COURSEWORK 3 – 5%

This assignment has three parts and graded over **5 pts**. Some general remarks:

- The assignment is due on 14 Dec. 2022. 1PM GMT, to be submitted via Blackboard (see the instructions on the course website).
- You should submit a PDF report. You can do this via two ways (your choice):
  1. Prepare a PDF. You can put relevant code for a question next to your derivations (like an IPython notebook). However **please format the code** – if you are using Word, make sure that the code is boxed and coloured. If you are using LaTeX, then use `pythonhighlight` package (or something similar) in LaTeX for readability (please do!). We do not have much handwritten derivations in this assignment, but you can include handwritten pages if you wish to, however, please organise your report cleanly in this case and submit a single PDF merged together.
  2. Prepare an IPython notebook and export it as a PDF (preferred). If you can't export your HTML notebook as PDF, use "Print" feature in browser (Chrome: File -> Print) and choose "Save as PDF".
- No length limit! But please be concise.
- You can reuse the code from the course material, but try to personalise your code in order to avoid having problems with plagiarism checks. You can use Python's functions for sampling random variables of all distributions of your choice.

Q1: SAMPLE AN INTERESTING CHAIN (1 PTS)

Consider weights

$$w_1 = 0.2993 \qquad \text{and} \qquad w_2 = 0.7007,$$

and the following matrices A_1, A_2 and vectors b_1, b_2

$$A_1 = \begin{bmatrix} 0.4 & -0.3733 \\ 0.06 & 0.6 \end{bmatrix} \qquad \text{and} \qquad b_1 = \begin{bmatrix} 0.3533 \\ 0.0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} -0.8 & -0.1867 \\ 0.1371 & 0.8 \end{bmatrix} \qquad \text{and} \qquad b_2 = \begin{bmatrix} 1.1 \\ 0.1 \end{bmatrix}$$

Now we will define a Markov chain on $\mathbb{R}^2$ and we will denote our chain with $(x_k)_{k\geq 0}$. For this, consider two deterministic functions:

$$f_1(x) = A_1 x + b_1$$
$$f_2(x) = A_2 x + b_2.$$

Simulate the following Markov process

$$i_n \sim \text{Discrete}(w_1, w_2)$$
$$x_{n+1} = f_{i_n}(x_n)$$

for $1 \leq n \leq N$ where $N = 10000$. You can use $x_0 = [0, 0]^\top$. Plot a scatter plot of the Markov chain. For a pretty plot, you can use the following plotting function

```
plt.scatter(x[0, 20:k], x[1, 20:k], s=0.1, color = [0.8, 0, 0])
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['bottom'].set_visible(False)
plt.gca().spines['left'].set_visible(False)
plt.gca().set_xticks([])
plt.gca().set_yticks([])
plt.gca().set_xlim(0, 1.05)
plt.gca().set_ylim(0, 1)
plt.show()
```

As you can see, I chose `burnin` as 20 iterations here. Be careful about matrix products in Python: `A*x` **is not performing matrix product**, you should use `A@x` where A is a 2×2 matrix and x is a 2×1 vector. Try to simulate from this system until you see a nice picture.

Q2: Sample a state-space model (4 pts)

In the lectures, we have talked about state-space models. Differently than a Markov process, in state-space models you have an *underlying* and *hidden* Markov process and some observation sequence. This question asks you to simulate synthetic data from such a process. This is required for pure *simulation* tasks.

Simulate a Gaussian time-series corrupted by noise (2 pts)

Simulate data from the following model

$$x_t | x_{t-1} \sim \mathcal{N}(x_t; a x_{t-1}, \sigma_x^2),$$
$$y_t | x_t \sim \mathcal{N}(y_t; x_t, \sigma_y^2).$$

For reproducibility, choose $x_0 = 1$ (instead of sampling from a prior μ_0). Choose $a = 0.9$, $\sigma_x = 0.01$ and $\sigma_y = 0.1$[6]. Simulate this system $T = 100$ time steps. Plot your x and y in the same graph with different colors and give some examples what the model output can model in the real world.

