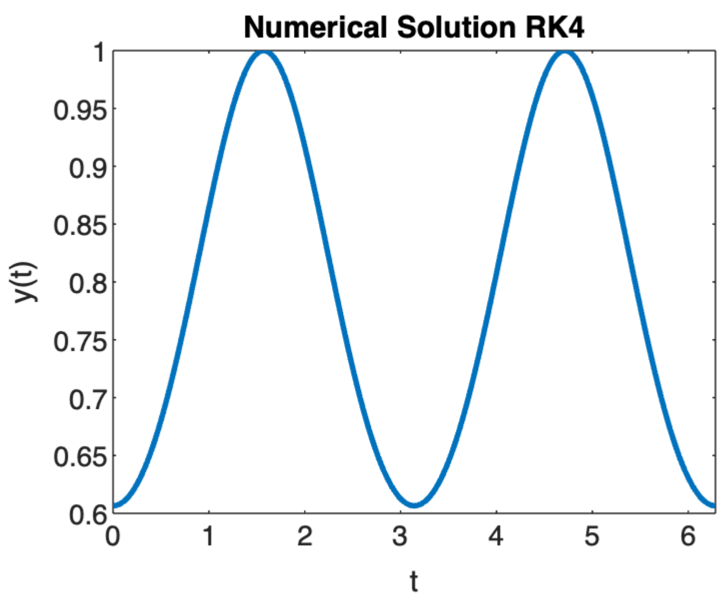
Problem 1.

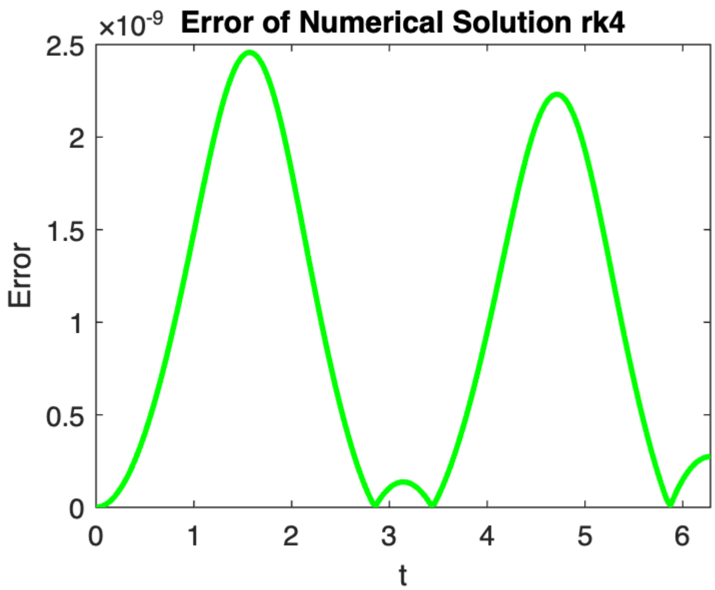
Part (a) and (b)

```
% function
f = @(t,y) cos(t)*sin(t)*y;

% initial conditions
y0 = 1/exp(1/2);
tspan = [0,2*pi];
N = 200;
h = (tspan(2) - tspan(1))/N;

% solve RK4 method
[t,y] = rk4(f,tspan,y0,h);

% Plot
plot(t, y, 'LineWidth', 2);
xlabel('t');
ylabel('y(t)');
title('Numerical Solution RK4');
```



