Classical Mechanics

Morten Hjorth-Jensen

Dec 13, 2020

CONTENTS

Morten Hjorth-Jensen, Department of Physics and Astronomy and National Superconducting Cyclotron Laboratory, Michigan State University, USA and Department of Physics, University of Oslo, Norway

Date: **Jan 22, 2020**

CHAPTER ONE

INTRODUCTION

Classical mechanics is a topic which has been taught intensively over several centuries. It is, with its many variants and ways of presenting the educational material, normally the first **real** physics course many of us meet and it lays the foundation for further physics studies. Many of the equations and ways of reasoning about the underlying laws of motion and pertinent forces, shape our approaches and understanding of the scientific method and discourse, as well as the way we develop our insights and deeper understanding about physical systems.

There is a wealth of well-tested (from both a physics point of view and a pedagogical standpoint) exercises and problems which can be solved analytically. However, many of these problems represent idealized and less realistic situations. The large majority of these problems are solved by paper and pencil and are traditionally aimed at what we normally refer to as continuous models from which we may find an analytical solution. As a consequence, when teaching mechanics, it implies that we can seldomly venture beyond an idealized case in order to develop our understandings and insights about the underlying forces and laws of motion.

CHAPTER

TWO

NUMERICAL ELEMENTS

On the other hand, numerical algorithms call for approximate discrete models and much of the development of methods for continuous models are nowadays being replaced by methods for discrete models in science and industry, simply because **much larger classes of problems can be addressed** with discrete models, often by simpler and more generic methodologies.

As we will see below, when properly scaling the equations at hand, discrete models open up for more advanced abstractions and the possibility to study real life systems, with the added bonus that we can explore and deepen our basic understanding of various physical systems

Analytical solutions are as important as before. In addition, such solutions provide us with invaluable benchmarks and tests for our discrete models. Such benchmarks, as we will see below, allow us to discuss possible sources of errors and their behaviors. And finally, since most of our models are based on various algorithms from numerical mathematics, we have a unique oppotunity to gain a deeper understanding of the mathematical approaches we are using.

With computing and data science as important elements in essentially all aspects of a modern society, we could then try to define Computing as **solving scientific problems using all possible tools, including symbolic computing, computers and numerical algorithms, and analytical paper and pencil solutions**. Computing provides us with the tools to develope our own understanding of the scientific method by enhancing algorithmic thinking.

CHAPTER

THREE

COMPUTATIONS AND THE SCIENTIFIC METHOD

The way we will teach this course reflects this definition of computing. The course contains both classical paper and pencil exercises as well as computational projects and exercises. The hope is that this will allow you to explore the physics of systems governed by the degrees of freedom of classical mechanics at a deeper level, and that these insights about the scientific method will help you to develop a better understanding of how the underlying forces and equations of motion and how they impact a given system. Furthermore, by introducing various numerical methods via computational projects and exercises, we aim at developing your competences and skills about these topics.

These competences will enable you to

- understand how algorithms are used to solve mathematical problems,
- derive, verify, and implement algorithms,
- understand what can go wrong with algorithms,
- use these algorithms to construct reproducible scientific outcomes and to engage in science in ethical ways, and
- think algorithmically for the purposes of gaining deeper insights about scientific problems.

All these elements are central for maturing and gaining a better understanding of the modern scientific process *per se*.

The power of the scientific method lies in identifying a given problem as a special case of an abstract class of problems, identifying general solution methods for this class of problems, and applying a general method to the specific problem (applying means, in the case of computing, calculations by pen and paper, symbolic computing, or numerical computing by ready-made and/or self-written software). This generic view on problems and methods is particularly important for understanding how to apply available, generic software to solve a particular problem.

However, verification of algorithms and understanding their limitations requires much of the classical knowledge about continuous models.

CHAPTER FOUR

A WELL-KNOWN EXAMPLES TO ILLUSTRATE MANY OF THE ABOVE CONCEPTS

Before we venture into a reminder on Python and mechanics relevant applications, let us briefly outline some of the abovementioned topics using an example many of you may have seen before in for example CMSE201. A simple algorithm for integration is the Trapezoidal rule. Integration of a function $f(x)$ by the Trapezoidal Rule is given by following algorithm for an interval $x \in [a, b]$

$$\int_a^b (f(x)dx = \frac{1}{2}\left[f(a) + 2f(a+h) + \cdots + 2f(b-h) + f(b)\right] + O(h^2),$$

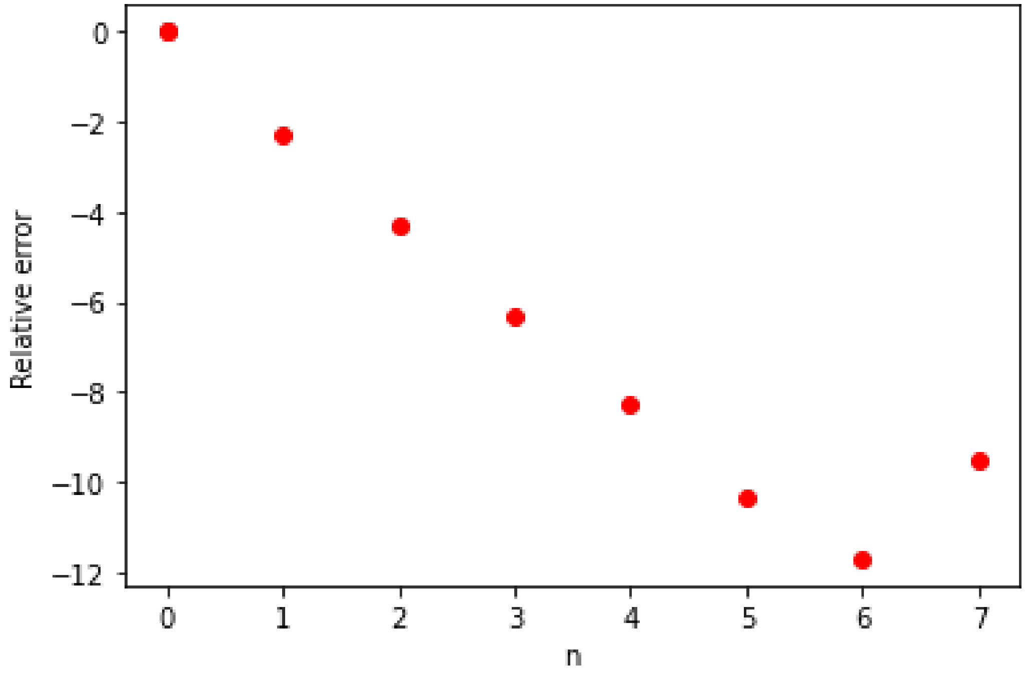
where h is the so-called stepsize defined by the number of integration points N as $h = (b-a)/(n)$. Python offers an extremely versatile programming environment, allowing for the inclusion of analytical studies in a numerical program. Here we show an example code with the **trapezoidal rule**. We use also **SymPy** to evaluate the exact value of the integral and compute the absolute error with respect to the numerically evaluated one of the integral $\int_0^1 dxx^2 = 1/3$. 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.

```
%matplotlib inline

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);
```

(continues on next page)

(continued from previous page)

```
# 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()
```



CHAPTER

FIVE

ANALYZING THE ABOVE EXAMPLE

This example shows the potential of combining numerical algorithms with symbolic calculations, allowing us to

- Validate and verify their algorithms.
- Including concepts like unit testing, one has the possibility to test and test several or all parts of the code.
- Validation and verification are then included *naturally* and one can develop a better attitude to what is meant with an ethically sound scientific approach.
- The above example allows the student to also test the mathematical error of the algorithm for the trapezoidal rule by changing the number of integration points. The students get **trained from day one to think error analysis**.
- With a Jupyter notebook you can keep exploring similar examples and turn them in as your own notebooks.

In this process we can easily bake in

1. How to structure a code in terms of functions
2. How to make a module
3. How to read input data flexibly from the command line
4. How to create graphical/web user interfaces
5. How to write unit tests (test functions or doctests)
6. How to refactor code in terms of classes (instead of functions only)
7. How to conduct and automate large-scale numerical experiments
8. How to write scientific reports in various formats (LaTeX, HTML)

The conventions and techniques outlined here will save you a lot of time when you incrementally extend software over time from simpler to more complicated problems. In particular, you will benefit from many good habits:

1. New code is added in a modular fashion to a library (modules)
2. Programs are run through convenient user interfaces
3. It takes one quick command to let all your code undergo heavy testing
4. Tedious manual work with running programs is automated,
5. Your scientific investigations are reproducible, scientific reports with top quality typesetting are produced both for paper and electronic devices.

CHAPTER

SIX

TEACHING TEAM, GRADING AND OTHER PRACTICALITIES

CHAPTER

SEVEN

GRADING AND DATES

CHAPTER

EIGHT

POSSIBLE TEXTBOOKS AND LECTURE NOTES

Recommended textbook:

- John R. Taylor, Classical Mechanics (Univ. Sci. Books 2005), see also the GitHub link of the course

Additional textbooks:

- Anders Malthe-Sørenssen, Elementary Mechanics using Python (Springer 2015) and the GitHub link of the course
- Alessandro Bettini, A Course in Classical Physics 1, Mechanics (Springer 2017) and the GitHub link of the course.

The books from Springer can be downloaded for free (pdf or ebook format) from any MSU IP address.

Lecture notes: Posted lecture notes are in the doc/pub folder here or at https://mhjensen.github.io/Physics321/doc/web/course.html for easier viewing. They are not meant to be a replacement for textbook. These notes are updated on a weekly basis and a **git pull** should thus always give you the latest update.

8.1 PHY321: Forces, Newton's Laws and Motion Example

Morten Hjorth-Jensen, Department of Physics and Astronomy and National Superconducting Cyclotron Laboratory, Michigan State University, USA and Department of Physics, University of Oslo, Norway

Date: **Feb 19, 2020**

8.1.1 Basic Steps of Scientific Investigations

An overarching aim in this course is to give you a deeper understanding of the scientific method. The problems we study will all involve cases where we can apply classical mechanics. In our previous material we already assumed that we had a model for the motion of an object. Alternatively we could have data from experiment (like Usain Bolt's 100m world record run in 2008). Or we could have performed ourselves an experiment and we want to understand which forces are at play and whether these forces can be understood in terms of fundamental forces.

Our first step consists in identifying the problem. What we sketch here may include a mix of experiment and theoretical simulations, or just experiment or only theory.

8.1.2 Identifying our System

Here we can ask questions like

1. What kind of object is moving
2. What kind of data do we have
3. How do we measure position, velocity, acceleration etc
4. Which initial conditions influence our system
5. Other aspects which allow us to identify the system

8.1.3 Defining a Model

With our eventual data and observations we would now like to develop a model for the system. In the end we want obviously to be able to understand which forces are at play and how they influence our specific system. That is, can we extract some deeper insights about a system?

We need then to

1. Find the forces that act on our system
2. Introduce models for the forces
3. Identify the equations which can govern the system (Newton's second law for example)
4. More elements we deem important for defining our model

8.1.4 Solving the Equations

With the model at hand, we can then solve the equations. In classical mechanics we normally end up with solving sets of coupled ordinary differential equations or partial differential equations.

1. Using Newton's second law we have equations of the type $\boldsymbol{F} = m\boldsymbol{a} = md\boldsymbol{v}/dt$
2. We need to define the initial conditions (typically the initial velocity and position as functions of time) and/or initial conditions and boundary conditions
3. The solution of the equations give us then the position, the velocity and other time-dependent quantities which may specify the motion of a given object.

We are not yet done. With our lovely solvers, we need to start thinking.

8.1.5 Analyze

Now it is time to ask the big questions. What do our results mean? Can we give a simple interpretation in terms of fundamental laws? What do our results mean? Are they correct? Thus, typical questions we may ask are

1. Are our results for say $\boldsymbol{r}(t)$ valid? Do we trust what we did? Can you validate and verify the correctness of your results?
2. Evaluate the answers and their implications
3. Compare with experimental data if possible. Does our model make sense?
4. and obviously many other questions.

The analysis stage feeds back to the first stage. It may happen that the data we had were not good enough, there could be large statistical uncertainties. We may need to collect more data or perhaps we did a sloppy job in identifying the degrees of freedom.

All these steps are essential elements in a scientific enquiry. Hopefully, through a mix of numerical simulations, analytical calculations and experiments we may gain a deeper insight about the physics of a specific system.

Let us now remind ourselves of Newton's laws, since these are the laws of motion we will study in this course.

8.1.6 Newton's Laws

When analyzing a physical system we normally start with distinguishing between the object we are studying (we will label this in more general terms as our **system**) and how this system interacts with the environment (which often means everything else!)

In our investigations we will thus analyze a specific physics problem in terms of the system and the environment. In doing so we need to identify the forces that act on the system and assume that the forces acting on the system must have a source, an identifiable cause in the environment.

A force acting on for example a falling object must be related to an interaction with something in the environment. This also means that we do not consider internal forces. The latter are forces between one part of the object and another part. In this course we will mainly focus on external forces.

Forces are either contact forces or long-range forces.

Contact forces, as evident from the name, are forces that occur at the contact between the system and the environment. Well-known long-range forces are the gravitional force and the electromagnetic force.

8.1.7 Setting up a model for forces acting on an object

In order to set up the forces which act on an object, the following steps may be useful

1. Divide the problem into system and environment.
2. Draw a figure of the object and everything in contact with the object.
3. Draw a closed curve around the system.
4. Find contact points—these are the points where contact forces may act.
5. Give names and symbols to all the contact forces.
6. Identify the long-range forces.
7. Make a drawing of the object. Draw the forces as arrows, vectors, starting from where the force is acting. The direction of the vector(s) indicates the (positive) direction of the force. Try to make the length of the arrow indicate the relative magnitude of the forces.
8. Draw in the axes of the coordinate system. It is often convenient to make one axis parallel to the direction of motion. When you choose the direction of the axis you also choose the positive direction for the axis.

8.1.8 Newton's Laws, the Second one first

Newton's second law of motion: The force $\boldsymbol{F}$ on an object of inertial mass m is related to the acceleration a of the object through

$$\boldsymbol{F} = m\boldsymbol{a},$$

where $\boldsymbol{a}$ is the acceleration.

Newton's laws of motion are laws of nature that have been found by experimental investigations and have been shown to hold up to continued experimental investigations. Newton's laws are valid over a wide range of length- and time-scales. We use Newton's laws of motion to describe everything from the motion of atoms to the motion of galaxies.

The second law is a vector equation with the acceleration having the same direction as the force. The acceleration is proportional to the force via the mass m of the system under study.

Newton's second law introduces a new property of an object, the so-called inertial mass m. We determine the inertial mass of an object by measuring the acceleration for a given applied force.

8.1.9 Then the First Law

What happens if the net external force on a body is zero? Applying Newton's second law, we find:

$$\boldsymbol{F} = 0 = m\boldsymbol{a},$$

which gives using the definition of the acceleration

$$\boldsymbol{a} = \frac{d\boldsymbol{v}}{dt} = 0.$$

The acceleration is zero, which means that the velocity of the object is constant. This is often referred to as Newton's first law. An object in a state of uniform motion tends to remain in that state unless an external force changes its state of motion. Why do we need a separate law for this? Is it not simply a special case of Newton's second law? Yes, Newton's first law can be deduced from the second law as we have illustrated. However, the first law is often used for a different purpose: Newton's First Law tells us about the limit of applicability of Newton's Second law. Newton's Second law can only be used in reference systems where the First law is obeyed. But is not the First law always valid? No! The First law is only valid in reference systems that are not accelerated. If you observe the motion of a ball from an accelerating car, the ball will appear to accelerate even if there are no forces acting on it. We call systems that are not accelerating inertial systems, and Newton's first law is often called the law of inertia. Newton's first and second laws of motion are only valid in inertial systems.

A system is an inertial system if it is not accelerated. It means that the reference system must not be accelerating linearly or rotating. Unfortunately, this means that most systems we know are not really inertial systems. For example, the surface of the Earth is clearly not an inertial system, because the Earth is rotating. The Earth is also not an inertial system, because it ismoving in a curved path around the Sun. However, even if the surface of the Earth is not strictly an inertial system, it may be considered to be approximately an inertial system for many laboratory-size experiments.

8.1.10 And finally the Third Law

If there is a force from object A on object B, there is also a force from object B on object A. This fundamental principle of interactions is called Newton's third law. We do not know of any force that do not obey this law: All forces appear in pairs. Newton's third law is usually formulated as: For every action there is an equal and opposite reaction.

8.1.11 Motion of a Single Object

Here we consider the motion of a single particle moving under the influence of some set of forces. We will consider some problems where the force does not depend on the position. In that case Newton's law $m\dot{\boldsymbol{v}} = \boldsymbol{F}(\boldsymbol{v})$ is a first-order differential equation and one solves for $\boldsymbol{v}(t)$, then moves on to integrate $\boldsymbol{v}$ to get the position. In essentially all of these cases we cna find an analytical solution.

