Introduction to Scientific Computing, PSCB57, Fall 2016
Assignment 4
Root finding and interpolation

- The deadline for this assignment is Friday, October 28th, 5pm.
- You must submit the assignment electronically at `http://rein.utsc.utoronto.ca/submit/`. The username is `pscb57`, the password is `2016`.
- Make sure your code passes all tests before submitting it. Your code needs to pass all tests in 3 seconds or less. If your code does not pass the automated tests, you will most likely not get any points for it.
- This assignment comes in multiple parts. Submit all your answers in one file.
- Only submit the functions you wrote, i.e. no print statements, tests or data files.
- Do not use any packages or libraries. Note that you can use the syntax `x**0.5` to calculate the square root of `x`.
- The entire submission has to be 90 lines of code or less (including comments).
- You must be present at the tutorial in the following week to take a quiz. If you do not show up or fail to pass the quiz, your assignment will be marked as 0% even if it was correct.
- The questions are motivated by astronomy, but you don't need to know anything about astronomy or physics to solve them.

Part 1

In this part, you will calculate the location of the *Lagrange Point L1* between the Earth and the Sun. This is a special location in the Solar System where the gravitational force from the Earth and the Sun equals the centrifugal force felt by a third body. The exact location of L1 can only be calculated numerically. To do this, implement a function `getL1(M_1,M_2)` that uses Newton's method to solve the following equation for r:

$$0 = -\frac{M_1}{(R-r)^2} + \frac{M_2}{r^2} + \frac{M_1}{R^2} - \frac{r\,(M_1+M_2)}{R^3} \tag{1}$$

where w.l.o.g. you may set $R = 1$. For Newton's methods, you need the derivate of this equation. Calculate the derivative analytically. As an initial guess for Newton's methods you may use $r_0 = 0.1$. Once your solution has converged to almost machine precision, return the approximate value of r.

M_1 and M_2 are the masses of the two massive bodies, i.e. the Sun and the Earth. Your function should work for any value of M_1 and M_2, but you may want to test your code with the values for the Sun and the Earth, $M_1 = 1.0$, $M_2 = 3.0 \cdot 10^{-6}$, for which the solution is $r \approx 0.00996655277$. For the physicists among you: we're working in units where $G = 1$.

Part 2

1. Go to NASA's Horizon website at `http://ssd.jpl.nasa.gov/?horizons` and download the trajectory of both the *Voyager 1* spacecraft and *Saturn*. To do this, click on 'web interface', then select the following options:

Ephemeris Type:	Vector Table
Target Body:	Voyager 1
Coordinate Origin:	Solar System Barycenter
Time Span:	Start=1980-11-10, Stop=1980-11-30, Step=1 d
Table Settings:	defaults
Display/Output:	plain text

   Repeat the above process, with the exact same options except for the target which you should change to *Saturn Barycenter*. Save the complete output as two text files, `voyager1.txt` and `saturn.txt`.

   The data format is to be understood as follows. The important part starts after the sequence `$$SOE` and ends before the sequence `$$EOE`. In between those statements, four lines correspond to one data point. For example:

```
2444553.500000000 = A.D. 1980-Nov-10 00:00:00.0000 TDB
 -9.464721239400E+00 -3.384345041628E-01 3.8004817290153E-01
 -8.896083480199E-03 -7.824948864272E-03 4.9991519987601E-04
 5.4742664767881E-02 9.4783975728853E+00 9.1551625552845E-03
```

   Here, the underlined quantities are the time and the x, y and z coordinates of the object. You may ignore all other quantities. You also don't have to worry about the units. For the physicists: the units of time and length in this data file are Julian days and astronomical units, respectively.

2. Write a python program to read in the two data files you just downloaded. You may want to search for the sequences `$$SOE`, then read in four lines at a time for every data point and stop when you encounter the sequence `$$EOE`. If you don't know how to do this, ask for help in the tutorial and we will show you how to do this part. You should create the following three variables:

   - `d_t`: A list of floats that correspond to the time column in your data files (note that this is the same for both files).
   - `d_sat_xyz`: A list of lists, containing the Cartesian xyz coordinates of Saturn as floats, e.g. the list should have the form: `[[-9.4721E+00, -3.848E-01, 3.813E-01], [-9.462E+00, -3.334E-01, 3.723E-01], ...]`.
   - `d_voy_xyz`: Same as above but for the coordinates of Voyager 1.

