MECH 230 Dynamics
Homework 1

Dr. Theresa Honein

Due Wednesday September 4, 2024

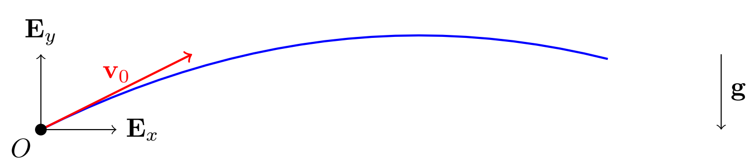
This homework will walk you through reproducing Figure 1.1 and 1.6 of

O. M. O'Reilly, *Engineering Dynamics: A Primer*, Third Edition
Springer-Verlag, New York, 2019.

The electronic version of this text is available for free here.

1. Read Matlab's ode45 tutorial.

2. Consider a particle of mass m that is launched into the air from a point with initial velocity $\mathbf{v}_0$ at $t = 0$. At this instant, $\mathbf{r} = \mathbf{r}_0 = x_0\mathbf{E}_x + y_0\mathbf{E}_y + z_0\mathbf{E}_z$. During the subsequent motion of the particle it is under the influence of a vertical gravitational force $-mg\mathbf{E}_y$. In SI units, g is approximately 9.81 meter per second per second (m s^{-2}). The particle is also under the influence of the drag force $\mathbf{F}_D = -\frac{1}{2}\rho C_d A v \mathbf{v}$.



   (a) Write expressions of the position, velocity, and acceleration vectors for the particle in Cartesian coordinates.

   (b) Draw a free body diagram of the particle and express the forces acting on it in vector form.

   (c) Write the expression for the balance of linear momentum (also known as Newton's second law or Euler's first law) for the particle in vector form.

   (d) Project the balance of linear momentum equation along the $\mathbf{E}_x$, $\mathbf{E}_y$, $\mathbf{E}_z$ directions to obtain three scalar differential equations for the motion of the particle.

(e) To solve these differential equations numerically, we start by defining the column vector

```
y = [y(1); y(2); y(3); y(4); y(5); y6]
```

where

```
y1 = x
y2 = y
y3 = z
y4 = dxdt
y5 = dydt
y6 = dzdt
```

Then,

```
dy1dt = y4;
dy2dt = ...(1);
dy3dt = ...(2);
dy4dt = -k*sqrt(y4^2+y5^2+y6^2)*y4;
dy5dt = ...(3);
dy6dt = ...(4);
```

where `k` replaces the expression $\frac{\rho C_d A}{2m}$. Using your answers in part (d), provide the missing expressions `(1)`, `(2)`, `(3)`, and `(4)`.

(f) In Matlab, create a script, in which you define the function the function `eom` as follows:

```
% defining the equation of motion
function dydt = eom(t,y,k,g)
    dydt = zeros(6,1);
    dydt(1) = y(4);
    dydt(2) = ...;
    dydt(3) = ...;
    dydt(4) = -k*sqrt(y(4)^2+y(5)^2+y(6)^2)*y(4);
    dydt(5) = ...;
    dydt(6) = ...;
end
```

where, again, you need to provide the missing expressions.

(g) To solve these equations of motion for several values of `k`, we use Matlab's `ode45` inside a for loop. At the beginning of your script, insert the following code snippet.

```
k = [0, 0.01, 0.05];
g = 9.81;       % graviational constant
tspan = [0 15]; % simulation time span
y0 = ...;       % vector of initial conditions

for i=1:3
    % solving the differential eom
    [t,y] = ode45(@(t,y) eom(t,y,k(i),g), tspan, y0);
end
```

Define the vector of initial conditions `y0` knowing that

$$\mathbf{r}_0 = \mathbf{0},$$
$$\mathbf{v}_0 = 5\mathbf{E}_x + 10\mathbf{E}_y.$$

(h) To reproduce Fig 1.6, add the following code snippet inside the for loop after solving the differential equations of motion.

```
%  Fig 1.6
v = sqrt(y(:,4).^2+y(:,5).^2+y(:,6).^2);    % calculating speed
figure(1)   % initializing figure 2
hold on     % plot new curves without erasing old ones
box on      % box figure
plot(t,v)   % plot speed as a function of time
axis([0 15 0 40])       % axis limits axis([xmin xmax ymin ymax])
xlabel('t (seconds)')   % x-axis label
ylabel('v (m/s)')       % y-axis label
quiver(2,5,-1,4,'k')    % arrow towards increasing k, see documentation
```

Add code to reproduce Fig 1.1 that includes the three projectile trajectories and two unit vectors $\mathbf{E}_x$ and $\mathbf{E}_y$. You will need to use the command `axis equal`.

(i) To calculate the terminal velocity of the particle, we set the accelerations to zero in the equations of motion and solve for the velocity vector:

$$\mathbf{v}_{\text{term}} = -v_{\text{term}}\mathbf{E}_y = -\left(\frac{2mg}{\rho C_d A}\right)^{1/2}\mathbf{E}_y.$$

Verify that this result agrees with your replica of Fig. 1.6.

Instructions Your homework submissions should consist of

1. A hard copy of your answers to questions (2a-e) and (2i).

2. A hard copy of your complete final code containing your implementation of questions (2f-h).
3. Hard copy of your reproductions of figures 1.1 and 1.6.