**Phys 265 - Lab 1b: MineCrafting** [13-mar-2024]

*Download and edit lab1b_lastname.py, which is posted on ELMS and in the* `https://github.com/astroumd/PHYS265-spring24` *repo. Submit three items to your github repo:*

1 Your completed version of `lab1b_lastname.py` (re-named appropriately)

2 A screenshot of your desktop after running your code.

3 A two-page summary (in PDF) of your final submission. Your narrative can follow the one given in Part 1 to Part 6 below. You should add your figures as well, but those are not part of the two pages of text.

Make sure your instructors have read access to your github repo name!

| `'Check for any updates to this document'` |
|---|

## Introduction

You are a scientist for a mining company that operates a vertical mine at the earth's equator. This is one of the deepest mines on earth; it's roughly 4 km to the bottom of the shaft. Your boss proposes to measure the vertical depth of the shaft by dropping a test mass and measuring the time to hit the bottom. Compute the time under simple assumptions of no-drag and uniform gravity (you can compute this on paper).
*Assume we don't need to wait for the sound to hit the top of the shaft, imagine an electrical signal that will instantaneously report when the test mass hits the bottom. For reference, the sound speed in normal air is about 343 m/s, so it would take a while to reach the top of the shaft.*

You want to impress your boss, so you start to take into account a few small effects that will make a difference in the result. First, you add a linear air drag force. How much of a difference does the drag force make?

Now make a small correction for the fact that gravity weakens as you go down the shaft. Here is where the density profile of the earth starts to impact the result. How much of a difference does this make? Let's first assume the density is constant. We will cover a non-homogenous earth later on.

Now compute the east-west deflection due to the Coriolis force. Since the mine shaft has a width of say 5 meters, this sets a limit on whether it's practical to drop a test mass the entire way. Report back to your boss and tell him or her if the idea is any good, or should we move to mining on the poles!

On the moon there is no significant air drag. How would this project work on the moon, assuming we drill from the moon's north pole to south pole and drop the test mass, so no Coriolis forces are at play. The moon actually rotates at the same rate as it revolves around the earth, so any Coriolis forces would be extremely small and neglible.

## All the way down?

Since 4 km is not challenging enough, imagine digging that hole all the way through the center of the earth to the other side. And now drop an object from rest down towards the center. What will happen, assuming there is only gravity acting on the object?

Assuming the earth has a constant density, what is the path of the object (you will plot radius and velocity as function of time), and how long will it take for the object to reach the other side (the crossing time). You can give the answer in real earth SI units, or you can normalize it with respect to an earth with unit radius, unit mass, and G=1. See also the section on **units** in this lab. In this case you should ignore Coriolis forces, or just drill the hole from the north to the south pole. How does this compare to the

orbital period, where you assume the object is on a circular orbit, in centrifugal balance with the attractive force of the earth:

$$\frac{V^2}{R} = \frac{GM}{R^2} \tag{1}$$

gives the orbital speed $V$ for given earth radius $R$ and earth mass $M$.

Now assume that the density is not constant. From geology we know that the density increases towards the center. Near the surface it is about 2-3 g/cm3, near the center more like 13. So lets assume something simple like

$$\rho = \rho_n (1 - r^2)^n \tag{2}$$

where $n$ is some exponent, for example take $n = 0$ to confirm your constant density earth can be reproduced. Now take a more centrally concentrated earth, for example $n = 2$ gets pretty close to how earth in fact is. We have a figure in our github repo showing the actual earth density profile.

Note you will need to normalize the density $\rho_n$ for each case $n > 0$, because the orbital period should not change because the total mass needs to be

$$M = \frac{4}{3}\pi\rho_0 R^3 \qquad (n = 0) \tag{3}$$

you can also use the `quad` integrator for this, though there is analytical solution as well, and for small $n$ relatively easily done with some algebra. Again, it's good practice to check your code this way, so you can feel comfortable using $n = 100$ or more generic density profiles.

What about a Hollow Earth? This was a popular idea some time ago. Jules Verne was apparently about only 87 miles below the surface. How does a hollow earth affect the falling object through the center of the earth?

### Coriolis force

Since the earth is rotating, there is a **Coriolis force** on the test particle as it picks up speed.

The effect of the Coriolis force becomes enormous as the test particle falls. For example, suppose the 'mine shaft' is on the earth's equator and pointed towards the earth's center. Well, the equator rotates at about half a kilometer per second. So once the test particle has fallen for several minutes, the mine shaft has to be extremely wide or else the test particle bumps into the wall.

If the mine shaft is on the north/south pole, then there is no Coriolis force. So in this case the original notion of computing the complete oscillation time works great, and there is no complication from a Coriolis force.

