Modelling of Particle with mass m with an external force acting on it.

Clear all data

```
clc;
clear all;
close all;
clf; %Clear all figures
```

Basic Definition

```
particle_mass = 1; % mass of particle in kg
initial_parameters = [0 ; 0; 0; 0]; % Initial (x,x_dot,y,y_dot) parameters of particle
applied_force = [1 ; 45]; % Applied force to the particle (F, thetha)
% where F is the magnitude and thetha is the angle of application of force wrt to +ve x
gravitational_acceleration = 9.8; %m/s2
graph_time = 10; % Time frame for plotting
```

Symbolic Definition

```
syms m x x_dot x_ddot y y_dot y_ddot  F thetha g 'real' % Creating Symbolic Variables
```

Defining State Space Matrix

```
X = sym('X',[4,1]); %Create a 4X1 State Space Matrix Vector
X(1) = x;
X(2) = x_dot;
X(3) = y;
X(4) = y_dot;
X
```

X =

$$\begin{pmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{pmatrix}$$

Systems Equations extracted from Euler-Lagrange Matrix which were solved using hand

```
eq1 = (m*x_ddot) - (F*cosd(thetha)) == 0;
eq2 = (m*y_ddot) + (m*g) - (F*sind(thetha)) == 0;
```

Symbolic Solution to form state space matrix

```
[x_ddot, y_ddot] = solve([eq1,eq2],[x_ddot,y_ddot]);
```

Calculating dX Matrix

```
dX = sym('dX',[4,1]); %Create a 4x1 dX Matrix
dX(1) = X(2,1);
dX(2) = x_ddot;
dX(3) = X(4,1);
dX(4) = y_ddot;
dX
```

dX =

$$\begin{pmatrix} \dot{x} \\ \frac{F\cos\left(\frac{\pi\,\mathrm{thetha}}{180}\right)}{m} \\ \dot{y} \\ -\frac{g\,m - F\sin\left(\frac{\pi\,\mathrm{thetha}}{180}\right)}{m} \end{pmatrix}$$

Substituting values fed by user

```
sub_dX = subs( ...
    dX, ...
    [F,thetha,m,g], ...
    [applied_force(1,1),applied_force(2,1),particle_mass,gravitational_acceleration])
```

sub_dX =

$$\begin{pmatrix} \dot{x} \\ \frac{\sqrt{2}}{2} \\ \dot{y} \\ \frac{\sqrt{2}}{2} - \frac{49}{5} \end{pmatrix}$$

```
%Substitue Values
fun_dX = @(t,x)[
    x(2);
    double(sub_dX(2,1));
    x(4);
    double(sub_dX(4,1))];
% Convert the symbolic matrix to double matrix
```

Calculating Differential Equation to get velocity

```
tspan = [0: 0.1: graph_time];
rsize = size(tspan)
```

```
rsize = 1×2
     1   101
```

```
[t,x] = ode45(fun_dX,tspan,initial_parameters);
```

Plotting Graphs

```
for i = 1:1:rsize(1,2)
    link1XCoordinates = [0 x(i,1)];
    link1YCoordinates = [0 x(i,2)];
    %plot(xunit, yunit,'k','LineStyle','--'); % Draw Circular Axes
    hold on;
    plot(link1XCoordinates, link1YCoordinates, 'ro','MarkerSize',2);
    pause(0.1); % pause to see realtime animation. Given in seconds
    hold off;
end
```