BK2-Physics: Simulations, 2021

Thomas Schrøder

January 15, 2024

Contents

1 A few math exercises:

Below are some math exercises you should do as home-work before session 2. It is math that will probably be useful in your data analysis - in this course and later in your studies. If you think they are trivial - that's fine. If you need to look them up in your high school math book, google or ask a friend - that's fine too...

- A: $\log_{10}(10) =$
- B: $\log_{10}(100) =$
- C: $\log_{10}(1000) =$
- D: $\log_{10}(0.1) =$
- E: $\log(ax) =$
- F: $\log(x^b) =$
- G: $\log(ax^b) =$
- H: $1/x = x^a$. What is a?:
- I: $\sqrt{x} = x^a$. What is a?:
- J: $\frac{1}{\sqrt{x}} = x^a$. What is a?:
- K: Plotting y vs. x gives a straight line. $y =$
- L: Plotting y vs. $\log(x)$ gives a straight line. $y =$
- M: Plotting $\log(y)$ vs. x gives a straight line. $y =$
- N: Plotting $\log(y)$ vs. $\log(x)$ gives a straight line. $y =$

2 Simulation of objects in 1 dimension

In this section you will learn how to simulate the motion of an object in 1 dimension. In the following section you will use this knowledge to simulate a mass attached to a spring using Python This simulation will be part of your mini-project. If you need to brush up on your Python-skills you can revisit your BK1-notes.

To see how to do the simulations, three concepts are needed, *velocity*, *acceleration* and *Newton's second law*. We will briefly go through these below. If you need more information on these subjects you can look at e.g. the textbook "Physics" by Ohanian, p26-34 (2. ed.).

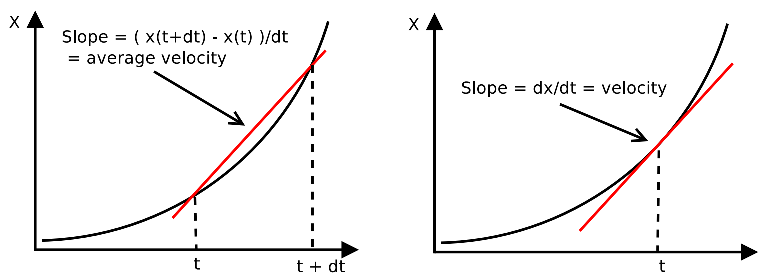
2.1 Velocity

Velocity is the change of position per time. Let $x(t)$ be the position of an object at the time t and $x(t+\Delta t)$ the position after some time interval Δt. The average velocity, $\bar{v}$, is then given by:

$$\bar{v} = \frac{x(t+\Delta t) - x(t)}{(t+\Delta t) - t} = \frac{x(t+\Delta t) - x(t)}{\Delta t}. \tag{1}$$

If we let the time interval Δt be very small, we get the *instantaneous* velocity, v, which is the position differentiated with respect to time, i.e. the slope of the position plotted as a function of time:

$$v = \frac{dx}{dt}. \tag{2}$$



2.2 Acceleration

Acceleration is the change of velocity per time. The average acceleration in a time interval is thus:

$$\bar{a} = \frac{v(t+\Delta t) - v(t)}{\Delta t}, \tag{3}$$

and the instantaneous acceleration, a, is:

$$a = \frac{dv}{dt} \tag{4}$$

2.3 Newton's second law

Newton's second law says that the total force, F, on an object is equal to its mass, m, times its acceleration:

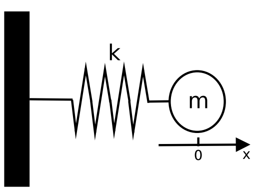
$$F = ma \tag{5}$$

When doing simulations, we determine the force from our knowledge of the system and then use Eq. (5) to determine the acceleration,

$$a = F/m. \tag{6}$$

From the acceleration we can then calculate the change of velocity and from the velocity we can calculate the change of position (this will be detailed below).

2.4 Modelling a mass attached to a spring



When we write up which forces that are present in a system, we are actually making a *model* of the system. You might have seen Hooke's Law for the restoring force of a spring:

$$F = -kx \tag{7}$$

where x is position measured relative to the resting position of the spring and k is the "spring constant".