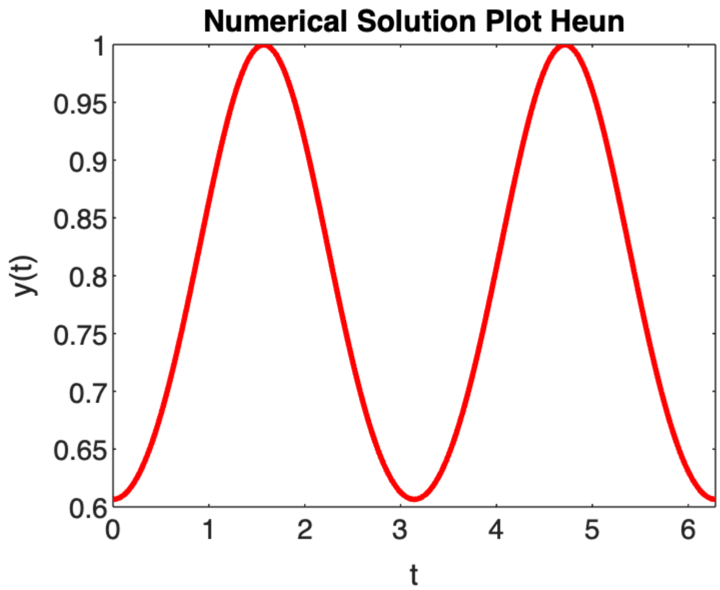
Part (c)

```
% Calculate the error
exact = 1./exp(1/2*cos(t).^2);
error = abs(exact-y);

% Plot error
plot(t,error, 'g-', 'lineWidth', 2);
xlabel('t');
ylabel('Error');
title('Error of Numerical Solution rk4');
```

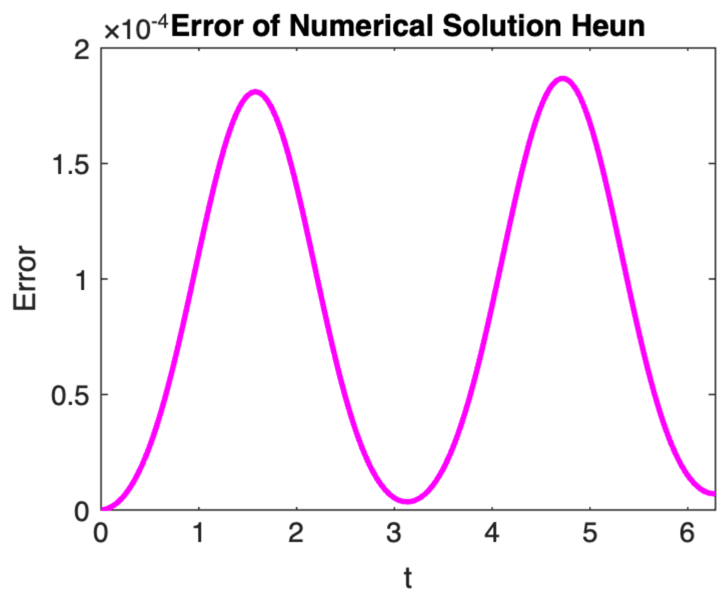


Part (d)

```
clear;

% function
f = @(t,y) cos(t)*sin(t)*y;

% initial conditions
y0 = 1/exp(1/2);
tspan = [0, 2*pi];
N = 200;
h = (tspan(2)-tspan(1))/N;

% heun function
[t,y] = heun(f, tspan, y0, h);

% Plot
plot(t,y, 'r-','lineWidth', 2);
xlabel('t');
ylabel('y(t)');
title('Numerical Solution Plot Heun');
```




```
% Calculate the error
exact = 1./exp(1/2*cos(t).^2);
error = abs(exact-y);

% Plot error
plot(t,error, 'm-', 'lineWidth', 2);
xlabel('t');
ylabel('Error');
title('Error of Numerical Solution Heun');
```

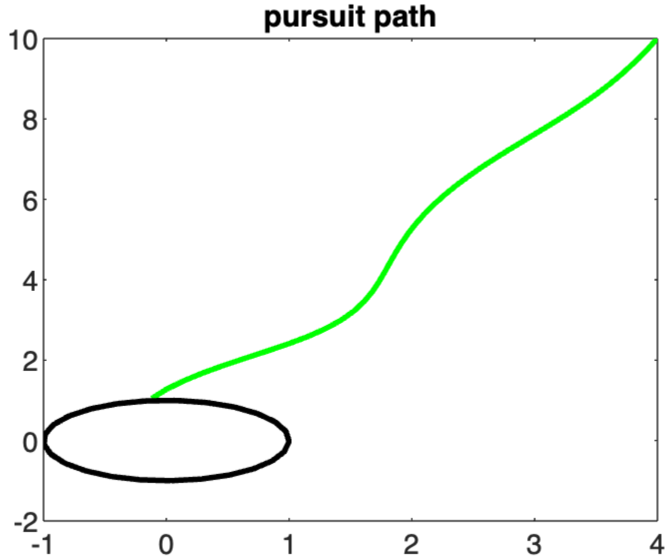



```
display('The error values for rk4 on the y -axis appear to be significantly
smaller, therefor leaving less room for error')
```

