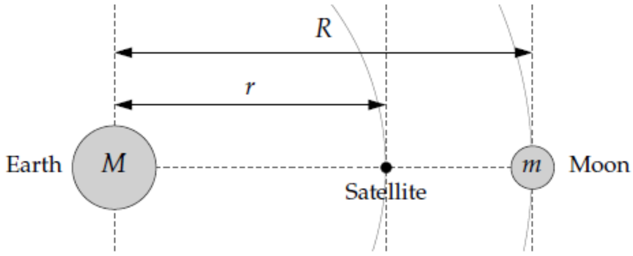
Anjelah Balachandran
Due 20th November 2014

Scientific Computing

Python Assessment 2

1a)



$m_{moon} = Mass\ of\ the\ Moon$

$M_{Earth} = Mass\ of\ the\ Earth$

$m_{satellite} = mass\ of\ the\ satellite$

Using Newton's Law of Gravitation to resolve forces

Newton's law f Gravitation is as follows:

$$F = \frac{GMm}{r^2}$$

Where F is the force between the masses, G is the Gravitational constant, M and m are respective masses of two separate bodies and r is the distance from the centre of each between them.

We form the following by resolving forces:

$$\frac{GMm_{sat}}{r^2} - \frac{Gmm_{sat}}{(R-r)^2} = m_{sat}a = m_{sat}\omega r^2$$

This is the "inward force" provided by the pull of the Earth

This is the "outward force" provided by the pull of the moon and as it is in the opposite direction to the force of pull from Earth it is negative. The value for the distance between the moon and the satellite is $(R-r)^2$ because R is the entire distance from the Moon to Earth and r is the distance between the Earth and the satellite therefore the difference between them gives the distance between the moon and the Satellite

The resultant force is the centripetal force that keeps the satellite in orbit where ω is the angular velocity

Anjelah Balachandran
Due 20th November 2014

Then dividing through by m_{sat}

$$\frac{GM}{r^2} - \frac{Gm}{(R-r)^2} = \omega r^2$$

1b) **Code attached** -I used the “Ridder method” to solve the fifth order polynomial equation.

```
 2
 3 from ridder import*
 4
 5 #Various Parameters:
 6 G=6.674*10**-11
 7 M=5.974*10**24
 8 m=7.348*10**22
 9 R=3.844*10**8
10 w=2.662*10**-6
```

In line 3 I imported ridder so that I could later use it to solve the equation which is a method based on linear interpolation. I then go on to list my various parameters I will use.

```
10 w=2.662*10**-6
11
12 def f(r):
13         return G*M*(R-r)**2-G*m*r**2-w**2*r**3*(R-r)**2 #
   fractions
14 root = ridder(f,0,R,tol=1.0e-9)
15 print 'Using the ridder function r is calculated to be:'
16 print '%e' %(root) #gets in scientific form
```

I’ve defined my function as the equation I obtained from part 1a that’s been rearranged such that there are no fractions (I brought the $m_{sat}\omega r^2$ over to the other side and multiplied everything through by $r^2(R-r)^2$). I then used the ridder function with limits of 0 and R to solve for r as we are going from Earth up to the Moon at a distance R away from Earth. Below is the solution after it has been run:

```
In [6]: run Py2Q1.py
Question 1b
Using the ridder function r is calculated to be:
3.260451e+08
```

Therefore the solution to five significant figures would be 3.2605×10^8m and this is a rough approximation of the location of L1.

Anjelah Balachandran
Due 20th November 2014

2a)

The taylor series approximations are as follows

$$f(x-h_1)=f(x)-h_1f'(x)+\frac{h_1^2}{2!}f''(x)-\frac{h_1^3}{3!}f'''(x)+\frac{h_1^4}{4!}f^{(4)}(x)+\cdots \quad (1)$$

$$f(x+h_2)=f(x)+h_2f'(x)+\frac{h_2^2}{2!}f''(x)+\frac{h_2^3}{3!}f'''(x)+\frac{h_2^4}{4!}f^{(4)}(x)+\cdots \quad (2)$$

Adding the two taylor approximations [(1) + (2)] gives

$$f(x-h_1)+\ f(x+h_2)=2f(x)+f'(x)(h_2-h_1)+f''(x)\frac{(h_1^2+h_2^2)}{2!}+f'''(x)\frac{(h_2^3-h_1^3)}{3!} \quad (3)$$

