

# PHY321: More on Motion and Forces with examples, begin Work and Energy discussion

Morten Hjorth-Jensen<sup>1,2</sup>

<sup>1</sup>Department of Physics and Astronomy and Facility for Rare Ion Beams (FRIB), Michigan State University, USA

<sup>2</sup>Department of Physics, University of Oslo, Norway

Jan 23, 2023

## Aims and Overarching Motivation

**Monday January 23, Lecture and examples.** We discuss various forces and their pertinent equations of motion

Recommended reading: Taylor 2.1-2.4. Málthe-Sørenssen chapters 6 and 7 contain many interesting examples with codes and solutions. We will cover in particular a falling object in two dimensions with linear air resistance relevant for homework 3.

- [Video of lecture](#)

**Wednesday January 25, Lecture and Examples.** We discuss other force models with examples such as the gravitational force and a spring force. See Málthe-Sørenssen chapter 7.3-7.5.

We discuss also exercises 5 and 6 from homework 2. We will continue with this on Friday.

**Friday January 27, summary and work on weekly assignments.** We start our discussion of energy and work, see Taylor 4.1 and 4.2 the first 20 minutes. Thereafter we solve exercises.

## Air Resistance in One Dimension

Last week we considered the motion of a falling object with air resistance. Here we look at both a quadratic in velocity resistance and linear in velocity. But first we give a qualitative argument about the mathematical expression for the

air resistance we used last Friday. The discussion here is also contained in the [video on how to make a model for a drag force](#).

Air resistance tends to scale as the square of the velocity. This is in contrast to many problems chosen for textbooks, where it is linear in the velocity. The choice of a linear dependence is motivated by mathematical simplicity (it keeps the differential equation linear) rather than by physics. One can see that the force should be quadratic in velocity by considering the momentum imparted on the air molecules. If an object sweeps through a volume  $dV$  of air in time  $dt$ , the momentum imparted on the air is

$$dP = \rho_m dV v, \quad (1)$$

where  $v$  is the velocity of the object and  $\rho_m$  is the mass density of the air. If the molecules bounce back as opposed to stop you would double the size of the term. The opposite value of the momentum is imparted onto the object itself. Geometrically, the differential volume is

$$dV = A v dt, \quad (2)$$

where  $A$  is the cross-sectional area and  $v dt$  is the distance the object moved in time  $dt$ .

## Resulting Acceleration

Plugging this into the expression above,

$$\frac{dP}{dt} = -\rho_m A v^2. \quad (3)$$

This is the force felt by the particle, and is opposite to its direction of motion. Now, because air doesn't stop when it hits an object, but flows around the best it can, the actual force is reduced by a dimensionless factor  $c_W$ , called the drag coefficient.

$$F_{\text{drag}} = -c_W \rho_m A v^2, \quad (4)$$

and the acceleration is

$$\frac{dv}{dt} = -\frac{c_W \rho_m A}{m} v^2. \quad (5)$$

For a particle with initial velocity  $v_0$ , one can separate the  $dt$  to one side of the equation, and move everything with  $vs$  to the other side. We did this in our discussion of simple motion and will not repeat it here.

On more general terms, for many systems, e.g. an automobile, there are multiple sources of resistance. In addition to wind resistance, where the force is proportional to  $v^2$ , there are dissipative effects of the tires on the pavement, and in the axel and drive train. These other forces can have components that scale

proportional to  $v$ , and components that are independent of  $v$ . Those independent of  $v$ , e.g. the usual  $f = \mu_K N$  frictional force you consider in your first Physics courses, only set in once the object is actually moving. As speeds become higher, the  $v^2$  components begin to dominate relative to the others. For automobiles at freeway speeds, the  $v^2$  terms are largely responsible for the loss of efficiency. To travel a distance  $L$  at fixed speed  $v$ , the energy/work required to overcome the dissipative forces are  $fL$ , which for a force of the form  $f = \alpha v^n$  becomes

$$W = \int dx f = \alpha v^n L. \quad (6)$$

For  $n = 0$  the work is independent of speed, but for the wind resistance, where  $n = 2$ , slowing down is essential if one wishes to reduce fuel consumption. It is also important to consider that engines are designed to be most efficient at a chosen range of power output. Thus, some cars will get better mileage at higher speeds (They perform better at 50 mph than at 5 mph) despite the considerations mentioned above.

## Going Ballistic, Projectile Motion or a Softer Approach, Falling Raindrops

As an example of Newton's Laws we consider projectile motion (or a falling raindrop or a ball we throw up in the air) with a drag force. Even though air resistance is largely proportional to the square of the velocity, we will consider the drag force to be linear to the velocity,  $\mathbf{F} = -m\gamma\mathbf{v}$ , for the purposes of this exercise.

Such a dependence can be extracted from experimental data for objects moving at low velocities, see for example Malthe-Sørensen chapter 5.6.

We will here focus on a two-dimensional problem.

### Two-dimensional falling object

The acceleration for a projectile moving upwards,  $\mathbf{a} = \mathbf{F}/m$ , becomes

$$\begin{aligned} \frac{dv_x}{dt} &= -\gamma v_x, \\ \frac{dv_y}{dt} &= -\gamma v_y - g, \end{aligned} \quad (7)$$

and  $\gamma$  has dimensions of inverse time.

If you on the other hand have a falling raindrop, how do these equations change? See for example Figure 2.1 in Taylor. Let us stay with a ball which is thrown up in the air at  $t = 0$ .

