Contents

Project Rules

Some useful facts:

- The project is worth 50% of the marks for the Computational Modeling of Physical Systems 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 WebCT submission is 17$^{\text{th}}$ January 4pm. The final laboratory session is 13$^{\text{th}}$ December.
- You should submit
  - a *well* documented user-friendly program(s) in .cpp format which is (are) relatively computationally efficient.
  - a document of approximately 3000 words (more than 3500 will be penalised, text in appendices is not included in this word count) in pdf format which 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 c++ code in your report*}
- Your program should run on unix g++, but should be portable
- for graphics in 2D/3D use Matlab, gnuplot or similar
- you may use GSL, but make sure that we know how to compile and build your programs

Please be considerate in terms of cpu on the servers with respect to other users and also consideration in how much demonstrator time you occupy. Marks will be deducted for people who are unreasonable on either of these counts (you will be warned, if this is likely).

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

1 Angular distribution of $e^+e^- \mapsto Z^0, \gamma \mapsto \mu^+\mu^-$

1. **Introduction**

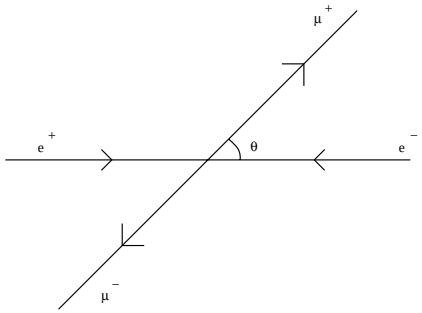
The differential cross-section for the reaction

$$e^+e^- \mapsto \mu^+\mu^- \tag{1}$$

as measured at the LEP accelerator complex is predicted to be of the form

$$\frac{d\sigma}{d\cos\theta} \propto 1 + a_1 \cos\theta + a_2 \cos^2\theta \tag{2}$$

where θ is the angle between the incoming positron and the outgoing μ^+:



The values of the parameters a_1 and a_2 are predicted by the standard electroweak theory; for example $a_1 \cos\theta$ is a parity-violating term resulting from interference between virtual Z^0 and γ intermediate states. Conversely, measurements of these parameters can be used to constrain the theory.

The object of this project is to determine how many 'events' (i.e. occurrences) of the form (1) are required to measure a_1 and a_2 to given accuracies. This type of calculation can help experimenters decide how much machine time will be needed to carry out a significant measurement.

2. **Outline procedure**

In outline, the procedure consists of using a random number generator to generate a given number of events of the form (1). These events are then treated as if they had been recorded in the actual experiment. The interval $-1 \leq \cos\theta \leq 1$ is divided into a number of equal 'bins' and the number of events in each bin is counted. Standard statistical theory is used to estimate the error in each number. A weighted least-squares-fit is then used to estimate a_1 and a_2 from these artificial data. This fit also provides errors on these parameters. It is assumed that the parameters are already known well enough for the purpose of estimating these errors although precise determination of a_1 and a_2 must of course await the real experiment. Any programs developed to analyse the 'Monte Carlo' data will still be useful when genuine data are available.

Several important computational techniques of general applicability will be encountered in the course of the project.

3. **Generation of Events**

   All program libraries contain a routine providing (pseudo) random numbers y distributed uniformly in the interval $[0, 1]$. In the present case values of $x = \cos\theta$ are required, distributed according to the probability density:

$$\rho(x) = \frac{3(1 + a_1 x + a_2 x^2)}{2(3 + a_2)} \tag{3}$$

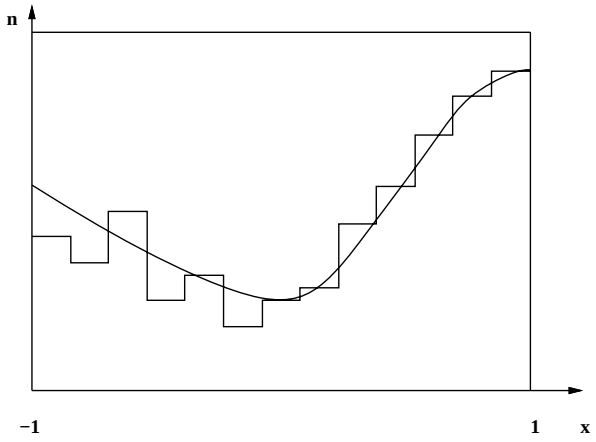
   which is the normalised version of (2):

$$\int_{-1}^{1} dx \rho(x) = 1$$

   Employing the mathematics developed in 'Maths 22', write a program to generate random numbers distributed according to (3) (you will need to call NAG routine G05CAF and solve equations with the help of 'Maths 16').

