Level 3 PHYS3561 Computing Project

Computing Project Booklet

2023/24

Contents

Part I

Condensed/Soft Matter Physics

Chapter 1

Silicon Bandstructure

1.1 Introduction

In this project a so-called plane-wave approach is employed to calculate the energies of the electron states in the simplest semiconductor material, silicon. As described the approach can also be employed for germanium and a form of tin called α-tin. All of these materials crystallise in a face-centred cubic (FCC) structure. A fairly simple extension of the method also makes it possible to deal with a whole range of III-V and II-VI semiconductors such as GaAs, ZnTe, etc., which have the same basic structure.

1.2 Background theory

As silicon is a periodic, crystalline material the crystal potential can be expressed as a 3D Fourier series:

$$V(\mathbf{r}) = \sum_{\mathbf{g}''} V_{\mathbf{g}''} e^{i\mathbf{g}''\cdot\mathbf{r}}$$

where the $''$ indexation on the reciprocal lattice vectors (RLVs) is introduced for later convenience. In general, RLVs are given by

$$\mathbf{g} = n_x\mathbf{a}^* + n_y\mathbf{b}^* + n_z\mathbf{c}^* \qquad n_x = 0, \pm1, \pm2, \ldots,\ n_y = 0, \pm1, \pm2, \ldots,\ n_z = 0, \pm1, \pm2, \ldots$$

with, in this case,

$$\mathbf{a}^* = \frac{2\pi}{A}(-1,1,1),\ \mathbf{b}^* = \frac{2\pi}{A}(1,-1,1),\ \mathbf{c}^* = \frac{2\pi}{A}(1,1,-1).$$

A is the FCC lattice constant of the crystal. For the FCC lattice the RLVs are then of the form

$$\mathbf{g} \ = \ \frac{2\pi}{A}(-n_x + n_y + n_z,\ n_x - n_y + n_z, n_x + n_y - n_z) \ = \ \frac{2\pi}{A}(n'_x, n'_y, n'_z)$$

where, because of the specific form of the RLVs, there are restrictions on the allowed combinations of integers n'_x, n'_y, n'_z. The reciprocal lattice of the original FCC crystal structure is body-centered cubic (BCC) so the RLVs in this case define a BCC lattice.

The wave functions of the electron states can be written in the Bloch form:

$$\psi_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}} u_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}} \sum_{\mathbf{g}'} a_{\mathbf{g}'}(\mathbf{k}) e^{i\mathbf{g}'\cdot\mathbf{r}} = \sum_{\mathbf{g}'} a_{\mathbf{g}'}(\mathbf{k}) e^{i(\mathbf{k}+\mathbf{g}')\cdot\mathbf{r}}$$

where $u_{\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{g}'} a_{\mathbf{g}'}(\mathbf{k}) e^{i\mathbf{g}'\cdot\mathbf{r}}$ has the same periodicity as $V(\mathbf{r})$ and $\mathbf{k}$ is a wavevector confined to the first Brillouin zone. The $'$ indexation has again been introduced for later convenience, to distinguish between that employed in the potential and wave function expansions. The electron energy, $E_{\mathbf{k}}$, and the wavefunction expansion coefficients, $a_{\mathbf{g}'}(\mathbf{k})$, can be found from the solution of the Schrödinger equation

$$\frac{-\hbar^2}{2me}\nabla^2\psi_{\mathbf{k}}(\mathbf{r}) + V(\mathbf{r})\psi_{\mathbf{k}}(\mathbf{r}) = E_{\mathbf{k}}\psi_{\mathbf{k}}(\mathbf{r})$$

where m is the electron mass and we have assumed that both the energy, $E_{\mathbf{k}}$, and potential, V, are in units of eV. Multiplying this equation by each of the $e^{-i(\mathbf{k}+\mathbf{g})\cdot\mathbf{r}}$ in turn and integrating over all space leads to a reformulation of the problem in matrix form with the following structure:

$$\begin{array}{c} \\ \mathbf{g} \\ \downarrow \end{array} \begin{array}{c} \mathbf{g}' \ \rightarrow \\ \\ \end{array} \begin{pmatrix} & & \\ & \frac{\hbar^2 |\mathbf{k}+\mathbf{g}|^2}{2me}\delta_{\mathbf{g},\mathbf{g}'} + V_{\mathbf{g}-\mathbf{g}'} & \\ & & \end{pmatrix} \begin{pmatrix} a_{\mathbf{g}1} \\ a_{\mathbf{g}2} \\ \vdots \\ a_{\mathbf{g}N} \\ \vdots \end{pmatrix} = E_{\mathbf{k}} \begin{pmatrix} a_{\mathbf{g}1} \\ a_{\mathbf{g}2} \\ \vdots \\ a_{\mathbf{g}N} \\ \vdots \end{pmatrix} \tag{1.1}$$

In the above matrix the 'kinetic energy' terms, $\frac{\hbar^2 |\mathbf{k}+\mathbf{g}|^2}{2me}$, appear only on the diagonal of the matrix where $\mathbf{g} = \mathbf{g}'$ and in each element of the matrix only a single Fourier component of the potential with $\mathbf{g}'' = \mathbf{g} - \mathbf{g}'$ is present. The eigenvalues of this matrix equation give the various $E_{\mathbf{k}} \equiv E^n_{\mathbf{k}}$ where we have now employed an additional band index, n, to indicate solutions with increasing energy. In theory, there are an infinite number of RLVs and $V_{\mathbf{g}''}$ corresponding to all possible n_x, n_y, n_z values but in practice, using something called a 'pseudopotential' (you may learn more about this in the Level 4 Advanced Theoretical Physics module but it is not essential here), it is possible to obtain reliable values of the energies in the first few bands using relatively few RLVs and $V_{\mathbf{g}''}$. In this case only the N lowest magnitude RLVs will be employed which leads to a finite $N \times N$ matrix eigenvalue problem. At a given wavevector $\mathbf{k}$ this can be straightforwardly solved using standard techniques to give eigenvalues corresponding to N electron energies in N energy bands and the associated eigenvectors corresponding to the wavefunctions. In equation (1) it can be imagined that $\mathbf{g}_1 = \mathbf{0}$ corresponds to the lowest magnitude RLV, that $\mathbf{g}_2, \ldots, \mathbf{g}_9$ correspond to the 8 next lowest magnitude RLVs of the form $\mathbf{g} = \frac{2\pi}{A}(\pm 1, \pm 1, \pm 1)$, and so on.

1.3 Bandstructure plots

By convention, the results of the solution of the eigenvalue equation (1) are displayed in the form of a *bandstructure* plot, in which $E^n_{\mathbf{k}}$ is plotted as a function of $\mathbf{k}$ for a number of energy bands along straight lines between various special high symmetry points in reciprocal (wavevector) space. Figure 1.1 shows the form of the Brillouin zone for silicon. The labelled points refer to the following positions at the centre and edges of the Brillouin zone:

$$\Gamma = \frac{2\pi}{A}(0,0,0),\ \mathrm{L} = \frac{2\pi}{A}(0.5,0.5,0.5),\ \mathrm{X} = \frac{2\pi}{A}(1,0,0),\ \mathrm{K} = \frac{2\pi}{A}(0.75,0.75,0).$$

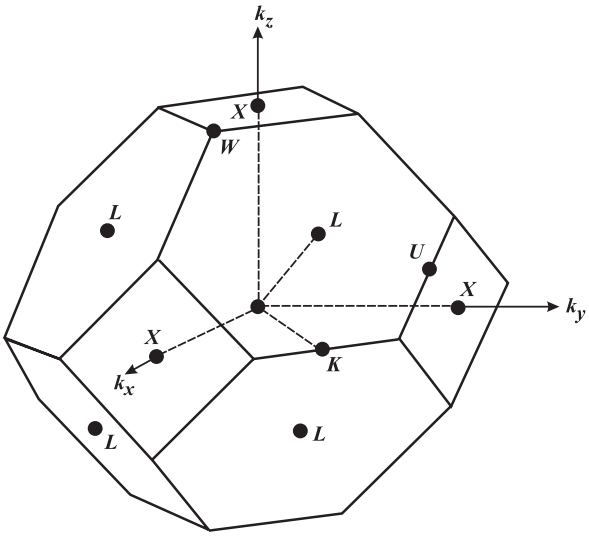


Figure 1.1: The FCC Brillouin zone showing the positions of some special points

Figure 1.2 shows an actual bandstructure plot for GaAs. Gallium arsenide is a direct gap material and the *forbidden energy gap*, the region in which there are no observable solutions, appears between the top valence band (band 4) and the lowest conduction band (band 5). This is clearly visible in the figure. At Γ the value of the energy gap for GaAs is about 1.5 eV. (If spin is included simply by imagining the existence of degenerate spin up/down energy bands the gap would instead appear between bands 8 and 9.)

1.4 Your Work Plan

1. **Design your program** The first task is to design your program. Use pseudocode techniques to design a program that you will use to investigate the band structure of various crystals.

   As well as an outline of the code structure, you should describe: (a) the names of the functions you will use and the variables they will take as arguments and what they will return (b) the data you will input to the program and how the program output will the results (c) the data structures the program will use to store its internal data

   Here are some suggestions for the components you should include:

   - a function to generate and store the reciprocal lattice vectors, ordered by increasing magnitude. As all RLVs include a common $2\pi/A$ factor you need not include this factor at this stage. The RLVs should come in obvious {0,0,0}, {1,1,1} etc groups.
   - a function to set up the $V_{\mathbf{g}''} = V_{\mathbf{g}-\mathbf{g}'}$ terms in the $N \times N$ matrix on the LHS of equation (1) employing the N lowest magnitude RLVs. For silicon the Fourier components $V_{\mathbf{g}''}$ are of the form:

     $$V_{\mathbf{g}''} = S(\mathbf{g}'')V(|\mathbf{g}''|)$$

     where the *structure factor* is $S(\mathbf{g}'') = \cos\left[\frac{\pi}{4}(n'_x + n'_y + n'_z)\right]$ and the *form factor* $V(|\mathbf{g}''|)$ only depends on the magnitude of $\mathbf{g}''$. You need only use the following 4 form factors:

     $$V_3 \text{ for } \mathbf{g}'' \text{ such that } n'^2_x + n'^2_y + n'^2_z = 3 \left(\Rightarrow \mathbf{g}'' \text{ is of the form } \frac{2\pi}{A}(\pm 1, \pm 1, \pm 1)\right)$$

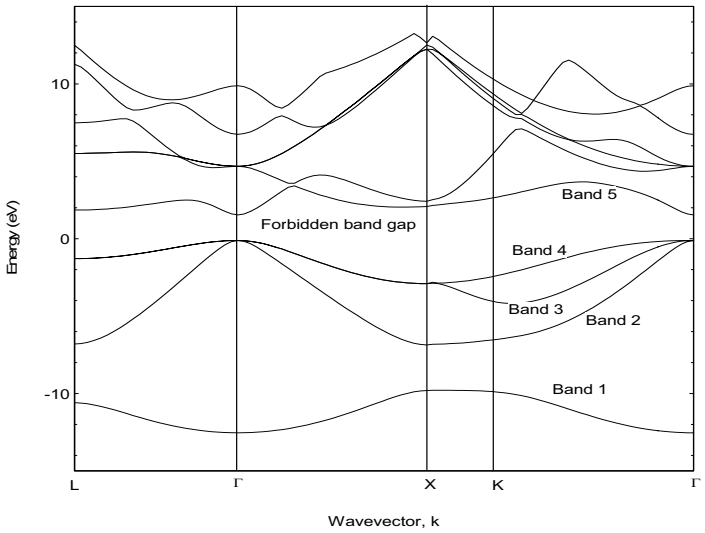


Figure 1.2: Bandstructure of GaAs

V_8 for $\mathbf{g}''$ such that $n_x'^2 + n_y'^2 + n_z'^2 = 8$ $\left(\Rightarrow \mathbf{g}''$ is of the form $\dfrac{2\pi}{A}(\pm 2, \pm 2, 0)\right)$ and

V_{11} for $\mathbf{g}''$ such that $n_x'^2 + n_y'^2 + n_z'^2 = 11$ $\left(\Rightarrow \mathbf{g}''$ is of the form $\dfrac{2\pi}{A}(\pm 3, \pm 1, \pm 1)\right)$.

V_0 (corresponding to $\mathbf{g}'' = \mathbf{0}$) is in effect the average potential in the crystal. Depending on its value all eigenvalues simply shift uniformly up or down. It can initially be set to zero but eventually it should be set such that the top of the valence band at Γ corresponds to zero energy, the convention used in plotting the results in Figure 1.2.

- functions to solve and print on the screen the eigenvalues of the matrix with given N and a single value of $\mathbf{k}$. This can be done by employing the "'numpy"' module using a statement like:

  EIGENVALUES=np.linalg.eigvalsh(MATRIX)

  where MATRIX is an $N \times N$ matrix. EIGENVALUES will then be an array containing the N eigenvalues of the matrix. Use an "import numpy as np" statement to ensure that numpy is available to your program. You should try this out on a simple 2×2 matrix to assure yourself that it works.

2. **Solve the milestone problem.** Use your program to calculate results for a *free electron* situation. Do this by setting $V_0 = V_3 = V_8 = V_{11} = 0$ with $N = 15$ and $A = 5.43 \times 10^{-10}$ m (the lattice constant of silicon). What does your program give for some of the energies (in eV) at the Γ, X, L and K points? These energies are easy to check in the free electron case and this should be done before proceeding further. When you are sure that they are correct extend your program so that, as well as a single point calculation, you can also plot a properly proportioned free electron band structure in the range from L to Γ to X (a subset of the range of the plot shown in Figure 2).

   At the milestone interview, your program should calculate the energies at the Γ, X, L and K points in the free electron model (using $N = 15$), and plot the L to Γ to X energy band structure.

3. **Extending your program** The milestone program is a useful test, but the purpose of your program is to investigate the band structure of semi-conductors.

   (a) Investigate the convergence of your program as a function of N by calculating the band gap between bands 4 and 5 at $\mathbf{k} = 0$ for a series of increasing values of N (15, 27, 51...) using the following values of the Si form factors:

   $$V_3 = -3.04768 \text{ eV}, \; V_8 = 0.74831 \text{ eV}, \; V_{11} = 0.97961 \text{ eV}.$$

   If the calculations are working correctly this energy gap should be about 3 eV. The results may be considered to be sufficiently converged once the value of N is such that the energy gap is changed by less than 0.01 eV compared to the previous value. At what value of N does this occur? Use this value of N for the following calculations.

   (b) Calculate the correct band structure of Si from L $\rightarrow \Gamma \rightarrow$ X. **You will need to use the expressions for V_0, V_3, V_8 and V_{11} given above.** In the case of Si you should find that the material has an indirect band gap between the valence band maximum at Γ and the conduction band minimum. At what k-value (in units of $2\pi/A$) and in what $\mathbf{k}$ space direction does the conduction band minimum occur and what (in eV) is the value of the forbidden energy gap?

4. **Research with your Program**

   With a basic working program you are in a position to do some research. Here are some suggestions for the starting point of your investigation. You should not look at all the points below, it is better to pick one topic an investigate it in depth.

   - You can apply your program to other elemental materials (like germanium). Non-elemental materials such as GaAs and other III-Vs and II-VIs require a slightly modified form of the potential. Chapter8 of Kittel's book contains a extensive discussion of Crystal structures. Starting from measurements of the band stucture of a crystal (for Si, see Vos et al., 2006), can you determine suitable values for the form-factors V_0, V_3 etc.
   - Calculation of, for example, density of states, is relatively straightforward and by making use of the eigenvectors a whole range of additional calculations can be performed and compared to experimental data. The density of states is discussed in Kittel's book. For example, you could use you band structure calculation to determine the effective masses of electrons and holes and compare this with observational data.
   - Your calculations of the Eigen states also allow you to detemine the wave functions of electrons within the crystal structure. For example, are the electrons localised around the atoms, or along the "bonds" that connect them? Many examples are shown in Kittel's and Ascroft's books. Compare your calculations to published examples.

References

Introduction to Solid State Physics, C. Kittel, Wiley.

Solid State Physics, N.W. Ashcroft and N. D. Mermin, Saunders.

Electronic Structure and Optical Properties of Semiconductors, M. L. Cohen and J. R. Chelikowsky, Springer-Verlag (This contains more detailed experimental and theoretical information on individual semiconductors and the pseudopotential approach.)

Vos et al., 2006, Phy Rev B. http://journals.aps.org/prb/abstract/10.1103/PhysRevB.73.085207. An example research paper carrying out an experimental investigation of the band structure of Si. You can readily search for other examples.

You may find more detailed information concerning the pseudopotential method at (for example): http://www.nanohub.org/resources/1524/download/ empericalpseudopotentialmethod_word.pdf

Chapter 2

Simulation of a Phase Transition

2.1 Introduction

The transitions from a solid to a liquid or from a material that is paramagnetic (look up the definition!) to one that is ferromagnetic have many similar properties. It would be impossible to follow the behaviour of all the particles in some macroscopic volume using a computer (why?). However, with a simplified model for the physical interactions and what is known as a *Monte-Carlo* approach, we can perform a representative simulation of the behaviour of the system as it moves from one state to another.

Here we concentrate on a magnetic system, which we shall study using the *Ising model*. This is described as follows: consider a two dimensional, regular lattice of $N_x \times N_y = N$ atoms, each with a net electronic spin $\mathbf{S}$. There are two contributions to the energy of the system.

1. There is the interaction between the spin on each atom and any externally applied magnetic field $\mathbf{B}$. The energy of this interaction is simply $-\sum_\alpha \mathbf{S} \cdot \mathbf{B}$, where we have absorbed the magnetic moment into the definition of the magnetic field.

2. There is also an 'exchange' interaction between particles arising from the Pauli exclusion principle. Remind yourself of the role that spin plays in the Pauli exclusion principle.

   The energy of this interaction can be written as $-J\mathbf{S}_\alpha \cdot \mathbf{S}_\beta$, where J characterises the strength of this exchange force. This force is very localised and is only significant between the four immediate nearest neighbours of a particle on the lattice. If $J > 0$ then it is energetically favourable for adjacent spins to be parallel ; if $J < 0$ then the spins are preferentially anti-parallel.

If we make the further simplification that the particles can only be either 'spin-up' or 'spin-down' along the axis defined by the external field, then the total energy of the system can be written as the sum of these two terms:

$$E = -B \sum_\alpha S_\alpha - J \sum_{\langle \alpha\beta \rangle} S_\alpha S_\beta \tag{2.1}$$

Here, the first summation runs over every location on the lattice, with α denoting a particular combination of the labels i and j, with $i = 1, N_x$ and $j = 1, N_y$. The second term is a double summation: again

α runs over every ij, but β only runs over the four nearest neighbours: $(i-1,j)$; $(i+1,j)$; $(i,j-1)$ and $(i,j+1)$.

Pause here for a moment and ensure that you understand the physics of the above equation. Sketch a diagram that describes the model that this equation represents. It should look like an $N \times N$ 'random' chess board where spin up and spin down can be represented by black and white squares, etc.

2.2 Macroscopic properties

In statistical mechanics, the macroscopic properties of a system, such as the total energy or the magnetisation, are calculated by looking at an ensemble (collection) of all the possible states of the system. Even with the simple Ising model, it would be infeasible to realise all the possible configurations of spin orientations and average their energy. If we restricted ourselves to a 10×10 lattice, the number of lattice sites is 100, each with two possible spins, giving a total of 2^{100} ($\sim 10^{30}$) distinct configurations. Estimate the time it would take your computer to calculate the energy of each of these states. You will see that we need to be selective in your choice of which states to look at. Fortunately, many of the possible states are very unlikely. For example if the temperature of the system is such that the thermal energy of the atoms is much higher than the interaction energy between spins, then the overall magnetisation will be close to zero - the thermal vibrations quickly disrupt any alignment of spins. Therefore in this case states which have a significant magnetisation or energy will be less likely to occur than states that have zero magnetisation.

This information is contained in a quantity called the *partition function* of the system - this is a *weighted* sum over all the possible states of the system, where the weight is the Boltzmann factor for the state:

$$Z = \sum_i \exp(-E_i/kT). \tag{2.2}$$

Hence, states which have a 'high' energy – relative to the thermal energy kT – make a small contribution to the partition function. We can use this to help us decide which configurations to consider. (What is kT at room temperature?)

2.3 The Metropolis Algorithm

In the algorithm developed by Metropolis et al, new states or configurations of the system are found by systematically moving through all the lattice sites. At each site, the spin of the atom is flipped and the change in energy of the whole system is calculated; then according to a simple rule, this spin flip is either kept or the spin is returned to its original value. To calculate the change in energy of the system after a spin-flip, we only need to consider the four nearest neighbours of the spin on the lattice, because this is the range of the exchange interaction.

In terms of the flipped spin S_α, the energy change is then given by

$$\Delta E = -2\left[Jf + B\right] S_\alpha\,, \tag{2.3}$$

where the factor of 2 arises because we have flipped the spin and the factor f is equal to the sum of the spins on the sites that are the nearest neighbours of the spin that has been flipped. f can take the values $-4, -2, 0, 2, 4$. Before moving on, convince yourself this is correct.

The Metropolis algorithm is straightforward:

1. Flip a given spin
2. Calculated this energy change
3. (1)We retain the spin flip if $\exp(-\Delta E/kT) > 1$ or
   (2) if not, we draw a uniform random number r between 0 and 1 and if $\exp(-\Delta E/kT) > r$ we keep the spin flip.
   (3) If neither of these conditions is met, we leave the spin with its original orientation
4. Move on to the next spin in the lattice.

Every time all the spins on the lattice have been looped through, in each case testing whether or not a spin-flip should be retained according to the rules outlined above, is called a thermalisation sweep. When a set number of thermalisation sweeps have been executed, properties of the lattice, such as the magnetisation per spin, can be computed:

$$M_S = \frac{1}{N_x N_y} \sum_{\alpha} S_\alpha \,. \tag{2.4}$$

2.4 Your Work Plan

1. **Design your program**. Your first task is to design the program described below using pseudocode techniques. As well as an outline of the code structure you should include names and descriptions of any functions that you will need, any data that may require input from the user, internal data structures and the form of the output. You should make use of the numerical python library (numpy) as much as possible. For example numpy.random.rand can be used for generating random numbers, etc.

   Note that the only parameter in the model is J/kT. Therefore you should write your code in terms of this parameter and not J and T separately.

2. **Solve the Milestone Problem**. Set up a 2D lattice of N_x by N_y spins. Initially, give the spins a spin-up or spin-down orientation at random. The code will then perform thermalisation sweeps, looping through every spin on the lattice, computing the energy change associated with a spin-flip and retaining the flip if the Metropolis conditions are met. Then every n sweeps, compute the magnetisation per spin of the lattice. Experiment with the dimensions of the grid *e.g.* try square grids of dimensions $10, 16, 32$. Experiment with the number of thermalisation sweeps used in between calculating the magnetisation of the lattice. Plot the magnetisation as a function of the number of thermalisation sweeps. How many thermalisation sweeps do you need to perform before equilibrium is reached? One test of the code would is to monitor the net magnetisation as the number of thermalisation sweeps increases. Start with $J = 0$ and $B = 0$ and show that there is no net magnetisation. You can then experiment with different values of J and B.

   At the milestone interview, your program must plot the magnetisation of the system as a function of the number of thermalisation sweeps for the case $J = 0.5$, $B = -0.05$. Use a 32 x 32 lattice for this test. You should find that the magnetisation per spin has converged to about -0.95 after 30 sweeps.

3. **Research With Your Program** Now you have a working code, you can investigate the relaxation of lattices of interacting molecules/atoms. High marks will only be given to reports that demonstrate your initiative. Here are some aspects that you may wish to consider including. You do not need to address all of these points: these are intended as the starting point for your own investigation.

   - Investigate the effect of varying the exchange force. Output the magnetisation per spin with no external field applied for values of J/kT in the range $0 \leq J/kT \leq 1.0$. Compare your answers with the analytic prediction of

   $$M/(N_x N_y) = \pm \frac{(1+z^2)^{0.25}(1-6z^2+z^4)^{0.125}}{\sqrt{(1-z^2)}}\,, \tag{2.5}$$

   where $z = \exp(-2J/kT)$ (Onsager 1943). Investigate how the behaviour of the model depends on the dimension of the matterial and the range of the interactions.
   - Consider what further Physics can be explored using your model. For example, by calculating the partition function, you can evaluate many physical properties such as the magnetisation and the specific heat capacity. The results behaviour you obtain is quite general (it is a second-order phase transition) and the characteristic dependence on temperature is seen in a wide range of physical situations. You will find a good introduction in the Landau and Giordano books on Computational Physics, and further discussion of critical exponents in Pathria's book.
   - Make an image of the magnetisation of the lattice, showing up and down spins as black and white squares. Investigate the patterns that emerge when the external field is zero. What determines the size of the coherent regions? Find ways of characterising the size distribution, for example calculating the correlation function. Show that the size of the regions undergoes a phase transition.
   - Investigate the role of an External Magnetic Field. Repeat the first exercise for different values of the external magnetic field. In particular, try setting B/kT equal to 0.1 and 1.0. Show that the presence of the magnetic field influences the net magnetisation. By applying the magnetic field you will observe a first order phase transition (as occur when water freees). Another important effect is *hysteresis*: the magnetisation of the system will be different depending on whether the applied field is increasing or decreasing. These issuses are discussed in the Giordano and Pathria books. For advanced discussion of Hysteresis and the associated *cracking noise* look at Sethna et al 2001.

References

Onsager 1943, Phys. Rev., 65, 117. http://journals.aps.org/pr/pdf/10.1103/PhysRev.65.117

Giordano N. J., Nakanishi H., Computational Physics, Pearson Prentice Hall.

Landau R. H. et al., Computational Physics, Wiley.

Pathria R. K., Statistical Mechanics, Elsevier.

Sethna J. P. et al. 2001, Nature 410, 242.
http://www.nature.com/nature/journal/v410/n6825/full/410242a0.html

Chapter 3

Colloidal Fluids: Simulations of Hard Disks and Hard Spheres

3.1 Background

Soft matter physics is often characterised by the importance of entropy. This is exemplified by the so-called hard sphere model, consisting of impenetrable particles which otherwise do not interact. If the particles have a diameter σ, the interaction potential between two such particles i and j is given by

$$U_{ij}(r_{ij}) = \begin{cases} \infty & \text{if } r_{ij} < \sigma \\ 0 & \text{if } r_{ij} \geq \sigma, \end{cases}$$

where r_{ij} is the center-to-center distance between the particles. The potential energy of the system is therefore always zero without any overlap.

Hard sphere model plays an important role in our understanding of liquids, as well as being one of the first ever systems to be studied carefully using computer simulations. It is worth mentioning here that understanding the structure of liquids is essential to understand the myriad phenomena that take place in the liquid state. Furthermore, despite its simple interaction potential, the hard sphere model proves to be anything but simple. It was generally believed prior to the 1950s that a fluid to solid phase transition must be due to attractive interactions between the molecules. Pioneering Monte Carlo simulations done at Los Alamos and Lawrence Livermore National Laboratories challenged this view by showing that hard spheres do have a fluid-solid phase transition. In fact, the phase transition is driven by entropy. This is a significant statement at that time, as it shows that the notion of "entropy is disorder", while adequate for many cases, is not always accurate.

In this project we will look at Monte Carlo simulations using the hard sphere model. The milestone problem requires you to study a structural signature of the fluid to solid phase transition in two dimensions. The hard sphere model remains a topic of current scientific interests, and there are a number of avenues you can choose to extend the project.

3.2 Monte Carlo Simulation

The method of choice in this project is the so-called Monte Carlo simulation. In Monte Carlo, we estimate the properties of the system we are interested in by taking an average over the possible states, or in more

technical terms, by taking an *ensemble average*. This is perhaps not the way many of us think about the average behaviour of a system. In experiments we usually take a series of measurements over a certain time interval and take an average of these measurements. In the world of computer simulations, this is precisely the idea behind Molecular Dynamics (and also several other methods), where the system is studied by computing the natural time evolution of the system numerically. The estimated properties of the system should be identical, either by taking time or ensemble averages, if the system is *ergodic*. Note that not all physical systems are ergodic. Such systems usually require special care to get reliable simulation results.

Since the number of states in condensed matter systems is usually very large, it is impractical, if not impossible, to visit all possible states. This is where the concept of importance sampling becomes important in Monte Carlo. Firstly, if we run our simulation sufficiently long, it will have sampled the phase space adequately, even if not all states have been occupied during the simulation. Secondly, in the context of a classical system, the probability of a state to be occupied is proportional to its Boltzmann factor, $\exp[-U/k_BT]$ [**?**], where U is the total potential energy of the system. It therefore makes sense that we sample lower energy states more often, since they have higher probabilities to be occupied. But how does this work in practice?

An algorithm which is used in most Monte Carlo simulations is due to Metropolis. Asssume you are initially in configuration I. In this algorithm, first generate a random perturbation to the system. Next, (i) accept the move if the energy of the new configuration J is lower, or (ii) accept the move with a probability $\exp[(U_I - U_J)/k_BT]$ if $U_J > U_I$. In practise, this is done by generating a random number, *rand*, between 0 and 1. If $rand < \exp[(U_I - U_J)/k_BT]$, the move is accepted. An important property of the Metropolis algorithm is that it satisfies detailed balance: at equilibrium, each elementary process should be equilibrated by its reverse process.