### Ways of solving these equations

We will go over two different ways to solve this equation. The first by direct integration, and the second as a differential equation. To do this by direct

integration, one simply multiplies both sides of the equations above by  $dt$ , then divide by the appropriate factors so that the  $v$ s are all on one side of the equation and the  $dt$  is on the other. For the  $x$  motion one finds an easily integrable equation,

$$\begin{aligned}\frac{dv_x}{v_x} &= -\gamma dt, \\ \int_{v_{0x}}^{v_x} \frac{dv_x}{v_x} &= -\gamma \int_0^t dt, \\ \ln\left(\frac{v_x}{v_{0x}}\right) &= -\gamma t, \\ v_x(t) &= v_{0x}e^{-\gamma t}.\end{aligned}\tag{8}$$

This is very much the result you would have written down by inspection. For the  $y$ -component of the velocity,

$$\begin{aligned}\frac{dv_y}{v_y + g/\gamma} &= -\gamma dt \\ \ln\left(\frac{v_y + g/\gamma}{v_{0y} - g/\gamma}\right) &= -\gamma t_f, \\ v_{fy} &= -\frac{g}{\gamma} + \left(v_{0y} + \frac{g}{\gamma}\right)e^{-\gamma t}.\end{aligned}\tag{9}$$

Whereas  $v_x$  starts at some value and decays exponentially to zero,  $v_y$  decays exponentially to the terminal velocity,  $v_t = -g/\gamma$ .

## Solving as differential equations

Although this direct integration is simpler than the method we invoke below, the method below will come in useful for some slightly more difficult differential equations in the future. The differential equation for  $v_x$  is straight-forward to solve. Because it is a first order differential equation there is one arbitrary constant,  $A$ , and by inspection the solution is

$$v_x = Ae^{-\gamma t}.\tag{10}$$

The arbitrary constants for equations of motion are usually determined by the initial conditions, or more generally boundary conditions. By inspection  $A = v_{0x}$ , the initial  $x$  component of the velocity.

## Differential Equations, contn

The differential equation for  $v_y$  is a bit more complicated due to the presence of  $g$ . Differential equations where all the terms are linearly proportional to a function,

in this case  $v_y$ , or to derivatives of the function, e.g.,  $v_y$ ,  $dv_y/dt$ ,  $d^2v_y/dt^2 \dots$ , are called linear differential equations. If there are terms proportional to  $v^2$ , as would happen if the drag forces were proportional to the square of the velocity, the differential equation is not longer linear. Because this expression has only one derivative in  $v$  it is a first-order linear differential equation. If a term were added proportional to  $d^2v/dt^2$  it would be a second-order differential equation. In this case we have a term completely independent of  $v$ , the gravitational acceleration  $g$ , and the usual strategy is to first rewrite the equation with all the linear terms on one side of the equal sign,

$$\frac{dv_y}{dt} + \gamma v_y = -g. \quad (11)$$

### Splitting into two parts

Now, the solution to the equation can be broken into two parts. Because this is a first-order differential equation we know that there will be one arbitrary constant. Physically, the arbitrary constant will be determined by setting the initial velocity, though it could be determined by setting the velocity at any given time. Like most differential equations, solutions are not “solved”. Instead, one guesses at a form, then shows the guess is correct. For these types of equations, one first tries to find a single solution, i.e. one with no arbitrary constants. This is called the *particular* solution,  $y_p(t)$ , though it should really be called “a” particular solution because there are an infinite number of such solutions. One then finds a solution to the *homogenous* equation, which is the equation with zero on the right-hand side,

$$\frac{dv_{y,h}}{dt} + \gamma v_{y,h} = 0. \quad (12)$$

Homogenous solutions will have arbitrary constants.

The particular solution will solve the same equation as the original general equation

$$\frac{dv_{y,p}}{dt} + \gamma v_{y,p} = -g. \quad (13)$$

However, we don't need find one with arbitrary constants. Hence, it is called a **particular** solution.

The sum of the two,

$$v_y = v_{y,p} + v_{y,h}, \quad (14)$$

is a solution of the total equation because of the linear nature of the differential equation. One has now found a *general* solution encompassing all solutions, because it both satisfies the general equation (like the particular solution), and has an arbitrary constant that can be adjusted to fit any initial condition (like the homogeneous solution). If the equations were not linear, that is if there were terms such as  $v_y^2$  or  $v_y \dot{v}_y$ , this technique would not work.

## More details

Returning to the example above, the homogenous solution is the same as that for  $v_x$ , because there was no gravitational acceleration in that case,

$$v_{y,h} = Be^{-\gamma t}. \quad (15)$$

In this case a particular solution is one with constant velocity,

$$v_{y,p} = -g/\gamma. \quad (16)$$

Note that this is the terminal velocity of a particle falling from a great height. The general solution is thus,

$$v_y = Be^{-\gamma t} - g/\gamma, \quad (17)$$

and one can find  $B$  from the initial velocity,

$$v_{0y} = B - g/\gamma, \quad B = v_{0y} + g/\gamma. \quad (18)$$

Plugging in the expression for  $B$  gives the  $y$  motion given the initial velocity,

$$v_y = (v_{0y} + g/\gamma)e^{-\gamma t} - g/\gamma. \quad (19)$$

It is easy to see that this solution has  $v_y = v_{0y}$  when  $t = 0$  and  $v_y = -g/\gamma$  when  $t \rightarrow \infty$ .