4. **Checking the first two steps**

   Using $a_1 = 0.5$ and $a_2 = 1$, generate about a thousand 'events'. By slicing the interval $(-1, 1)$ into about twenty equal intervals or *bins*, count the number of events in each bin. Plot a histogram of the results as depicted, and overlay the *scaled* exact distribution using NAGGraphics.



5. **Weighted Least-Squares Fit**

   The values of n displayed in the histogram must be fitted to a function of the form:

$$n_i = c_0 + c_1 x_i + c_2 x_i^2$$

   where x_i is the value of x at the centre of the i'th bin. Using the mathematics developed in 'Maths 23', together with the idea that the variance of the error at each point is expected to scale as $< \epsilon_i^2 > \sim n_i$, find a least squares fit to your data and assess the errors.

   Note: Professional numerical analysts (see for example the NAG manual) assert that least-squares problems should not be solved in this way. However there does not seem to be a suitable alternative routine in the NAG library. In the present case, rounding errors in accumulating A and **B** should not be over-large. The three parameters are strongly over-determined by at least 20 bins so most physicists would be happy to use the present method.

6. **Results**

   Compute $a_1 = c_1/c_0$ and $a_2 = c_2/c_0$ and their errors and compare with the input values. Re-draw the histogram showing the data and overlay the theoretical curve using the fitted values of a_1 and a_2. Alternatively plot the new curve on top of the previous plot.

   Repeat the whole process with N increased to say 4000 so that the dependence of the errors on N can be deduced. (In fact this should already be clear from 'Maths 23'.) It may be advisable to increase the number of bins correspondingly. How many events would be required to determine a_1 and a_2 to a relative accuracy of 1% ?

7. **Programming Considerations**

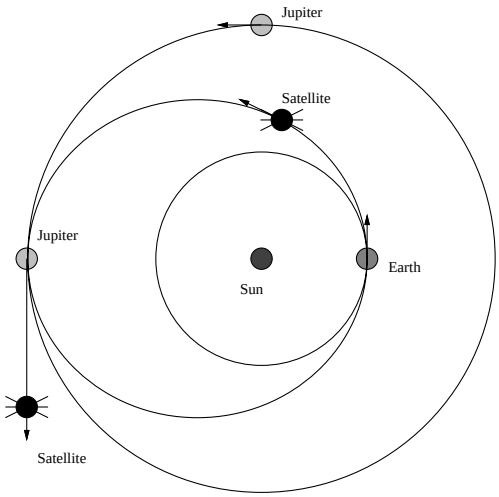
   To give the program a clear structure different subroutines should be used for different parts of the procedure. Use of separate subroutines should be considered for the following features:
   (a) generation of events;
   (b) solution of implicit equation;
   (c) allocation of events to bins;
   (d) plotting histogram;
   (e) least-squares-fit;
   (f) analysis of results.
   You may even prefer to write several programs!

   It will also be helpful to define functions for computing $\rho(x)$ and $\int^x dx'\rho(x')$ used in 'Maths 22'.

2 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 'Maths28'. 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 'Maths27', 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 'Maths27'. 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 =	$0.6985 \times 10^{8} m$

3 Payloads

This project involves a rudimentary model for firing a satellite out of the Earth's gravitational field using a 'rocket propulsion system'. The basic physical ideas are developed in 'Maths28', where the problem is mapped onto:

$$\frac{d^2\mathbf{s}}{d\tau^2} = \frac{t_b}{t_g^2}\left[\frac{t_c\hat{\mathbf{U}}}{1-\tau} - \frac{t_b\mathbf{s}}{|\ \mathbf{s}\ |^3}\right]$$

where:

$$t_b = \frac{m_0}{r}$$

is the burn-time that would exhaust the entire rocket, mass m_0, burned at a rate r,

