;

clear all;

% 已知参数

H\_values = linspace(0, 15000, 1000); % 定义高度范围

M\_0\_values = [0.85, 2.1]; % 飞行马赫数

T\_t4 = 1900;

pi\_c = 15;

Qr = 43000000;

pi\_d = 0.98;

e\_c = 0.85; % e\_c 取10个值

pi\_b = 0.95;

eta\_b = 0.98;

e\_t\_values = linspace(0.833, 0.867, 10);

eta\_m = 0.98;

pi\_n = 0.97;

gamma\_c = 1.4;

C\_pc = 1004;

gamma\_t = 1.33;

C\_pt = 1156;

R\_c = (gamma\_c - 1) / gamma\_c \* C\_pc;

R\_t = (gamma\_t - 1) / gamma\_t \* C\_pt;

% 初始化存储数组

thrust\_values = zeros(length(e\_t\_values), length(H\_values), length(M\_0\_values));

% 遍历 M\_0

figure;

hold on;

colors = lines(10); % 生成 10 种不同颜色

for k = 1:length(M\_0\_values)

M\_0 = M\_0\_values(k);

% 遍历 e\_c

for j = 1:length(e\_t\_values)

e\_t = e\_t\_values(j);

% 遍历高度 H

for i = 1:length(H\_values)

H = H\_values(i);

% 计算环境温度

if H < 11000

T\_0 = 288.15 - 0.0065 \* H;

else

T\_0 = 216.65;

end

% 计算气动力参数

a\_0 = sqrt(gamma\_c \* R\_c \* T\_0);

V\_0 = a\_0 \* M\_0;

t\_r = 1 + (gamma\_c - 1) / 2 \* M\_0^2;

pi\_r = t\_r^(gamma\_c / (gamma\_c - 1));

t\_lambda = T\_t4 \* C\_pt / (T\_0 \* C\_pc);

t\_c = pi\_c^((gamma\_c - 1) / (gamma\_c \* e\_c));

eff\_c = (pi\_c^((gamma\_c - 1) / gamma\_c) - 1) / (t\_c - 1);

f = (t\_lambda - t\_r \* t\_c) / ((eta\_b \* Qr) / (C\_pc \* T\_0) - t\_lambda);

t\_t = 1 - (t\_r \* (t\_c - 1) / t\_lambda) / (eta\_m \* (1 + f));

pi\_t = t\_t^(gamma\_t / ((gamma\_t - 1) \* e\_t));

eta\_t = (1 - t\_t) / (1 - t\_t^(-e\_t));

P\_t9\_9 = pi\_c \* pi\_r \* pi\_t \* pi\_n \* pi\_b \* pi\_d;

M\_9 = sqrt(2 / (gamma\_t - 1) \* (P\_t9\_9^((gamma\_t - 1) / gamma\_t) - 1));

T\_9\_T\_0 = t\_lambda \* t\_t / P\_t9\_9^((gamma\_t - 1) / gamma\_t) \* C\_pc / C\_pt;

V\_9\_a\_0 = M\_9 \* sqrt(T\_9\_T\_0 \* R\_t / R\_c \* gamma\_t / gamma\_c);

% 计算推力

thrust\_values(j, i, k) = a\_0 \* ((1 + f) \* V\_9\_a\_0 - M\_0);

end

% 画出推力随高度变化的曲线

if k == 1

plot(H\_values, squeeze(thrust\_values(j, :, k)), 'Color', colors(j, :), 'LineWidth', 1.5);

else

plot(H\_values, squeeze(thrust\_values(j, :, k)), '--', 'Color', colors(j, :), 'LineWidth', 1.5);

end

end

end

% 设置图例、标签和标题

title('Thrust vs Altitude for Different e\_t and M\_0');

xlabel('Altitude (m)');

ylabel('Thrust (N)');

legend\_labels = cell(1, 20);

for j = 1:10

legend\_labels{j} = sprintf('M\_0=0.85, e\_t=%.3f', e\_t\_values(j));

legend\_labels{j + 10} = sprintf('M\_0=2.1, e\_t=%.3f', e\_t\_values(j));

end

legend(legend\_labels, 'Location', 'northeastoutside');

grid on;

hold off;

**part(e)**

%%

clc;

clear;

clear all;

