PHY2001

Planetary Motion

Thomas Tsoi (02592751)

1. Problem

According to Newton's second law and the law of gravitation, we have

$$m\frac{d^2\vec{r}}{dt^2} = -\frac{GMm}{r^2}\vec{e}_r \ ,$$

which describes the motion of a planet with mass m revolving around the sun of mass M.

Assume that the orbit is circular, we have

$$\frac{GMm}{R^2} = mRw^2 = mR\left(\frac{2p}{T}\right)^2$$

$$\frac{GM}{R^3} = \frac{4p^2}{T^2}$$

$$\frac{R^3}{T^2} = \frac{GM}{4p^2}$$

By choosing the units such that R=1 and T=1, we obtain

$$\frac{1}{1} = \frac{GM}{4p^2}$$

$$GM = 4p^2$$

We shall then try to solve and investigate the motion of the planet by using a spreadsheet.

2. Methodology

From the equation obtained above, we see that

$$m\frac{d^2\vec{r}}{dt^2} = -\frac{GMm}{r^2}\vec{e}_r$$

$$\frac{d^2\vec{r}}{dt^2} = -\frac{GM}{r^3}\vec{r}$$

which is a second order differential equation. Discretizing the ODE, we have

$$x(n+1) = x(n) + v_x(n)\Delta t$$

$$v_x(n+1) = a_x(n) + v_x(n)\Delta t$$

$$a_x(n+1) = -GM\frac{x(n)}{r^3} = -4p^2\left(x^2+y^2\right)^{3/2}\cdot x(n)$$

$$y(n+1) = y(n) + v_y(n)\Delta t$$

$$v_y(n+1) = a_y(n) + v_y(n)\Delta t$$

$$a_y(n+1) = -GM\frac{y(n)}{r^3} = -4p^2\left(x^2+y^2\right)^{3/2}\cdot y(n)$$

By iterating these equations with initial conditions $\{x=1,\ y=0,\ v_x=0,\ v_y=6\}$ in a spreadsheet, we can solve the ODE numerically.

3. Results

When Δt is large, the energy and angular momentum are not conserved. As Δt decreases, angular momentum is preciously conserved and total energy is almost conserved (with small oscillating differences, which further decrease in magnitude as Δt decreases).

Particularly, when $\Delta t = 0.008$, the results are shown in *Figure 1* and *Chart 1* in the attached sheet. When only a few periods are observed, the orbit seems to be closed.

4. Comments and Discussions

The orbit seems to be closed when only a few periods are observed, but if the period is repeated for many times, we see that it is not completely closed, as shown in *Figure 2* and *Figure 3*. It is noticed that f we decrease the value of Δt, the orbits look more closed, therefore such non-closure is due to the non-infinitesimal time interval Δt.

By adjusting the initial conditions, it is observed the planet can be flung away when its total energy is larger or equal to 0, an example is shown in *Figure 4*. This means that if the magnitude of kinetic energy is larger than that of the potential energy, the planet is flung away. In other words, since energy is conserved throughout the motion, if in the initial conditions, the velocity components are too large relative to their positions x and y, the planet is flung away.