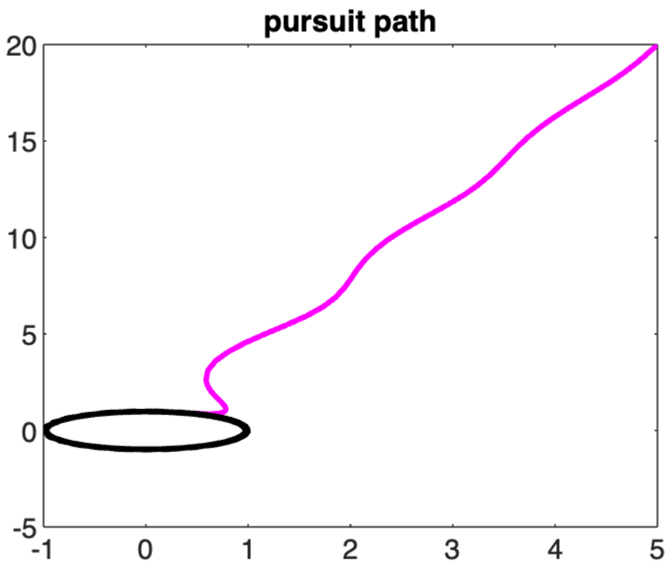
```
The error values for rk4 on the y -axis appear to be significantly smaller, therefor leaving less room for
```

Problem 2.

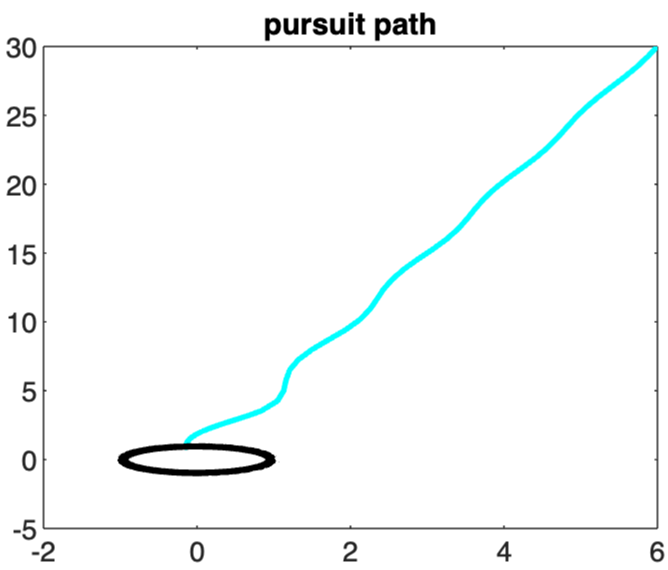
```
clear;

% pursuit path traced out by the fox
tspan = [0 10];
y0 = [4;10];
[x,y] = ode45(@chase,tspan,y0);

plot(y(:,1),y(:,2), 'g-', cos(x),sin(x),'k','linewidth', 2)
title('pursuit path')
```




```
tspan = [0 20];
y0 = [5;20];
[x,y] = ode45(@chase,tspan,y0);

plot(y(:,1),y(:,2), 'm-', cos(x),sin(x),'k','linewidth', 2)
title('pursuit path')
```




```
tspan = [0 30];
y0 = [6;30];
[x,y] = ode45(@chase,tspan,y0);

plot(y(:,1),y(:,2), 'c-', cos(x),sin(x),'k','linewidth', 2)
title('pursuit path')
```

