```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%% Final Project, 1 Stage %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%% 1D Rocket Motion %%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%% Alex Bartholomew %%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear all
close all

%% Universal constants

G = 6.674e-11;                  % gravitational constant
M_earth = 5.972e24;             % mass of the Earth
mol = 0.029;                    % molar mass of air
R = 8.314;                      % gas constant
P0 = 101325;                    % standard pressure (Pa), sea level
T0 = 300;                       % Florida temperature (K), sea level
rho0 = (mol*P0)/(R*T0);         % air density, sea level
g0 = 9.81;                      % gravitational acceleration, sea level

%% Local constants

% specifications for Falcon 1e

FT = 560000;                  % rocket thrust, in Newtons
C0 = 0.75;                       % drag coefficient, see notes
d = 1.7;                       % rocket diameter, in meters
A = pi*(d/2)^2;                 % rocket cross-sectional area
L = 27.4;                      % rocket length, in meters
m0 = 46760;                    % initial mass, in kg ("wet mass")
empty = 3120;                  % mass when fuel is expended, in kg ("dry mass")
Isp = 221;                      % specific impulse, in seconds
dm = FT/(g0*Isp);               % mass flow rate, dm/dt



%% Euler-Cromer Method

dt = 0.1;                       % time step
z0 = 1;                         % intial altitude
v0 = 0;                         % initial velocity
v = v0;
z = z0;
V = v;
Z = z;
m = m0;
Rho=rho0;
T=T0;
P=P0;
M = m;
Thrust = FT/m;
Drag = 0;
g=g0;
grav=g;
```

```
nextstage=0;
tmax = 500;

for t=dt:dt:tmax

    g = (G*M_earth)/((z+6371000)^2);       % gravitational acceleration, g(z)
    m = m - (dm*dt);                                  % changing mass, m(t)
    [rho,temp,press] = density(z);
                % changing air density by barometric formula, rho(z)
    Cd = CD(v,temp,C0);
    thrust = FT/m;
    drag = 0.5*rho*(v^2)*Cd*A/m;
    if v < 0            % flip drag force vector if rocket falls
        drag = drag*-1;
    end

    v = v + (thrust - drag - g)*dt;  % new velocity
    z = z + v*dt;                    % new altitude


    V = [V,v];
    Z = [Z,z];
    M = [M,m];
    grav = [grav,g];
    Thrust = [Thrust,thrust];
    Drag = [Drag,drag];
    Rho=[Rho,rho];
    T=[T,temp];
    P=[P,press];

    t1=t;

    if z < 0              % rocket crashes or fails to launch
        break
    elseif m < empty      % rocket runs out of fuel, mass becomes stable
        FT = 0;
        dm = 0;
    end

end

%% Plot the trajectory

t = 0:dt:t;
line = zeros(1,size(t,2));

subplot(2,1,1)
plot(t,Z/1000)
title('Rocket altitude')
ylim([0,1.5*max(Z)/1000])
xlabel('time (s)')
ylabel('altitude (km)')
subplot(2,1,2)
plot(t,V)
title('Rocket velocity')
```

```
ylim([1.5*min(V),1.5*max(V)])
xlabel('time (s)')
ylabel('velocity (m/s)')

%% Plot the forces

figure
subplot(3,1,1)
plot(t,Thrust.*(M/1000))
title('Thrust force')
xlabel('time (s)')
ylabel('force (kN)')
ylim([-0.5*max(Thrust.*M/1000),1.5*max(Thrust.*M/1000)])
subplot(3,1,2)
plot(t,Drag.*M,t,line,'--k')
title('Drag force')