There are a number of Monte Carlo moves available in the literature. For our problem, it is adequate to consider random Cartesian moves, where the change in the x-coordinate of particle i is given by

$$\Delta x_i = d(\xi - 0.5),$$

where ξ is a random number between 0 and 1, and d is the amplitude of the possible displacements. A similar formula can be written for the y and z coordinates of the particles. Usually you want to choose d such that $25 - 50\%$ of the moves are accepted. Since we do not know this a priori, in practise d is regularly updated every N_{MC} Monte Carlo steps. As a guide, larger step sizes tend to be rejected more often. On the other hand, small step sizes may have a higher chance to be accepted, but you have not moved very far in the phase space.

In our hard sphere model, the potential energy of the system is zero when there is no overlap, and infinite if there is overlap. Thus, the accept/reject criterion is reduced to detecting any possible overlap in the configuration of the particles. Note that kinetic energy is not used in Monte Carlo [**?**] (Why is this?). It is also customary to apply periodic boundary conditions when investigating bulk properties using computer simulations. The simulation box is replicated in space such that when a particle leaves the cell from one side, under periodic boundary conditions, the particle will re-enter the cell from the opposite side.

To analyse the configurations generated by the simulation, a particularly useful tool is the so-called pair-correlation function. Mathematically, it is defined as

$$g(r) = \frac{n(r)}{n_{\text{ideal}}(r)},$$

where $n_{\text{ideal}}(r) = \rho 4\pi r^2 \delta r$ in three dimensions and $\rho 2\pi r \delta r$ in two dimensions, $n(r)$ is the number of particles within a shell δr at a distance r, and ρ is the number density of the system. Thus, $g(r)$ is the probability

relative to the ideal gas case of observing $n(r)$ particles at a distance r. For getting $n(r)$ reliably, you will have to generate many configurations of the particles and average over all those configurations. You also want to optimise the choice for δr. If δr is too small, your results will be very noisy. If it is too large, you will lose a lot of information.

3.3 Your Work Plan

3.3.1 Design Your Program

Your first task is to design the program described below using pseudocode techniques. As well as an outline of the code structure you should include names and descriptions of any functions that you will need, any data that may require input from the user, internal data structures and the form of the output. You should make use of the numerical python library (numpy) as much as possible. For example numpy.random.rand can be used for generating random numbers, etc.

You will need separate subroutines/functions which carry out the following tasks:

- Generate starting configurations. The simplest option is to put the particles in a lattice (Hint: what lattice type do you expect to see when the system forms a crystal?). Alternatively, you can place the particles in a random configuration. Using this strategy, however, it may prove difficult to avoid overlaps between particles at high density.
- Generate Monte Carlo moves. You can start by moving one particle per step, and increase (decrease) the amplitude d by 5% if the fraction of accepted moves after $N_{MC} = 1000$ steps is more than 50% (less than 25%).
- Compute the energy of the system. In practise, you only need to compute the total energy once at the beginning of the simulation. It is a lot faster to track the change in energy associated to the Monte Carlo moves, rather than computing the total energy every single step. For the hard sphere model, this subroutine is reduced to detecting overlaps between the particles.
- Accept/reject criterion according Metropolis algorithm. Due to the peculiarity of the hard sphere potential, the energy scale (and thus temperature) is irrelevant here, since the potential energy of the system is either zero or infinite (But it is important if you choose other potentials).
- Compute the pair-correlation function. Make sure that you only make measurements after the system is properly equilibrated. A good starting point for δr is of order one-tenth of the particle radius. When computing the distance between two particles, make sure you take into account the periodic boundary condition.

3.3.2 Solve the Milestone Problem

Using the hard sphere model, you will now investigate the pair correlation function, $g(r)$, as a function of the area/volume fraction, η, occupied by the particles. If L and N are respectively the size of your simulation box and the total number of particles, then $\eta = \frac{N\pi\sigma^2}{4L^2}$ in two dimensions and $\eta = \frac{N\pi\sigma^3}{6L^3}$ in three dimensions.

For simplicity, let us focus here on two dimensions, and we shall use an ensemble where the total number of particles and the size of the simulation box are constants.

Initialise your code such that it contains 100 particles and the area fraction occupied by the particles is $\eta = 0.68$. To remove the bias of your starting configuration, it is important that you equilibrate your system before taking any measurement. A typical equilibration run is of order $10^4 - 10^5$ steps, though this can be system-dependent.

After the system is properly equilibrated, run your Monte Carlo simulation for 5×10^5 steps and sample the pair correlation function every 1000 steps. Compute and plot the average of the pair correlation function for $r = 0 \ldots 5\sigma$. You should observe that $g(r) = 0$ for $r < \sigma$; it has its highest value at $r = \sigma$; and has several peaks, including at $r/\sigma \sim 0.19$, 0.28, and 0.37.

3.3.3 Extend Your Program

You have a basic working program, but you should consider the following points before you start your research work:

- How do you make sure that your simulation results have converged?
- The fact that $g(\sigma) > 1$ suggests that there is an effective attractive interaction between the particles. How do you explain this observation?
- Repeat the milestone analysis for $\eta = 0.69$, 0.70, 0.71, 0.72. Compute and plot the average of the pair correlation function for $r = 0 \ldots 5\sigma$. You may need to sample $g(r)$ over a larger number of simulation steps with increasing η. Furthermore, you will observe the appearance of a new peak. This is a structural signature of the fluid to solid transition. Determine the area fraction where this phase transition occurs.

3.3.4 Research With Your Program

Now that you have a working code, there are several avenues to extend this project. Here I list several suggestions, but you are free to choose a different direction. You do not need to address all of these points. A good source for inspiration and references is the sklogwiki pages for hard spheres and related problems: http://www.sklogwiki.org/SklogWiki/index.php/Hard_sphere_model. High marks will be given to reports that demonstrate your initiative.

- **Dimensionality**: Compare the simulation results for two and three dimensions. Do you expect to observe anything interesting in 1D?
- **Advanced Methods**: (i) Study the same problem using NPT ensemble (constant number of particles, pressure, temperature). A nice advanced calculation you can do then is to look for coexistence between the fluid and solid phases. (ii) Improve the efficiency of the simulation, e.g. by creating a neighbour list. This is particularly useful for high density.
- **New Interactions**: (i) Investigate the difference between the hard sphere model described here to a Lennard-Jones potential, a very popular model in computational chemistry. (ii) Study the effect of shapes, e.g. rods. There are novel liquid crystalline phases simply due to the shapes of the molecules.
- **Mixture**: Study binary hard spheres (or disks), where the two components have different diameters. You can investigate when phase separation occurs, and when the system is completely mixed.

- **Thermodynamics**: Compute the pressure and/or free energy of the fluid and solid phases. Note: Computing the free energy of the solid phase is quite challenging.

3.3.5 References

T. Guénault, *Statistical Physics*, Springer (2007).

D. Frenkel and B. Smit, *Understanding Molecular Simulation*, Academic Press (2002).

Additional references can be found in:
http://www.sklogwiki.org/SklogWiki/index.php/Hard_sphere_model

Part II

Atomic and Optical Physics

Chapter 4

Light-Matter Interactions in atomic physics

4.1 Introduction

What do the energy levels of an atom, ion or molecule do when I apply a magnetic, electric or laser field? This is a very common question in atomic, molecular and optical physics and vital for applications such as quantum technology, but not always so easy to answer because even 'simple' atoms are relatively complex - due to the effects of electron and nuclear spin, etc.

The aim of this project is to be begin by calculating the effect of magnetic fields on simple atoms such as hydrogen and other alkalis, and then explore more complex problems. One example might be the tensor electric polarizability of an alkali atom ground state.

4.2 Uncoupled basis

The starting point is to assume that the atomic energy levels are all degenerate and then add in various interactions. Essentially, this is *degenerate perturbation theory*, however, the label 'perturbation' could be misleading, as this typically means something that is 'small' such that one can perform a *perturbative expansion to first or second order*. However, in our case we will find the eigenvalues numerically, and the energies are exact to all orders.

We will use the **uncoupled basis** where the states are labelled according to their coarse structure quantum numbers n and ℓ and the magnetic quantum number for each component of angular momentum (orbital m_ℓ, electron spin, m_s, and nuclear spin, m_I). Thus the state labels are $|n\ell : m_\ell m_s m_I\rangle$ In this basis the atomic Hamiltonian can be written as

$$\hat{H} \quad = \quad \hat{H}_0 + \hat{H}_{\rm fs} + \hat{H}_{\rm hfs} + \hat{H}' \ , \tag{4.1}$$

where H_0 is a diagonal matrix consisting of the coarse structure energy levels, $E_{n\ell}$. $H_{\rm fs}$ and $H_{\rm hfs}$ are the fine and hyperfine interactions, and H' is the interaction induced by the external field. Ignoring $H_{\rm fs}$, $H_{\rm hfs}$ and H', each $n\ell$ level has a degeneracy of $D_{n\ell} = (2\ell+1)(2I+1)(2s+1)$, consequently the Hamiltonian for each $n\ell$ level can be written as a sub-matrix of dimension $D_{n\ell} \times D_{n\ell}$.

Example: For the hydrogen ground state $I = 1/2$, $\ell = 0$, the unperturbed s-state matrix $H_{\rm s}$ is a 4×4 with elements $\langle i,j|H_0 + H_{\rm fs} + H_{\rm hfs}|i,j\rangle$, where $|i,j\rangle$ are the spin states $\left|m_s = +\frac{1}{2}, m_I = +\frac{1}{2}\right\rangle = |+,+\rangle$, $|+,-\rangle$, $|-,+\rangle$, and $|-,-\rangle$. For $\ell = 1$, $H_{\rm p}$ is a 12×12 matrix where the four spin components are repeated three times, once for each m_ℓ state.

4.3 Spin matrices

A key property of atomic systems is *angular momentum* whether from spin or orbital motion. In quantum mechanics, angular momentum or spin is described by a 'spinor', which mathematically is written as a matrix. The classic example is the spin-1/2 matrices known as the Pauli matrices. To describe atoms we need to build matrices that characterize any angular momentum, j.

All angular momentum operators satisfy the same set of commutation rules and we can derive the general result (for an arbitrary angular momentum j, see e.g. *Quantum mechanics*, Ballantyne, p. 121)

$$\hat{j}_+|j,m\rangle \;=\; \sqrt{j(j+1) - m(m+1)}|j,m+1\rangle \;,$$

where $\hat{j}_+ = \hat{j}_x + i\hat{j}_y$. Consequently the matrix $\hat{j}_+$ is only non-zero along the upper diagonal with elements $\sqrt{j(j+1) - m(m+1)}$, e.g. for $j = \frac{3}{2}$ we have (note that it is the m value along the top that is used to calculate the matrix element):

$$\begin{array}{c|cccc} & \frac{3}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{3}{2} \\ \frac{3}{2} & 0 & \sqrt{3} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \sqrt{4} & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & \sqrt{3} \\ -\frac{3}{2} & 0 & 0 & 0 & 0 \end{array} \;.$$

$\hat{j}_-$ is given by the transpose of $\hat{j}_+$, $\hat{j}_x = \frac{1}{2}(\hat{j}_+ + \hat{j}_-)$, $\hat{j}_y = -\frac{i}{2}(\hat{j}_+ - \hat{j}_-)$ and $\hat{j}_z = \frac{1}{2}(\hat{j}_+\hat{j}_- - \hat{j}_-\hat{j}_+)$. Using these results one can construct a function to generate the x, y and z components for any j.

4.4 Zeeman effect

In an external magnetic field, $\boldsymbol{\mathcal{B}}$, the ground state energy levels are shifted by both the hyperfine interaction and the Zeeman effect. The total perturbation is given by the sum of these two terms, i.e.,

$$\hat{H}_{\rm hfs} + \hat{H}' \;=\; \mathcal{A}\hat{\boldsymbol{I}} \cdot \hat{\boldsymbol{J}} - (\boldsymbol{\mu}_I + \boldsymbol{\mu}_J) \cdot \boldsymbol{\mathcal{B}} \;, \tag{4.2}$$

As $\mu_N = (m_e/m_p)\mu_B$ and m_p is 3 orders of magnitude larger than m_e we will neglect $\boldsymbol{\mu}_I$.[1] If the magnetic field is along the z-axis, i.e., $\boldsymbol{\mathcal{B}} = (0,0,\mathcal{B})$, and for $L = 0$, $\boldsymbol{\mu}_J = \boldsymbol{\mu}_S$, using $\boldsymbol{\mu}_S = g_S\mu_B\hat{\boldsymbol{S}}$ with $g_S \approx 2$, we get

$$\hat{H}_{\rm hfs} + \hat{H}' \;=\; \mathcal{A}\hat{\boldsymbol{I}} \cdot \hat{\boldsymbol{J}} + 2\mu_B\mathcal{B}\hat{S}_z \;. \tag{4.3}$$

[1] For MRI, one can think of H nuclei (protons) in the 'mean-field' of the other electrons and nuclei. An applied field splits each level into a 'spin up' and a 'spin down' component with a splitting

$$\Delta E \;=\; g_I\mu_N\mathcal{B} \;.$$

For a field of 1 Tesla, the frequency splitting is 42.6 MHz.

We want to find the expectation value of this operator for all possible ground state sub-levels. A plot of these expectation values versus field is known as the *Breit-Rabi diagram*. For a single electron atom with nuclear spin I, one can obtain a simplified equation of the eigenenergy

$$E = \pm\frac{1}{2}\mathcal{A}\left(1+2\mathcal{M}x+x^2\right)^{1/2} , \tag{4.4}$$

where $\mathcal{M} = 2m/(2I+1)$ and we have shifted the zero of the energy scale by adding $\frac{1}{4}\mathcal{A}$. Note that this equation includes two unphysical solutions (for $m = \pm(l+1/2)$), which are not eigenstates. This can be understood by following through the matrix derivation in the appendix.

The energy levels as a function of the external field for ^{1}H and ^{133}Cs are shown in Fig. 4.1. You should be able to reproduce this figure from Eq. 4.4.

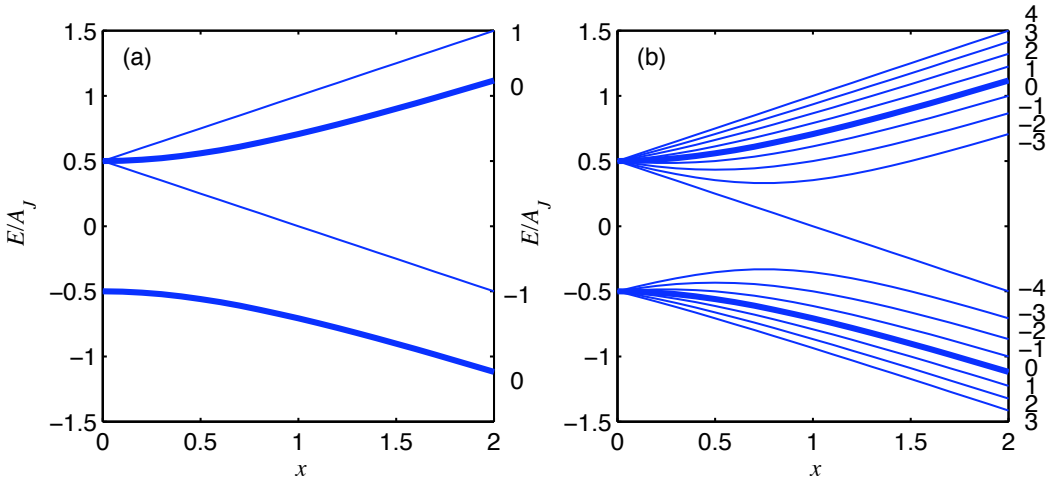


Figure 4.1: Breit-Rabi diagram showing the energy shift as a function of an external magnetic field, $x = \mu_B\mathcal{B}/2$, in units of the hyperfine splitting, $A_J = \mathcal{A}/(2I+1)$. The two panels show (a) H with $I = \frac{1}{2}$; and (b) Cs, $I = \frac{7}{2}$. Bold: magnetically insensitive $m_F = 0$ levels.

4.5 Numerical methods

The trick to finding the expectation values is to write all the operators appearing in Eq. 4.3 as matrices in the uncoupled basis. We construct these matrices using our rules for generating j_x, j_y and j_x for any j then adding components to find the total electric angular momentum $\hat{\boldsymbol{J}}$ or total atomic angular momentum $\hat{\boldsymbol{F}}$ using the addition rule, see Appendix A. First, we rewrite the hyperfine interaction as

$$\hat{\boldsymbol{I}}\cdot\hat{\boldsymbol{J}} = \tfrac{1}{2}\left[\hat{\boldsymbol{F}}^2 - \hat{\boldsymbol{I}}^2 - \hat{\boldsymbol{J}}^2\right] \tag{4.5}$$

and use $\hat{\boldsymbol{J}}^2 = \hat{J}_x^2 + \hat{J}_y^2 + \hat{J}_z^2$ to construct the matrix. Finally we add all the terms together and find the eigenvalues as function of the external magentic field.

4.6 Milestone

Write a code to produce matrices for an angular momentum j, add angular momenta and plot a Breit-Rabi diagram for H, Fig. 1(a). Compare to the analytical formula, Eq. 4.4 and make a residuals plot to test the accuracy of your numerics.

4.7 Extending the milestone

Some interesting topics include calculating the energy levels of atoms in electric fields and laser fields, which has relevance to application such as atomic clocks. The electric field problem is treated in W. L. Virgo, *Simultaneous Stark and Zeeman effects in atoms with hyperfine structure*, Am. J. Phys. **81**, 936 (2013). Some extension include the following:

- Calculate the Breit-Rabi diagram for an excitation state.
  Examples can be found in `http://steck.us/alkalidata/`
- Calculate the ground state level shifts in an electric field, see also `http://steck.us/alkalidata/` and for some discussion S. Ulzega *et al.*, *Simultaneous Stark and Zeeman effects in atoms with hyperfine structure*, Europhys. Lett. **78**, 69901 (2006).
- Calculate the energy levels of nitrogen vacancy centres in diamond, sometimes referred to as Nature's ion and thought to be a useful solid-state system for quantum computing. See Fig. 4 in arXiv:1302.3288
- Calculate both the ground and excited state shifts in a laser field to find magic wavelengths for use in applications such as *optical lattice clock*, see M. Takamoto *et al.*, *An optical lattice clock*, Nature **435**, 321(2005).

Appendix: Adding angular momenta

Once we have built the matrices for an arbitrary angular momentum it is easy to add two or more angular momenta. The components of the sum of two angular momenta, e.g., j_1 and j_2, in matrix form are given by

$$\hat{J}_i = \hat{j}_{1i} \otimes \mathbf{1}_{(2j_2+1)} + \mathbf{1}_{(2j_1+1)} \otimes \hat{j}_{2i} \ ,$$

where $\mathbf{1}_{(2j+1)}$ is a unit (identity) matrix with dimension $2j+1$. First we will do one example analytically for the simplest case of two spin-$\frac{1}{2}$ particles, (e.g. the ^{1}H ground state), and then develop a general method for any state.

Two spin-$\frac{1}{2}$ particles.

Consider two spin-$\frac{1}{2}$ particles (labelled 1 and 2). Each particle may be either in the 'spin-up' state, $|\uparrow\rangle$ or the 'spin-down' state, $|\downarrow\rangle$. We write the wavefunction of the first spin as $|\psi\rangle_1 = a_1|\uparrow\rangle + b_1|\downarrow\rangle$, and similarly for the second. We can also write this as a vector,

$$|\psi\rangle_1 = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \ , \text{ and } |\psi\rangle_2 = \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \ .$$

The combined state of the two spins is

$$|\psi\rangle = |\psi\rangle_1 \otimes |\psi\rangle_2 = a_1a_2|\uparrow\uparrow\rangle + a_1b_1|\uparrow\downarrow\rangle + a_2b_1|\downarrow\uparrow\rangle + a_2b_2|\downarrow\downarrow\rangle$$

or in vector notation

$$|\psi\rangle = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \otimes \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1a_2 \\ a_1b_2 \\ b_1a_2 \\ b_1b_2 \end{pmatrix},$$

where $\otimes$ is a tensor product (make copies of the second matrix with positions and prefactors given by the first). An operator in this 4-dimensional vector space is given by a 4×4 matrix. We want to construct the operator corresponding to the total angular momentum. The angular momentum of a spin-$\frac{1}{2}$ (in units of $\hbar$) is $\hat{\boldsymbol{J}} = \frac{1}{2}\hat{\boldsymbol{\sigma}}$, where $\hat{\boldsymbol{\sigma}} = (\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z)$ is a 'spinor' with components equal to the Pauli spin matrices. The total angular momentum is given by the sum of the contributions from each particle,

$$\hat{\boldsymbol{J}} = \tfrac{1}{2}(\hat{\boldsymbol{\sigma}}_1 \otimes \mathbf{1}_2 + \mathbf{1}_2 \otimes \hat{\boldsymbol{\sigma}}_2), \tag{4.6}$$

where $\mathbf{1}_2$ is a 2×2 unit (identity) matrix. Note that the operator order specifies which term acts on which spin: e.g., $\hat{\sigma} \otimes \mathbf{1}_2$ means that we apply $\hat{\sigma}$ to the first particle and the identity matrix (i.e. do nothing) to the second; whereas $\mathbf{1}_2 \otimes \hat{\sigma}_x$ means do nothing to the first particle and apply $\hat{\sigma}_x$ to the second.

To construct the matrix for $\hat{\boldsymbol{J}}^2$ consider each component separately,

$$\begin{aligned}
\hat{J}_x &= \tfrac{1}{2}(\hat{\sigma}_x \otimes \mathbf{1}_2 + \mathbf{1}_2 \otimes \hat{\sigma}_x) \qquad (4.7)\\
&= \tfrac{1}{2}\left[\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right] \\
&= \tfrac{1}{2}\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \\
\hat{J}_y &= \tfrac{1}{2}(\hat{\sigma}_y \otimes \mathbf{1}_2 + \mathbf{1}_2 \otimes \hat{\sigma}_y) \\
&= \tfrac{1}{2}\left[\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right] \\
&= \tfrac{1}{2}\begin{pmatrix} 0 & -i & -i & 0 \\ i & 0 & 0 & -i \\ i & 0 & 0 & -i \\ 0 & i & i & 0 \end{pmatrix}, \qquad (4.8)\\
\hat{J}_z &= \tfrac{1}{2}(\hat{\sigma}_z \otimes \mathbf{1}_2 + \mathbf{1}_2 \otimes \hat{\sigma}_z) \\
&= \tfrac{1}{2}\left[\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right] \\
&= \tfrac{1}{2}\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \end{pmatrix}. \qquad (4.9)
\end{aligned}$$

Next, we square the matrices[2]

$$\begin{aligned}\hat{J}_x^2 &= \frac{1}{4}\begin{pmatrix} 0&1&1&0\\ 1&0&0&1\\ 1&0&0&1\\ 0&1&1&0 \end{pmatrix}\begin{pmatrix} 0&1&1&0\\ 1&0&0&1\\ 1&0&0&1\\ 0&1&1&0 \end{pmatrix}\\ &= \frac{1}{2}\begin{pmatrix} 1&0&0&1\\ 0&1&1&0\\ 0&1&1&0\\ 1&0&0&1 \end{pmatrix}.\end{aligned}$$

Similarly,

$$\begin{aligned}\hat{J}_y^2 &= \frac{1}{2}\begin{pmatrix} 1&0&0&-1\\ 0&1&1&0\\ 0&1&1&0\\ -1&0&0&1 \end{pmatrix},\\ \hat{J}_z^2 &= \frac{1}{2}\begin{pmatrix} 2&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&2 \end{pmatrix},\end{aligned}$$

and the total angular momentum

$$\hat{\boldsymbol{J}}^2 = \hat{J}_x^2+\hat{J}_y^2+\hat{J}_z^2 = \begin{pmatrix} 2&0&0&0\\ 0&1&1&0\\ 0&1&1&0\\ 0&0&0&2 \end{pmatrix}.$$

To find the eigenvalues we put

$$\begin{vmatrix} 2-\lambda&0&0&0\\ 0&1-\lambda&1&0\\ 0&1&1-\lambda&0\\ 0&0&0&2-\lambda \end{vmatrix} = 0\ .$$

which gives

$$(2-\lambda)^2(\lambda^2-2\lambda)=0\ ,$$

giving three eigenvalues with $\lambda=2$ and one with $\lambda=0$. As the eigenvalues of the total angular momentum are $J(J+1)$, these correspond to $J=1$ and $J=0$, respectively. For $\lambda=0$

$$\begin{pmatrix} 2&0&0&0\\ 0&1&1&0\\ 0&1&1&0\\ 0&0&0&2 \end{pmatrix}\begin{pmatrix} a\\ b\\ c\\ d \end{pmatrix} = 0\ ,$$

which gives $a=d=0$ and $c=-b$, i.e., the eigenvector is

$$|0,0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 0\\ 1\\ -1\\ 0 \end{pmatrix},$$

[2]We could have avoided squaring a 4×4 matrix by making use of $\hat{\sigma}_x^2=\mathbf{1}_2$ and $\hat{\sigma}_{x1}=\hat{\sigma}_{x2}=\hat{\sigma}_x$, to rearrage (4.7) as

$$\begin{aligned}\hat{J}_x^2 &= \tfrac{1}{4}(\hat{\sigma}_x\otimes\mathbf{1}_2+\mathbf{1}_2\otimes\hat{\sigma}_x)^2\\ &= \tfrac{1}{4}(\hat{\sigma}_x^2\otimes\mathbf{1}_2+\hat{\sigma}_x\otimes\hat{\sigma}_x+\hat{\sigma}_x\otimes\hat{\sigma}_x+\mathbf{1}_2\otimes\hat{\sigma}_x^2)\\ &= \tfrac{1}{2}(\mathbf{1}_2\otimes\mathbf{1}_2+\hat{\sigma}_x\otimes\hat{\sigma}_x)\ ,\end{aligned}$$

Substituting

$$\hat{\sigma}_x\otimes\hat{\sigma}_x = \begin{pmatrix} 0&1\\ 1&0 \end{pmatrix}\otimes\begin{pmatrix} 0&1\\ 1&0 \end{pmatrix} = \begin{pmatrix} 0&0&0&1\\ 0&0&1&0\\ 0&1&0&0\\ 1&0&0&0 \end{pmatrix},$$

we obtain,

$$\hat{J}_x^2 = \tfrac{1}{2}(\mathbf{1}_2\otimes\mathbf{1}_2+\hat{\sigma}_x\otimes\hat{\sigma}_x) = \frac{1}{2}\begin{pmatrix} 1&0&0&1\\ 0&1&1&0\\ 0&1&1&0\\ 1&0&0&1 \end{pmatrix}.$$

where we have used the label $|j,m\rangle$ for the eigenvector, with m being the eigenvalues of $\hat{J}_z$. We will verify the m quantum numbers below. For $\lambda = 2$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = 0 \ ,$$

so either $a = 1$, $b = c = d = 0$, or $a = d = 0$ and $b = c$, or $d = 1$, $a = b = c = 0$, i.e., the eigenvectors are

$$|1,1\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \ , \ |1,0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \ , \text{ and } |1,-1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \ .$$

What about $\hat{J}_z$? Using (4.9) we find

$$\begin{aligned} \hat{J}_z|1,1\rangle &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ &= 1 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = 1|1,1\rangle \ . \end{aligned}$$

Similarly, $\hat{J}_z|1,-1\rangle = -1|1,-1\rangle$, $\hat{J}_z|1,0\rangle = 0$, and $\hat{J}_z|0,0\rangle = 0$.

For the hydrogen ground state the total angular momentum is labelled as F and its projection as m_F. By finding the eigenvectors of the total angular momentum matrix $\hat{F}^2$, we have related the eigenstates in the *coupled basis*, labelled $|F,m_F\rangle$ (which also happens to the eigenbasis of the hyperfine interaction), to those in the *uncoupled basis* $|m_I,m_s\rangle$.

Appendix B: Zeeman effect - coupled basis

The operator $\hat{S}_z$ only acts on the electron spin part of the wavefunction (and not on the nuclear spin part), therefore to calculate the matrix elements, $\langle F', m'_F|\hat{S}_z|F,m_F\rangle$, we need to write the $|F,m_F\rangle$ states in terms of their uncoupled spin components, $|m_I,m_S\rangle$, i.e.,

$$|F,m_F\rangle = U|m_I,m_S\rangle \ . \tag{4.10}$$

Using our previous result

$$\begin{aligned} |1,1\rangle &= |++\rangle \ , \\ |1,0\rangle &= \tfrac{1}{\sqrt{2}}\left(|-+\rangle + |+-\rangle\right) \ , \\ |0,0\rangle &= \tfrac{1}{\sqrt{2}}\left(|-+\rangle - |+-\rangle\right) \ , \\ |1,-1\rangle &= |--\rangle \ , \end{aligned} \tag{4.11}$$

and writing the spin wavefunction

$$|\psi\rangle = a|++\rangle + b|+-\rangle + c|-+\rangle + d|--\rangle \ ,$$

as a vector $|\psi\rangle = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$, then both

$$\begin{aligned} \hat{S}_z &= \tfrac{1}{2}\hat{\mathbf{1}} \otimes \hat{\sigma}_z \\ &= \tfrac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ &= \tfrac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} . \end{aligned}$$

and

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} .$$

are 4×4 matrices.[3] Using Eqs. (4.10) and (4.11) The matrix elements

$$\langle F', m'_F | \hat{S}_z | F, m_F \rangle = \langle m'_I, m'_S | U^\dagger \hat{S}_z U | m_I, m_S \rangle ,$$

are given by

$$U^\dagger \hat{S}_z U = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} ,$$

Using this result we can write Eq. (4.3) in matrix form

$$H' = \begin{pmatrix} \frac{1}{4}\mathcal{A}_J + \mu_B \mathcal{B} & 0 & 0 & 0 \\ 0 & \frac{1}{4}\mathcal{A}_J & -\mu_B \mathcal{B} & 0 \\ 0 & -\mu_B \mathcal{B} & -\frac{3}{4}\mathcal{A}_J & 0 \\ 0 & 0 & 0 & \frac{1}{4}\mathcal{A}_J - \mu_B \mathcal{B} \end{pmatrix} .$$

The off-diagonal terms in the centre are because

$$\begin{aligned} \hat{S}_z |0,0\rangle &= \hat{S}_z \tfrac{1}{\sqrt{2}} \left(|-,+\rangle + |+,-\rangle \right) \\ &= \tfrac{1}{\sqrt{2}} \left(\tfrac{1}{2} |-,+\rangle - (-\tfrac{1}{2}) |+,-\rangle \right) = \tfrac{1}{2} |1,0\rangle . \end{aligned} \tag{4.12}$$

and similarly $\hat{S}_z |1,0\rangle = \frac{1}{2}|0,0\rangle$. In contrast, $\hat{S}_z |1,\pm 1\rangle = \pm\frac{1}{2}|1,\pm 1\rangle$.