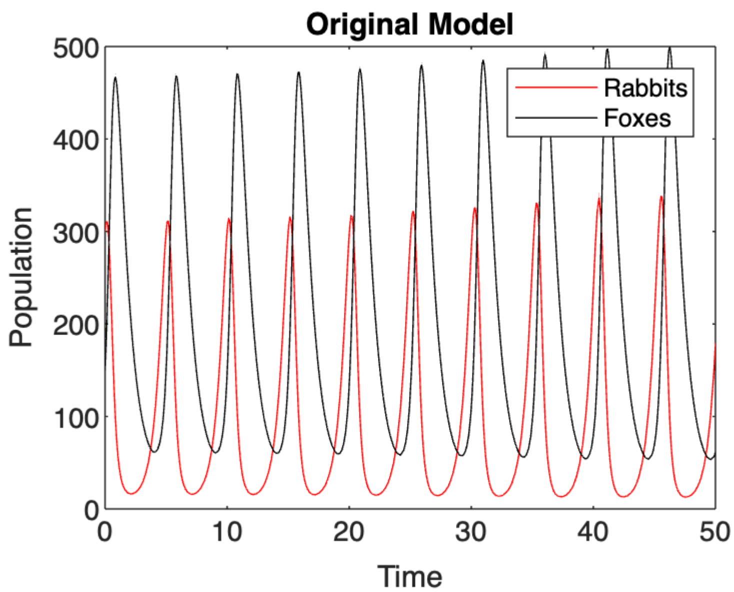


Problem 3.

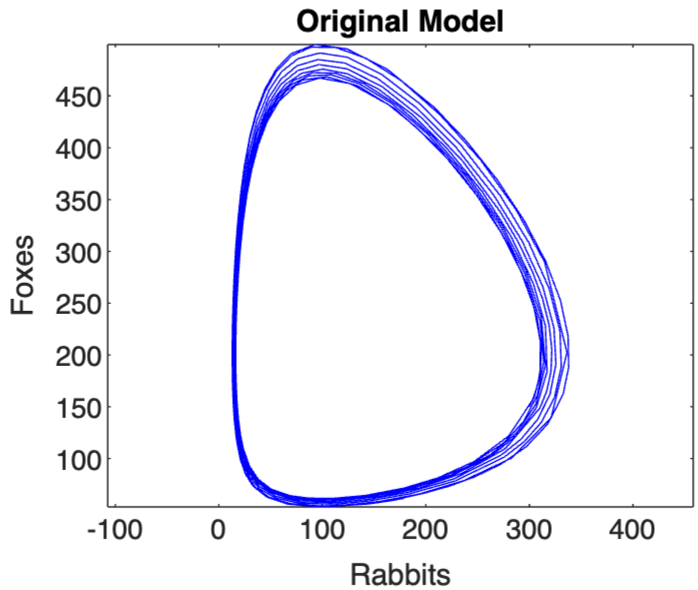
Part (a)

```
tspan = [0 50];
y0 = [300;150];
```

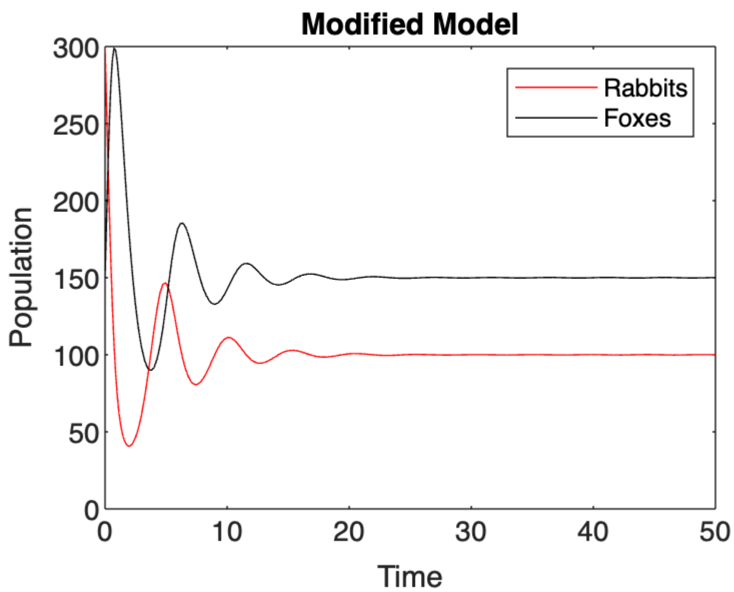
```
% Solve the original system using ode45
[t,y] = ode45(@lotkaVoltera,tspan,y0);

% Solve the modified system using ode45
[t2,y2] = ode45(@lotkaVoltera2,tspan,y0);

% Plot rabbits and foxes vs time original model
figure(1), plot(t,y(:,1),'r-',t,y(:,2),'k-')
xlabel('Time')
ylabel('Population')
legend('Rabbits','Foxes')
title('Original Model')
```