$$t_c = \frac{|\ \mathbf{U}\ |}{g}$$

is the critical time which must be greater than the burn time, t_b, in order to force take-off, in terms of the expulsion velocity of the fuel, $\mathbf{U}$, and the gravitational acceleration at the Earth's surface, g,

$$t_g = \left[\frac{r_0}{g}\right]^{1/2}$$

is the natural gravitational time-scale, and corresponds to the period that a mass would have if it oscillated through the Earth under the action of the Earth's gravitational field. The time t_c is a function of the style of fuel and may be assumed 'constant', while the burn-time, t_b, is a variable to be investigated. The *payload* is the fraction of the mass of the rocket which actually ends up in space:

$$\frac{m_{final}}{m_0} = 1 - \tau_{final}$$

where τ_{final} is the time at which the burn ceases. The final 'energy' of the satellite after the burn is:

$$E = \frac{1}{2}\ |\ \frac{d\mathbf{s}}{d\tau}\ |^2 - \frac{t_b^2}{t_g^2}\frac{1}{|\ \mathbf{s}\ |}$$

which must be positive for the satellite to escape from orbit.

1. **Analytical** With the help of 'Maths28' and going to the limit $\tau_{final} \mapsto 1$, show that escape from orbit is always possible for a sufficiently small payload.

2. **Numerical** Using a Runge-Kutta algorithm, write a program to solve for the motion of the rocket under a uniform gravitational field, using and overlaying the exact solution to verify your calculation. Now extend your calculation to the problem with the Earth's gravitational field varying with height. Extend your calculation to a *variable* choice of step length, and assess the result.

3. **Investigation** Investigate the payload achievable for a given burn-time, which achieves escape velocity. What sort of payload could escape from Jupiter?

4. **Data**

Gravitational Constant =	$0.667 \times 10^{-10} m^3 kg^{-1} s^{-2}$
Mass of the Earth =	$0.5976 \times 10^{25} kg$
Radius of the Earth =	$0.6368 \times 10^{7} m$
Earth's Surface Gravitational acceleration =	$0.9829 \times 10^{1} m s^{-2}$
The gravitational time-scale, t_g=	$0.8049 \times 10^{3} s$

4 Contact Interactions between quantum particles in a box

The many-body problem often starts out with an investigation of how only two particles interact. In this project the behaviour of two quantum *bosonic* particles trapped in a one-dimensional infinite square well but subject to an extremely short-range or *contact*-interaction may be studied. The Hamiltonian is:

$$H = \frac{\hat{p}_1^2}{2m} + \frac{\hat{p}_2^2}{2m} + V\delta[x_1 - x_2]$$

and there is a restriction that the wavefunction vanishes on the boundary of the box:

$$\psi(x_1, x_2) \quad \Rightarrow \quad \psi(x_1, 0) = 0 = \psi(x_1, a) \quad \psi(0, x_1) = 0 = \psi(a, x_2)$$

In this project we use the *relaxation method*. This first converts a (possibly partial) differential equation into a set of linear equations $\mathbf{Ay} = \mathbf{b}$ (one for each point of a grid) using finite difference approximations to the derivatives; for example,

$$\frac{d^2y}{dx^2} \approx \frac{1}{\delta^2}[y(x-\delta) + y(x+\delta) - 2y(x)]$$

so the equation $d^2y/dx^2 = \lambda y$ can be approximated by

$$y_{i+1} + y_{i-1} - (2 + \delta^2\lambda)y_i = 0$$

where $y_k = y(k\delta)$. It then solves this set of equations by iteration, starting from some initial guess at the solution. The *successive over-relaxation* iteration is

$$y_i \to \frac{\omega}{A_{ii}}(b_i - \sum_{j \neq i} A_{ij}y_j) + (1-\omega)y_i$$

where i repeatedly runs through the grid points, and $0 < \omega < 2$ is a parameter; for example, for $d^2y/dx^2 = \lambda y$,

$$y_i \to \frac{\omega(y_{i+1} + y_{i-1})}{2 + \delta^2\lambda} + (1-\omega)y_i$$

The special case $\omega = 1$ is called *Gauss-Seidel*, but convergence is often faster with $\omega > 1$.