[3] U is simply a 'rotation' matrix that rotates any state in the $\{m_I, m_S\}$ basis onto the $\{F, m_F\}$ basis. Similarly, we can easily convert any operator written in the uncoupled $\{m_I, m_S\}$ basis, e.g. $\hat{S}_z^{\{m_I, m_S\}}$ into the coupled $\{F, m_F\}$ basis using

$$\hat{S}_z^{\{F, m_F\}} = U^\dagger \hat{S}_z^{\{m_I, m_S\}} U .$$

So a matrix that is easy to write down in the uncoupled basis, such as

$$\hat{S}_z^{\{m_I, m_S\}} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} ,$$

can be easily converted to the coupled basis

$$\begin{aligned} \hat{S}_z^{\{F, m_F\}} &= U^\dagger \hat{S}_z^{\{m_I, m_S\}} U \\ &= \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} . \end{aligned}$$

Chapter 5

Qubit Dynamics and Quantum Computing

5.1 Introduction

Qubits are two-level quantum systems, which form the basic building blocks of quantum technologies such as quantum communications and quantum computing. In practice, a qubit can be realised using a 'spin' in a solid-state system, a quantum dot in a semiconductor, an atom, ion or molecule, an 'artificial atom' in a superconductor, or even the two polarization states of light [see e.g. Chapter 7 in *Quantum Computation and Quantum Information*, M. A. Nielsen and I. L. Chuang (CUP 2000)]. An isolated qubit is described by a wave function of the form

$$|\psi\rangle = a|0\rangle + b|1\rangle, \tag{5.1}$$

with dynamics determined by the Schrödinger equation

$$i\hbar\frac{d|\psi\rangle}{dt} = \hat{H}|\psi\rangle. \tag{5.2}$$

Representing the state of the system as a column vector, $\hat{H}$ is then a 2×2 matrix describing the interaction between the qubit and a control field. For an atomic qubit, this is a laser field and (in a frame oscillating with the laser frequency)

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},\ \hat{H} = \frac{\hbar}{2}\begin{pmatrix} \Delta & \Omega \\ \Omega & -\Delta \end{pmatrix}, \tag{5.3}$$

where Ω is known as the *Rabi frequency* and is proportional to the laser field amplitude, and Δ is the *detuning* of the laser frequency ω_L from the qubit resonant frequency ω_{01}. For $\Delta \equiv \omega_L - \omega_{01} = 0$, the qubit evolves such that the populations in states $|0\rangle$ and $|1\rangle$ oscillate sinusoidally, with the Rabi frequency Ω, from zero to unit occupation. Write down the algebra and convince yourself this is correct.

We cannot completely isolate a qubit from the external world, however. This interaction leads to *decoherence* [see, e.g., Chap. 4 in S. Haroche and J. M. Raimond *Exploring the Quantum* (OUP 2006)], or 'collapse' of the wave function.[1] A simple example is spontaneous emission. If state $|1\rangle$ is an excited state which decays to a ground state $|0\rangle$, after some random time (where we call the average the excited

[1] Another interesting book is Joos *et al.*, *Decoherence and the appearance of the classical world* (Springer 2003).

state lifetime) the atom will decay by emitting a photon. If we detect this photon we know the atom is back in the ground state, i.e., $a = 1$ and $b = 0$. The collapse of the wave function $|\psi\rangle$ to the ground state $|0\rangle$ is known as a *quantum jump* and the complete time sequence $|\psi(t)\rangle$ as a *quantum trajectory*. Averaging many quantum trajectories yields the time-averaged response of a single qubit observed over multiple experimental runs, or, equivalently, the average response of a large ensemble of noninteracting qubits observed in parallel [Fig. 1 or Fig. 4.4 in Haroche and Raimond 2006 (ibid.)].

5.2 Density Matrix

If the state of the system can be described by a wavevector $|\psi\rangle$ (a *pure state*), propagated by the Schrödinger equation [Eq. (5.2)], we can define a *density matrix* $\rho = |\psi\rangle\langle\psi|$, propagated by

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[\hat{H}, \rho], \text{ where } \rho = \begin{pmatrix} |a|^2 & ab^* \\ a^*b & |b|^2 \end{pmatrix} \tag{5.4}$$

and $[\hat{H}, \rho] \equiv \hat{H}\rho - \rho\hat{H}$ is a commutator. The diagonal elements $\rho_{00} = |a|^2$ and $\rho_{11} = |b|^2$ give the populations in states $|0\rangle$ and $|1\rangle$, respectively, and the off-diagonal $\rho_{01} = ab^*$ and $\rho_{10} = \rho_{01}^* = a^*b$ terms are known as *coherences*. The density matrix ρ and its equation of motion can also describe statistical mixtures or *mixed states*. For example, if a random half of an ensemble of atoms were prepared in state $|\psi\rangle$ and the other half in a different state $|\chi\rangle$, this would correspond to $\rho = |\psi\rangle\langle\psi|/2 + |\chi\rangle\langle\chi|/2$. The case of $|\psi\rangle = |0\rangle$ and $|\chi\rangle = |1\rangle$ would then correspond to

$$\rho = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}. \tag{5.5}$$

This is very different to all atoms being prepared in an equal superposition state, e.g., $\rho = |\psi\rangle\langle\psi|$ where $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$; although the *populations* are the same, the *coherences* (associated with the existence of coherent superpositions between the basis states) are zero in the mixed state. As we are interested in the averaged dynamics of an ensemble of atoms subject to randomly occurring decay processes, which will quickly lead to a mixed state even if all atoms are initially prepared in, e.g., state $|0\rangle$, it is natural to choose a density matrix approach to do so.

5.2.1 Optical Bloch Equations

A standard approach to model an ensemble of two-level systems with decay employs the *optical Bloch equations* (OBE) to describe the time evolution of the density matrix ρ. Setting $\Delta = 0$ in Eq. (5.4), and extracting equations of motion for ρ_{11} and ρ_{01}, if we also include terms to describe decay of the populations and coherences, we obtain

$$\frac{d\rho_{11}}{dt} = i\frac{\Omega}{2}(\rho_{10} - \rho_{01}) - \Gamma\rho_{11}, \tag{5.6a}$$

$$\frac{d\rho_{01}}{dt} = i\frac{\Omega}{2}(\rho_{00} - \rho_{11}) - \frac{\Gamma}{2}\rho_{01}, \tag{5.6b}$$

where Γ is the *decay rate*. These are the optical Bloch equations [see, e.g., *The Quantum Theory of Light*, R. Loudon, 3rd edition (OUP 2000), Eq. 2.8.2], where we note $\dot{\rho}_{00} = -\dot{\rho}_{11}$, and $\dot{\rho}_{10} = \dot{\rho}_{01}^*$. A full derivation can be found in the appendix. Rather than solve this numerically, we will use an analytical solution (see appendix), and compare this to our quantum trajectory code.

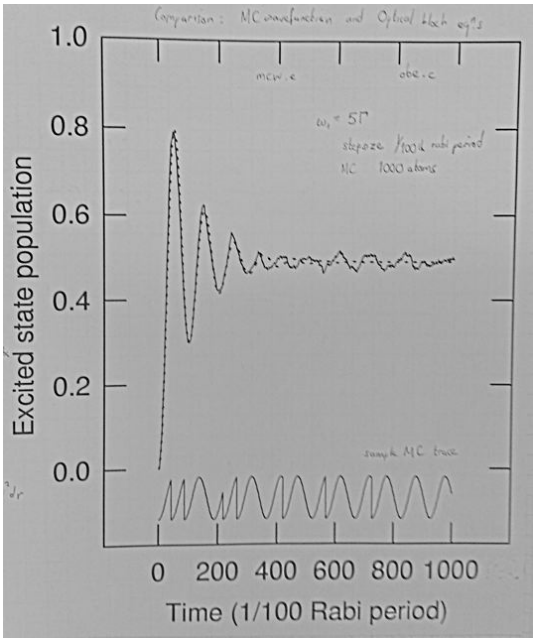


Figure 5.1: Comparison between the Monte Carlo wave function (MCW) and optical Bloch equation (OBE) approach for excited of an ensemble of qubit with a damping rate equal to one fifth of the Rabi frequency. The Monte Carlo wave function is an average of 1000 runs. A typical single run is shown offset below.

5.2.2 Monte Carlo wave function

Our first goal is to produce a plot similar to Fig. 1 by writing a quantum trajectory code and comparing the output, averaged over many realisations, to the analytical solution of the OBE. In the Monte Carlo wave function (MCW) method we solve a form of Schrödinger equation with a non-Hermitian *effective Hamiltonian* for the two-level system that incorporates the decay

$$i\hbar\frac{d|\psi\rangle}{dt} = \hat{H}_{\rm eff}|\psi\rangle,\ \hat{H}_{\rm eff} = \frac{\hbar}{2}\begin{pmatrix} \Delta & \Omega \\ \Omega & -\Delta - i\Gamma \end{pmatrix}, \tag{5.7}$$

and use a Monte Carlo algorithm to implement quantum jumps. The non-Hermitian nature of the effective Hamiltonian means that the norm-squared $\langle\psi|\psi\rangle$ of the wavevector will decay with time. The simplest approach is to evolve the wavevector by a very small time step δt to produce $|\psi'(n+\delta t)\rangle$, and to then generate a random number r between zero and one. If $r > \langle\psi'(t+\delta t)|\psi'(t+\delta t)\rangle$ then a quantum jump occurs and the qubit collapses to state $|0\rangle$. The wavevector must then be renormalised, whether or not a quantum jump event has occurred, to produce $|\psi(t+\delta t)\rangle$. The evolution, governed by Eq. (5.7), restarts using this final renormalised state as an initial condition, and the process is repeated.

The time evolution of the density matrix can then be built up by averaging:

$$\rho(n\,\delta t) = \frac{1}{N}\sum_{k=1}^{N}|\psi_k(n\,\delta t)\rangle\langle\psi_k(n\,\delta t)|, \tag{5.8}$$

where $|\psi_k\rangle$ describes the kth of N separate quantum trajectories, individually determined by the process described above.

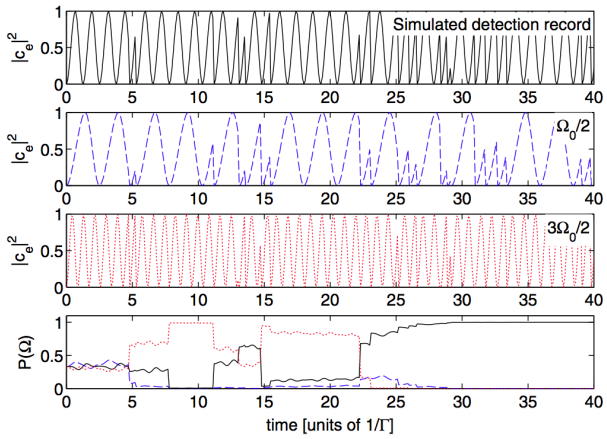


Figure 5.2: Estimation of the Rabi frequency from the intervals between detection events using Bayesian logic. From *Estimation of atomic interaction parameters by photon counting*, A H Kiilerich and K Mølmer, Phys. Rev. A **89**, 052110 (2014).

5.3 Numerical methods

Use the standard built-in ordinary differential equation solver to model Rabi oscillations of a qubit. Subsequently, you can incorporate quantum jumps using the Monte Carlo algorithm described above. The implementation of this Monte Carlo algorithm is discussed in more detail in Section 3 of K. Mølmer, Y. Castin and J. Dalibard, *Monte Carlo wave-function method in quantum optics* [J. Opt. Soc. Am. B **10**, 524 (1993)]. There is also a later paper by the same authors, K. Mølmer and Y. Castin, *Monte Carlo wave functions in quantum optics* [Quantum. Semiclass. Opt. **8**, 49 (1996)].

5.4 Milestone

Write your own MCW code and make a plot of damped Rabi oscillations for $\Gamma = 0.1\Omega$ as in Fig. 1. For example, run 1000 Monte Carlo realisations. Also plot residuals of the difference between the numerical and the analytical OBE solution discussed in the appendix. You should be able to show that the residuals are less than 1%.

5.5 Research with your Program

Plot the residuals between the OBE and MCW codes and investigate the convergence with number of runs. Investigate possible systematic differences and explore different renormalisation strategies. One interesting direction is to see what we can learn simple by looking at the time between 'collapse' events (i.e., clicks on a photon counter). Estimation of the Rabi frequency from the clicks is illustrated in Fig. 2.

5.5.1 Further ideas

- *EIT*: Extend your MCW code to a 3 level system (qutrit), and see if you can model the phenomenon of electromagnetically induced transparency (EIT).
- *Quantum gates*: Try to model the errors due to unwanted quantum jumps in a simulation of a CNOT quantum gate.
- *Bayesian analysis of photon statistics*: A. H. Kiilerich and K. Mølmer, *Estimation of atomic interaction parameters by photon counting*, Phys. Rev. A **89**, 052110 (2014).
- *Photon correlations*: calculate a second correlation function for light emitted by a single atom using the quantum regression theorem, see K. Mølmer and Y. Castin, *Monte Carlo wave functions in quantum optics*, Quantum Semiclass. Opt. **8**, 49 (1996).

Appendix: OBE Solution

Background

If we accept that spontaneous emission is an exponential decay process with decay rate Γ, such that in an ensemble of M atoms, all initially prepared in state $|1\rangle$, the expected number of atoms still in state $|1\rangle$ at time t is given by $Me^{-\Gamma t}$, then this is consistent with the excited state population of an atom decaying as $|b(t)|^2 = e^{-\Gamma t}|b(0)|^2$. This is in turn consistent with $b(t) = e^{-\Gamma t/2}b(0)$, and therefore with the coherences decaying as $a(t)b^*(t) = e^{-\Gamma t/2}a(0)b^*(0)$. Setting $\rho_{11} = |b|^2$, $\rho_{01} = ab^*$ and differentiating obtains

$$\frac{d\rho_{11}}{dt} = -\Gamma\rho_{11}, \qquad \frac{d\rho_{01}}{dt} = -\frac{\Gamma}{2}\rho_{01}, \tag{5.9}$$

which describes the decay part of Eq. (5.6).

We also introduce the *jump operator* which transfers the state of the system from being wholly or partially in the excited state $|1\rangle$ to be entirely within the ground state

$$\hat{C} = \sqrt{\frac{\Gamma}{2}}|0\rangle\langle 1| = \sqrt{\frac{\Gamma}{2}}\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \tag{5.10}$$

and thus, when followed by an appropriate renormalisation, describes the change associated with a quantum jump.

Master Equation

Standard practice is to combine the coherent and incoherent (decay) dynamics within a *master equation*

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[\hat{H},\rho] - \frac{1}{2}\left(\hat{C}^\dagger\hat{C}\rho + \rho\hat{C}^\dagger\hat{C} - 2\hat{C}\rho\hat{C}^\dagger\right), \tag{5.11}$$

which essentially consists of adding decay dynamics to Eq. (5.4). Any such master equation, for any Hamiltonian and where there can be a sum of similar decay-governing terms involving distinct jump operators $-\sum_k(\hat{C}_k^\dagger\hat{C}_k\rho + \rho\hat{C}_k^\dagger\hat{C}_k - 2\hat{C}_k\rho\hat{C}_k^\dagger)/2$ is said to be in *Lindblad form*. Such master equations, including Eq. (5.11), have the physically desirable properties of being trace preserving and completely

positive, which essentially means that the populations (in any basis) cannot become negative and will always sum to one.

While the evolution of a density matrix propagated by a master equation in Lindblad form is on these terms physically *reasonable*, whether it is more or less approximately *correct* depends on the dynamical situation being modelled (see D. Manzano, *A short introduction to the Lindblad master equation* [AIP Advances **10**, 025106 (2020)]). An assumption built into such a master equation description is that the system is coupled only weakly to the environment, and that there is no buildup of correlations, i.e., no memory in the environment of the system interacting with it which can then feed back into the system. This, however, is typically quite well fulfilled by isolated atomic systems coupled via spontaneous decay to the infinite-dimensional free electromagnetic field.

Solving the Differential Equation

The optical Bloch equations are the equations of motion for the matrix elements of ρ as given by Eq. (5.11):

$$\frac{d\rho_{01}}{dt} = \frac{d\rho_{10}^*}{dt} = \frac{i}{2}\,\Omega(\rho_{00} - \rho_{11}) - \frac{\Gamma}{2}\rho_{01}, \tag{5.12a}$$

$$\frac{d\rho_{11}}{dt} = i\frac{\Omega}{2}(\rho_{01} - \rho_{10}) - \Gamma\rho_{11}, \tag{5.12b}$$

$$\frac{d\rho_{00}}{dt} = -\,i\frac{\Omega}{2}(\rho_{01} - \rho_{10}) + \Gamma\rho_{11}. \tag{5.12c}$$

We observe that $\dot\rho_{00} + \dot\rho_{11} = 0$, demonstrating that total probability is indeed conserved. We therefore define new quantities $\rho_{\text{diff}} = \rho_{00} - \rho_{11}$ and $\rho_{\text{coh}} = \text{Im}(\rho_{01}) = (\rho_{01} - \rho_{10})/2i$. With this we can rationalise the above into two coupled differential equations:

$$\frac{d\rho_{\text{coh}}}{dt} = \frac{\Omega}{2}\rho_{\text{diff}} - \frac{\Gamma}{2}\rho_{\text{coh}}, \tag{5.13a}$$

$$\frac{d\rho_{\text{diff}}}{dt} = -\,2\Omega\rho_{\text{coh}} + \Gamma(1 - \rho_{\text{diff}}), \tag{5.13b}$$

Making ρ_{coh} the subject of Eq. (5.13b), we can substitute this into Eq. (5.13a). We can then substitute the result into the expression resulting from differentiating Eq. (5.13b), yielding

$$\frac{d^2\rho_{\text{diff}}}{dt^2} = -\frac{3\Gamma}{2}\frac{d\rho_{\text{diff}}}{dt} - \left(\Omega^2 + \frac{\Gamma^2}{2}\right)\rho_{\text{diff}} + \frac{\Gamma^2}{2}. \tag{5.14}$$

This is an inhomogeneous second-order linear differential equation, and can be generally solved using standard methods to give

$$\begin{aligned}\rho_{\text{diff}} = e^{-3\Gamma t/4}\Bigg[&A\cos\left(\sqrt{\Omega^2 - \frac{\Gamma^2}{16}}\,t\right)\\ &+B\sin\left(\sqrt{\Omega^2 - \frac{\Gamma^2}{16}}\,t\right)\Bigg] + \frac{\Gamma^2}{2\Omega^2 + \Gamma^2}.\end{aligned} \tag{5.15}$$

Assuming the initial condition $\rho = |0\rangle\langle 0|$ at $t = 0$, it also follows that $\rho_{\text{diff}} = 1$, $\rho_{\text{coh}} = 0$, and that therefore $d\rho_{\text{diff}}/dt = 0$ at $t = 0$. The integration constants are then given by

$$A = \frac{2\Omega^2}{2\Omega^2 + \Gamma^2}, \qquad B = \frac{3\Gamma A}{\sqrt{16\Omega^2 - \Gamma^2}}. \tag{5.16}$$

Noting that $\rho_{11} = (1 - \rho_{\text{diff}})/2$, this probability becomes

$$\rho_{11} = \frac{\Omega^2}{2\Omega^2 + \Gamma^2} \left\{ 1 - e^{-3\Gamma t/4} \left[\cos\left(\sqrt{\Omega^2 - \frac{\Gamma^2}{16}} t \right) + \frac{3\Gamma}{\sqrt{16\Omega^2 - \Gamma^2}} \sin\left(\sqrt{\Omega^2 - \frac{\Gamma^2}{16}} t \right) \right] \right\}. \tag{5.17}$$

For completeness, note that $\rho_{00} = 1 - \rho_{11}$, and that ρ_{coh} can be determined by substituting Eq. (5.15) into Eq. (5.13b). Finally, defining $\rho_{\text{real}} = (\rho_{01} + \rho_{10})/2$, Eq. (5.12a) yields $d\rho_{\text{real}}/dt = -\Gamma\rho_{\text{real}}/2$, which is solved by $\rho_{\text{real}} = De^{-\Gamma t/2}$; taking $\rho = |0\rangle\langle 0|$ as the initial condition then means the integration constant $D = 0$. The off-diagonal terms ρ_{01} and ρ_{10} can then be deduced from ρ_{real} and ρ_{coh}.

Comparing to Monte Carlo

The simplest approach essentially follows K. Mølmer and Y. Castin, *Monte Carlo wave functions in quantum optics* [Quantum. Semiclass. Opt. **8**, 49 (1996)] section 2.2. Using Eq. (5.7), propagate the wavevector for a very short time δt. If the initial state is given by $|\psi(t)\rangle$, then the result of this time evolution is given by definition as $|\psi'(t + \delta t)\rangle = \exp(-i\hat{H}_{\text{eff}}\,\delta t\,/\hbar)|\psi(t)\rangle$. However, if δt is very small compared to the relevant dynamical timescales (which in the present context means $\delta t \ll \Gamma^{-1}$, Ω^{-1}), we can approximate

$$|\psi'(t + \delta t)\rangle \approx \left(1 - \frac{i}{\hbar}\hat{H}_{\text{eff}}\,\delta t\right)|\psi(t)\rangle. \tag{5.18}$$

This is just the result of expanding the exponential to first order in δt.

Then, choose a random number r between 0 and and 1. If r is less than the norm-squared $\langle\psi'(t + \delta t)|\psi'(t + \delta t)\rangle$, then there is no decay and we simply renormalise. Hence, in this case:

$$|\psi(t + \delta t)\rangle = \frac{|\psi'(t + \delta t)\rangle}{\sqrt{\langle\psi'(t + \delta t)|\psi'(t + \delta t)\rangle}}. \tag{5.19}$$

If the random number is greater than or equal to $\langle\psi'(t + \delta t)|\psi'(t + \delta t)\rangle$ then there is a decay. We apply the jump operator, and then renormalise. Hence, in this case:

$$|\psi(t + \delta t)\rangle = \frac{\hat{C}|\psi'(t + \delta t)\rangle}{\sqrt{\langle\psi'(t + \delta t)|\hat{C}^\dagger\hat{C}|\psi'(t + \delta t)\rangle}}. \tag{5.20}$$

We repeat the process iteratively to generate $|\psi(t + 2\,\delta t)\rangle$, $|\psi(t + 3\,\delta t)\rangle$, and so on, until some chosen total evolution time is reached. Carrying out multiple such total time evolutions, we can build up an averaged density matrix according to Eq. (5.8), and compare for example the resulting ρ_{11} with the result of Eq. (5.17).

Alternatively (see R. Dum, P. Zoller and H. Ritsch, *Monte Carlo simulation of the atomic master equation for spontaneous emission* [Phys. Rev. A **45**, 4879 (1992)] section II C, noting also figures 1 and 2) one can start off by choosing a random number r between 0 and 1. Then, propagate the wavevector using Eq. (5.7) for a time Δt, such that at this time $1 - \langle\psi'(t + \Delta t)|\psi'(t + \Delta t)\rangle$ is equal to (or, in a practical sense, very slightly greater than) r (this necessarily involves monitoring the value of $\langle\psi'|\psi'\rangle$). At this point

apply the jump operator, the interpretation being that this is when a quantum jump takes place, and renormalise. Hence

$$|\psi(t+\Delta t)\rangle = \frac{\hat{C}|\psi'(t+\Delta t)\rangle}{\sqrt{\langle\psi'(t+\Delta t)|\hat{C}^\dagger\hat{C}|\psi'(t+\Delta t)\rangle}}. \tag{5.21}$$

Similarly, we repeat the process iteratively until some total evolution time of interest, and build up the density matrix by averaging (noting that the wavevectors must be renormalised to do so).

Chapter 6

Quantum optimization and applied algorithms

6.1 Introduction

Quantum computing has advanced significantly in recent years, and while no conclusive quantum advantage has been demonstrated at the time this document has been written, real world applications appear to be on the horizon. There are several potential areas for near term applications of quantum computing, of which optimization is one promising area. For this chapter, we will investigate how quantum tools can be used to solve classical optimization problems.

The first step in quantum optimization is to map the problem to a physical model which can be encoded on a quantum computer. In practice, the most common choice is the Ising model, which is known to be able to map all NP-hard[1] problems:

$$H_{\text{Ising}} = \sum_{k=1}^{n} \sum_{j=k+1}^{n} J_{kj} \sigma_k^z \sigma_j^z + \sum_{j=1}^{n} h_j \sigma_j^z \tag{6.1}$$

with $\sigma_j^z = \left(\bigotimes_{k=1}^{j-1} I_2 \right) \otimes \sigma^z \otimes \left(\bigotimes_{k=j+1}^{n} I_2 \right)$, where

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{6.2}$$

is the 2×2 identity matrix and

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{6.3}$$

The $\otimes$ symbol indicates a tensor product, which is reviewed in Appendix 1 along with the $\bigotimes_{k=1}^{j-1}$ notation. The coupling strengths J_{kj} form a matrix, and the field strengths h_j form a vector specifying a particular Ising model. Optimization problems can be encoded in the matrix J and the vector h in ways which will be explained in section 6.3. Since H_{Ising} is a Hamiltonian and the operators σ_j^z have no physical dimensions,

[1] Non-deterministic polynomial-time hard (often mistakenly referred to as 'non-polynomial hard'). The exact definition of this problem class is not important for this project, but this is the 'hardest' class of conventional optimisation problems, and it is suspected (but not proven) that no efficient algorithms exist for this class of problems.

The J_{kj}'s and h_j's should have the physical dimensions of an energy. However, it is customary, in this area of Physics, to treat this Hamiltonian as having no physical dimensions either, which simplifies the description. Accordingly, the J_{kj}'s and h's will be assumed to be pure numbers in the following.

When J and h encode a problem, they encode it in such a way that the minimum energy with respect to H_{Ising} is the best solution. When represented as a matrix, H_{Ising} is a $2^n \times 2^n$ diagonal matrix where each diagonal entry gives the energy of a classical bitstring from $|00...0\rangle$ to $|11...1\rangle$. (By classical bitstring, we mean a state which is labelled by a binary number and can represent the state of a classical system.) Calculating each of the entries on the diagonal, hence the corresponding energies, might not require much computational efforts. However, writing the entire matrix out in this way is only possible for very small problems. To see why, consider a 100-bit optimization problem, which is actually still quite small compared to what one typically encounters in the 'real world'. The number of energies will be $2^{100} \sim 10^{30}$. In Physics terms, this is more than the number of water molecules in a large tanker truck full of water. Counting the molecules in such a truck is clearly not possible. Finding the solution to such an optimization problem by just calculating all the energies and taking the lowest is similarly impossible: even if you had a fast computer which could check one solution every nanosecond and started at the beginning of the universe, you would currently be less than one thousandth of the way to being done.

6.2 The Quantum Adiabatic Algorithm

Clearly, there are better ways (i.e., *algorithms*) to solve optimization problems than just checking every possibility. Also, in most cases you just need a good solution, not the *best* solution. Coming up with clever classical ways to solve optimization problems (e.g., genetic algorithms, swarm algorithms and many others) is a huge area of active research, but these problems could also potentially be solved with the help of quantum mechanics. So far, all we have shown is how to state an optimization problem as a Hamiltonian, we have not shown how to add quantum effects. Consider the effect of adding single bit flip operations to H_{Ising}, to obtain the Hamiltonian of a *transverse field Ising model*:

$$H(t) = -A(t)\sum_j \sigma_j^x + B(t)H_{\text{Ising}} \tag{6.4}$$

where $A(t)$ and $B(t)$ are time-dependent controls, and $\sigma_j^x = \left(\bigotimes_{k=1}^{j-1} I_2\right) \otimes \sigma^x \otimes \left(\bigotimes_{k=j+1}^{n} I_2\right)$ where

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{6.5}$$

is the usual Pauli matrix. (The functions $A(t)$ and $B(t)$ are often given physical units of energies, so as to give $H(t)$ the physical dimensions of an energy; however, for simplicity we won't do this. Accordingly, we will take the "time" t to be dimensionless, too.)

If $B = 0$ and $A > 0$, then the ground state $|\phi_0(A > 0, B = 0)\rangle$ of Eq. (6.4) will just be the highest energy state of each σ^x individually. In other words,

$$|\phi_0(A > 0, B = 0)\rangle = \bigotimes_{j=1}^{n} |+\rangle\,, \tag{6.6}$$

where

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{6.7}$$

On the other hand, when $B > 0$ and $A = 0$, then by definition $|\phi_0(A = 0, B > 0)\rangle$ will be the solution to the optimization problem which has been encoded into J and h in Eq. (6.1).[2]

It is important to note that $H(t)$ acts and $|\phi_0\rangle$ lives in a space of 2^n-component column vectors. This space can be spanned by the 2^n-component column vectors $|0\rangle$, $|1\rangle$, $|2\rangle$,..., $|2^n - 1\rangle$, where $|i\rangle$ is a column vector of zeros except the i-th element which is 1. In terms of these vectors,

$$|\phi_0(A > 0, B = 0)\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle \,. \tag{6.8}$$

Note, also that the exact ground state of H_{Ising} is one of these vectors.