One can also integrate the two equations to find the coordinates  $x$  and  $y$  as functions of  $t$ ,

$$\begin{aligned} x &= \int_0^t dt' v_{0x}(t') = \frac{v_{0x}}{\gamma} (1 - e^{-\gamma t}), \\ y &= \int_0^t dt' v_{0y}(t') = -\frac{gt}{\gamma} + \frac{v_{0y} + g/\gamma}{\gamma} (1 - e^{-\gamma t}). \end{aligned} \quad (20)$$

If the question was to find the position at a time  $t$ , we would be finished. However, the more common goal in a projectile equation problem is to find the range, i.e. the distance  $x$  at which  $y$  returns to zero. For the case without a drag force this was much simpler. The solution for the  $y$  coordinate would have been  $y = v_{0y}t - gt^2/2$ . One would solve for  $t$  to make  $y = 0$ , which would be  $t = 2v_{0y}/g$ , then plug that value for  $t$  into  $x = v_{0x}t$  to find  $x = 2v_{0x}v_{0y}/g = v_0 \sin(2\theta_0)/g$ . One follows the same steps here, except that the expression for  $y(t)$  is more complicated. Searching for the time where  $y = 0$ , and we get

$$0 = -\frac{gt}{\gamma} + \frac{v_{0y} + g/\gamma}{\gamma} (1 - e^{-\gamma t}). \quad (21)$$

This cannot be inverted into a simple expression  $t = \dots$ . Such expressions are known as “transcendental equations”, and are not the rare instance, but are the norm. In the days before computers, one might plot the right-hand side of

the above graphically as a function of time, then find the point where it crosses zero.

Now, the most common way to solve for an equation of the above type would be to apply Newton's method numerically. This involves the following algorithm for finding solutions of some equation  $F(t) = 0$ .

1. First guess a value for the time,  $t_{\text{guess}}$ .
2. Calculate  $F$  and its derivative,  $F(t_{\text{guess}})$  and  $F'(t_{\text{guess}})$ .
3. Unless you guessed perfectly,  $F \neq 0$ , and assuming that  $\Delta F \approx F' \Delta t$ , one would choose
4.  $\Delta t = -F(t_{\text{guess}})/F'(t_{\text{guess}})$ .
5. Now repeat step 1, but with  $t_{\text{guess}} \rightarrow t_{\text{guess}} + \Delta t$ .

If the  $F(t)$  were perfectly linear in  $t$ , one would find  $t$  in one step. Instead, one typically finds a value of  $t$  that is closer to the final answer than  $t_{\text{guess}}$ . One breaks the loop once one finds  $F$  within some acceptable tolerance of zero. A program to do this will be added shortly.

## Motion in a Magnetic Field

This case is just another example of motion under a specific force.

Another example of a velocity-dependent force is magnetism,

$$\begin{aligned}\mathbf{F} &= q\mathbf{v} \times \mathbf{B}, \\ F_i &= q \sum_{jk} \epsilon_{ijk} v_j B_k.\end{aligned}\tag{22}$$

For a uniform field in the  $z$  direction  $\mathbf{B} = B\hat{z}$ , the force can only have  $x$  and  $y$  components,

$$\begin{aligned}F_x &= qBv_y \\ F_y &= -qBv_x.\end{aligned}\tag{23}$$

The differential equations are

$$\begin{aligned}\dot{v}_x &= \omega_c v_y, \omega_c = qB/m \\ \dot{v}_y &= -\omega_c v_x.\end{aligned}\tag{24}$$

One can solve the equations by taking time derivatives of either equation, then substituting into the other equation,

$$\begin{aligned}\ddot{v}_x &= \omega_c \dot{v}_y = -\omega_c^2 v_x, \\ \ddot{v}_y &= -\omega_c \dot{v}_x = -\omega_c^2 v_y.\end{aligned}\tag{25}$$

The solution to these equations can be seen by inspection,

$$\begin{aligned}v_x &= A \sin(\omega_c t + \phi), \\v_y &= A \cos(\omega_c t + \phi).\end{aligned}\tag{26}$$

One can integrate the equations to find the positions as a function of time,

$$\begin{aligned}x - x_0 &= \int_{x_0}^x dx = \int_0^t dt v(t) \\&= \frac{-A}{\omega_c} \cos(\omega_c t + \phi), \\y - y_0 &= \frac{A}{\omega_c} \sin(\omega_c t + \phi).\end{aligned}\tag{27}$$

The trajectory is a circle centered at  $x_0, y_0$  with amplitude  $A$  rotating in the clockwise direction.

The equations of motion for the  $z$  motion are

$$\dot{v}_z = 0,\tag{28}$$

which leads to

$$z - z_0 = V_z t.\tag{29}$$

Added onto the circle, the motion is helical.

Note that the kinetic energy,

$$T = \frac{1}{2}m(v_x^2 + v_y^2 + v_z^2) = \frac{1}{2}m(\omega_c^2 A^2 + V_z^2),\tag{30}$$

is constant. This is because the force is perpendicular to the velocity, so that in any differential time element  $dt$  the work done on the particle  $\mathbf{F} \cdot d\mathbf{r} = dt \mathbf{F} \cdot \mathbf{v} = 0$ .

One should think about the implications of a velocity dependent force. Suppose one had a constant magnetic field in deep space. If a particle came through with velocity  $v_0$ , it would undergo cyclotron motion with radius  $R = v_0/\omega_c$ . However, if it were still its motion would remain fixed. Now, suppose an observer looked at the particle in one reference frame where the particle was moving, then changed their velocity so that the particle's velocity appeared to be zero. The motion would change from circular to fixed. Is this possible?

The solution to the puzzle above relies on understanding relativity. Imagine that the first observer believes  $\mathbf{B} \neq 0$  and that the electric field  $\mathbf{E} = 0$ . If the observer then changes reference frames by accelerating to a velocity  $\mathbf{v}$ , in the new frame  $\mathbf{B}$  and  $\mathbf{E}$  both change. If the observer moved to the frame where the charge, originally moving with a small velocity  $v$ , is now at rest, the new electric field is indeed  $\mathbf{v} \times \mathbf{B}$ , which then leads to the same acceleration as one had before. If the velocity is not small compared to the speed of light, additional  $\gamma$  factors come into play,  $\gamma = 1/\sqrt{1 - (v/c)^2}$ . Relativistic motion will not be considered in this course.