xlabel('time (s)')
ylabel('force (N)')
subplot(3,1,3)
plot(t,grav)
title('Gravitational acceleration')
xlabel('time (s)')
ylabel('g (m/s^2)')


%% Define density function


function [x,y,z] = density(z)

    mol = 0.029;
    R = 8.314;
    h = z/1000;                  % h, altitude in km

    if h <= 11                   % pressure and temperature values by altitude
        T = 288.15 - 6.5*h;
        P = 101325*((288.15/(288.15-6.5*h))^(34.1632/-6.5));
    elseif 11 < h && h <= 20
        T = 216.65;
        P = 22632.06*exp(-34.1632*(h-11)/216.65);
    elseif 20 < h && h <= 32
        T = 196.65 + 0.001*z;
        P = 5474.889 * ((216.65/(216.65+(h-20)))^(34.1632));
    elseif 32 < h && h <= 47
        T = 139.05 + 2.8*h;
        P = 868.0187 * ((228.65/(228.65+2.8*(h-32)))^(34.1632/2.8));
    elseif 47 < h && h <= 51
        T = 270.65;
        P = 110.9063 * exp(-34.1632*(h-47)/270.65);
    elseif 51 < h && h <= 71
        T = 413.45 - 2.8*h;
        P = 66.93887*((270.65/(270.65-2.8*(h-51)))^(34.1632/-2.8));
    else %71 < h && h <= 86
        T = 356.65 - 2.0*h;
        P = 3.956420*((214.65/(214.65-2*(h-71)))^(34.1632/-2));
```

```
    end

    rho = (mol*P)/(R*T);

    if 86 < h && h <= 91
        P = exp(-4.22012E-08*h^5 + 2.13489E-05*h^4 - 4.26388E-03*h^3 +
            0.421404*h^2 - 20.8270*h + 416.225);
        rho = exp(7.5691E-08*h^5 - 3.76113E-05*h^4 + 0.0074765*h^3 -
            0.743012*h^2 + 36.7280*h - 729.346 );
        T = 186.8673;
    elseif 91 < h && h <= 100
        P = exp(-4.22012E-08*h^5 + 2.13489E-05*h^4 - 4.26388E-03*h^3 +
            0.421404*h^2 - 20.8270*h + 416.225);
        rho = exp(7.5691E-08*h^5 - 3.76113E-05*h^4 + 0.0074765*h^3 -
            0.743012*h^2 + 36.7280*h - 729.346 );
        T = 263.1905-76.3232*sqrt(1 - ((h-91)/-19.9429)^2);
    elseif 100 < h && h <= 110
        P = exp(-4.22012E-08*h^5 + 2.13489E-05*h^4 - 4.26388E-03*h^3 +
            0.421404*h^2 - 20.8270*h + 416.225);
        rho = exp(7.5691E-08*h^5 - 3.76113E-05*h^4 + 0.0074765*h^3 -
            0.743012*h^2 + 36.7280*h - 729.346 );
        T = 263.1905-76.3232*sqrt(1 - ((h-91)/-19.9429)^2);
    elseif 110 < h && h <= 120
        rho = exp(-8.854164E-05*h^3 + 0.03373254*h^2 - 4.390837*h +
            176.5294);
        P = 0;
        T = 240 + 12*(h-110);
    elseif 120 < h && h <= 150
        P = 0;
        rho = exp(3.661771E-07*h^4 - 2.154344E-04*h^3 + 0.04809214*h^2 -
            4.884744*h + 172.3597);
        T = 1000 - 640*exp(-0.01875*(h-120)*(6356.766 + 120)/(6356.766+h));
    elseif 150 < h %&& h <= 200
        P = 0;
        rho = 02.0763e-09;
        T = 1000 - 640*exp(-0.01875*(h-120)*(6356.766 + 120)/(6356.766+h));
    end

    x = rho;
    y = T;
    z = P;
end

%% Define Drag Coefficient Function

function x = CD(v,T,C0)

    cs = sqrt(1.4*287*T);            % sound speed as function of temperature
    Mach = v/cs;                        % Mach number

    if Mach < 1
        Cd = C0/sqrt(1-Mach^2);              % Prandtl-Glauert Rule
    elseif Mach == 1
        Mach = 0.99999;                      % eliminate the singularity
        Cd = C0/sqrt(1-Mach^2);
```

```
    elseif Mach > 1
        Cd = C0/sqrt(Mach^2 - 1);
    end

    x = Cd;
end
```