8.1.12 Air Resistance in One Dimension

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

$dP = \rho_m dV v,$ (8.1)

where v is the velocity of the object and ρ_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

$dV = Avdt,$ (8.2)

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

8.1.13 Resulting Acceleration

Plugging this into the expression above,

$\frac{dP}{dt} = -\rho_m A v^2.$(8.3)

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.

$$F_{\text{drag}} = -c_W \rho_m A v^2, (8.4)$$

and the acceleration is

to

$$\frac{dv}{dt} = -\frac{c_W \rho_m A}{m} v^2 .(8.5)$$

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

to

$$W = \int dx\, f = \alpha v^n L .(8.6)$$

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.

8.1.14 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, $\boldsymbol{F} = -m\gamma\boldsymbol{v}$, for the purposes of this exercise. The acceleration for a projectile moving upwards, $\boldsymbol{a} = \boldsymbol{F}/m$, becomes

$$to$$

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

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

and γ 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 $t = 0$.

8.1.15 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 vs 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,

$$to$$

$$\frac{dv_x}{v_x} =$$

$$-\gamma dt,$$

$$\int_{v_{0x}}^{v_x} \frac{dv_x}{v_x} =$$

$$-\gamma \int_0^t dt,$$

$$\ln\left(\frac{v_x}{v_{0x}}\right) =$$

$$-\gamma t,$$

$$v_x(t) =$$

$$v_{0x} e^{-\gamma t}.$$

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

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

$$to$$

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

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

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

8.1.16 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 first order there is one arbitrary constant, A, and by inspection the solution is

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

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.

8.1.17 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, $d^2v_y/dt^2\cdots$, are called linear differential equations. If there are terms proportional to v^2, as would happen if the drag force 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,

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

8.1.18 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 {\it 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 {\it homogenous} equation, which is the equation with zero on the right-hand side,

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

Homogenous solutions will have arbitrary constants.

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

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

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

The sum of the two,

$$v_y=v_{y,p}+v_{y,h},$$

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 homogneous solution). If the equation were not linear, e.g if there were a term such as v_y^2 or $v_y\dot{v}_y$, this technique would not work.

8.1.19 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,

$$v_{y,h}=Be^{-\gamma t}.$$

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

$$v_{y,p}=-g/\gamma.$$

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

$$v_y = Be^{-\gamma t} - g/\gamma,$$

and one can find B from the initial velocity,

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

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

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

It is easy to see that this solution has $v_y = v_{0y}$ when $t = 0$ and $v_y = -g/\gamma$ when $t \to \infty$. One can also integrate the two equations to find the coordinates x and y as functions of t,

$$to$$

$$x =$$

$$\int_0^t dt'\, v_{0x}(t') = \frac{v_{0x}}{\gamma}\left(1 - e^{-\gamma t}\right),$$

$$y =$$

$$\int_0^t dt'\, v_{0y}(t') = -\frac{gt}{\gamma} + \frac{v_{0y} + g/\gamma}{\gamma}\left(1 - e^{-\gamma t}\right).$$

=

$$\int_0^t dt'\, v_{0x}(t') = \frac{v_{0x}}{\gamma}\left(1 - e^{-\gamma t}\right) ,=y$$
$$\int_0^t dt'\, v_{0y}(t') = -\frac{gt}{\gamma} + \frac{v_{0y} + g/\gamma}{\gamma}\left(1 - e^{-\gamma t}\right).$$

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

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

This cannot be inverted into a simple expression $t = \cdots$. 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, $t_{\rm guess}$.
2. Calculate F and its derivative, $F(t_{\rm guess})$ and $F'(t_{\rm guess})$.
3. Unless you guessed perfectly, $F \neq 0$, and assuming that $\Delta F \approx F'\Delta t$, one would choose
4. $\Delta t = -F(t_{\rm guess})/F'(t_{\rm guess})$.
5. Now repeat step 1, but with $t_{\rm guess} \rightarrow t_{\rm guess} + \Delta t$.

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 $t_{\rm guess}$. One breaks the loop once one finds F within some acceptable tolerance of zero. A program to do this will be added shortly.

8.1.20 Motion in a Magnetic Field

Another example of a velocity-dependent force is magnetism,

to

$\boldsymbol{F} =$

$q\boldsymbol{v} \times \boldsymbol{B},$

$F_i =$

$q\sum_{jk} \epsilon_{ijk} v_j B_k.$

=

$$F = q\boldsymbol{v} \times \boldsymbol{B}, F_i$$
$$q \sum_{jk} \epsilon_{ijk} v_j B_k.$$

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

$$to$$

$$F_x =$$
$$qBv_y$$
$$F_y =$$
$$-qBv_x.$$

$$=$$
$$=qBv_y F_y$$
$$-qBv_x.$$

The differential equations are

$$to$$

$$\dot{v}_x =$$
$$\omega_c v_y, \omega_c = qB/m$$
$$\dot{v}_y =$$
$$-\omega_c v_x.$$

$$=$$
$$\omega_c v_y, =\omega_c = qB/m\dot{v}_y$$
$$-\omega_c v_x.$$

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

to

$\ddot{v}_x = \omega_c \dot{v}_y = -\omega_c^2 v_x,$

$\ddot{v}_y =$

$-\omega_c \dot{v}_x = -\omega_c v_y.$

=

$-\omega_c \dot{v}_x = -\omega_c v_y.$

The solution to these equations can be seen by inspection,

to

$v_x =$

$A \sin(\omega_c t + \phi),$

$v_y =$

$A \cos(\omega_c t + \phi).$

=

$A \sin(\omega_c t + \phi), v_y$

$A \cos(\omega_c t + \phi).$

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

$$to$$

$$x-x_0 =$$

$$\int_{x_0}^{x} dx = \int_0^t dtv(t)$$

$$=$$

$$\frac{-A}{\omega_c}\cos(\omega_c t+\phi),$$

$$y-y_0 =$$

$$\frac{A}{\omega_c}\sin(\omega_c t+\phi).$$

$$=$$

$$\int_{x_0}^{x} dx = \int_0^t dtv(t)$$

$$\frac{-A}{\omega_c}\cos(\omega_c t+\phi), y-y_0$$

$$\frac{A}{\omega_c}\sin(\omega_c t+\phi).$$

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

$$\dot{v_z} = 0,$$

which leads to

$$z - z_0 = V_z t.$$

Added onto the circle, the motion is helical.

Note that the kinetic energy,

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

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 $\boldsymbol{F} \cdot dr = dt\boldsymbol{F} \cdot v = 0$.

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 $\boldsymbol{B} \neq 0$ and that the electric field $\boldsymbol{E} = 0$. If the observer then changes reference frames by accelerating to a velocity $\boldsymbol{v}$, in the new frame $\boldsymbol{B}$ and $\boldsymbol{E}$ 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 $\boldsymbol{v} \times \boldsymbol{B}$, which then leads to the same acceleration as one had before. If the velocity is not small compared to the speed of light, additional γ factors come into play, $\gamma = 1/\sqrt{1-(v/c)^2}$. Relativistic motion will not be considered in this course.

8.1.21 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

$$F = -kx.$$

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

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

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

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

$$x(t) = Acos(\omega_0 t + \nu),$$

where A is the amplitude and ν the phase constant. 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.

8.1.22 Simple Example, Block tied to a Wall

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

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

We are now going to solve these equations using first the standard forward Euler method. Later we will try to improve upon this.

8.1.23 Simple Example, Block tied to a Wall

Before proceeding however, it is important to note that in addition to the exact solution, we have at least two further tests which can be used to check our solution.

Since functions like cos are periodic with a period 2π, then the solution $x(t)$ has also to be periodic. This means that

$$x(t+T)=x(t),$$

with T the period defined as

$$T=\frac{2\pi}{\omega_0}=\frac{2\pi}{\sqrt{k/m}}.$$

Observe that T depends only on k/m and not on the amplitude of the solution.

8.1.24 Simple Example, Block tied to a Wall

In addition to the periodicity test, the total energy has also to be conserved.

Suppose we choose the initial conditions

$$x(t=0)=1\ \mathrm{m}\qquad v(t=0)=0\ \mathrm{m/s},$$

meaning that block is at rest at $t=0$ but with a potential energy

$$E_0=\frac{1}{2}kx(t=0)^2=\frac{1}{2}k.$$

The total energy at any time t has however to be conserved, meaning that our solution has to fulfil the condition

$$E_0=\frac{1}{2}kx(t)^2+\frac{1}{2}mv(t)^2.$$

We will derive this equation in our discussion on energy conservation.

8.1.25 Simple Example, Block tied to a Wall

An algorithm which implements these equations is included below.

- Choose the initial position and speed, with the most common choice $v(t=0)=0$ and some fixed value for the position.

- Choose the method you wish to employ in solving the problem.
- Subdivide the time interval $[t_i, t_f]$ into a grid with step size

$$h = \frac{t_f - t_i}{N},$$

where N is the number of mesh points.

- Calculate now the total energy given by

$$E_0 = \frac{1}{2}kx(t=0)^2 = \frac{1}{2}k.$$

- Choose ODE solver to obtain x_{i+1} and v_{i+1} starting from the previous values x_i and v_i.
- When we have computed $x(v)_{i+1}$ we upgrade $t_{i+1} = t_i + h$.
- This iterative process continues till we reach the maximum time t_f.
- The results are checked against the exact solution. Furthermore, one has to check the stability of the numerical solution against the chosen number of mesh points N.

8.1.26 Simple Example, Block tied to a Wall, python code

The following python program (code will be added shortly)

```
#
# This program solves Newtons equation for a block sliding on
# an horizontal frictionless surface.
# The block is tied to the wall with a spring, so N's eq takes the form:
#
#  m d^2x/dt^2 = - kx
#
# In order to make the solution dimless, we set k/m = 1.
# This results in two coupled diff. eq's that may be written as:
#
#  dx/dt = v
#  dv/dt = -x
#
# The user has to specify the initial velocity and position,
# and the number of steps. The time interval is fixed to
# t \in [0, 4\pi) (two periods)
#
```

8.1.27 The classical pendulum and scaling the equations

The angular equation of motion of the pendulum is given by Newton's equation and with no external force it reads

$$ml\frac{d^2\theta}{dt^2} + mgsin(\theta) = 0,$$

with an angular velocity and acceleration given by

$$v = l\frac{d\theta}{dt},$$

and

$$a = l\frac{d^2\theta}{dt^2}.$$

8.1.28 More on the Pendulum

We do however expect that the motion will gradually come to an end due a viscous drag torque acting on the pendulum. In the presence of the drag, the above equation becomes

$$ml\frac{d^2\theta}{dt^2} + \nu\frac{d\theta}{dt} + mgsin(\theta) = 0,$$

where ν is now a positive constant parameterizing the viscosity of the medium in question. In order to maintain the motion against viscosity, it is necessary to add some external driving force. We choose here a periodic driving force. The last equation becomes then

$$ml\frac{d^2\theta}{dt^2} + \nu\frac{d\theta}{dt} + mgsin(\theta) = Asin(\omega t),$$

with A and ω two constants representing the amplitude and the angular frequency respectively. The latter is called the driving frequency.

8.1.29 More on the Pendulum

We define

$$\omega_0 = \sqrt{g/l},$$

the so-called natural frequency and the new dimensionless quantities

$$\hat{t} = \omega_0 t,$$

with the dimensionless driving frequency

$$\hat{\omega} = \frac{\omega}{\omega_0},$$

and introducing the quantity Q, called the *quality factor*,

$$Q = \frac{mg}{\omega_0 \nu},$$

and the dimensionless amplitude

$$\hat{A} = \frac{A}{mg}$$

8.1.30 More on the Pendulum

We have

$$\frac{d^2\theta}{d\hat{t}^2} + \frac{1}{Q}\frac{d\theta}{d\hat{t}} + sin(\theta) = \hat{A}cos(\hat{\omega}\hat{t}).$$

This equation can in turn be recast in terms of two coupled first-order differential equations as follows

$$\frac{d\theta}{d\hat{t}} = \hat{v},$$

and

$$\frac{d\hat{v}}{d\hat{t}} = -\frac{\hat{v}}{Q} - sin(\theta) + \hat{A}cos(\hat{\omega}\hat{t}).$$

These are the equations to be solved. The factor Q represents the number of oscillations of the undriven system that must occur before its energy is significantly reduced due to the viscous drag. The amplitude $\hat{A}$ is measured in units of the maximum possible gravitational torque while $\hat{\omega}$ is the angular frequency of the external torque measured in units of the pendulum's natural frequency.

8.2 Energy, Momentum and Conservation Laws

Morten Hjorth-Jensen, Department of Physics and Astronomy and National Superconducting Cyclotron Laboratory, Michigan State University, USA and Department of Physics, University of Oslo, Norway

Date: **Feb 26, 2020**

8.2.1 Work, Energy, Momentum and Conservation laws

Energy conservation is most convenient as a strategy for addressing problems where time does not appear. For example, a particle goes from position x_0 with speed v_0, to position x_f; what is its new speed? However, it can also be applied to problems where time does appear, such as in solving for the trajectory $x(t)$, or equivalently $t(x)$.

8.2.2 Work and Energy

Material to be added here.

8.2.3 Energy Conservation

Energy is conserved in the case where the potential energy, $V(\boldsymbol{r})$, depends only on position, and not on time. The force is determined by V,

$$\boldsymbol{F}(\boldsymbol{r}) = -\nabla V(\boldsymbol{r}).$$

The net energy, $E = V + K$ where K is the kinetic energy, is then conserved,

$$to$$

$$\frac{d}{dt}(K+V) =$$

$$\frac{d}{dt}\left(\frac{m}{2}(v_x^2+v_y^2+v_z^2)+V(\boldsymbol{r})\right)$$

$$=$$

$$m\left(v_x\frac{dv_x}{dt}+v_y\frac{dv_y}{dt}+v_z\frac{dv_z}{dt}\right)+\partial_xV\frac{dx}{dt}+\partial_yV\frac{dy}{dt}+\partial_zV\frac{dz}{dt}$$

$$=$$

$$v_xF_x+v_yF_y+v_zF_z-F_xv_x-F_yv_y-F_zv_z=0.$$

$$=$$

$$\frac{d}{dt}\left(\frac{m}{2}(v_x^2+v_y^2+v_z^2)+V(\boldsymbol{r})\right)$$

$$m\left(v_x\frac{dv_x}{dt}+v_y\frac{dv_y}{dt}+v_z\frac{dv_z}{dt}\right)+\partial_xV\frac{dx}{dt}+\partial_yV\frac{dy}{dt}+\partial_zV\frac{dz}{dt}$$

$$v_xF_x+v_yF_y+v_zF_z-F_xv_x-F_yv_y-F_zv_z=0.$$

The same proof can be written more compactly with vector notation,

$$to$$

$$\frac{d}{dt}\left(\frac{m}{2}v^2+V(\boldsymbol{r})\right)=$$

$$m\boldsymbol{v}\cdot\dot{\boldsymbol{v}}+\nabla V(\boldsymbol{r})\cdot\dot{\boldsymbol{r}}$$

$$=$$

$$\boldsymbol{v}\cdot\boldsymbol{F}-\boldsymbol{F}\cdot\boldsymbol{v}=0.$$

=

$=m\boldsymbol{v}\cdot\dot{\boldsymbol{v}}+\nabla V(\boldsymbol{r})\cdot\dot{\boldsymbol{r}}$

$\boldsymbol{v}\cdot\boldsymbol{F}-\boldsymbol{F}\cdot\boldsymbol{v}=0.$

Inverting the expression for kinetic energy,

$$v=\sqrt{2K/m}=\sqrt{2(E-V)/m},$$

allows one to solve for the one-dimensional trajectory $x(t)$, by finding $t(x)$,

$$t=\int_{x_0}^{x}\frac{dx'}{v(x')}=\int_{x_0}^{x}\frac{dx'}{\sqrt{2(E-V(x'))/m}}.$$

Note this would be much more difficult in higher dimensions, because you would have to determine which points, x, y, z, the particles might reach in the trajectory, whereas in one dimension you can typically tell by simply seeing whether the kinetic energy is positive at every point between the old position and the new position.

