MAS2008 SAMPLE ASSIGNMENT:
THE LOTKA-VOLTERRA MODEL

1. Assignment

In this assigment you will investigate the Lotka-Volterra model, which governs the time evolution of the populations of two species, which we will take to be sharks and fish. The model is given by equations

$$\dot{F} = \alpha F - \beta S F$$
$$\dot{S} = -\gamma S + \delta S F,$$

depending on four parameters $\alpha, \beta, \gamma, \delta$ (which are all strictly positive real numbers). You will need to submit a Jupyter notebook containing text cells with explanations and mathematical arguments, as well as code cells with Python code to complete various tasks. Further details of the submission format are in Section 2.

1.1. **Background.** Find internet sources to explain the meaning of these equations, and the interpretation of the parameters $\alpha, \beta, \gamma, \delta$. Digest these sources and write your own explanation.

1.2. **Theorems.** Prove the following theorems about the Lotka-Volterra model. For some of them you will be able to find proofs on the internet, but you will need to rewrite them in your own words using the same notation and terminology as in this brief. For some of them it is unlikely that you will find exactly the right statement on the internet (although you may find something similar) so you will need to write your own proof.

Theorem 1. *Put $F_0 = \gamma/\delta$ and $S_0 = \alpha/\beta$. Then there is a constant solution given by $(F, S) = (F_0, S_0)$ for all t.*

Theorem 2. *If (F, S) is a solution for the Lotka-Volterra equations, then the quantity $V = \beta S + \delta F - \alpha \log(S) - \gamma \log(F)$ does not change over time.*

Theorem 3. *Suppose we have a solution $(F(t), S(t))$ to the Lotka-Volterra equations. Suppose that this is periodic of period T for some $T > 0$, so that $F(t+T) = F(t)$ and $S(t+T) = S(t)$ for all t. Then the average value of F is $F_0 = \gamma/\delta$, and the average value of S is $S_0 = \alpha/\beta$.*

Theorem 4. *Suppose we have a solution (F, S) to the Lotka-Volterra equations. Put $f = F/F_0 = \gamma^{-1}\delta F$ and $s = S/S_0 = \alpha^{-1}\beta S$ and $\tau = \sqrt{\alpha\gamma}t$ and $m = \sqrt{\alpha/\gamma}$, and write p' for $dp/d\tau$. The functions f and s then satisfy the following simplified equations:*

$$f' = m(1 - s)f$$
$$s' = -m^{-1}(1 - f)s.$$

1.3. **Solving the equations.** All of the following tasks are based on the rescaled LV equations

$$\dot{f} = m(1 - s)f$$
$$\dot{s} = -m^{-1}(1 - f)s.$$

(These are as in Theorem 4 except that we have replaced τ by t.)

Write a function `solve_lotka_volterra` as follows:

- The arguments should include the parameter `m`, a starting value `x0` (with $0 < x0 < 1$) and a time limit `t_max`. All arguments should have reasonable default values
- Your function should call an appropriate function from the `scipy` package to solve the rescaled LV equations for t from 0 to `t_max` starting with `f=s=x0` at `t=0`.
- If you find it necessary you can include extra arguments to specify the number of evaluation points, the error tolerance and so on.

- The return value should be a tuple `(t, f, s)`, where `t` is an array of t values between `0` and `t_max`, and `f` is an array of values of the variable f at those t values, and `s` is an array of values of the variable s.

Write a function `show_lotka_volterra()` which accepts the same arguments as `solve_lotka_volterra()`, and produces a `matplotlib` picture illustrating the solution.

- The left half of the picture should show the graph of f against t in one colour, and the graph of s against t in another colour.
- The right hand half should show the curve traced by the vector (f, s) as t varies.
- You should include appropriate labels on the curves and/or the axes.
- The value of m should be displayed somewhere in the plot.
- You should think about whether it would be helpful to include any other elements in the plot.

1.4. **Finding the period.** Provided that we start in the meaningful zone where $f, s > 0$, all solutions of the rescaled LV equations are periodic: there is a constant T such that $f(t+T) = f(t)$ and $s(t+T) = s(t)$ for all t. This can be found as follows: we start at a point $(f, s) = (x_0, x_0)$ where $f = s$, then we cross the line $f = s$ again after roughly half a period, then we meet the line $f = s$ for the third time when $t = T$. Write a function `lotka_volterra_period(m)` that finds T.

- If we start with `t_max` being too small, then our solution may not cover a full period so we will not be able to find the times when the solution crosses the line $f = s$. If this occurs then your function should handle the problem by itself: it should increase `t_max` (perhaps by doubling it) and then try again.
- By following the above ideas you can find an index i such that the solution crosses the line $f = s$ between $t = t_i$ and $t = t_{i+1}$, so $t_i < T < t_{i+1}$. As a crude approach, you could return t_i or $(t_i + t_{i+1})/2$ as an estimate of T. However, for full credit you should do linear interpolation to give a better approximation of the precise time between t_i and t_{i+1} when the line $f = s$ is crossed.

1.5. **Checking the theorems.** How do Theorems 2 and 3 simplify in the case of the rescaled equations? Choose a value of m and x_0, find the corresponding solution, and check the predictions of those theorems by numerical calculation.

1.6. **Approximating the period.** Fix $x_0 = 0.5$ and calculate the period T for a range of values of m, say from 1/3 to 3. Plot T against m. The dependence can be approximated moderately well by an equation of the form $\log(T) = A\log(m)^2 + B$ for some constants A and B. Use an appropriate curve-fitting approach to find values of A and B, and plot the curve $T_{\text{approx}} = \exp(A\log(m)^2 + B)$ along with the actual plot of T against m.

2. Guidance

- You may discuss the assignment with other students, but you must not copy code or text from them. You must write your own notebook in your own words based on your own understanding. You must also mention any collaboration in the acknowledgements section of your notebook, including the names of people with whom you worked.
- You may search the internet for information, but you must mention all sources that you have used in your acknowledgements, with specific URLs.
- You can use code that you find on the internet or that is given to you by an AI assistant, but you must acknowledge it. You must also ensure that the code you submit complies precisely with the notation and terminology used in this briefing document, and that the function names, arguments and return values are exactly as specified. You will probably need to modify code obtained from elsewhere to achieve this.
- All acknowledgements must appear in a separate markdown cell at the top of your notebook, with heading "Acknowledgements".
- All nontrivial functions should have docstrings.
- For all code that implements a nonobvious algorithm, you should add detailed comments in the code to prove that you understand how the algorithm works.

- When developing your notebook, you will probably move backwards and forwards, inserting things and executing code in different places. However, before submission you should tidy up your code. Remove anything that is not needed and check that the rest can be executed in order from the top to the bottom without errors and that this generates all the required plots and prints all the required messages.
- Upload your notebook using Blackboard.
- Do not include your own name or registration number in your notebook. (Blackboard will ensure that your work is tagged with your name at the point when that becomes necessary.)