FYS-MEK1110 - Mandatory assignment 2

William Dugan

March 5, 2022

1 Ball on a spring

a Free-body diagram

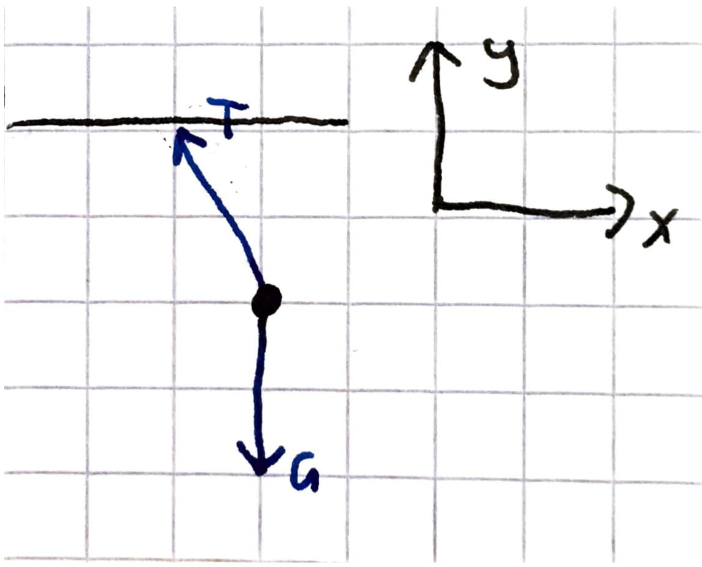

Figure 1: Force due to gravity G and tension in spring T on ball.

b Net force

We have $\vec{F_{net}} = \sum \vec{F} = \vec{G} + \vec{T}$. Since we define our coordinate system as shown in Figure 1, we identify G to be $-mg$ in the opposite direction of our y-axis. The tension in the string is given by Hooke's law and is $k * \Delta x$ in the opposite direction of our position vector. Hence

$$\vec{F_{net}} = \sum \vec{F} = -mg\vec{j} - k(r - L_0)\frac{\vec{r}}{r}. \tag{1}$$

c Component forces

With $r = \sqrt{x^2(t) + y^2(t)}$ we get

$$\begin{aligned}
\vec{F_{net}} &= -mg\vec{j} - k(\sqrt{x^2+y^2} - L_0)\frac{x\vec{i} + y\vec{j}}{\sqrt{x^2+y^2}} \\
&= \left(-k(\sqrt{x^2+y^2} - L_0)\frac{x(t)}{\sqrt{x^2+y^2}}\right)\vec{i} + \left(-mg\vec{j} - k(\sqrt{x^2+y^2} - L_0)\frac{y(t)}{\sqrt{x^2+y^2}}\right)\vec{j}
\end{aligned}$$

If we split this into its separate components we get

$$F_x = \left[-k\left(1 - \frac{L_0}{\sqrt{x^2(t)+y^2(t)}}\right)x(t)\right]\vec{i} \tag{2}$$

$$F_y = \left[-mg - k\left(1 - \frac{L_0}{\sqrt{x^2(t)^2(t)}}\right)y(t)\right]\vec{j} \tag{3}$$

d Position expressed by θ

If we were to express the position of the ball by using polar coordinates given by θ and r instead of Cartesian coordinates, we would need to know the length of the spring as well. This means that the angle θ does not give a sufficient description of the balls position.

e No movement nor acceleration

If $\theta = 0$ and $\vec{v} = \vec{a} = \vec{0}$, the ball would simply be resting at its equilibrium position. This position is given by $\vec{r} = (0, -L_0)$.

f Expressing the acceleration

From Newton's second law we have

$$\sum \vec{F} = m\vec{a} \tag{4}$$

We get the acceleration by dividing equation 1 by the ball's mass. This gives us

$$\vec{a_{net}} = \frac{\vec{F_{net}}}{m} = -g\vec{j} - \frac{k}{m}\left(1 - \frac{L_0}{r}\right)\vec{r}. \tag{5}$$

The components of acceleration in x and y direction is easily found by dividing F_x and F_y by the ball's mass, giving us

$$a_x = -\frac{k}{m}\left(1 - \frac{L_0}{\sqrt{x^2(t)+y^2(t)}}\right)x(t) \tag{6}$$

$$a_y = -g - \frac{k}{m}\left(1 - \frac{L_0}{\sqrt{x^2(t)+y^2(t)}}\right)y(t) \tag{7}$$

g Differential equation for $\vec{a}(t)$

We want to solve the following equations using the Euler-Cromer method

$$\begin{aligned} v(t+\Delta t) &= v(t) + a(t)\Delta t \\ r(t+\Delta t) &= r(t) + v(t)\Delta t \\ &= r(t) + v(t+\Delta t)\Delta t. \end{aligned}$$

To solve this we need the initial positions, velocities and acceleration (as well as $t(0) = t_0$). The initial velocity is $\vec{v} = \vec{0}$. The initial position is given by

$$r_0 = (L_0 sin(\theta_0), -L_0 cos(\theta_0)) = (sin(\pi/6), -cos(\pi/6)).$$

We will calculate the initial acceleration at the start of our integration loop, so there is no need to perform any further calculations.