1. **Analytical** Solve the single particle problem:

$$H = -\frac{d^2}{dx^2} + 2\alpha\delta[x] \tag{1}$$

where the wavefunction vanishes on the boundary of the box:

$$\psi(x) \quad \Rightarrow \quad \psi(-1) = 0 = \psi(1)$$

showing that the energy satisfies:

$$\frac{\sqrt{\epsilon}}{\tan\sqrt{\epsilon}} = -\alpha$$

This problem corresponds to the motion of one of the particles whilst the other is fixed. Write a program to find ϵ given α (e.g. using Newton-Raphson).

2. **Numerical** Write a program to solve (1) numerically employing relaxation. The eigenvalue must be *simultaneously* obtained, and this can be accomplished by demanding that an appropriate *scale* is conserved:

$$\int_{-1}^{1} dx \mid \psi(x) \mid^2 \qquad or \qquad \int_{-1}^{1} dx \psi(x)$$

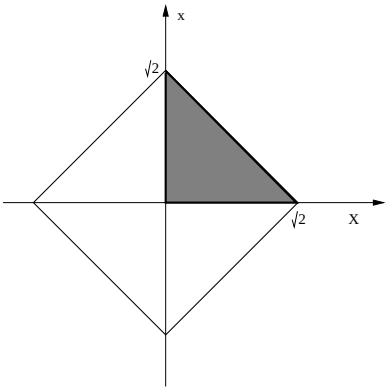
for example. Ensure the validity of your calculations by comparison with the exact results.

3. **Investigation** Rescale and reparameterise the interacting boson problem into the form:

$$H = -\frac{\partial^2}{\partial X^2} - \frac{\partial^2}{\partial x^2} + 2\alpha\delta[x]$$

where the boundary conditions become:

$$\psi(X,x) \quad \Rightarrow \quad \psi\left(-\sqrt{2}+x,x\right) = 0 = \psi\left(-\sqrt{2}-x,x\right) \quad \psi\left(\sqrt{2}+x,x\right) = 0 = \psi\left(\sqrt{2}-x,x\right)$$



Solve this problem using relaxation in the depicted region under the symmetry assumptions that the wavefunction is invariant under $X \mapsto -X$ and $x \mapsto -x$.

The simple extension of the above finite-difference approximation to two dimensions is

$$\nabla^2 \approx \frac{1}{\delta^2}[y(X-\delta,x) + y(X+\delta,x) + y(X,x-\delta) + y(X,x+\delta) - 4y(x)]$$

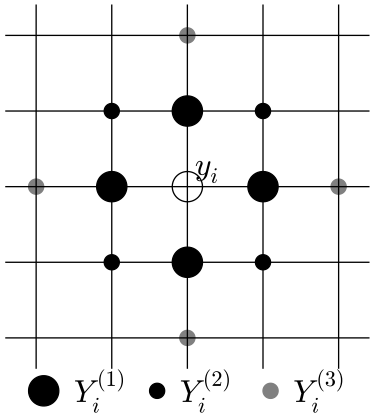
$$y(X,x) \to \frac{\omega[y(X-\delta,x) + y(X+\delta,x) + y(X,x-\delta) + y(X,x+\delta)]}{4+\delta^2\lambda} + (1-\omega)y(X,x)$$

but to obtain sufficient accuracy for this question you may need to use a higher-order method: defining the symmetric shell sums

$$Y_i^{(1)} = y(X_i-\delta,x_i) + y(X_i+\delta,x_i) + y(X_i,x_i-\delta) + y(X_i,x_i+\delta)$$

$$Y_i^{(2)} = y(X_i-\delta,x_i-\delta) + y(X_i+\delta,x_i-\delta) + y(X_i-\delta,x_i+\delta) + y(X_i+\delta,x_i+\delta)$$

$$Y_i^{(3)} = y(X_i-2\delta,x_i) + y(X_i+2\delta,x_i) + y(X_i,x_i-2\delta) + y(X_i,x_i+2\delta)$$



we have the finite-difference approximations

$$Y_i^{(1)} + \frac{1}{4}Y_i^{(2)} = 5y + \frac{3}{2}\delta^2\nabla^2 y + \frac{1}{8}\delta^4\nabla^4 y + O(\delta^6)$$

$$Y_i^{(1)} + \frac{3}{8}Y_i^{(2)} + \frac{1}{32}Y_i^{(3)} = \frac{45}{8}y + \frac{15}{8}\delta^2\nabla^2 y + \frac{3}{16}\delta^4\nabla^4 y + \frac{1}{96}\delta^6\nabla^6 y + O(\delta^8)$$

