```
%=========================================================================
=
%
%   This code uses the input numbers for estimated Mach number (M_est), gamma
(gamma),
%   allowable error (allow_error), exit pressure (P_e) in Pa, combustion temperature
%   and pressure (T_o, P_o) in K and Pa respectively, and the molecular weight (MW)
%   to analyze a rocket nozzle. It first checks for choked flow based on exit Mach
%   number (M_e). The function 'nozzle_geometry' takes nozzle profile points and
%   returns the necessary geometry.An iterative solution function, 'solver', returns

%   an array of Mach numbers for the points along the nozzle profile (Mach).
%   These Mach numbers are used to find the temperature and pressure (Temp and
Press)
%   at each point, as well as the rocket performance (m_dot, Thrust, and Isp).
%   Mach number, pressure and temperature distribution along the nozzle are plotted.
%
%


% MAE 5420
% Homework #4
%=========================================================================
==


clc
clear all

% Input Numbers
M_est = 3.0;
gamma = 1.22;
allow_error = 0.0001;

P_e = 16872.7           % Pa
P_o = 20.4E6;           % Pa
P_inf = 0;              % vacuum
T_o = 3500;             % K
MW = 22;                % kg/kgmole

% Determine if nozzle is choked using input conditions
x = P_o/P_e;
y = (gamma - 1)/gamma;
z = 2 / (gamma - 1);

M_e = sqrt(z*(x^y - 1));

if M_e < 1
   disp('Nozzle is not choked!')
else

        % Determine nozzle profile and physical perameters
        [x, points, A_e, A_Astar, location] = nozzle_geometry2;

        % Find Mach number at each location along the nozzle
        % Converging section
        for i = 1:location
                M = M_est/10;
                Mach(i) = solver2(M, A_Astar(i), allow_error, gamma);
        end

        % Diverging section
        for i = location:points
```

```
            M = M_est;
            Mach(i) = solver2(M, A_Astar(i), allow_error, gamma);
        end

        % Calculate temperature and pressure for each point along the nozzle profile
        b = 1 + ((gamma - 1) * Mach.^2 / 2);
        p = gamma/(gamma - 1);

        Temp = T_o./b;
        Press = P_o./(b.^p);

        % Find exit conditions
        T_e = Temp(points);
        P_e = Press(points);
        M_e = Mach(points);

        % Find rocket characteristics from calculated exit conditions
        [m_dot, Thrust, Isp] = Performance2(P_e, P_o, P_inf, T_e, T_o, A_e, MW,
M_e);

        % plot
        plot(x, Mach);
        xlabel('Nozzle Location Along x-axis')
        ylabel('Mach Number')
        title('Mach Number Along SSME Nozzle')

        figure
        plot(x, Press);
        xlabel('Nozzle Location Along x-axis')
        ylabel('Pressure (kPa)')
        title('Pressure Along SSME Nozzle')

        figure
        plot(x, Temp);
        xlabel('Nozzle Location Along x-axis')
        ylabel('Temperature (K)')
        title('Temperature Along SSME Nozzle')

end
```

```
function [x, points, A_e, A_Astar, location] = nozzle_geometry2

%=========================================================================
=
%
%   This function uses predefined nozzle profile to calculate area (A_x) in meters
%   and area ratio (A_Astar) at each point. It also finds the total number of
%   coordinates (points) and the smallest z value, cooresponding to the throat
%   (location). The exit area (A_e) is the only area returned for use later and is
%   in m^2.
%
% Nicole Ortmann
% 24 SEPTEMBER 2008
%=========================================================================
==

% Define nozzle coordinates, x and z, given in centimeters
x_cm = [0.00, 4.00, 8.00, 12.00, 16.00, 20.00, 25.00, 30.00, 50.00, 70.00, 90.00,
...
        100.00, 120.00, 140.00, 160.00, 180.00, 200.00, 220.00, 240.00, 260.00, ...