Based on the statements in the previous paragraph, and some basic quantum mechanics, there is a general quantum algorithm for optimization problems, assuming a physical system which can implement the transverse field Ising model with sufficient controls is available. The *adiabatic theorem of quantum mechanics* states that a quantum system initially in the ground state of a time-dependent Hamiltonian remains in the ground state of the Hamiltonian as this operator changes, *provided the change is slow enough* (some other technical criteria — which transverse Ising systems always satisfy — must also be met). If we start in $|\phi_0(A > 0, B = 0)\rangle$ and change A and B *slowly enough* until we end with $B > 0$ and $A = 0$, we will find a final state which is almost equal to $|\phi_0(A = 0, B > 0)\rangle$, which is the solution to our problem. [It would normally be equal in the limit where $A(t)$ and $B(t)$ vary infinitely slowly. For finite time scales, the final state will be a superposition of $|\phi_0(A = 0, B > 0)\rangle$ with a number of other states, but for a sufficiently slow variation $|\phi_0(A = 0, B > 0)\rangle$ will have a much larger weight in this superposition than any of the other states.]

Solving an optimization problem along these lines would involve implementing $H(t)$ on a quantum platform, start with $B(t) = 0$ and $A(t) > 0$ and the state $|\phi_0(A > 0, B = 0)\rangle$ at $t = 0$, slowly vary $A(t)$ and $B(t)$ from $t = 0$ to $t = t_{\text{fin}}$, and measure the state at that time in the $|j\rangle$-basis (i.e., use a measuring apparatus which would find the system in one of the $|j\rangle$ states).

This techniques is known as *adiabatic quantum computing* (AQC) or the *quantum adiabatic algorithm* (QAA). Note that, while AQC definitely works, if you change the Hamiltonian slowly enough, it won't be useful in practice if it takes much longer than classical computers solving the same problem. In practice we would still be happy if it doesn't find the absolute best solution but finds a better solution than classical computers could obtain, or if it finds a solution a classical computer can also obtain but finds it much faster.

6.3 Encoding Optimization problems into Hamiltonians

To explain how optimization problems can be mapped to the J and h terms in Eq. (6.1), we will consider a simple case, known as the maximum independent set problem. The easiest way to think of maximum independent set is to think of a *graph* with vertices connected by edges, for example, the one shown in Fig. 6.1. The goal of finding the maximum independent set[3] is to colour in as many vertices as possible such that no two coloured vertices share an edge.

[2] If there is a tie for the best solution, then $|\phi_0(A = 0, B > 0)\rangle$ could be a superposition of the corresponding states.

[3] Not to be confused with *maximal* independent set, which is not a hard problem. A maximal independent set is one that cannot be extended by colouring one more vertex without having two vertices sharing an edge. Fig. 6.1(b) is an example of maximal independent set.

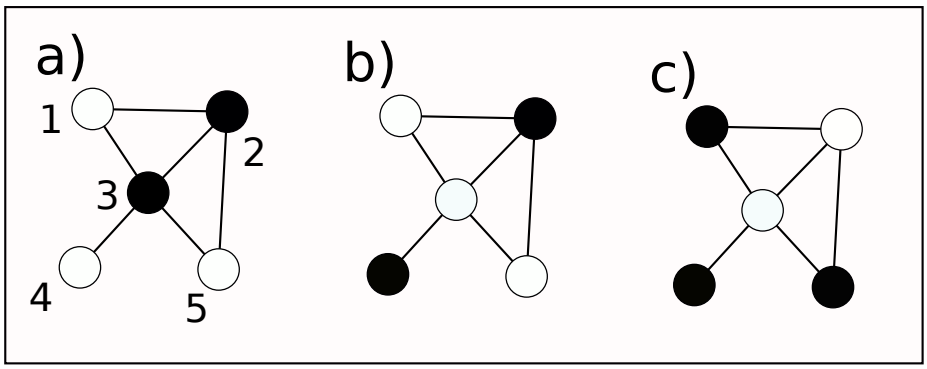


Figure 6.1: A graph with different sets of vertices coloured in black. a) A set of vertices which are not independent. b) A set of vertices which are independent, but is not the maximum independent set. c) The maximum independent set of vertices for this graph. The numbering scheme used in Eq. (6.10) is shown in a).

While finding maximum independent set may seem like a silly colouring problem, it could actually come up in the real world. For instance, imagine a stock broker wants to build a large diverse portfolio of stocks, including pairs of stocks whose values are likely to be highly correlated (maybe stocks of two different companies which produce the same product or rely on the same resource). This problem maps to a graph, where each vertex is a stock and the edges represent high correlation between stocks. Finding the largest portfolio of uncorrelated stocks in this example is exactly the maximum independent set problem.

To express the maximum independent set problem as an Ising model, we first need to think how to encode the concept of an independent set. To do this, first construct a two qubit Hamiltonian which penalizes (higher energy) the $|11\rangle$ state relative to the $|00\rangle$, $|01\rangle$, and $|10\rangle$ states. This constraint is achieved by

$$H_2 = -\sigma^z \otimes I_2 - I_2 \otimes \sigma^z + \sigma^z \otimes \sigma^z = -\sigma_1^z - \sigma_2^z + \sigma_1^z \sigma_2^z, \tag{6.9}$$

where the second form drops the identity operators and the $\otimes$ symbols for brevity. This Hamiltonian ensures the lowest energy state does not contain two ones.

Consider a graph defined by the upper triangular adjacency matrix M, such that a 1 entry indicates an edge between corresponding vertices and a 0 entry indicates no edge (see Appendix 2). If the qubits represent vertices, and the qubit state $|1\rangle$ means that this vertex is coloured, we can use operators like the Hamiltonian of Eq. (6.9) to penalise adjacent vertices that are both coloured. By defining $J = M$ and $h_k = -\sum_j (M_{kj} + M_{jk})$, a Hamiltonian of the form of Eq. (6.1) will enforce independent sets. In other words, when expressed as a bitstring, if the ones in a state form an independent set on the graph defined by M, they will get one energy, but they will get a higher energy if they do not. Try writing it out by hand for a three qubit example, such as o–o–o, see figure 6.3 and Appendix 3.

Finally, to construct a *maximum* independent set Hamiltonian, rather than just an independent set Hamiltonian, we need to give a lower energy to states with more qubits in the $|1\rangle$ state. To do this, we now set $h_k = -(\sum_j M_{kj} + M_{jk}) + \kappa$ (a Greek kappa, not k) where $0 < \kappa < 1$.

There are many more complicated ways of encoding optimization problems into Ising Hamiltonians, but maximum independent set is one of the simplest, and gives a flavour of how this process works.

6.4 Time-dependent Hamiltonian simulation

For the milestone, you will be simulating the quantum adiabatic algorithm solving a small instance of maximum independent set. To start with, consider the graph depicted in Fig. 6.1, which is defined by the following adjacency matrix (see Appendix 2)

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{6.10}$$

First, construct H_{Ising} for this graph, and verify that the lowest energy state is indeed the maximum independent set $|10011\rangle = |19\rangle$. In vector form, this is a column vector of length 2^5 with a one in the 19^{th} position $(18\,\text{zeros}\,, 1, 13\,\text{zeros})^T$ (see Appendix 3).

Once you have verified that you have constructed the classical problem Hamiltonian correctly, the next step is to simulate an adiabatic algorithm solving the problem. To simulate the adiabatic algorithm, you have to simulate evolution with a time dependent Hamiltonian. How the states evolve in time can be described as governed by an evolution operator $U(t,0)$ so that $|\psi(t)\rangle = U(t,0)\,|\psi(t=0)\rangle$. This evolution operator can be written as an infinite product of matrix exponentials: For an evolution from $t = 0$ to $t = t_{\max}$,

$$U(t_{\max},0) = \lim_{q\to\infty} \mathcal{T} \prod_{j=1}^{q} \exp\left\{-i\frac{t_{\max}}{q}\, H\left(\frac{j\,t_{\max}}{q}\right)\right\}, \tag{6.11}$$

where $\mathcal{T}$ indicates that the product is time ordered. If we are willing to accept some numerical error, we can set q to be large but not strictly infinite, thus obtaining a discrete version of the continuous-time evolution in a form convenient for numerical calculations. If we start in state $|\psi(t=0)\rangle$, then the state at time $k\,t_{\max}/q$ can be written as

$$\left|\psi\left(t = \frac{k\,t_{\max}}{q}\right)\right\rangle \approx \mathcal{T} \prod_{j=1}^{k} \exp\left\{-i\frac{t_{\max}}{q}\, H\left(\frac{j\,t_{\max}}{q}\right)\right\} |\psi(t=0)\rangle\,, \tag{6.12}$$

where we use the convention that the right-hand most term of the product is that with $j = 1$ and the left-hand most that with $j = k$ (thus term j acts on the left of term $j-1$, which itself acts on the left of term $j-2$, etc.). Mathematically, exponentiating a matrix is well defined (at least for square matrices) through the power series expansion of the exponential function. Fortunately, Python has a very efficient function to exponentiate matrices: `scipy.linalg.expm`.

6.5 Milestone project

For the milestone you will simulate an adiabatic algorithm which solves the maximum independent set problem on the graph defined by the adjacency matrix given in Eq. (6.10), using Eq. (6.12). For this exercise, set $A(t) = 1 - t/t_{\max}$ and $B(t) = t/t_{\max}$ where $t_{\max}$ is the total runtime of the algorithm. Plot the probability of 'success' (i.e., the probability that the measured result is the maximum independent set) versus $t/t_{\max}$ for several different values of $t_{\max}$. This plot should include both values of $t_{\max}$ for which the success probability is barely changed from random guessing, values for which the success probability

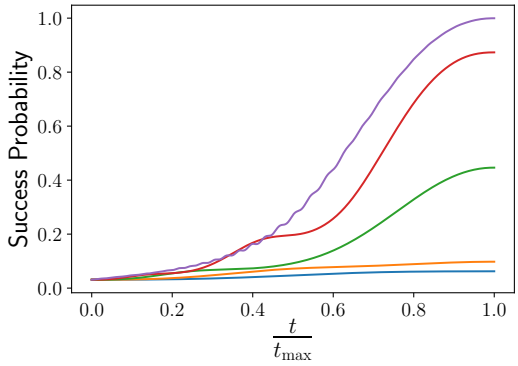


Figure 6.2: Success probability versus time for solving the maximum independent set problem on the graph given in Fig. 6.1 and Eq. (6.10) for different total runtimes: $t_{\max} = 1$ (blue), $t_{\max} = 2$ (orange), $t_{\max} = 5$ (green), $t_{\max} = 10$ (red), $t_{\max} = 100$ (purple). This plot was created using $\kappa = 1/2$.

is greater than 0.95, and at least one value where the probability is somewhere in between. Find an acceptable value of q by trial and error (q does not have to scale with $t_{\max}$). You may aim at reproducing Fig. 6.2, which shows the success probabilities versus time for different total runtimes.

6.6 Milestone extensions

There are a wide variety of possible extensions which would be appropriate, including many not listed here. All of the listed ones are are suitable for Level 3 computer projects, but those which are likely to be more challenging are marked with a †):

- Map other problems such as maximum 2 satisfiability using the mapping from this experimental paper [1] and † more complicated problems such as those described in [2].
- Apply sparse matrix techniques to simulate adiabatic quantum computing on larger systems using `scipy.sparse.linalg.expm_multiply`, possibly also looking at problems beyond maximum independent set.
- Investigate the effect of different annealing schedules, in other words different functional forms of $A(t)$ and $B(t)$ rather than just the linear example used for the milestone, for example:

$$A(\alpha, s', t) = \frac{1 - t/t_{\max}}{\alpha + (1-\alpha)\,(1 - t/t_{\max})\,(1-s')^{-1}}\,, \tag{6.13a}$$

$$B(\alpha, s', t) = \frac{t/t_{\max}}{\alpha + (1-\alpha)\,t/t_{\max}\,(s')^{-1}}\,, \tag{6.13b}$$

  where $0 < s' < 1$ controls where the algorithm slows down (it translates to a value of $t/t_{\max}$ in the original $A = t/t_{\max}$, $B = 1 - t/t_{\max}$ parametrization) and $0 < \alpha \leq 1$ controls how much it slows down. How does this affect the performance? Is it better to slow down when the energy gap between the ground and first excited state is large, or when it is small?
- Reproduce some of the results obtained in early proof-of-principle problems for adiabatic quantum computing [3]. Note that some of these problems are not formulated as Ising models.

- Explore QAOA (quantum approximate optimisation algorithm) rather than adiabatic protocols. In QAOA, $-\sum_j \sigma_j^x$ and H_{Ising} are applied in an alternating fashion rather than simultaneously — see Refs. [4, 5].
- Explore quantum walks on graphs [6](† possibly also including marked states and decoherence [7]). Quantum walks have recently been considered as a tool to solve optimization problems [8], but more results are known for walks on graphs.
- † Write code to perform path integral quantum annealing (PIQA) [9], which can be applied to much larger problems than the matrix simulation methods used here. Potentially reproduce a version of the PIQA plots in [10].

Warning: the use of the terms *adiabatic quantum computing* versus *quantum annealing* is not standardized in the literature, and different authors use these terms differently. Read carefully, to understand what is actually being done in a paper.

Appendix 1: A review of tensor products

Tensor products provide a powerful mathematical tool for a range of Physics models. The basic idea behind a tensor product is that it separates operations occurring on different degrees of freedom of a physical system. Tensor products are required to complete the milestone. Note that the Python function `numpy.kron` performs tensor products. Terminology in Python for matrix operations is not the same as usually used in Physics, and you should always check that it is actually doing what you want. In particular, matrix multiplication using * is usually elementwise; you need to use something like `numpy.dot` to do what we consider normal matrix multiplication.

Python has functions to perform all the operations you need (use them, they are very efficient implementations). However, it is necessary to understand how they work, in order to check that your code is correct. Effectively, the action of the tensor product $a \otimes b$ is to replace each element of a, a_{jk}, with the matrix $a_{jk} \times b$. Since each element is replaced by a matrix, the total size of the resulting matrix is $\text{length}(a) \times \text{length}(b)$. Unlike standard matrix multiplication, where the multiplication dimension needs to match, tensor products can be performed between pairs of matrices (or vectors) of any size. Like all types of multiplication, tensor products are *associative*, $a \otimes b \otimes c = a \otimes (b \otimes c) = (a \otimes b) \otimes c$.

As an example, consider the smallest non-trivial matrix tensor product, which is a tensor product of two 2×2 matrices, such as matrices representing quantum mechanical operators acting on qubits (blue ket vectors are added as a visual aid) :

$$a \otimes b = \begin{matrix} \\ \langle 0| \\ \langle 1| \end{matrix} \begin{matrix} \begin{matrix} |0\rangle & |1\rangle \end{matrix} \\ \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \end{matrix} \otimes \begin{matrix} \\ \langle 0| \\ \langle 1| \end{matrix} \begin{matrix} \begin{matrix} |0\rangle & |1\rangle \end{matrix} \\ \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix} \end{matrix} =$$

$$\begin{matrix} \\ \langle 00| \\ \langle 01| \\ \langle 10| \\ \langle 11| \end{matrix} \begin{matrix} \begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix} \\ \begin{pmatrix} a_{00}b_{00} & a_{00}b_{01} & a_{01}b_{00} & a_{01}b_{01} \\ a_{00}b_{10} & a_{00}b_{11} & a_{01}b_{10} & a_{01}b_{11} \\ a_{10}b_{00} & a_{10}b_{01} & a_{11}b_{00} & a_{11}b_{01} \\ a_{10}b_{10} & a_{10}b_{11} & a_{11}b_{10} & a_{11}b_{11} \end{pmatrix} \end{matrix}. \tag{6.14}$$

The action of the tensor product, denoted by $\otimes$, is to combine the spaces in which these two operators act. In this way, a composite representation can be constructed where states of the two-qubit system can be written as single state vectors: $|\alpha\beta\rangle = |\alpha\rangle \otimes |\beta\rangle$.

Each degree of freedom can be addressed independently using the tensor product, for instance, if we replace matrix a in Eq. (6.14) with a 2×2 identity matrix, the resulting matrix,

$$\begin{array}{c} \\ \langle 00| \\ \langle 01| \\ \langle 10| \\ \langle 11| \end{array} \begin{array}{c} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{pmatrix} b_{00} & b_{01} & 0 & 0 \\ b_{10} & b_{11} & 0 & 0 \\ 0 & 0 & b_{00} & b_{01} \\ 0 & 0 & b_{10} & b_{11} \end{pmatrix} \end{array}, \tag{6.15}$$

can only flip or apply phase differences to the second qubit. Similarly, if instead b were replaced by the 2×2 identity matrix the resulting matrix,

$$\begin{array}{c} \\ \langle 00| \\ \langle 01| \\ \langle 10| \\ \langle 11| \end{array} \begin{array}{c} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{pmatrix} a_{00} & 0 & a_{01} & 0 \\ 0 & a_{00} & 0 & a_{01} \\ a_{10} & 0 & a_{11} & 0 \\ 0 & a_{10} & 0 & a_{11} \end{pmatrix} \end{array}, \tag{6.16}$$

can only flip or apply phase differences to the first qubit.

This generalises for tensor products of more degrees of freedom. The matrices quickly become unwieldy to write out explicitly, but computers can store and manipulate very large matrices. An operation a on the jth qubit of an n qubit system can be written $a_j = (\bigotimes_{k=1}^{j-1} I_2) \otimes a \otimes (\bigotimes_{k=j+1}^{n} I_2)$ where I_2 is a 2×2 identity matrix and $\bigotimes_{k=1}^{j-1}$ indicates repeated tensor products in the same way $\prod_{k=1}^{j-1}$ would represent repeated multiplication. In particular $\bigotimes_{k=n-q+1}^{n} I_2 = I_2 \otimes I_2 ... (\text{total of q times}) ... \otimes I_2 = I_2^{\otimes q}$.

Any Hermitian operator of size 2^n can be constructed from sums and products of Pauli operations on different sites $\sigma_j^{\{x,y,z\}}$ plus the identity operation. The operational meaning of $\sigma_j^{\{x,y,z\}}$ is to perform a Pauli $\{x, y, z\}$ operation on the jth qubit while doing nothing (the identity) to the others. Simultaneous operations on qubits j and k where $j \neq k$ can be represented as $\sigma_j^{\{x,y,z\}} \sigma_k^{\{x,y,z\}}$. Pauli operators acting on different qubits commute with each other, i.e., $[\sigma_j^{\{x,y,z\}}, \sigma_{k \neq j}^{\{x,y,z\}}] = 0$.

Coding tip: write a function (or three) to create $\sigma_j^{\{x,y,z\}}$ once, and call it in later functions, rather than try to 'hard code' the creation of each term from tensor products. Such functions can be written using less than 10 lines of code, if written well.

Appendix 2: Graphs and adjacency matrices

Graphs are formal mathematical objects consisting of vertices (usually represented visually by circles) connected by edges (usually represented by line segments). Graphs can be drawn on a two dimensional plane by choosing (x, y) coordinates for the vertices and drawing appropriate edges between them. This positioning of the vertices is important for human understanding but is not part of the definition of the graph itself. Rearranging the vertices keeps the graph the same. The power of graphs is that they allow

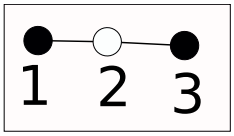


Figure 6.3: A simple graph with three vertices, coloured to show maximum independent set and numbered to match Eq. (6.18).

an abstract representation of the relationships between different entities. A family tree, for instance, is one type of graph which provides information about ancestry and lineage. In general graphs can have different types of edges (relationships between vertices) or weights assigned to each edge, and can either be directed (the relationship has a direction, e.g., parent to child in a family tree) or undirected. The maximum independent set problem is defined on a very simple class of graphs where edges are undirected and for which there is only one type of (unweighted) edge, any two vertices can only be unconnected to each other or connected by a single edge, and vertices are not allowed to connect to themselves (no 'self loops').

While graphs provide a powerful tool for humans to visualize real problems and systems, they are not a way of representing information which computers can easily operate on directly. Fortunately, since a graph is defined by the connections between vertices rather than the positioning of those vertices, a graph can efficiently be represented by a matrix, and matrices can be efficiently manipulated by computers. Consider the simple graphs which are used to define maximum independent sets. Each pair of vertices either share an edge or do not share an edge. The information contained in a graph of this kind with n vertices can be expressed as $n\,(n-1)$ binary variables (numbers which can only be 0 or 1). In practice however, it is more convenient to organize these variables into an $n \times n$ *adjacency matrix*, an array of numbers which has ones in the $(j, k > j)$ position if there is an edge between the vertices j and k and has zeros everywhere else. (Depending on the application, adjacency matrices may be defined with entries below the diagonal, too, for traversing edges from k to j.) The upper triangular definition is convenient for application to maximum independent set Hamiltonians, but we could have used either convention.

To construct an adjacency matrix, one must assign labels in some order to the vertices, as was done in Fig. 6.1(a). This ordering is arbitrary, but once chosen must be used consistently. The matrix in Eq. (6.10) is reproduced below, with the rows and columns labelled as a reference

$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}. \tag{6.17}$$

As an example, if we look in the first row and second column, there is a 1 entry in this matrix, and indeed, there is an edge between vertex 1 and vertex 2 in Fig. 6.1. On the other hand, there is a zero in the first row and fourth column, and there is indeed no edge shared between vertex 1 and vertex 4.

Appendix 3: Examples of Hamiltonians

Given a graph, the vector h and matrix J which encode the problem in the Ising Hamiltonian can be defined by a repeated use of Eq. (6.9). Mathematically, this Hamiltonian can be expressed as a $2^n \times 2^n$ matrix for n vertices, although for large problems this matrix will not be practical to construct simply because of its size. Already for the maximum independent set problem defined in the main text, the matrix will be $2^5 \times 2^5 = 32 \times 32$, rather large to write out explicitly on the page.

Instead, consider a three qubit problem defined by the graph depicted in Fig. 6.3, which has the adjacency matrix

$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \end{array}. \tag{6.18}$$

The Hamiltonian for the maximum independent set is therefore represented by the following arrays:

$$J = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \end{array}, \qquad h = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \\ \begin{pmatrix} -1+\kappa \\ -2+\kappa \\ -1+\kappa \end{pmatrix} \end{array}. \tag{6.19}$$

Straightforward calculations show that the *p*roblem Hamiltonian for this problem (the 8×8 Ising Hamiltonian) takes the following form,

$$H_{\text{problem}} = \begin{array}{cc} & \begin{array}{cccccccc} |0\rangle & |1\rangle & |2\rangle & |3\rangle & |4\rangle & |5\rangle & |6\rangle & |7\rangle \\ |000\rangle & |001\rangle & |010\rangle & |011\rangle & |100\rangle & |101\rangle & |110\rangle & |111\rangle \end{array} \\ \begin{array}{cc} \langle 0| & \langle 000| \\ \langle 1| & \langle 001| \\ \langle 2| & \langle 010| \\ \langle 3| & \langle 011| \\ \langle 4| & \langle 100| \\ \langle 5| & \langle 101| \\ \langle 6| & \langle 110| \\ \langle 7| & \langle 111| \end{array} & \begin{pmatrix} -2+3\kappa & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2+\kappa & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2+\kappa & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2-\kappa & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2+\kappa & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2-\kappa & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2-\kappa & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6-3\kappa \end{pmatrix} \end{array}. \tag{6.20}$$

Note that this matrix is diagonal, because it has been constructed entirely from σ_z operators plus identity terms. The σ_x terms in the transverse field in Eq. (6.4) will appear as off-diagonal entries (see below). The bras and kets labelling the rows and columns depict two different ways of expressing the states, either by listing the states of each of the three qubits (blue), or by converting this state to a decimal number (green). For visual clarity, the lowest energy eigenvalue has been coloured orange (recall that $0 < \kappa < 1$). By examining the problem Hamiltonian, we can observe that the maximum independent set is the state $|101\rangle \equiv |5\rangle \equiv (0,0,0,0,1,0,0)^T$, where the last expression is the state expressed as a vector.

Finally, the adiabatic evolution Hamiltonian can be constructed. Setting $\kappa = 1/2$ and plugging into

Eq. (6.4). This yields

$$H(t) = -A(t)\sum_j \sigma_j^x + B(t)H_{\text{problem}} =$$

$$\begin{array}{c} \\ \langle 000| \\ \langle 001| \\ \langle 010| \\ \langle 011| \\ \langle 100| \\ \langle 101| \\ \langle 110| \\ \langle 111| \end{array} \begin{array}{c} \begin{array}{cccccccc} |000\rangle & |001\rangle & |010\rangle & |011\rangle & |100\rangle & |101\rangle & |110\rangle & |111\rangle \end{array} \\ \begin{pmatrix} -\frac{1}{2}B(t) & -A(t) & -A(t) & 0 & -A(t) & 0 & 0 & 0 \\ -A(t) & -\frac{3}{2}B(t) & 0 & -A(t) & 0 & -A(t) & 0 & 0 \\ -A(t) & 0 & -\frac{3}{2}B(t) & -A(t) & 0 & 0 & -A(t) & 0 \\ 0 & -A(t) & -A(t) & \frac{3}{2}B(t) & 0 & 0 & 0 & -A(t) \\ -A(t) & 0 & 0 & 0 & -\frac{3}{2}B(t) & -A(t) & -A(t) & 0 \\ 0 & -A(t) & 0 & 0 & -A(t) & -\frac{5}{2}B(t) & 0 & -A(t) \\ 0 & 0 & -A(t) & 0 & -A(t) & 0 & \frac{3}{2}B(t) & -A(t) \\ 0 & 0 & 0 & -A(t) & 0 & -A(t) & -A(t) & \frac{9}{2}B(t) \end{pmatrix} \end{array}. \tag{6.21}$$

Bibliography

[1] S. Santra, G. Quiroz, G. Ver Steeg, and D. A. Lidar, *Max 2-SAT with up to 108 qubits*. New Journal of Physics **16**, 045006 (2014).

[2] V. Choi, *Different adiabatic quantum optimization algorithms for the NP-complete exact cover and 3SAT problems*. arXiv:1010.1221 (2010).

[3] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser *Quantum computation by adiabatic evolution*. arXiv:quant-ph/0001106 (2000).

[4] E. Farhi, J. Goldstone, and S. Gutmann, *A quantum approximate optimization algorithm*. arXiv:1411.4028 (2014).

[5] S. Hadfield, *Quantum algorithms for scientific computing and approximate optimization* (PhD. thesis). arXiv:1805.03265 (2018).

[6] B. Tregenna, W. Flanagan, R. Maile, and V. Kendon, *Controlling discrete quantum walks: coins and initial states*. New Journal of Physics **5**, 83 (2003).

[7] V. Kendon, *Decoherence in quantum walks – a review*. Mathematical Structures in Computer Science **17**, 1169 (2007).

[8] A. Callison, N. Chancellor, F. Mintert, and V. Kendon, *Finding spin-glass ground states using quantum walks*. arXiv:1903.05003 (2019).

[9] R. Martonak, G. E. Santoro, and E. Tosatti *Quantum annealing by the path-integral Monte Carlo method: the two-dimensional random Ising model*. Physical Review B **66**, 094203 (2002).

[10] N. Chancellor *Modernizing quantum annealing using local searches*. New Journal of Physics **19**, 023024 (2017).

Chapter 7

Solitons

7.1 Introduction

Formally, a *soliton* is a solitary wave which asymptotically preserves its shape and velocity upon nonlinear interaction with other solitary waves, or more generally, with another (arbitrary) localized disturbance.

Here, a solitary wave is a traveling wave solution f, which has the form $f(x,t) = g(x - vt) = g(z)$, and whose asymptotic states for $z \to \pm\infty$ are equal.

However, the formal definition is commonly not used in physics, and the subtle distinction between solitons and solitary waves is dropped. Depending on the community, the term soliton refers to localised solutions of a 1D non-linear wave equation where dispersion is exactly cancelled by the nonlinearity or is cancelled by nonlinearity together with linear confinement. Examples include water waves, optical solitons in fibres, and matter-wave solitons in atomic Bose-Einstein condensates.

Here we will focus on matter-wave solitons produced using atomic Bose-Einstein condensates, first because they provide a model system to study the particle and wave like properties of solitons and second because there are exciting experiments happening here in Durham!

