Computer Model of a Spring-Mass System

OBJECTIVES

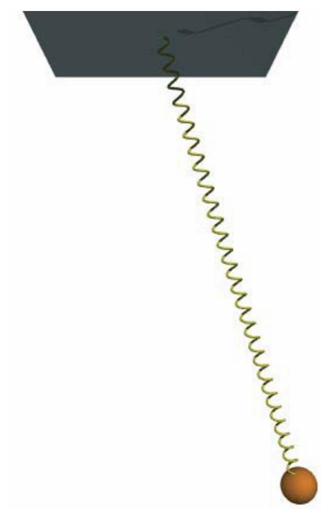
In this activity you will use an iterative computer model to predict the period of oscillation of a mass-spring system. You will also create graphs that display position and energy as a function of time.

Using the VIDLE editor, open the shell file, "Model3_Shell.py" and save it on your computer as "Model3.py".

PLANNING

1. Forces acting on the mass

The picture at right shows a snapshot of a mass oscillating on a spring at a particular instant. The mass is moving at this instant.



In this model, there are two forces acting on the mass: the **force of the spring** and the **force of gravity**. Our motion loop will need to calculate both of these in order to predict the motion of the mass.

2. Outline of the program

Let us recall the typical structure of an iterative calculation program exhibited by the program shell.

`## CONSTANTS`

Define any constants of nature.

`## OBJECTS AND INITIAL VALUES`

Here we create the ceiling, spring, and ball, as well as assign initial values like the unstretched spring length, mass of the ball, and so on. We also calculate the period analytically for later comparison, using the formulas

$$\omega=\sqrt{\frac{k_s}{m}} \text{ and } T=\frac{2\pi}{\omega}$$

`deltat` is defined to be 1/10000 times the period for accurate calculation.

`## DISPLAY AND GRAPHS`

This section creates windows and initializes graphs. You will not need to make any changes to this section

`## CALCULATIONS`

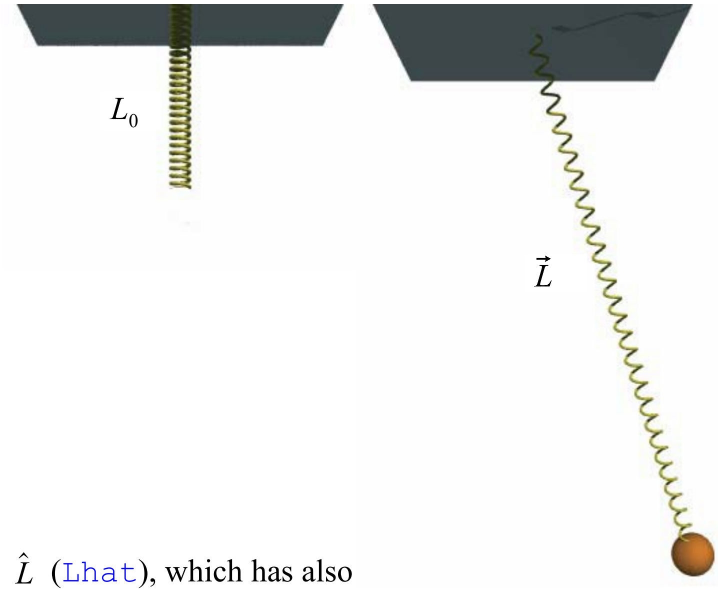
The iterative loop goes here. Inside the loop, we will update the forces, the momentum, the position, various energies, and time.

Think about the following question: In the cart program, we dealt with several constant forces (Fgrav and Ffan). These forces could have been defined either outside or inside of the loop, the latter choice telling the program to assign the same values many times. For our current spring-mass program, why must the definitions of forces be inside the loop? What is different about spring-mass physics?

Edited by M. Haley

3. Calculating the spring force as a vector

Consider the diagrams at right. When the spring is stretched as shown on the far right, the force on the ball due to the spring is directed along the spring, toward the point where the spring is attached to the ceiling.



The vector $\vec{L}$

We define a vector pointing from the attachment point (spring.pos) to the center of the ball (ball.pos), as shown in the right diagram. The VPython statement to define the vector $\vec{L}$ is

```
L = ball.pos - ceiling.pos
```

and the direction of $\vec{L}$ is given by the unit vector $\hat{L}$ (`Lhat`), which has also been defined.

