Contents

Project Rules

Some useful facts:

- The project is worth 60% of the marks for the Scientific Computing 2 module.
- 80% of the marks will be awarded for following the project outline given, and 20% for extensions to the project that you have generated.
- The deadline for Canvas submissions is 4$^{\text{th}}$ December, 23:59, and the final laboratory session is 12$^{\text{th}}$ November.
- You should submit:
  - a *well* documented user-friendly python program(s) which is (are) relatively computationally efficient.
  - a report of approximately 3000 words. Lengths greater than 3500 words will be penalised (further details are given below). The report should include, at the minimum:
    - the background physics
    - details of algorithms
    - tests of effectiveness, *e.g.* how do you know your code is giving you the right answer? How did you choose the parameters? How did you optimise it?
    - an analysis of the results and a comparison with the literature.

  {*Please remember to include your name in both the code and report. Do not include the full Python code in a .pdf report.*}
- Your program should run on any system with Anaconda installed. If you use any environments beyond NumPy, SciPy and MatPlotLib make it clear which were used. (*Hint: test that the code runs from a clean start before submitting it.*)

You may submit your report as a .pdf document, with the code submitted in a separate file or files, or else you may submit your report and code in the form of a Jupyter notebook. The report should contain the same elements whichever format it used.

Word counting:
Consistent word counting is difficult because different tools do this in different ways. MS Word and PDF-based counters will count absolutely everything, while tools like TexCount (used by Overleaf for its word count) exclude many elements of the document from the count (and in some cases don't have an option to override this behaviour). So whatever rules we defined, some people would have to correct the count their tool gives in order to be completely consistent.

So to be explicit, references, equations, headings, captions and text in appendices[1] are not

[1] Remember that appendices are only for supplementary information, and the report has to be complete and coherent without them

included in this word count[2]. Abstracts and footnotes are included. Since TeXCount's word counting will exclude footnotes by default, if you include any of these please add the line

```
%TC:macro \footnote [text]
```

in your LaTeX file (before the `\begin{document}` command is fine).

And finally should we be concerned in any way about your submission (for instance plagiarism) you will be required to have a viva.

[2]This is erring towards generosity in the word count. Plus it's impossible to make Overleaf count some of these things...

1 Random Walks: Diffusion Limited Aggregates

This project is about random walks, diffusion and fractals:
Fractal dimension is defined through the idea that:

$$M \sim V^d$$

where M is the total mass in volume V and then d is the ratio of the fractal dimension to the real dimension. For your calculations you need to count how many spheres are inside a certain radius, yielding the Mass versus Volume data.

Choose either the square or triangular lattice for this project.

1. **Numerical**
   (a) Write a program to create a random walk on your chosen lattice of given length.
   (b) Write a program to plot your random walk as a collection of symbols connected by lines.
   (c) Write a program to evaluate the distances between all pairs of points at a fixed number of steps apart along your walk.
   (d) Try to fit your 'distribution' of distances to a Gaussian distribution and find the best value for the variance.
   (e) Assess the assertion that the variance scales as N for N steps.

   (*If generating the walk takes a very long time you could consider writing the points to a file which you can then read back and analyse. If it does not, then don't worry.*)

   At this point the 'tired' student can quit! The more active individual can attempt one of the two following problems:

2. **Self Avoiding Random Walk**
   (a) For your chosen lattice, write a program to find a self-avoiding random walk.
   (b) It is often asserted that the variance of a self-avoiding random walk scales with a different power of n. What do you think of this assertion?