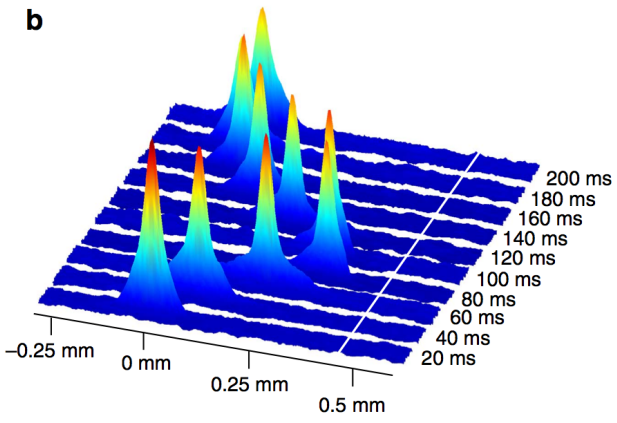


Figure 7.1: Image showing the atomic density for a matter wave soliton reflected from a laser beam 'barrier'. From Marchant, A. L. *et al. Controlled formation and reflection of a bright solitary matter-wave.* Nat. Commun. **4**, 1865 (2013). doi: 10.1038/ncomms2893

7.2 NLSE

To model solitons we will use the non-linear Schrödinger equation (NLSE). This description is appropriate for optical solitons and matter-wave 'solitons'. Due to the analogy with the linear Schrödinger equation we will discuss matter wave solitons in more detail here. For a single atom, the de Broglie wave in the position basis satisfies a 1D wave equation of the form

$$\mathrm{i}\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2} + V\psi \tag{7.1}$$

where V is an external potential. This is the linear Schrödinger equation. For N interacting atoms in a Bose-Einstein condensate (BEC) the effect of interactions is to add a term proportional to the density giving a non-linear Schrödinger equation (NLSE) of the form

$$\mathrm{i}\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2} + V\psi + g|\psi|^2\psi \ , \tag{7.2}$$

where g is a constant relating to the strength of the interactions. The constant g may be either positive or negative, but only a negative sign corresponding to attractive interactions leads to localised solutions. This equation can be written in dimensionless form

$$\mathrm{i}\frac{\partial\tilde{\psi}}{\partial t} = -\frac{\partial^2\tilde{\psi}}{\partial x^2} + \tilde{V}\psi - \tilde{g}|\tilde{\psi}|^2\tilde{\psi} \ . \tag{7.3}$$

This equation also describe solitons in optical fibres, where 'effective' interaction between photons arises due to the **optical Kerr effect** in the optical fibre. Is the non-linearity due to the optical Kerr effect generally positive or negative? Why?

7.3 Numerical methods

We need to solve an equation of the form

$$\mathrm{i}\frac{\partial\tilde{\psi}}{\partial t} = \mathbf{H}\tilde{\psi} \ . \tag{7.4}$$

An approximate solution to this equation has the form

$$\tilde{\psi}(x, t+\delta t) \approx \mathrm{e}^{-\mathrm{i}\mathbf{H}\delta t}\tilde{\psi}(x,t) \ . \tag{7.5}$$

For a NLSE this is only approximate because $\mathbf{H}$ depends on $\tilde{\psi}$.

7.3.1 Split-step Fourier

In the split-step Fourier method we split the 'Hamiltonian' into a kinetic and potential term using the approximation

$$\mathrm{e}^{-\mathrm{i}(\mathbf{T}+\mathbf{V})\delta t} \approx \mathrm{e}^{-i\mathbf{V}\delta t/2}\mathrm{e}^{-i\mathbf{T}\delta t}\mathrm{e}^{-i\mathbf{V}\delta t/2} \ . \tag{7.6}$$

The Fourier 'trick' is to recognise that in Fourier space the operator $\partial/\partial x$ becomes k, so we can write

$$\tilde{\psi}(x, t+\delta t) = \mathrm{e}^{-\mathrm{i}\mathbf{V}\delta t/2}\mathcal{F}^{-1}\left[e^{-\mathrm{i}k^2\delta t}\mathcal{F}\left[\mathrm{e}^{-\mathrm{i}\mathbf{V}\delta t/2}\tilde{\psi}(x,t)\right]\right] \ .$$

This is exactly the same as the angular spectrum method used in the L2 optics course (see Optics f2f).

7.4 Milestone

A good way to start is to first implement the diffraction alone, and propagate an initial Gaussian [$\psi = \exp(-x^2)$] profile and compare to the analytical solution. Then, use a linear harmonic oscillator to exactly cancel the diffraction, and check that the profile $|\psi|^2$ of the wavefunction does not change upon propagation.

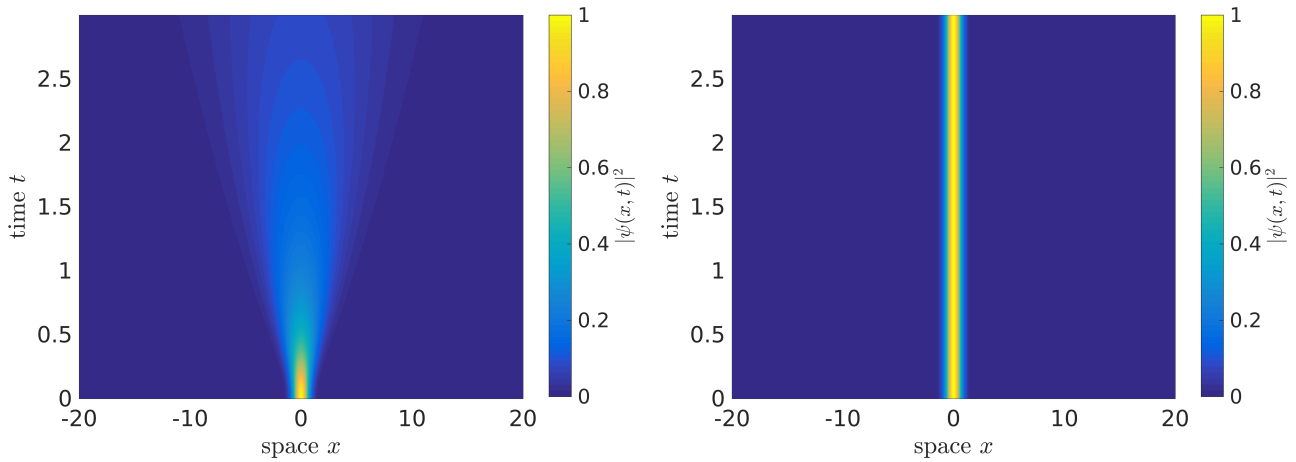


Figure 7.2: Colormap on left showing a gaussian input into the linear Schrödinger equation. Right: the same input but with the parabolic confining potential added.

Now try the NLSE. The soliton solution is in the form of a sech-function, see e.g. `https://en.wikipedia.org/wiki/Soliton_(optics)`. Plot a colormap showing the intensity (modulus squared of the amplitude).

Finally, the your Milestone program should create propagating a sech soliton

$$\psi = \frac{\sqrt{2\xi}}{cosh(x\sqrt{\xi})} e^{ivx} \tag{7.7}$$

(with family parameter $\xi = 1$ and velocity $v = L/4$, where L is the box length, for example). Show that its shape does not change as it propogates and that its norm ($\int |\psi|^2 dx$) is conserved to better than 0.1% accuracy over a time $t = \frac{5}{\xi}$.

7.5 Extending the milestone

Now that you have created a code that accurately propogates solitions, you can extend you investigation in many different directions. For example, Figure 7.3 illustrates what happens when two solitions colide.

The next steps are to test the accuracy of your numerics and then apply the optimised code to address specific physics questions. Some interesting topics include solitons in optical fibres and the dynamics of matter wave solitons in harmonic traps, reflection from a barrier (including 'quantum' reflection), collisions, etc.

Here are some more ideas that you could use as the starting point for your research.

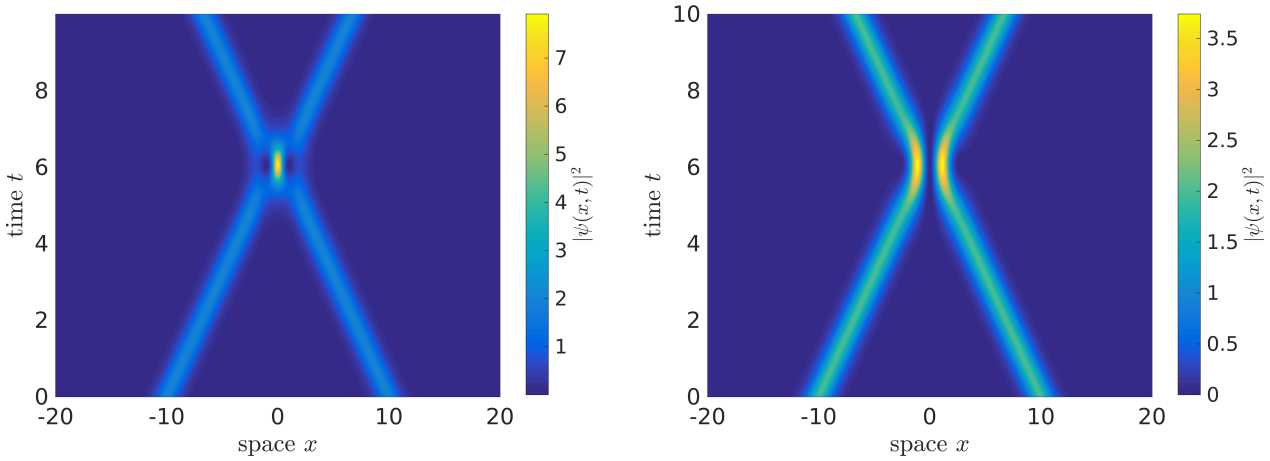


Figure 7.3: These figures illustrate the colision of two solitions. In the left hand panel, the solitions are in phase and combine to form a bright spot as they collide. In the right panel, the solitions are out of phase and appear to bounce off each other.

- Investigate the importance of the amplitude of the initial wave. Show that solitons may be 'dark' as well as bright.
- Transverse modes in an optical fibre.
- Collisions of matter wave solitons, A. D. Martin *et al.*, *Bright solitary-matter-wave collisions in a harmonic trap: Regimes of soliton like behavior* Phys. Rev. A **77**, 013620 (2008).
- Quantum reflection, S. L. Cornish *et al.*, *Quantum reflection of bright matter-wave solitons* Physica D **238**, 1299 (2009).
- Optical solitons and freak waves, B. Kibler *et al.*, *Observation of Kuznetsov-Ma soliton dynamics in optical fibre*, Scientific Reports **2**, 463 (2012).
- Modulational instability and self-organization of light, F. Maucher *et al.*, Phys. Rev. Lett. **116**, 163902 (2016), F. Maucher *et al*, Optical Data Processing and Storage, **3**, 13 (2017).
- Brownian motion and radiative loss of solitons in fibres, V Folli, C Conti, Phys. Rev. Lett. **104** 193901 (2010), F. Maucher *et al* Phys. Rev. A 85, 063803 (2012)

Part III

Astrophysics

Chapter 8

The Black Hole Accretion Disc Spectrum

8.1 Background

Black holes are black, so cannot be seen directly which makes finding and studying them very difficult. They manifest themselves only by their gravitational effect on other bodies. This is at its most spectacular when material falls (accretes) onto the black hole. Some of the immense gravitational potential energy of the infalling material can be converted into radiation, producing intensely luminous, high energy emission before the material disapears forever below the event horizon. Black holes are fundamentally important of their impact on the evolution of galaxies, and because they are an exciting prediction of General Relativity.

The simplest type of accretion flow is a *disc*, where the material is in Keplarian motion, with gravity balanced by the centripetal force. This azimulthal velocity is very much larger than the radial velocity but it is this slow radial infall of material which powers the gravitational energy release.

8.2 Continuum Emission from the Disk

A simple approximation to the spectrum of this flow can be derived if the gravitational potential energy liberated as a mass ΔM slides in time dt inwards from R to $R - dR$ is radiated as black body radiation. For a black hole of mass M this energy is $GPE = GM\Delta M/R^2 \times dR$ (Energy=Force x distance), which gives a luminosity $GM\dot{M}/R^2 \times dR$, where $\dot{M} = \Delta M/\Delta t$ is the rate at which mass accretes through the ring. However, the virial theorem says that only half of this can be radiated (the rest is retained as kinetic energy of the Keplarian orbits) so the luminosity from this ring is $dL = GM\dot{M}/2R^2 \times dR$.

The surface area, dA of this small annulus in the disc over which this energy is dissipated is $dA = 2 \times 2\pi RdR$ where the factor 2 comes as the disc ring has a top and bottom face, both of which radiate. Blackbody radiation has $L = \sigma_{SB}T^4 \times$ area, so the temperature of this ring is

$$T(R)^4 = \frac{GM\dot{M}}{8\pi R^3 \sigma_{SB}} \tag{8.1}$$

A blackbody emitting isotropically has a luminosity per unit frequency per unit area of

$$F_\nu(T) = \pi B_\nu(T) = \frac{2\pi h\nu^3/c^2}{[exp(h\nu/kT) - 1]} \tag{8.2}$$

Thus the total luminosity per unit frequency from the disc is

$$L_\nu = \int_{R_{in}}^{R_{out}} F_\nu(T) dA = \int_{R_{in}}^{R_{out}} F_\nu(T(R)) 4\pi R dR \tag{8.3}$$

The derivation above for the disc temperature did not consider the effects of viscous forces between different annuli in the disc. The effect of the viscous forces is to transport energy and angular momentum outwards in the disc, as well as heating the disc locally. Including these effects gives a revised equation for the radial temperature distribution in the accretion disc of (e.g. Pringle 1981)

$$T(R)^4 = \frac{3GM\dot{M}}{8\pi\sigma_{SB}R^3}[1 - (\frac{R_{in}}{R})^{1/2}] \tag{8.4}$$

In General Relativity, the inner radius of the disc, R_{in} is set by the last stable orbit around the black hole. This is at $6R_g$, where $R_g = GM/c^2$, assuming that the black hole is not spinning, but decreases to $1.23R_g$ for a maximally spinning black hole.

There is a maximum mass accretion rate at which these disc equations break down. When the luminosity of the disc is high enough then the radiation pressure from the outflowing photon flux can be stronger than gravity, halting the inward accretion flow. This is termed the Eddington limit, $L_{Edd} = 4\pi GMm_p c/\sigma_T$ where σ_T is the Thompson cross-section for interaction between an electron and photon, and m_p is the mass of a proton. The mass accretion rate associated with this luminosity depends on the efficiency of the disc, η, in radiating the rest mass energy of the incoming material so $L_{Edd} = \eta\dot{M}_{Edd}c^2$.

8.3 Numerical techniques

It is more meaningful to rewrite the equations in scaled units, where $r = R/R_g$. Then the temperature and luminosity equations become

$$T(r)^4 = \frac{3GM\dot{M}}{8\pi\sigma_{SB}r^3R_g^3}[1 - (\frac{r_{in}}{r})^{1/2}] \tag{8.5}$$

$$L_\nu = \int_{r_{in}}^{r_{out}} F_\nu(T(r))\, 4\pi R_g^2\, r dr \tag{8.6}$$

It is very important to evaluate the radial integral on a *logarithmic* grid, as a large range in radii need to be considered. The grid should run from $log_{10}(r_{in})$ to $log_{10}(r_{out})$ in constant steps $\Delta log_{10}(r)$. The temperature should be calculated at the midpoint of each bin rather than the edge, as this avoids the zero at r_{in}.

The black body emission covers a wide range of frequencies, so a logarithmic scale in frequency is required to calculate the blackbody spectrum from each radius, and the total disc spectrum.

8.4 Your Work Plan

1. **Design Your Program.**

Your first task is to design your program. To begin with, focus on the continuum emission from the disk. Use the pseudocode techniques to design an program that you will be able to use to investigate the spectrum of an accretion disc around a black hole.

As well as an outline of the code structure, you should include: (a) names and description of the functions you will use: what variables will they take as arguements, what will they return? (b) what data will you input into the program, how will the program output the results? (c) what data structures will the program use to store its internal data?

Remember to make your program sufficiently flexible to allow you to tackle new problems as well as the example below.

2. **Solve the Milestone Problem: A simple continuum disc spectrum**

Write a code to calculate the spectrum of an accretion disc extending from $R_{in} = 6R_g$ to $R_{out} = 10^5 R_g$ around a $10M_\odot$ black hole (where the solar mass $M_\odot = 2 \times 10^{30}$ kg) accreting at material at a rate of 10^{15} kg/s from a companion star. Compute this over a frequency range from $10^{14} - 10^{19}$ Hz.

Plot your spectrum and integrate this to derive the total luminosity from the system. You should find that the luminosity is roughly 7×10^{30} W.

At the milestone interview, your program must be able to make a plot of the accretion disk spectrum (in the form $\log(\nu L_\nu)$ vs. $\log \nu$) and compute the total luminosity.

3. **Research with your Program**

Now you have a working code, you can investigate the spectrum of the accretion disc and use your results to write your report. High marks will only be given to reports that demonstrate your initiative. Here are some aspects that you may wish to consider. You should not look at all of these points: these are intended as the starting point for your own investigation. It is better to focus on a narrower topic in more depth.

- It is of great interest to understand how the observed emission from the disk can be used to infer physical parameters (eg., Gierlinski & Done, 2004) Use your program to investigate how the spectrum changes as a function of r_{in} and $\dot{M}$. For example, how does the spectrum of an Eddington limited accretion disc change with M?
- What range of masses might you expect from black holes resulting from collapse of the most massive stars? What range of black hole masses might form in the centers of galaxies? Use your program to compare the observed spectra of both stellar mass and supermassive black holes to predictions based on their mass (eg. Shakura & Sunyaev 1976).
- The temperature of the accretion disc was derived above using Newtonian physics. In full GR this becomes (e.g. Abramowicz & Fragile 2013)

$$T^4(r) = \frac{3GM\dot{M}}{8\pi\sigma_{SB} r^3 R_g^3} \frac{(A-B)}{C} \tag{8.7}$$

where

$$
\begin{aligned}
A &= 1 - \frac{y_{ms}}{y} - \frac{3a_* \ln(y/y_{ms})}{2y} \\
B &= \frac{3(y_1 - a_*)^2 \ln[(y - y_1)/(y_{ms} - y_1)]}{y \quad y_1(y_1 - y_2)(y_1 - y_3)} \\
&\quad + \frac{3(y_2 - a_*)^2 \ln[(y - y_2)/(y_{ms} - y_2)]}{y \quad y_2(y_2 - y_1)(y_2 - y_3)} \\
&\quad + \frac{3(y_3 - a_*)^2 \ln[(y - y_3)/(y_{ms} - y_3)]}{y \quad y_3(y_3 - y_1)(y_3 - y_2)} \\
C &= 1 - \frac{3}{r} + \frac{2a_*}{r^{3/2}} \\
y &= \sqrt{r} \\
y_1 &= 2\cos[(\arccos(a_*) - \pi)/3] \\
y_2 &= 2\cos[(\arccos(a_*) + \pi)/3] \\
y_3 &= -2\cos(\arccos(a_*)/3) \\
r_{ms} &= 3 + z_2 \mp \sqrt{(3 - z_1)(3 + z_1 + 2z_2)} \\
y_{ms} &= \sqrt{r_{ms}} \\
z_1 &= 1 + (1 - a_*^2)^{1/3}[(1 + a_*)^{1/3} + (1 - a_*)^{1/3}] \\
z_2 &= \sqrt{3a_*^2 + z_1^2}
\end{aligned}
$$

and the $\mp$ sign change in r_{ms} is for prograde (disc spinning in the same direction as the black hole) and retrograde rotation, respectively. Your code should recover $r_{ms} = 6$ for $a_* = 0$ and 1.23 for $a* = 0.998$. How does spin change the inner radius and the efficiency of the black hole? How different are these spectra to those produced from the approximate temperature distribution used before? This is a controversial issue (eg., Ebisawa et al. 1991, Shafee et al. 2006).

References

Abramowicz, M. A., Fragile, P. C., 2013, https://link.springer.com/article/10.12942/lrr-2013-1

Ebisawa K.,Mitsuda, K., Hanawa, T., 1991, ApJ, 367, 213

Gierlinski M., Done, C., 2004, MNRAS, 347, 885

Pringle, J. E., 1981, Ann.Rev.Astron.Astroph, 19,137.

Shafee, R., McClintock, J. E., Narayan R., Davis, S.W. Li, L.-X., Remillard, R., 2006, ApJ, 636, L113

Shakura N, & Sunyaev R. A., 1976, MNRAS, 175, 613

Chapter 9

Supernova Cosmology

9.1 Recent Progress

Recently, rapid progress has been made in measuring the geometry of the Universe. This has in turn lead to an accurate determination of its total mass content and demonstrated that the recent history of the Universe has been dominated by a period of accelerating expansion. These measurements have convinced astronomers of the existence of "dark matter" (required to explain the high mass density of the Universe) and "dark energy" (required to explain the recent acceleration). Before starting work on this project you should read more about these terms, and our understanding of the history of the Universe. A good introduction can be found in Liddle: Introduction to Modern Cosmology.

One of the key pieces of observational evidence is the measurement of the brightness of large samples of extragalactic supernovae. Supernovae have a well defined peak brightness, so that the "luminosity distance" to a supernova can be determined by comparing the flux, f, received with the known peak luminosity,

$$d_L = \left(\frac{L_{\rm peak}}{4\pi f} \right)^{1/2} . \tag{9.1}$$

By measuring d_L for a large sample of supernovae covering different distances, the relation between distance and recession velocity (or redshift) can be measured and compared with theoretical models.

For nearby supernovae, we expect a roughly linear relationship between distance and redshift, known as Hubble's Law. However, if we extend the measurement to more distant objects, the relation becomes curved, and strongly dependent on the matter content of the Universe. Given a set of theoretical models for the matter content of the Universe, we can compare each model with the observed supernova data in order to select the best model.

Some of the first results based on large samples of sufficiently distant supernovae were presented by Schmidt et al., 1998 and Perlmutter et al in 1999. They found the surprising result that the Universe contained a lot of matter ($\Omega_m = 0.28$) but that the dominant contribution to the energy density was in the form of a "Cosmological Constant" or vacuum energy. Their results have been subsequently substantiated by larger surveys and by measurements of the Cosmic Microwave Background (CMB).

9.2 Distances in an Expanding Universe

The aim of this project is to be able to use measurements of supernovae in order to constrain the geometry and mass content of the Universe. Our starting point will be the relation between redshift and distance.

9.2.1 The Friedmann Equation

The expansion rate of the Universe is given by the Friedmann equation. This relates the rate of change of the "scale factor" R (effectively the size of the Universe) to the energy density of the Universe:

$$\left(\frac{\dot{R}}{R}\right)^2 = \frac{8\pi G}{3c^2}\left(\rho_{\rm mass}\, c^2 + \rho_{\rm DE}\, c^2\right) - \frac{kc^2}{R^2}, \tag{9.2}$$

where $\rho_{\rm mass}c^2$ and $\rho_{\rm DE}c^2$ are the energy densities in normal matter and "dark energy" respectively (we have omitted the negligible contribution from radiation); k is a constant equal to -1, 0 or 1 for negatively curved, spatially flat and positively curved universes respectively.

The expansion factor of the Universe is defined by

$$a(t) = R(t)/R_0, \tag{9.3}$$

where R_0 is the present day value of the scale factor R, and the redshift, z of a supernova is determined by the expansion factor of the Universe when it occurs

$$a = \frac{1}{1+z}. \tag{9.4}$$

The left hand side of the Friedmann equation can be rewritten in terms of the "Hubble parameter",

$$H = \left(\frac{\dot{R}}{R}\right) \equiv \left(\frac{\dot{a}}{a}\right). \tag{9.5}$$

Note that H is a function of time, and that it changes as the Universe expands. We will make the time dependence explicit below.

An important scale in the Friedmann equation is the "critical density". This is the density the Universe would have if k were zero:

$$\rho_{\rm crit}(t) = \frac{3H^2(t)}{8\pi G}. \tag{9.6}$$

Cosmologists often express the densities appearing in the Friedmann equation in terms of the critical density such that:

$$\Omega_M(t) = \rho_{\rm mass}(t)/\rho_{\rm crit}(t) \tag{9.7}$$

and

$$\Omega_{\rm DE}(t) = \rho_{\rm DE}(t)/\rho_{\rm crit}(t). \tag{9.8}$$

In order to solve the Friedmann equation, we need to know how the matter and dark energy densities depend on time. In general, this is non-trivial, but we can write down simple expressions for their dependence on the expansion factor. Since matter is conserved as the Universe expands,

$$\rho_{\rm mass}(a) = a^{-3}\rho_{\rm mass,0}, \tag{9.9}$$

where the subscript 0 means the value at the present day.

In the simplest models, the dark energy is a fixed property of space (the "cosmological constant") so that it is independent of the expansion factor, i.e.

$$\rho_{\rm DE}(a) \equiv \rho_\Lambda. \tag{9.10}$$

Combining the above equations, we can rewrite the Friedmann equation as

$$H^2 = \frac{H_0^2 \Omega_{M,0}}{a^3} + H_0^2 \Omega_\Lambda - \frac{kc^2}{R^2}. \tag{9.11}$$

Note that

$$\frac{kc^2}{R_0^2} = H_0^2 \left(\Omega_{M,0} + \Omega_\Lambda - 1\right). \tag{9.12}$$

9.2.2 The Distance-Redshift Relation

An important distance for this project is the "comoving distance" between the point in space at which the supernova exploded and the observer. This is equivalent to the present-day separation of these points, but it is not the same as the distance traveled by the photons from the supernova. The comoving distance to a supernova with redshift z, corresponding to a comoving coordinate η, is given by

$$R_0 \eta = \int_{a_1}^{1} \frac{cdt}{a} \equiv c \int_0^z \frac{dz'}{H(z')}. \tag{9.13}$$

The peak flux of a supernova with luminosity L is then given by

$$f = \frac{L_{\rm peak}}{4\pi [R_0 S(\eta)]^2 (1+z)^2}. \tag{9.14}$$

The two factors of $(1+z)$ allow for the reduction of the energy of each photon as it is redshifted by the expansion of the Universe, and the lower frequency with which each photon is emitted. The factor $S(\eta)$ is needed to allow for the effects of General Relativity in distorting space time. If the Universe is "flat", however, (meaning $k = 0$) $S(\eta) = \eta$.

9.3 Comparing Models of the Universe

Using equations 9.11, 9.13 and 9.14, you can now compute the flux expected from a supernova with measured redshift z. Different values of $L_{\rm peak}$, H_0, $\Omega_{M,0}$, Ω_Λ will give different predictions, and our task is to decide which values are most compatible with the measured supernova data.

The best approach to this problem is to compute the χ^2 statistic for each set of values. This is based on comparing the differences between the observed and model fluxes to the observational errors

$$\chi^2 = \sum \frac{(f_{\rm obs} - f_{\rm model})^2}{\sigma_{\rm obs}^2} \tag{9.15}$$

The smallest value of χ^2 will correspond to the best model. For a one parameter problem, two models are equally good descriptions of the data if $|\chi_1^2 - \chi_2^2| < 1$, and one is a much better fit if $|\chi_1^2 - \chi_2^2| > 9$. Note

this use of the χ^2 statistic is distinct from its use to ask whether a model is a good fit to the data. The use of χ^2 to compare to models is closely related to the Bayesian approach to statistics.

In our problem, $L_{\rm peak}$ and H_0 are degenerate (they have the same effect on the predicted flux) and we can only determine their combined value. The best approach is to assume $H_0 = 75\,{\rm km/s/Mpc}$ and to then determine $L_{\rm peak}$ using the local supernova data (eg., $z < 0.1$). For such low redshifts, the comoving distance is accurately approximated by $R_0\eta \approx cz/H_0$. This value of $L_{\rm peak}$ can then be combined with data on distant supernova data to infer the cosmological parameters. Note that if we assume the Universe is flat ($k = 0$), $\Omega_{M,0} = 1 - \Omega_{\Lambda,0}$, so that we only have one parameter to determine from the high redshift data. To determine the best fit value, we must loop over possible values finding the minimum χ^2. To determine the uncertainty, we must find the points on either side that are $\Delta\chi^2 = 1$ greater than the minimum value.

9.4 Your Work Plan

1. **Design Your Program** Your first task is to design a program that you can use to determine the Ω_Λ from the supernova data of Perlmutter et al. *The supernova data are available on the DUO webpage for the computing lab.* You can use the Numpy function "loadtxt" to read the file. The columns of the file give the supernova name, its redshift, the effective peak magnitude and the magnitude error. The "effective" magnitude has already been corrected for the stretching of supernovae light curves, for the band-pass of the original observations and for galactic dust extinction. You can relate the magnitude m to the detected flux, f, using

$$m = m_0 - 2.5\log_{10}(f) \tag{9.16}$$

   where m_0 is a constant (if f is in units of $\rm erg\,cm^{-2}s^{-1}\AA^{-1}$, $m_0 = -20.45$, but note that your results do not depend on this value).

   As well as an outline of the code structure, you should include: (a) names and description of the functions you will use: what variables will they take as arguments, what will they return? (b) what data will you input into the program, how will the program output the results? (c) what data structures will the program use to store its internal data, what system of units will you use? Remember to make your program sufficiently flexible to allow you to tackle new problems as well as the Milestone problem below.

2. **Solve the Milestone Problem.** Your milestone program should solve separately for the value of $L_{\rm peak}$ and $\Omega_{\Lambda,0}$ (you may assume $H_0 = 75\,{\rm km/s/Mpc}$). To determine $L_{\rm peak}$, use the supernova data with $z < 0.1$, and the approximation $R_0\eta = cz/H_0$. Then adopt this value to determine $\Omega_{\Lambda,0}$ using supernova data with $z > 0.1$ under the assumption that $k = 0$.

   You should write your own code to determine cosmological distances, but you may find it helpful to check your results using Ned Wright's web-based calculator. You may wish to use the Scipy integration routines to solve the integral 9.13, but it is best to determine Ω_Λ by evaluating χ^2 for a list of trial values (automated minimisation routines may get trapped in local minima). Always check your results by plotting the observed and predicted supernova magnitudes against redshift.

   At the milestone interview, your program must use the data provide to (i) find the best-fit values of $L_{\rm peak}H_0^2$, then (ii) use this value to determine the best-fit value of Ω_Λ (assuming $k = 0$). Your program should plot the supernova data as a function of redshift and compare this against the best-fit relation you have calculated.

3. **Research with your program.** Now you have a working code, you can investigate supernova cosmology and the geometry of the Universe. High marks will only be given to reports that demonstrate your initiative. Here are some ideas that you may wish to consider. You should not need to address all of these points: these are intended only to provide some possible starting points for your own investigation.

