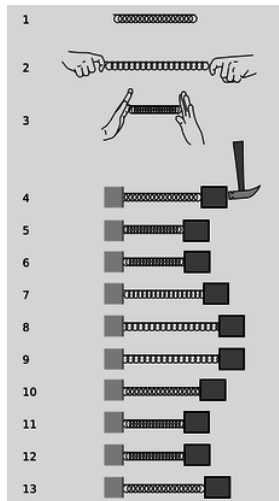


3.5: Oscillations

Let's revisit the example of the stretched spring from the previous section. We know that its energy is a form of electrical energy of interacting atoms, which is nice conceptually but doesn't help us to solve problems, since we don't know how the energy, *[Math Processing Error]*, depends on the length of the spring. All we know is that there's an equilibrium (figure a/1), which is a local minimum of the function *[Math Processing Error]*. An extremely important problem which arises in this connection is how to calculate oscillatory motion around an equilibrium, as in a/4-13. Even if we did special experiments to find out how the spring's energy worked, it might seem like we'd have to go through just as much work to deal with any other kind of oscillation, such as a sapling swinging back and forth in the breeze.



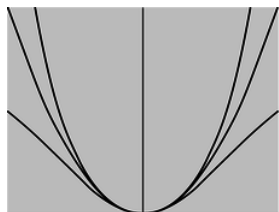
a / The spring has a minimum-energy length, 1, and energy is required in order to compress or stretch it, 2 and 3. A mass attached to the spring will oscillate around the equilibrium, 4-13.

Surprisingly, it's possible to analyze this type of oscillation in a very general and elegant manner, as long as the analysis is limited to *small* oscillations. We'll talk about the mass on the spring for concreteness, but there will be nothing in the discussion at all that is restricted to that particular physical system. First, let's choose a coordinate system in which *[Math Processing Error]* corresponds to the position of the mass where the spring is in equilibrium, and since interaction energies like *[Math Processing Error]* are only well defined up to an additive constant, we'll simply define it to be zero at equilibrium:

[Math Processing Error]

Since *[Math Processing Error]* is an equilibrium, *[Math Processing Error]* must have a local minimum there, and a differentiable function (which we assume *[Math Processing Error]* is) has a zero derivative at a local minimum:

[Math Processing Error]



b / Three functions with the same curvature at *[Math Processing Error]*=0.

There are still infinitely many functions that could satisfy these criteria, including the three shown in figure b, which are *[Math Processing Error]*, *[Math Processing Error]*, and *[Math Processing Error]*. Note, however, how all three functions are virtually identical right near the minimum. That's because they all have the same curvature. More specifically, each function has its second derivative equal to 1 at *[Math Processing Error]*, and the second derivative is a measure of curvature. We write *[Math Processing Error]* for the second derivative of the energy at an equilibrium point,

[Math Processing Error]

Physically, k is a measure of stiffness. For example, the heavy-duty springs in a car's shock absorbers would have a high value of k . It is often referred to as the spring constant, but we're only using a spring as an example here. As shown in figure b, any two functions that have k , k , and k , with the same value of k , are virtually indistinguishable for small values of k , so if we want to analyze small oscillations, it doesn't even matter which function we assume. For simplicity, we'll just use k from now on.

Now we're ready to analyze the mass-on-a-spring system, while keeping in mind that it's really only a representative example of a whole class of similar oscillating systems. We expect that the motion is going to repeat itself over and over again, and since we're not going to include frictional heating in our model, that repetition should go on forever without dying out. The most interesting thing to know about the motion would be the period, T , which is the amount of time required for one complete cycle of the motion. We might expect that the period would depend on the spring constant, k , the mass, m , and the amplitude, A , defined in figure c.¹¹



c / The amplitude would usually be defined as the distance from equilibrium to one extreme of the motion, i.e., half the total travel.

In examples like the brachistochrone and the Apollo 11 mission, it was generally necessary to use numerical techniques to determine the amount of time required for a certain motion. Once again, let's dust off the time3 function from page 93 and modify it for our purposes. For flexibility, we'll define the function $u(k, x)$ as a separate Python function. We really want to calculate the time required for the mass to come back to its starting point, but that would be awkward to set up, since our function works by dividing up the distance to be traveled into tiny segments. By symmetry, the time required to go from one end to the other equals the time required to come back to the start, so we'll just calculate the time for half a cycle and then double it when we return the result at the end of the function. The test at lines 16-19 is necessary because otherwise at the very end of the motion we can end up trying to take the square root of a negative number due to rounding errors.

```
import math
def u(k,x):
    return .5*k*x**2

def osc(m,k,a,n):
    x=a
    v=0
    dx = -2.*a/n
    t=0
    e = u(k,x)+.5*m*v**2
    for i in range(n):
        x_old = x
        v_old = v
        x = x+dx
        kinetic = e-u(k,x)
        if kinetic<0. :
            v=0.
            print "warning, K=",kinetic,"<0"
        else :
            v = -math.sqrt(2.*kinetic/m)
        v_avg = (v+v_old)/2.
        dt=dx/v_avg
        t=t+dt
    return 2.*t
```

