Lab-VI

Devansh Shukla
I18PH021

1 Aim

Write and execute an octave program to simulate/solve planetary motion.

2 Theory

3 Program

3.1 Simple pendulum

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PLanetaryMotion
%
% Program to solve/simulate planetary motion using RK45 method
% Author: Devansh Shukla I18PH021
% 9th March, 2022
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Defining global parameters\
global G M;

% simplifing mass to obtain GM = 100.05 m^3/s^2
G = 6.67 * 1e-11 % m^3/kg/s^2 -- Newtons graviations constant
M = 1.5 * 1e12 % kg mass
printf("Solving using RK45 method")

% initial position of planet
r = 10;

% time parameters
t0 = 1;
tf = 2800;
dt = 0.05;

% function to calculate acceleration in x direction
function f = Fx(t, x, y, vx ,vy)
   global G M;
   r = sqrt (x**2 + y**2) ;
   f = -G*M*x/r**3;
endfunction

% function to calculate acceleration in y direction
function f = Fy(t, x, y, vx ,vy)
   global G M;
   r = sqrt (x**2 + y**2) ;
   f = -G*M*y/r**3;
endfunction

idx = 1;

figure();
hold on;
grid on;
set(gcf, 'PaperSize', [6, 3]);
set(gca,'XMinorTick','on','YMinorTick','on')

for e=0:0.1:0.7
   x(1) = r;
   y(1) = 0;
   vx (1) = 0;
   vy (1) = sqrt ((e + 1)*G*M/r); % velocity formula

   for t=t0:tf
      dx1 = dt*vx(t);
      dy1 = dt*vy(t);

      dvx1 = dt*Fx(t*dt, x(t), y(t), vx(t), vy(t));
      dvy1 = dt*Fy(t*dt, x(t), y(t), vx(t), vy(t));

      dvx2 = dt*Fx(t*dt + dt/2, x(t) + dx1/2, y(t) + dy1 /2, vx(t) + dvx1/2, vy(t) + dvy1/2);
      dvy2 = dt*Fy(t*dt + dt/2, x(t) + dx1/2, y(t) + dy1 /2, vx(t) + dvx1/2, vy(t) + dvy1/2);

      dx2 = dt*(vx(t) + dvx1 /2);
      dy2 = dt*(vy(t) + dvy1 /2);

      dvx3 = dt*Fx(t*dt + dt/2, x(t) + dx2/2, y(t) + dy2/2, vx(t) + dvx2/2, vy(t) + dvy2/2);
      dvy3 = dt*Fy(t*dt + dt/2, x(t) + dx2/2, y(t) + dy2/2, vx(t) + dvx2/2, vy(t) + dvy2/2);

      dx3 = dt*(vx(t) + dvx2 /2);
      dy3 = dt*(vy(t) + dvy2 /2);

      dvx4 = dt*Fx(t*dt + dt, x(t) + dx3, y(t) + dy3, vx(t) + dvx3, vy(t) + dvy3);
```

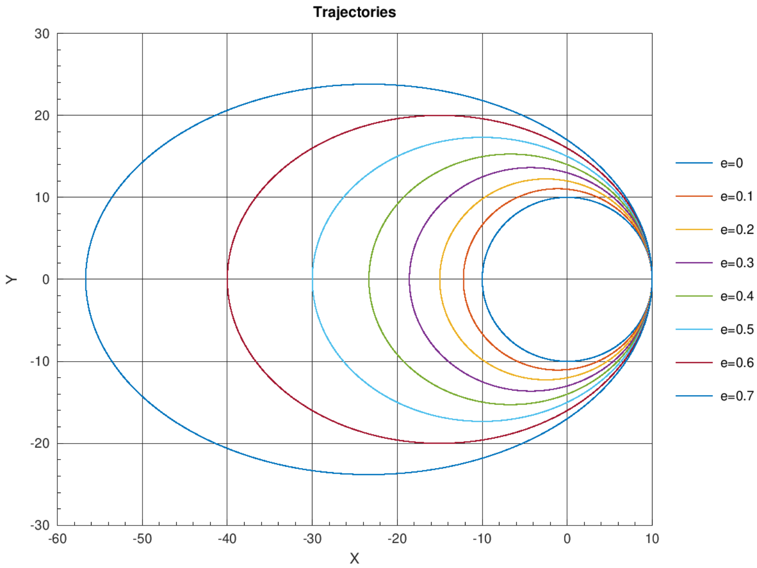
```
        dvy4 = dt*Fy(t*dt + dt, x(t) + dx3, y(t) + dy3, vx(t) + dvx3, vy(t) + dvy3);

