Appendix 1 - Assignment 1

Due Date: 05 Sep 19 16:00 Week 7
Submit your assignment as a single PDF file plus the set of Python source code files that can be run for verification. A link for upload will be provided on BlackBoard.

Question 1 (25 marks)

Non-linear ODEs arise commonly in engineering problems. A very simple example is Y''-2Y^3=0. In this case the equation has a solution 1/(1+t). Check this result. In the second half of the course you will formulate methods to solve the ODE numerically (which is essential when no analytic solution exists. For a small mesh size (h=0.2) for 0<t<1 the numerical equations to be solved are:

$$y_2 - 2y_1 + 1 - 2y_1^3 h^2 = 0$$
$$y_3 - 2y_2 + y_1 - 2y_2^3 h^2 = 0$$
$$y_4 - 2y_3 + y_2 - 2y_3^3 h^2 = 0$$
$$\tfrac{1}{2} - 2y_4 + y_3 - 2y_4^3 h^2 = 0$$

Write a Newton's solver to find a solution to the equations, and compare your solution to the solution of the underlying ODE (Y=1/(1+h i) where i=1,2,3,4). A set of four points is sufficient.

Use the starting guess y=(1,1,1,1) and stop the program when the maximum absolute error *in the solution of the four equations* is less than 0.00001. Report the maximum absolute error for each step of your method.

This question will be marked based on your code and results. No discussion is needed.

Question 2 (25 marks)

An experiment yields the data x,y

1.000,4.521
1.250,4.835
1.500,4.934
1.750,5.047
2.000,5.043
2.250,5.039
2.500,5.325
2.750,5.801

Your task is to write a general program to compute the least squares estimate of a function of the form

$$y(x) = \sum_{0}^{N-1} a_n \, x^n$$

For general N. I recommend using the vector-based method of generating the normal equations discussed on the tutorial sheet - but you may use other methods as well.

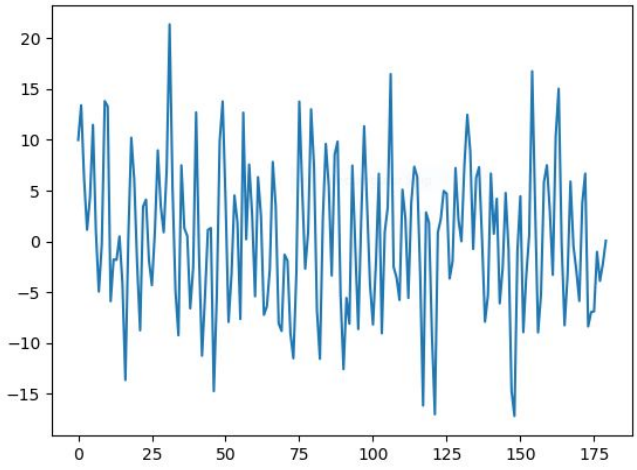
Demonstrate your code by plotting the data against your line of best fit for the case N=2 (a line), N=4 (a cubic), and N=8. Calculate the norm of the error in your fit for all values of N. (17 marks)

Comment on the results, and provide a one paragraph justification of which value of N is best. Consider the concepts of under- and over-fitting in your justification. (8 marks)

Question 3 (25 marks)

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 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. (14 marks)

(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 0.45 seconds long. Therefore the time interval per sample is 0.0025 seconds as the duration of the signal is taken to be from t_o to t_N not t_o to t_{N-1}.

Hint: If there was a single sinusoid of period 0.45 seconds what would it's DFT be. (8 marks)

(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. Note that it is not particularly large because we have not exploited the fact the original signal is real. (3 marks)

Question 4 (25 marks)

Consider placing C units of mass at the point x=50 on a lattice of length 100 (labelled 0 up to 99 in python). At each time step a fraction “p” of the mass at a lattice point steps to the left, and fraction “p” steps to the right (so 1-2p stays at the point). Write a simulation program to determine the mass at each point on the array at later times. You need only consider the sites 1 to 98 and ignore the mass steps from site 1 to 0 or 99 to 98 (for this part of the question).

The mass is known to approximately follow a Gaussian distribution at later times

$$Mass = C \frac{1}{\sqrt{4\pi pt}} exp(-\frac{(x-50)^2}{4pt})$$

(a) Plot your results at time 50 and time 200 in comparison to the theoretical distribution. Also compute the total mass on the lattice at each time. This is useful to help you debug the code. Use the value C=1 in your computations. (20 marks)

(b) Compute the lost mass at T=200 and at T=4000, and confirm that the total mass is C=1. Plot your result at T=4000 against

$$0.00735 \times sin\left(\frac{x\pi}{100}\right)$$

Comment on the form of the curve (do not attempt to explain the value 0.00735). ((5 marks).