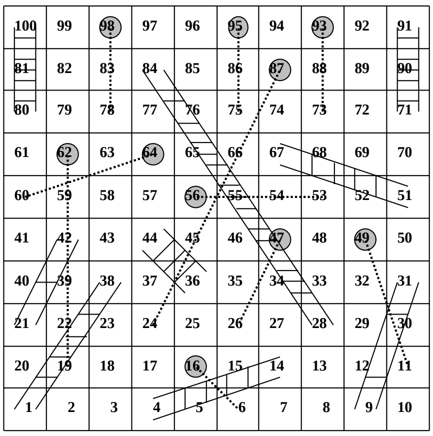
OR

3. **Diffusion limited aggregation**
   (a) Write a program to create a diffusion limited aggregate: Particles are introduced at a position in space and perform random walks. A 'seed' is placed somewhere else in the system and is 'sticky'. Each particle introduced wanders until it 'sticks', then another is introduced until it sticks and so on. The object that is created by this process is of interest to various people.
   (b) Write a program to picture your diffusion limited aggregate.
   (c) Write a program to fit a 'fractal' dimension to your aggregate.
   (d) Do you believe that your aggregate is fractal?

2 Snakes and Ladders

There is a popular board game for children: *Snakes and Ladders.*

- Find the average duration of a game and order the squares according to the average length of time until victory. Assess the spread in duration by calculating the *variance* of the duration.
- Your code should be able to cope with a random initialisation of the ”snakes and ladders” board-(*i.e.* number of squares, ladders and snakes and end conditions. You should consider this in terms of ”transfer matrices” and use this as the basis of your algorithm.

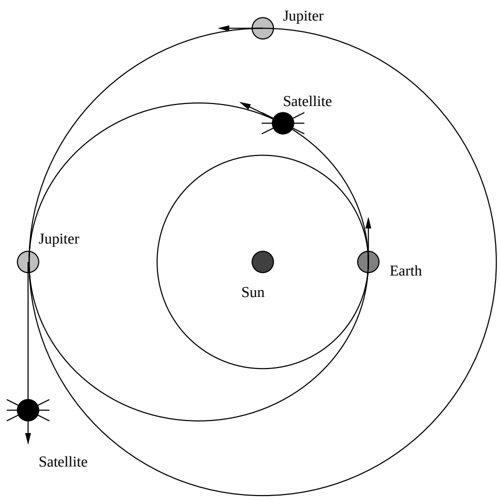


Note that the second task is initially very mathematical, relying on you *formalising* the problem into mathematics.

3 Satellites beyond Jupiter

This project addresses the issue of how to use the existence of the planet Jupiter to save NASA money. A satellite is made on Earth and then needs to be transported to its eventual 'home'. At first sight, one might calculate the difference in potential energy between the intial and eventual positions of the satellite and then provide that in kinetic energy, at great expense to NASA. Remind yourself of the 'expensive nature' of payloads and why saving at 'blast-off' is crucial by reading 'Payloads.pdf'. Although with only two bodies there is no way to avoid the payment in potential energy, with three bodies, the satellite can gain kinetic energy created from the potential energy between the other two. It is this effect that you will investigate: In simple terms, the satellite can pick up kinetic energy by 'bouncing off' Jupiter.

The basic picture is as depicted:



In order to minimise the expense, the first task is to decide *when* to launch in order to achieve close approach to Jupiter with the minimum orbit depicted. The next task is to find out the best configuration for the satellite and Jupiter when they meet, in order to achieve the maximum impulse to the satellite. The final task is to understand in simple terms what the basic concepts are which solve the problem.

1. **Analytical** With the help of 'TwoAndThreeBodyProblems.pdf', formulate the problem of *when* to launch the satellite, based upon the radii of the orbits of Earth and Jupiter, assuming that both planets have *circular* orbits and ignoring *all* potentials except that of the sun.

   Provide the equations of motion for the satellite under the assumptions that; the mass of the satellite is negligible in comparison to planetary masses, the Earth's gravitational field may be neglected and that all motion is coplanar.

   Should you launch the satellite in phase with the Earth's orbit or opposed to it?

