

Assignment 4:
Introduction to *Mathematica*
Introduction to Data Analysis for Physics

Evan Ott and Will Beason

Spring 2014

Submission Requirements

Submit the assignment to data.analysis.physics@gmail.com by Wednesday at 5pm. Just submit the *Mathematica* document you create (typically a .nb file).

Problem 1

In the Error Analysis section of the textbook (which we haven't gone through yet), there's an example that shows that acceleration due to gravity on the surface of the Earth is just about constant (http://www.cs.utexas.edu/~evanott/PHY110C_Textbook/static/data_analysis/Analysis/error.html#example-watermelon-drop).

Let's use *Mathematica* to see if we can prove this. From Newton, we know that for us near Earth:

$$|a_g(r)| = G \frac{M}{r^2}$$

with radius of the Earth being around 6370km. If we consider our "elevation", e , to be our distance above or below this "mean Earth radius" given by

$$R_{\oplus} = 6.370 \times 10^6 m$$

such that our real radius to the center of the earth is

$$r = e + R_{\oplus}$$

we can make a Taylor series expansion to see how much gravity changes as a function of elevation. Note that $M = 5.97 \times 10^{24} kg$ and $G = 6.67 \times 10^{-11} Nm^2 kg^{-2}$.

Remember that when you use the `Series` command, you may need to make the result a regular, computable function by using `Normal` on the result.

a

Use `Series[...]` to make a first-order approximation to $|a_g(e)|$. What is the acceleration due to gravity in Austin ($e = 150m$)? Denver ($e = 1609m$)? Mount Everest ($e = 8848m$)?

b

Compare your first-order approximations to the “true” values computed with $|a_g(r)| = G\frac{M}{r^2}$. Do they over- or under-estimate acceleration due to gravity? By what factor?

c

Repeat steps **a** and **c** for a zeroth-order approximation and second-order approximation. From this, what can you conclude about the approximation of using $a_g = 9.81\frac{m}{s^2}$ near the surface of Earth - perhaps the most fundamental, often unexplained, approximation in introductory physics courses?

d

No problem here, but if this problem made you wonder about how Earth’s gravitational field *actually* varies, check out “New ultrahigh-resolution picture of Earth’s gravity field” in *Geophysical Research Letters* by Hirt et al (2013). The figures show in particular how gravity varies over Mount Everest. In the supplemental information, Figure 3 shows how gravity varies in units of $10^{-5}ms^{-2}$ over the planet. The following link will use the UT proxy service to open the article (so you can look at it with EID even if you’re not on the campus WiFi). <http://onlinelibrary.wiley.com.ezproxy.lib.utexas.edu/enhanced/doi/10.1002/grl.50838/>

Problem 2

With this problem, we’re going to build on what we did last week with the planetary simulation, but only looking at one part: Halley’s comet. The data is on the website for just Halley’s comet with sun always centered at $(0, 0, 0)$, giving a 2D representation of the problem. The data is just (x, y) for the comet, again with each row being separated by 1800s. In this problem, We’ll be using a well-known model for orbits of a comet and see how well it fits our data.

Using the gravitational force $F(r) = GMm/r^2$, and assuming polar coordinates

$$r^2 = x^2 + y^2, \quad \tan(\phi) = \frac{y}{x}$$

we can[1] determine that the radius as a function of angle for a comet is:

$$r(\phi) = \frac{c}{1 + \epsilon \cos(\phi)}$$

where c is a length-scale constant, and ϵ is the unitless eccentricity.

a

Transform your data so that you have a listing of points (ϕ, r) rather than (x, y) . The coordinate transformations above are not quite good enough to specify which solution for ϕ is preferred. In this case, we want ϕ to always be increasing, so once you have calculated values of ϕ , you may want to use `Mod[myAngleList + 2Pi, Pi]`, which, for the values provided, will ensure that ϕ increases and that there are no jumps in the data (to do this for a larger data set would require a more complicated correction function, but here, this is sufficient). Once you have that, use a `ListPlot` on the (x, y) version and a `ListPolarPlot` on the (ϕ, r) data (preferably overlaying them with the `Show` function) to ensure that you have transformed the data correctly (graphs should be identical).

b

Use the `FindFit` function to get fitted values for c and e in $r(\phi) = \frac{c}{1 + \epsilon \cos(\phi)}$. You should be able to use the default least-squares method of curve fitting. Plot this fitted function against the data - does it appear to match well by visual inspection?

c

Using the following:

$$\begin{aligned} m_{\text{Halley}} &= M_H = 2.2 * 10^{14} kg \\ m_{\text{sun}} &= M_{\odot} = 1.989 * 10^{30} kg \\ G &= 6.67 \times 10^{-11} Nm^2 kg^{-2} \\ a &= \frac{c}{1 - e^2} \\ \tau^2 &= \frac{4\pi^2}{GM_{\odot}} a^3 \end{aligned}$$

calculate the period of orbit for Halley's comet based on your data (above, period is simply τ). Does it match well with our knowledge of Halley's comet arriving every 75-76 years?

d

Finally, let's consider whether our model seems to accurately portray what's going on in the simulation. Consider the difference between data and fitted function $r_{obs} - r(\phi)$ for each point in the dataset. Looking at this quantity versus ϕ , t , or r_{obs} , is there a visible pattern to the residuals? How big is this difference in comparison to the actual value of the radius ($(r_{obs} - r(\phi))/r_{obs}$)? Is it negligible?

References

- [1] Taylor, John R. *Classical Mechanics*, Section 8.6 "Kepler Orbits" (2005)