Subtracting the two taylor approximations [(1) – (2)] gives

$$f(x+h_2)-f(x-h_1)=f'(x)(h_2+h_1)+f''(x)\frac{(h_2^2-h_1^2)}{2!}+f'''(x)\frac{(h_2^3+h_1^3)}{3!} \quad (4)$$

I can rearrange equation (4) to give $f'(x)$

$$f'(x)(h_2+h_1)=\ f(x+h_2)-f(x-h_1)$$

Divide through by (h_2+h_1)

$f'(x)=\ \frac{f(x+h_2)-f(x-h_1)}{(h_2+h_1)}$ + truncation error (5)

The truncation error for $f'(x)$ is as follows

Error on $f'(x)=f''(x)\frac{(h_2^2-h_1^2)}{2!((h_2+h_1)}$ (6)

The error on $f'(x)$ is therefore 1st order as $\frac{{h_2}^2}{h_2}\sim h_2$

I can rearrange equation (3) to give $f''(x)$

$$f''(x)\frac{(h_1^2+h_2^2)}{2!}=f(x-h_1)+\ f(x+h_2)-2f(x)-f'(x)(h_2-h_1)$$

Divide through by $\frac{(h_1^2+h_2^2)}{2!}$

$f''(x)=\frac{2(f(x-h1)+\ f(x+h2)-2f(x)-f'(x)(h_2-h_1))}{(h_1^2+h_2^2)}$ + truncation error

(7)

The truncation error for $f''(x)$ is as follows

Error on $f''(x)=f'''(x)\frac{2(h_2^3-h_1^3)}{3!((h_1^2+h_2^2)}=f'''(x)\frac{(h_2^3-h_1^3)}{3((h_1^2+h_2^2)}$ (8)

The error on $f''(x)$ is therefore 1st order as $\frac{{h_2}^3}{{h_2}^2}\sim h_2$

Anjelah Balachandran
Due 20th November 2014

2b) **Code attached**

	$x-h_1$	x	$x+h_2$
x	0.97	1.00	1.05
$f(x)$	0.85040	0.84147	0.82612

```
h1=0.03 #because 1-0.97 as we take x0 to be 1
h2=0.05 #because 1.05-1.00 again as we take x0 to be 1
fxh1=0.85040
fx=0.84147
fxh2=0.82612

```

I defined my variables by calculating h_1 to be $1.00 - 0.97 = 0.03$ and h_2 to be $1.05 - 1.0 = 0.05$. I defined $f(x)$ at h_1 to be fxh1, $f(x)$ at x to be fx and $f(x)$ at h_2 to be fxh2.

```
y1=(fxh2-fxh1)/(h2+h1)

print "f'(1.00)=",y1
print ""

y2= (2*(fxh2+fxh1-(2*fx)-(y1*(h2-h1))))/(h2**2+h1**2)
print ""
print "f''(1.00)=",y2


```

I defined $f'(x)$ as y1 and $f''(x)$ as y2. I then went on to produce the solution for $f'(1)$ and $f''(1)$ i.e the variables defined earlier were substituted into the equations and an answer was generated. I used the solution for y1 and substituted it into y2 in order to ensure complete accuracy, as $f'(x)$ is in the y2 equation. The solutions obtained are as follows:

```
Question 2b

f'(1.00)= -0.3035


f''(1.00)= -0.205882352941
```

Anjelah Balachandran
Due 20th November 2014

3a) Deriving the central difference approximations for $f'(x)$ accurate to O(h^4). The centred approximation is as follows:

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} + o(h) \qquad (1)$$

We want to compute some quantity G dependent on some parameter h

$G(h) = g(h) + E(h)$ where $g(h)$ is the approximation and $E(h)$ is the error that takes the form $E(h) = ch^p$, i.e

$$\underbrace{f'(x)}_{G(h)} = \underbrace{\frac{f(x+h)-f(x-h)}{2h}}_{g(h)} + \underbrace{O(h)}_{E(h) = \ ch^p}$$

