MATH3024 Complex Systems 2021 Project
DEPT. OF MATHEMATICS AND STATISTICS
The University of Western Australia

There are two parts for this project.
Questions 1 and 2 are compulsory.
You are free to choose ONE part of question 3 to work on.

Note: There are two submissions related to this project.
The interim report is worth 10%. Due date: **5pm 24 Sept.**.
The interim report should contain:
Your work so far on Q1 & Q2 as well as a research plan for the additional question.
The final report together with your source code (including comments) make up the remaining 40%. Due date: **5pm 22 Oct.**
The source code with comments (for all questions) contributes 15%.
The report is worth 25%. 20% are given for the correctness of the results.
The remaining 5% are for presentation and spelling.
The reports word count should stay below 3,000.

1. (a) Explain and comment the code dynsys2021.py.
   (b) Extend the code so that you have two systems coupled with each other (use a one-to-one coupling of the x-coordinate, i.e., $x \leftrightarrow x$, and couple the difference of the coordinates into the systems). Explore the synchronization for varying coupling strength $\sigma \in [0.0, 10.0]$.
   (c) Try following additional permutations of coupling the two dynamical systems: y to y, and z to z. (Hint: you may need to change/extend the coupling strength interval.)
   (d) Using the coupling of 4 to x components (as in (b)), explore how the synchronization regime changes for additive Gaussian white noise in the coupling force. Vary the noise intensity $D = [0.05, 0.2]$ in increments of 0.05.
   Note: If the dynamical system is $\dot{x} = f(x) + c(y - x)$ the additive case would be: $\dot{x} = f(x) + c(y - x + \sqrt{D}\xi)$, where $\xi \sim N(0, 1)$.
   Note: Stochastic differential equations should be simulated using an Euler-Maruyama algorithm.
   (e) Implement a Pecora-Carroll drive-response synchronization setup using this system and explore the synchronization using (i) x-driving, (ii) y-driving, and (iii) z-driving.

20 mks

2. For parts a), b), and c) include interpretations of each plot in the context of the network model parameters and network connectivity.

   (a) Consider the Erdös-Renyi $G(n, p)$ random graph model with $n = 300$ vertices, $p \in [0.0, 1.0]$. Generate plots of assortativity, transitivity, centralities, entropy of degree distribution, diameter, percolation (size of giant component) with respect to p and average degree $\langle k \rangle$.

   (b) Consider a small world (Watts-Strogatz) random graph model with $n = 300$ vertices, initial degree $k = 4$, and rewiring probability $p \in [0.0, 1.0]$. Repeat the calculation of the network statistics in Part a) but generate the plots with respect to the rewiring probability.

   (c) Consider a Barabasi-Albert random graph model with $m = 6$ and $n = 300$ vertices, and call it G_0. Study the rewiring processes "Maximal assortative case I", and "Maximal disassortative case I" as described in the paper [S. Zhou and R. J. Mondragón, "Structural constraints in complex networks," *New Journal of Physics*, **9**, 173 (2007).].

      i. Rewire G_0 using double edge swaps to create as maximally assortative network as possible, call it G_a. (A plot of assortativity verses number of swaps may be useful here.)

      ii. Rewire G_0 using double edge swaps to create as maximally disassortative network as possible, call it G_d. (Again a plot of assortativity verses number of swaps may be informative.)

      iii. Visualize G_0, G_a, and G_d.

      iv. For G_0, G_a, and G_d add one vertex with degree $k = 1$ to each of them assigning the edge at random to an existing vertex. Calculate the transitivity and assortativity of the new networks. Reset G_0, G_a, and G_d and now add a degree $k = 2$ vertex to each of them assigning the edges to existing vertices at random. Recalculate transitivity and assortativity. Repeat the above process for $k = 1, 2, \ldots, 100$ and generate plots of assortativity and transitivity with respect to k. Provide an explanation of what you observe and a possible interpretation in terms of the rich club of the resulting networks.

(d) Implement a basic sandpile model on a random 4-regular network of $n = 400$ vertices, and produce plots of the avalanche size distribution until saturation is reached. That is, all vertices have four or more grains.

   i. Introduce "dissipation" whereby 20% of vertices lose one grain and only redistribute three grains during an avalanche. Regenerate plots of the avalanche size distribution.

   ii. Now consider a random Barabasi-Albert network with $n = 400$ vertices and $m = 3$. Adapt the "dissipative" sandpile model above so that avalanches occur when a vertex of degree k has k or more grains but loses one grain during an avalanche and only redistributes $k - 1$ grains to connected neighbours. Summarize the results with plots of the avalanche size distribution.

20 mks

3. Each of the following parts contains a link to the Complexity Explorables of Dirk Borckmann. Have a look at the different projects I selected and choose ONE to work on. For all projects the aim is to implement the system yourself in Python. The code does not need to be interactive but you want to explore the full parameter space and summarize the found phenomena using parameter sweeps and present your results in figures.

   (a) Flock'n Roll:
   Collective behavior and swarming
   Observe the emergent behaviour and collective state the system aquires with respect to the parameters. Summarize this emergent behaviour with respect to the parameters using illustrative plots. The reference [I. D. Couzin, J. Krause, R. James, G. D. Ruxton and N. R. Franks, "Collective Memory and Spatial Sorting in Animal Groups," *J. theor. Biol.* (2002) **218**, 1–11.] will be useful to decide on good summary statistics. Sayama's `abm-swarm.py` may be a useful starting point.
   30 mks

   (b) I herd you!:
   How herd immunity works
   Investigate how vaccination not only protects you but also helps others using four network models. The disease model is SIS and numerically study how the epidemic threshold changes with respect to vaccine uptake, transmissibility of the disease, and model network structure. Sayama's `net-SIS.py` and related codes may be a good starting point.
   30 mks

   (c) Jujujajáki networks:
   The emergence of communities in weighted networks
   A mechanism that may explain community structures in social networks. Investigate how the structure of the networks, i.e., their connectivity, transitivity, degree correlations, community structure etc., depends on the model parameters. Networkx has community structure assignment routines, and the two cited papers in the explorable will be useful to decide on useful summary statistics and visualizations.
   30 mks