For a given physical spring, Eq. (7) holds only up to a certain value of x. Beyond this displacement the actual force is often less than the one given by Eq. (7). Thus Eq. (7) is a *model* of the spring; a model that works well for not-too-large displacements. Eq. (7) is a good model for many other systems than just physical springs, some of which you will encounter in this course. Because of its usefulness, the model is also given a more fancy name, "the harmonic oscillator".

2.5 Simulation: The Leap-Frog method

Here we will develop the method for simulation called the "Leap-Frog" method. As a starting point for using the method, one needs to know

- t_0 - the time at which the simulation starts (often set to zero)
- x_0 - the initial position, i.e. the position at time t_0
- v_0 - the initial velocity, i.e. the velocity at time t_0

This information as a whole is called "the initial conditions."

The Leap-Frog method works by making a number of "timesteps" of length Δt. In other words the time is discretized, and the time at the i'th timestep is:

$$t_i = i \cdot \Delta t \tag{8}$$

Given the initial conditions mentioned above and the model of the system, one can calculate the corresponding values of t, x, and v a time step Δt later. An important 'trick' is to use the average velocity in a time interval as the instantaneous velocity in the *middle* of the time interval (which we here will call $v_{i+\frac{1}{2}}$):

$$\begin{aligned} v_{i+\frac{1}{2}} &= \frac{x_{i+1} - x_i}{\Delta t} \quad \Leftrightarrow & (9) \\ x_{i+1} &= x_i + v_{i+\frac{1}{2}} \cdot \Delta t & (10) \end{aligned}$$

In similar manner we can write the acceleration:

$$\begin{aligned} a_i &= \frac{v_{i+\frac{1}{2}} - v_{i-\frac{1}{2}}}{\Delta t} \quad \Leftrightarrow & (11) \\ v_{i+\frac{1}{2}} &= v_{i-\frac{1}{2}} + a_i \cdot \Delta t & (12) \end{aligned}$$

We have now developed the Leap-Frog method, which can be written:

$$\begin{aligned} t_{i+1} &= t_i + \Delta t & (13) \\ v_{i+\frac{1}{2}} &= v_{i-\frac{1}{2}} + a_i \cdot \Delta t & (14) \\ x_{i+1} &= x_i + v_{i+\frac{1}{2}} \cdot \Delta t & (15) \end{aligned}$$

Note that the velocity is updated before the position. Why?

2.6 The time step, Δt

It is important to understand that one makes a small error every time the Leap-Frog method is used to calculate a new time step. This is because the average velocity, Eq. 9, clearly is only an approximation of the instantaneous velocity ($v \equiv dx/dt$). The magnitude of the error is in general dependent on the time step, Δt; a smaller Δt leads to a smaller error.

Usually one wants to calculate the positions, $x(t)$, and the velocities, $v(t)$, to a given precision in a given time interval. How do we find out if the simulation is precise enough? This is easy if we know the analytical solution: we just compare the simulation to the analytical result. In fact, one should always test the program with a simulation where we *do* know the analytical result. However the reason for doing a simulation is usually that we have a problem that we do *not* know the analytical solution to. In that case one can do the following: Make a simulation with the time step Δt and another simulation with a time step that is e.g., a factor of 10 smaller. Now, compare the two results: If they are identical to the precision we want, then we are done and we can trust the results. If the results are not similar enough, we try again with even smaller time steps.

2.7 Leap-Frog using Python

Even for simple equations of motion, the Leap-Frog method requires a lot of calculations, and it is therefore only practical if a computer is used to do them. The python function below implements the method in Python:

```
def LeapFrog(m, k, dt, t_end, v0, x0):
    """ Run a numerical simulation of a mass and a spring using Leap-Frog.
    Parameters:
        m:
        k:
        dt:
        t_end:
        v0:
        x0:
    Returns:
        ttable:
        xtable:
        vtable:
    """

    n_steps = int(t_end/dt)  # Number of steps

    ttable = np.zeros(n_steps)
    xtable = np.zeros(n_steps)
    vtable = np.zeros(n_steps)

    t = 0.0
    x = x0
    v = v0

    for i in range(n_steps):
        ttable[i] = t
        xtable[i] = x
        vtable[i] = v

        t = t + dt
        a = ...
        v = v + a*dt  # a = dv/dt
        x = x + v*dt  # v = dx/dt


    return ttable,xtable,vtable
```

Exercise 1. Download the jupyter-notebook BKphysics.ipynb from moodle, and open it in Jupyter.

