%

% Variable List:

% Delta = Time step (s)

% t = Time (s)

% Thrust = Thrust (N)

% Mass = Mass (kg)

% Mass\_Rocket\_With\_Motor = Mass with motor (kg)

% Mass\_Rocket\_Without\_Motor = Mass without motor (kg)

% Theta = Angle (deg)

% C = Drag coefficient

% Rho = Air density (kg/m^3)

% A = Rocket projected area (m^2)

% Gravity = Gravity (m/s^2)

% Launch\_Rod\_Length = Length of launch rod (m)

% n = Counter

% Fn = Normal force (N)

% Drag = Drag force (N)

% Fx = Sum of forces in the horizontal direction (N)

% Fy = Sum of forces in the vertical direction (N)

% Vx = Velocity in the horizontal direction (m/s)

% Vy = Velocity in the vertical direction (m/s)

% Ax = Acceleration in the horizontal direction (m/s^2)

% Ay = Acceleration in the vertical direction (m/s^2)

% x = Horizontal position (m)

% y = Vertical position (m)

% Distance\_x = Horizontal distance travelled (m)

% Distance\_y = Vertical travelled (m)

% Distance = Total distance travelled (m)

% Memory\_Allocation = Maximum number of time steps expected

clear, clc % Clear command window and workspace

itki=load('itki.txt');

% Parameters

Delta = 0.001; % Time step

Memory\_Allocation = 30000; % Maximum number of time steps expected

% Preallocate memory for arrays

t = zeros(1, Memory\_Allocation);

Thrust = zeros(1, Memory\_Allocation);

Mass = zeros(1, Memory\_Allocation);

Theta = zeros(1, Memory\_Allocation);

Fn = zeros(1, Memory\_Allocation);

Drag = zeros(1, Memory\_Allocation);

Fx = zeros(1, Memory\_Allocation);

Fy = zeros(1, Memory\_Allocation);

Ax = zeros(1, Memory\_Allocation);

Ay = zeros(1, Memory\_Allocation);

Vx = zeros(1, Memory\_Allocation);

Vy = zeros(1, Memory\_Allocation);

x = zeros(1, Memory\_Allocation);

y = zeros(1, Memory\_Allocation);

Distance\_x = zeros(1, Memory\_Allocation);

Distance\_y = zeros(1, Memory\_Allocation);

Distance = zeros(1, Memory\_Allocation);

C = 0.4; % Drag coefficient

Rho = 1.2; % Air density (kg/m^3)

A = 4.9\*10^-4; % Rocket projected area (m^2)

Gravity = 9.81; % Gravity (m/s^2)

Launch\_Rod\_Length = 1; % Length of launch rod (m)

Mass\_Rocket\_With\_Motor = 0.01546; % Mass with motor (kg)

Mass\_Rocket\_Without\_Motor = 0.0117; % Mass without motor (kg)

G = 6.674e-11; % gravitational constant

M\_earth = 5.972e24; % mass of the Earth

Theta(1) = 85; % Initial angle (deg)

Vx(1) = 2\*cosd(85); % Initial horizontal speed (m/s)

Vy(1) = 2\*sind(85); % Initial vertical speed (m/s)

x(1) = 0; % Initial horizontal position (m)

y(1) = 0.1; % Initial vertical position (m)

Distance\_x(1) = 0; % Initial horizontal distance travelled (m)

Distance\_y(1) = 0; % Initial vertical distance travelled (m)

Distance(1) = 0; % Initial distance travelled (m)

Mass(1) = Mass\_Rocket\_With\_Motor; % Initial rocket mass (kg)

m(1)=25;

m(2)=25;

n = 1; % Initial time step

while y(n) > 0 % Run until rocket hits the ground

n = n+1; % Increment time step

t(n)= (n-1)\*Delta; % Elapsed time

% Determine rocket thrust and mass based on launch phase

% if t(n) <= 0.1 % Launch phase 1

% Thrust(n) = 56\*t(n);

% Mass(n) = Mass\_Rocket\_With\_Motor;

% elseif t(n) > 0.1 && t(n) < 0.5 % Launch phase 2

% Thrust(n) = 5.6;

% Mass(n) = Mass\_Rocket\_With\_Motor;

% elseif t(n) >= 0.5 && t(n) < 3.5 % Launch phase 3

% Thrust(n) = 0;

% Mass(n) = Mass\_Rocket\_With\_Motor;

% elseif t(n) >= 3.5 % Launch phase 4

% Thrust(n) = 0;

% Mass(n) = Mass\_Rocket\_Without\_Motor; % Rocket motor ejects

% end

if n<422

Thrust(n)=itki(n,2);

else

Thrust(n)=0;

end

dm=1.2;

g(n) = (G\*M\_earth)/((Vy(n)+6371000)^2); % gravitational acceleration, g(z)

% m(n) = m(n) - (dm\*Delta); % changing mass, m(t)

% Normal force calculations

if Distance(n-1) <= Launch\_Rod\_Length % Launch rod normal force

Fn(n) = Mass(n)\*g(n)\*cosd(Theta(1));

else

Fn(n) = 0; % No longer on launch rod

end

%% finding density

mol = 0.029;

R = 8.314;

h = y(n)/1000; % h, altitude in km

if h <= 11 % pressure and temperature values by altitude

T(n) = 288.15 - 6.5\*h; %6.5 is temperature lapse rate

P (n)= 101325\*((288.15/(288.15-6.5\*h))^(34.1632/-6.5));

elseif 11 < h && h <= 20

T(n) = 216.65;

P(n) = 22632.06\*exp(-34.1632\*(h-11)/216.65);

elseif 20 < h && h <= 32

T(n) = 196.65 + 0.001\*z;