2. **Numerical** Using a Runge-Kutta algorithm, write a program to solve the two body problem for the satellite's motion from Earth to Jupiter. Assess your calculations using the exact analysis provided in 'TwoAndThreeBodyProblems.pdf'. Now extend your calculation to incorporate Jupiter's gravitational potential, paying special attention to *real* conservation laws for the chosen limit and approximate conservation laws which would be true in the absence of Jupiter. You will need to have an algorithm to provide an automatic step-length control: The use of two Runge-Kutta algorithms to assess the error is a very efficient method. Pay particular attention to picturing the answer.

3. **Investigation** Devise a strategy for giving the satellite the *biggest* kick from Jupiter's gravitational potential. Does the satellite approach 'dangerously close' to Jupiter? Can the satellite escape from the Solar system? Provide a simple argument to predict the maximum impulse from Jupiter, and the corresponding minimum planetary orbit to achieve escape from the solar system.

4. **Data**

Gravitational Constant	$0.667 \times 10^{-10} m^3 kg^{-1} s^{-2}$
Mass of the Sun =	$0.1984 \times 10^{31} kg$
Mass of the Earth =	$0.5976 \times 10^{25} kg$
Mass of Jupiter =	$0.1903 \times 10^{28} kg$
Radius of the Earth's Orbit =	$0.1495 \times 10^{12} m$
Radius of Jupiter's Orbit =	$0.7778 \times 10^{12} m$
Radius of the Earth =	$0.6368 \times 10^{7} m$
Radius of Jupiter =	$0.6985 \times 10^{8} m$

5. **Additional Resources** The following notes, from MIT, are quite long but provide a clear description of some of the theory here:// http://web.mit.edu/8.01t/www/materials/modules/guide17.p http://web.mit.edu/8.01t/www/materials/modules/guide17Appendix.pdf//

   Update, Nov 2019: Those links are broken. I'm leaving them in case this is temporary, but there is a very detailed Astrodynamics module here:

   https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-346-astrodynamics-fall-2008/lecture-notes/

   That said, the mechanics covered in our own Classical Mechanics modules should be sufficient to enable you do set up this problem correctly.

4 Strange Attractor

In this project the computer will be employed to investigate a *non-integrable, dissipative* dynamical system. The system is a forced, damped, harmonic oscillator:

$$\frac{d^2\theta}{dt^2} + \nu\frac{d\theta}{dt} + \sin\theta = T\sin(2\pi ft)$$

where f is the forcing frequency, T is the forcing amplitude and ν is the friction coefficient.

1. **Analytical**
   Solve the integrable system:
   $$\frac{d^2\theta}{dt^2} + \nu\frac{d\theta}{dt} + \theta = T\sin(2\pi ft)$$
   which corresponds to small oscillations of the real system, for its motion given general starting conditions: $\theta(t = 0)$ and $\frac{d\theta}{dt}(t = 0)$. What is the *attractor?* (The steady-state of the oscillator once any initial transients have decayed.) A Poincare-section may be defined by taking snapshots of the state of the system:
   $$\left(\theta(t_n), \frac{d\theta}{dt}(t_n)\right)$$
   at times $t_n = \frac{n}{f}$, viz at the same time in each forcing cycle, and then overlaying them as points on a two-dimensional surface. What would be expected for such a plot from your attractor?

2. **Numerical**
   Employing the Runge-Kutta integrate the following differential equations:

   (a) $\ddot{y} = y$, subject to $y(0) = 1 = \dot{y}(0)$ for $t \in (0, 10)$.

   (b) $\ddot{\theta} + \sin\theta = 0$, subject to $\theta(0) = 0$, $\dot{\theta}(0) = 2$ for $t \in (0, 10)$.

   (c) $\ddot{\theta} + \nu\dot{\theta} + \theta = T\sin Ft$, for user defined boundary conditions.

   (d) $\ddot{\theta} + \nu\dot{\theta} + \sin\theta = T\sin Ft$, for user defined boundary conditions.

   Use (a) to verify your programs and to assess the Runge-Kutta technique. Use (b) to comment on the role of non-linearity in the dynamics. Verify your analytic calculations using (c) and provide the previously defined Poincare-section as a sequence of two-dimensional points. (You will have to wait for the transients to decay.)