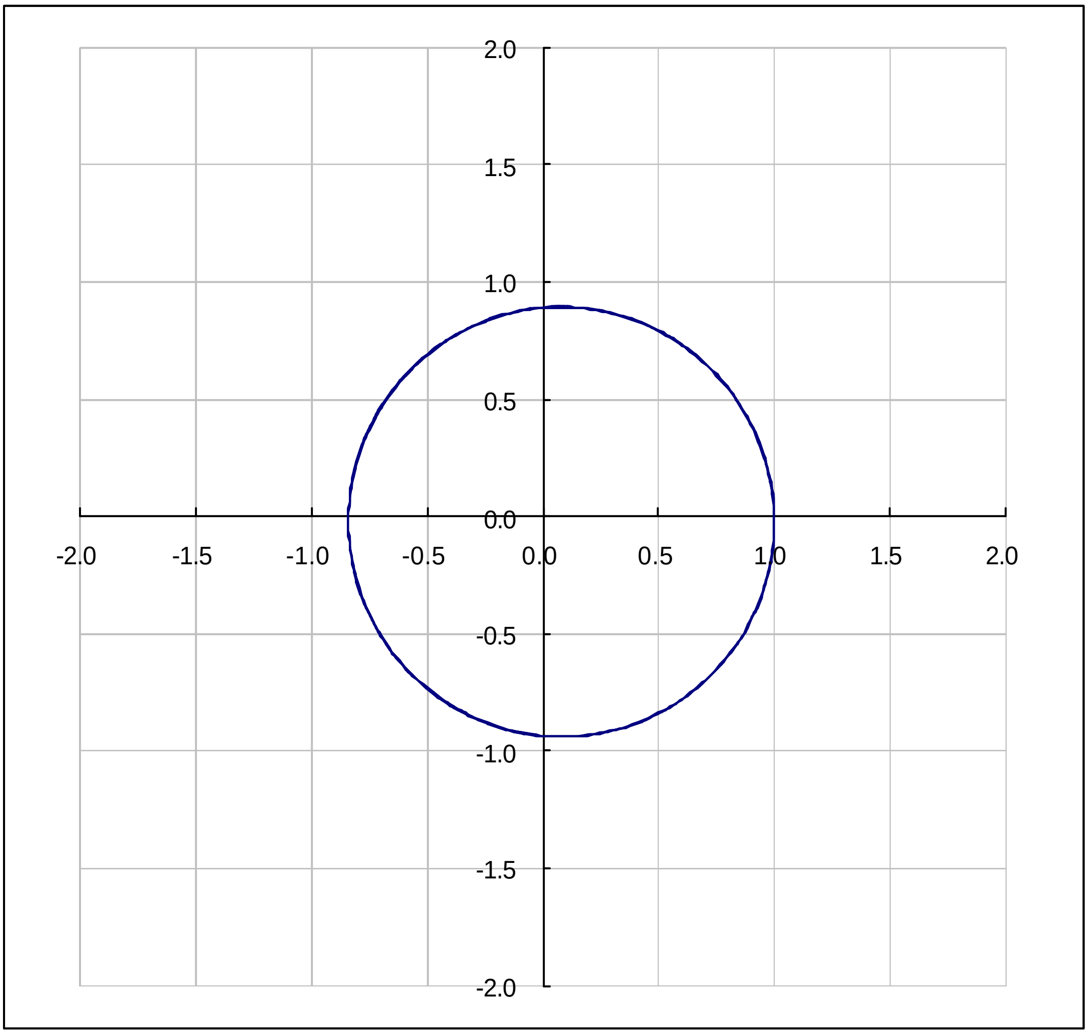

Figure 1

t	*x*	*y*	*E*	*L*
0.000	1.000000000	0.000000000	-21.47841760	6.00000000
0.008	0.997473381	0.048000000	-21.48279754	6.00000000
0.016	0.992416123	0.095900000	-21.48715396	6.00000000
0.024	0.984828985	0.143512029	-21.49148330	6.00000000
0.032	0.974717613	0.190777999	-21.49578157	6.00000000
0.040	0.962092664	0.237551994	-21.50004423	6.00000000
0.048	0.946969936	0.283709259	-21.50426618	6.00000000
0.056	0.929370516	0.329124514	-21.50844163	6.00000000
0.064	0.909320938	0.373672078	-21.51256409	6.00000000
0.072	0.886853355	0.417225998	-21.51662627	6.00000000