   - Much larger supernova data-sets are now available (eg., http://supernova.lbl.gov). You can improve the accuraccy of the measurement, and investigate the effect of intrinsic variation of SN peak brightness, and/or investigate the systematic uncertainties by using supernova with different intrinsic properties such as colour or host galaxy morphology.
   - In the milestone project, you fitted one parameter at a time. This makes minimisation of χ^2 simple, but this approach is unsatisfactory since the errors in the parameters may be correlated, and it would be better to consider simultaneously minimising χ^2 in a two, three or higher dimensional parameter space (eg. Wall & Jenkins). There are many approaches to multi-dimensional minimisation. Pick one to investigate, but beware of the solution becoming trapped in a local minimum.
   - You do not need to assume that the Universe is flat. However, if $k = \pm 1$, there is a small correction to the distance due to the Universe's geometry, as well as the obvious factor in the Friedmann equation. In general,

     $$S(\eta) = \begin{cases} \sin(\eta), & \text{if k=1,} \\ \eta, & \text{if k=0,} \\ \sinh(\eta), & \text{if k=-1.} \end{cases}$$

     With sufficient supernova, or supernovae at very high redshift, it is possible to determine whether the Universe is flat (an important prediction of "inflation").
   - The χ^2 approach allows you to determine error bars on the model paramaters as well as the best fit values. However, general cosmological models contain many parameters so you must think carefully about how to take into account the correlations between parameters. The Markov-Chain Monte Carlo approach (eg., Ottosen 2012) samples the parameter space with a probability that depends on the likelihood that the model is a good fit to the data. This will lead you into the topic of Bayesian analysis of data.
   - Modern theories suggest that the Dark Energy may not be a simple vaccuum energy. Solutions such as "Qunitessence" (eg., Zlatev et al 1999) allow the apparent value Ω_Λ to be a function of expansion factor. What limits can be placed on the nature of Quintessence? You could also investigate the limits that you can set on alternatives to the theory of General Relativity, for example the class of $F(R)$ models (eg., Li et al., 2012).
   - Other data sets, in particular the power-spectrum of the cosmic microwave background (eg., Ade et al. 2013) play an important role in constraining cosmological paramaters. By adapting the χ^2 analysis to allow for this additional information, we are tackling the problem using Bayesian Statistics. Investigate how the "prior" information from the CMB experiments influences your selection of the best fitting parameters.

References

Liddle. "An Introduction to Cosmology" (Wiley)

Perlmutter et al., 1999, ApJ, 517, 565, http://adsabs.harvard.edu/abs/1999ApJ...517..565P

Schmidt et al., 1998, ApJ, 507, 46, http://adsabs.harvard.edu/abs/1998ApJ...507...46S

Ade et al., 2013, http://arxiv.org/abs/1303.5076

LBNL supernova project web-page. http://supernova.lbl.gov/

Ottosen T. A., 2012, http://arxiv.org/pdf/1206.6905v1.pdf

Ned Wright's cosmology calculator. http://www.astro.ucla.edu/ wright/CosmoCalc.html

Wall & Jenkins. "Practical Statistics for Astronomers" (Cambridge University Press)

Zlatev I., Wang L., Steinhardt P. J., 1999, Phys. Rev. Lett., 82, 896

Li, B., 2012, Ast. & Grav., 53, 37. http://adsabs.harvard.edu/abs/2012A%26G....53d..37L

Chapter 10

Gravitational Collapse: simulating interactions between many bodies

10.1 Background physics

Gravity is an interesting force. Because the force is always attractive, systems of particles that are brought together by gravity have unusual properties. In particular, the system may be dominated by collective effects where the motion of a particle depends on the locations of all other particles in the system. Such systems have a negative specific heat - when energy is lost from the system, its temperature *increases*. The time evolution and properties of such systems are too complex to predict analytically.

Computer models of physical systems, in which the interactions of many particles are followed in time have made significant contributions to research in many areas such as plasma physics, semiconductors, fluid dynamics and astrophysics. This is due to a combination of advances in both the speed of computers and in numerical techniques used to rapidly calculate the inter-particle forces. In this project we consider the gravitational interaction between N massive particles.

For a system containing N particles of mass m_i, Newton's law of gravity gives the gravitational force on the i^{th} particle due to the other $N-1$ particles as

$$\mathbf{F}_i = G \sum_{j \neq i} -m_i \; m_j \; \frac{\mathbf{r_i} - \mathbf{r_j}}{|\mathbf{r_i} - \mathbf{r_j}|^3} = \sum_{j \neq i} \mathbf{F}_{ij}, \tag{10.1}$$

where $\mathbf{r_i}$ is the position vector of the i^{th} and G is Newton's gravitational constant. Once we know the force on a particle, its acceleration is given by Newton's second law:

$$\mathbf{F_i} = m_i \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t}. \tag{10.2}$$

The goal of this project is to write a computer program which can follow the motion of N particles over time, taking into account the gravitational forces between them.

10.2 Numerical solution: the leapfrog approximation

Since the force on a particle in a gravitational system depends only upon the positions of the other particles and not on the velocity (as would be the case in for a particle moving in a magnetic field, for example), we can use a particularly simple scheme to solve or numerically integrate equations (1) and (2), and thus advance the motion of the particles with time.

The method is called the leapfrog finite difference approximation. It is the basis for the integrator used in most astrophysical simulations of gravitational systems. In particular, the leap-frog scheme is used in cosmological simulations of the formation of dark matter structures in the Universe, which can use billions of particles. There are higher order, more accurate integration schemes, such as the Runge-Kutta method for solving differential equations, and these are used in models of the evolution of the solar system. However, these require the storage of more variables, and hence take up more computer memory, which is a drawback when designing simulations that will run with large numbers of particles.

In the leapfrog scheme, we increment the time variable t in steps of dt; we refer to the n^{th} timestep as $t_0 + n\,dt$, where t_0 is the initial time. We calculate the positions and velocities half a timestep out of phase, hence the name *leapfrog*. (Note this scheme is similar to the mid-point version of Euler's method, in which variables are evolved using the value of a gradient estimated half way across the time step.) We use the positions of the particles at timestep n to compute the forces F_{ij}. These forces are then used to update the particle velocities from their values at the timestep labelled $n-1/2$ to the new values at timestep $n+1/2$:

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^{n-1/2} + \mathbf{F}_i\,dt/m_i. \tag{10.3}$$

We then use these new particle velocities to update the particle positions to timestep n

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^{n} + \mathbf{v}_i^{n+1/2}\,dt. \tag{10.4}$$

For this scheme to work, it is important to note that we need to specify the initial velocities at time $t = -\frac{1}{2}dt$ even though we specify the initial positions at time $t = 0$. While this makes little difference if we start from a random set of particle positions and velocities, it is important in the circular orbit test problem that we consider below.

By applying this scheme, we do not know the particle positions and velocities explicitly at the same time. However, if we wish to compute the energy of a system of particles, we need to know their potential energy and kinetic energy at a given epoch, and so in this case we would need to estimate the velocities and positions at the *same* timestep.

10.3 Some numerical points and tips

Equation (1) needs to be modified when particles come close together. If we did not modify the equation, very short timesteps would have to be used to follow close encounters between particles due to the huge accelerations which the particles would experience. The gravitational force is modified or, technically speaking, *softened* by including a constant ϵ^2 in the denominator of equation (1)

$$\mathbf{F}_i = G\sum_{j\neq i} -m_i\ m_j\ \frac{\mathbf{r_i} - \mathbf{r_j}}{\left[|\mathbf{r_i} - \mathbf{r_j}|^2 + \epsilon^2\right]^{3/2}}. \tag{10.5}$$

This effectively makes the particles behave like hard spheres with radii $\epsilon/2$. Note that the expression for the potential energy of two particles needs also to be adapted if we wish to check for energy conservation.

The gravitational force given by Equation (1) is antisymmetric. The force on particle i due to particle j is the negative of the force on particle j due to particle i: $F_{ij} = -F_{ji}$. Hence, when you have computed F_{ij}, you have F_{ji} for free! This trick can cut the number of operations by a factor of two. Be careful to write the force calculation so that you do not re-calculate the particle separation unnecessarily.

A system of particles interacting through a conservative force should conserve energy over time. In a numerical calculation, energy will not be conserved exactly. In practice, the energy will be conserved to within some tolerance e.g. a variation of $x\%$ of the initial energy, where x depends on the timestep chosen. The size of x should reduce as more steps are used to evolve system over a fixed time interval; if this doesn't happen, then there is probably a bug in your code!

To compute the energy of a system of particles, as commented upon in the previous section, the positions and velocities of the particles need to be calculated at the same time. This can be done by modifying Equation 4 to define a new position vector which has half of the increment.

Finally, when calculating forces, note that a particle cannot exert a gravitational force on itself. When calculating the gravitational potential energy, only count the contribution from each *pair* of particles once.

10.4 Your Work Plan

1. **Design your program** Before writing any Python code, design a code to follow the gravitational interactions and motions of N particles. Write a pseudocode, a list of the instructions or steps to be followed to solve this problem.

   Things to bear in mind: (1) The initial conditions (starting positions and velocities, particles masses) will be different for each problem: the pseudo code should indicate the different ways in which these may be set, but does not need to go into detail. (2) The guts of the program, i.e. the calculation of the force between particles, the update of the positions and velocities, and the calculation of the system energy, should be general for the case of N particles. (3) The output will be different for each problem. Think about which quantities you need to store as arrays. This is determined in part by the algorithms used but also by the output that you need to plot for each task e.g. for the N particle case, we tend to plot the particle positions at a given time, rather than plotting their orbits, which we would do in the case of $N = 2$.

   In order to simplify running your program, it is helpful for you to arrange to read the input paramaters (such as the intitial particle positions and velocities) from a file, rather than having to type them in each time.

2. **Solve the Milestone Problem** Write a Python code to model the motion of a light particle around a heavier particle. Following your pseudocode design, as much as your code as possible should be written for N particles - only the initial conditions and the output should be specific for 2 or 3 particles. In this way, we can test the main part of the code and use it without modification in the next problem.

   Solve the motion of the Earth about the Sun. Assume that the Earth follows a circular orbit. (Mass of the Earth $= 5.9742 \times 10^{24}$kg, mass of the Sun $= 1.9889 \times 10^{30}$kg, Earth-Sun distance: 1.4960×10^{11}m). You can determine the velocity the Earth needs to follow a circular orbit analytically. What is the period of the orbit? For this problem, if your code works and your analytic calculation of the velocity is correct, the two particles should never come together, so the softening can be set to zero. Show that your simulation produces a circular orbit with the expected period.

It is important to note, that the initial velocities need to be specified half a timestep before the initial position. Thus if the initial position of the Earth (at $t = 0$) is on the x-axis, there will still be significant initial velocity components (at $t = -\frac{1}{2}dt$) in both y and x directions.

You should follow the system for 100 orbits to show that the numerical calculation is stable and that the errors do not grow over time. A useful diagnostic is the energy of the system, so you should also plot the energy of the system as a function of time (KE, PE, total energy, and on a separate plot, the percentage change in total energy, compared to the initial energy, as a function of time). You should also plot the radius of the particle as a function of time. Examine your results for different sizes of time step. It should easily be possible to conserve energy to better than a percent.

Once you have a program that works well, you can introduce a third particle representing Jupiter. Jupiter has a mass of 1.8986×10^{27}kg and orbits at a radius of approximately 7.7854×10^{11}m. Initialise your calculation assuming that the Earth and Jupiter are co-linear but located on opposite sides of the sun. The presence of Jupiter will perturb the Earth's orbit slightly and the system's potential and kinetic energy will fluctuate; however, the total energy should be constant to better than a percent.

In your milestone interview, your program should evolve the Sun, Earth, Jupiter system for 100 orbits using a time step of 10 days. Use your program to plot the Sun-Earth separation as a function of time, and to plot the KE, PE and total energy as a function of time. Compute the change in the total energy after 100 orbits as a percentage of the initial energy of the system, and also the percentage change in the Sun-Earth separation.

3. **Extend your program** Now that you have a working code, you can apply it to a system that cannot be followed analytically, the gravitational collapse of N particles. The only parts of your code which will need development are the initial conditions and the output.

   The starting point is to put N particles within a sphere of radius R with random x, y and z co-ordinates. Give the particles zero velocity to start with and equal mass. You need to devise an algorithm to assign an x, y and z position at random, where each co-ordinate lies within the range $-R$ to R. Then check to see if a particle is within a sphere of radius R from the origin; retain co-ordinates which satisfy this condition. (**Tip:** set the softening to $0.1R$. Start off with $N = 100$.)

   Plot the $x - y$ positions of the particles after selected numbers of steps (e.g. if you use 1000 steps, make a plot every 200 steps). Plot the KE, PE, total energy of the particles as a function of time. Experiment with the size of the timestep. You should aim to conserve energy to better than 10%.

4. **Research with your program.** Now you can apply your code to model the collapse of a gravitating system. The points below are suggestions for some of the things you could look at. You should not need to consider all of them. It is better to focus on one topic in more depth.

   - A good example of a system of N particles is a globular cluster, a system of stars born in a compact configuration (see Vesperini 2010 for a recent review). Assuming the stars are born with zero velocity, how long does it take the cluster to collapse to an equilibrium state in which the rms (root-mean-square) radius is no longer contracting? What happens if you continue to evolve the system for longer? What happens if the stars have different masses?
   - Follow the collapse of a system with a large initial radius. Explore whether the final system satisfies the virial theorem ($2T + W = 0$, where T is the total kinetic energy of the particles and W is their potential energy) and investigate the thermodynamics of the system when particles lose energy or interact with each other (eg., Hut 1997)

- A topical question is whether globular clusters contain black holes. Explore what happens if you add a few very massive particles to your globular clusters with mass 100 times greater than that of a typical star. Explore what happens if you allow these particles to merge with each other if they come sufficiently close (eg., Gultekin et al., 2004).
- By colliding two systems of particles you can simulate the effects of collisions between globular clusters or galaxies (eg., Barnes et al. 1991). What happens? Does the final system differ from that formed in the collapse of a uniform sphere? An elegant analytic theory is presented in Lynden-Bell 1967. The colliding systems (star clusters or galaxies) are normally expected to be internally in dynamical equilibrium before the collision, so you should check this by first evolving your initial systems in isolation. (Note that real galaxies have quite complicated structures, with disk, bulge and dark matter halo. You may therefore want to first investigate collisions of idealized spherical galaxies before attempting more realistic cases.)
- By starting your simulation from almost uniform, periodic initial conditions, you can simulate the formation of structure in the Universe (eg., Aarseth et al 1979, Frenk & White 2012). You will need to modify your equations to take the periodicity and expansion of the universe into account. How does the collapse of structure in the Universe differ from the collapse of a spherical distribution?

References

Landau et al. Computational physics.

Binney J., Tremaine S., 1987, Galactic Dynamics

Barger V.D. and Olsson, M.G.: Classical Mechanics, A Modern Perspective (McGraw-Hill)

A nice explanation of the Lagrangian points can be found at:
http://www.esa.int/esaSC/SEMM17XJD1E_index_0.html

Vesperini 2010, Phil. Trans. Royal Soc. A, 368, 829.
http://rsta.royalsocietypublishing.org/content/368/1913/829.full.pdf

Hut 1997, Complexity, 3, 38,
http://arxiv.org/pdf/astro-ph/9704286v1.pdf

Gultekin K., Miller M. C., Hamilton D. P., 2004, 616, 221,
http://adsabs.harvard.edu/abs/2004ApJ...616..221G

Barnes J., Hernquist L., Schweizer F., 1991, Sci. Am., 265, 40,
http://adsabs.harvard.edu/abs/1991SciAm.265...40B

Lynden-Bell D., 1967, MNRAS, 136, 101
http://adsabs.harvard.edu/abs/1967MNRAS.136..101L

Aarseth S. J., Turner E. L., Gott, J. R., III, 1979, ApJ, 228, 664,
http://adsabs.harvard.edu/abs/1979ApJ...228..664A

Frenk C. S., White S. D. M., 2012, Ann. Phys., 524, 507,
http://adsabs.harvard.edu/abs/2012AnP...524..507F

Chapter 11

The Restricted Three-Body Problem (Rocket to the Moon)

11.1 Background

The goal of this project is to integrate very accurately the equations of motion that govern gravitating particles. You will tackle problems in which accuracy is of key importance. This will allow you to predict confidently the position of particles over many orbital periods.

Why would you want to be able to do this? While the motion of two point masses around each other can be solved exactly, gravitating systems of three or more particles have no known analytical solution and are inherently chaotic: small variations in the initial position or velocity result rapidly in increasingly large deviations as a function of time. We will concentrate on systems of three particles. In the simplest case, two very massive particles orbit each other, defining a time-evolving gravitational potential, and we are interested in the motion of the third, lighter body. This is called the 'restricted three-body problem' - restricted because we assume that the third body does not affect the orbit of the other two. This has many interesting applications, from the stability of satellite orbits, the formation of planetary rings, the stability of the solar system, to the destruction of stellar disks in galaxy mergers.

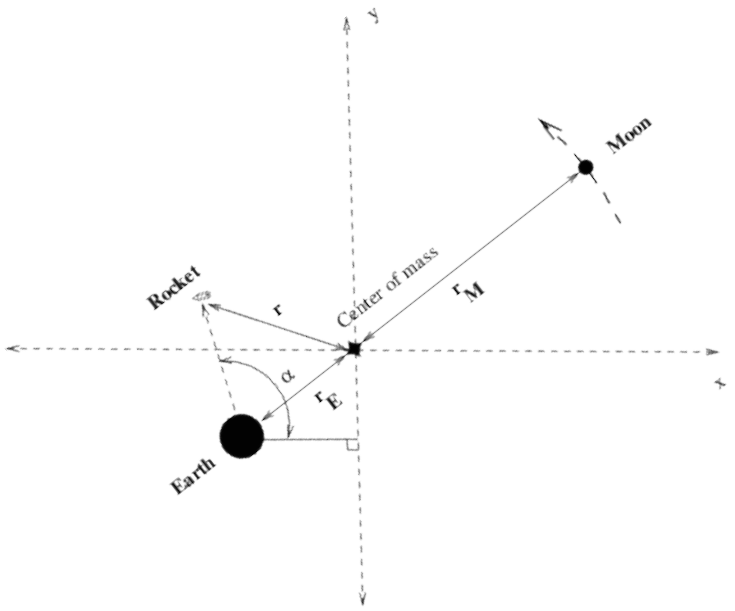
We will start with a simpler example, that of sending a rocket to the Moon. We will solve for the ballistic motion of the rocket, as it is attracted gravitationally by both Earth and Moon, treating all as point particles. The Earth and Moon follow elliptical orbits about a common centre of mass. Approximating this orbit as circular, the distance of each body from the centre of mass can be written in terms of d, the Earth-Moon distance, as

$$r_E = d\frac{m_M}{(m_M + m_E)}$$

and

$$r_M = d\frac{m_E}{(m_M + m_E)}.$$

Here, r_E and r_M are the distances of the Earth and Moon to the centre of mass, and m_E and m_M are the masses of the Earth and Moon, respectively. The period of the orbit, T, can be worked out using Kepler's third law (using the sum of the Earth and Moon masses as the gravitating mass).



Let us now consider the forces exerted on a spacecraft (whose mass, m, is negligible compared with the Earth or the Moon!), again with the centre of mass as the origin of the co-ordinate system. For simplicity we assume that the rocket moves in the orbital plane of the Earth-Moon system. All we need is Newton's law of gravity:

$$\mathbf{F} = -Gmm_E\frac{(\mathbf{r}-\mathbf{r_E})}{|\mathbf{r}-\mathbf{r_E}|^{\mathbf{3}}} - Gmm_M\frac{(\mathbf{r}-\mathbf{r_M})}{|\mathbf{r}-\mathbf{r_M}|^{\mathbf{3}}},,$$

and hence

$$\frac{d^2\mathbf{r}}{dt^2} = -Gm_E\frac{(\mathbf{r}-\mathbf{r_E})}{|\mathbf{r}-\mathbf{r_E}|^{\mathbf{3}}} - Gm_M\frac{(\mathbf{r}-\mathbf{r_M})}{|\mathbf{r}-\mathbf{r_M}|^{\mathbf{3}}}\,, \tag{11.1}$$

where variables in bold type now represent vector quantities. In terms of Cartesian coordinates x and y in the orbital plane,

$$\frac{d^2x}{dt^2} = -Gm_E\frac{(x-x_E)}{d_E^3} - Gm_M\frac{(x-x_M)}{d_M^3}\,,$$

where

$$d_E = \sqrt{(x-x_E)^2+(y-y_E)^2}$$

is the distance from the spacecraft to the Earth, and

$$d_M = \sqrt{(x-x_M)^2+(y-y_M)^2}$$

is that from the spacecraft to Moon.

Your first task is to write a programme to solve for the motion of the spacecraft as a function of time in the gravitational field of the Earth and Moon.

11.2 Numerical solution: Solving differential equations

There are many different ways solving the differential equations of motion numerically. The simplest is via a Taylor expansion. Given the position and velocity of the rocket at time t_n, the Taylor expansion in the small time-step $a = t_{n+1} - t_n$ is

$$\mathbf{x}_{n+1} = \mathbf{x}_n + a\dot{\mathbf{x}}_n + \frac{a^2}{2}\ddot{\mathbf{x}}_n + \mathcal{O}(a^3) \tag{11.2}$$

$$\dot{\mathbf{x}}_{n+1} = \dot{\mathbf{x}}_n + a\ddot{\mathbf{x}}_n + \mathcal{O}(a^2) \tag{11.3}$$

where $\mathcal{O}(a^n)$ indicates the error term resulting from truncating the expansion depend on the step size a to the n^{th} power, and $\dot{\mathbf{x}} \equiv \mathrm{d}\mathbf{x}/\mathrm{d}t$, $\ddot{\mathbf{x}} \equiv \mathrm{d}^2\mathbf{x}/\mathrm{d}t^2$. $\ddot{\mathbf{x}}$ is calculated using equation (11.1).

A more complicated but more accurate method is the fourth-order Runge-Kutta formula,

$$\mathbf{z}_1 = \mathbf{x}_n + \frac{a}{2}\dot{\mathbf{x}}_n \qquad \dot{\mathbf{z}}_1 = \dot{\mathbf{x}}_n + \frac{a}{2}\ddot{\mathbf{x}}_n \tag{11.4}$$

$$\mathbf{z}_2 = \mathbf{x}_n + \frac{a}{2}\dot{\mathbf{z}}_1 \qquad \dot{\mathbf{z}}_2 = \dot{\mathbf{x}}_n + \frac{a}{2}\ddot{\mathbf{z}}_1 \tag{11.5}$$

$$\mathbf{z}_3 = \mathbf{x}_n + a\dot{\mathbf{z}}_2 \qquad \dot{\mathbf{z}}_3 = \dot{\mathbf{x}}_n + a\ddot{\mathbf{z}}_2 \tag{11.6}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{a}{6}(\dot{\mathbf{x}}_n + 2\dot{\mathbf{z}}_1 + 2\dot{\mathbf{z}}_2 + \dot{\mathbf{z}}_3) + \mathcal{O}(a^5) \qquad \dot{\mathbf{x}}_{n+1} = \dot{\mathbf{x}}_n + \frac{a}{6}(\ddot{\mathbf{x}}_n + 2\ddot{\mathbf{z}}_1 + 2\ddot{\mathbf{z}}_2 + \ddot{\mathbf{z}}_3) + \mathcal{O}(a^5) \tag{11.7}$$

In this method the velocity and acceleration are evaluated at 3 test points $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$ across the time-step, in addition to at the starting point: $\dot{\mathbf{z}}_n$, $\ddot{\mathbf{z}}_n$ are evaluated at time t_n, $\dot{\mathbf{z}}_1$, $\ddot{\mathbf{z}}_1$, $\dot{\mathbf{z}}_2$, $\ddot{\mathbf{z}}_2$ at time $t_n + a/2$, and $\dot{\mathbf{z}}_3$, $\ddot{\mathbf{z}}_3$ at time $t_n + a$. By taking a weighted average over these 4 points we greatly improve the accuracy of the position and velocity estimated at the next time-step. The error terms in this method are of order a^5. If carefully applied, this method will give us the accuracy to evolve the position of the 'rocket' over many orbital time periods, allowing you to tackle much more interesting problems.

For both the Taylor expansion and Runge-Kutta methods, the time-step size a should be 'small' enough to track the orbit accurately. You should think carefully how to best limit a. Given initial values for $\mathbf{x} = (x, y)$ and $\dot{\mathbf{x}} = (dx/dt, dy/dt)$, with either the Taylor expansion or Runge-Kutta methods, we can march the orbit forward in time and calculate the position and velocity up to any given time. For each location along the orbit, we can calculate the acceleration from the position by using the equations of motion. Note that for the Runge-Kutta method we can only calculate $\mathbf{z}_2$ and $\dot{\mathbf{z}}_2$ after we know *both* $\mathbf{z}_1$ and $\dot{\mathbf{z}}_1$; this means we must apply the equations 11.4, 11.5, 11.6, and 11.7 in that order.

There is the added complication that the positions of the Earth and Moon are continually changing along their circular orbit, so their positions also need updating at each time-step. Since we assume that the Earth and Moon move on circular orbits about the centre of mass, it is straightforward to write down

expressions for the x and y co-ordinates of these bodies as a function of time, using simple harmonic motion.

11.3 Implementation in Python

Although you should start to write your program using a simple implementation of these schemes, you are welcome to use the scipy library of routines that solve differential equations. For example, the routine scipy.integrate.ode will solve the differential equations for you. More help is given at the back of the book, or you can look at:

http://docs.scipy.org/doc/scipy/reference/tutorial/integrate.html

However, these routines are a 'black box' and you should check that they work by comparing with your own simple code. You will need to understand and establish what limits the accuracy of these integrators.

This challenge is all about accuracy. If you are out by a few kilometre, the rocket will crash into the Moon's surface or escape into space. If you are integrating the 3-body system for a long period of time, small integration errors will rapidly build up and your results will be nonsense. As you write your program, think carefully about how to assess the accuracy of your results.

11.4 Your Work Plan

1. **Design your program.** First, design a code to follow the motion of a rocket in the gravitational field of the Earth and Moon. Write down a pseudo-code which sets out the steps you will need to go through to solve the equations of motion.

   Your outline of the code structure should include: (a) Names and descriptions of the functions needed along with the variables used by the function and the variables returned by the function. (b) What data will you need to input to the code and how will you show the results? (c) What data structures will you use to store information in the code?

   Some things to think about: We have assumed that the Earth and Moon follow circular orbits about their centre of mass. Therefore, at a given time, we can write down the x and y co-ordinates of the Earth and Moon using the solutions of simple harmonic motion and given the period of the orbit calculated using Kepler's third law. Write your program in such a way that it would be easy to adapt to another method for calculating the orbit.

   Since you will use many time steps, you may need to think carefully about data storage. Whilst all of the time steps are needed to calculate the motion of the rocket, not all of the steps need to be output to a file or shown on a plot. Also, note that to apply the Runge-Kutta algorithm, only a small number of variables need to be stored. We do not need to retain these variables for steps before the one we are advancing from. Hence, we can be selective about how many values of x and y we store in a vector (or 1-D array) to use in a plot of the orbit. Be aware that computational resources are limited. You should focus on extracting the maximum accuracy from the available resource.

2. **Solve the milestone problem.** Write a Python code to solve the equations of motion and follow the orbit of a rocket about the Earth-Moon system. You will need to solve for the position of the Earth and Moon along their orbits, and then us these to calculate the acceleration on the rocket.

   We will test the program by investigating the stability of the lunar L2 'Lagrangian' point. The L2 point lies on line connecting Earth and Moon, past the Moon as seen from Earth. The additional

gravitational force exerted by the Moon compensates for the greater distance from the Earth and a satellite (or rocket) at this point will orbit Earth with the same period as the Moon, locked into the same relative position. An example of a satellite in the lunar L2 point is the Artemis spacecraft (see http://www.nasa.gov/mission_pages/themis/news/artemis-orbit.html). Satellites are often place at the Sun-Earth's L2 point, an example is the *Planck* spacecraft. The lunar L2 point has also been proposed for radio SETI observations. Throughout this test, we will ignore (tidal) effects arising from the gradient in the Sun's gravitational field. This is a good approximation since the Earth-Sun distance is so much larger than the radius of the Moon's orbit, but see below for a possible improvements to the scheme. The distance of the lunar L2 Lagrangian point measured *from the Moon* is approximately

$$\Delta r = d \left(\frac{M_m}{3M_E} \right)^{1/3} . \tag{11.8}$$

To confirm that your program works correctly, you can solve for the motion of the rocket when it is initially placed at L2, and moving with speed such that its orbital period is that of the Moon. Solve for the position of the rocket over 1 lunar period. Plot the orbit of the Moon and the rocket: you will find that they initially orbit together but then separate. The expression for Δr is only approximate, and you can manually vary the initial position by a few percent to obtain better results. Now plot the position of the satellite and show that for a suitable choice of Δr it remains less than 100,000 km from the Moon over one period.

Hint: If you use the Taylor expansion algorithm, you should get reasonable results with a time-step of $a = 10$ seconds. The use of the more accurate Runge-Kutta method (or the method within SciPy) would allow you to use a larger time-step and trace out the motion of the rocket more quickly.

In your milestone interview, your program should solve for the orbit of a rocket at the L2 point to high accuracy over one lunar orbit. Use *both* the Taylor expansion *and* the Runge-Kutta (or SciPy) routine. Determine the optimal value of Δr (the distance of the L2 point from the Moon) to better than 1% accuracy, and plot the orbit of the Moon and the rocket. Determine the separation of the rocket relative to the Moon after 1 orbit.