3. **Investigation** Analyse the Poincare-section of the system's attractor by fixing ν in the *not* over-damped regime and increasing T until the attractor becomes non-trivial. By sitting on a point of the attractor-section and counting the number of other points as a function of range from the initial point, assess the dimension of the attractor. By using the 'box-counting' construction, assess the dimension of the attractor. Is the attractor fractal? (Ie: Is the attractor a *strange* attractor?)

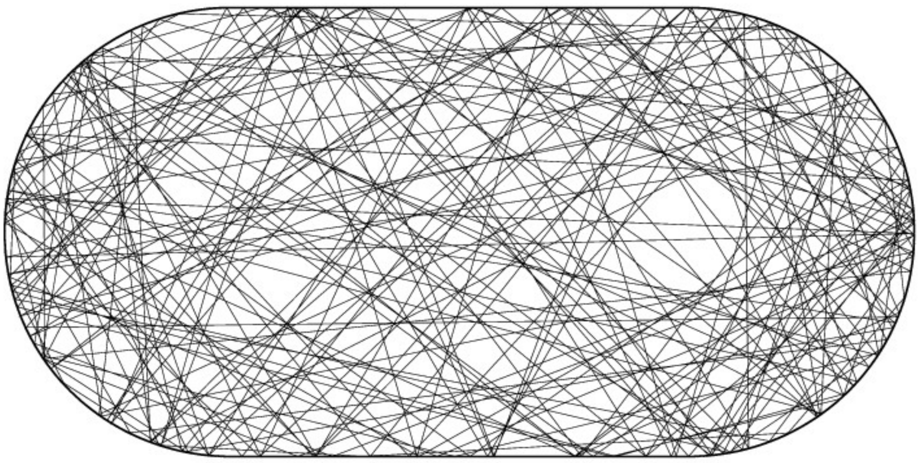
5 Mathematical Billiards

In this project the "classical" mathematical game of billiards is to be investigated. The billiard is a point particle which you initialise with a velocity (assume no dissipation). The ball then collides with the walls of the system under the assumption that the angle of incidence and reflection are equal.

Your task is to generate and investigate the trajectories that are possible, highlighting any special ones and categorise them with help from the literature (you may find a general book on chaos is useful).

You should repeat your study using an elliptical geometry - how does the system differ?

It is then up to you to consider further geometries - a possible one would be the stadium billiard which has two semi-circular caps attached to a rectangular section.



In order to identify and quantify chaotic behavior you should plot and analyse the trajectories in a suitable phase space (i.e. do not only plot the position of the ball as shown above but find other parameters, such as velocity, angles, ...). A chaotic system can produce fractal images when presented like this!

6 Forest fire model

In nature large amplitude events *e.g.* earthquakes measuring 8 on the Richter scale, stock market crash are rare. The question is whether these events were caused by a special set of circumstances or are they part of a pattern of events that would have occured without external intervention? The idea of *self-organised criticality*(SOC)[1] is that it is often possible to demonstrate that these events are part of a distribution of events. Thus if s is the magnitude of an event then the system is critical if $N(s)$ follows a power law:

$$N(s) \sim s^{-\alpha}$$

which is scale invariant.

1. **Analytical** Understand what is meant by scale invariance and show analytically that this is not what you expect if you combine a large number of independently random events.

   An example of a SOC system is the forest fire model. Consider trees on a two dimensional grid, which sites can have a live tree, a burning tree or an empty site.

2. **Numerical**Write a code that will generate this idealised forest and decide on an update regime which will set fire to the trees (does it depend on the neigbouring trees, how?). Update the whole system in "one go"(synchonously).

3. **Investigation** Once you are in a critical regime (how will you define this?) - you should analyse the area and circumference dependence of your clusters of live trees - and see if there are other parameters that you can use to parameterise your system.

References

[1] H.J.Jensen, *Self-organized criticality : emergent complex behaviour in physical and biological systems*, CUP 1998