## Sliding Block tied to a Wall

Another classical case is that of simple harmonic oscillations, here represented by a block sliding on a horizontal frictionless surface. The block is tied to a wall with a spring. If the spring is not compressed or stretched too far, the force on the block at a given position  $x$  is

$$F = -kx.$$

## Back and Forth, Sliding Block with no friction

The negative sign means that the force acts to restore the object to an equilibrium position. Newton's equation of motion for this idealized system is then

$$m \frac{d^2x}{dt^2} = -kx,$$

or we could rephrase it as

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x = -\omega_0^2x,$$

with the angular frequency  $\omega_0^2 = k/m$ .

We will derive the above force when we start studying **harmonic oscillations**.

## Final rewrite

With the position  $x(t)$  and the velocity  $v(t) = dx/dt$  we can reformulate Newton's equation in the following way

$$\frac{dx(t)}{dt} = v(t),$$

and

$$\frac{dv(t)}{dt} = -\omega_0^2x(t).$$

With initial conditions  $x(t_0) = x_0$  and  $v(t_0) = v_0$  we can in turn solve the differential equations.

## Analytical Solution

The above differential equation has the advantage that it can be solved analytically with general solutions on the form

$$x(t) = A \cos \omega_0 t + B \sin \omega_0 t,$$

and

$$v(t) = -\omega_0 A \sin \omega_0 t + \omega_0 B \cos \omega_0 t,$$

where  $A$  and  $B$  are constants to be determined from the initial conditions.

This provides in turn an important test for the numerical solution and the development of a program for more complicated cases which cannot be solved analytically.

We will discuss the above equations in more detail when we discuss harmonic oscillations.

## Summarizing the various motion problems 1

The examples we have discussed above were included in order to illustrate various methods (which depend on the specific problem) to find the solutions of the equations of motion. We have solved the equations of motion in the following ways:

**Solve the differential equations analytically.** We did this for example with the following object in one or two dimensions or the sliding block. Here we had for example an equation set like

$$\frac{dv_x}{dt} = -\gamma v_x,$$

and

$$\frac{dv_y}{dt} = -\gamma v_y - g,$$

and  $\gamma$  has dimension of inverse time.

## Summarizing the various motion problems 2

**Integrate the equations.** We could also integrate directly in case we can separate the degrees of freedom in a an easy way. Take for example one of the equations in the previous slide

$$\frac{dv_x}{dt} = -\gamma v_x,$$

which we can rewrite in terms of a left-hand side which depends only on the velocity and a right-hand side which depends only on time

$$\frac{dv_x}{v_x} = -\gamma dt.$$

Integrating we have (since we can separate  $v_x$  and  $t$ )

$$\int_{v_0}^{v_t} \frac{dv_x}{v_x} = - \int_{t_0}^{t_f} \gamma dt,$$

where  $v_f$  is the velocity at a final time and  $t_f$  is the final time. In this case we found, after having integrated the above two sides that

$$v_f(t) = v_0 \exp -\gamma t.$$

## Summarizing the various motion problems 3

**Solve the differential equations numerically.** Finally, using for example Euler's method, we can solve the differential equations numerically. If we can compare our numerical solutions with analytical solutions, we have an extra check of our numerical approaches.

The example code on the next slide is relevant for homework 3. Here we deal with a falling object in two dimensions. Except for the derivations above with an air resistance which is linear in the velocity, homework 3 uses a quadratic velocity dependence.

## Code example using Euler's methods

**Note:** this code needs some additional expressions and will not run

```
# Common imports
import numpy as np
import pandas as pd
from math import *
import matplotlib.pyplot as plt
import os
from pylab import plt, mpl
plt.style.use('seaborn')
mpl.rcParams['font.family'] = 'serif'

#define the gravitational acceleration
g = 9.80655 #m/s^2
# The mass and the drag constant D
D = 0.00245 #mass/length kg/m
m = 0.2 #kg, mass of falling object
DeltaT = 0.001
#set up final time, here just a number we have chosen
tfinal = 1.0
# set up number of points for all variables
n = ceil(tfinal/DeltaT)
# set up arrays for t, a, v, and y and arrays for analytical results
# Note the brute force setting up of arrays for x and y, vx, vy, ax and ay
# For hw3 you should think of using the 2-dim vectors you used in homework 2
t = np.zeros(n)
vy = np.zeros(n)
y = np.zeros(n)
vx = np.zeros(n)
x = np.zeros(n)
# Initial conditions
vx[0] = 10.0 #m/s
vy[0] = 0.0 #m/s
y[0] = 10.0 #m
x[0] = 0.0 #m
# Start integrating using Euler's method
for i in range(n-1):
    # expression for acceleration, note the absolute value and division by mass
    # ax = You need to set up the expression for force and thereby the acceleration in the x-direction
    # ay = You need to set up the expression for force and thereby the acceleration in the y-direction
    # update velocity and position
    vx[i+1] = vx[i] + DeltaT*ax
    x[i+1] = x[i] + DeltaT*vx[i]
    vy[i+1] = vy[i] + DeltaT*ay
```

```

        y[i+1] = y[i] + DeltaT*vy[i]
        # update time to next time step and compute analytical answer
        t[i+1] = t[i] + DeltaT
    # Here you need to set up the analytical solution for y(t) and x(t)

    if ( y[i+1] < 0.0):
        break
    data = {'t[s]': t,
            'Relative error in y': abs((y-yanalytic)/yanalytic),
            'vy[m/s]': vy,
            'Relative error in x': abs((x-xanalytic)/xanalytic),
            'vx[m/s]': vx
    }
    NewData = pd.DataFrame(data)
    display(NewData)
    # save to file
    NewData.to_csv(outfile, index=False)
    #then plot
    fig, axs = plt.subplots(4, 1)
    axs[0].plot(t, y)
    axs[0].set_xlim(0, tfinal)
    axs[0].set_ylabel('y')
    axs[1].plot(t, vy)
    axs[1].set_ylabel('vy[m/s]')
    axs[1].set_xlabel('time[s]')
    axs[2].plot(t, x)
    axs[2].set_xlim(0, tfinal)
    axs[2].set_ylabel('x')
    axs[3].plot(t, vx)
    axs[3].set_ylabel('vx[m/s]')
    axs[3].set_xlabel('time[s]')
    fig.tight_layout()
    plt.show()