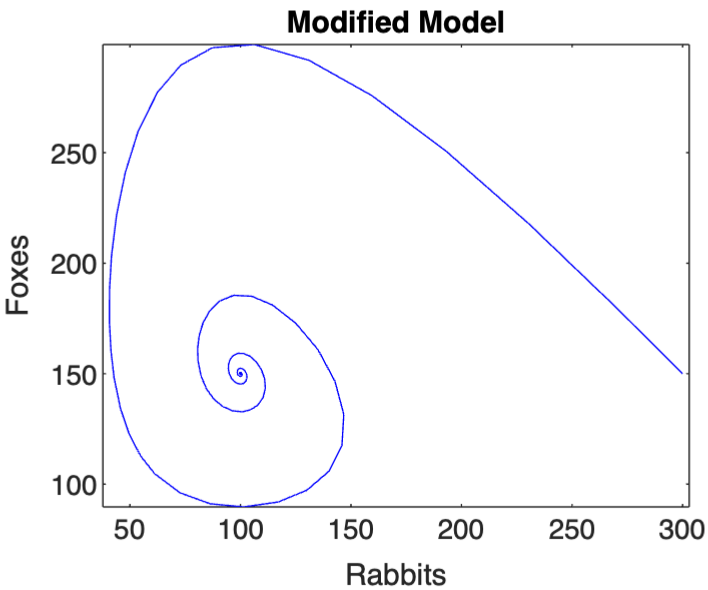

```
% Plot phase portraits for the original model
figure(2), plot(y(:,1),y(:,2),'b-')
xlabel('Rabbits')
ylabel('Foxes')
title('Original Model')
axis equal
```




```
% Plot rabbits and foxes vs time modified model
figure(3), plot(t2,y2(:,1),'r-',t2,y2(:,2),'k-')
xlabel('Time')
ylabel('Population')
legend('Rabbits','Foxes')
title('Modified Model')
```




```
% Plot phase portraits for the modified model
figure(4), plot(y2(:,1),y2(:,2),'b-')
xlabel('Rabbits')
ylabel('Foxes')
title('Modified Model')
axis equal
```



Problem 4.

Part (a)

```
clear;

g = 9.81;
y0 = 1000; %initial position
yp0 = 0; %initial velocity

%Equation for velocity and position
yp = @(t) -g*t + yp0;
y = @(t) -g/2*t.^2 + y0;

%Time paratrooper hits the ground:
tground = sqrt(2*y0/g)
```

