ME5309 Aircraft Engines and Rocket Propulsion

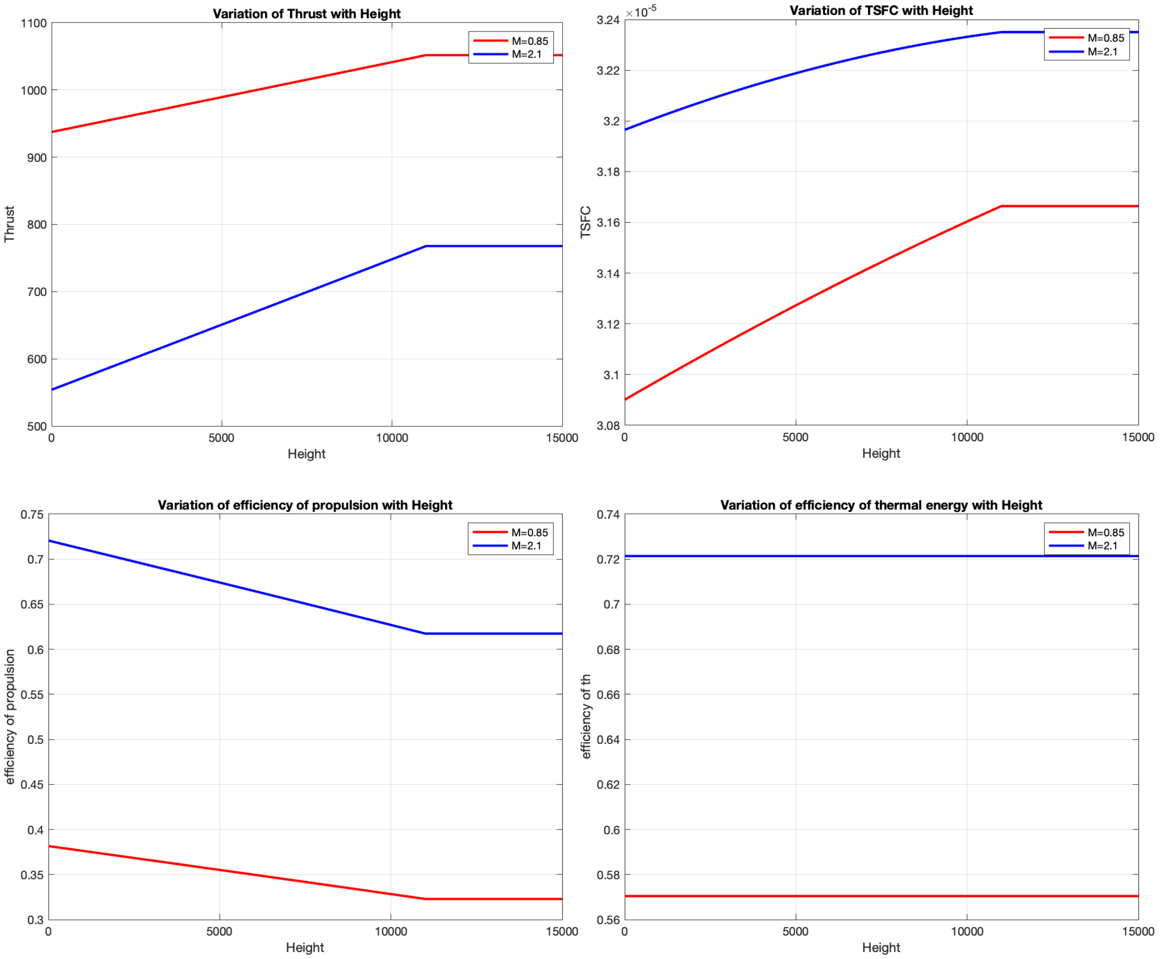
Assignment 2: Ideal and Non-Ideal Cycle Analysis

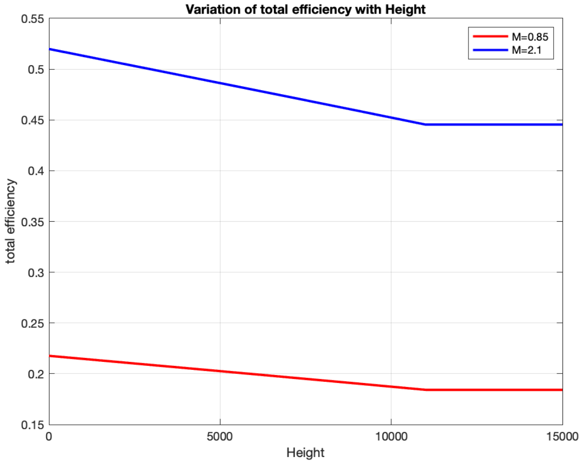
Part (a)