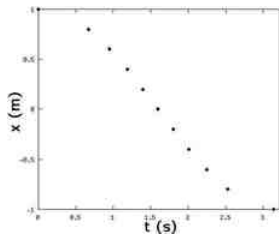
```
>>> print(osc(1.,1.,1.,100000))
warning, K= -1.43707268307e-12 <0
6.2831854132667919
```

The first thing to notice is that with this particular set of inputs ($m=1$ kg, $\omega=1$, and $x_0=1$), the program has done an excellent job of computing T . This is Mother Nature giving us a strong hint that the problem has an algebraic solution, not just a numerical one. The next interesting thing happens when we change the amplitude from 1 m to 2 m:

```
>>> print(osc(1.,1.,2.,100000))
warning, K= -5.7482907323e-12 <0
6.2831854132667919
```

Even though the mass had to travel double the distance in each direction, the period is the same to within the numerical accuracy of the calculation!

With these hints, it seems like we should start looking for an algebraic solution. For guidance, here's a graph of $x(t)$ as a function of t , as calculated by the osc function with $n=10$.

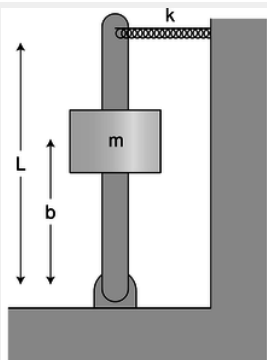


This looks like a cosine function, so let's see if $x(t) = A \cos(\omega t + \phi)$ is a solution to the conservation of energy equation --- it's not uncommon to try to “reverse-engineer” the cryptic results of a numerical calculation like this. The symbol ω (Greek omega), called angular frequency, is a standard symbol for the number of radians per second of oscillation. Except for the factor of ω , it is identical to the ordinary frequency f , which has units of f or Hz (Hertz). The phase angle ϕ is to allow for the possibility that $x(t)$ doesn't coincide with the beginning of the motion. The energy is

$E = \frac{1}{2} k A^2$

According to conservation of energy, this has to be a constant. Using the identity $\cos^2 \theta + \sin^2 \theta = 1$, we can see that it will be a constant if we have $\omega^2 = k/m$, or $\omega = \sqrt{k/m}$, i.e., $\omega = \sqrt{k/m}$. Note that the period is independent of amplitude.

Example 23: A spring and a lever



d / Example 23. The rod pivots on the hinge at the bottom.

[Math Processing Error] What is the period of small oscillations of the system shown in the figure? Neglect the mass of the lever and the spring. Assume that the spring is so stiff that gravity is not an important effect. The spring is relaxed when the lever is vertical.

[Math Processing Error] This is a little tricky, because the spring constant *[Math Processing Error]*, although it is relevant, is *not* the *[Math Processing Error]* we should be putting into the equation *[Math Processing Error]*. The *[Math Processing Error]* that goes in there has to be the second derivative of *[Math Processing Error]* with respect to the position, *[Math Processing Error]*, of the mass that's moving. The energy *[Math Processing Error]* stored in the spring depends on how far the *tip* of the lever is from the center. This distance equals *[Math Processing Error]*, so the energy in the spring is

[Math Processing Error]

and the *[Math Processing Error]* we have to put in *[Math Processing Error]* is

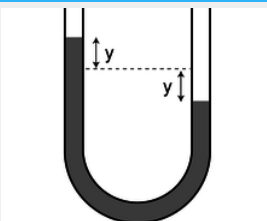
[Math Processing Error]

The result is

[Math Processing Error]

The leverage of the lever makes it as if the spring was stronger, and decreases the period of the oscillations by a factor of *[Math Processing Error]*.

Example 24: Water in a U-shaped tube



e / Water in a U-shaped tube.

[Math Processing Error] What is the period of oscillation of the water in figure e?

[Math Processing Error] In example 13 on p. 89, we found *[Math Processing Error]*, so the “spring constant,” which really isn't a spring constant here at all, is

[Math Processing Error]

This is an interesting example, because *[Math Processing Error]* can be calculated without any approximations, but the kinetic energy requires an approximation, because we don't know the details of the pattern of flow of the water. It could be very complicated. There will be a tendency for the water near the walls to flow more slowly due to friction, and there may also be swirling, turbulent motion. However, if we make the approximation that all the water moves with the same velocity as the surface, *[Math Processing Error]*, then the mass-on-a-spring analysis applies. Letting *[Math Processing Error]* be the total length of the filled part of the tube, the mass is *[Math Processing Error]*, and we have

[Math Processing Error]

Contributors and Attributions

Benjamin Crowell (Fullerton College). *Conceptual Physics* is copyrighted with a CC-BY-SA license.

This page titled 3.5: Oscillations is shared under a CC BY-SA license and was authored, remixed, and/or curated by Benjamin Crowell.