Consider a simple harmonic oscillator potential, $V(x) = kx^2/2$, with a particle emitted from $x = 0$ with velocity v_0. Solve for the trajectory $t(x)$,

to

$t =$

$$\int_0^x \frac{dx'}{\sqrt{2(E - kx^2/2)/m}}$$

$=$

$$\sqrt{m/k}\int_0^x \frac{dx'}{\sqrt{x_{\max}^2 - x'^2}}, \quad x_{\max}^2 = 2E/k.$$

=

$$\int_0^x \frac{dx'}{\sqrt{2(E - kx^2/2)/m}}$$

=

$$\sqrt{m/k}\int_0^x \frac{dx'}{\sqrt{x_{\max}^2 - x'^2}}, \quad x_{\max}^2 = 2E/k.$$

Here $E = mv_0^2/2$ and $x_{\max}$ is defined as the maximum displacement before the particle turns around. This integral is done by the substitution $\sin\theta = x/x_{\max}$.

to

$(k/m)^{1/2}t =$

$\sin^{-1}(x/x_{\max}),$

$x =$

$$x_{\max}\sin\omega t, \quad \omega = \sqrt{k/m}.$$

=

=$\sin^{-1}(x/x_{\max}), x$

$$x_{\max}\sin\omega t, \quad \omega = \sqrt{k/m}.$$

8.2.4 Conservation of Momentum

Newton's third law which we met earlier states that **For every action there is an equal and opposite reaction**, is more accurately stated as **If two bodies exert forces on each other, these forces are equal in magnitude and opposite in direction**.

This means that for two bodies i and j, if the force on i due to j is called $\boldsymbol{F}_{ij}$, then

$$\boldsymbol{F}_{ij} = -\boldsymbol{F}_{ji}.$$

Newton's second law, $\boldsymbol{F} = m\boldsymbol{a}$, can be written for a particle i as

$$\boldsymbol{F}_i = \sum_{j \neq i} \boldsymbol{F}_{ij} = m_i \boldsymbol{a}_i,$$

where $\boldsymbol{F}_i$ (a single subscript) denotes the net force acting on i. Because the mass of i is fixed, one can see that

$$\boldsymbol{F}_i = \frac{d}{dt} m_i \boldsymbol{v}_i = \sum_{j \neq i} \boldsymbol{F}_{ij}.$$

Now, one can sum over all the particles and obtain

$$to$$

$$\frac{d}{dt} \sum_i m_i v_i =$$

$$\sum_{ij, i \neq j} \boldsymbol{F}_{ij}$$

$$=$$

$$0.$$

=

$$= \sum_{ij, i \neq j} \boldsymbol{F}_{ij}$$

0.

The last step made use of the fact that for every term ij, there is an equivalent term ji with opposite force. Because the momentum is defined as $m\boldsymbol{v}$, for a system of particles,

$$\frac{d}{dt}\sum_i m_i\boldsymbol{v}_i = 0, \quad \text{for isolated particles.}$$

By “isolated” one means that the only force acting on any particle i are those originating from other particles in the sum, i.e. “no external” forces. Thus, Newton’s third law leads to the conservation of total momentum,

to

$\boldsymbol{P} =$

$\sum_i m_i\boldsymbol{v}_i,$

$\frac{d}{dt}\boldsymbol{P} =$

$0.$

=

=$\sum_i m_i\boldsymbol{v}_i, \frac{d}{dt}\boldsymbol{P}$

$0.$

Consider the rocket of mass M moving with velocity v. After a brief instant, the velocity of the rocket is $v + \Delta v$ and the mass is $M - \Delta M$. Momentum conservation gives

$$to$$

$$Mv =$$
$$(M - \Delta M)(v + \Delta v) + \Delta M(v - v_e)$$
$$0 =$$
$$-\Delta M v + M \Delta v + \Delta M(v - v_e),$$
$$0 =$$
$$M \Delta v - \Delta M v_e.$$

=
$(M = \Delta M)(v + \Delta v) + \Delta M(v - v_e)0$
$-\Delta M v + M \Delta v + \Delta M(v - v_e), 0$
$M \Delta v - \Delta M v_e.$

In the second step we ignored the term $\Delta M \Delta v$ because it is doubly small. The last equation gives

$$to$$

$$\Delta v =$$
$$\frac{v_e}{M} \Delta M,$$
$$\frac{dv}{dt} =$$
$$\frac{v_e}{M} \frac{dM}{dt}.$$

=
$=\frac{v_e}{M} \Delta M, \frac{dv}{dt}$
$\frac{v_e}{M} \frac{dM}{dt}.$

Integrating the expression with lower limits $v_0 = 0$ and M_0, one finds

$$to$$

$$v =$$

$$v_e \int_{M_0}^{M} \frac{dM'}{M'}$$

$$v =$$

$$-v_e \ln(M/M_0)$$

$$=$$

$$-v_e \ln[(M_0 - \alpha t)/M_0].$$

=

$$=v_e \int_{M_0}^{M} \frac{dM'}{M'} v$$

$$=-v_e \ln(M/M_0)$$

$$-v_e \ln[(M_0 - \alpha t)/M_0].$$

Because the total momentum of an isolated system is constant, one can also quickly see that the center of mass of an isolated system is also constant. The center of mass is the average position of a set of masses weighted by the mass,

$$\bar{x} = \frac{\sum_i m_i x_i}{\sum_i m_i}.$$

The rate of change of $\bar{x}$ is

$$to$$

$$\dot{\bar{x}} =$$

$$\frac{1}{M} \sum_i m_i \dot{x}_i = \frac{1}{M} P_x.(8.16)$$

=

$$\frac{1}{M}\sum_i m_i \dot{x}_i = \frac{1}{M}P_x.$$

Thus if the total momentum is constant the center of mass moves at a constant velocity, and if the total momentum is zero the center of mass is fixed.

8.2.5 Conservation of Angular Momentum

Consider a case where the force always points radially,

$$\boldsymbol{F}(\boldsymbol{r}) = F(r)\hat{r},$$

where $\hat{r}$ is a unit vector pointing outward from the origin. The angular momentum is defined as

$$\boldsymbol{L} = \boldsymbol{r} \times \boldsymbol{p} = m\boldsymbol{r} \times \boldsymbol{v}.$$

The rate of change of the angular momentum is

$$to$$

$$\frac{d\boldsymbol{L}}{dt} =$$

$$m\boldsymbol{v} \times \boldsymbol{v} + m\boldsymbol{r} \times \dot{\boldsymbol{v}}$$

$$=$$

$$m\boldsymbol{v} \times \boldsymbol{v} + \boldsymbol{r} \times \boldsymbol{F} = 0.$$

=

=$m\boldsymbol{v} \times \boldsymbol{v} + m\boldsymbol{r} \times \dot{\boldsymbol{v}}$

$m\boldsymbol{v} \times \boldsymbol{v} + \boldsymbol{r} \times \boldsymbol{F} = 0.$

The first term is zero because $\boldsymbol{v}$ is parallel to itself, and the second term is zero because $\boldsymbol{F}$ is parallel to $\boldsymbol{r}$.

As an aside, one can see from the Levi-Civita symbol that the cross product of a vector with itself is zero. Here, we consider a vector

$$to$$

$$\boldsymbol{V} =$$

$$\boldsymbol{A} \times \boldsymbol{A},$$

$$V_i =$$

$$(\boldsymbol{A} \times \boldsymbol{A})_i = \sum_{jk} \epsilon_{ijk} A_j A_k.$$

=

$$=\boldsymbol{A} \times \boldsymbol{A}, V_i$$

$$(\boldsymbol{A} \times \boldsymbol{A})_i = \sum_{jk} \epsilon_{ijk} A_j A_k.$$

For any term i, there are two contributions. For example, for i denoting the x direction, either j denotes the y direction and k denotes the z direction, or vice versa, so

$$V_1 = \epsilon_{123} A_2 A_3 + \epsilon_{132} A_3 A_2.$$

This is zero by the antisymmetry of ϵ under permutations.

If the force is not radial, $\boldsymbol{r} \times \boldsymbol{F} \neq 0$ as above, and angular momentum is no longer conserved,

$$\frac{d\boldsymbol{L}}{dt} = \boldsymbol{r} \times \boldsymbol{F} \equiv \boldsymbol{\tau},$$

where $\boldsymbol{\tau}$ is the torque.

For a system of isolated particles, one can write

$$to$$

$$\frac{d}{dt}\sum_i \boldsymbol{L}_i =$$

$$\sum_{i\neq j} \boldsymbol{r}_i \times \boldsymbol{F}_{ij}$$

$$=$$

$$\frac{1}{2}\sum_{i\neq j} \boldsymbol{r}_i \times \boldsymbol{F}_{ij} + \boldsymbol{r}_j \times \boldsymbol{F}_{ji}$$

$$=$$

$$\frac{1}{2}\sum_{i\neq j} (\boldsymbol{r}_i - \boldsymbol{r}_j) \times \boldsymbol{F}_{ij} = 0,$$

=

=$\sum_{i\neq j} \boldsymbol{r}_i \times \boldsymbol{F}_{ij}$

=$\frac{1}{2}\sum_{i\neq j} \boldsymbol{r}_i \times \boldsymbol{F}_{ij} + \boldsymbol{r}_j \times \boldsymbol{F}_{ji}$

$\frac{1}{2}\sum_{i\neq j} (\boldsymbol{r}_i - \boldsymbol{r}_j) \times \boldsymbol{F}_{ij} = 0,$

where the last step used Newton's third law, $\boldsymbol{F}_{ij} = -\boldsymbol{F}_{ji}$. If the forces between the particles are radial, i.e. $\boldsymbol{F}_{ij} \mid\mid (\boldsymbol{r}_i - \boldsymbol{r}_j)$, then each term in the sum is zero and the net angular momentum is fixed. Otherwise, you could imagine an isolated system that would start spinning spontaneously.

One can write the torque about a given axis, which we will denote as $\hat{z}$, in polar coordinates, where

$$to$$

$$x =$$

$r\sin\theta\cos\phi, \quad y = r\sin\theta\cos\phi, \quad z = r\cos\theta,$ (8.17)

=

$r\sin\theta\cos\phi, \;\; y = r\sin\theta\cos\phi, \;\; z = r\cos\theta,$

to find the z component of the torque,

to

$\tau_z =$

$xF_y - yF_x$

$=$

$-r\sin\theta\left\{\cos\phi\partial_y - \sin\phi\partial_x\right\}V(x,y,z).$

=

$xF_y = yF_x$

$-r\sin\theta\left\{\cos\phi\partial_y - \sin\phi\partial_x\right\}V(x,y,z).$

One can use the chain rule to write the partial derivative w.r.t. ϕ (keeping r and θ fixed),

to

$\partial_\phi =$

$$\frac{\partial x}{\partial\phi}\partial_x + \frac{\partial_y}{\partial\phi}\partial_y + \frac{\partial z}{\partial\phi}\partial_z$$

$=$

$-r\sin\theta\sin\phi\partial_x + \sin\theta\cos\phi\partial_y.$

=

$=\frac{\partial x}{\partial\phi}\partial_x + \frac{\partial_y}{\partial\phi}\partial_y + \frac{\partial z}{\partial\phi}\partial_z$

$-r\sin\theta\sin\phi\partial_x + \sin\theta\cos\phi\partial_y.$

Combining the two equations,

to

$\tau_z =$
$-\partial_\phi V(r,\theta,\phi)$.(8.15)

=
$-\partial_\phi V(r,\theta,\phi).$

Thus, if the potential is independent of the azimuthal angle ϕ, there is no torque about the z axis and L_z is conserved.

8.2.6 Symmetries and Conservation Laws

When we derived the conservation of energy, we assumed that the potential depended only on position, not on time. If it depended explicitly on time, one can quickly see that the energy would have changed at a rate $\partial_t V(x,y,z,t)$. Note that if there is no explicit dependence on time, i.e. $V(x,y,z)$, the potential energy can depend on time through the variations of x,y,z with time. However, that variation does not lead to energy non-conservation. Further, we just saw that if a potential does not depend on the azimuthal angle about some axis, ϕ, that the angular momentum about that axis is conserved.

Now, we relate momentum conservation to translational invariance. Considering a system of particles with positions, $\boldsymbol{r}_i$, if one changed the coordinate system by a translation by a differential distance $\boldsymbol{\epsilon}$, the net potential would change by

$$to$$

$$\delta V(\boldsymbol{r}_1, \boldsymbol{r}_2 \cdots) =$$

$$\sum_i \boldsymbol{\epsilon} \cdot \nabla_i V(\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots)$$

$$=$$

$$-\sum_i \boldsymbol{\epsilon} \cdot \boldsymbol{F}_i$$

$$=$$

$$-\frac{d}{dt} \sum_i \boldsymbol{\epsilon} \cdot \boldsymbol{p}_i.$$

=

$$=\sum_i \boldsymbol{\epsilon} \cdot \nabla_i V(\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots)$$

$$=-\sum_i \boldsymbol{\epsilon} \cdot \boldsymbol{F}_i$$

$$-\frac{d}{dt} \sum_i \boldsymbol{\epsilon} \cdot \boldsymbol{p}_i.$$

Thus, if the potential is unchanged by a translation of the coordinate system, the total momentum is conserved. If the potential is translationally invariant in a given direction, defined by a unit vector, $\hat{\epsilon}$ in the $\boldsymbol{\epsilon}$ direction, one can see that

$$to$$

$$\hat{\epsilon} \cdot \nabla_i V(\boldsymbol{r}_i) =$$

0.(8.14)

=

0.

The component of the total momentum along that axis is conserved. This is rather obvious for a single particle. If $V(\boldsymbol{r})$ does not depend on some coordinate x, then the force in the x direction is $F_x = -\partial_x V = 0$, and momentum along the x direction is constant.

We showed how the total momentum of an isolated system of particle was conserved, even if the particles feel internal forces in all directions. In that case the potential energy could be written

to

$V = \sum_{i,j\leq i} V_{ij}(\boldsymbol{r}_i - \boldsymbol{r}_j).(8.14)$

In this case, a translation leads to $\boldsymbol{r}_i \rightarrow \boldsymbol{r}_i + \boldsymbol{\epsilon}$, with the translation equally affecting the coordinates of each particle. Because the potential depends only on the relative coordinates, δV is manifestly zero. If one were to go through the exercise of calculating δV for small $\boldsymbol{\epsilon}$, one would find that the term $\nabla_i V(\boldsymbol{r}_i - \boldsymbol{r}_j)$ would be canceled by the term $\nabla_j V(\boldsymbol{r}_i - \boldsymbol{r}_j)$.

The relation between symmetries of the potential and conserved quantities (also called constants of motion) is one of the most profound concepts one should gain from this course. It plays a critical role in all fields of physics. This is especially true in quantum mechanics, where a quantity A is conserved if its operator commutes with the Hamiltonian. For example if the momentum operator $-i\hbar\partial_x$ commutes with the Hamiltonian, momentum is conserved, and clearly this operator commutes if the Hamiltonian (which represents the total energy, not just the potential) does not depend on x. Also in quantum mechanics the angular momentum operator is $L_z = -i\hbar\partial_\phi$. In fact, if the potential is unchanged by rotations about some axis, angular momentum about that axis is conserved. We return to this concept, from a more formal perspective, later in the course when Lagrangian mechanics is presented.

8.2.7 Bulding a code for the Earth-Sun system

We will now venture into a study of a system which is energy conserving. The aim is to see if we (since it is not possible to solve the general equations analytically) we can develop stable numerical algorithms whose results we can trust!

We solve the equations of motion numerically. We will also compute quantities like the energy numerically.

We start with a simpler case first, the Earth-Sun system in two dimensions only. The gravitational force F_G on the earth from the sun is

$$\boldsymbol{F}_G = -\frac{GM_\odot M_E}{r^3}\boldsymbol{r},$$

where G is the gravitational constant,

$$M_E = 6\times 10^{24}\mathrm{Kg},$$

the mass of Earth,

$$M_{\odot} = 2 \times 10^{30}\mathrm{Kg},$$

the mass of the Sun and

$$r = 1.5 \times 10^{11}\mathrm{m},$$

is the distance between Earth and the Sun. The latter defines what we call an astronomical unit **AU**. From Newton's second law we have then for the x direction

$$\frac{d^2x}{dt^2} = -\frac{F_x}{M_E},$$

and

$$\frac{d^2y}{dt^2} = -\frac{F_y}{M_E},$$

for the y direction.

Here we will use that $x = r\cos(\theta)$, $y = r\sin(\theta)$ and

$$r = \sqrt{x^2 + y^2}.$$

We can rewrite

$$F_x = -\frac{GM_{\odot}M_E}{r^2}\cos(\theta) = -\frac{GM_{\odot}M_E}{r^3}x,$$

and

$$F_y = -\frac{GM_{\odot}M_E}{r^2}\sin(\theta) = -\frac{GM_{\odot}M_E}{r^3}y,$$

for the y direction.

We can rewrite these two equations

$$F_x = -\frac{GM_{\odot}M_E}{r^2}\cos(\theta) = -\frac{GM_{\odot}M_E}{r^3}x,$$

and

$$F_y = -\frac{GM_{\odot}M_E}{r^2}\sin(\theta) = -\frac{GM_{\odot}M_E}{r^3}y,$$

as four first-order coupled differential equations

43

<<<!!MATH_BLOCK

44

<<<!!MATH_BLOCK

45

<<<!!MATH_BLOCK

$$\frac{dy}{dt} = v_y.$$

8.2.8 Building a code for the solar system, final coupled equations

The four coupled differential equations

47

<<<!!MATH_BLOCK

48

<<<!!MATH_BLOCK

49

<<<!!MATH_BLOCK

$$\frac{dy}{dt} = v_y,$$

can be turned into dimensionless equations or we can introduce astronomical units with 1 AU = 1.5×10^{11}.