To use these, note that if $\nabla^2 y = \lambda y$ then $\nabla^4 y = \lambda^2 y$, etc., giving the iterations

$$y_i \to \frac{\omega(Y_i^{(1)} + \frac{1}{4}Y_i^{(2)})}{5 + \frac{3}{2}\delta^2\lambda + \frac{1}{8}\delta^4\lambda^2} + (1-\omega)y_i$$

$$y_i \to \frac{\omega(Y_i^{(1)} + \frac{3}{8}Y_i^{(2)} + \frac{1}{32}Y_i^{(3)})}{\frac{45}{8} + \frac{15}{8}\delta^2\lambda + \frac{3}{16}\delta^4\lambda^2 + \frac{1}{96}\delta^6\lambda^3} + (1-\omega)y_i$$

Relate the two limits $\alpha \mapsto 0$ and $\alpha \mapsto \infty$ to the non-interacting *fermion* problem. Investigate the wavefunction under both repulsive and attractive interactions. Depict the wavefunctions and plot the dependence on α of the ground-state energy.

5 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?)

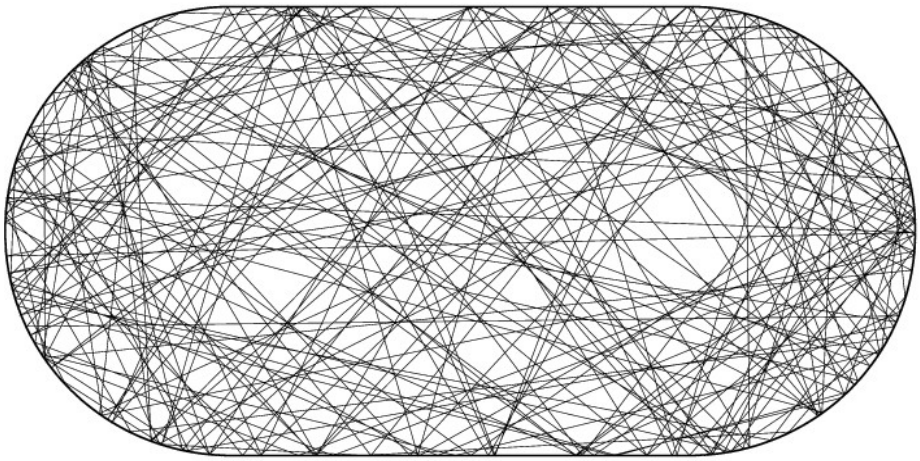
6 Mathematical Billiards

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!

7 Sand-pile 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.

   A common example of SOC is the sand-pile. If you keep adding grains to a pile of sand, at some point it will start displacing grains as you add more sand, but the size of these "avalanches" can be of any order of magnitude up to the size of the system.

2. **Numerical**

   (a) Write a code that will generate an idealised sand pile on a *lattice* in one-dimension. What rules will you use to update the neighbouring sites? Where will the sand be added? How will the sand be lost? How will you know you are in the critical regime?

   (b) Now generalise your system to a two dimensional grid. Does the lattice symmetry you choose matter?

3. **Investigation** Once you are in a critical regime you need to find how $N(s)$ varies, and consider how you are scaling the system size. Vary your rules and see how this affects your results. Is there an entropic type quantity you could measure?

References

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

8 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

9 Finite-Difference Time-Doman Simulation

This project involves looking at the Maxwell equations and how they evolve in time. At the end of this project you should have created a 1-Dimensional solver for the Maxwell equations, where you are able to inject a source signal into a 1D space and see how it evolves. Several problems will need to be overcome, like how to efficiently solve the equations and using appropriate boundary conditions.