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

$$T = \begin{cases} 288.15 - 0.0065H \ (H < 11000m) \\ 216.65K \ (11000 \leq H \leq 20100m) \end{cases}$$

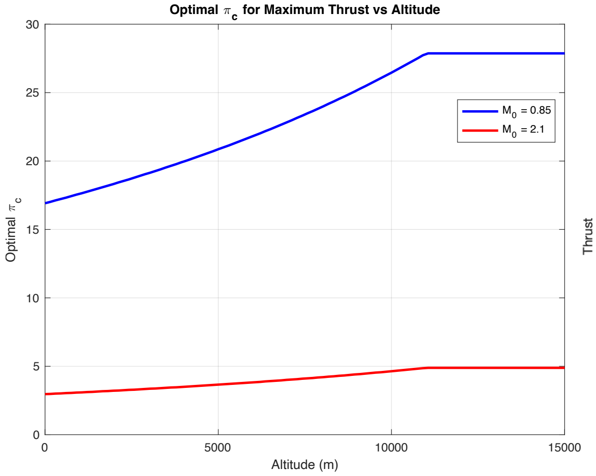
And do it in the MATLAB, the coding will be attached behind.

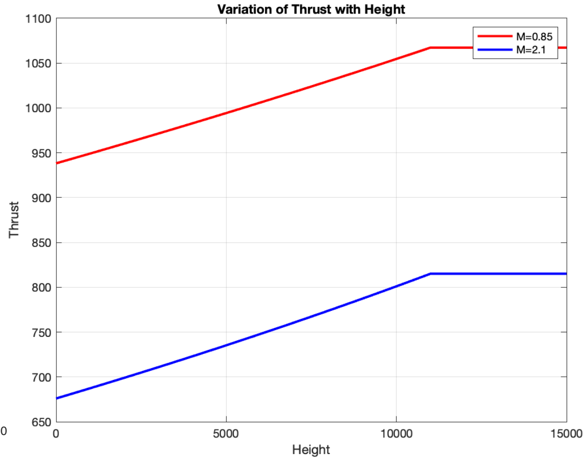


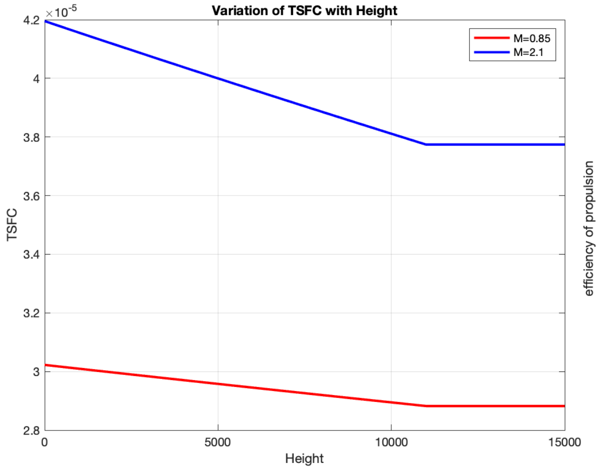


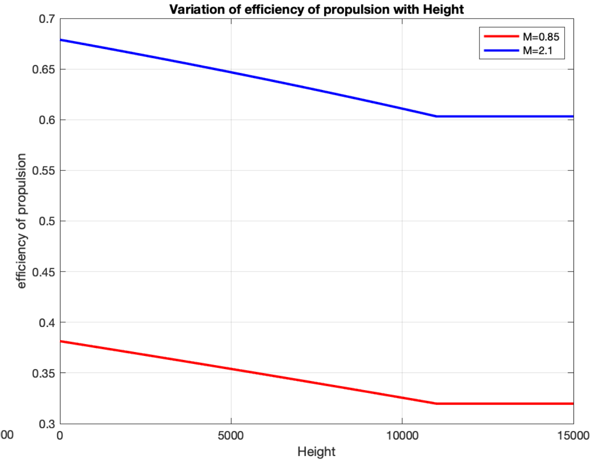
Part (b)