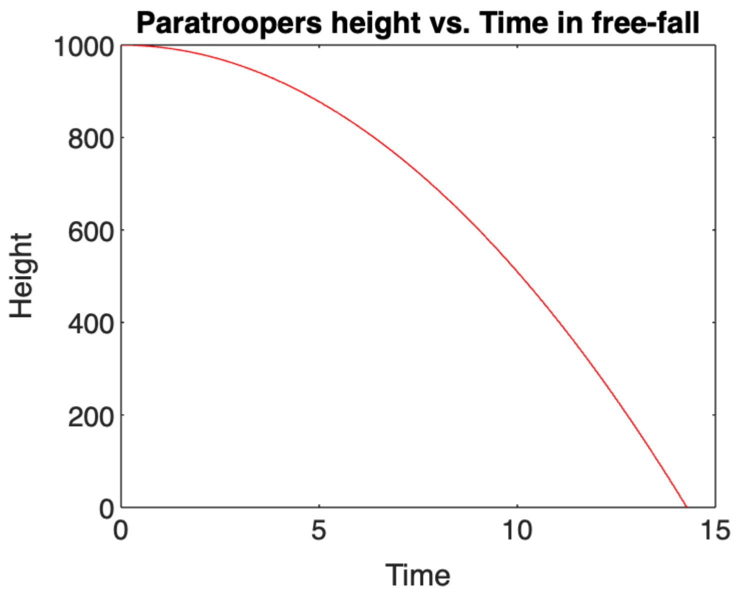
```
tground = 14.2784
```

```
%Imapcat velocity
impact_velocity = yp(tground)
```

```
impact_velocity = -140.0714
```

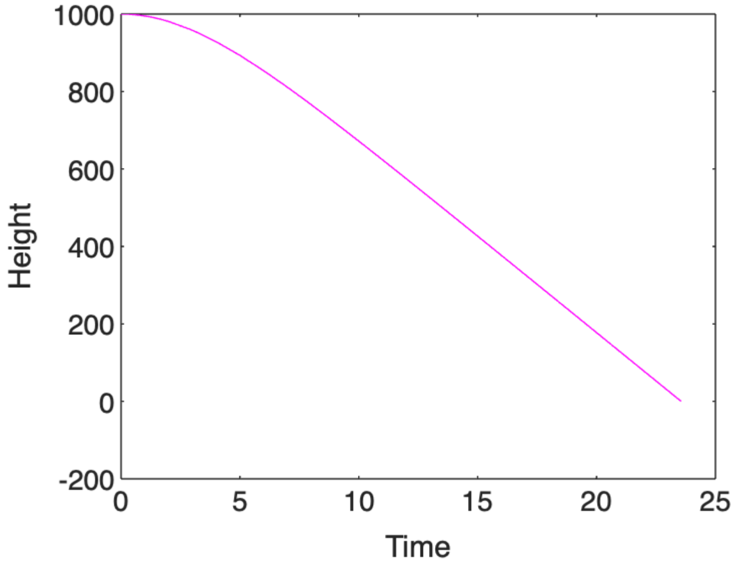
```
t = linspace(0,tground,100);
plot(t,y(t),'r-')
```

```
xlabel('Time')
ylabel('Height')
title('Paratroopers height vs. Time in free-fall')
```



Part (b)

```
tspan = [0 inf];
yy0 = [y0;yp0];
opts = odeset('Events',@ground);
[t,yy] = ode45(@paratrooper,tspan,yy0,opts);

height = yy(:,1);
velocity = yy(:,2);
plot(t,height,'m-')
xlabel('Time')
ylabel('Height')
```

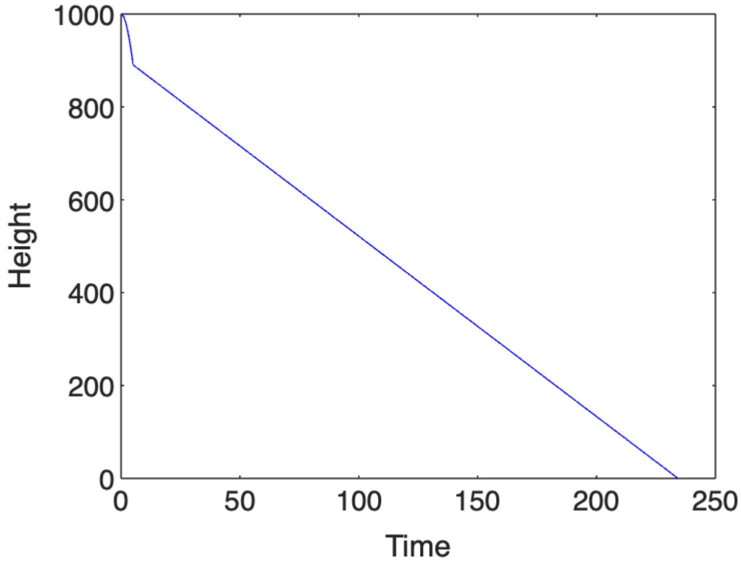



```
tground_b = t(end);
impact_velocity_b = velocity(end)
```

```
impact_velocity_b = -49.9076
```