The Richardson extrapolation formula at $h_2=\frac{h_1}{2}$ is as follows:

$$G = \frac{2^p g\left(\frac{h_1}{2}\right) - g(h_1)}{2^p - 1} \qquad (2)$$

We are approximating O(h^2) and so the power of h is 2, thus p is 2. Applying p=2 into equation (2) gives:

$$G = \frac{4g\left(\frac{h_1}{2}\right) - g(h_1)}{4-1} = \frac{4g\left(\frac{h_1}{2}\right) - g(h_1)}{3} \qquad (3)$$

I will substitute (3) into (1) to give:

$$g(h) = \frac{4f\left(x+\frac{h_1}{2}\right) - 4f\left(x-\frac{h_1}{2}\right)}{2 \times \frac{3h_1}{2}} - \frac{(f(x+h_1) - f(x-h_1))}{2 \times 3h_1}$$

$$g(h) = \frac{4f\left(x+\frac{h_1}{2}\right) - 4f\left(x-\frac{h_1}{2}\right)}{3h_1} - \frac{(f(x+h_1) - f(x-h_1))}{6h_1}$$

I will now put it in the form $G(h) = g(h) + O(h)$:

$$G(h) = \frac{4f\left(x+\frac{h_1}{2}\right) - 4f\left(x-\frac{h_1}{2}\right)}{3h_1} - \frac{(f(x+h_1) - f(x-h_1))}{6h_1} + O(h^4) \qquad (4)$$

Anjelah Balachandran
Due 20th November 2014

3b) **Code attached-**

Since we are approximating the derivative for tanh(x), I have substituted tanh(x) into (4) from part a

$$G(h) = \frac{4tanh\left(x + \frac{h_1}{2}\right) - 4tanh\left(x - \frac{h_1}{2}\right)}{3h_1} - \frac{\left(tanh(x + h_1) - tanh(x - h_1)\right)}{6h_1} + O(h^4)$$

```
import math as m

#Variables

h1=0.5
h2=0.1
m.tanh(1+h1)
def G(h):
        x=1.0
        return (((4*m.tanh(x+(h/2)))-(4*m.tanh(x-(h/2))))/(3*h))-((m.tanh(x+h)-m.tanh(x-h))/(6*h))

print "at x=1 and h=0.5"
print "G(0.5) =",G(h1)
print "at x=1 and h=0.1"
print "G(0.1) =",G(h2)
```

I imported math as m such that I can apply my tanh function. I defined a function G(h) that returned the equation I formed from part a. The function has 1 parameter h, which we choose suitably when we call the function (our values for h1 and h2). The values at h=0.5 and h=0.1 are as follows:

```
Question 3b
at x=1 and h=0.5
G(0.5) = 0.420682134726
at x=1 and h=0.1
G(0.1) = 0.419975498528
```

The true value of the first derivative of tanh(x) at 1 is 0.41997434161402606939

The error for when h=0.5

$$Actual\ value - my\ obtained\ value = 0.41997434 - 0.42068213 = -0.00070779$$

The error for when h=0.1

$$Actual\ value - my\ obtained\ value = 0.41997434 - 0.41997551 = -0.000001157$$

The errors do behave as expected. As 0.1<0.5, the width of the partition is smaller (and so there are more of them). As a result of having more (and smaller) partitions, the error (or over/underestimate) is less extreme, thus closer to the true value.

Anjelah Balachandran
Due 20th November 2014

4a) A perfect black body of unit area radiates electromagnetic an amount of thermal energy per second where:

$$I(\omega) = \frac{\hbar}{4\pi^2 c^2} \frac{\omega^3}{(e^{\frac{\hbar\omega}{K_bT}} - 1)} \quad (1)$$

$$\int I(\omega) d\omega = W \quad (2)$$

Therefore we have

$$W = \int I(\omega) d\omega = \int \frac{\hbar}{4\pi^2 c^2} \frac{\omega^3}{(e^{\frac{\hbar\omega}{K_bT}} - 1)} d\omega \quad (3)$$

I will perform an integration by substitution where I will let $x = \frac{\hbar\omega}{K_bT}$

Differentiating x with respect to ω gives

$\frac{dx}{d\omega} = \frac{\hbar}{K_bT}$ rearranging this for $d\omega$ gives $d\omega = \frac{K_bT}{\hbar} dx$