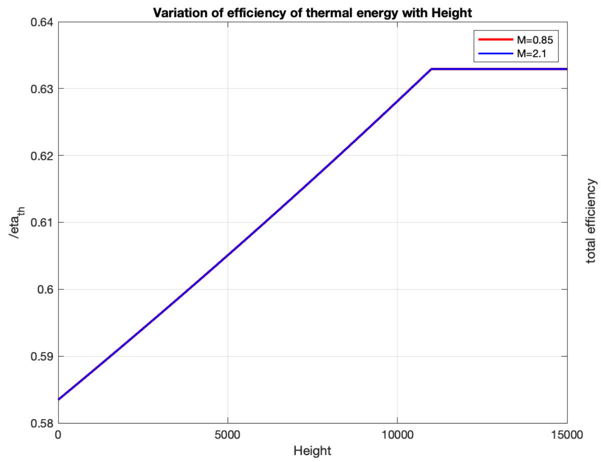
For this case, at different altitude the maximum specific thrust happens in different height, so first of all is to fix the height and change the parameter π_c to find out the best specific thrust performance at this altitude, and change the altitude to do the same over and over again. And the result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

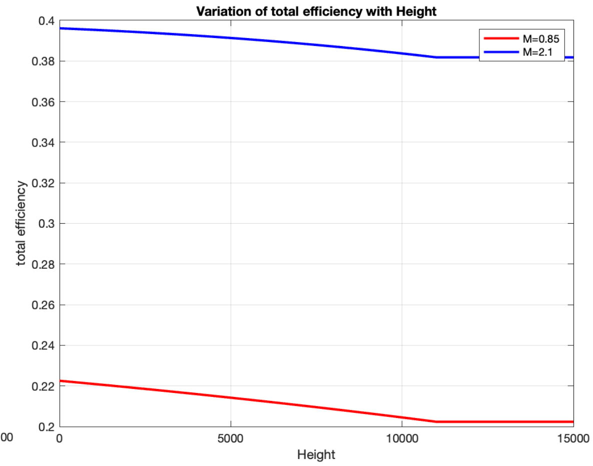






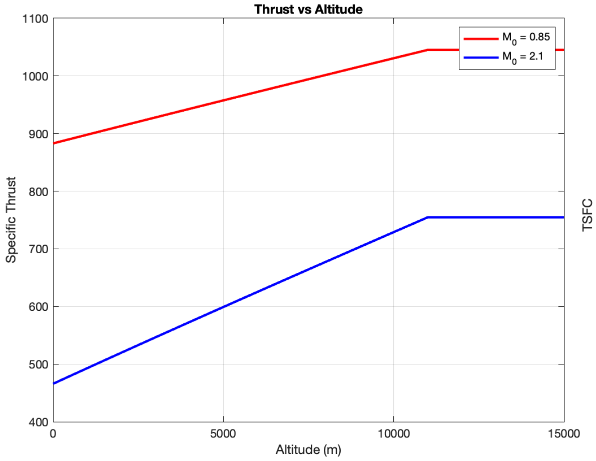


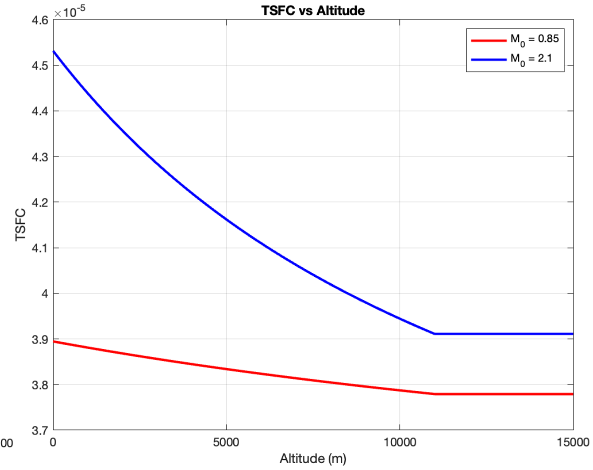


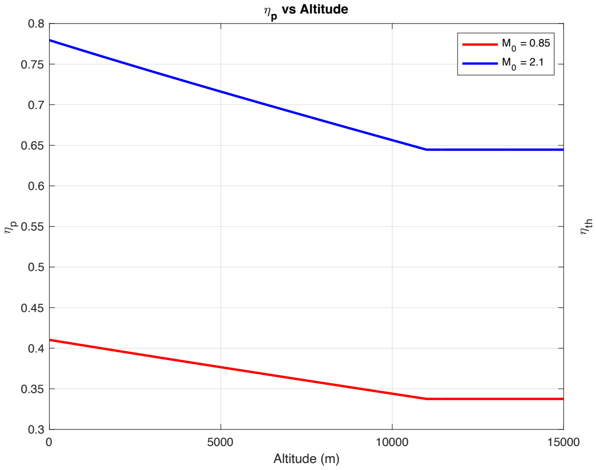


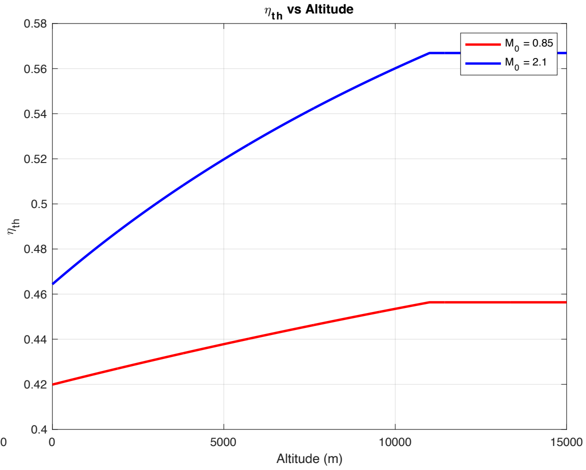
Part (c)

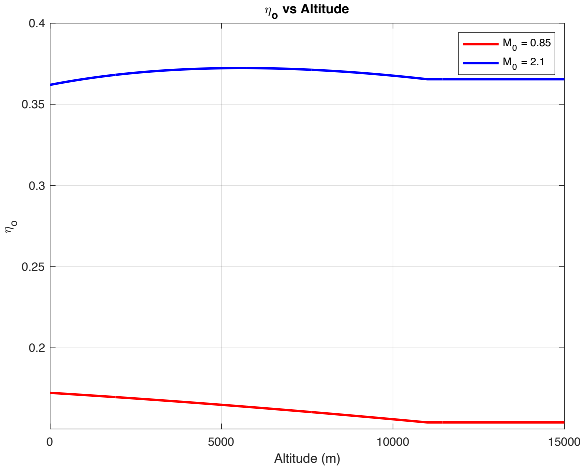
The turbojet is not ideal, so we need to do more calculations, and add them into the code.







