Appendix 1

Assignment 1 Problems.

Final version (I will advise of minor corrections).

Question 1

We will be developing a code to solve a complex problem modelling a hanging cable. The steps below will help you build a progressively more sophisticated code. Not all steps need to handed in.

Step 1.

Write a Newton's solver to find a solution to the following system of four equations

```
t=z[0]
u=z[1]
v=z[2]
w=z[3]
vec[0]=t**4+u**4-1
vec[1]=t**2-u**2+1
vec[2]=v**4+w**4-1
vec[3]=v**2-w**2+1.
```

Use an exact calculation for the Jacobian. You code should use python functions to evaluate the above vector of functions and the Jacobian. Your Newton's solver should use a loop. Report the solutions you found. Do not submit code.

Step 2.

Write a function to find a numerical version of the Jacobian based on forward differences with Delta x = 0.02. Use the new code to resolve the problem in Step 1. Comment on the difference with Step 1.

Step 3.

Find the two solutions to the equation vec=0 where the equations are:

```
t=z[0]
u=z[1]
v=z[2]
w=z[3]
vec[0]=2*(u-t)**2-4*(u-t)+v**2+3*w**2+6*w+2
vec[1]=(u-t)**2+v**2-2*v+2*w**2-5
vec[2]=3*(u-t)**2-12*u**2+v**2+3*w**2+8
vec[3]=u**2-v**2+t
```

Comment on why you might prefer to use the numerical code for the Jacobian.

Step 4.

Write down the total energy (potential and elastic) of a hanging cable made up of three weightless springs connecting two objects of mass m1 and m2.
The end points of the cable are (x0,y0) and (x3,y3). The masses are located at (x1,y1) and (x2,y2). The springs have Hooke's constants k01, k12, k23 and natural lengths L01, L12 and L23.

The equilibrium position of the cable is obtained by minimising the total energy of the system. Formulate the problem mathematically, and state the equations to be solved.

Step 5

In scientific coding it is useful to develop test cases which you can use to test your code.

Case 1. Consider the case where m1=m2=1kg, k01,k23=100N/m and k12=0.1 with Lij=0.9 for all springs. Provide a calculation of the approximate equilibrium positions and give a short physical description.

Case 2. Consider the case where all springs have kij=100, and length Lij=0.2 and the masses are taken to have m1=m2=0.01. Provide an approximate calculation of the equilibrium positions and give a short physical description.

Step 6

Consider the parameters given below. The dimensions of the lengths are in metres, and spring constants are in N/m. Write efficient code to evaluate your equations.The function representing the four equations has value
Test f [37.2 4.98 -67.6 25.1]

At
Test x (x1,y1,x2,y2)= [1.1 0.9 1.7 0.8].
Use this to confirm your equations are working. Report your values of the test function to five significant figures.

```
global x0,y0,x3,y3,m1,m2,g,k01,k12,k23,L01,L12,L23
x0=0.
y0=1.
x3=3.
y3=1.
g=9.8
m1=1
m2=3
L01=0.9
L12=0.8
L23=0.7
k01=90
k12=100
k23=80
```

Step 7

Test your code by considering the exact problems you studied in Step 5. Provide the results that show your code is working. Comment briefly on the difference between your “exact” results and the approximations of Step 5.

Step 8

Use you code to find the equilibrium position of the masses for the problem in Step 6. Report and plot the positions of your masses and provide a qualitative physical description of the configuration.

My code for plotting:

```
xpos=numpy.array( [x0,x1,x2,x3] )
ypos=numpy.array( [y0,y1,y2,y3] )

plt.figure(1)
plt.ylim(-0.05, 1.05)
plt.xlim(-0.05, 3.05)
plt.plot(xpos,ypos,'o')
plt.show()
```