Substituting this into (3) gives

$$W = \int \frac{\hbar}{4\pi^2 c^2} \frac{\omega^3}{(e^{\frac{\hbar\omega}{K_bT}} - 1)} \frac{K_bT}{\hbar} dx$$

Since $x = \frac{\hbar\omega}{K_bT}$ we can rearrange for ω^3 as this is a substitution I will need to make

$\omega = \frac{xK_bT}{\hbar} \rightarrow \omega^3 = \frac{x^3 K_b^3 T^3}{\hbar^3}$

Now substituting this into our integral gives

$$W = \int \frac{\hbar}{4\pi^2 c^2} \frac{x^3}{(e^{\frac{\hbar\omega}{K_bT}} - 1)} \frac{K_b^3 T^3}{\hbar^3} \quad \frac{K_bT}{\hbar} dx$$

Since $x = \frac{\hbar\omega}{K_bT}$ we can substitute x into $(e^{\frac{\hbar\omega}{K_bT}} - 1)$ to give $(e^x - 1)$. Finally substituting this into our integral removes all ω terms and our substitution is complete:

$$W = \int \frac{\hbar}{4\pi^2 c^2} \frac{x^3}{(e^x - 1)} \frac{K_b^3 T^3}{\hbar^3} \frac{K_bT}{\hbar} dx \quad (4)$$

Simplifying (4) gives

$$W = \int \frac{K_b^4 T^4}{4\pi^2 c^2 \hbar^3} \frac{x^3}{(e^x - 1)} dx$$

as required.

Anjelah Balachandran
Due 20th November 2014

4b) Evaluating the integral from 0 to infinity

$$W = \int_0^\infty \frac{K_b^4 T^4}{4\pi^2 c^2 \hbar^3} \frac{x^3}{(e^x - 1)} dx$$

```
2 import numpy as np
3 import scipy.integrate
4
5 def W(x):
6     return (x**3/(np.exp(x)-1))
7 I=scipy.integrate.quad(W,0,np.inf)
8 print I
9 print "Work=",I[0]
10 print "error=",I[1]
11 #constants I need to times the integral by:
12 Kb=float (1.380648813*10**(-23))
13 h=float(1.05457172647*10**(-34))
14 c=float(299792458)
15
```

I imported the scipy.integrate package to compute my integral and defined my function as $W(x)$that returned the equation $\frac{x^3}{(e^x-1)}$ and used the scipy.integrate package to compute the integral between 0 and infinity shown by "0,np.inf". This generated a solution that gave me the value for work (without constants) and the error on this value.

```
(6.4939394022668298, 2.6284700289248249e-09)
Work= 6.49393940227
error= 2.62847002892e-09
```

The error is many magnitudes smaller than our value for work, therefore we can say that the approximation is accurate.

As this only gave me an exact value for the integral without the constants, I went on to define the constants (lines 2,3,4) and then multiply this by my exact integrated solution. As scipy.integrate.quad outputs an array with two items, the first being the solution and the second being the error, my calculation was done using only the solution (the first item in the array) and thus I used I[0] multiplied by the constants:

```
17 w=float(I[0]*(Kb**4)/(4.0*(np.pi)**2*c**2*h**3))
```

The value for work was therefore calculated to be: 5.67037281787e-08 T^4

(When calculating the work done in Python, I didn't reference the T^4 as it is a constant variable that will cancel when finding the final solution in part c)

4c) Stefan's law:

$$W = \sigma T^4$$

Since we have W from part 4b to equal $5.67037281787 \times 10^{-8}\, T^4$ we can equate this to σT^4. This gives the following:

$$5.67037281787 \times 10^{-8}\, T^4 = \sigma T^4$$

T^4 Will cancel giving Stefan Boltzmann's constant $\sigma = 5.67037281787 \times 10^{-8}$ W m^{-2} K^{-4}. The known value of the Stefan-Boltzman constant is 5.67037321 × 10^{-8} W m^{-2} K^{-4}, this is therefore a good agreement as the values are very close to one another.

Error percentage =$\frac{Actual\ Value - Value\ obtained}{Actual\ Value} \times 100 = \frac{5.67037321 \times 10-8 - 5.67037281787\times10^{-8}}{5.67037321 \times 10-8} \times 100$

= 6.9154x 10^{-8} is the relative error and 6.9154x 10^{-6} % is the relative percentage error.