% Given parameters

H = 10000; % Altitude in meters

M\_0\_values = [0.85, 2.1]; % Mach numbers

T\_t4 = 1900; % Turbine inlet temperature in Kelvin

pi\_c\_values = linspace(1, 100, 100); % Range of compression ratios

Qr = 43000000; % Fuel heating value in J/kg

pi\_d = 0.98;

e\_c = 0.85;

pi\_b = 0.95;

eta\_b = 0.98;

e\_t = 0.85;

eta\_m = 0.98;

pi\_n = 0.97;

gamma\_c = 1.4;

C\_pc = 1004;

gamma\_t = 1.33;

C\_pt = 1156;

R\_c = (gamma\_c - 1) / gamma\_c \* C\_pc;

R\_t = (gamma\_t - 1) / gamma\_t \* C\_pt;

% Calculate ambient temperature T\_0

if H < 11000

T\_0 = 288.15 - 0.0065 \* H;

else

T\_0 = 216.65;

end

% Preallocate arrays for thrust

thrust\_values = zeros(length(M\_0\_values), length(pi\_c\_values));

% Create figure

figure;

hold on;

colors = lines(length(M\_0\_values)); % Generate different colors for the curves

for j = 1:length(M\_0\_values)

M\_0 = M\_0\_values(j);

% Compute initial parameters

a\_0 = sqrt(gamma\_c \* R\_c \* T\_0);

V\_0 = a\_0 \* M\_0;

t\_r = 1 + (gamma\_c - 1) / 2 \* M\_0^2;

pi\_r = t\_r^(gamma\_c / (gamma\_c - 1));

for i = 1:length(pi\_c\_values)

pi\_c = pi\_c\_values(i);

t\_lambda = T\_t4 \* C\_pt / (T\_0 \* C\_pc);

t\_c = pi\_c^((gamma\_c - 1) / (gamma\_c \* e\_c));

eff\_c = (pi\_c^((gamma\_c - 1) / gamma\_c) - 1) / (t\_c - 1);

f = (t\_lambda - t\_r \* t\_c) / (((eta\_b \* Qr) / (C\_pc \* T\_0)) - t\_lambda);

t\_t = 1 - (t\_r \* (t\_c - 1) / t\_lambda) / (eta\_m \* (1 + f));

pi\_t = t\_t^(gamma\_t / ((gamma\_t - 1) \* e\_t));

eta\_t = (1 - t\_t) / (1 - t\_t^(-e\_t));

P\_t9\_9 = pi\_c \* pi\_r \* pi\_t \* pi\_n \* pi\_b \* pi\_d;

M\_9 = sqrt(2 / (gamma\_t - 1) \* (P\_t9\_9^((gamma\_t - 1) / gamma\_t) - 1));

T\_9\_T\_0 = t\_lambda \* t\_t / P\_t9\_9^((gamma\_t - 1) / gamma\_t) \* C\_pc / C\_pt;

V\_9\_a\_0 = M\_9 \* sqrt(T\_9\_T\_0 \* R\_t / R\_c \* gamma\_t / gamma\_c);

% Compute thrust

thrust\_values(j, i) = a\_0 \* ((1 + f) \* V\_9\_a\_0 - M\_0);

end

% Plot thrust vs pi\_c

plot(pi\_c\_values, thrust\_values(j, :), 'Color', colors(j, :), 'LineWidth', 1.5, ...

'DisplayName', sprintf('M\_0 = %.2f', M\_0));

% Find and mark the maximum thrust point

[max\_thrust, max\_idx] = max(thrust\_values(j, :));

best\_pi\_c = pi\_c\_values(max\_idx);

% Highlight the maximum thrust point

plot(best\_pi\_c, max\_thrust, 'o', 'Color', colors(j, :), 'MarkerSize', 8, 'MarkerFaceColor', colors(j, :));

text(best\_pi\_c, max\_thrust, sprintf('\\leftarrow Max Thrust: %.1f N (\\pi\_c = %.1f)', max\_thrust, best\_pi\_c), ...

'Color', colors(j, :), 'FontSize', 10);

end

% Labels and title in English

xlabel('\pi\_c (Compression Ratio)', 'FontSize', 12);

ylabel('Thrust (N)', 'FontSize', 12);

title('Thrust vs Compression Ratio for Different Mach Numbers', 'FontSize', 14);

legend('Location', 'best');

grid on;

hold off;