```

## Work, Energy, Momentum and Conservation laws

The previous three cases have shown us how to use Newton's laws of motion to determine the motion of an object based on the forces acting on it. For two of the cases there is an underlying assumption that we can find an analytical solution to a continuous problem. With a continuous problem we mean a problem where the various variables can take any value within a finite or infinite interval.

Unfortunately, in many cases we cannot find an exact solution to the equations of motion we get from Newton's second law. The numerical approach, where we discretize the continuous problem, allows us however to study a much richer set of problems. For problems involving Newton's laws and the various equations of motion we encounter, solving the equations numerically, is the standard approach.

It allows us to focus on the underlying forces. Often we end up using the same numerical algorithm for different problems.

Here we introduce a commonly used technique that allows us to find the velocity as a function of position without finding the position as a function of time—an alternate form of Newton's second law. The method is based on a

simple principle: Instead of solving the equations of motion directly, we integrate the equations of motion. Such a method is called an integration method.

This allows us also to introduce the **work-energy** theorem. This theorem lets us find the velocity as a function of position for an object even in cases when we cannot solve the equations of motion. This introduces us to the concept of work and kinetic energy, which is the energy related to the motion of an object.

And finally, later, we will link the work-energy theorem with the principle of conservation of energy.

## The Work-Energy Theorem

Let us define the kinetic energy  $K$  with a given velocity  $\mathbf{v}$

$$K = \frac{1}{2}mv^2,$$

where  $m$  is the mass of the object we are considering. We assume also that there is a force  $\mathbf{F}$  acting on the given object

$$\mathbf{F} = \mathbf{F}(\mathbf{r}, \mathbf{v}, t),$$

with  $\mathbf{r}$  the position and  $t$  the time. In general we assume the force is a function of all these variables. Many of the more central forces in Nature however, depend only on the position. Examples are the gravitational force and the force derived from the Coulomb potential in electromagnetism.

## Rewriting the Kinetic Energy

Let us study the derivative of the kinetic energy with respect to time  $t$ . Its continuous form is

$$\frac{dK}{dt} = \frac{1}{2}m \frac{d\mathbf{v} \cdot \mathbf{v}}{dt}.$$

Using our results from exercise 3 of homework 1, we can write the derivative of a vector dot product as

$$\frac{dK}{dt} = \frac{1}{2}m \frac{d\mathbf{v} \cdot \mathbf{v}}{dt} = \frac{1}{2}m \left( \frac{d\mathbf{v}}{dt} \cdot \mathbf{v} + \mathbf{v} \cdot \frac{d\mathbf{v}}{dt} \right) = m \frac{d\mathbf{v}}{dt} \cdot \mathbf{v}.$$

We know also that the acceleration is defined as

$$\mathbf{a} = \frac{\mathbf{F}}{m} = \frac{d\mathbf{v}}{dt}.$$

We can then rewrite the equation for the derivative of the kinetic energy as

$$\frac{dK}{dt} = m \frac{d\mathbf{v}}{dt} \cdot \mathbf{v} = \mathbf{F} \cdot \frac{d\mathbf{r}}{dt},$$

where we defined the velocity as the derivative of the position with respect to time.

## Discretizing

Let us now discretize the above equation by letting the instantaneous terms be replaced by a discrete quantity, that is we let  $dK \rightarrow \Delta K$ ,  $dt \rightarrow \Delta t$ ,  $d\mathbf{r} \rightarrow \Delta\mathbf{r}$  and  $d\mathbf{v} \rightarrow \Delta\mathbf{v}$ .

We have then

$$\frac{\Delta K}{\Delta t} = m \frac{\Delta \mathbf{v}}{\Delta t} \mathbf{v} = \mathbf{F} \frac{\Delta \mathbf{r}}{\Delta t},$$

or by multiplying out  $\Delta t$  we have

$$\Delta K = \mathbf{F} \Delta \mathbf{r}.$$

We define this quantity as the **work** done by the force  $\mathbf{F}$  during the displacement  $\Delta\mathbf{r}$ . If we study the dimensionality of this problem we have mass times length squared divided by time squared, or just dimension energy.

## Difference in kinetic energy

If we now define a series of such displacements  $\Delta\mathbf{r}$  we have a difference in kinetic energy at a final position  $\mathbf{r}_n$  and an initial position  $\mathbf{r}_0$  given by

$$\Delta K = \frac{1}{2}mv_n^2 - \frac{1}{2}mv_0^2 = \sum_{i=0}^n \mathbf{F}_i \Delta \mathbf{r},$$

where  $\mathbf{F}_i$  are the forces acting at every position  $\mathbf{r}_i$ .

The work done by acting with a force on a set of displacements can then be as expressed as the difference between the initial and final kinetic energies.

This defines the **work-energy** theorem.

