Project 3: Binary Numbers and Computational Errors

Due Dates

Week 5, Monday at 4:59

Your submission should be a single zip file containing all the .py files containing your code as well as a Single typed PDF document with your test results and answers to the problems/questions. Word documents will not be accepted. Your zip file should be in the form <lastname>_project_< N >.zip where you replace <lastname> with your last name and <N> with the project number. Upload your zip file via the link provided on the project webpage found at http://urminsky.ca.

Documentation: *Make sure that your code is well enough commented so that it can be easily understood by almost anyone, including yourself, if you look at it again at the end of the course (when you've forgotten how you coded it.)*

Project Briefing

You should **not** use any of built in root-finding functions in any code you did not write.

Part 1

Design, write and test a PYTHON script to find the roots of an arbitrary mathematical function $y = f(x)$ using the bisection method.

Your solution should include a function root_bisect of the form

```
def root_bisect(func,low,high):
    #your solution to the bisect method
    ...
    return approx_root
```

where low and high define the interval which contain the root. Here func refers to an arbitrary python function of one real variable. For example,

```
def f(x):
    return (x-1)+1
```

In this way we can pass $f(x)$ to root_bisect as

```
root_bisect(f,0.5,1.5)
```

When designing your program, think carefully about any special cases. For example, what if the initial guess does not bracket a root? or what if the initial guess is a root.

Use your program to find all real roots of the following functions

i. $f(x) = \sin(\sqrt{x}) - x$

ii. $f(x) = -2 + 6x - 4x^2 + 0.5x^3$

iii. $f(x) = \sin x + \cos(1 + x^2) - 1$

iv. $f(x) = x^{10} - 1$

v. $f(x) = -(x-2)^2$

In some cases there is more than one root. You will need to re-run the program for each root. Be careful of your initial guesses in these cases. It is a good idea to plot each function beforehand in order to make a reasonable initial guess.

Use the following termination criterion: fractional estimated error $\epsilon_a \leq \epsilon_s$ where $\epsilon_s = 1 \times 10^{-6}$. In each case, record the value of the root and the number of iterations taken. When does the bisection method work well and when does it fail? Comment briefly.

Part 2 - page 267 from Newman

Planck's radiation law tells us that the intensity of radiation per unit area and per unit wavelength λ from a black body at temperature T is

$$I(\lambda) = \frac{2\pi hc^2\lambda^{-5}}{e^{hc/\lambda k_B T} - 1}$$

where h is Planck's constant, c is the speed of light, and k_B is Boltzmann's constant.

(a) Show by differentiating that the wavelength λ at which the emitted radiation is strongest is the solution of the equation

$$5e^{-hc/\lambda k_B T} + \frac{hc}{\lambda k_B T} - 5 = 0.$$

Make the substitution $x = hc/\lambda k_B T$ and hence show that the wavelength of maximum radiation obeys the *Wien displacement law*

$$\lambda = \frac{b}{T}$$

where $b = hc/k_B x$ is the *Wien displacement constant* , and x is the solution to the nonlinear equation

$$5e^{-x} + x - 5 = 0.$$

(b) Write a program to solve this equation to an accuracy of $\epsilon = 10^{-6}$ using your program from Part 1 and hence find a value for the displacement constant.

(b) This method can be used to estimate the surface temperatures of astronomical bodies such as the sun. The wavelength peak in the Sun's emitted radiation falls at $\lambda = 502$ nm. From the equations above and your value of the displacement constant, estimate the surface temperature of the Sun.