Using the equations from circular motion (with $r = 1\mathrm{AU}$)

$$\frac{M_E v^2}{r} = F = \frac{GM_\odot M_E}{r^2},$$

we have

$$GM_\odot = v^2 r,$$

and using that the velocity of Earth (assuming circular motion) is $v = 2\pi r/\mathrm{yr} = 2\pi \mathrm{AU}/\mathrm{yr}$, we have

$$GM_\odot = v^2 r = 4\pi^2 \frac{(\mathrm{AU})^3}{\mathrm{yr}^2}.$$

8.2.9 Building a code for the solar system, discretized equations

The four coupled differential equations can then be discretized using Euler's method as (with step length h)

54

<<<!!MATH_BLOCK

55

<<<!!MATH_BLOCK

56

<<<!!MATH_BLOCK

$$y_{i+1} = y_i + hv_{y,i},$$

8.2.10 Code Example with Euler's Method

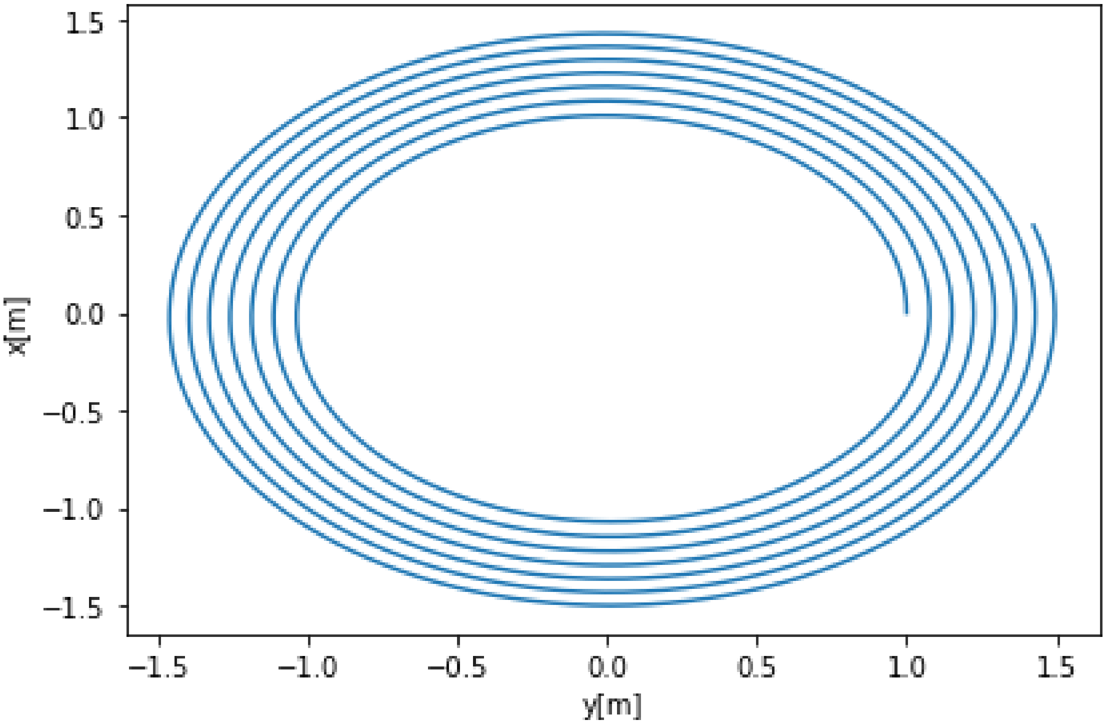
The code here implements Euler's method for the Earth-Sun system using a more compact way of representing the vectors. Alternatively, you could have spelled out all the variables v_x, v_y, x and y as one-dimensional arrays.

```
%matplotlib inline

# Common imports
import numpy as np
import pandas as pd
from math import *
import matplotlib.pyplot as plt
import os

# Where to save the figures and data files
PROJECT_ROOT_DIR = "Results"
FIGURE_ID = "Results/FigureFiles"
DATA_ID = "DataFiles/"

if not os.path.exists(PROJECT_ROOT_DIR):
    os.mkdir(PROJECT_ROOT_DIR)

if not os.path.exists(FIGURE_ID):
    os.makedirs(FIGURE_ID)

if not os.path.exists(DATA_ID):
    os.makedirs(DATA_ID)

def image_path(fig_id):
    return os.path.join(FIGURE_ID, fig_id)

def data_path(dat_id):
    return os.path.join(DATA_ID, dat_id)

def save_fig(fig_id):
    plt.savefig(image_path(fig_id) + ".png", format='png')


DeltaT = 0.001
#set up arrays
tfinal = 10 # in years
n = ceil(tfinal/DeltaT)
# set up arrays for t, a, v, and x
t = np.zeros(n)
v = np.zeros((n,2))
r = np.zeros((n,2))
# Initial conditions as compact 2-dimensional arrays
r0 = np.array([1.0,0.0])
v0 = np.array([0.0,2*pi])
r[0] = r0
v[0] = v0
Fourpi2 = 4*pi*pi
# Start integrating using Euler's method
for i in range(n-1):
    # Set up the acceleration
    # Here you could have defined your own function for this
    rabs = sqrt(sum(r[i]*r[i]))
    a =  -Fourpi2*r[i]/(rabs**3)
    # update velocity, time and position using Euler's forward method
    v[i+1] = v[i] + DeltaT*a
    r[i+1] = r[i] + DeltaT*v[i]
    t[i+1] = t[i] + DeltaT
```

(continues on next page)

(continued from previous page)

```
# Plot position as function of time
fig, ax = plt.subplots()
#ax.set_xlim(0, tfinal)
ax.set_ylabel('x[m]')
ax.set_xlabel('y[m]')
ax.plot(r[:,0], r[:,1])
fig.tight_layout()
save_fig("EarthSunEuler")
plt.show()
```



8.2.11 Problems with Euler's Method

We notice here that Euler's method doesn't give a stable orbit. It means that we cannot trust Euler's method. In a deeper way, as we will see in homework 5, Euler's method does not conserve energy. It is an example of an integrator which is not symplectic.

Here we present thus two methods, which with simple changes allow us to avoid these pitfalls. The simplest possible extension is the so-called Euler-Cromer method. The changes we need to make to our code are indeed marginal here. We need simply to replace

```
    r[i+1] = r[i] + DeltaT*v[i]
```

in the above code with the velocity at the new time t_{i+1}

```
    r[i+1] = r[i] + DeltaT*v[i+1]
```

By this simple caveat we get stable orbits. Below we derive the Euler-Cromer method as well as one of the most utlized algorithms for sovling the above type of problems, the so-called Velocity-Verlet method.

8.2.12 Deriving the Euler-Cromer Method

Let us repeat Euler's method. We have a differential equation

$$y'(t_i) = f(t_i, y_i)$$

and if we truncate at the first derivative, we have from the Taylor expansion

$$y_{i+1} = y(t_i) + (\Delta t)f(t_i, y_i) + O(\Delta t^2),$$

which when complemented with $t_{i+1} = t_i + \Delta t$ forms the algorithm for the well-known Euler method. Note that at every step we make an approximation error of the order of $O(\Delta t^2)$, however the total error is the sum over all steps $N = (b - a)/(\Delta t)$ for $t \in [a, b]$, yielding thus a global error which goes like $NO(\Delta t^2) \approx O(\Delta t)$.

To make Euler's method more precise we can obviously decrease Δt (increase N), but this can lead to loss of numerical precision. Euler's method is not recommended for precision calculation, although it is handy to use in order to get a first view on how a solution may look like.

Euler's method is asymmetric in time, since it uses information about the derivative at the beginning of the time interval. This means that we evaluate the position at y_1 using the velocity at v_0. A simple variation is to determine x_{n+1} using the velocity at v_{n+1}, that is (in a slightly more generalized form)

$$y_{n+1} = y_n + v_{n+1} + O(\Delta t^2)$$

and

$$v_{n+1} = v_n + (\Delta t)a_n + O(\Delta t^2).$$

The acceleration a_n is a function of $a_n(y_n, v_n, t_n)$ and needs to be evaluated as well. This is the Euler-Cromer method.

Exercise: go back to the above code with Euler's method and add the Euler-Cromer method.

8.2.13 Deriving the Velocity-Verlet Method

Let us stay with x (position) and v (velocity) as the quantities we are interested in.

We have the Taylor expansion for the position given by

$$x_{i+1} = x_i + (\Delta t)v_i + \frac{(\Delta t)^2}{2}a_i + O((\Delta t)^3).$$

The corresponding expansion for the velocity is

$$v_{i+1} = v_i + (\Delta t)a_i + \frac{(\Delta t)^2}{2}v_i^{(2)} + O((\Delta t)^3).$$

Via Newton's second law we have normally an analytical expression for the derivative of the velocity, namely

$$a_i = \frac{d^2x}{dt^2}|_i = \frac{dv}{dt}|_i = \frac{F(x_i, v_i, t_i)}{m}.$$

If we add to this the corresponding expansion for the derivative of the velocity

$$v_{i+1}^{(1)} = a_{i+1} = a_i + (\Delta t)v_i^{(2)} + O((\Delta t)^2) = a_i + (\Delta t)v_i^{(2)} + O((\Delta t)^2),$$

and retain only terms up to the second derivative of the velocity since our error goes as $O(h^3)$, we have

$$(\Delta t)v_i^{(2)} \approx a_{i+1} - a_i.$$

We can then rewrite the Taylor expansion for the velocity as

$$v_{i+1} = v_i + \frac{(\Delta t)}{2}\left(a_{i+1} + a_i\right) + O((\Delta t)^3).$$

8.2.14 The velocity Verlet method

Our final equations for the position and the velocity become then

$$x_{i+1} = x_i + (\Delta t)v_i + \frac{(\Delta t)^2}{2}a_i + O((\Delta t)^3),$$

and

$$v_{i+1} = v_i + \frac{(\Delta t)}{2}\left(a_{i+1} + a_i\right) + O((\Delta t)^3).$$

Note well that the term a_{i+1} depends on the position at x_{i+1}. This means that you need to calculate the position at the updated time t_{i+1} before the computing the next velocity. Note also that the derivative of the velocity at the time t_i used in the updating of the position can be reused in the calculation of the velocity update as well.

8.2.15 Adding the Velocity-Verlet Method

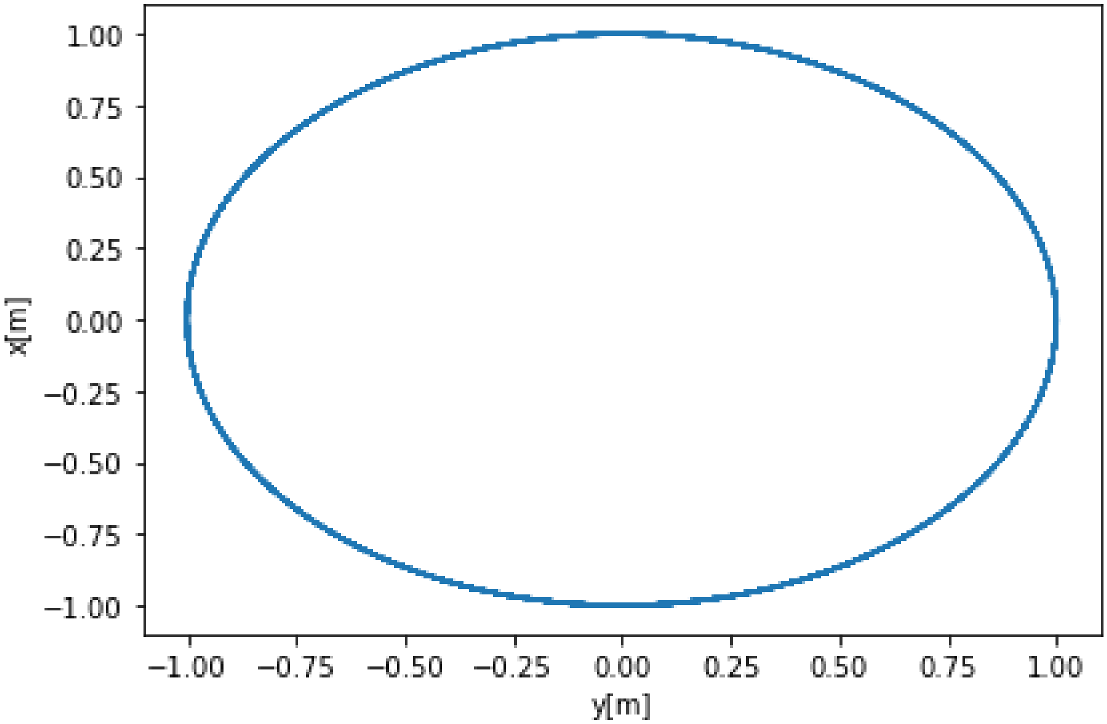
We can now easily add the Verlet method to our original code as

```
DeltaT = 0.01
#set up arrays
tfinal = 10
n = ceil(tfinal/DeltaT)
# set up arrays for t, a, v, and x
```

(continues on next page)

(continued from previous page)

```
t = np.zeros(n)
v = np.zeros((n,2))
r = np.zeros((n,2))
# Initial conditions as compact 2-dimensional arrays
r0 = np.array([1.0,0.0])
v0 = np.array([0.0,2*pi])
r[0] = r0
v[0] = v0
Fourpi2 = 4*pi*pi
# Start integrating using the Velocity-Verlet  method
for i in range(n-1):
    # Set up forces, air resistance FD, note now that we need the norm of the vecto
    # Here you could have defined your own function for this
    rabs = sqrt(sum(r[i]*r[i]))
    a =  -Fourpi2*r[i]/(rabs**3)
    # update velocity, time and position using the Velocity-Verlet method
    r[i+1] = r[i] + DeltaT*v[i]+0.5*(DeltaT**2)*a
    rabs = sqrt(sum(r[i+1]*r[i+1]))
    anew = -4*(pi**2)*r[i+1]/(rabs**3)
    v[i+1] = v[i] + 0.5*DeltaT*(a+anew)
    t[i+1] = t[i] + DeltaT
# Plot position as function of time
fig, ax = plt.subplots()
ax.set_ylabel('x[m]')
ax.set_xlabel('y[m]')
ax.plot(r[:,0], r[:,1])
fig.tight_layout()
save_fig("EarthSunVV")
plt.show()
```



You can easily generalize the calculation of the forces by defining a function which takes in as input the various variables. We leave this as a challenge to you.

8.2.16 Studying Energy Conservation

In order to study the conservation of energy, we will need to perform a numerical integration, unless we can integrate analytically. Here we present the Trapezoidal rule as a the simplest possible approximation.

8.2.17 Numerical Integration

It is also useful to consider methods to integrate numerically. 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

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

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

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

which results in

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

8.2.18 Numerical Integration

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.

8.2.19 Newton-Cotes Quadrature or equal-step methods

The integral

$$I = \int_a^b f(x)dx$$

has a very simple meaning. The integral is the area enscribed 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.

8.2.20 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 $f(x)\approx a_0+a_1x$.

8.2.21 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.

8.2.22 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

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

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

8.2.23 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

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

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

8.2.24 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 $\xi \in [a, b]$.

8.2.25 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))
```

```
Relative error=  4.112453549290521e-05
```

8.2.26 The trapezoidal rule

The other integral gives

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

and adding up we obtain

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

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

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

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 $f_h=f(x_0+h)$.

8.2.27 Error in the trapezoidal rule

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

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

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 $\xi\in[a,b]$.

8.2.28 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 $f(a+h) + f(a+2h) + f(a+3h) + \cdots + f(b-h)$. Each step in the loop corresponds to a given value $a + nh$.
- Multiply the final result by h and add $hf(a)/2$ and $hf(b)/2$.

8.2.29 Trapezoidal Rule

We use the same function and integrate now using the trapoezoidal 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))
```

```
Relative error=  8.224805627923717e-05
```

8.2.30 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

$$f(x) \approx P_2(x) = a_0 + a_1 x + a_2 x^2.$$

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. (*18*) we obtain

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

which is Simpson's rule.

8.2.31 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. (*17*) to be

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

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

8.2.32 Mathematical expressions for the truncation error

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

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

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.

8.2.33 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 $\frac{h}{3}$.