## From the discrete version to the continuous version

If we take the limit  $\Delta\mathbf{r} \rightarrow 0$ , we can rewrite the sum over the various displacements in terms of an integral, that is

$$\Delta K = \frac{1}{2}mv_n^2 - \frac{1}{2}mv_0^2 = \sum_{i=0}^n \mathbf{F}_i \Delta \mathbf{r} \rightarrow \int_{\mathbf{r}_0}^{\mathbf{r}_n} \mathbf{F}(\mathbf{r}, \mathbf{v}, t) d\mathbf{r}.$$

This integral defines a path integral since it will depend on the given path we take between the two end points. We will replace the limits with the symbol  $c$  in order to indicate that we take a specific contour in space when the force acts on the system. That is the work  $W_{n0}$  between two points  $\mathbf{r}_n$  and  $\mathbf{r}_0$  is labeled as

$$W_{n0} = \frac{1}{2}mv_n^2 - \frac{1}{2}mv_0^2 = \int_c \mathbf{F}(\mathbf{r}, \mathbf{v}, t) d\mathbf{r}.$$

Note that if the force is perpendicular to the displacement, then the force does not affect the kinetic energy.

Let us now study some examples of forces and how to find the velocity from the integration over a given path.

Thereafter we study how to evaluate an integral numerically.

## Studying the Work-energy Theorem numerically

In order to study the work- energy, we will normally need to perform a numerical integration, unless we can integrate analytically. Here we present some of the simpler methods such as the **rectangle** rule, the **trapezoidal** rule and higher-order methods like the Simpson family of methods.

### Example of an Electron moving along a Surface

As an example, let us consider the following case. We have classical electron which moves in the  $x$ -direction along a surface. The force from the surface is

$$\mathbf{F}(x) = -F_0 \sin\left(\frac{2\pi x}{b}\right)\mathbf{e}_1.$$

The constant  $b$  represents the distance between atoms at the surface of the material,  $F_0$  is a constant and  $x$  is the position of the electron.

Using the work-energy theorem we can find the work  $W$  done when moving an electron from a position  $x_0$  to a final position  $x$  through the integral

$$W = \int_{x_0}^x \mathbf{F}(x')dx' = - \int_{x_0}^x F_0 \sin\left(\frac{2\pi x'}{b}\right)dx',$$

which results in

$$W = \frac{F_0 b}{2\pi} \left[ \cos\left(\frac{2\pi x}{b}\right) - \cos\left(\frac{2\pi x_0}{b}\right) \right].$$

### Finding the Velocity

If we now use the work-energy theorem we can find the the velocity at a final position  $x$  by setting up the differences in kinetic energies between the final position and the initial position  $x_0$ .

We have that the work done by the force is given by the difference in kinetic energies as

$$W = \frac{1}{2}m(v^2(x) - v^2(x_0)) = \frac{F_0 b}{2\pi} \left[ \cos\left(\frac{2\pi x}{b}\right) - \cos\left(\frac{2\pi x_0}{b}\right) \right],$$

and labeling  $v(x_0) = v_0$  (and assuming we know the initial velocity) we have

$$v(x) = \pm \sqrt{v_0^2 + \frac{F_0 b}{m\pi} \left[ \cos\left(\frac{2\pi x}{b}\right) - \cos\left(\frac{2\pi x_0}{b}\right) \right]},$$

Choosing  $x_0 = 0$ m and  $v_0 = 0$ m/s we can simplify the above equation to

$$v(x) = \pm \sqrt{\frac{F_0 b}{m\pi} \left[ \cos\left(\frac{2\pi x}{b}\right) - 1 \right]},$$

## Harmonic Oscillations

Another well-known force (and we will derive when we come to Harmonic Oscillations) is the case of a sliding block attached to a wall through a spring. The block is attached to a spring with spring constant  $k$ . The other end of the spring is attached to the wall at the origin  $x = 0$ . We assume the spring has an equilibrium length  $L_0$ .

The force  $F_x$  from the spring on the block is then

$$F_x = -k(x - L_0).$$

The position  $x$  where the spring force is zero is called the equilibrium position. In our case this is  $x = L_0$ .

We can now compute the work done by this force when we move our block from an initial position  $x_0$  to a final position  $x$

$$W = \int_{x_0}^x F_x dx' = -k \int_{x_0}^x (x' - L_0) dx' = \frac{1}{2}k(x_0 - L_0)^2 - \frac{1}{2}k(x - L_0)^2.$$

If we now bring back the definition of the work-energy theorem in terms of the kinetic energy we have

$$W = \frac{1}{2}mv^2(x) - \frac{1}{2}mv_0^2 = \frac{1}{2}k(x_0 - L_0)^2 - \frac{1}{2}k(x - L_0)^2,$$

which we rewrite as

$$\frac{1}{2}mv^2(x) + \frac{1}{2}k(x - L_0)^2 = \frac{1}{2}mv_0^2 + \frac{1}{2}k(x_0 - L_0)^2.$$

What does this mean? The total energy, which is the sum of potential and kinetic energy, is conserved. Wow, this sounds interesting. We will analyze this next week in more detail when we study energy, momentum and angular momentum conservation.

## Numerical Integration

This material is optional. We will not cover this in the lectures and we will not use it in exercises or exams. It is included here for the same of completeness.

Let us now see how we could have solved the above integral numerically.

There are several numerical algorithms for finding an integral numerically. The more familiar ones like the rectangular rule or the trapezoidal rule have simple geometric interpretations.

Let us look at the mathematical details of what are called equal-step methods, also known as Newton-Cotes quadrature.

## Newton-Cotes Quadrature or equal-step methods

The integral

$$I = \int_a^b f(x) dx \tag{31}$$



has a very simple meaning. The integral is the area encribed by the function  $f(x)$  starting from  $x = a$  to  $x = b$ . It is subdivided in several smaller areas whose evaluation is to be approximated by different techniques. The areas under the curve can for example be approximated by rectangular boxes or trapezoids.