        280.00, 300.00, 305.00];

z_cm = [17.5, 15.5, 13.00, 12.25, 13.00, 15.5, 18.50, 22.00, 33.00, 41.50, 50.50,
...
        54.50, 61.00, 68.00, 74.50, 80.50, 86.00, 91.00, 97.00, 101.00, 105.00, ...
        107.50, 107.85];

% Convert geometery to meters
x = x_cm * 1E-2;    % meters
z = z_cm * 1E-2;    % meters

% Total number of points along the axis of the nozzle, points
points = length(x);

% Find the throat location, location
location = find (z == min(z));

% Define area at each point along the nozzle, A_x
A_x = z.^2 * pi;    % m^2

% Define area at the throat (A*), Astar and the area ratio (A/A*, A_Astar along
nozzle
Astar = A_x(location);
A_Astar = A_x/Astar;

% Find Area of the exit, A_e
A_e = A_x(points);  % m^2
```

```
function [Mach] = solver2(M_est, A_Astar, allow_error, gamma)

%=========================================================================================
=
%
%   This function is a numerical solver for Mach number given A/A*, gamma and the
%   allowable error. The initial estimate of Mach number is also a required
parameter.
%   The numerical solution is based on Newton's method and iterates a function to
%   convergence. A single value for Mach number is returned.
%
% Nicole Ortmann
% 24 SEPTEMBER 2008
%=========================================================================================
==

M = M_est;
error = 100;        %Initial abitrary value to start while loop


% Iterate approximation until convergence
while (error >= allow_error)
    % Break A/A* equation into smaller pieces
    a = 2 / (gamma + 1);
    b = 1 + (M^2 * (gamma - 1) / 2);
    c = (gamma + 1)/(2 * (gamma - 1));

    % Break derivative equation into smaller pieces
    d = 2^((1 - 3 * gamma)/(2 - 2 * gamma));
    e = M^2 * (2 + (M^2 * (gamma - 1)));
    f = M^2 - 1;

    F_M = (1/M) * (a*b)^c - A_Astar;
    deriv = d * (f/e) * (b/(gamma + 1))^c;
    M = M - (F_M/deriv);
    error = abs((1/M) * (a*b)^c - A_Astar)/A_Astar;
end

Mach = M;
```

```
function [m_dot, Thrust, Isp] = Performance2(P_e, P_o, P_inf, T_e, T_o, A_e, MW,
M_e)

%=========================================================================
=
%
%   This function uses combustion (pressure (P_o) in Pa, and temperature (T_o) in
%   Kelvin) and exit conditions (area (A_e) in meters, temperature (T_e) in Kelvin,
%   Mach number (M_e), and pressure (P_o) in Pa) to calculate rocket performance.
%   Gas characteristics are given by molecular weight (MW) and the ambient pressure,

%   (P_inf), can be specified if not vacuum. It returns single values for
%   mass flow rate (m_dot) in kg/s, thrust (Thrust) in Newtons, and the specific
%   impulse (Isp) in seconds.
%
% Nicole Ortmann
% 24 SEPTEMBER 2008
%=========================================================================
==

% Givens
g_o = 9.8067              % m/s^2
Ru = 8314.4126;           % J/Kkgmole
gamma = 1.22;

% Calculate the gas specific constant, Rg
Rg = Ru/MW   % J/kgK


% Find exit velocity, V_e
V_e = M_e * (sqrt(gamma * Rg * T_e))   % m/s


% Calculate mass flow rate, m_dot
b = 1 + (M_e^2 * (gamma - 1) / 2);
c = (gamma + 1)/(2 * (gamma - 1));

q = sqrt(gamma/Rg);
r = P_o/(sqrt(T_o));

m_dot = q*r*M_e/(b^c)*A_e  % kg/s


% Calculate Thrust
Thrust = m_dot * V_e + A_e * (P_e - P_inf)     % Newtons


% Calculate specific impulse, Isp
Isp = 1/g_o * Thrust/m_dot     % sec
```