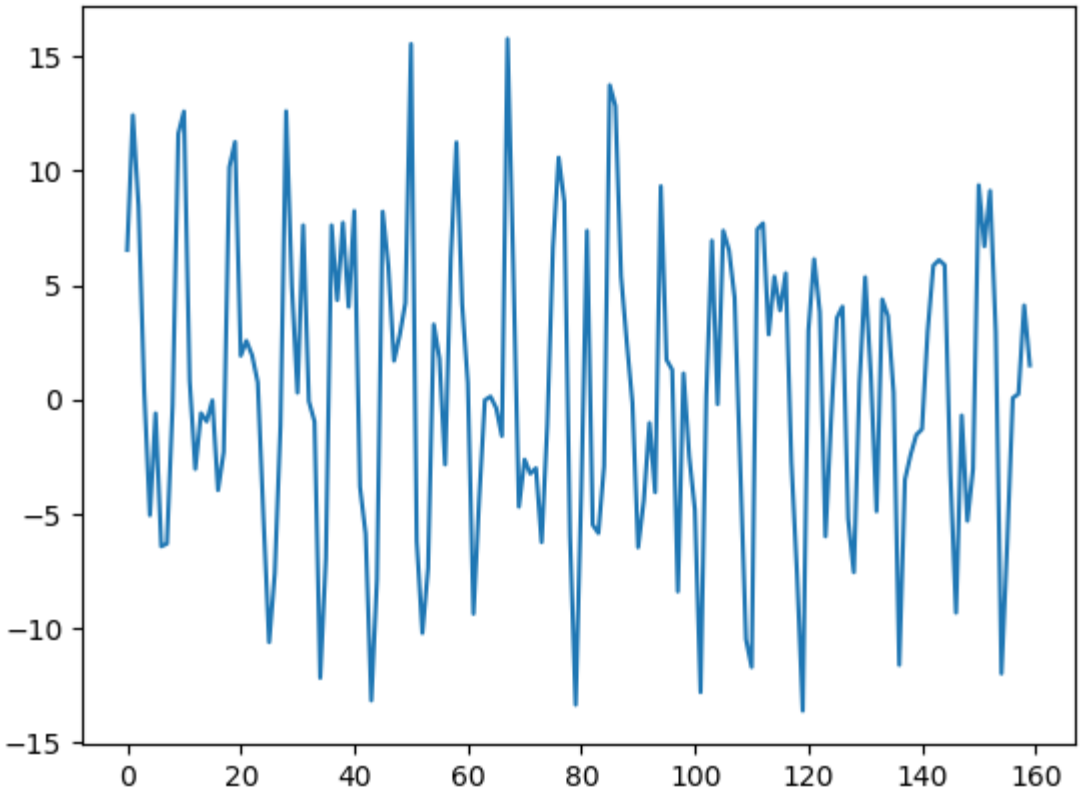
Step 9

Criticize your code, and explain how you would modify your code if the cable had 20 masses.

Question 2

Background
https://www.quora.com/How-do-submarines-differentiate-between-friendly-and-enemy-submarines-How-do-they-do-this-while-remaining-undetected-Do-they-listen-for-the-frequency-of-the-propellers-What-if-the-boat-is-stationary
http://www.acoustics.asn.au/conference_proceedings/AAS2015/papers/p80.pdf

A (fictitious) acoustic time signal recorded in a submarine is given in the second appendix.



In order to detect the presence of another ship or submarine you need to find the spectrum of the signal to see which frequencies are present, and in particular determine if any might correspond to an enemy.

This type of mathematical problem is common in many branches of engineering and science. We will use a Discrete Fourier transform.

(a) Read in the data, and find the DFT using numpy.fft.fft. Plot the real and imaginary parts of the signal and the energy spectrum. The spectrum $E_j = |a_j|$ is the magnitude of each term in the signal.

(b) Your task is to detect the presence of unusual harmonics in the submarine signal (located in an appendix of this file). - this would correspond to the noise of a propellor with a particular frequency. Find any significant harmonics in your signal. Report their frequency in hz (cycles per second). You need to know that the signal is 40 seconds long. Hint: If there was a single sinusoid of period 40 seconds what would it's DFT be.

(c) To roughly illustrate compression using the DFT clip out all the signals whose energy is Less than 10% of the maximum energy. Report the number of terms in the DFT which you set to zero. Plot the "compressed signal" and discuss the comparison with the original signal. Report the compression ratio of the signal. Note that it is not particularly large because we have not exploited the fact the original signal is real.

Question 3.

Find the Fourier series of the function
$f(x) = \pi + x$ for $-\pi < x < 0$ and $f(x) = \pi - x$ for $0 \leq x < \pi$.
Write a program to compute the series for n terms. Show a figure of the case n=1 (1 trig term), and the case n=5). Comment.
Rewrite your trigonometric series for $f(0)$ to show how you can use the series to estimate π.
Evaluate your series up to 10000 points.

Appendix 2 - Data for Assignment

```
data = numpy.loadtxt('data2.txt')   #The online file has 160 data points!
plt.figure(1)
plt.plot(data,'-')
```

6.540897348598405436e+00
1.243189495161463576e+01
8.453586973099067592e+00
3.495854923738575537e-01
-5.052576940215681489e+00
-5.995946544984718241e-01
-6.411289560448475200e+00
-6.292004907396538727e+00
-2.570143090026589250e-01
1.162217738545337298e+01
1.259371743961255774e+01