## Formulas and Data

Coriolis force on earth:

$$F_c = -2\,\vec{\Omega} \times \vec{v} \tag{4}$$

where $\Omega$ is the earth rotation rate (1.1547e-5 $s^{-1}$) for a vector along $\hat{z}$. This implies

$$F_{cx} = -2\Omega v_y \tag{5}$$
$$F_{cy} = +2\Omega v_x \tag{6}$$

Some examples of Rotation and Revolution (listen to Monty Python's **Galaxy Song**):

- our sun revolves clock wise around the galactic center
  speed around 230 km/s

- our earth revolves counter clock wise around the sun
  speed around 30 km/s

- our moon revolves counter clock wise around the earth
  moon shadow on earth 0.72 km/s (1600 mph), or about twice the speed of sound

- our earth rotates counter clock wise (mathematically $\Omega > 0$)
  0.463 km/s at the equator

```
Gravitational Constant = 6.6743e-11 m3 / (kg s2)

Earth:  mass   =       5.972e+24 kg
        radius =       6378.1    km   (ignore flattening)

Moon:   mass   =       7.35e+22  kg
        radius =       1738.1    km   (ignore flattening)

Kola Superdeep Borehole            =  12.3 km  (now closed)
AngloGold Ashanti Mponeng gold mine = 3.9 km (still going deeper)
```

### Units

There are a few common ways to deal with units (in Python):

1 Use the actual SI (or any) units. Be aware in astronomy and particle physics values are often rather extreme large/small.

2 Use dimensionless units, and then scale to the real units afterwards. You should always use a dimensional analysis as a double check, it's very easy to make a scaling mistake here.

3 Use something like `astropy.units`, but not all python code knows how to process such numpy arrays. If we use the familiar `np.sum()` or **Euler** method, it will correctly work with the Astropy Units, but functions like `quad()` or `solve_ivp()` will not work. Thus method 1. or 2. is sadly to be preferred. For good measure he's an contrived example to retrieve the earth mass:

```
from astropy.constants import G, M_earth, R_earth

from astropy import units as u
r=np.linspace(0,6400,101) * u.km
rho0 = M_earth / (4*np.pi*R_earth**3/3) # circular reasoning to use M_earth
    here
dr = r[1]-r[0]
s = 4*np.pi*rho0*r*r
m = dr*(s[1:].sum()+s[:-1].sum())/2   # this is now in units km^3.kg/m^3
m = m.to("kg")                        # unify, to get to kg

print(m)
6.0341996 x 10^24 kg                  # 5.972e+24 is the correct answer

print(m.value)
6.034199580563992e+24
```

## Code Files

Download from ELMS or github the skeleton file `lab1b_lastname.py`. This is the file you will edit and eventually submit back to us, preferably github.

The skeleton code contains a few helper functions that are documented as such. Be sure to read their documentation. The code then guides you through a series of figures, with things to explore which you can add to your report. Pay particular attention to the accuracy.

### Part 1

Similar to our earlier Pisa tower experiments, here we explore our local constant gravity earth and confirm the code gives the same answer as the theoretical value. We use a `zero` level finder utility that you will find in the code. We did not discuss the `events=` keyword for `solve_ivp()`, which could also be used to solve for the time hitting the bottom of the shaft.

Figure 1 will show the height of the object above the bottom of the mine shaft as function of time. Also draw the black zero level line so we visually can see when it hits rock bottom. Pick a time just a bit past the splashdown so the plot looks like. Be sure in all figures to label your axes and put a descriptive title on your plot.

### Part 2

We now modify gravity so it depends on the depth in the mine shaft. Does gravity get stronger or weaker as you go down? What effect does this have on the time reaching the bottom of the mine shaft. The code will happily plunge right through the bottom of the pit, but using the `zero` level finder tool we can find when this happens.

In this part it will be useful to consider how you define the position. It should be the earth radius at t=0, falling down, but for plotting purposes it will be useful to plot the height above the bottom of the shaft.

Add drag to your equation to see the effect how delayed it will arrive. How can we calibrate the drag coefficient?

Figure 2, like Figure 1, will show the path for zero drag and one or two values of the drag, all in one diagram. Add the drag value to the legend.

### Part 3

We consider a mine shaft on the equator and see what the effect of the Coriolis force is.

Figure 3 should show a plot the path of the object in X and Y, where X is along the length of mine shaft, and Y along the width of the mine shaft. You can plot a number of equally time spaced points along the curve.

If need be, for the X and Y limits to show just around the shaft. Your plot will then not have equal scales in X (4 km) and Y (5 m).

### Part 4

We consider a mine shaft from pole to pole, so Coriolis force can be neglected. We first use a constant density earth.

Figure 4: plot position and velocity as function of time. Do the curves look familiar? At what time does the object reach the center of the earth, and at what speed? Use the `zero` level function and `np.max()`.

Here it might be more easier to use dimensionless units and scale to earth units later on.

### Part 5

Continue on using different density profiles from eq. (2). Remember to properly scale $\rho_n$ for different values of $n$ so we use the same total mass of the earth. How does the density profile affect the travel time?

Figures 5, 6 and 7 should plot the density profile as function of radius, force profile as function of radius, and combined position/velocity plot as function of time resp. In the latter plot (and associated numpy arrays) you can derive the time to reach the center and the speed achieved there. Plot solutions for $n = 0, 1, 2$ in the same plot.

Try higher values of $n$, for example $n = 10$ or $n = 100$. Report this in terms of travel time and maximum speed achieved. Do you feel confident about the accuracy of the solution?

### Part 6

With all of the framework in place, compute the travel time to the center of the moon in case we dig a pole-to-pole mine shaft on the moon. What is the density (we assume a constant density) of the moon compared to that of earth? This problem can actually be done on paper, and compare the answer with that in Part 4 for the homogenous earth.