Chart 1

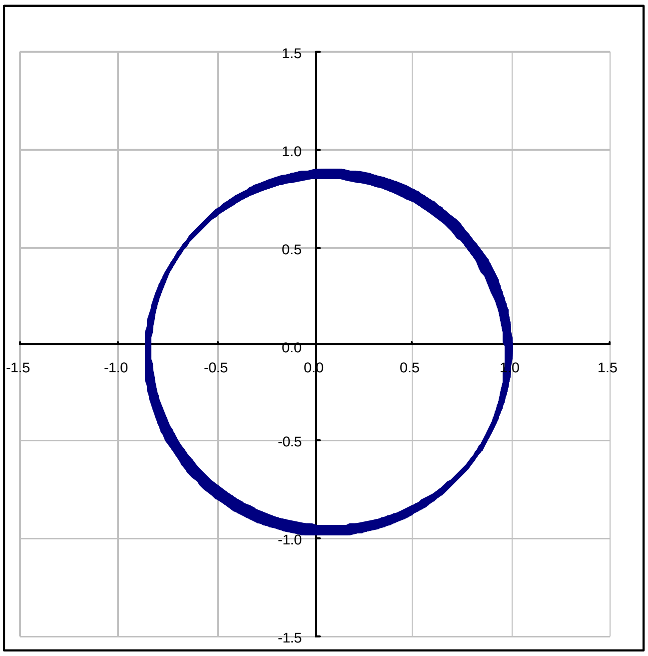

Figure 2

12500 Loops

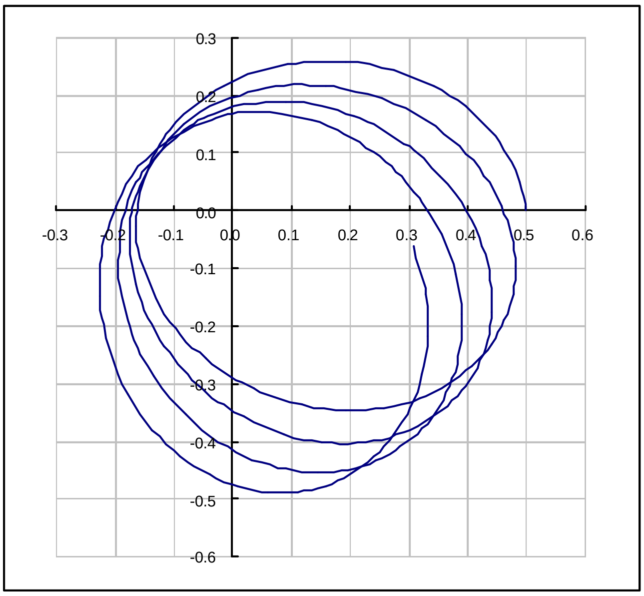

Figure 3

Initial conditions: $x = 0.5$, $y = 0$, $v_x=0$, $v_y=6$

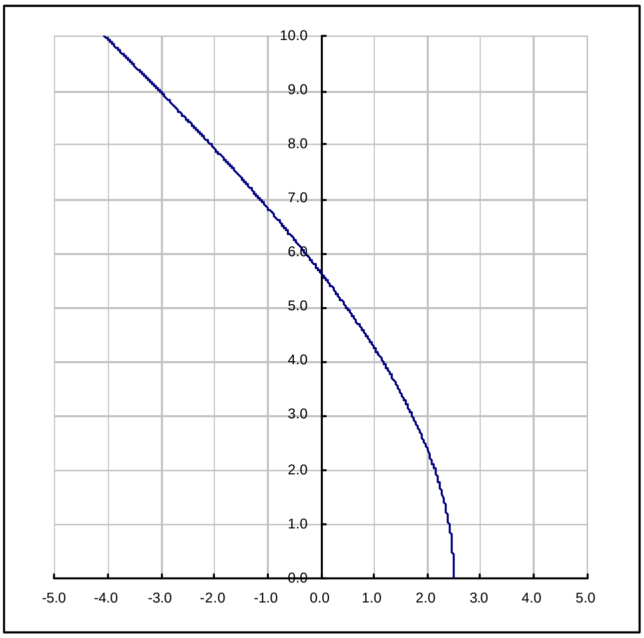

Figure 4

Initial conditions: $x = 2.5$, $y = 0$, $v_x=0$, $v_y=6$

Fortran Program

```
PROGRAM Planetary_Motion

INTEGER :: i, n
REAL(8) :: x_x, x_y, v_x, v_y, a_x, a_y, E, L, dt
REAL(8), PARAMETER :: pi = 3.14159265358979323846264338327950Q0

dt = 0.008Q0
n  = 5

x_x = 1.Q0
x_y = 0.Q0
v_x = 0.Q0
v_y = 6.Q0

OPEN(UNIT=10, FILE='result.dat', STATUS='REPLACE')

DO i=0, CEILING(REAL(n,8)/dt)

    a_x = -x_x*4*pi**2/SQRT(x_x**2+x_y**2)**3
    a_y = -x_y*4*pi**2/SQRT(x_x**2+x_y**2)**3

    E = (v_x**2+v_y**2)/2.Q0 - 4*pi**2/SQRT(x_x**2+x_y**2)
    L = x_x*v_y - x_y*v_x

    WRITE(10, '(4F20.14)') x_x, x_y, E, L

    v_x = v_x + a_x*dt
    v_y = v_y + a_y*dt

    x_x = x_x + v_x*dt
    x_y = x_y + v_y*dt

END DO

END PROGRAM
```