The stretch is a scalar value, which in this case is positive, since the length of the spring is currently longer than its relaxed length.

$$s = |\vec{L}| - L_0$$

For positive values of *s*, the spring force is in the direction of $-\hat{L}$ (toward the ceiling). So the force on the ball due to the spring is:

$$\vec{F}_{spring} = -(k_s s)\hat{L}$$

For negative values of *s*, the spring is compressed, and the spring force is in the direction of $+\hat{L}$ (away from the ceiling). Since *s* is a negative number, the same equation describes the spring force:

$$\vec{F}_{spring} = -(k_s s)\hat{L}$$

Write the code to define `Fspring` using the above equation.

4. Kinetic, potential, and total energy

Following the definition of the forces is the momentum principle and position update, as we have seen in our previous model. Now we will calculate the various energies present in the system so that we may graph these quantities as a function of time.

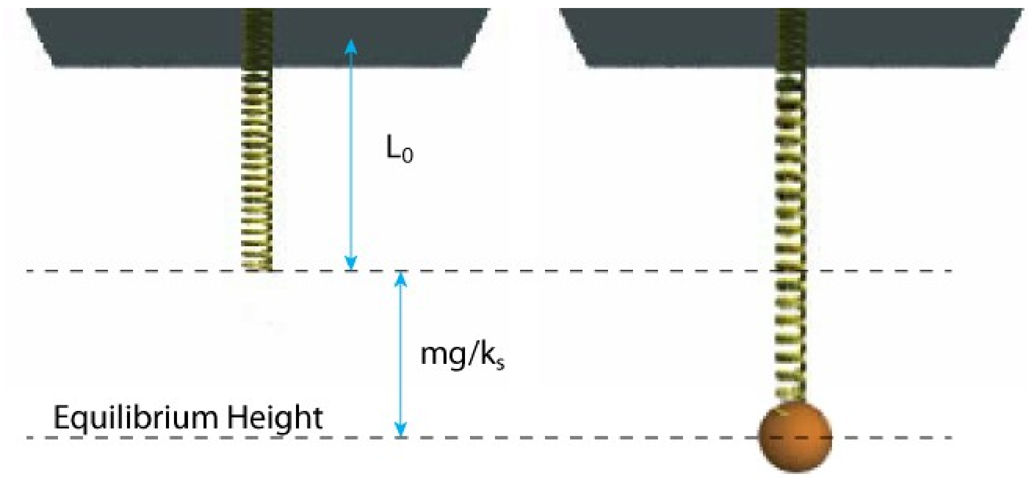
Note that in addition to kinetic energy, there are **two** kinds of potential energy present: gravitational and spring. Write the lines of code necessary to define each of these three energies. Since $\vec{v} = \vec{p}/m$, kinetic energy can also be written as $K = \frac{1}{2}mv^2 = \frac{p^2}{2\text{m}}$. You may find it useful to use the following syntax for exponentiation: `x**y` for x^y. For height in the gravitational potential energy near the surface of Earth equation, use the vertical position of the ball (`ball.pos.y`).

The code below what you have just written will graph these quantities.

5. Using graphs to view position versus time and energy versus time

If you have done everything correctly so far, you should see three windows open when you run your program. These are the scene window depicting the spring and mass, the position versus time graph, and energy versus time graphs. As each of these graphs is updated in sync with the scene, you should verify the graphs are doing what you expect as the mass oscillates up and down.

6. Equilibrium Point



The equilibrium point of the system is where the net force on the mass is zero. One difference between the horizontal mass-spring system and the vertical mass-spring system is the location of the equilibrium point. For the horizontal mass-spring, the equilibrium point is at L_0. However, since the vertical mass has weight, the spring must stretch by some amount to cancel the force of gravity. We can solve for this height by using Hooke's Law:

$$0 = F_{net,y} = k_s s - mg$$
$$s = mg/k_s$$

The vertical oscillation of the mass will be centered around this point.

7. Reflection

At this point, you may begin answering the reflection questions for Model 3.

Remember that you only get one try per question on a reflection assignment, so be sure discuss with your group or an instructor before answering!