3. Create a function to interpolate the distance between Saturn and Voyager 1 in-between data points. Name the function `getDistance(t)`. It should return the interpolated distance at the time t. To do this, first calculate the distance between the two bodies at the two nearest datapoints, then use the linear interpolation method to approximate the distance at time t. You may assume that t is in the range where you have data points.

4. Implement a function `getTime(d)` which uses the bisection method and your function `getDistance(t)` from above to find when Voyager 1 was exactly a distance d away from Saturn. Stop the bisection when you have narrowed down the time interval to within 0.0001 days. You may assume that the answer is always in the range where you have data points.

   You can test your function by searching for the time when Voyager 1 was 0.03542 AU away from Saturn. The answer should be $\approx$ 2444560.23090 JD, which is Nov. 16, 1980. The following image was taken at that time at a distance of 5.3 million kilometres.

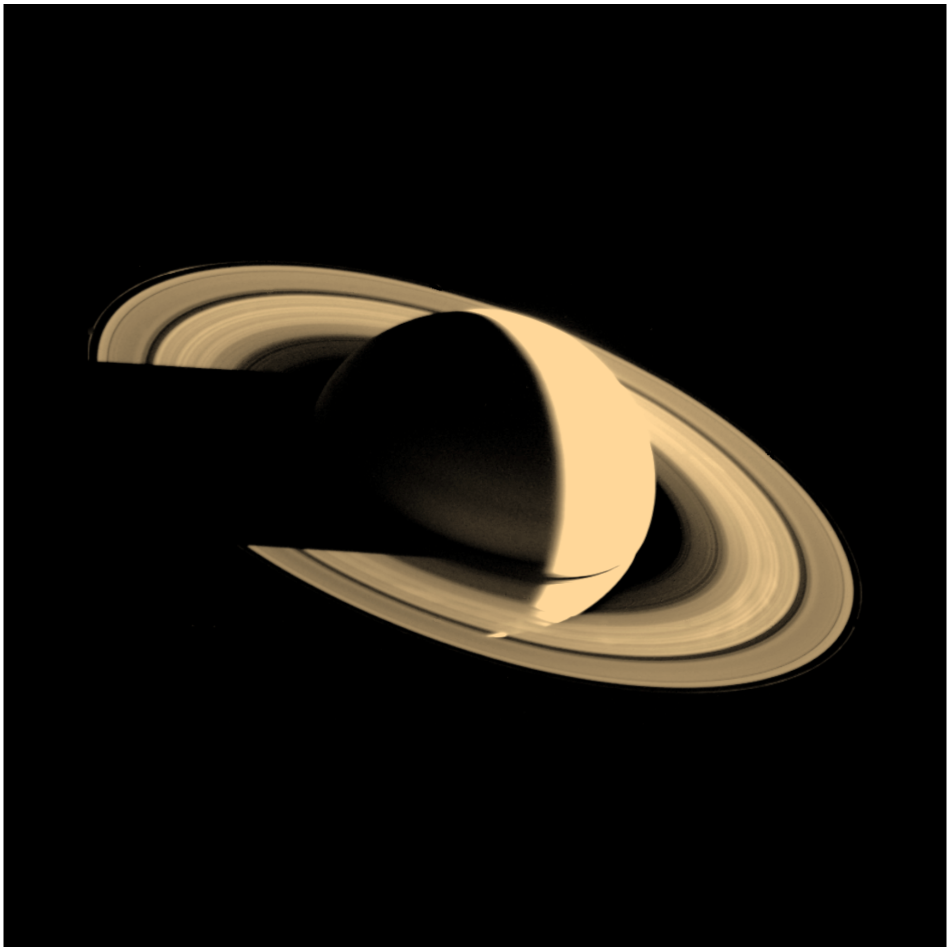

Figure 1: A picture taken by Voyager 1 looking back at Saturn on Nov. 16, 1980, four days after the spacecraft flew past the planet, to observe the appearance of Saturn and its rings from this unique perspective. A few of the spoke-like ring features discovered by Voyager appear in the rings as bright patches in this image, taken at a distance of 5.3 million kilometres (3.3 million miles) from the planet. Saturn's shadow falls upon the rings, and the bright Saturn crescent is seen through all but the densest portion of the rings. From Saturn, Voyager 1 is on a trajectory taking the spacecraft out of the ecliptic plane, away from the Sun and eventually out of the Solar System. Image credit: NASA/JPL