8.2.34 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)
```

(continues on next page)

(continued from previous page)

```
    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))
```

```
Relative error=  5.412252157986472e-09
```

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.

8.3 Oscillations

Morten Hjorth-Jensen, Department of Physics and Astronomy and National Superconducting Cyclotron Laboratory, Michigan State University, USA and Department of Physics, University of Oslo, Norway

Date: **Feb 22, 2020**

8.3.1 Harmonic Oscillator

The harmonic oscillator is omnipresent in physics. Although you may think of this as being related to springs, it, or an equivalent mathematical representation, appears in just about any problem where a mode is sitting near its potential energy minimum. At that point, $\partial_x V(x) = 0$, and the first non-zero term (aside from a constant) in the potential energy is that of a harmonic oscillator. In a solid, sound modes (phonons) are built on a picture of coupled harmonic oscillators, and in relativistic field theory the fundamental interactions are also built on coupled oscillators positioned infinitesimally close to one another in space. The phenomena of a resonance of an oscillator driven at a fixed frequency plays out repeatedly in atomic, nuclear and high-energy physics, when quantum mechanically the evolution of a state oscillates according to e^{-iEt} and exciting discrete quantum states has very similar mathematics as exciting discrete states of an oscillator.

The potential energy for a single particle as a function of its position x can be written as a Taylor expansion about some point x_0

$$V(x) = V(x_0) + (x-x_0)\left.\partial_x V(x)\right|_{x_0} + \frac{1}{2}(x-x_0)^2\left.\partial_x^2 V(x)\right|_{x_0} + \frac{1}{3!}\left.\partial_x^3 V(x)\right|_{x_0} + \cdots$$

If the position x_0 is at the minimum of the resonance, the first two non-zero terms of the potential are

$$to$$

$$V(x) \approx$$

$$V(x_0) + \frac{1}{2}(x - x_0)^2 \; \partial_x^2 V(x)\big|_{x_0},$$

$$=$$

$$V(x_0) + \frac{1}{2}k(x - x_0)^2, \quad k \equiv \partial_x^2 V(x)\big|_{x_0},$$

$$F =$$

$$-\partial_x V(x) = -k(x - x_0).$$

$V{=}(x_0) + \frac{1}{2}(x - x_0)^2 \; \partial_x^2 V(x)\big|_{x_0},$
$V{=}(x_0) + \frac{1}{2}k(x - x_0)^2, \quad k \equiv \partial_x^2 V(x)\big|_{x_0}, F$
$-\partial_x V(x) = -k(x - x_0).$

Put into Newton's 2nd law (assuming $x_0 = 0$),

$$to$$

$$m\ddot{x} =$$

$$-kx,$$

$$x =$$

$A\cos(\omega_0 t - \phi), \quad \omega_0 = \sqrt{k/m}.$(8.25)

=
=$-kx, x$
$A\cos(\omega_0 t - \phi), \quad \omega_0 = \sqrt{k/m}.$

Here A and ϕ are arbitrary. Equivalently, one could have written this as $A\cos(\omega_0 t) + B\sin(\omega_0 t)$, or as the real part of $Ae^{i\omega_0 t}$. In this last case A could be an arbitrary complex constant. Thus, there are 2 arbitrary constants (either A and B or A and ϕ, or the real and imaginary part of one complex constant. This is the expectation for a second order differential equation, and also agrees with the physical expectation that if you know a particle's initial velocity and position you should be able to define its future motion, and that those two arbitrary conditions should translate to two arbitrary constants.

A key feature of harmonic motion is that the system repeats itself after a time $T = 1/f$, where f is the frequency, and $\omega = 2\pi f$ is the angular frequency. The period of the motion is independent of the amplitude. However, this independence is only exact when one can neglect higher terms of the potential, $x^3, x^4 \cdots$. Once can neglect these terms for sufficiently small amplitudes, and for larger amplitudes the motion is no longer purely sinusoidal, and even though the motion repeats itself, the time for repeating the motion is no longer independent of the amplitude.

One can also calculate the velocity and the kinetic energy as a function of time,

$$to$$

$$\dot{x} =$$

$$-\omega_0 A\sin(\omega_0 t - \phi),$$

$$K =$$

$$\frac{1}{2}m\dot{x}^2 = \frac{m\omega_0^2 A^2}{2}\sin^2(\omega_0 t - \phi),$$

$$=$$

$$\frac{k}{2}A^2\sin^2(\omega_0 t - \phi).$$

$$=$$

$$= -\omega_0 A\sin(\omega_0 t - \phi), K$$

$$= \frac{1}{2}m\dot{x}^2 = \frac{m\omega_0^2 A^2}{2}\sin^2(\omega_0 t - \phi),$$

$$\frac{k}{2}A^2\sin^2(\omega_0 t - \phi).$$

The total energy is then

$$E = K + V = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}kx^2 = \frac{1}{2}kA^2.$$

The total energy then goes as the square of the amplitude.

A pendulum is an example of a harmonic oscillator. By expanding the kinetic and potential energies for small angles find the frequency for a pendulum of length L with all the mass m centered at the end by writing the eq.s of motion in the form of a harmonic oscillator.

The potential energy and kinetic energies are (for x being the displacement)

$$to$$

$$V =$$

$$mgL(1-\cos\theta) \approx mgL\frac{x^2}{2L^2},$$

$$K =$$

$$\frac{1}{2}mL^2\dot{\theta}^2 \approx \frac{m}{2}\dot{x}^2.$$

=

$=mgL(1-\cos\theta) \approx mgL\frac{x^2}{2L^2}, K$

$\frac{1}{2}mL^2\dot{\theta}^2 \approx \frac{m}{2}\dot{x}^2.$

For small x Newton's 2nd law becomes

$$m\ddot{x} = -\frac{mg}{L}x,$$

and the spring constant would appear to be $k = mg/L$, which makes the frequency equal to $\omega_0 = \sqrt{g/L}$. Note that the frequency is independent of the mass.

8.3.2 Damped Oscillators

We consider only the case where the damping force is proportional to the velocity. This is counter to dragging friction, where the force is proportional in strength to the normal force and independent of velocity, and is also inconsistent with wind resistance, where the magnitude of the drag force is proportional the square of the velocity. Rolling resistance does seem to be mainly proportional to the velocity. However, the main motivation for considering damping forces proportional to the velocity is that the math is more friendly. This is because the differential equation is linear, i.e. each term is of order x, $\dot{x}$, $\ddot{x}\cdots$, or even terms with no mention of x, and there are no terms such as x^2 or $x\ddot{x}$. The equations of motion for a spring with damping force $-b\dot{x}$ are

$$m\ddot{x} + b\dot{x} + kx = 0.$$

Just to make the solution a bit less messy, we rewrite this equation as

$$\ddot{x}+2\beta\dot{x}+\omega_0^2x=0,\quad \beta\equiv b/2m,\ \omega_0\equiv\sqrt{k/m}.$$

Both β and ω have dimensions of inverse time. To find solutions (see appendix C in the text) you must make an educated guess at the form of the solution. To do this, first realize that the solution will need an arbitrary normalization A because the equation is linear. Secondly, realize that if the form is

$$x=Ae^{rt}$$

that each derivative simply brings out an extra power of r. This means that the Ae^{rt} factors out and one can simply solve for an equation for r. Plugging this form into Eq. (*4*),

$$r^2+2\beta r+\omega_0^2=0.$$

Because this is a quadratic equation there will be two solutions,

$$r=-\beta\pm\sqrt{\beta^2-\omega_0^2}.$$

We refer to the two solutions as r_1 and r_2 corresponding to the $+$ and $-$ roots. As expected, there should be two arbitrary constants involved in the solution,

$$x=A_1e^{r_1t}+A_2e^{r_2t},$$

where the coefficients A_1 and A_2 are determined by initial conditions.

The roots listed above, $\sqrt{\omega_0^2 - \beta_0^2}$, will be imaginary if the damping is small and $\beta < \omega_0$. In that case, r is complex and the factor ert will have some oscillatory behavior. If the roots are real, there will only be exponentially decaying solutions. There are three cases:

Underdamped: $\beta < \omega_0$

$$to$$

$$x =$$

$$A_1 e^{-\beta t} e^{i\omega' t} + A_2 e^{-\beta t} e^{-i\omega' t}, \;\; \omega' \equiv \sqrt{\omega_0^2 - \beta^2}$$

$$=$$

$$(A_1 + A_2) e^{-\beta t} \cos \omega' t + i(A_1 - A_2) e^{-\beta t} \sin \omega' t.$$

$$=$$

$$A_1 e^{-\beta t} e^{i\omega' t} + A_2 e^{-\beta t} e^{-i\omega' t}, \;\; \omega' \!= \!\equiv \sqrt{\omega_0^2 - \beta^2}$$

$$(A_1 + A_2) e^{-\beta t} \cos \omega' t + i(A_1 - A_2) e^{-\beta t} \sin \omega' t.$$

Here we have made use of the identity $e^{i\omega' t} = \cos \omega' t + i \sin \omega' t$. Because the constants are arbitrary, and because the real and imaginary parts are both solutions individually, we can simply consider the real part of the solution alone:

$$to$$

$$x =$$

$$B_1 e^{-\beta t} \cos \omega' t + B_2 e^{-\beta t} \sin \omega' t,$$

$$\omega' \equiv$$

$$\sqrt{\omega_0^2 - \beta^2}.$$

$$=$$

$$B_1 e^{-\beta t} \cos \omega' t + B_2 e^{-\beta t} \sin \omega' t, \omega'$$

$$\sqrt{\omega_0^2 - \beta^2}.$$

Critical dampling: $\beta = \omega_0$

In this case the two terms involving r_1 and r_2 are identical because $\omega' = 0$. Because we need to arbitrary constants, there needs to be another solution. This is found by simply guessing, or by taking the limit of $\omega' \to 0$ from the underdamped solution. The solution is then

$$x = Ae^{-\beta t} + Bte^{-\beta t}.$$

The critically damped solution is interesting because the solution approaches zero quickly, but does not oscillate. For a problem with zero initial velocity, the solution never crosses zero. This is a good choice for designing shock absorbers or swinging doors.

Overdamped: $\beta > \omega_0$

$$to$$

$$x =$$

$$A_1 \exp -(\beta + \sqrt{\beta^2 - \omega_0^2})t + A_2 \exp -(\beta - \sqrt{\beta^2 - \omega_0^2})t(8.29)$$

=

$$A_1 \exp -(\beta + \sqrt{\beta^2 - \omega_0^2})t + A_2 \exp -(\beta - \sqrt{\beta^2 - \omega_0^2})t$$

This solution will also never pass the origin more than once, and then only if the initial velocity is strong and initially toward zero.

Given b, m and ω_0, find $x(t)$ for a particle whose initial position is $x = 0$ and has initial velocity v_0 (assuming an underdamped solution).

The solution is of the form,

$$to$$

$$x =$$

$$e^{-\beta t}\left[A_1\cos(\omega' t) + A_2\sin\omega' t\right],$$

$$\dot{x} =$$

$$-\beta x + \omega' e^{-\beta t}\left[-A_1\sin\omega' t + A_2\cos\omega' t\right].$$

$$\omega' \equiv$$

$$\sqrt{\omega_0^2 - \beta^2}, \quad \beta \equiv b/2m.$$

$=$

$e^{-\beta t}\left[A_1\cos(\omega' t) + A_2\sin\omega' t\right], \dot{x}$
$-\beta x + \omega' e^{-\beta t}\left[-A_1\sin\omega' t + A_2\cos\omega' t\right].\omega'$
$\sqrt{\omega_0^2 - \beta^2}, \quad \beta \equiv b/2m.$

From the initial conditions, $A_1 = 0$ because $x(0) = 0$ and $\omega' A_2 = v_0$. So

$$x = \frac{v_0}{\omega'} e^{-\beta t}\sin\omega' t.$$

8.3.3 Our Sliding Block Code

Here we study first the case without additional friction term and scale our equation in terms of a dimensionless time τ.

Let us remind ourselves about the differential equation we want to solve (the general case with damping due to friction)

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx(t) = 0.$$

We divide by m and introduce $\omega_0^2 = \sqrt{k/m}$ and obtain

$$\frac{d^2x}{dt^2} + \frac{b}{m}\frac{dx}{dt} + \omega_0^2 x(t) = 0.$$

Thereafter we introduce a dimensionless time $\tau = t\omega_0$ (check that the dimensionality is correct) and rewrite our equation as

$$\frac{d^2x}{d\tau^2} + \frac{b}{m\omega_0}\frac{dx}{d\tau} + x(\tau) = 0,$$

which gives us

$$\frac{d^2x}{d\tau^2} + \frac{b}{m\omega_0}\frac{dx}{d\tau} + x(\tau) = 0.$$

We then define $\gamma = b/(2m\omega_0)$ and rewrite our equations as

$$\frac{d^2x}{d\tau^2} + 2\gamma\frac{dx}{d\tau} + x(\tau) = 0.$$

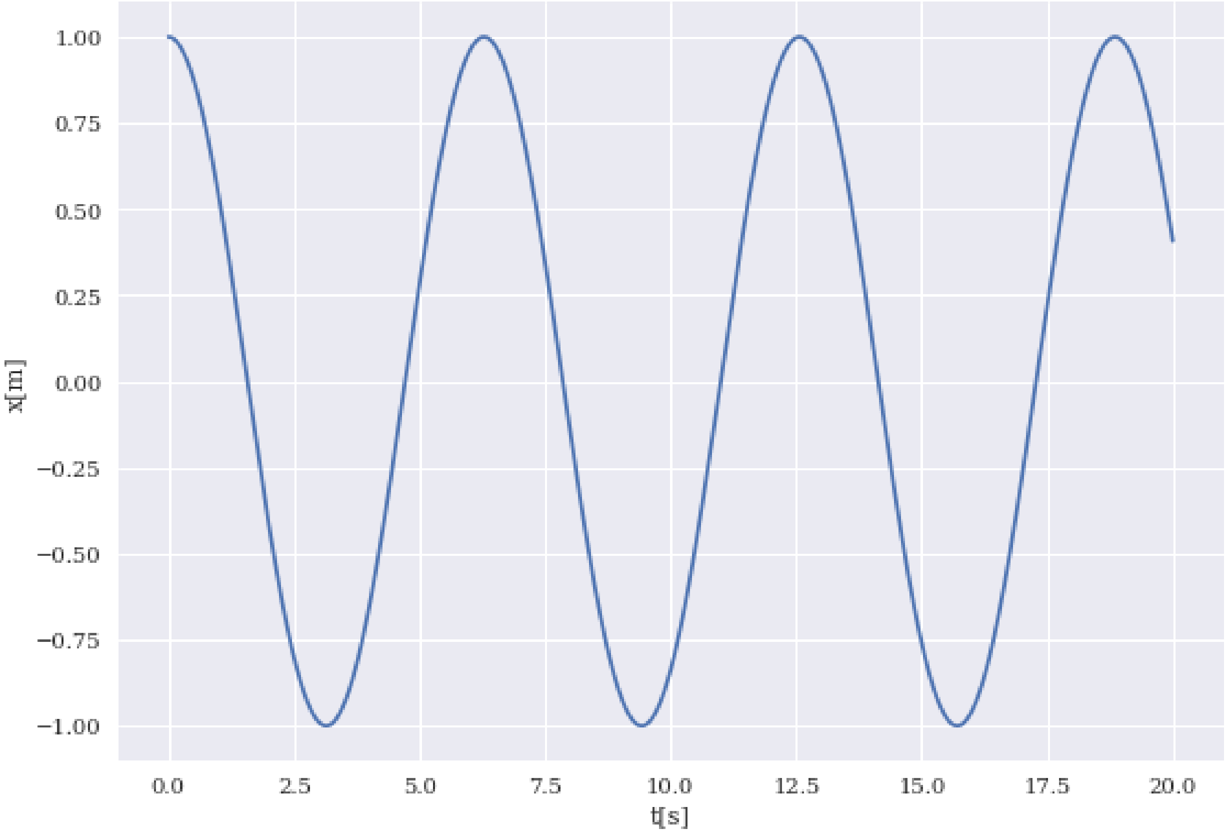
This is the equation we will code below. The first version employs the Euler-Cromer method.

