# Contents

# Project Rules

Some useful facts:

- The project is worth 50% of the marks for the Numerical 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 Canvas submission is 15$^{\text{th}}$ January 4:30pm. The final laboratory session is 11$^{\text{th}}$ 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, Python, 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 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. Order the points and write them to disc.

   (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.

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

1. **Introduction**

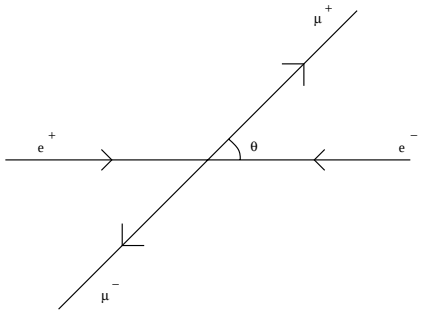
   The differential cross-section for the reaction

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

   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 $\theta$ is the angle between the incoming positron and the outgoing $\mu^+$:

   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 $\gamma$ 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:

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

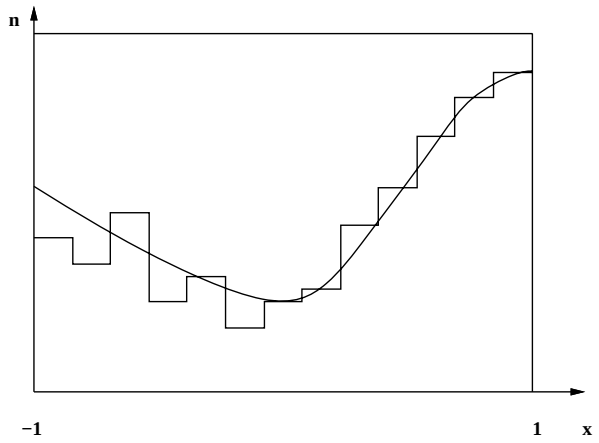
   which is the normalised version of (2):

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

   Employing the mathematics developed in 'Maths 22', write a program to generate random numbers distributed according to (3) (you will need to call a GSL routine 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.




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 assert that least-squares problems should not be solved in this way. 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**

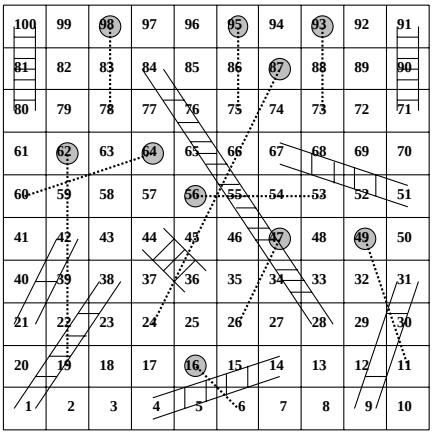
   To give the program a clear structure different subroutines should be used for different parts of the procedure. Use of separate subroutines should be considered for the following features:
   (a) generation of events;
   (b) solution of implicit equation;
   (c) allocation of events to bins;
   (d) 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'.

# 3 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.

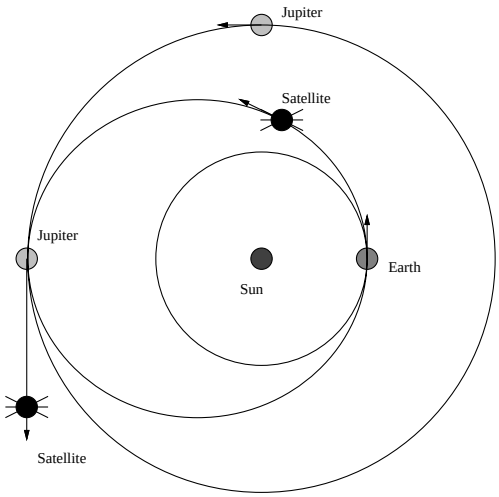



Note that the second task is initially very mathematical, relying on you *formalising* the problem into mathematics.

# 4 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**

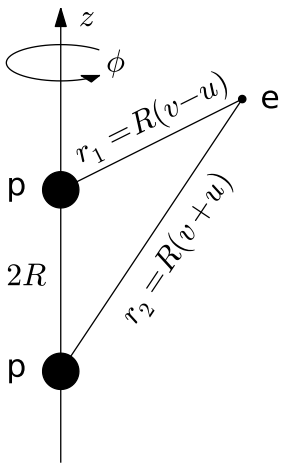
| | |
|---|---|
| Gravitational Constant | $0.667 \times 10^{-10} m^3 kg^{-1} s^{-2}$ |
| Mass of the Sun = | $0.1984 \times 10^{31} kg$ |
| Mass of the Earth = | $0.5976 \times 10^{25} kg$ |
| Mass of Jupiter = | $0.1903 \times 10^{28} kg$ |
| Radius of the Earth's Orbit = | $0.1495 \times 10^{12} m$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

# 5 The Hydrogen Molecule: $H_2^+$

In this project the energies and eigenstates of a single electron bound to a pair of protons is to be investigated. We consider the Hamiltonian for the electron motion at fixed proton positions (the Born-Oppenheimer approximation, valid because the proton is much heavier than the electron):

$$H = -\frac{\hbar^2}{2M}\nabla^2 - \frac{e^2}{r_1} - \frac{e^2}{r_2} + \frac{e^2}{2R}$$

where $M, e$ are the electron mass and charge (in $4\pi\epsilon_0 = 1$ units), $r_1, r_2$ are the electron-proton distances, and $2R$ is the proton-proton distance.