## Basic philosophy of equal-step methods

In considering equal step methods, our basic approach is that of approximating a function  $f(x)$  with a polynomial of at most degree  $N - 1$ , given  $N$  integration points. If our polynomial is of degree 1, the function will be approximated with  $f(x) \approx a_0 + a_1x$ .

## Simple algorithm for equal step methods

The algorithm for these integration methods is rather simple, and the number of approximations perhaps unlimited!

- Choose a step size  $h = (b - a)/N$  where  $N$  is the number of steps and  $a$  and  $b$  the lower and upper limits of integration.
- With a given step length we rewrite the integral as

$$\int_a^b f(x)dx = \int_a^{a+h} f(x)dx + \int_{a+h}^{a+2h} f(x)dx + \cdots + \int_{b-h}^b f(x)dx.$$

- The strategy then is to find a reliable polynomial approximation for  $f(x)$  in the various intervals. Choosing a given approximation for  $f(x)$ , we obtain a specific approximation to the integral.
- With this approximation to  $f(x)$  we perform the integration by computing the integrals over all subintervals.

## Simple algorithm for equal step methods

One possible strategy then is to find a reliable polynomial expansion for  $f(x)$  in the smaller subintervals. Consider for example evaluating

$$\int_a^{a+2h} f(x)dx,$$

which we rewrite as

$$\int_a^{a+2h} f(x)dx = \int_{x_0-h}^{x_0+h} f(x)dx. \quad (32)$$

We have chosen a midpoint  $x_0$  and have defined  $x_0 = a + h$ .

## The rectangle method

A very simple approach is the so-called midpoint or rectangle method. In this case the integration area is split in a given number of rectangles with length  $h$  and height given by the mid-point value of the function. This gives the following simple rule for approximating an integral

$$I = \int_a^b f(x)dx \approx h \sum_{i=1}^N f(x_{i-1/2}), \quad (33)$$

where  $f(x_{i-1/2})$  is the midpoint value of  $f$  for a given rectangle. We will discuss its truncation error below. It is easy to implement this algorithm, as shown below

## Truncation error for the rectangular rule

The correct mathematical expression for the local error for the rectangular rule  $R_i(h)$  for element  $i$  is

$$\int_{-h}^h f(x)dx - R_i(h) = -\frac{h^3}{24} f^{(2)}(\xi),$$

and the global error reads

$$\int_a^b f(x)dx - R_h(f) = -\frac{b-a}{24} h^2 f^{(2)}(\xi),$$

where  $R_h$  is the result obtained with rectangular rule and  $\xi \in [a, b]$ .

## Codes for the Rectangular rule

We go back to our simple example above and set  $F_0 = b = 1$  and choose  $x_0 = 0$  and  $x = 1/2$ , and have

$$W = \frac{1}{\pi}.$$

The code here computes the integral using the rectangle rule and  $n = 100$  integration points we have a relative error of  $10^{-5}$ .

```
from math import sin, pi
import numpy as np
from sympy import Symbol, integrate
# function for the Rectangular rule
def Rectangular(a,b,f,n):
    h = (b-a)/float(n)
    s = 0
    for i in range(0,n,1):
        x = (i+0.5)*h
        s = s+ f(x)
    return h*s
# function to integrate
def function(x):
```

```

    return sin(2*pi*x)
# define integration limits and integration points
a = 0.0; b = 0.5;
n = 100
Exact = 1./pi
print("Relative error= ", abs( (Rectangular(a,b,function,n)-Exact)/Exact))

```

## The trapezoidal rule

The other integral gives

$$\int_{x_0-h}^{x_0} f(x)dx = \frac{h}{2} (f(x_0) + f(x_0 - h)) + O(h^3),$$

and adding up we obtain

$$\int_{x_0-h}^{x_0+h} f(x)dx = \frac{h}{2} (f(x_0 + h) + 2f(x_0) + f(x_0 - h)) + O(h^3), \quad (34)$$

which is the well-known trapezoidal rule. Concerning the error in the approximation made,  $O(h^3) = O((b-a)^3/N^3)$ , you should note that this is the local error. Since we are splitting the integral from  $a$  to  $b$  in  $N$  pieces, we will have to perform approximately  $N$  such operations.

This means that the *global error* goes like  $\approx O(h^2)$ . The trapezoidal reads then

$$I = \int_a^b f(x)dx = h (f(a)/2 + f(a+h) + f(a+2h) + \cdots + f(b-h) + f_b/2), \quad (35)$$

with a global error which goes like  $O(h^2)$ .

Hereafter we use the shorthand notations  $f_{-h} = f(x_0 - h)$ ,  $f_0 = f(x_0)$  and  $f_h = f(x_0 + h)$ .

## Error in the trapezoidal rule

The correct mathematical expression for the local error for the trapezoidal rule is

$$\int_a^b f(x)dx - \frac{b-a}{2} [f(a) + f(b)] = -\frac{h^3}{12} f^{(2)}(\xi),$$

and the global error reads

$$\int_a^b f(x)dx - T_h(f) = -\frac{b-a}{12} h^2 f^{(2)}(\xi),$$

where  $T_h$  is the trapezoidal result and  $\xi \in [a, b]$ .

## Algorithm for the trapezoidal rule

The trapezoidal rule is easy to implement numerically through the following simple algorithm

- Choose the number of mesh points and fix the step length.
- calculate  $f(a)$  and  $f(b)$  and multiply with  $h/2$ .
- Perform a loop over  $n = 1$  to  $n - 1$  ( $f(a)$  and  $f(b)$  are known) and sum up the terms  $f(a + h) + f(a + 2h) + f(a + 3h) + \dots + f(b - h)$ . Each step in the loop corresponds to a given value  $a + nh$ .
- Multiply the final result by  $h$  and add  $hf(a)/2$  and  $hf(b)/2$ .