```
%matplotlib inline

# Common imports
import numpy as np
import pandas as pd
from math import *
import matplotlib.pyplot as plt
import os

# Where to save the figures and data files
PROJECT_ROOT_DIR = "Results"
FIGURE_ID = "Results/FigureFiles"
DATA_ID = "DataFiles/"

if not os.path.exists(PROJECT_ROOT_DIR):
    os.mkdir(PROJECT_ROOT_DIR)

if not os.path.exists(FIGURE_ID):
    os.makedirs(FIGURE_ID)

if not os.path.exists(DATA_ID):
    os.makedirs(DATA_ID)

def image_path(fig_id):
    return os.path.join(FIGURE_ID, fig_id)

def data_path(dat_id):
    return os.path.join(DATA_ID, dat_id)

def save_fig(fig_id):
    plt.savefig(image_path(fig_id) + ".png", format='png')


from pylab import plt, mpl
plt.style.use('seaborn')
mpl.rcParams['font.family'] = 'serif'

DeltaT = 0.001
#set up arrays
tfinal = 20 # in years
n = ceil(tfinal/DeltaT)
# set up arrays for t, v, and x
t = np.zeros(n)
v = np.zeros(n)
x = np.zeros(n)
# Initial conditions as simple one-dimensional arrays of time
x0 =  1.0
v0 = 0.0
x[0] = x0
v[0] = v0
gamma = 0.0
# Start integrating using Euler-Cromer's method
```

(continues on next page)

(continued from previous page)

```
for i in range(n-1):
    # Set up the acceleration
    # Here you could have defined your own function for this
    a =  -2*gamma*v[i]-x[i]
    # update velocity, time and position
    v[i+1] = v[i] + DeltaT*a
    x[i+1] = x[i] + DeltaT*v[i+1]
    t[i+1] = t[i] + DeltaT
# Plot position as function of time
fig, ax = plt.subplots()
#ax.set_xlim(0, tfinal)
ax.set_ylabel('x[m]')
ax.set_xlabel('t[s]')
ax.plot(t, x)
fig.tight_layout()
save_fig("BlockEulerCromer")
plt.show()
```



When setting up the value of γ we see that for $\gamma = 0$ we get the simple oscillatory motion with no damping. Choosing $\gamma < 1$ leads to the classical underdamped case with oscillatory motion, but where the motion comes to an end.

Choosing $\gamma = 1$ leads to what normally is called critical damping and $\gamma > 1$ leads to critical overdamping. Try it out and try also to change the initial position and velocity. Setting $\gamma = 1$ yields a situation, as discussed above, where the solution approaches quickly zero and does not oscillate. With zero initial velocity it will never cross zero.

8.3.4 Sinusoidally Driven Oscillators

Here, we consider the force

$$F=-kx-b\dot{x}+F_0\cos\omega t,$$

which leads to the differential equation

$$\ddot{x}+2\beta\dot{x}+\omega_0^2x=(F_0/m)\cos\omega t.$$

Consider a single solution with no arbitrary constants, which we will call a {\it particular solution}, $x_p(t)$. It should be emphasized that this is {\bf A} particular solution, because there exists an infinite number of such solutions because the general solution should have two arbitrary constants. Now consider solutions to the same equation without the driving term, which include two arbitrary constants. These are called either {\it homogenous solutions} or {\it complementary solutions}, and were given in the previous section, e.g. Eq. (*9*) for the underdamped case. The homogenous solution already incorporates the two arbitrary constants, so any sum of a homogenous solution and a particular solution will represent the {\it general solution} of the equation. The general solution incorporates the two arbitrary constants A and B to accommodate the two initial conditions. One could have picked a different particular solution, i.e. the original particular solution plus any homogenous solution with the arbitrary constants A_p and B_p chosen at will. When one adds in the homogenous solution, which has adjustable constants with arbitrary constants A' and B', to the new particular solution, one can get the same general solution by simply adjusting the new constants such that $A'+A_p=A$ and $B'+B_p=B$. Thus, the choice of A_p and B_p are irrelevant, and when choosing the particular solution it is best to make the simplest choice possible.

To find a particular solution, one first guesses at the form,

$$x_p(t)=D\cos(\omega t-\delta),$$

and rewrite the differential equation as

$$D\left\{-\omega^2\cos(\omega t-\delta)-2\beta\omega\sin(\omega t-\delta)+\omega_0^2\cos(\omega t-\delta)\right\}=\frac{F_0}{m}\cos(\omega t).$$

One can now use angle addition formulas to get

$$to$$

$$D\left\{(-\omega^2\cos\delta + 2\beta\omega\sin\delta + \omega_0^2\cos\delta)\cos(\omega t)\right.$$

$$\left.+(-\omega^2\sin\delta - 2\beta\omega\cos\delta + \omega_0^2\sin\delta)\sin(\omega t)\right\} =$$

$$\frac{F_0}{m}\cos(\omega t).$$

$$+(-\omega^2\sin\delta - 2\beta\omega = \cos\delta + \omega_0^2\sin\delta)\sin(\omega t)\}$$

$$\frac{F_0}{m}\cos(\omega t).$$

Both the $\cos$ and $\sin$ terms need to equate if the expression is to hold at all times. Thus, this becomes two equations

$$to$$

$$D\left\{-\omega^2\cos\delta + 2\beta\omega\sin\delta + \omega_0^2\cos\delta\right\} =$$

$$\frac{F_0}{m}$$

$$-\omega^2\sin\delta - 2\beta\omega\cos\delta + \omega_0^2\sin\delta =$$

$$0.$$

$$=$$

$$\frac{F_0}{m} - \omega^2 \sin\delta - 2\beta\omega\cos\delta + \omega_0^2\sin\delta$$

$$=$$

$$0.$$

After dividing by $\cos\delta$, the lower expression leads to

$$\tan\delta = \frac{2\beta\omega}{\omega_0^2 - \omega^2}.$$

Using the identities $\tan^2 + 1 = \csc^2$ and $\sin^2 + \cos^2 = 1$, one can also express $\sin\delta$ and $\cos\delta$,

$$to$$

$$\sin\delta =$$

$$\frac{2\beta\omega}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\omega^2\beta^2}},$$

$$\cos\delta =$$

$$\frac{(\omega_0^2 - \omega^2)}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\omega^2\beta^2}}$$

=

$$= \frac{2\beta\omega}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\omega^2\beta^2}}, \cos\delta$$

$$\frac{(\omega_0^2 - \omega^2)}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\omega^2\beta^2}}$$

Inserting the expressions for $\cos\delta$ and $\sin\delta$ into the expression for D,

$$D = \frac{F_0/m}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\omega^2\beta^2}}.$$

For a given initial condition, e.g. initial displacement and velocity, one must add the homogenous solution then solve for the two arbitrary constants. However, because the homogenous solutions decay with time as $e^{-\beta t}$, the particular solution is all that remains at large times, and is therefore the steady state solution. Because the arbitrary constants are all in the homogenous solution, all memory of the initial conditions are lost at large times, $t >> 1/\beta$.

The amplitude of the motion, D, is linearly proportional to the driving force (F_0/m), but also depends on the driving frequency ω. For small β the maximum will occur at $\omega = \omega_0$. This is referred to as a resonance. In the limit $\beta \rightarrow 0$ the amplitude at resonance approaches infinity.

8.3.5 Alternative Derivation for Driven Oscillators

Here, we derive the same expressions as in Equations (*13*) and (*16*) but express the driving forces as

to

$F(t) =$

$F_0 e^{i\omega t}$, (8.31)

=

$F_0 e^{i\omega t}$,

rather than as $F_0 \cos \omega t$. The real part of F is the same as before. For the differential equation,

to

$\ddot{x} + 2\beta\dot{x} + \omega_0^2 x =$

$\frac{F_0}{m} e^{i\omega t}$, (8.31)

=

$\frac{F_0}{m} e^{i\omega t}$,

one can treat $x(t)$ as an imaginary function. Because the operations d^2/dt^2 and d/dt are real and thus do not mix the real and imaginary parts of $x(t)$, Eq. (*17*) is effectively 2 equations. Because $e^{\omega t} = \cos \omega t + i \sin \omega t$, the real part of the solution for $x(t)$ gives the solution for a driving force $F_0 \cos \omega t$, and the imaginary part of x corresponds to the case where the driving force is $F_0 \sin \omega t$. It is rather easy to solve for the complex x in this case, and by taking the real part of the solution, one finds the answer for the $\cos \omega t$ driving force.

We assume a simple form for the particular solution

$$x_p = De^{i\omega t},$$

where D is a complex constant.

From Eq. (*17*) one inserts the form for x_p above to get

$$to$$

$$D\left\{-\omega^2 + 2i\beta\omega + \omega_0^2\right\} e^{i\omega t} = (F_0/m)e^{i\omega t},$$
$$D = \frac{F_0/m}{(\omega_0^2 - \omega^2) + 2i\beta\omega}.$$

The norm and phase for $D = |D|e^{-i\delta}$ can be read by inspection,

$$|D| = \frac{F_0/m}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}}, \quad \tan\delta = \frac{2\beta\omega}{\omega_0^2 - \omega^2}.$$

This is the same expression for δ as before. One then finds $x_p(t)$,

$$to$$

$$x_p(t) =$$
$$\Re\frac{(F_0/m)e^{i\omega t - i\delta}}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}}$$
$$=$$
$$\frac{(F_0/m)\cos(\omega t - \delta)}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}}.$$

=

$$=\Re\frac{(F_0/m)e^{i\omega t - i\delta}}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}}$$

$$\frac{(F_0/m)\cos(\omega t-\delta)}{\sqrt{(\omega_0^2-\omega^2)^2+4\beta^2\omega^2}}.$$

This is the same answer as before. If one wished to solve for the case where $F(t)=F_0\sin\omega t$, the imaginary part of the solution would work

$$to$$

$$x_p(t)=$$

$$\Im\frac{(F_0/m)e^{i\omega t-i\delta}}{\sqrt{(\omega_0^2-\omega^2)^2+4\beta^2\omega^2}}$$

$$=$$

$$\frac{(F_0/m)\sin(\omega t-\delta)}{\sqrt{(\omega_0^2-\omega^2)^2+4\beta^2\omega^2}}.$$

=

$$=\Im\frac{(F_0/m)e^{i\omega t-i\delta}}{\sqrt{(\omega_0^2-\omega^2)^2+4\beta^2\omega^2}}$$
$$\frac{(F_0/m)\sin(\omega t-\delta)}{\sqrt{(\omega_0^2-\omega^2)^2+4\beta^2\omega^2}}.$$

Consider the damped and driven harmonic oscillator worked out above. Given F_0, m, β and ω_0, solve for the complete solution $x(t)$ for the case where $F=F_0\sin\omega t$ with initial conditions $x(t=0)=0$ and $v(t=0)=0$. Assume the underdamped case.

The general solution including the arbitrary constants includes both the homogenous and particular solutions,

$$to$$

$$x(t)=$$

$$\frac{F_0}{m}\frac{\sin(\omega t-\delta)}{\sqrt{(\omega_0^2-\omega^2)^2+4\beta^2\omega^2}}+A\cos\omega' te^{-\beta t}+B\sin\omega' te^{-\beta t}.$$

$$=$$

$$\frac{F_0}{m}\frac{\sin(\omega t-\delta)}{\sqrt{(\omega_0^2-\omega^2)^2+4\beta^2\omega^2}}+A\cos\omega' te^{-\beta t}+B\sin\omega' te^{-\beta t}.$$

The quantities δ and ω' are given earlier in the section, $\omega' = \sqrt{\omega_0^2 - \beta^2}, \delta = \tan^{-1}(2\beta\omega/(\omega_0^2 - \omega^2)$. Here, solving the problem means finding the arbitrary constants A and B. Satisfying the initial conditions for the initial position and velocity:

$$to$$

$$x(t=0)=0=$$

$$-\eta\sin\delta+A,$$

$$v(t=0)=0=$$

$$\omega\eta\cos\delta-\beta A+\omega' B,$$

$$\eta\equiv$$

$$\frac{F_0}{m}\frac{1}{\sqrt{(\omega_0^2-\omega^2)^2+4\beta^2\omega^2}}.$$

$$=$$

$$=\!\!-\eta\sin\delta+A, v(t=0)=0$$

$$\omega\eta\cos\delta-\beta A+\omega' B,\eta$$

$$\frac{F_0}{m}\frac{1}{\sqrt{(\omega_0^2-\omega^2)^2+4\beta^2\omega^2}}.$$

The problem is now reduced to 2 equations and 2 unknowns, A and B. The solution is

$$to$$

$$A=$$

$$\eta\sin\delta,\quad B=\frac{-\omega\eta\cos\delta+\beta\eta\sin\delta}{\omega'}.(8.29)$$

=

$\eta\sin\delta, \quad B = \frac{-\omega\eta\cos\delta + \beta\eta\sin\delta}{\omega'}.$

8.3.6 Resonance Widths; the Q factor

From the previous two sections, the particular solution for a driving force, $F = F_0\cos\omega t$, is

to

$x_p(t) =$

$\frac{F_0/m}{\sqrt{(\omega_0^2-\omega^2)^2+4\omega^2\beta^2}}\cos(\omega_t-\delta),$

$\delta =$

$\tan^{-1}\left(\frac{2\beta\omega}{\omega_0^2-\omega^2}\right).$

=

$\frac{F_0/m}{\sqrt{(\omega_0^2-\omega^2)^2+4\omega^2\beta^2}}\cos(\omega_t-\delta), \delta$

=

$\tan^{-1}\left(\frac{2\beta\omega}{\omega_0^2-\omega^2}\right).$

If one fixes the driving frequency ω and adjusts the fundamental frequency $\omega_0 = \sqrt{k/m}$, the maximum amplitude occurs when $\omega_0 = \omega$ because that is when the term from the denominator $(\omega_0^2-\omega^2)^2+4\omega^2\beta^2$ is at a minimum. This is akin to dialing into a radio station. However, if one fixes ω_0 and adjusts the driving frequency one minimize with respect to ω, e.g. set

$$\frac{d}{d\omega}\left[(\omega_0^2-\omega^2)^2+4\omega^2\beta^2\right] = 0,$$

and one finds that the maximum amplitude occurs when $\omega = \sqrt{\omega_0^2 - 2\beta^2}$. If β is small relative to ω_0, one can simply state that the maximum amplitude is

$x_{\rm max} \approx \frac{F_0}{2m\beta\omega_0}$.

to

$\frac{4\omega^2\beta^2}{(\omega_0^2-\omega^2)^2+4\omega^2\beta^2} = \frac{1}{2}$.(8.30)

For small damping this occurs when $\omega = \omega_0 \pm \beta$, so the $FWHM \approx 2\beta$. For the purposes of tuning to a specific frequency, one wants the width to be as small as possible. The ratio of ω_0 to $FWHM$ is known as the {\it quality} factor, or Q factor,

$Q \equiv \frac{\omega_0}{2\beta}$.

8.3.7 Numerical Studies of Driven Oscillations

Solving the problem of driven oscillations numerically gives us much more flexibility to study different types of driving forces. We can reuse our earlier code by simply adding a driving force. If we stay in the x-direction only this can be easily done by adding a term $F_{\rm ext}(x,t)$. Note that we have kept it rather general here, allowing for both a spatial and a temporal dependence.

Before we dive into the code, we need to briefly remind ourselves about the equations we started with for the case with damping, namely

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx(t) = 0,$$

with no external force applied to the system.

Let us now for simplicty assume that our external force is given by

$$F_{\rm ext}(t) = F_0\cos{(\omega t)},$$

where F_0 is a constant (what is its dimension?) and ω is the frequency of the applied external driving force. **Small question:** would you expect energy to be conserved now?

Introducing the external force into our lovely differential equation and dividing by m and introducing $\omega_0^2 = \sqrt{k/m}$ we have

$$\frac{d^2x}{dt^2} + \frac{b}{m}\frac{dx}{dt} + \omega_0^2 x(t) = \frac{F_0}{m}\cos(\omega t),$$

Thereafter we introduce a dimensionless time $\tau = t\omega_0$ and a dimensionless frequency $\tilde{\omega} = \omega/\omega_0$. We have then

$$\frac{d^2x}{d\tau^2} + \frac{b}{m\omega_0}\frac{dx}{d\tau} + x(\tau) = \frac{F_0}{m\omega_0^2}\cos(\tilde{\omega}\tau),$$

Introducing a new amplitude $\tilde{F} = F_0/(m\omega_0^2)$ (check dimensionality again) we have

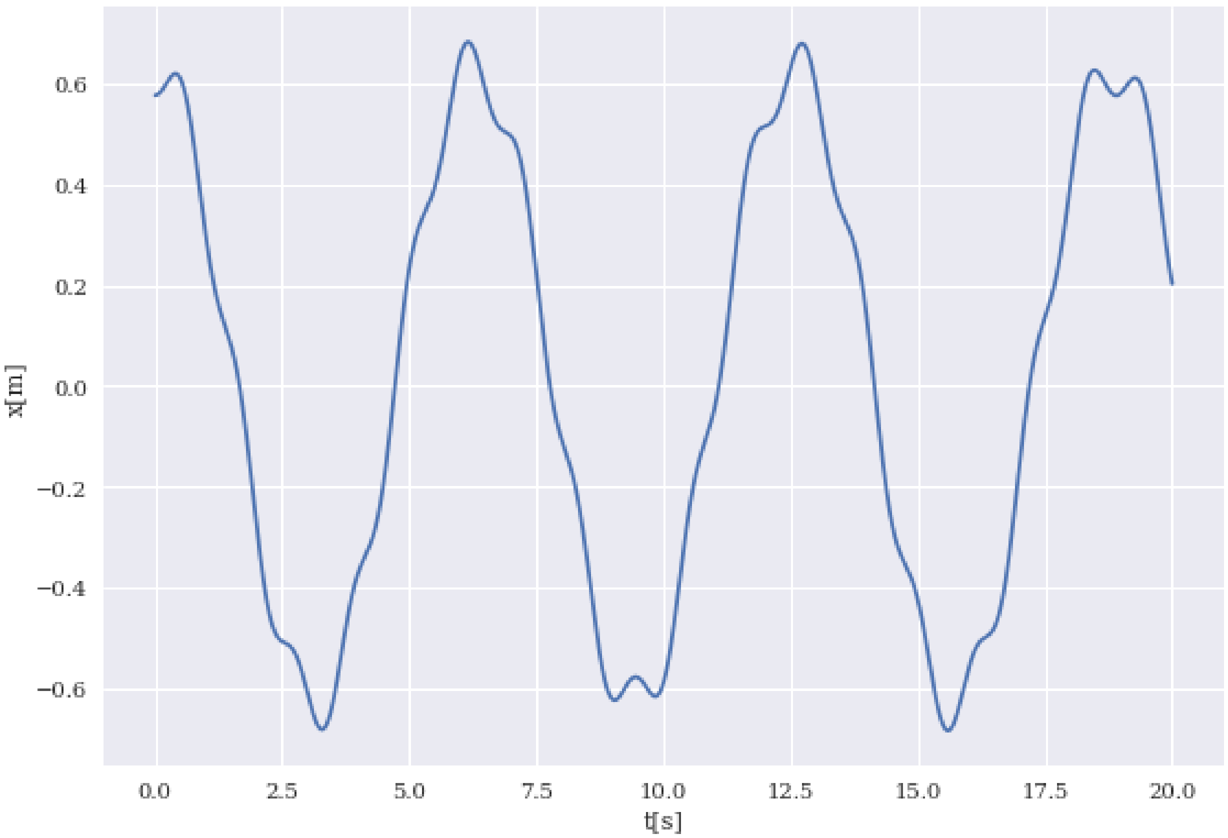
$$\frac{d^2x}{d\tau^2} + \frac{b}{m\omega_0}\frac{dx}{d\tau} + x(\tau) = \tilde{F}\cos(\tilde{\omega}\tau).$$

Our final step, as we did in the case of various types of damping, is to define $\gamma = b/(2m\omega_0)$ and rewrite our equations as

$$\frac{d^2x}{d\tau^2} + 2\gamma\frac{dx}{d\tau} + x(\tau) = \tilde{F}\cos(\tilde{\omega}\tau).$$

This is the equation we will code below using the Euler-Cromer method.

