作业要求

Assessment

For the Modelling and simulation coursework, you will pick a dynamic system to study that represents an in-context problem and report on your analysis of that system in a report of up to 2200 words with accompanying code notebook in a GitHub repository.

Deadline: 09/12/2024 16:00 Submission on Blackboard.

The assessment criteria of the assignment are derived from the module's learning outcomes: Below, the **module learning outcomes** are in bold as top level bullet points, and how these form the assessement criteria of the module are the nested bullets.

- **Construct numerical and mathematical models that capture the key features of a design engineering problem.**
    - Define the scope of the system you are studying in terms of its own context.
    - Identify the variables, parameter, and assumptions of the system.
    - Iteratively increase the complexity of the system, starting with the smallest meaningful model, and increasing in complexity to capture new features, behaviours, or interventions.
- **Apply a range of numerical and computational methods.**
    - Implement your system model numerically for analysis and prediction.
    - Evaluate characteristic units of the system, e.g. characteristic time, for scaling / non-dimensionalisation.
- **Analyse algorithms for stability, accuracy, and computational complexity.**
    - Analyse the characteristics of the system in terms of fixed points, regions of phase space, and chaos.
    - Identify any bifurcations in the system and the system parameters that drive them.
    - Investigate the sensitivity to parameters and initial conditions with ensemble simulations.
- **Interpret simulation results and their implications in their wider design engineering context.**
    - Explain numerical results and how they describe the system being modelled.

    - Justify interventions in the context by making predictions of how your interventions will affect the state of your system.
- **Represent simulation results and data graphically for understanding, exploration, and communication.**
    - E.g. with systems diagrams, time evolution, phase space diagrams, bifurcation diagrams.
    - Use of animation or interactive plots for communication.
    - Present Wolfram notebook code in an clear and structured manner (link to repository).
    - Proper referencing of sources.

Theme – *Zombies!*

The theme for this run of the module is modelling a zombie apocalypse. The book Mathematical modelling of zombies (Online access from Library), is a collection of pedagogic papers investigating the zombie apocalypse as a dynamical system, usually with analogies to epidemiological modelling for real-world diseases. The starting point being a Susceptible-Infected-Removed (SIR) model, which has also been the starting point to model the COVID pandemic. Have a read of these, you are to make a report in the same style.

You can take your model in numerous directions as you wish, some suggestions being,

- Different strains of zombies and how that affects their deadliness.
- Spatial dependence of infection – can we run and hide?
- Seasonal variation – Winter is coming!
- Defensive strategies and interventions.
- Long term zombie strategy.

If zombies aren't your thing, you're welcome to choose a different theme, so long as you are able to meet the assessment criteria above. Discuss with me if you want to check you're heading in the right direction.

课程内容

Week 1 – Floating Point Numbers

Exercises

1. Install Wolfram Mathematica.
2. Is 0.1 + 0.2 greater or less than 0.3, does this depend on the size of the floating point representation?
3. Calculate the optimal step size and error for central difference first derivative. How does each quantity compare to the forward difference case?
4. How would you acually calculate sin and cos numerically?
5. Understand and explain how the Fast inverse square root (Quake 3) works. How would it look in 16 bit precision?

Resources

- Float Toy
- What Every Computer Scientist Should Know About Floating-Point Arithmetic
- Fast inverse square root (Quake 3)

More Resources

- https://www.coursera.org/learn/linear-algebra-machine-learning?specialization=mathematics-machine-learning
- https://www.coursera.org/learn/multivariate-calculus-machine-learning?specialization=mathematics-machine-learning
- https://www.linkedin.com/learning/hands-on-start-to-wolfram-mathematica/

Week 2 – Love, Predators, and Dynamic Systems

Matrix Diagonalisation

Exercises

Romeo and Juliet

- Implement the Romeo and Juliet system yourself in Mathematica
- Explore the parameter space of the matrix – Can you identify different regimes of the system. What do they correspond to in terms of the problem domain?
- Can you construct a parameter space diagram.
- Calculate the eigenvectors and eigenvalues of the matrix, draw them at the origin of the phase space diagram, how does this correspond to the dynamics?

Predator Prey

- Explore different variations of the predator-prey model.