Develop a stochastic volatility model and simulate (2 pts)

This question is intentionally vague: I would like to be able to develop a model of a phenomenon that is described to you. I will give fewer details[7].

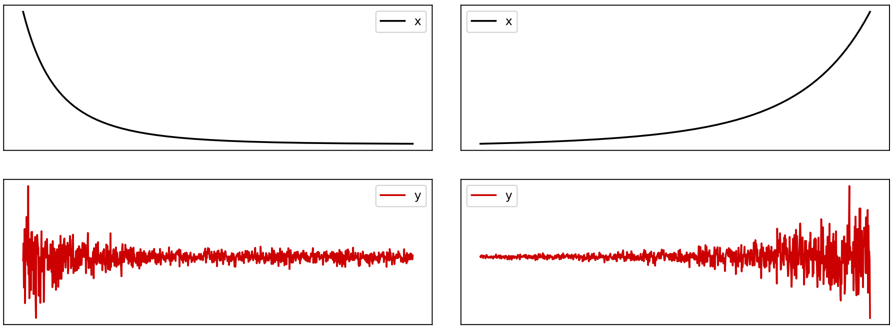
Imagine you are asked to develop a *volatility* model. Intuitively, volatility is a hidden quantity (that nobody knows about) which controls the variance of the observed stock (or option) prices. If the volatility is high, the stock prices will have high variance, if the volatility is low, the prices will have lower variance. So what you need to do essentially is:

- To define a Markov transition kernel to model the volatility variable, defined it as x_t (similar to above). Note that a naive Gaussian kernel would give you negative values and is therefore not suitable. You can use a Gaussian kernel but then you need to figure out how to use it modelling variance.
- Define your likelihood – this is more clear. If higher volatility (x_t) means higher variance in observed prices (y_t), how can you model it?

[6] Please note that these are standard deviations.
[7] Again, not because I'd like this to be hard – but I'd like you to use your creativity.

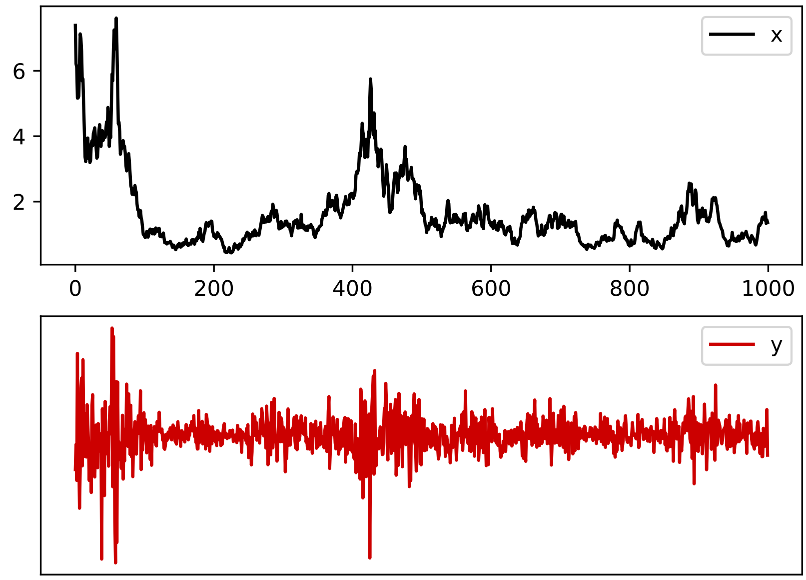
- Simulate data and show your results meaningfully model volatility.

As a sanity check, your model should exhibit the following behaviour if you choose x as a decaying or growing system.



Two examples of a simulated data from a volatility model. You can see that as x decays, the variance of y decays – or if x grows, the variance y grows (it becomes more erratic).

However, this is **not** the only figures I'm asking. The model is more important. Hopefully you will be able to generate some realistic data from your model, such as below.



You are free to search the literature for volatility models and use any model you wish in published literature (only state-space models, not more general ones!). As I said, there is no **correct** answer to this question, I will grade it with respect to your understanding of developing a model.