```
DeltaT = 0.001
#set up arrays
tfinal = 20 # in years
n = ceil(tfinal/DeltaT)
# set up arrays for t, v, and x
t = np.zeros(n)
v = np.zeros(n)
x = np.zeros(n)
# Initial conditions as one-dimensional arrays of time
x0 = sqrt(1./3.)
v0 = 0.0
x[0] = x0
v[0] = v0
gamma = 0.0
Omegatilde = 8./sqrt(2.)
Ftilde = 1.75
# Start integrating using Euler-Cromer's method
for i in range(n-1):
    # Set up the acceleration
    # Here you could have defined your own function for this
    a =  -2*gamma*v[i]-x[i]+Ftilde*cos(t[i]*Omegatilde)
    # update velocity, time and position
    v[i+1] = v[i] + DeltaT*a
    x[i+1] = x[i] + DeltaT*v[i+1]
    t[i+1] = t[i] + DeltaT
# Plot position as function of time
fig, ax = plt.subplots()
ax.set_ylabel('x[m]')
ax.set_xlabel('t[s]')
ax.plot(t, x)
fig.tight_layout()
save_fig("ForcedBlockEulerCromer")
plt.show()
```



In the above example we have focused on the Euler-Cromer method. This method has a local truncation error which is proportional to Δt^2 and thereby a global error which is proportional to Δt. We can improve this by using the Runge-Kutta family of methods. The widely popular Runge-Kutta to fourth order or just **RK4** has indeed a much better truncation error. The RK4 method has a global error which is proportional to Δt.

Let us revisit this method and see how we can implement it for the above example.

8.3.8 Differential Equations, Runge-Kutta methods

Runge-Kutta (RK) methods are based on Taylor expansion formulae, but yield in general better algorithms for solutions of an ordinary differential equation. The basic philosophy is that it provides an intermediate step in the computation of y_{i+1}.

To see this, consider first the following definitions

$$\frac{dy}{dt} = f(t, y),$$

and

$$y(t) = \int f(t,y)dt,$$

and

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} f(t,y)dt.$$

To demonstrate the philosophy behind RK methods, let us consider the second-order RK method, RK2. The first approximation consists in Taylor expanding $f(t,y)$ around the center of the integration interval t_i to t_{i+1}, that is, at $t_i + h/2$, h being the step. Using the midpoint formula for an integral, defining $y(t_i + h/2) = y_{i+1/2}$ and$t_i + h/2 = t_{i+1/2}$, we obtain

$$\int_{t_i}^{t_{i+1}} f(t,y)dt \approx hf(t_{i+1/2}, y_{i+1/2}) + O(h^3).$$

This means in turn that we have

$$y_{i+1} = y_i + hf(t_{i+1/2}, y_{i+1/2}) + O(h^3).$$

However, we do not know the value of $y_{i+1/2}$. Here comes thus the next approximation, namely, we use Euler's method to approximate $y_{i+1/2}$. We have then

$$y_{(i+1/2)} = y_i + \frac{h}{2}\frac{dy}{dt} = y(t_i) + \frac{h}{2}f(t_i, y_i).$$

This means that we can define the following algorithm for the second-order Runge-Kutta method, RK2.

60

<<<!!MATH_BLOCK

$k_2 = hf(t_{i+1/2}, y_i + k_1/2),$

with the final value

$y_{i+i} \approx y_i + k_2 + O(h^3).$

The difference between the previous one-step methods is that we now need an intermediate step in our evaluation, namely $t_i + h/2 = t_{(i+1/2)}$ where we evaluate the derivative f. This involves more operations, but the gain is a better stability in the solution.

The fourth-order Runge-Kutta, RK4, has the following algorithm

63

<<<!!MATH_BLOCK

$$k_3 = hf(t_i + h/2, y_i + k_2/2) \qquad k_4 = hf(t_i + h, y_i + k_3)$$

with the final result

$$y_{i+1} = y_i + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right).$$

Thus, the algorithm consists in first calculating k_1 with t_i, y_1 and f as inputs. Thereafter, we increase the step size by $h/2$ and calculate k_2, then k_3 and finally k_4. The global error goes as $O(h^4)$.

However, at this stage, if we keep adding different methods in our main program, the code will quickly become messy and ugly. Before we proceed thus, we will now introduce functions that enbody the various methods for solving differential equations. This means that we can separate out these methods in own functions and files (and later as classes and more generic functions) and simply call them when needed. Similarly, we could easily encapsulate various forces or other quantities of interest in terms of functions. To see this, let us bring up the code we developed above for the simple sliding block, but now only with the simple forward Euler method. We introduce two functions, one for the simple Euler method and one for the force.

Note that here the forward Euler method does not know the specific force function to be called. It receives just an input the name. We can easily change the force by adding another function.

```python
def ForwardEuler(v,x,t,n,Force):
    for i in range(n-1):
        v[i+1] = v[i] + DeltaT*Force(v[i],x[i],t[i])
        x[i+1] = x[i] + DeltaT*v[i]
        t[i+1] = t[i] + DeltaT
```

```python
def SpringForce(v,x,t):
#   note here that we have divided by mass and we return the acceleration
    return  -2*gamma*v-x+Ftilde*cos(t*Omegatilde)
```

It is easy to add a new method like the Euler-Cromer

```
def ForwardEulerCromer(v,x,t,n,Force):
    for i in range(n-1):
        a = Force(v[i],x[i],t[i])
        v[i+1] = v[i] + DeltaT*a
        x[i+1] = x[i] + DeltaT*v[i+1]
        t[i+1] = t[i] + DeltaT
```

and the Velocity Verlet method (be careful with time-dependence here, it is not an ideal method for non-conservative forces))

```
def VelocityVerlet(v,x,t,n,Force):
    for i in range(n-1):
        a = Force(v[i],x[i],t[i])
        x[i+1] = x[i] + DeltaT*v[i]+0.5*a
        anew = Force(v[i],x[i+1],t[i+1])
        v[i+1] = v[i] + 0.5*DeltaT*(a+anew)
        t[i+1] = t[i] + DeltaT
```

Finally, we can now add the Runge-Kutta2 method via a new function

```
def RK2(v,x,t,n,Force):
    for i in range(n-1):
# Setting up k1
        k1x = DeltaT*v[i]
        k1v = DeltaT*Force(v[i],x[i],t[i])
# Setting up k2
        vv = v[i]+k1v*0.5
        xx = x[i]+k1x*0.5
        k2x = DeltaT*vv
        k2v = DeltaT*Force(vv,xx,t[i]+DeltaT*0.5)
# Final result
        x[i+1] = x[i]+k2x
        v[i+1] = v[i]+k2v
        t[i+1] = t[i]+DeltaT
```

```
  File "<ipython-input-7-ffedbda27704>", line 14
    t[i+1] = t[i]+DeltaT
                        ^
TabError: inconsistent use of tabs and spaces in indentation
```

Finally, we can now add the Runge-Kutta2 method via a new function

```
def RK4(v,x,t,n,Force):
    for i in range(n-1):
# Setting up k1
        k1x = DeltaT*v[i]
        k1v = DeltaT*Force(v[i],x[i],t[i])
# Setting up k2
        vv = v[i]+k1v*0.5
        xx = x[i]+k1x*0.5
        k2x = DeltaT*vv
        k2v = DeltaT*Force(vv,xx,t[i]+DeltaT*0.5)
# Setting up k3
        vv = v[i]+k2v*0.5
        xx = x[i]+k2x*0.5
        k3x = DeltaT*vv
        k3v = DeltaT*Force(vv,xx,t[i]+DeltaT*0.5)
```

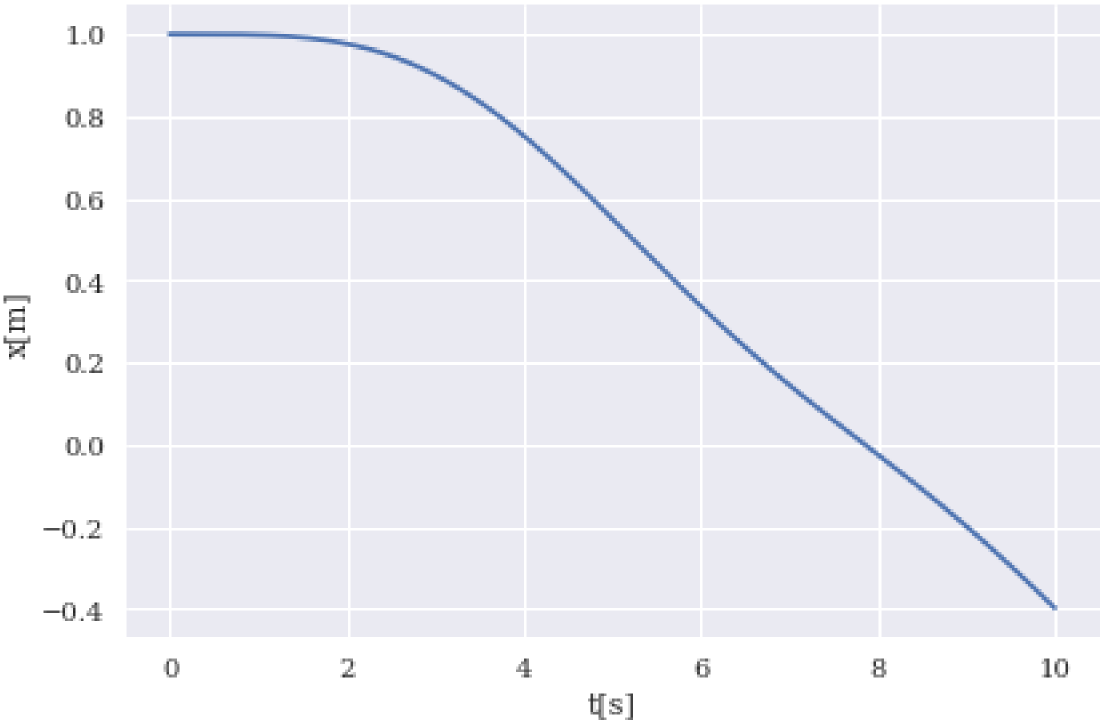
(continues on next page)

(continued from previous page)

```
# Setting up k4
        vv = v[i]+k3v
        xx = x[i]+k3x
        k4x = DeltaT*vv
        k4v = DeltaT*Force(vv,xx,t[i]+DeltaT)
# Final result
        x[i+1] = x[i]+(k1x+2*k2x+2*k3x+k4x)/6.
        v[i+1] = v[i]+(k1v+2*k2v+2*k3v+k4v)/6.
        t[i+1] = t[i] + DeltaT
```

The Runge-Kutta family of methods are particularly useful when we have a time-dependent acceleration. If we have forces which depend only the spatial degrees of freedom (no velocity and/or time-dependence), then energy conserving methods like the Velocity Verlet or the Euler-Cromer method are preferred. As soon as we introduce an explicit time-dependence and/or add dissipitave forces like friction or air resistance, then methods like the family of Runge-Kutta methods are well suited for this. The code below uses the Runge-Kutta4 methods.

```
DeltaT = 0.001
#set up arrays
tfinal = 10 # in years
n = ceil(tfinal/DeltaT)
# set up arrays for t, v, and x
t = np.zeros(n)
v = np.zeros(n)
x = np.zeros(n)
# Initial conditions (can change to more than one dim)
x0 = 1.0
v0 = 0.0
x[0] = x0
v[0] = v0
gamma = 0.0
Omegatilde = 0.2
Ftilde = 1.0

# Start integrating using Euler's method
# Note that we define the force function as a SpringForce
RK4(v,x,t,n,SpringForce)

# Plot position as function of time
fig, ax = plt.subplots()
ax.set_ylabel('x[m]')
ax.set_xlabel('t[s]')
ax.plot(t, x)
fig.tight_layout()
save_fig("ForcedBlockRK4")
plt.show()
```



8.3.9 Principle of Superposition and Periodic Forces (Fourier Transforms)

If one has several driving forces, $F(t) = \sum_n F_n(t)$, one can find the particular solution to each F_n, $x_{pn}(t)$, and the particular solution for the entire driving force is

$$x_p(t) = \sum_n x_{pn}(t).$$

This is known as the principal of superposition. It only applies when the homogenous equation is linear. If there were an anharmonic term such as x^3 in the homogenous equation, then when one summed various solutions, $x = (\sum_n x_n)^2$, one would get cross terms. Superposition is especially useful when $F(t)$ can be written as a sum of sinusoidal terms, because the solutions for each sinusoidal (sine or cosine) term is analytic, as we saw above.

Driving forces are often periodic, even when they are not sinusoidal. Periodicity implies that for some time τ

to

$F(t+\tau) = F(t)$.(8.41)

One example of a non-sinusoidal periodic force is a square wave. Many components in electric circuits are non-linear, e.g. diodes, which makes many wave forms non-sinusoidal even when the circuits are being driven by purely sinusoidal sources.

The code here shows a typical example of such a square wave generated using the functionality included in the **scipy** Python package. We have used a period of $\tau = 0.2$.