        dx4 = dt*(vx(t) + dvx3);
        dy4 = dt*(vy(t) + dvy3);

        x(t + 1) = x(t) + (dx1 + 2*dx2 + 2*dx3 + dx4)/6;
        y(t + 1) = y(t) + (dy1 + 2*dy2 + 2*dy3 + dy4)/6;
        vx(t + 1) = vx(t) + (dvx1 + 2*dvx2 + 2*dvx3 + dvx4 )/6;
        vy(t + 1) = vy(t) + (dvy1 + 2*dvy2 + 2*dvy3 + dvy4 )/6;

    endfor
    plot(x, y, "linewidth", 2)
    drawnow

    for i=2:tf-1
        if ((sign (y(i + 1)) ~= sign (y(i))) && (x(i) > 0))
            time_period(idx) = (2*i -1) *dt /2;
            break;
        endif
    endfor
    for i=2:tf-1
        if (sign (y(i +1)) ~= sign (y(i)))
            semi_major_axis(idx) = (r -(x(i+1)+x(i))/2)/2;
            break ;
        endif
    endfor
    idx = idx + 1;
endfor

title("Trajectories");
xlabel("X");
ylabel("Y");
legend("e=0", "e=0.1", "e=0.2", "e=0.3", "e=0.4", "e=0.5", "e=0.6", "e=0.7", "location", "eastoutside");
legend boxoff
set(gcf, 'renderer', 'painters');
print("-dpng", "planet_traj.png");
hold off


figure();
hold on;
grid on;
set(gcf, 'PaperSize', [6, 3]);
set(gca,'XMinorTick','on','YMinorTick','on')
plot(time_period.^2 ,semi_major_axis.^3, "marker", "+", "linewidth", 2);
xlabel("T^2");
ylabel("a^3");
title("Keplers third law (T^2 vs a^3)");
print("-dpng", "planet_kepler.png");
hold off
```

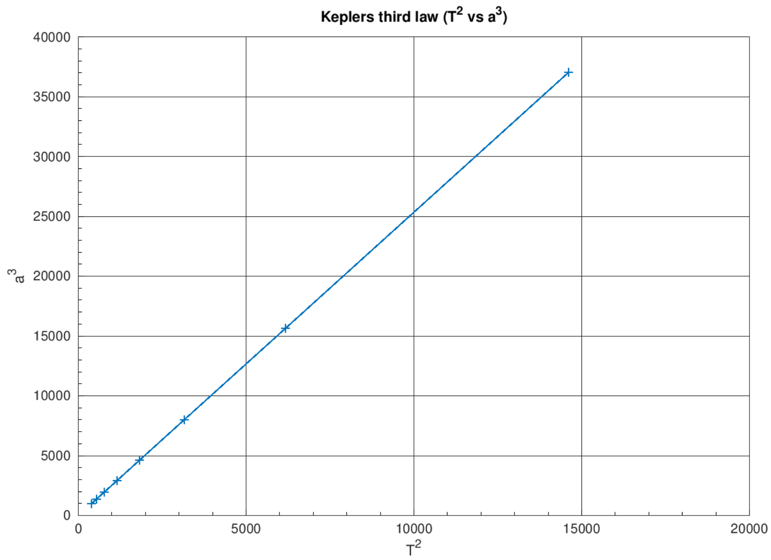
4 Results

4.1 Terminal output

```
(escape) devansh@ds:~/GitHub/Vault/OctaveLab/Programs/outputs$ octave ../PlanetaryMotion.m
G =    6.6700e-11
M =    1.5000e+12
Solving using RK45 method
```

4.2 Plots





5 Remarks

The programs can be used to trace and simulate the platenary motion by defining the inital parameters.