- For each system, find the fixed points, and linearise about them - what are their characteristics.
- How much can be solved analytically – can you determine the regions in parameter space?
- Build a Mathematica toy for a Preditor-Prey model.

Week 3 - Animations, Pendula, Runge Kutta Formula

- Modify the particle in a bouncing box to have be on a TV screen with a 3:4 ratio as seen in The Office
- Observe how the single pendulum gains energy and changes modes between simple harmonic motion and wrap-around behaviours.
- Model a cannon-ball under the effects of air resistance.
    - The ball is subject to gravity F = mg, vertically
    - A good air resistane model has a force proportional to the absolute square of velocity opposing the motion, F = -γ |v|^2 v_hat. (v_hat is the unit vector in the velocity direction)
- Have the cannon ball elastically bounce off the floor.
- Add air resistance to the pendulum.
    - Using the model above
    - Using a linear in velocity model, F = -γ v
- Observe the effect of the unstable point in the pendulum by setting the initial condition to {π, 0} which is the inverted state.
    - Mathematically, it should stay here forever, does it?
    - what controls for how long it is upright?

Week 4 – Lagrangian Mechanics, Double Pendulum, Runge Kutta Fehlberg Method

- Look up and derive for the double pendulum system (Massless rods, massive ends)
    - The Euler-Lagrange equations.
    - From here derive the equations of motion.
- Find all stationary points of this system,
    - What states do these correspond to?

  - What is the total energy, E, of each of these states?
  - Linearise about each point, what is their character?
  - For the orbit solution, what is the frequency of oscillation? What is the eigenvector of these modes. How does this correspond to translation and vibration?
- Experiment with nearby initial conditions, how quickly do they diverge
- Can you find chaotic solutions? Can you find non-chaotic areas of parameter space?
- Read about Poincaré Map for the double pendulum.
- Can you reproduce this tweet? https://x.com/j_bertolotti/status/1411987574109913092
  - There's a higher resolution and Mathematica code for it on wikimedia commons: https://commons.wikimedia.org/wiki/File:Double_pendulum_predicting_dynamics.gif ,
  - See if you can work out what the code is doing. Bit of a warning - it takes a long time to fully run. Edit the table function to make it shorter Table[..., {t, 0, 1, 0.3}].
- Run the Mathematica code provided in this repo with an initial condition set at state = {-12.934153936353134, 10.59615330654864, -0.19807807300308317, -2.98557249710416} – what happens - how did I construct this?

Week 5 – Bifurcations

- Recall the week 2 predator-prey discussion.
- Play with the notebook for different bifurcation types.
- Run the catastrophe simulation, pausing mid-way to change parameters, then resume.
  - Do you observe the hysteresis effect?

Week 6 – Limit Cycles

- Perform dimensional analysis on the predator-prey systems given in *Danby*[1]. Then anaylse for fixed points character. Plot orbits, is there any special behaviour?

    - Starter system,
        - $\dot{r}=\alpha rr-prfrf$
        - $\dot{f}=-\delta ff+pfrrf$
    - Logistic Growth for Prey,
        - $\dot{r}=\alpha rr(1-rr\infty)-prf$
        - $\dot{f}=-\delta ff+prf$
    - Logistic Growth for Both,
        - $\dot{r}=\alpha rr(1-rr\infty)-prf$
        - $\dot{f}=\alpha ff(1-sfr)$
    - Saturatable Predation,
        - $\dot{r}=\alpha rr-pfrrpr+rp$
        - $\dot{f}=-\delta ff+pfrrpr+rp$
- We analysed in class the system of logistic growth for both species with saturable predation,
    - $\dot{r}=\alpha rr(1-rr\infty)-prf$
    - $\dot{f}=\alpha ff(1-sfr)$
    - See how far you can get with an analytic analysis of the fixed points.
        - How many are there? Can Mathematica find them all? Do they have critical values?
    - This system reduces to the "Catastophic" bifurcation system of last week when $\dot{f}\rightarrow 0$. Explore this correspondance.

[1] Danby, J. M. A. Computer Modeling : From Sports to Spaceflight, from Order to Chaos. Richmond, Va: Willmann-Bell, 1997. Print. Chapter 6