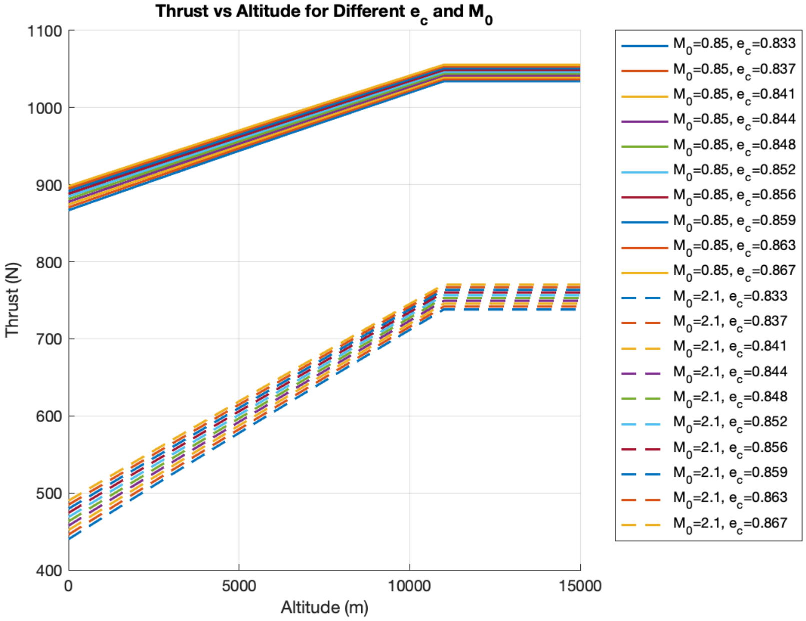




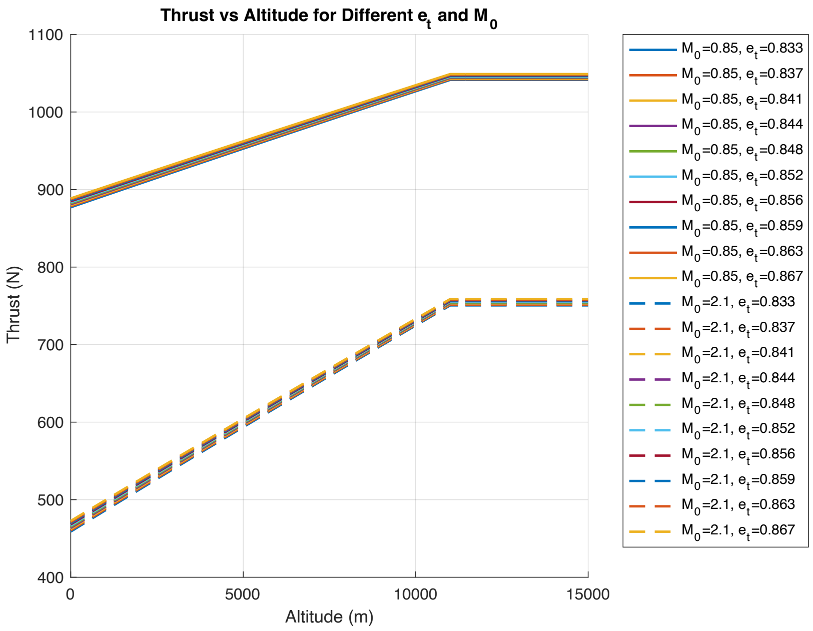
Part (d)

In this part, e_c and e_t have a variance of $\pm 2\%$. So for $e_c, e_t \in [0.833, 0.867]$.

I. For $e_c \in [0.833, 0.867]$



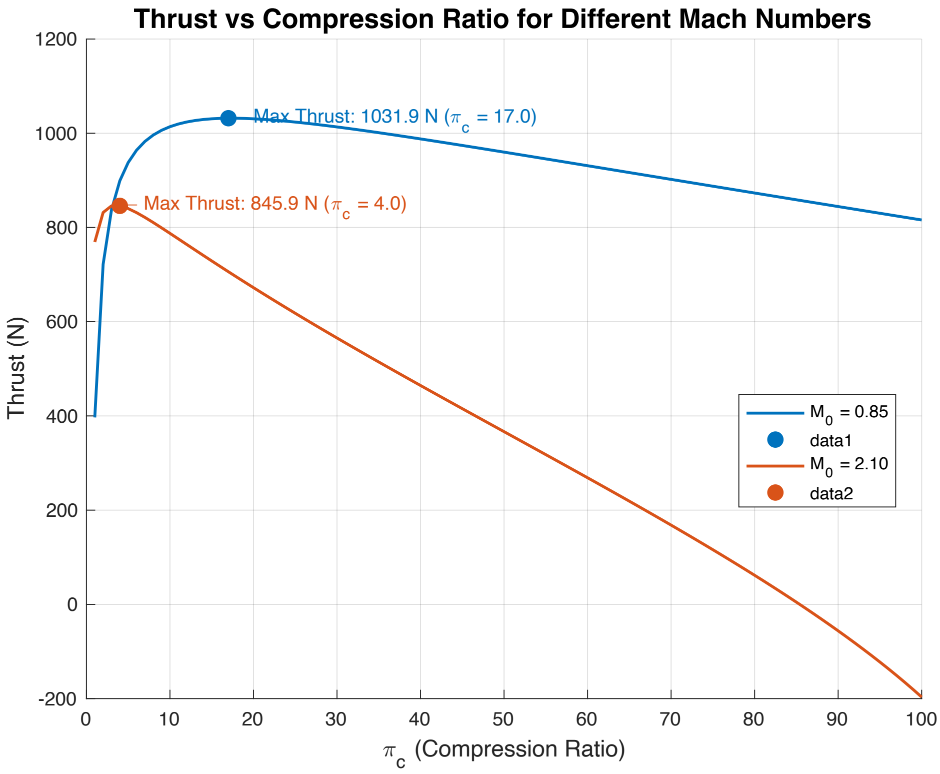
II. For $e_t \in [0.833, 0.867]$



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

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;
```