Exercise 2. Go through the Leapfrog function in the notebook and make sure you understand the meaning of each line. Fill in the comments describing the parameters of the function.

Exercise 3. Put in the calculation of the acceleration at the appropiate place. This should explain to Python that it is a mass attached to a spring you want to simulate. **Hint:** Think about what physical laws are already in the function, and what is not (yet):

3 Simulating a mass attached to a spring

Exercise 4. Follow the instructions in the notebook, to use the LeapFrog function to simulate a mass attached to a spring, and plot the results. Choose a spring constant k and a mass m. Use for example the initial conditions $x_0 = 0.1$, $v_0 = 0$.

Investigate how small time steps are needed to get a satisfying solution of the equations of motion.

Do the results agree with the behavior you expect from a spring?

Exercise 5. What is the period of the oscillations of your spring (for your chosen values of k and m).

Exercise 6. What happens if you multiply k and m by the same number and run the simulation again. Why? **Hint:** Look at how k and m is used in your program.

Exercise 7. In general the period of the oscillations depends on both k and m. Given your results in Exercise 6, we can argue that the period actually depends only on a single parameter - what is that parameter?

__**Exercise 8.**__ Use simulations to answer the following research-question: How does the period of the oscillations, T, depend on k and m? Note, that we are seeking a *mathematical* expression descriping the connection between T, k, and m, not just the trends like "larger m leads to larger T". Think of this as being in the Lab and doing a lot of experiments (here: simulations), followed by carefull data-analysis to find a mathematical description of your data. For the data analysis you might find section 4 in this note useful.

Exercise 9. Compare your results with the analytical solution. To find the analytical simulation, first combine Eqs. (2), (4) and (7) to derive that the position as a function of time fulfills the differential equation:

$$x(t) = -\frac{m}{k}\frac{d^2x(t)}{dt^2} \tag{16}$$

Now, from your simulations, you might be able to "guess" a function $x(t)$ and show that it is consistent with this differential equation.

__**Exercise 10.**__ Use simulations to answer the following research-question: Under which conditions does the spring-mass system you been simulating exhibit a resonance? To test this, you should add a small extra oscillating force – we call the extra force a "pertubation" of the system:

```
Fe =  0.01*cos(2*pi*t/Te);
```

Te is the period of this extra force – it should be a new input-parameter to your program. Think of the extra force as mimicking the effect of the wind or pedastrians on a bridge. Modify your program so that this extra force is included in the simulation, and save it under a new name.

The following simulations should be purely *driven* oscillations, thus $x_0 = 0$ and $v_0 = 0$, and it will be useful to make simulations where t_{end} is eg. 50 times the natural period for your chosen m and k. Try varying the period of the extra force, Te – Is there a period which results in large effects even though the pertubation is small? Try plotting the maximum value of $x(t)$ versus Te. What is happening?

Exercise 11. (optional) Add friction to your spring. There are different ways to model friction. Assume that the friction force is proportional to the velocity. What effect does the friction have? (you should probably start by removing the pertubation from exercise 10).

4 Fitting by transforming to straight lines:

When fitting empirical data to a hypothesis formulated as a mathematical formula, it is very convenient if a transformation can be done so that the data according to the hypothesis should become a straight line. A fit a to straight line is very easy to judge by the human eye, and in many cases this is what you would plot in a scientific paper, to convince the reader that your hypothesis is fulfilled.

Below we show the mathematical transformation for the two most common cases.

4.1 Exponential functions

Lets assume that the hypothesis is that you have an exponential growth or decrease (depending on the sign of b):

$$y = a\exp(bx)$$

Now we take the logarithm on both sides of the equation:

$$\log(y) = \log(a\exp(bx)) = \log(a) + \log(\exp(bx)) = \log(a) + bx$$

From this we conclude that if we plot x on the first axis and $\log(y)$ on the second axis, we should get a straight line with slope b and the constant being $\log(a)$.

4.2 Power-law

Lets assume that the hypothesis is a power-law (in Danish: Potenslov):

$$y = ax^b$$

Again we take the logarithm on both sides of the equation:

$$\log(y) = \log(ax^b) = \log(a) + \log(x^b) = \log(a) + b\log(x)$$

From this we conclude that if we plot $\log(x)$ on the first axis and $\log(y)$ on the second axis we should get a straight line with slope b and the constant being $\log(a)$.