```
import numpy as np
import math
from scipy import signal
import matplotlib.pyplot as plt

# number of points
n = 500
# start and final times
t0 = 0.0
tn = 1.0
# Period
t = np.linspace(t0, tn, n, endpoint=False)
SqrSignal = np.zeros(n)
SqrSignal = 1.0+signal.square(2*np.pi*5*t)
plt.plot(t, SqrSignal)
plt.ylim(-0.5, 2.5)
plt.show()
```

For the sinusoidal example studied in the previous subsections the period is $\tau = 2\pi/\omega$. However, higher harmonics can also satisfy the periodicity requirement. In general, any force that satisfies the periodicity requirement can be expressed as a sum over harmonics,

$$F(t) = \frac{f_0}{2} + \sum_{n>0} f_n \cos(2n\pi t/\tau) + g_n \sin(2n\pi t/\tau).$$

From the previous subsection, one can write down the answer for $x_{pn}(t)$, by substituting f_n/m or g_n/m for F_0/m into Eq.s (*20*) or (*21*) respectively. By writing each factor $2n\pi t/\tau$ as $n\omega t$, with $\omega \equiv 2\pi/\tau$,

$$F(t) = \frac{f_0}{2} + \sum_{n>0} f_n \cos(n\omega t) + g_n \sin(n\omega t).$$

The solutions for $x(t)$ then come from replacing ω with $n\omega$ for each term in the particular solution in Equations (*13*) and (*16*),

$$to$$

$$x_p(t) =$$

$$\frac{f_0}{2k} + \sum_{n>0} \alpha_n \cos(n\omega t - \delta_n) + \beta_n \sin(n\omega t - \delta_n),$$

$$\alpha_n =$$

$$\frac{f_n/m}{\sqrt{((n\omega)^2 - \omega_0^2) + 4\beta^2 n^2 \omega^2}},$$

$$\beta_n =$$

$$\frac{g_n/m}{\sqrt{((n\omega)^2 - \omega_0^2) + 4\beta^2 n^2 \omega^2}},$$

$$\delta_n =$$

$$\tan^{-1}\left(\frac{2\beta n\omega}{\omega_0^2 - n^2\omega^2}\right).$$

=

$$\frac{f_0}{2k} + \sum_{n>0} \alpha_n = \cos(n\omega t - \delta_n) + \beta_n \sin(n\omega t - \delta_n), \alpha_n$$

$$\frac{f_n/m}{\sqrt{((n\omega)^2 - \omega_0^2) + 4\beta^2 n^2 \omega^2}}, \beta_n$$

$$\frac{g_n/m}{\sqrt{((n\omega)^2 - \omega_0^2) + 4\beta^2 n^2 \omega^2}}, \delta_n$$

$$\tan^{-1}\left(\frac{2\beta n\omega}{\omega_0^2 - n^2\omega^2}\right).$$

Because the forces have been applied for a long time, any non-zero damping eliminates the homogenous parts of the solution, so one need only consider the particular solution for each n.

The problem will considered solved if one can find expressions for the coefficients f_n and g_n, even though the solutions are expressed as an infinite sum. The coefficients can be extracted from the function $F(t)$ by

$$to$$

$$f_n =$$

$$\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt\, F(t)\cos(2n\pi t/\tau),$$

$$g_n =$$

$$\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt\, F(t)\sin(2n\pi t/\tau).$$

=

$$=\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt\, F(t)\cos(2n\pi t/\tau), g_n$$

$$\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt\, F(t)\sin(2n\pi t/\tau).$$

To check the consistency of these expressions and to verify Eq. (*37*), one can insert the expansion of $F(t)$ in Eq. (*36*) into the expression for the coefficients in Eq. (*37*) and see whether

$$to$$

$$f_n =?$$

$$\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt\, \left\{\frac{f_0}{2} + \sum_{m>0} f_m\cos(m\omega t) + g_m\sin(m\omega t)\right\}\cos(n\omega t).(8.42)$$

=?

$$\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt\, \left\{\frac{f_0}{2} + \sum_{m>0} f_m\cos(m\omega t) + g_m\sin(m\omega t)\right\}\cos(n\omega t).$$

Immediately, one can throw away all the terms with g_m because they convolute an even and an odd function. The term with $f_0/2$ disappears because $\cos(n\omega t)$ is equally positive and negative over the interval and will integrate to zero. For all the terms $f_m \cos(m\omega t)$ appearing in the sum, one can use angle addition formulas to see that $\cos(m\omega t)\cos(n\omega t) = (1/2)(\cos[(m+n)\omega t] + \cos[(m-n)\omega t]$. This will integrate to zero unless $m = n$. In that case the $m = n$ term gives

$$\int_{-\tau/2}^{\tau/2} dt\ \cos^2(m\omega t) = \frac{\tau}{2},$$

and

$$to$$

$$f_n =?$$

$$\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt\ f_n/2$$

$$=$$

$$f_n \checkmark.$$

=?

$$=\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt\ f_n/2$$

$$f_n \checkmark.$$

The same method can be used to check for the consistency of g_n.

Consider the driving force:

$$F(t) = At/\tau,\ \ -\tau/2 < t < \tau/2,\quad F(t+\tau) = F(t).$$

Find the Fourier coefficients f_n and g_n for all n using Eq. (*37*).

Only the odd coefficients enter by symmetry, i.e. $f_n = 0$. One can find g_n integrating by parts,

$$to$$

$$g_n =$$

$$\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt\ \sin(n\omega t)\frac{At}{\tau}$$

$$u =$$

$$t,\ dv = \sin(n\omega t)dt,\ v = -\cos(n\omega t)/(n\omega),$$

$$g_n =$$

$$\frac{-2A}{n\omega\tau^2}\int_{-\tau/2}^{\tau/2} dt\ \cos(n\omega t) + 2A\frac{-t\cos(n\omega t)}{n\omega\tau^2}\Big|_{-\tau/2}^{\tau/2}.$$

=

$$\frac{2}{\tau}\int_{-\tau/2}^{\tau/2} dt{=}\sin(n\omega t)\frac{At}{\tau}u$$

$$t,\ dv = \sin{=}(n\omega t)dt,\ v = -\cos(n\omega t)/(n\omega), g_n$$

$$\frac{-2A}{n\omega\tau^2}\int_{-\tau/2}^{\tau/2} dt\ \cos(n\omega t) + 2A\frac{-t\cos(n\omega t)}{n\omega\tau^2}\Big|_{-\tau/2}^{\tau/2}.$$

The first term is zero because $\cos(n\omega t)$ will be equally positive and negative over the interval. Using the fact that $\omega\tau = 2\pi$,

$$to$$

$$g_n =$$

$$-\frac{2A}{2n\pi}\cos(n\omega\tau/2)$$

$$=$$

$$-\frac{A}{n\pi}\cos(n\pi)$$

$$=$$

$$\frac{A}{n\pi}(-1)^{n+1}.$$

=

$$=-\frac{2A}{2n\pi}\cos(n\omega\tau/2)$$

$$=-\frac{A}{n\pi}\cos(n\pi)$$
$$\frac{A}{n\pi}(-1)^{n+1}.$$

8.3.10 Fourier Series

More text will come here, chpater 5.7-5.8 of Taylor are discussed during the lectures. The code here uses the Fourier series discussed in chapter 5.7 for a square wave signal. The equations for the coefficients are are discussed in Taylor section 5.7, see Example 5.4. The code here visualizes the various approximations given by Fourier series compared with a square wave with period $T = 0.2$, witth 0.1 and max value $F = 2$. We see that when we increase the number of components in the Fourier series, the Fourier series approximation gets closes and closes to the square wave signal.

```
import numpy as np
import math
from scipy import signal
import matplotlib.pyplot as plt

# number of points                                                                      ␣
↪
n = 500
# start and final times                                                                 ␣
↪
t0 = 0.0
tn = 1.0
# Period                                                                                ␣
↪
T =0.2
# Max value of square signal                                                            ␣
↪
Fmax= 2.0
# Width of signal                                                                       ␣
↪
Width = 0.1
t = np.linspace(t0, tn, n, endpoint=False)
SqrSignal = np.zeros(n)
FourierSeriesSignal = np.zeros(n)
SqrSignal = 1.0+signal.square(2*np.pi*5*t+np.pi*Width/T)
a0 = Fmax*Width/T
FourierSeriesSignal = a0
Factor = 2.0*Fmax/np.pi
for i in range(1,500):
    FourierSeriesSignal += Factor/(i)*np.sin(np.pi*i*Width/T)*np.cos(i*t*2*np.pi/T)
plt.plot(t, SqrSignal)
plt.plot(t, FourierSeriesSignal)
plt.ylim(-0.5, 2.5)
plt.show()
```

8.3.11 Solving differential equations with Fouries series

The material here was discussed during the lecture of February 19 and 21. It is also covered by Taylor in section 5.8.

8.3.12 Response to Transient Force

Consider a particle at rest in the bottom of an underdamped harmonic oscillator, that then feels a sudden impulse, or change in momentum, $I = F\Delta t$ at $t = 0$. This increases the velocity immediately by an amount $v_0 = I/m$ while not changing the position. One can then solve the trajectory by solving Eq. (*9*) with initial conditions $v_0 = I/m$ and $x_0 = 0$. This gives

$$x(t) = \frac{I}{m\omega'}e^{-\beta t}\sin\omega' t, \;\; t > 0.$$

Here, $\omega' = \sqrt{\omega_0^2 - \beta^2}$. For an impulse I_i that occurs at time t_i the trajectory would be

$$x(t) = \frac{I_i}{m\omega'}e^{-\beta(t-t_i)}\sin[\omega'(t-t_i)]\Theta(t-t_i),$$

where $\Theta(t-t_i)$ is a step function, i.e. $\Theta(x)$ is zero for $x < 0$ and unity for $x > 0$. If there were several impulses linear superposition tells us that we can sum over each contribution,

$$x(t) = \sum_i \frac{I_i}{m\omega'}e^{-\beta(t-t_i)}\sin[\omega'(t-t_i)]\Theta(t-t_i)$$

Now one can consider a series of impulses at times separated by Δt, where each impulse is given by $F_i\Delta t$. The sum above now becomes an integral,

$$to$$

$$x(t) =$$

$$\int_{-\infty}^{\infty} dt'\ F(t')\frac{e^{-\beta(t-t')}\sin[\omega'(t-t')]}{m\omega'}\Theta(t-t')$$

$$=$$

$$\int_{-\infty}^{\infty} dt'\ F(t')G(t-t'),$$

$$G(\Delta t) =$$

$$\frac{e^{-\beta\Delta t}\sin[\omega'\Delta t]}{m\omega'}\Theta(\Delta t)$$

=

$$\int_{-\infty}^{\infty} dt'=F(t')\frac{e^{-\beta(t-t')}\sin[\omega'(t-t')]}{m\omega'}\Theta(t-t')$$

$$\int_{-\infty}^{\infty} dt'=F(t')G(t-t'), G(\Delta t)$$

$$\frac{e^{-\beta\Delta t}\sin[\omega'\Delta t]}{m\omega'}\Theta(\Delta t)$$

The quantity $e^{-\beta(t-t')}\sin[\omega'(t-t')]/m\omega'\Theta(t-t')$ is called a Green's function, $G(t-t')$. It describes the response at t due to a force applied at a time t', and is a function of $t-t'$. The step function ensures that the response does not occur before the force is applied. One should remember that the form for G would change if the oscillator were either critically- or over-damped.

When performing the integral in Eq. (*44*) one can use angle addition formulas to factor out the part with the t' dependence in the integrand,

to

$$x(t) = \frac{1}{m\omega'}e^{-\beta t}\left[I_c(t)\sin(\omega' t) - I_s(t)\cos(\omega' t)\right],$$

$$I_c(t) \equiv \int_{-\infty}^{t} dt'\ F(t')e^{\beta t'}\cos(\omega' t'),$$

$$I_s(t) \equiv \int_{-\infty}^{t} dt'\ F(t')e^{\beta t'}\sin(\omega' t').$$

=

$$\frac{1}{m\omega'}e^{-\beta t}\left[I_c(t)\sin(\omega' t) - I_s(t)\cos(\omega' t)\right], I_c(t) \int_{-\infty}^{t} dt'\ F(t')e^{\beta t'}\cos(\omega' t'), I_s(t) \int_{-\infty}^{t} dt'\ F(t')e^{\beta t'}\sin(\omega' t').$$

If the time t is beyond any time at which the force acts, $F(t' > t) = 0$, the coefficients I_c and I_s become independent of t.

Consider an undamped oscillator ($\beta \to 0$), with characteristic frequency ω_0 and mass m, that is at rest until it feels a force described by a Gaussian form,

to

$$F(t) = F_0 \exp\left\{\frac{-t^2}{2\tau^2}\right\}.$$

=

$$F_0 \exp\left\{\frac{-t^2}{2\tau^2}\right\}.$$

For large times ($t >> \tau$), where the force has died off, find $x(t)$.\ Solve for the coefficients I_c and I_s in Eq. (*45*). Because the Gaussian is an even function, $I_s = 0$, and one need only solve for I_c,

$$to$$

$$I_c =$$

$$F_0 \int_{-\infty}^{\infty} dt'\; e^{-t'^2/(2\tau^2)} \cos(\omega_0 t')$$

$$=$$

$$\Re F_0 \int_{-\infty}^{\infty} dt'\; e^{-t'^2/(2\tau^2)} e^{i\omega_0 t'}$$

$$=$$

$$\Re F_0 \int_{-\infty}^{\infty} dt'\; e^{-(t'-i\omega_0\tau^2)^2/(2\tau^2)} e^{-\omega_0^2\tau^2/2}$$

$$=$$

$$F_0 \tau \sqrt{2\pi} e^{-\omega_0^2\tau^2/2}.$$

=

$$F_0 \int_{-\infty}^{\infty} dt'\; e^{-t'^2/(2\tau^2)} \cos(\omega_0 t')$$

$$\Re F_0 \int_{-\infty}^{\infty} dt'\; e^{-t'^2/(2\tau^2)} e^{i\omega_0 t'}$$

$$\Re F_0 \int_{-\infty}^{\infty} dt'\; e^{-(t'-i\omega_0\tau^2)^2/(2\tau^2)} e^{-\omega_0^2\tau^2/2}$$

$$F_0 \tau \sqrt{2\pi} e^{-\omega_0^2\tau^2/2}.$$

The third step involved completing the square, and the final step used the fact that the integral

$$to$$

$$\int_{-\infty}^{\infty} dx\; e^{-x^2/2} =$$

$$\sqrt{2\pi}.$$

=

$\sqrt{2\pi}.$

To see that this integral is true, consider the square of the integral, which you can change to polar coordinates,

$$to$$

$$I =$$

$$\int_{-\infty}^{\infty} dx\; e^{-x^2/2}$$

$$I^2 =$$

$$\int_{-\infty}^{\infty} dxdy\; e^{-(x^2+y^2)/2}$$

$$=$$

$$2\pi \int_{0}^{\infty} rdr\; e^{-r^2/2}$$

$$=$$

$$2\pi.$$

=

=$\int_{-\infty}^{\infty} dx\; e^{-x^2/2} I^2$

=$\int_{-\infty}^{\infty} dxdy\; e^{-(x^2+y^2)/2}$

=$2\pi \int_{0}^{\infty} rdr\; e^{-r^2/2}$

$2\pi.$

Finally, the expression for x from Eq. (*45*) is

$$to$$

$$x(t >> \tau) =$$

$$\frac{F_0\tau}{m\omega_0}\sqrt{2\pi}e^{-\omega_0^2\tau^2/2}\sin(\omega_0 t).$$

=

$$\frac{F_0\tau}{m\omega_0}\sqrt{2\pi}e^{-\omega_0^2\tau^2/2}\sin(\omega_0 t).$$

8.3.13 The classical pendulum and scaling the equations

Let us end our discussion of oscillations with another classical case, the pendulum.

The angular equation of motion of the pendulum is given by Newton's equation and with no external force it reads

$$ml\frac{d^2\theta}{dt^2} + mgsin(\theta) = 0,$$

with an angular velocity and acceleration given by

$$v = l\frac{d\theta}{dt},$$

and

$$a = l\frac{d^2\theta}{dt^2}.$$

We do however expect that the motion will gradually come to an end due a viscous drag torque acting on the pendulum. In the presence of the drag, the above equation becomes

$$ml\frac{d^2\theta}{dt^2} + \nu\frac{d\theta}{dt} + mgsin(\theta) = 0,$$

where ν is now a positive constant parameterizing the viscosity of the medium in question. In order to maintain the motion against viscosity, it is necessary to add some external driving force. We choose here a periodic driving force. The last equation becomes then

$$ml\frac{d^2\theta}{dt^2} + \nu\frac{d\theta}{dt} + mgsin(\theta) = Asin(\omega t),$$

with A and ω two constants representing the amplitude and the angular frequency respectively. The latter is called the driving frequency.

We define

$$\omega_0 = \sqrt{g/l},$$

the so-called natural frequency and the new dimensionless quantities

$$\hat{t} = \omega_0 t,$$

with the dimensionless driving frequency

$$\hat{\omega} = \frac{\omega}{\omega_0},$$

and introducing the quantity Q, called the *quality factor*,

$$Q = \frac{mg}{\omega_0 \nu},$$

and the dimensionless amplitude

$$\hat{A} = \frac{A}{mg}$$

8.3.14 More on the Pendulum

We have

$$\frac{d^2\theta}{d\hat{t}^2} + \frac{1}{Q}\frac{d\theta}{d\hat{t}} + sin(\theta) = \hat{A}cos(\hat{\omega}\hat{t}).$$

This equation can in turn be recast in terms of two coupled first-order differential equations as follows

$$\frac{d\theta}{d\hat{t}} = \hat{v},$$

and

$$\frac{d\hat{v}}{d\hat{t}} = -\frac{\hat{v}}{Q} - sin(\theta) + \hat{A}cos(\hat{\omega}\hat{t}).$$

These are the equations to be solved. The factor Q represents the number of oscillations of the undriven system that must occur before its energy is significantly reduced due to the viscous drag. The amplitude $\hat{A}$ is measured in units of the maximum possible gravitational torque while $\hat{\omega}$ is the angular frequency of the external torque measured in units of the pendulum's natural frequency.