h Numerical solution to differential equation

```
import numpy as np
import matplotlib.pyplot as plt

m = 0.1             # kg
L0 = 1              # m
k = 200             # N/m
g = 9.81            # m/s^2

theta0 = np.pi/6    # rad
r0 = L0*np.sin(theta0), -L0*np.cos(theta0)

T = 10
dt = 0.001
n = int(np.ceil(T/dt))

t = np.linspace(0, T, n)
r = np.zeros((n, 2)); r[0] = r0
v = np.zeros_like(r)
a = np.zeros_like(r)

for i in range(n-1):
    r_ = np.linalg.norm(r[i])
    a[i] = -k*(1-L0/r_)/m * r[i,0], -g-k*(1-L0/r_)/m * r[i, 1]
    v[i+1] = v[i] + a[i]*dt
    r[i+1] = r[i] + v[i+1]*dt

plt.plot(r[:,0], r[:,1])
plt.xlabel('x [m]')
plt.ylabel('y [m]')
plt.show()
```

i Results from program

The program is set to run with $\Delta t = 0.001$s for 10s.

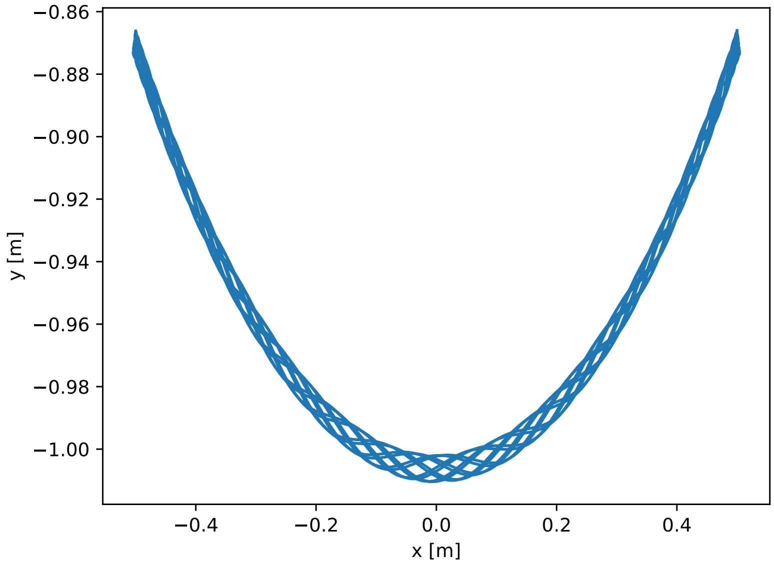

Figure 2: Plot of x, y for $t \in [0, 10]$

In Figure 2 we can see that the ball oscillates in an expected manner. The movement in the x direction is what we would expect from a standard pendulum on a string, but we can clearly see that it oscillates in the y-direction as well. This is what created the webbing / knitting pattern we observe.

j Changes to Δt

With $\Delta t = 0.01$s it seems as the time taken for a vertical oscillation decreases, giving us a tighter webbing pattern, but the motion is generally the same as before. Using $\Delta t = 0.1$s I receive an Runtime Warning: Overflow and the resulting plot is a straight line with $x, y >> 10^{200}$. I chose to not implement Euler's method as it is proven to be far less accurate when dealing with periodic motion.

k Changes to spring constant k

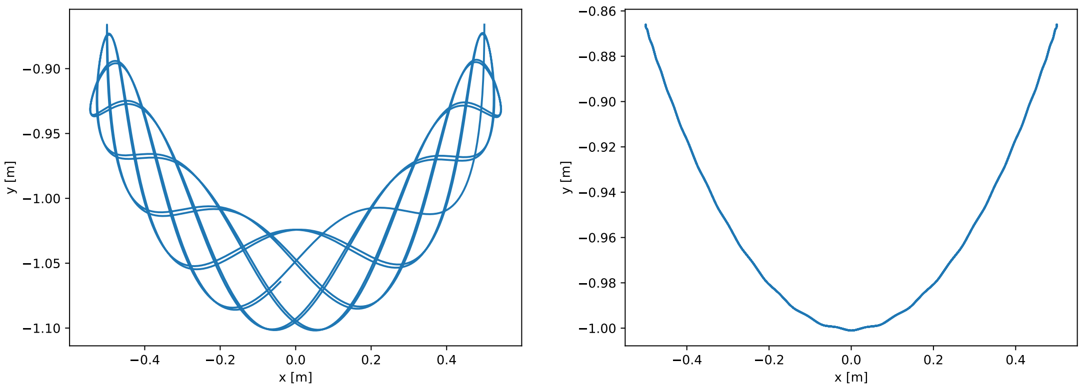

Figure 3: Plot of x, y for $t \in [0, 10]$ with $k = 20$ and $k = 2000$ respectively.

Figure 3 shows how a stiffer spring (higher k) results in a motion more similar to a pendulum on a non-elastic string. When trying to run the program with $k = 2 * 10^6$ I received another Runtime Warning: Overflow. This shows that even though our model will be closer to a non-elastic spring by simply increasing the spring constant, it lacks in efficiency and the model becomes inaccurate at some point.