Part (c)

```
tspan = [0 inf];
yy0 = [y0;yp0];
opts = odeset('Events',@ground);
[t,yy] = ode45(@paratrooper2,tspan,yy0,opts);

height = yy(:,1);
velocity = yy(:,2);
plot(t,height,'b-')
xlabel('Time')
ylabel('Height')
```




```
tground_b = t(end);
impact_velocity_b = velocity(end)
```

```
impact_velocity_b = -3.8812
```

```

```

Functions

```
function [t,y] = heun(f,tspan,y0,h)
t0 = tspan(1);
tend = tspan(2);
numsteps = ceil((tend-t0)/h);
y = zeros(1,numsteps+1);
y(1) = y0;
t = zeros(1,numsteps+1);
t(1) = t0;
for n = 1:numsteps
    s1 = f(t(n),y(n));
    s2 = f(t(n)+h,y(n)+h*s1);
    y(n+1) = y(n) + h/2*(s1 + s2);
    t(n+1) = t0 + n*h;
end
end

function [t,y] = rk4(f,tspan,y0,h)
    t0 = tspan(1);
    tend = tspan(2);
    numsteps = ceil((tend-t0)/h);
    y = zeros(1,numsteps+1);
```

```
    y(1) = y0;
    t = zeros(1,numsteps+1);
    t(1) = t0;

    for n = 1:numsteps
        s1 = f(t(n), y(n));
        s2 = f(t(n)+0.5*h, y(n)+0.5*h*s1);
        s3 = f(t(n)+0.5*h, y(n)+0.5*h*s2);
        s4 = f(t(n)+h, y(n)+h*s3);

        y(n+1) = y(n)+h/6*(s1+2*s2+2*s3+s4);
        t(n+1) = t0+n*h;
    end
end

function g_1 = chase(t,y)

g_1 = zeros(2,1);
R = [cos(t);sin(t)];
F = [y(1);y(2)];
nRF = norm(R-F,2);
F_1 = (R-F)/nRF;

g_1(1) = F_1(1);
g_1(2) = F_1(2);

end

function g = lotkaVoltera(t,y)

alpha = 0.01;
g = zeros(2,1);

g(1) = 2*y(1) - alpha*y(1)*y(2);
g(2) = -y(2) + alpha*y(1)*y(2);

end

function g = lotkaVoltera2(t,y)

alpha = 0.01;
R = 400;
g = zeros(2,1);

g(1) = 2*(1-y(1)/R)*y(1) - alpha*y(1)*y(2);
g(2) = -y(2) + alpha*y(1)*y(2);

end
```

```
function f = paratrooper(t,y);

%total mass of paratrooper
m = 17+15+80;
g = 9.81;
CD1 = 1.2;
A1 = 0.6;
rho = 1.225;
FD = 0.5*rho*CD1*A1*y(2)^2;
f = zeros(2,1);
f(1) = y(2);
f(2) = -g + 1/m*FD;

end

function [position,isstop,direction] = ground(t,y)

position = y(1);
isstop = 1;
direction = [];

end

function f = paratrooper2(t,y);

%total mass of paratrooper
m = 17+15+80;
g = 9.81;
CD1 = 1.2;
CD2 = 1.75;
A1 = 0.6;
d = 9.3;
A2 = 0.25*pi*d^2;
rho = 1.225;

if t < 5
FD = 0.5*rho*CD1*A1*y(2)^2;
else
    FD = 0.5*rho*CD2*A2*y(2)^2;
end
f = zeros(2,1);
f(1) = y(2);
f(2) = -g + 1/m*FD;
end
```