3. **Research with your program.**

Now you have a working code, you can investigate various aspects of the 3-body problem. High marks will only be given to reports that demonstrate initiative. Here are some ideas that you may wish to consider. You probably do not have space in your report to address more than one of these points: it is better to address one particular aspect in detail. Remember that you will need to be able to demonstrate the accuracy of your results.

- You now have the tools to send a rocket to the Moon. Start with the rocket in a circular orbit around Earth, with radius of, say, 7000 km (note that this is *not* the distance from the rocket to the centre of mass!). Assume that the rocket is given an instantaneous velocity boost which sets it on a new orbit that takes it to the Moon. Use your code to estimate the required boost. Minimising the amount of fuel required is critical to the success of the mission. You should not be happy simply to reach the same orbital radius as the Moon or indeed crash into it. To extend your work, consider using the Lagrangian points, or travelling to Mars using a sling-shot around the Moon. Do not mix up units of miles and kilometres.

- A class of asteroids called Trojans is associated with the Jupiter-Sun Lagrangian points, see for example http://astronomy.swin.edu.au/cosmos/T/Trojan+Asteroids. Investigate the stability of the orbits of these bodies.
- Giant planets, such as Saturn and Jupiter, are surrounded by thin disks of rocks and dust debris. These rings have very detailed structure that results from resonances in the gravitational potential created by the planet and its moons (Colombo et al. 1968). You could use your program to understand how such structures develop and investigate their long-term stability. How do your results compare to the observed rings?
- Gravitating systems of several massive bodies are inevitably chaotic. Yet the solar system has been stable for long enough for life to evolve on Earth. Use your program to investigate the long term stability of the solar system. This is a challenging problem because you will need to simultaneously evolve several planets over long timescales with extreme accuracy. Laskar, 2012, provides a recent review.
- Many planets have been discovered around nearby stars, including some planets in binary star systems. You could investigate the stability of planetary orbits in such binary star systems, and the implications for whether life is possible on such planets.
- In a seminal paper, Toomre & Toomre (1972) investigated galaxy collisions using the restricted 3 body method. A good starting point is to approximate the galaxies as point masses, and then calculate orbits of stars using your program. A better way is to represent the potential of each galaxy by an extended dark matter halo (see Narasimhan et.al 1997 for an example of how to do this). By ignoring the gravitational interactions between individual stars, you can compute the orbits of stars independently, and so avoid needing to solve the full N-body problem. Initialise locations and velocities of the stars by placing them in a stable disk (check stability by evolving the galaxy in isolation), then compute their location as a function of time. 'When mice collide' is a spectacular 'Astronomy Picture of the day' image, located at http://apod.nasa.gov/apod/ap020506.html.

Constants

In order to avoid difficulties in comparing results for the Milestone problem, please adopt the following values of the physical and astronomical constants in your calculation.

The mass of the Earth is $m_E = 5.9742 \times 10^{24}\ kg$

The mass of the Moon is $m_M = 7.35 \times 10^{22}\ kg$

Newton's constant of gravitation is $G = 6.6726 \times 10^{-11}\ Nm^2kg^{-2}$

The Earth-Moon distance is $d = 3.844 \times 10^5\ km$

References

Toomre, A., & Toomre, J. 1972, ApJ, 178, 623

Colombo, G., Franklin, F. A., & Munford, C. M. 1968, AJ, 73, 111

Laskar, 2012, http://arxiv.org/abs/1209.5996

DeVries P.L.: A First Course in Computational Physics (Wiley)

Barger V.D. and Olsson, M.G.: Classical Mechanics, A Modern Perspective (McGraw-Hill)

Narasimhan, K. S. et.al. 1997, J. Astrophys. Astr., 18, 23

A nice explanation of the Lagrangian points can be found at:
http://www.esa.int/esaSC/SEMM17XJD1E_index_0.html

Chapter 12

White Dwarfs and Neutron Stars

12.1 Background

White dwarfs and neutron stars are two of the long-lived end states of stellar evolution (the third being a black hole). Both are characterized by matter at extreme densities ranging from 10^5 to 10^{14} times that of the Earth. Since they are both quasi-equilibrium states, however, their interior structure can be derived from a set of differential equations which describe the balance between internal pressure forces and the attractive force of gravity which acts to compress the stellar remnant. Typical white dwarfs have masses between 0.5 and $1.4\,M_\odot$ and radii of $\sim$ 5000 km (similar to the Earth). Their density is of the order of 10^9 kg m^{-3} at which degeneracy effects due to the Pauli exclusion principle start to come into play. Neutron stars are even more compact objects. Their density is the order of 3×10^{17} kg m^{-3}, comparable to that of the matter inside an atomic nucleus, and high enough that non-Newtonian gravity effects also become important. A typical neutron star has a mass of $1.5\,M_\odot$ and a radius of 10 km. Neutron stars are often observed as pulsars, emitting regularly spaced pulses of electromagnetic radiation because they rotate rapidly and radiate only in particular directions. Above masses of $\sim 3\,M_\odot$, neutron stars are believed to collapse into black holes.

12.2 Newtonian Formulation

The interior structure of a normal star under hydrostatic equilibrium can be determined by solving the following equations of stellar structure which can be derived using classical Newtonian mechanics (Carrol & Ostlie, 2007):

$$\frac{dp}{dr} = -\frac{G\rho(r)\mathcal{M}(r)}{r^2} = -\frac{G\epsilon(r)\mathcal{M}(r)}{c^2 r^2} \tag{12.1}$$

$$\frac{d\mathcal{M}}{dr} = 4\pi r^2 \rho(r) = \frac{4\pi r^2 \epsilon(r)}{c^2} \tag{12.2}$$

$$\mathcal{M}(r) = 4\pi \int_0^r r'^2 dr' \rho(r') = 4\pi \int_0^r r'^2 dr' \epsilon(r')/c^2 \tag{12.3}$$

Here G is Newton's gravitational constant, $p(r)$ is the pressure at distance r from the centre of the star, $\rho(r)$ is the mass density at r and $\epsilon(r)$ is the corresponding energy density. The quantity $\mathcal{M}(r)$ is the

total mass inside the sphere of radius r. In these equations we have departed slightly from Newtonian physics, defining the energy density $\epsilon(r)$ in terms of the mass density $\rho(r)$ according to the famous Einstein equation from special relativity,

$$\epsilon(r) = \rho(r)c^2 \tag{12.4}$$

This allows the same equations to continue to be used when one takes into account contributions of the interaction energy between the particles making up compact stellar remnants.

To solve this set of equations for $p(r)$ and $\mathcal{M}(r)$ one can integrate outwards from the origin ($r = 0$) to the point $r = R$ where the pressure goes to zero. This point defines R as the radius of the star. One needs an initial value of the pressure at $r = 0$, let us call it p_0, and then R and the total mass of the star, $\mathcal{M}(R) \equiv M$, will depend on the value of p_0. To be able to perform the integration, one also needs to know the energy density $\epsilon(r)$ in terms of the pressure $p(r)$. This relationship is the equation of state (EoS) for the matter making up the star. Much of the effort required in applying these equations to the conditions appropriate to white dwarfs and neutron stars is in developing an appropriate EoS for the problem at hand.

12.3 General Relativistic Corrections

The Newtonian formulation presented above works well in regimes where the mass of the star is not so large that it significantly "warps" space-time. General relativistic (GR) effects become important when the (dimensionless) quantity GM/c^2R becomes non-negligible. This is the case for the typical neutron star dimensions given above.

The relativistic corrections to the equation of hydrostatic equilibrium (Eq. (12.1)) can be expressed in terms of three additional (dimensionless) factors to give the Tolman-Oppenheimer-Volkov (TOV) equation (Shapiro & Teukolsky 1983, Weinberg 1972):

$$\frac{dp}{dr} = -\frac{G\epsilon(r)\mathcal{M}(r)}{c^2r^2}\left[1 + \frac{p(r)}{\epsilon(r)}\right]\left[1 + \frac{4\pi r^3 p(r)}{\mathcal{M}(r)c^2}\right]\left[1 - \frac{2G\mathcal{M}(r)}{c^2 r}\right]^{-1} \tag{12.5}$$

The factors in square brackets represent GR corrections. The first two of these reduce to 1 when $p \ll \epsilon$, while the third reduces to 1 when $G\mathcal{M}/c^2r \ll 1$. Note that the correction factors are all positive definite so that non-Newtonian gravity becomes stronger at every r; general relativity strengthens the relentless pull of gravity. The coupled non-linear equations for $p(r)$ and $\mathcal{M}(r)$ can still be integrated from $r = 0$ for a starting value of p_0 to the point where $p(R) = 0$, to determine the star mass $M = \mathcal{M}(R)$ and radius R for this value of p_0.

These equations invoke a balance between gravitational forces and the internal pressure. The pressure is a function of the EoS, and for certain conditions it may not be sufficient to withstand the gravitational attraction. Thus the structure equations also imply there is a maximum mass that a star or compact remnant can have.

12.3.1 White Dwarf Stars

A white dwarf star is a low- or medium-mass star near the end of its lifetime, having burned up, through nuclear processes, most of its hydrogen and helium forming carbon, silicon and (perhaps) iron. They

typically have a mass less than 1.4 times that of our Sun ($M_\odot = 1.989 \times 10^{30}$ kg). They are also much smaller than our Sun, with radii of the order of 10^4 km (to be compared with $R_\odot = 6.96 \times 10^5$ km). Since $GM/c^2R \approx 10^{-4}$ for such a typical white dwarf, it suffices to solve the Newtonian structure equations, Eqs. (12.1)-(12.3).

The reason a dwarf star is small is because, having burned up all the nuclear fuel it can, there is no longer enough thermal pressure to prevent its gravity from crushing it down. As the density increases, the electrons in the atoms are pushed closer together, which then tend to fall into the lowest energy levels available to them. Eventually the Pauli Exclusion Principle takes over, and electron degeneracy pressure provides the means for stabilizing the star against its gravitational attraction (Shapiro & Teukolsky, 1983). This is the physics behind the EoS which one needs in order to integrate Eqs. (12.1) and (12.2) for a compact stellar remnant like a white dwarf.

12.4 Fermi Gas Model for Electrons

For free electrons the number of states dn available at momentum k per unit volume is:

$$dn = \frac{d^3k}{(2\pi\hbar)^3} = \frac{4\pi k^2 dk}{(2\pi\hbar)^3} \tag{12.6}$$

For a Fermi gas at zero temperature, all states are filled up to the Fermi momentum k_F. Integrating, one gets the electron number density:

$$n = \frac{8\pi}{(2\pi\hbar)^3}\int_0^{k_F} k^2 dk = \frac{k_F^3}{3\pi^2\hbar^3} \tag{12.7}$$

The additional factor of two comes in because there are two spin states for each electron energy level. The value of k_F at the centre of a star is a parameter which varies according to the star's total mass and its composition, but which can be set in the calculations to follow. The electrons become relativistic roughly when $k_F = m_e c$.

Each electron is neutralized by a proton, which in turn is accompanied in its atomic nucleus by a neutron (or perhaps more as in the case of a nucleus like $^{56}Fe_{26}$). Thus, neglecting the electron mass m_e with respect to the nucleon mass m_N, the mass density of the star is essentially given by:

$$\rho = n m_N A/Z \tag{12.8}$$

where A/Z is the number of nucleons per electron. For $^{12}C_6$, $A/Z = 2.00$, while for ^{56}Fe, $A/Z = 2.15$. Note that, since n is a function of k_F, so is ρ. Conversely, given a value of ρ,

$$k_F = \hbar\left(\frac{3\pi^2\rho}{m_N}\frac{Z}{A}\right)^{1/3} \tag{12.9}$$

The contribution to the energy density from the electrons (including their rest masses) is:

$$\begin{aligned}\epsilon_{\rm elec}(k_F) &= \frac{8\pi}{(2\pi\hbar)^3}\int_0^{k_F}(k^2c^2 + m_e^2c^4)^{1/2}k^2dk \\ &= \frac{m_e^4c^5}{\pi^2\hbar^3}\int_0^{k_F/m_ec}(u^2+1)^{1/2}u^2du\end{aligned} \tag{12.10}$$

The *total* energy density due to both nucleons and electrons is then

$$\epsilon = nm_N(A/Z)c^2 + \epsilon_{\rm elec}(k_F) \tag{12.11}$$

To get our desired EoS, we need an expression for the pressure, which can be shown to be (Shapiro & Teukolsky, 1983):

$$\begin{aligned} p(k_F) &= \frac{8\pi c^2}{3(2\pi\hbar)^3}\int_0^{k_F}(k^2c^2+m_e^2c^4)^{-1/2}k^4dk \\ &= \frac{m_e^4c^5}{3\pi^2\hbar^3}\int_0^{k_F/m_ec}(u^2+1)^{-1/2}u^4du \end{aligned} \tag{12.12}$$

In the relativistic case when $k_F \gg m_ec$ Eq. (12.12) simplifies to:

$$p(k_F) \approx \frac{m_e^4c^5}{3\pi^2\hbar^3}\int_0^{k_F/m_ec}u^3du = \frac{m_e^4c^5}{12\pi^2\hbar^3}(k_F/m_ec)^4 = \frac{\hbar c}{12\pi^2}\left(\frac{3\pi^2Z\rho}{m_NA}\right)^{4/3} = K_{\rm rel}\,\epsilon^{4/3} \tag{12.13}$$

where

$$K_{\rm rel} = \frac{\hbar c}{12\pi^2}\left(\frac{3\pi^2Z}{Am_Nc^2}\right)^{4/3} \tag{12.14}$$

A star having a simple EoS like $p = K\epsilon^\gamma$ is called a "polytrope", and the relativistic electron Fermi gas model gives a polytropic EoS with $\gamma = 4/3$. A polytropic EoS allows the structure equations to be solved (numerically) in a relatively straight-forward way. There is another polytropic EoS corresponding to the non-relativistic limit, where $k_F \ll m_ec$. In a way similar to the derivation of Eq. (12.13), one finds

$$p = K_{\rm non-rel}\epsilon^{5/3} \quad \text{where} \quad K_{\rm non-rel} = \frac{\hbar^2}{15\pi^2m_e}\left(\frac{3\pi^2Z}{Am_Nc^2}\right)^{5/3} \tag{12.15}$$

12.5 Integrating the Polytrope Numerically

The Milestone Problem to calculate numerically the Chandrasekhar mass limit requires you to integrate the coupled first-order ordinary differential equations (ODEs), Eqs. (12.1) and (12.2), out from the origin, $r = 0$, to the point R where the pressure falls to zero, $p(R) = 0$. To do this you will need two initial values, $p(0)$ (which must be positive) and $\mathcal{M}(0)$ (which by definition must be 0). The star's radius, R, and its mass $M = \mathcal{M}(R)$ (it is convenient to convert these to units of km and $M_\odot$) will vary, depending on the choice for $p(0)$. The central pressure $p(0)$ can be derived from the central density using Eq. (12.4) together with either Eq. (12.14) or (12.15) depending on the value of k_F.

Python provides a suite of numerical differential equation integration routines via the the *scipy.integrate* module. However, you will have to ensure that these two equations are solved simultaneously. There are some hints on how to do this for another classical coupled first order, non-linear, ODE problem (the 'predator-prey' problem) available here:

https://scipy-cookbook.readthedocs.io/items/LoktaVolterraTutorial.html

12.6 Your Work Plan

12.6.1 Design Your Program

Your first task is to design your program. Use the pseudocode techniques to design a program that you will be able to use to study the interior structure of white dwarfs under the relativistic and non-relativistic polytrope approximations. As well as an outline of the code structure, you should include: (a) names and description of the functions you will use: what variables will they take as arguments, what will they return ? (b) what data will you input into the program, how will the program output the results ? (c) what data structures will the program use to store its internal data ? Remember to make your program sufficiently flexible to allow you to tackle new problems as well as the milestone example below.

12.6.2 Solve the Milestone Problem: The Chandrasekhar Mass Limit for White Dwarfs

Write a code to derive the mass-radius relations between total mass and total radius for white dwarfs under the relativistic and non-relativistic polytrope approximations. You should do this for a realistic range of white dwarf masses and remember to switch between the non-relativistic (low mass) and relativistic (high mass) forms of the polytropic EoS as appropriate (you can do this based on the input value of the central density). You can assume that the electrons make a negligible contribution to the total energy density. Assume that the chemical composition is pure $^{12}C_6$. Plot the relation and us it to derive the so-called Chandrasekhar mass limit (maximum mass) for white dwarfs and compare your results with literature values.

12.6.3 Research with your program

Now you have a working code, you can modify it to deal with more realistic scenarios. There are several options suggested below for developing the program. Choose one of them.

- The relativistic and non-relativistic polytropic equations of state in the Milestone Problem are assumed to hold throughout the star. In practice, this is not very physical since, even in the relativistic case, the pressure (and density) drops to zero at the surface of the star. Evaluate Eq. (12.12) numerically over the range $0.01 \leq k_F/m_e c \leq 10$ and use a log-log plot to demonstrate that the power law relativistic and non-relativistic approximations discussed above hold asymptotically in the limits $k_F \gg m_e c$ and $k_F \ll m_e c$. Hence find a fitting function of the form:

$$\epsilon(p) = A_{\rm NR} p^{3/5} + A_{\rm R} p^{3/4} \tag{12.16}$$

  to define an EoS covering all relevant pressure regimes. Use this fitted function for $\epsilon(p)$ along with the structure equations to derive the interior pressure profile and enclosed mass distribution for a white dwarf like Sirius B with $M = 0.978 M_\odot$ (assume the composition is still pure $^{12}C_6$). Compare these profiles with those obtained using the relativistic and non-relativistic polytropic equations of state defined in the Milestone Problem. Some useful information may be found in Tooper (1964).

- The analysis in the Milestone Problem can easily be extended to even more compact objects such as neutron stars. In this case we must include the general relativistic (GR) contributions represented by the three dimensionless factors in the TOV equation Eq. (12.5). Following the approach in

Oppenheimer (1939), for a pure neutron star Fermi gas EoS one can proceed much as in the white dwarf case, substituting the neutron mass m_n for the electron mass m_e. When $k_F \ll m_n c$ one again finds a polytrope with $\gamma = 5/3$ and the K corresponding to that in Eq. (12.15) is:

$$K_{\rm non-rel} = \frac{\hbar^2}{15\pi^2 m_n}\left(\frac{3\pi^2}{m_n c^2}\right)^{5/3} \tag{12.17}$$

In the relativistic regime, $k_F \gg m_n c$ there is again a polytropic EoS, but with $\gamma = 1$ and $K_{\rm rel} = 1/3$ ($p = \epsilon/3$ should be a familiar result for a relativistic photon gas). Using an approach analogous to that for the white dwarfs in the Milestone Problem, derive the mass-radius relations for pure neutron stars under the above approximations and compare them with the results obtained if the GR corrections in the TOV equation are ignored.

- An EoS for a non-interacting neutron Fermi gas which works for all values of the relativity parameter $x = k_F/m_n c$ can be obtained by fitting to the energy density as a function of pressure, with each given as a transcendental function of k_F, using a function of the form:

$$\epsilon(p) = A_{\rm NR} p^{3/5} + A_{\rm R} p \tag{12.18}$$

For low pressures the non-relativistic first term dominates over the second as required. Create a table of exact ϵ and p values as a function of k_F and use these to fit for the A-values (an accuracy of better than 1% over most of the range of k_F should be sufficient). Use these to calculate $p(r)$ and $\mathcal{M}(r)$ (r in km) for a pure neutron star using a Fermi gas EoS valid for all values of k_F. Using the general relativistic corrections (TOV equation) make a long run of calculations for a range of $p(0)$ values and then make a (parametric) plot of M vs R as a function of the central pressure. Hence derive the maximum mass of a pure neutron star with a Fermi gas EoS and estimate the value of $p(0)$ at which this occurs. A discussion of the stability of such models can be found in Weinberg (1972).

- Modifications to the EoS are also possible which include the (non-negligible) effects of nucleon-nucleon interactions. These can be estimated by constructing a simple model for the nuclear potential that reproduces the features of (normal) nuclear matter (e.g. Prakash 1997). For non-symmetric nuclear matter, where $n_n \neq n_p$ a good fit can be obtained with a polytropic fit of the form

$$p(\epsilon) = \kappa_0\, \epsilon^{\gamma} \tag{12.19}$$

This polytrope can be used in solving the TOV equation for a pure neutron star with nuclear interactions. Calculate the mass-radius plot for this model and compare with the results for the Fermi gas neutron star model.

- One way (of several) to interpret the observed acceleration of the universal expansion is via Einstein's cosmological constant, which contributes a term $\Lambda g_{\mu\nu}$ to the right-hand side of Einstein's field equation, the basic equation of GR. If Λ *is* non-zero, it is nonetheless surprisingly small. We can also examine the effect of a non-zero cosmological constant on the internal structure of a neutron star. It turns out that the only modification to the TOV equation (Vinayaraj & Kuriakose, 2008) is in the correction factor:

$$\left[1 + \frac{4\pi r^3 p(r)}{\mathcal{M}(r)c^2}\right] \rightarrow \left[1 + \frac{4\pi r^3 p(r)}{\mathcal{M}(r)c^2} - \frac{\Lambda r^3}{2G\mathcal{M}(r)}\right] \tag{12.20}$$

Estimate a value of Λ (in the appropriate units) from recent cosmological data (e.g. Planck microwave background results) and then examine what its effect might be on the structure of a typical neutron star by comparing the results with this modified TOV equation against the case of zero cosmological constant.

12.7 References

Carrol, B. & Ostlie, D., 2007, *An Introduction to Modern Astrophysics* (2nd ed) (Pearson).

Shapiro. S.L. & Teukolsky, S.A., 1983, *Black Holes, White Dwarfs and Neutron Stars: The Physics of Compact Objects* (Wiley-Interscience).

Weinberg, S., 1972, *Gravitation and Cosmology* (John Wiley & Sons).

Tooper, R.F., 1964, ApJ, 140, 434.

Oppenheimer, J.R. & Volkoff, G.M., 1939, Phys. Rev., 55, 374.

Prakash, M., 1997, AIP Conference Proceedings, Vol. 412, 1007.

Vinayaraj, K. & Kuriakose, V.C., 2008, arXiv0802.1155.

Part IV

Particle Physics

Chapter 13

Quarkonium

13.1 Background

In the same way that an electron and proton are bound together by the electromagnetic force to make a hydrogen atom, a quark and antiquark can be bound together by the strong force to form a bound state called *quarkonium.*

Just as we can infer the Coulomb force between the electron and proton by studying the energy levels of the Hydrogen atom we can study the strong force between charm and bottom quarks by looking at the energy levels of the charmonium and bottomonium systems. By studying quarks that are heavy in comparison to their binding energy, we can apply methods from non-relativistic quantum mechanics that have been very successful in explaining the properties of the Hydrogen atom.

If we neglect spin effects the wavefunction for quarkonium satisfies the non-relativistic three-dimensional Schrödinger equation,

$$-\frac{\hbar^2}{2\mu}\nabla^2\psi + [V(r) - E_{nl}]\,\psi = 0, \tag{13.1}$$

where μ is the reduced mass, $1/\mu = 1/m_1 + 1/m_2$, $m_{1,2}$ are the masses of the quarks and r is the distance between the quarks.

A suitable parameterisation of the interquark potential is

$$-\frac{4\alpha_S}{3r} + \beta r, \tag{13.2}$$

where α_S is the dimensionless strong coupling constant.[1] The second term represents a confining potential and is added to ensure that the quarks are bound and cannot exist as free particles.

The spherically symmetry of the potential ensures that

$$\psi = R_{n\ell}(r)Y_{\ell m}(\theta,\phi), \tag{13.3}$$

where n, ℓ and m represent the usual quantum numbers describing the different energy levels. Here $Y_{\ell m}(\theta,\phi)$ are the standard spherical harmonics and the radial wavefunction $u_{n\ell}(r) = rR_{n\ell}(r)$ satisfies

$$\frac{d^2u_{n\ell}}{dr^2} - \frac{l(l+1)}{r^2}u_{n\ell} + 2\mu(E_{n\ell} - V(r))u_{n\ell} = 0. \tag{13.4}$$

[1] This is the equivalent of the fine-structure constant in electromagnetism for strong interactions.

In this project we will use natural units $\hbar = c = 1$ so that μ is measured in GeV/c^2, $E_{n\ell}$ in GeV, r in GeV^{-1} and β in GeV^2. The wavefunctions are normalised such that,

$$\int_0^\infty r^2 |R_{n\ell}(r)|^2 dr = \int_0^\infty |u_{n\ell}|^2 dr = 1. \tag{13.5}$$

The aim of this project is to study the properties of these quarkonium states using numerical calculations of the wavefunctions to obtain the energy levels. Once we have calculated the binding energy, the mass of the assoicated quarkonium states is

$$M_{n\ell} = m_1 + m_2 + E_{n\ell}. \tag{13.6}$$

A more detailed discussion of the physics of charmonium can be found in (eg.) Donoghue et al. 1992.

13.2 Numerical Solution

In order to solve Schrödinger's equation numerical we need to specify boundary conditions. In particular, as the wavefunction is well behaved at $r = 0$, we know that

$$u_{n\ell}(0) = 0. \tag{13.7}$$

We will also take

$$\frac{du_{n\ell}(0)}{dr} \neq 0, \tag{13.8}$$

so that if we do not care about the normalisation of the wavefunction, we can choose,

$$\frac{du_{n\ell}(0)}{dr} = 1. \tag{13.9}$$

Formally this is strictly only valid for $l = 0$ however in practice the loss of numerical accuracy due to this choice is small.

As $r \to \infty$

$$u_{n\ell}(r) \to 0, \qquad \frac{du_{n\ell}(r)}{dr} \to 0. \tag{13.10}$$

These equations can be solved numerical using standard techniques for ordinary differential equations. The scipy.integrate module contains a good routine for solving a system of linear first order differential equations, odeint, which you can use. An example of how to use this function is given in the appendix. In order to use standard packages for solving differential equations we need to transform the 2nd order differential equation we wish to solve into two first order equations. This is easily achieved by defining $v_{n\ell} = \frac{du_{n\ell}}{dr}$, so that we have two equations

$$\frac{du_{n\ell}}{dr} = v_{n\ell}, \tag{13.11a}$$

$$\frac{dv_{n\ell}}{dr} = \frac{l(l+1)}{r^2} u_{n\ell} - 2\mu(E_{n\ell} - V(r))u_{n\ell}. \tag{13.11b}$$

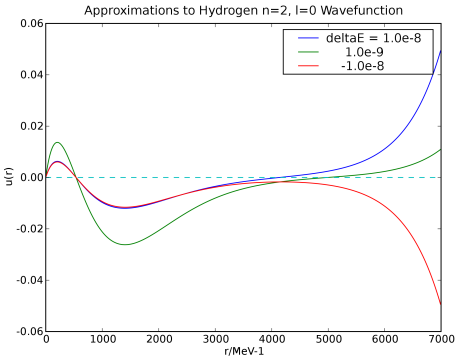


Figure 13.1: The effect of choosing an approximate energy on the function $u(r)$. When the energy is too high, the function crosses the axis twice. When it is too low, there is an extra turning point. For a very small offset in energy, the wavefunction is almost perfect, but just crosses the axis and diverges at large r. The exact solution will get close to the axis but never cross it.

You will need the solution out to a reasonable distance from the origin, $r_{\max} = 15\,\mathrm{GeV}^{-1}$, should be sufficient but you should check the effect of this choice on your results.[2]

If we do not choose the correct energy eigenvalue the solution will not satisfy Equation 13.10, in particular it will diverge to either positive or negative infinity. We know this happens because the wavefunction will have either too many nodes (crossing the x-axis) or turning points. For $l = 0$, the number of nodes and turning points should be $n - 1$ and n, respectively. By counting the number of nodes and turning points at some trial values of the energy we can iterate towards the correct solution. If the trial energy is too large then the number of nodes will be too large. If the trial energy is too small, then number of nodes will be correct or too small, and there will be too many turning points. We can then guess a new energy and repeat the procedure. This is illustrated in Fig. 13.1.

Fortunately, we do not need to know the "correct" number of nodes and turning points at the outset. The best procedure to use is to use three test values for the energy $E_{1,2,3}$, with $E_2 = \frac{1}{2}(E_1 + E_3)$. We can then check the number of nodes and turning points at each value. If the number of nodes and turning points changes between E_1 and E_2 we can set $E_3 = E_2$ and repeat the procedure, or if these change between E_2 and E_3 we can set $E_1 = E_2$ and repeat. This will allow you iterate until you have an accurate answer for the energy. It is important to check that $u_{n\ell}$ has the correct behaviour by plotting it. (Note that the solution will never be exact so you should not be surprised if $u_{n\ell}(r)$ diverges at large r). The value of β can be calculated in a similar way, but as it occurs with opposite sign to $E_{n\ell}$, if there are too many nodes then β is too small etc.. This approach requires that the correct value lies between E_1 and E_3.

The wavefunction should be normalised as in Equation 13.5. You can achieve this by integrating your result, however you must truncate the integral at a suitable value of r if $u_{n\ell}(r)$ starts to diverge. You could simply sum-up an array of $|u(r)|^2$ values and multiply by the step size, or you could use the simps routine in the scipy.integrate package can be used to compute the numerical integral.

[2] In practice you will probably find it necessary to start a small distance away from the origin to avoid numerical instabilities for l states.

13.3 Work Plan

1. **Design your program.** The first task is to design your program. Use pseudocode techniques to design a program that you will use to investigate quarkonium systems.

   As well as an outline of the code structure, you should include: (a) the names and descriptions of the functions you will use: what variables will they takes as arguments and what will they return? (b) what data you will input to the program, how the program output will the results? (c) what data structures the program will use to store its internal data?