Using separation of variables in the coordinate system

$$v = \frac{1}{2R}(r_1 + r_2) \qquad v \in [1, \infty)$$

$$u = \frac{1}{2R}(r_2 - r_1) \qquad u \in [-1, 1]$$

$$\phi = \text{azimuthal angle} \qquad \phi \in [0, 2\pi)$$

the eigenstates can be written as

$$\psi(\mathbf{r}) = \frac{1}{\sqrt{(2\pi)}} e^{im\phi} \left[v^2 - 1\right]^{\frac{m}{2}} \left[1 - u^2\right]^{\frac{m}{2}} V(v)U(u),$$

where $m$ is an integer and $U, V$ satisfy

$$(v^2 - 1)\frac{d^2V}{dv^2} + 2(m+1)v\frac{dV}{dv} + \left[\left(\epsilon - \frac{\alpha}{2}\right)(v^2 - 1) + 2\alpha v + m(m+1) - \lambda\right] V = 0$$

$$(1 - u^2)\frac{d^2U}{du^2} - 2(m+1)u\frac{dU}{du} + \left[\left(\epsilon - \frac{\alpha}{2}\right)(1 - u^2) - m(m+1) + \lambda\right] U = 0$$

Here $\alpha = 2Me^2R/\hbar^2$, and $\epsilon, \lambda$ are separation parameters; the energy is $E = \hbar^2\epsilon/2MR^2$. (The details of this separation can be found in the file 'Maths31', but are not required for this project.)

1. **Analytical** Verify that the generalised Legendre polynomials ($\partial^m P_l(u)/\partial u^m$ where $P_l(u)$ is a Legendre polynomial) solve the equation in $U(u)$ for a particular choice of energy $\epsilon$.

   Verify that:
   $$V(v) \sim \exp\left(-\left[\frac{\alpha}{2} - \epsilon\right]^{\frac{1}{2}} v\right)$$
   asymptotically as $v \mapsto \infty$, and show that this solution can be exact for a specific choice of $\lambda$ and $\epsilon$.

2. **Numerical** Write a subroutine to solve for $U(u)$ (e.g. using Runge-Kutta) and use the generalised Legendre polynomials to test your program.
   Note: Show that the solutions must be either symmetric or anti-symmetric under $u \mapsto -u$ and use this idea to find the appropriate eigenvalue. Write a subroutine to solve for $V(v)$ and use the exact asymptotic solution to test your program.
   Note: Use the asymptotic solution:
   $$V(v) \sim A\exp\left(\left[\frac{\alpha}{2} - \epsilon\right]^{\frac{1}{2}} v\right) + B\exp\left(-\left[\frac{\alpha}{2} - \epsilon\right]^{\frac{1}{2}} v\right)$$
   and the idea that $A \mapsto 0$ corresponds to bound states as the appropriate boundary condition.

3. **Investigation** Develop a program to solve for the eigenstates of the non-relativistic $H_2^+$ molecule. Depict the wavefunctions in cylindrical coordinates $(z, \rho)$, where the $z$ axis is defined to pass through both protons. Find the ground-state separation of the two protons and also their separation in the first excitation.

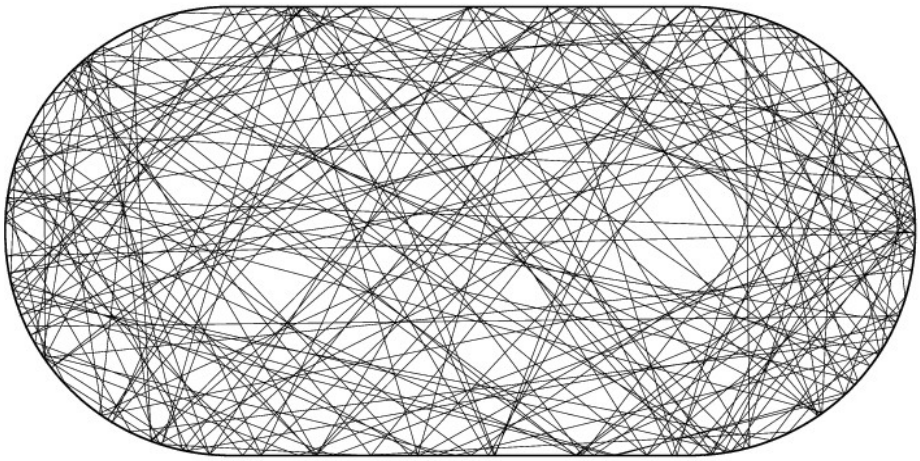
# 6 Mathematical Billiards

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-Domain 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,

$$\nabla \times \vec{E} = -\mu \frac{d\vec{H}}{dt} \tag{1}$$

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

   (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