## Trapezoidal Rule

We use the same function and integrate now using the trapezoidal rule.

```
import numpy as np
from sympy import Symbol, integrate
# function for the trapezoidal rule
def Trapez(a,b,f,n):
    h = (b-a)/float(n)
    s = 0
    x = a
    for i in range(1,n,1):
        x = x+h
        s = s+ f(x)
    s = 0.5*(f(a)+f(b)) +s
    return h*s
# function to integrate
def function(x):
    return sin(2*pi*x)
# define integration limits and integration points
a = 0.0; b = 0.5;
n = 100
Exact = 1./pi
print("Relative error= ", abs( (Trapez(a,b,function,n)-Exact)/Exact))
```

## Simpsons' rule

Instead of using the above first-order polynomials approximations for  $f$ , we attempt at using a second-order polynomials. In this case we need three points in order to define a second-order polynomial approximation

$$f(x) \approx P_2(x) = a_0 + a_1x + a_2x^2.$$

Using again Lagrange's interpolation formula we have

$$P_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1 + \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0.$$

Inserting this formula in the integral of Eq. (32) we obtain

$$\int_{-h}^{+h} f(x)dx = \frac{h}{3} (f_h + 4f_0 + f_{-h}) + O(h^5),$$

which is Simpson's rule.

### Simpson's rule

Note that the improved accuracy in the evaluation of the derivatives gives a better error approximation,  $O(h^5)$  vs.  $O(h^3)$ . But this is again the *local error approximation*. Using Simpson's rule we can easily compute the integral of Eq. (31) to be

$$I = \int_a^b f(x)dx = \frac{h}{3} (f(a) + 4f(a+h) + 2f(a+2h) + \dots + 4f(b-h) + f_b), \quad (36)$$

with a global error which goes like  $O(h^4)$ .

### Mathematical expressions for the truncation error

More formal expressions for the local and global errors are for the local error

$$\int_a^b f(x)dx - \frac{b-a}{6} [f(a) + 4f((a+b)/2) + f(b)] = -\frac{h^5}{90} f^{(4)}(\xi),$$

and for the global error

$$\int_a^b f(x)dx - S_h(f) = -\frac{b-a}{180} h^4 f^{(4)}(\xi).$$

with  $\xi \in [a, b]$  and  $S_h$  the results obtained with Simpson's method.

### Algorithm for Simpson's rule

The method can easily be implemented numerically through the following simple algorithm

- Choose the number of mesh points and fix the step.
- calculate  $f(a)$  and  $f(b)$
- Perform a loop over  $n = 1$  to  $n - 1$  ( $f(a)$  and  $f(b)$  are known) and sum up the terms  $4f(a+h) + 2f(a+2h) + 4f(a+3h) + \dots + 4f(b-h)$ . Each step in the loop corresponds to a given value  $a + nh$ . Odd values of  $n$  give 4 as factor while even values yield 2 as factor.
- Multiply the final result by  $\frac{h}{3}$ .

## Code example

```
from math import sin, pi
import numpy as np
from sympy import Symbol, integrate
# function for the trapezoidal rule
def Simpson(a,b,f,n):
    h = (b-a)/float(n)
    sum = f(a)/float(2);
    for i in range(1,n):
        sum = sum + f(a+i*h)*(3+(-1)**(i+1))
    sum = sum + f(b)/float(2)
    return sum*h/3.0
# function to integrate
def function(x):
    return sin(2*pi*x)
# define integration limits and integration points
a = 0.0; b = 0.5;
n = 100
Exact = 1./pi
print("Relative error= ", abs( (Simpson(a,b,function,n)-Exact)/Exact))
```

We see that Simpson's rule gives a much better estimation of the relative error with the same amount of points as we had for the Rectangle rule and the Trapezoidal rule.

## Symbolic integration

We could also use the symbolic mathematics. Here Python comes to our rescue with **SymPy**, which is a Python library for symbolic mathematics.

Here's an example on how you could use **SymPy** where we compare the symbolic calculation with an integration of a function  $f(x)$  by the Trapezoidal rule. Here we show an example code that evaluates the integral  $\int_0^1 dx x^2 = 1/3$ . The following code for the trapezoidal rule allows you to plot the relative error by comparing with the exact result. By increasing to  $10^8$  points one arrives at a region where numerical errors start to accumulate.

```
from math import log10
import numpy as np
from sympy import Symbol, integrate
import matplotlib.pyplot as plt
# function for the trapezoidal rule
def Trapez(a,b,f,n):
    h = (b-a)/float(n)
    s = 0
    x = a
    for i in range(1,n,1):
        x = x+h
        s = s+ f(x)
    s = 0.5*(f(a)+f(b)) +s
    return h*s
# function to compute pi
def function(x):
    return x*x
# define integration limits
```

```

a = 0.0; b = 1.0;
# find result from sympy
# define x as a symbol to be used by sympy
x = Symbol('x')
exact = integrate(function(x), (x, a, b))
# set up the arrays for plotting the relative error
n = np.zeros(9); y = np.zeros(9);
# find the relative error as function of integration points
for i in range(1, 8, 1):
    npts = 10**i
    result = Trapez(a,b,function,npts)
    RelativeError = abs((exact-result)/exact)
    n[i] = log10(npts); y[i] = log10(RelativeError);
plt.plot(n,y, 'ro')
plt.xlabel('n')
plt.ylabel('Relative error')
plt.show()

```