P(n) = 5474.889 \* ((216.65/(216.65+(h-20)))^(34.1632));

elseif 32 < h && h <= 47

T(n) = 139.05 + 2.8\*h;

P(n) = 868.0187 \* ((228.65/(228.65+2.8\*(h-32)))^(34.1632/2.8));

elseif 47 < h && h <= 51

T(n) = 270.65;

P(n) = 110.9063 \* exp(-34.1632\*(h-47)/270.65);

elseif 51 < h && h <= 71

T(n) = 413.45 - 2.8\*h;

P(n) = 66.93887\*((270.65/(270.65-2.8\*(h-51)))^(34.1632/-2.8));

else %71 < h && h <= 86

T(n) = 356.65 - 2.0\*h;

P(n) = 3.956420\*((214.65/(214.65-2\*(h-71)))^(34.1632/-2));

end

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

if 86 < h && h <= 91

P(n) = 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(n) = 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(n) = 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(n) = 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(n) = 263.1905-76.3232\*sqrt(1 - ((h-91)/-19.9429)^2);

elseif 100 < h && h <= 110

P(n) = 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(n) = 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(n) = 263.1905-76.3232\*sqrt(1 - ((h-91)/-19.9429)^2);

elseif 110 < h && h <= 120

rho(n) = exp(-8.854164E-05\*h^3 + 0.03373254\*h^2 - 4.390837\*h + 176.5294);

P(n) = 0;

T(n) = 240 + 12\*(h-110);

elseif 120 < h && h <= 150

P(n) = 0;

rho(n) = exp(3.661771E-07\*h^4 - 2.154344E-04\*h^3 + 0.04809214\*h^2 - 4.884744\*h + 172.3597);

T(n) = 1000 - 640\*exp(-0.01875\*(h-120)\*(6356.766 + 120)/(6356.766+h));

elseif 150 < h %&& h <= 200

P(n) = 0;

rho(n) = 02.0763e-09;

T(n) = 1000 - 640\*exp(-0.01875\*(h-120)\*(6356.766 + 120)/(6356.766+h));

end

%%

% Drag force calculation

Drag(n)= 0.5\*C\*Rho\*A\*(Vx(n-1)^2+Vy(n-1)^2); % Calculate drag force

% Sum of forces calculations

Fx(n)= Thrust(n)\*cosd(Theta(n-1))-Drag(n)\*cosd(Theta(n-1))...

-Fn(n)\*sind(Theta(n-1)); % Sum x forces

Fy(n)= Thrust(n)\*sind(Theta(n-1))-(Mass(n)\*g(n))-...

Drag(n)\*sind(Theta(n-1))+Fn(n)\*cosd(Theta(n-1)); % Sum y forces

% Acceleration calculations

Ax(n)= Fx(n)/Mass(n); % Net accel in x direction

Ay(n)= Fy(n)/Mass(n); % Net accel in y direction

% Velocity calculations

Vx(n)= Vx(n-1)+Ax(n)\*Delta; % Velocity in x direction

Vy(n)= Vy(n-1)+Ay(n)\*Delta; % Velocity in y direction

% Position calculations

x(n)= x(n-1)+Vx(n)\*Delta; % Position in x direction

y(n)= y(n-1)+Vy(n)\*Delta; % Position in y direction

% Distance calculations

Distance\_x(n) = Distance\_x(n-1)+abs(Vx(n)\*Delta); % Distance in x

Distance\_y(n) = Distance\_y(n-1)+abs(Vy(n)\*Delta); % Distance in y

Distance(n) = (Distance\_x(n)^2+Distance\_y(n)^2)^(1/2); % Total distance

% Rocket angle calculation

Theta(n)= atand(Vy(n)/Vx(n)); % Angle defined by velocity vector

end

figure('units','normalized','outerposition',[0 0 1 1]) % Maximize plot window

% Figure 1

subplot(3,3,1)

plot(x(1:n),y(1:n));

xlim([0 inf]);

ylim([0 inf]);

xlabel({'Range (m)'});

ylabel({'Altitude (m)'});

title({'Trajectory'});

% Figure 2

subplot(3,3,2)

plot(t(1:n),Vx(1:n));

xlabel({'Time (s)'});

ylabel({'Vx (m/s)'});

title({'Vertical Velocity'});

% Figure 3

subplot(3,3,3)

plot(t(1:n),Vy(1:n));

xlabel({'Time (s)'});

ylabel({'Vy (m/s)'});

title({'Horizontal Velocity'});

% Figure 4

subplot(3,3,4)

plot(t(1:n),Theta(1:n));

xlabel({'Time (s)'});

ylabel({'Theta (Deg)'});

title({'Theta'});

% Figure 5

subplot(3,3,5)

plot(Distance(1:n),Theta(1:n));

xlim([0 2]);

ylim([59 61]);

xlabel({'Distance (m)'});

ylabel({'Theta (Deg)'});

title({'Theta at Launch'});

% Figure 6

subplot(3,3,6)

plot(t(1:n),Mass(1:n));

ylim([.0017 .02546]);

xlabel({'Time (s)'});

ylabel({'Mass (kg)'});

title({'Rocket Mass'});

% Figure 7

subplot(3,3,7)

plot(t(1:n),Thrust(1:n));

xlim([0 0.8]);

xlabel({'Time (s)'});

ylabel({'Thrust (N)'});

title({'Thrust'});

% Figure 8

subplot(3,3,8)

plot(t(1:n),Drag(1:n));

xlabel({'Time (s)'});

ylabel({'Drag (N)'});

title({'Drag Force'});

% Figure 9

subplot(3,3,9)

plot(Distance(1:n),Fn(1:n));

xlim([0 2]);

xlabel({'Distance (m)'});

ylabel({'Normal Force (N)'});

title({'Normal Force'});