(c) (Do not hand in) Confirm that

$$c(x,t) = C \frac{1}{\sqrt{4\pi pt}} exp(-\frac{(x-50)^2}{4pt})$$

Satisfies the diffusion equation

$$\frac{\partial c}{\partial t} = p\frac{\partial^2 c}{\partial x^2} \ .$$

Appendix 1 - Data for Assignment

```
data = numpy.loadtxt('data.txt')   #The online file has 180 data points!
plt.figure(1)
plt.plot(data,'-')
```

9.98724354366794 7421e+00
1.34049104652682 1961e+01
6.16969151432765 3614e+00
1.14832842368891 5967e+00
4.35269837018698 6132e+00
1.14821240170441 7829e+01
9.71267505201182 5461e-01
-4.93679285194095 6517e+00
-2.65121594801308 0359e-01
1.38155871847387 4947e+01
1.32956924803426 8902e+01
-5.88348200942991 2729e+00
-1.76951580358788 1519e+00
-1.81483921870118 7029e+00
5.03577916515813 9614e-01
-4.09116441326069 9462e+00
-1.36396846987179 5897e+01
3.81161437748129 6612e-01
1.02163441883181 1431e+01
6.09161628073754 7129e+00
-2.29369956806122 9321e+00
-8.74925871326123 0724e+00
3.43221731673001 0737e+00
4.12106852377318 8552e+00
-1.88108699889642 6499e+00
-4.30247760743289 3869e+00
9.45833276761196 4398e-01
8.96613889445148 0823e+00
3.48899405274490 6593e+00
9.03628014160909 3208e-01
6.94546431389783 1587e+00
2.13642562259514 7900e+01
5.26755267325698 2843e+00
-4.90928357520652 9804e+00

-9.237776478298023619e+00
7.498157003582328173e+00
1.302620817655704810e+00
5.513687067764677652e-01
-6.588789656067187472e+00
-2.596959971271362466e+00
1.270876651021050030e+01
-1.460555374107379789e+00
-1.124844645674230037e+01
-5.455770373149257502e+00
1.145265705672336320e+00
1.320648890741753512e+00
-1.473396010521326538e+01
-5.502445810126888048e+00
9.892216824344266968e+00
1.378648628865621362e+01
3.892921315098117674e+00
-7.924365839429180269e+00
-3.244012374782762276e+00
4.534450318310955019e+00
1.910943832074602389e+00
-7.640236444792759762e+00
1.268101454012760421e+01
2.143832249611758445e-01
7.574887614563384552e+00
2.628507235875043424e+00
-5.402826437709069118e+00
6.332033510292798262e+00
2.544226373673209363e+00
-7.227969065481523181e+00
-6.396505943997229160e+00
-2.734660405694756768e+00
7.829120763522185555e+00
3.457561223563077490e+00
-8.066845603859270852e+00
-8.816907137117542703e+00
-1.297560413485399300e+00
-1.896025868768696832e+00
-9.056930014741473300e+00
-1.150666404076718585e+01
-3.226035365598467930e+00
1.376787360271863925e+01
5.069296363389677040e+00

-2.684347376340194380e+00
9.105951056503721608e-01
1.301250400138211383e+01
7.335819073404870494e+00
-6.841712869401051833e+00
-1.155587560456141460e+01
2.330716111962127357e+00
9.604401330876227050e+00
5.406073931328389648e+00
-3.360006074898999273e+00
8.555622884512249726e+00
9.839378597799036896e+00
-5.282736014128967206e+00
-1.257087895131846444e+01
-5.558252890032655102e+00
-8.101860924941437503e+00
7.472139082421951350e+00
-6.187796335720885388e-01
-8.626919585308881366e+00
2.112370970483592014e+00
1.134509336190035533e+01
3.661432225690159026e+00
-4.267071349282812065e+00
-8.178768437455303797e+00
-2.141033167544401650e+00
6.671535906056370813e+00
-9.042811231284469287e+00
9.349792175137960903e-01
3.364390124425888473e+00
1.647443209412781684e+01
-2.507508700651045253e+00
-3.551153459004377844e+00
-5.768294356333894868e+00
5.104247089335974152e+00
2.246983158356754995e+00
-5.563400393110962128e+00
3.855303015525190169e+00
7.366929273103508535e+00
6.364618774133141699e+00
-3.038267123866243757e+00
-1.615514400368299164e+01
2.860477470276936884e+00
1.853208690987784069e+00

-9.152962232469928239e+00
-1.701603516642196112e+01
8.763809632736009325e-01
2.202258119070212405e+00
4.980523668024864570e+00
4.709072515474852061e+00
-3.650828190205646084e+00
-1.941659265908209253e+00
7.230010962473746083e+00
2.144789468825352596e+00
1.822172405498988235e-02
7.452451099508453858e+00
1.247041942180738694e+01
8.885641881082390725e+00
-7.452700630101535273e-01
6.282183725487252346e+00
7.318611389734038575e+00
2.317894055049330437e-01
-7.905836671753839973e+00
-5.297323588790337823e+00
6.689228104893055082e+00
7.553246937839011022e-01
4.206910604495008421e+00
-6.092043870275335671e+00
-2.624875731991682226e+00
4.772815464991005285e+00
-8.090683976869824656e-01
-1.470412768323493680e+01
-1.718067489862992403e+01
-9.642864723342969846e-01
4.449310019633980318e+00
-8.930826497849061241e+00
-3.338273006310933155e+00
6.427951538829744793e-01
1.675711215490241912e+01
3.544088988982552646e+00
-8.960714247034362145e+00
-5.195669990925959603e+00
5.849760069873754631e+00
7.506431773738678537e+00
3.025327681942764801e+00
-3.292336159090587078e+00
1.039253753861260776e+01

1.503801257672754588e+01
-2.196068136298584272e-01
-8.258431390232832570e+00
-3.401825537027566071e+00
5.903409326388889156e+00
-4.398916243201960730e-01
-2.878590196739376328e+00
-5.860933718543889448e+00
3.761122982375680213e+00
6.684503262742381047e+00
-8.366577852397467652e+00
-6.950747814387728596e+00
-6.886544959219708772e+00
-1.020650563819176071e+00
-3.885215076205138285e+00
-2.166443391768472448e+00
7.547487463371124750e-02