1. To start with let us derive our main equations. Starting with the Maxwell equations in a vacuum,

$$\nabla \times \vec{E} = -\mu \frac{d\vec{H}}{dt} \tag{2}$$

$$\nabla \times \vec{H} = \epsilon \frac{d\vec{E}}{dt} \tag{3}$$

   (a) Take the equations above and expand into cartesian coordinates

   (b) Reduce these to a 1-dimensional set of equations assuming that the $\vec{E}$ and $\vec{H}$ only change in the $\hat{z}$ direction

You should now have two equations that state how the $\vec{E}$ and $\vec{H}$ are related to each other spatially and temporally. Therefore if we initially had a 1-D space where we specify the $\vec{E}$ and $\vec{H}$ values at a chosen spatial resolution we could calculate how the fields vary in time. This can turn into a complicated problem, especially when working in higher dimensions. However an alogorithm was developed to compute this efficiently and is called the Yee algorithm which makes use of spatially and temporally separating the field components and using a finite-difference approximation technique to solve the above equations.

2. Understanding the Yee alogorithm and how it uses the finite-difference approximation to differential equations is essential for programming this method correctly.

   (a) Discuss and depict how the Yee algorithm works

   (b) Derive the central finite-difference approximation. Discuss the physical reasoning as to why we use the central instead of the forward and backward approximations.

   (c) As with most numerical techniques there are stability conditions that if not satisfied result in unphysical results. Investigate any stability conditions that effect the 1D FDTD.

You should develop your own computer program in C++ to work as stated by the Yee algorithm. Your program should output the 1D space containing the 2 electomagnetic field components to a file that can be plotted and analysed.

As with any simulation you create you must validate it against a known result. You should pick some basic optical effect that can easily be tested in 1D, such as reflection or transmission from a dielectric interface.

FDTD is a very popular engineering and scientific tool and is documented all over the web, a quick internet search should return online videos, lecture courses and e-books. One useful e-book in particular can be found at http://www.eecs.wsu.edu/~schneidj/ufdtd/.

Some ideas for further study could be:

1. Lossy materials
2. Absorbing boundary conditions (1D Mur Boundaries)
3. Total-Field/Scattered-Field Boundaries
4. Higher-order spatial FD approximations

10 Detecting Black Holes

This project involves using various data analysis techniques to detect and extract properties of a signal buried in noisy data. While these techniques are used widely, for example in radar and voice recognition, we will consider a gravitational wave signal from two black holes colliding. Data files containing the data with a signal in it and the noise power spectrum of the detector will be provided. Extra necessary information will be provided in a suplimentary document.

1. To detect a signal that is hidden in noise we use a process called matched filtering. The idea being that we compare some measured data to a model signal, known as a template. To see how well they match we need to compute the inner product between the template and the data, defined in the frequency domain as

$$< a|b >= 4\mathcal{R}\int_0^\infty \frac{a(f)b^*(f)}{S_n(f)}df$$

where $S_n(f)$ is the noise power spectrum. The signal to noise ratio (SNR) is then defined as

$$\text{SNR} = \max_{\bar{\theta}} \frac{< d|h(\bar{\theta}) >}{\sqrt{< h(\bar{\theta})|h(\bar{\theta}) >}}$$

   (a) Describe the gravitational wave source and its parameters and describe how we are trying to detect these with laser interferometers.

   (b) Plot the data, noise PSD and an example template waveform.

   (c) Search over the chirp mass to find the where the SNR is greatest.

MCMC - Markov Chain Monte Carlo

If the SNR is large enough then we consider this a detection. Having detected a source we will want to estimate the physical parameters of the black hole binary system. As the signal is buried in noise we know that the maximum SNR will not necessarily correspond to the correct parameters. Instead we must map out the probability density function using Bayes theorem.

$$\begin{aligned} p\left(h(\bar{\theta}|d)\right) &= \frac{p\left(d|h(\bar{\theta})\right)p\left(h(\bar{\theta})\right)}{p\left(d\right)} \\ &= \frac{\text{Likelihood of data given model } * \text{ prior information about model}}{\text{normalising factor}} \end{aligned}$$

while we will give a relatively simple case here, if we are searching over many parameters, then this mapping becomes computationally intensive. For the full model of two black holes colliding there are 9 parameters to search over. We must instead turn to stochastic methods, in particular here we consider an MCMC algorithm.

2. Implementing the MCMC.

(a) Discuss how the MCMC algorithm works for mapping posterior density functions (PDF) when we use the Metropolis-Hastings step condition

(b) Implement the algorithm to map the PDF of the chirp mass

Some suggestions for further work

1. Search over a second parameter (η or distance)
2. Improve the stepping function in the MCMC