   Your program will need to (1) solve the Schrödinger equation numerically for specific values of the quark masses, α_S, β and the energy; (2) calculate the number of nodes and turning points; (3) guess a new energy (or β) and iterate to find a bound solution. Once you have this basic building block you can write functions to find either the energy given β or β given the energy, and plot the resulting wavefunctions.

2. **Solve the Milestone Program.** It is extremely useful to check this program by using it to calculate the energy levels and wavefunctions of the hydrogen atom, where we have known analytic results. You can do this be making the following replacements $\mu = m_e$, $\frac{4}{3}\alpha_S \to \alpha = 1/137$, $\beta \to 0$. *You should take care with your units as the radius of the Hydrogen atom is very different to that of Charmonium.* You will find it easier to use MeV rather than GeV as your natural units, in this case. You should obtain the wavefunctions and energy levels of the $(n, \ell) = (1, 0)$, $(2, 0)$ and $(2, 1)$ states and check them against the known analytic results. In natural units $E_{n\ell} = -\mu\alpha^2/2n^2$ and the Bohr radius $a_0 = 1/(\mu\alpha)$. The wavefunctions can be found in many quantum mechanics books, for example, Appendix A of Sakurai (1994).

   For the milestone your program should calculate the energy of the $(2, 1)$ state and plot the corresponding radial probability density function, $|u_{n\ell}|^2$.

3. **Extend your program**

   Now it is time to apply your program to study the properties of Charmonium. As a starting point, you should calculate β to 3 d.p. using the spin average Charmonium mass given in the appendix. You can then use this to give predictions for the energy of the $(n, \ell) = (1, 1)$ and $(2, 0)$ levels and produce plots of the normalised wavefunctions for the $(n, \ell) = (1, 0),\ (1, 1)$ and $(2, 0)$ levels.

4. **Research with your program**

   Once you have your basic program working for Charmonium there are a range of things you can study. You should not look into all of these issues, they are intended as a starting point for your own investigations. It is more important to investigate one point in depth than to cover a wide range of phenomena.

   - The spectra of charmonium and bottomonium states can be studied and compared to the observed values. The first state containing a bottom and charm quark, the B_c meson, been confirmed at Fermilab with mass 6.276 GeV, you can look at the properties of these states using your program. Remember that the potential you are using is an approximation to the full QCD interactions of the quarks. You will find recent summary of known quarkonia transitions in Eichten et al 2007. The latest data is available on-line at www.pdg.gov.
   - The choice we used for the potential is only one of a number which display the correct behaviour at both small and large distances. You can investigate the effect of different choices of potential

on your results. Another popular choice is

$$V(r) = cr^d + A \qquad \begin{cases} c \approx 6.870\,\text{GeV}, \\ d \approx 0.1, \\ A \approx -8.064\,\text{GeV}. \end{cases} \tag{13.12}$$

Constant values are derived from Martin (1981). Bear in mind that it may be necessary to tune some of these constants to reflect your specific mass choice. A discussion of the potentials for quarkonium can be found in Chapter XIII of Donoghue et al. (1992).

- The quarks have spin-$\frac{1}{2}$, therefore the s-wave ($l = 0$) states can be divided into spin-singlets and spin-triplets (as in the Helium atom). These states do not have exactly the same energies and are split by a hyperfine spin-spin interaction. This hyperfine splitting is

$$\Delta(n^3S_1 - n^1S_0) = \frac{8\alpha_S}{9m_q^2}|R_{n\ell}(0)|^2. \tag{13.13}$$

It thus depends on the amplitude of the wave function at the origin. In order to determine this, note that

$$\lim_{r\to 0} R_{n\ell}(r) = \lim_{r\to 0} \frac{u_{n\ell}(r)}{r} = \frac{du_{n\ell}(0)}{dr}. \tag{13.14}$$

The transition rates between the states provide deep insight into the nature of the strong force. More discussion can be found in Donoghue et al. (1992).

- Quarkonia have relatively short lifetimes. The lifetime of a state is largely determined by the probability that the quarks overlap, ie. $|\Phi(r = 0)|^2$ (eg., Eichten et al. 2007). Your program allows you to determine the (ground state) wavefunction and hence to compare the decay rates of different systems. By analogy with the hydrogen atom you can compute the wave function and use this to compute other properties of quarkonia.

Appendix: Quarkonium Data

The data for charmonium is,

$$\begin{aligned} m_c &= 1.34\text{GeV/c}^2 \\ \alpha_S(4m_c^2) &= 0.40, \\ M_{n=1,l=0} &= 3.068\text{GeV/c}^2 \end{aligned} \tag{13.15}$$

The data for bottomonium is,

$$\begin{aligned} m_b &= 4.70\text{GeV/c}^2 \\ \alpha_S(4m_b^2) &= 0.28, \end{aligned} \tag{13.16}$$

Recent experimental values for the mass of quarkonium states can be found in Amsler (2008).

References

F. Halzen and A. D. Martin, "Quarks And Leptons: An Introductory Course In Modern Particle Physics, Chapter 2.13," *Wiley (1984).*

J. J. Sakurai, "Modern Quantum Mechanics," *Addison-Wiley (1994).*

C. Amsler *et al.* [Particle Data Group], "Review of particle physics," Phys. Lett. B **667** (2008) 1. Available on line at http://pdg.lbl.gov/

A. Martin, "A simultaneous fit of $b\bar{b}$, $c\bar{c}$, $s\bar{s}$ (bcs pairs) and $c\bar{s}$ spectra," Phys. Lett. B **100**, 6, (1981).

J. F. Donoghue, E. Golowich and B. R. Holstein, "Dynamics Of The Standard Model," Camb. Monogr. Part. Phys. Nucl. Phys. Cosmol. **2**, 1 (1992). Available on google books. Pages 350-359 are relevant.

Rosenfeld R., 2013. http://www.ifsc.usp.br/ lattice/Natal_1.pdf. A recent quick overview of the standard model and QCD.

Chapter 14

3-particle Quantum Mechanics

14.1 Background

In quantum mechanics the Schrödinger equation can only be solved exactly to give the energy levels for a relatively small number of physical systems, such as the hydrogen atom. However there are a wide range of approximate and numerical approaches which enable more complicated systems to be studied. The aim of this project is to use one such approach to study the energy levels of the helium atom, historically this method was first used in 1929 by Hylleraas. The *variational method* is described in the Schiff and (more briefly) Sakurai text books.

If we construct a set of N states Φ_i which depend on an arbitrary parameter κ, and are not necessarily eigenstates of the Hamiltonian, we can compute the expectation value of the Hamiltonian, H,

$$\langle \Phi_i | H | \Phi_j \rangle = H_{ij}. \tag{14.1}$$

We can now find the eigenstates of Hamiltonian, which we assume to be a linear combination of the states Φ_i, as $\Psi = \sum_{j=1}^{N} a_j \Phi_j$, such that

$$H|\Psi\rangle = \sum_{j=1}^{N} a_j H|\Phi_j\rangle = E|\Psi\rangle = E\sum_{j=1}^{N} a_j |\Phi_j\rangle, \tag{14.2}$$

where E is the energy of the eigenstate. If we now take the expectation value with $\langle \Phi_i|$ we can write this as a matrix equation,

$$\sum_{j=1}^{N} H_{ij} a_j = E a_i. \tag{14.3}$$

This equation can be written in terms of matrices with H_{ij} being the ith row and jth column of a matrix H and a_j the jth entry of a column vector. The problem of finding the energy eigenvalues now amounts to finding the eigenvalues of the matrix H.

The theorem of Hylleraas and Undheim states that the eigenvalues E of Equation 14.3 are greater than or equal to the exact eigenvalues of the Schrödinger equation regardless of the number of states used or the parameter κ, *i.e.*

$$E_{\text{exact}} \leq E. \tag{14.4}$$

This approach will allow us to find not only an estimate for the true energy eigenvalues but also assess the quality of the approximation using the following approach:

1. using different values of the parameter κ and obtaining a minimum for the eigenvalue which will be the best estimate;

2. plotting the variation of the eigenvalue as a function of κ and checking the result converges with the minimum becoming broader and flatter as more basis states are used.

14.2 Numerical Solution

The Hamiltonian for the Helium atom can be written as

$$H = -\frac{\hbar^2}{2m_e}\left(\Delta_1 + \Delta_2\right) - \frac{Ze^2}{|\vec{r}_1|} - \frac{Ze^2}{|\vec{r}_2|} + \frac{e^2}{|\vec{r}_1 - \vec{r}_2|}, \tag{14.5}$$

where

$$\Delta_i = \frac{1}{r_i^2}\frac{\partial}{\partial r_i}\left(r_i^2\frac{\partial}{\partial r_i}\right) + \frac{1}{r_i^2\sin\theta_i}\frac{\partial}{\partial\theta_i}\left(\sin\theta_i\frac{\partial}{\partial\theta_i}\right) + \frac{1}{r_i^2\sin^2\theta_i}\frac{\partial^2}{\partial^2\phi_i} \tag{14.6}$$

is the Laplace operator for the ith electron, $\vec{r}_i$ is position vector of the ith electron relative to the nucleus, r_i is the magnitude of the position vector of the ith electron, Z is the nuclear charge number, m_e the electron mass and e is the magnitude of the electron's charge in "natural" units (where the prefactor $\frac{1}{4\pi\epsilon_0}$ has been absorbed into the definition of electric charge). **For this problem we will use Angstroms as the unit of distance and eV as the unit of energy**, in these units $e^2 = 14.4\text{eV}\text{Å}$ and the Bohr radius $r_0 = \frac{\hbar^2}{e^2 m_e} = 0.52917\text{Å}$.

We need to make a choice of the basis states, one choice physically appealing choice would be to use a product of single particle states using the wavefunctions of the hydrogen atom for each of the electrons. However a better choice is to use the wavefunctions

$$\tilde{\Phi}_{jkm}(\vec{r}_1,\vec{r}_2) = (r_1+r_2)^j(r_1-r_2)^k\,|\vec{r}_1-\vec{r}_2|^m\exp\left[-\frac{Z}{\kappa r_0}(r_1+r_2)\right] \qquad (j,k,m\geq 0). \tag{14.7}$$

The tilde here indicates that these basis states are not orthogonal. As before the wavefunction can be written as

$$\Psi(\vec{r}_1,\vec{r}_2) = \sum_{j,k,m}\tilde{a}_{jkm}\tilde{\Phi}_{jkm}(\vec{r}_1,\vec{r}_2). \tag{14.8}$$

In the original calculation Hylleraas only used three states, $(j,k,m) = (0,0,0)$, $(0,0,1)$, and $(0,2,0)$. For brevity we will often denote (j,k,m) by n and (j',k',m') by n'.

As we are using a non-orthogonal basis Equation 14.3 becomes

$$\sum_{n'}\tilde{H}_{nn'}\tilde{a}_{n'} = E\sum_{n'}N_{nn'}\tilde{a}_{n'}, \tag{14.9}$$

where

$$\tilde{H}_{nn'} = \langle\tilde{\Phi}_n|H|\tilde{\Phi}_{n'}\rangle, \tag{14.10a}$$
$$N_{nn'} = \langle\tilde{\Phi}_n|\tilde{\Phi}_{n'}\rangle. \tag{14.10b}$$

These equations are much more conveniently written in a matrix notation. If we take $\tilde{H}$ to be a matrix whose entry in the nth row and n'th column is $\tilde{H}_{nn'}$, N to be a matrix whose entry in the nth row and n'th

column is $N_{nn'}$ and $\tilde{A}$ to be a column vector whose n'th entry is $\tilde{a}_{n'}$. We can write a matrix eigenvalue equation

$$\tilde{H}\tilde{A} = EN\tilde{A}. \tag{14.11}$$

This equation can be solved by first finding the eigenvalues, β_i, and eigenvectors $y^i_{n'}$ of $N_{nn'}$,

$$\sum_{n'} N_{nn'} y^i_{n'} = \beta_i y^i_n, \tag{14.12}$$

for the ith eigenvector and eigenvalue of N. This allows us to write matrices

$$\beta_{ii'} = \delta_{ii'}\beta_i \qquad\qquad Y_{ni} = y^i_n, \tag{14.13}$$

i.e. the matrix β is a diagonal matrix whose diagonal elements are the eigenvalues of N and Y is a matrix whose columns are the eigenvectors of N.

This allows us to write N as

$$N = Y\beta Y^T, \tag{14.14}$$

where Y^T is the transpose of Y.

The matrix β is diagonal and therefore we can define the matrices $\beta^{\frac{1}{2}}$ and $\beta^{-\frac{1}{2}}$ which have elements

$$\beta^{\frac{1}{2}}_{ii'} = \delta_{ii'}\sqrt{\beta_i}, \qquad\qquad \beta^{-\frac{1}{2}}_{ii'} = \delta_{ii'}\frac{1}{\sqrt{\beta_i}}. \tag{14.15}$$

We can rearrange Equation 14.11 by multiplying from the left by $\beta^{-\frac{1}{2}}Y^T$ and inserting the identity matrix $I = Y\beta^{-\frac{1}{2}}\beta^{\frac{1}{2}}Y^T$ between H and $\tilde{A}$ giving

$$Pz = Ez, \tag{14.16}$$

where $P = \beta^{-\frac{1}{2}}Y^T\tilde{H}Y\beta^{-\frac{1}{2}}$ and $z = \beta^{\frac{1}{2}}Y^T\tilde{A}$.

The energy levels of the helium atom can now be found by obtaining the eigenvalues of the matrix P.

14.3 Implementing your program in Python

The numpy module contains matrices which we can use to store the matrices we need and the scipy.linalg module contains the eig routine for finding the eigenvalues and eigenvectors, in fact the eigenvectors are returned in the form of the matrix Y we require. An example of using this routine is given at the back of the book. The eigenvalues are not always sorted so that the smallest one is first, which does not matter for β since the eigenvalues and eigenvectors are in the same order, but does for the energy levels so you will need to sort the energy eigenvalues using numpy.sort() before you use them.

The analytic expressions for the expectation values you will need for N and H are given in the appendix.

14.4 Work Plan

1. **Design Your Program.** You need to a design program to find the energy levels of helium. The first component you is to be able to calculate the energy levels for a given set of states and value of κ. To do this you will need:

(a) A calculation of the matrix N;

(b) A calculation of the matrices β, $\beta^{\frac{1}{2}}$, $\beta^{-\frac{1}{2}}$, Y and Y^T (using the eigenvalues and eigenvectors of N);

(c) A calculation of the matrix $\tilde{H}$ and then P;

(d) The eigenvalues of the matrix P.

You should use pseudo-code techniques to design the program. As well as an outline of the code structure, you should include: (a) the names and descriptions of the functions you will use: what variables will they takes as arguments and what will they return? (b) what data you will input to the program, how the program output will the results? (c) what data structures the program will use to store its internal data?

2. **Solve the Milestone Program.**

   (a) To test your program is working correctly it is easiest to just use one basis state $(0, 0, 0)$. In this case the energy of the ground state is

   $$E = \frac{e^2}{r_0}\left(\frac{4}{\kappa^2} - \frac{27}{4\kappa}\right). \tag{14.17}$$

   Details of the analytic calculation of this result can be found in standard books on quantum mechanics such as Schiff or Sakurai.

   (b) However, this calculation does not properly test the eigenvector manipulation of your code, so you should now produce results for the three basis state used by Hylleraas in his original calculation (ie., the three states, $(j, k, m) = (0, 0, 0)$, $(0, 0, 1)$, and $(0, 2, 0)$).

   For the milestone your program should compute the ground state energy using the Hylleraas' three basis state and plot a graph showing the variation of the energy with κ.

3. **Extending your program** Once you have a working program you can investigate the energy levels of the helium atom and use your results to write your report. Note that the symmetry of the wave function is very important since the electrons are identical particles.

   Study the variation of the ground-state energy with κ and the selection of basis states. If you include more basis states *with the correct symmetry under the exchange of the two electrons* you should be able to reproduce the experimental result $E = 2.904e^2/r_0$. You can investigate how many states you need to obtain a result with sufficient accuracy and what choice of values of (j, k, m) gives good results.

4. **Research with your program**

   Now you have a working code, you can investigate multi-electron systems in more detail. You should not look at all the points below, they are intended as suggestions for the starting point for your investigation. It is more important to investigate one point in depth than to cover a wide range of phenomena.

   - Hyleraas' method is very powerful in that the higher energy levels are also calculated so you can study these energy levels, and "predict" the spectrum of the Helium atom. Compare the results you obtain to experimental results. Remember that your calculations are non-relativistic and

ignore vaccum (QED) interaction terms. The method can also study the energy levels of other two electron systems, *e.g.* H^- and Li^+. You can find data for atomic spectra on the NIST website
http://www.nist.gov/pml/data/asd.cfm

- By making an appropriate choice of basis states you can study the energy levels of both para- (electron spins antialigned) and ortho- (electron spins aligned) helium. This allows you to study the quantum mechanical interaction of identical particles - see Sakuraii, for example. Remember that you must use basis functions with appropriate symmetry under the exchange of the electrons.
- Using the eigenvalues you obtain allows you to reconstruct the wave-function of the electrons. This is of considerable interest, both in order to study the structure of the orbitals compared to the single electron case, to study the relative positions of the two electrons, and to compute transition probabilities and branching ratios. In some excited states, the outer electron sees a shielded nuclear charge. In these systems you should be able to approximate the solution by treating the electrons separately using a shielded nuclear charge for the outer electron.
- Hyleraas method can be extended to atoms with more electrons by adding additional terms to the Hamiltonian. The 3-electron case is discussed in Wilson 1933 (which gives relevant expressions for the matrices) and Ritter et al. 1960 (which gives references to earlier work). Three electron states can be approximated by treating the outer electron as seeing a shielded nuclear charge.
- More exotic atoms can be created by replacing the electrons by muons. Because the muon is heavier, the "atom" is smaller and the orbitals tend to overlap more. You can readily use your program to investigate how this affects the emission spectrum. Deviations from this calculation can be used to investigate QED effects and the charge structure of the nucleus (eg., Borie 2011).

14.5 Appendix: Matrix Elements

14.5.1 The General Case

You will need a number of expectation values to calculate the energy levels. The expectation value

$$N_{nn'} = \langle \tilde{\Phi}_{jkm} | \tilde{\Phi}_{j'k'm'} \rangle = N_{J,K,M}, \tag{14.18}$$

where

$$N_{J,K,M} = \begin{cases} 2\pi^2 \left(J+K+M+5\right)! \left(\frac{1}{2\lambda}\right)^{J+K+M+6} \times & K{=}0,2,4,\ldots, \\ \left(\frac{1}{M+2}\right) \times \left(\frac{1}{K+1} - \frac{1}{K+3} - \frac{1}{K+M+3} + \frac{1}{K+M+5}\right) & \\ 0 & K{=}1,3,5,\ldots, \end{cases} \tag{14.19}$$

where $J = j + j'$, $K = k + k'$, $M = m + m'$ and $\lambda = \frac{Z}{\kappa r_0}$.

The expectation value of the Hamiltonian operator has three components

$$H = T + C + W \tag{14.20}$$

where

$$C = -\frac{Ze^2}{r_1} - \frac{Ze^2}{r_2}, \qquad T = -\frac{\hbar^2}{2m_e}\left(\Delta_1 + \Delta_2\right), \qquad W = \frac{e^2}{|\vec{r}_1 - \vec{r}_2|}. \tag{14.21}$$

The expectation value for the Coulomb interaction of the electrons with the nucleus, C, is

$$\tilde{C}_{nn'} = \langle\tilde{\Phi}_{jkm}|C|\tilde{\Phi}_{j'k'm'}\rangle = -Ze^2\hat{C}_{JKM}, \tag{14.22}$$

where

$$\hat{C}_{JKM} = \begin{cases} 8\pi^2(J+K+M+4)!\left(\frac{1}{2\lambda}\right)^{J+K+M+5}\left(\frac{1}{M+2}\right)\times & K{=}0,2,4,\ldots \\ \left(\frac{1}{K+1}-\frac{1}{K+M+3}\right) & \\ 0 & K{=}1,3,5,\ldots. \end{cases} \tag{14.23}$$

The expectation value for the Coulomb interaction between the electrons is

$$\tilde{W}_{nn'} = \langle\tilde{\Phi}_{jkm}|W|\tilde{\Phi}_{j'k'm'}\rangle = e^2 N_{J,K,M-1}. \tag{14.24}$$

Finally the expectation of the kinetic energy of the electrons is

$$\begin{aligned} \tilde{T}_{nn'} = \langle\tilde{\Phi}_{jkm}|T|\tilde{\Phi}_{j'k'm'}\rangle \;=\; & \frac{\hbar^2}{2m_e}\left\{2\left[\lambda^2 N_{J,K,M} - J\lambda N_{J-1,K,M}\right.\right. \\ & \left. +jj' N_{J-2,K,M} + kk' N_{J,K-2,M} + mm' N_{J,K,M-2}\right] \\ & +\frac{1}{2}\left[-M\lambda\left(\hat{C}_{J,K,M} - \hat{C}_{J,K+2,M-2}\right)\right. \\ & +(mj'+m'j)\left(\hat{C}_{J-1,K,M} - \hat{C}_{J-1,K+2,M-2}\right) \\ & \left.\left. +(mk'+m'k)\left(\hat{C}_{J+1,K,M-2} - \hat{C}_{J-1,K,M}\right)\right]\right\}. \end{aligned} \tag{14.25}$$

In principle when $M = 0$ we need a number of expressions evaluated with $M = -2$, however these are always multiplied by zero and it is sufficient to set $N_{J,K,M}$ and $\hat{C}_{J,K,M} = 0$ in these cases.

14.5.2 The special case of one basis state

When we have only one basis sate, there is only one matrix element and the expressions above are greatly simplified.

$$N_{00} = \langle\tilde{\Phi}_{000}|\tilde{\Phi}_{000}\rangle = \pi^2\left(\frac{1}{\lambda}\right)^6 \tag{14.26}$$

$$\tilde{C}_{00} = \langle\tilde{\Phi}_{000}|C|\tilde{\Phi}_{000}\rangle = -Ze^2\,2\pi^2\left(\frac{1}{\lambda}\right)^5 \tag{14.27}$$

$$\tilde{W}_{00} = \langle\tilde{\Phi}_{000}|W|\tilde{\Phi}_{000}\rangle = e^2\,\frac{5\pi^2}{8}\left(\frac{1}{\lambda}\right)^5 \tag{14.28}$$

$$\tilde{T}_{00} = \langle\tilde{\Phi}_{000}|T|\tilde{\Phi}_{000}\rangle \;=\; \frac{\hbar^2}{2m_e}\,2\pi^2\left(\frac{1}{\lambda}\right)^4 \tag{14.29}$$

14.5.3 The special case of Hylleraas' three basis state

When we have Hylleraas' basis sate ($(j,k,m) = (0,0,0)$, $(0,0,1)$, and $(0,2,0)$) there are now 9 matrix elements, but the expressions above are still somewhat simplified. You sould use the matrices below to check that your own matrix calculation is working correctly.

$$N = \pi^2\lambda^{-6}\begin{pmatrix} 1 & \frac{35}{16}\lambda^{-1} & \frac{3}{2}\lambda^{-2} \\ \frac{35}{16}\lambda^{-1} & 6\lambda^{-2} & \frac{77}{16}\lambda^{-3} \\ \frac{3}{2}\lambda^{-2} & \frac{77}{16}\lambda^{-3} & 9\lambda^{-4} \end{pmatrix} \tag{14.30}$$

$$\tilde{C} = -Ze^2\,2\pi^2\lambda^{-5}\begin{pmatrix} 1 & \frac{30}{16}\lambda^{-1} & \frac{3}{2}\lambda^{-2} \\ \frac{30}{16}\lambda^{-1} & \frac{9}{2}\lambda^{-2} & \frac{70}{16}\lambda^{-3} \\ \frac{3}{2}\lambda^{-2} & \frac{70}{16}\lambda^{-3} & 9\lambda^{-4} \end{pmatrix} \tag{14.31}$$

$$\tilde{W} = e^2\,\frac{5\pi^2}{8}\lambda^{-5}\begin{pmatrix} 1 & 1.6\lambda^{-1} & 0.9\lambda^{-2} \\ 1.6\lambda^{-1} & 3.5\lambda^{-2} & 2.4\lambda^{-3} \\ 0.9\lambda^{-2} & 2.4\lambda^{-3} & 3.9\lambda^{-4} \end{pmatrix} \tag{14.32}$$

$$\tilde{T} = \frac{\hbar^2}{2m_e}\,2\pi^2\left(\frac{1}{\lambda}\right)^4\begin{pmatrix} 1 & \frac{25}{16}\lambda^{-1} & \frac{3}{2}\lambda^{-2} \\ \frac{25}{16}\lambda^{-1} & 4\lambda^{-2} & \frac{73}{16}\lambda^{-3} \\ \frac{3}{2}\lambda^{-2} & \frac{73}{16}\lambda^{-3} & 15\lambda^{-4} \end{pmatrix} \tag{14.33}$$

References

For completeness, I'm giving the original references, but these are in German (and not very helpful anway!). It is far better to start from a modern quantum mechanics text book.

E. A. Hylleraas and B. Undheim, Z. Phys. **65** (1930) 759. (in German), see translation in Macdonald Phy Rev D, 1933

E. A. Hylleraas and J. Midtal, 1958, Phys Rev. 103, 829

L. I. Schiff, "Quantum Mechanics", *McGraw-Hill.*

J. J. Sakurai, "Modern Quantum Mechanics" *Addison-Wiley (1994).*

Wilson E. B., 1933, Journal of Chemical Physics, 1, 210

Ritter Z. W., Pauncz R., Appel K., 1961, Journal of Chemical Physics, 35, 2

Borie E., 2011. http://arxiv.org/pdf/1103.1772v7.pdf

Chapter 15

Feynman Path Integral

15.1 Introduction

In his popular books, Feynman records how he was frustrated by the lack of an intuitive approach to quantum mechanics. It seemed to him that the Schrodinger equation appeared out of no-where and had no relation to the schemes used in classical mechanics. He felt that there ought to be another approach that is more closely connected to classical ideas of wave propagation. Eventually, he came up with an alternative formulation of quantum mechanics based on path integrals (eg., Feynman & Hibbs 1965). This project will allow you to explore these ideas, and there relation to the Schrodinger approach to quantum mechanics. To keep things simple we will just be looking at the 1-D case.

The mathematics behind the path integral is described in much more detail in Landau's text book on Computational Physics, and we will only include a brief outline here. You are strongly recommended to read through Landau's description and to investigate the minimum Action approach to classical mechanics on which the path integral formalism is based. Path integrals are also described in Sakurai's book. You should clearly separate the theoretical ideas behind the path integral approach and the mathematical tricks that are required to make the problem soluble in finite computing time. You should use the program you develop to investigate this approach in more detail comparing your results to classical mechanics and to results obtained using Schrodinger's equation.

Although the path integral approach is elegant, it is difficult to use in practice, so that most quantum mechanical solutions are based on Schrodinger's methodology of finding energy eigenstates of the Hamiltonian. In the study of the strong force, however, Path Integral methods offer the only practical means of solution. Path integral methods are also key to understanding quantum field theories. You can read more about the advanced applications in Lepage 2005 and on the UK and US QCD web sites (http://www.ukqcd.ac.uk, http://www.usqcd.org).

15.2 Feynman Path Integrals

Applying Schrodinger's equation, the wavefunction of a particle propagates in space-time from point a to point b according to the relation

$$\Phi(x_b, t_b) = \int G(x_b, t_b; x_a, t_a)\Phi(x_a, t_a)dx_a \tag{15.1}$$

For a free particle (ie., when V=0 everywhere),

$$G(x_b, t_b; x_a, t_a) = G(b, a) = \sqrt{\frac{m}{2\pi\hbar i(t_b - t_a)}} e^{i\frac{m(x_b - x_a)^2}{2\hbar(t_b - t_a)}} \tag{15.2}$$

This formulation is similar to Huygens' principle in optics. Each point on the wavefront emits a spherical wave; the new wavefront is by superposing these wavelets.

Feynman realised that this could be understood in another way, similar to Fermat's principle of stationary paths, or Hamilton's principle of least action. In classical mechanics, the *action* is defined as

$$S[x(t)] = \int_{t_a}^{t_b} L[x(t), \dot{x}(t)]dt \tag{15.3}$$

where $L = T[x, \dot{x}] - V[x]$ is the Lagrangian of the system. The square brackets indicate that S and L are *functionals* of the entire path, not just dependent on one particular value.

For a free particle,

$$S[b, a] = \frac{m(x_b - x_a)^2}{2(t_b - t_a)} \tag{15.4}$$

This is very similar to Eq. 15.2, suggesting a more general equation for G,

$$G(b, a) = \sqrt{\frac{m}{2\pi\hbar i(t_b - t_a)}} e^{iS[b,a]/\hbar} \tag{15.5}$$

Feynman suggested that a quantum mechanical version of the classical theory could be created by summing the expression over all possible paths that connect a and b.

$$G(b, a) = \sum_{\text{paths}} e^{iS[b,a]/\hbar} \tag{15.6}$$

This is the Feynman Path Integral (remember that S[a,b] is an integral over the path, not just dependent on a and b. We can interpret the exponential as giving a weighted "probability" to each path connecting a and b.

If we knew the wave-function at some time, t_a (for example it might be a delta function if we have just observed the particle to be at point a), we can use this formalism to determine the wavefunction at a new time.

15.3 Numerical Solution

Solving the path integrals directly is complicated. We start by discretizing space so that rather than considering an integral over continuous paths, we consider paths in which the particle 'hops' between points. The other issue is that the exponential involves multiplying by i: this makes the contributions from different paths oscillate. The integral is therefore extremely difficult to evaluate in this form, indeed it may not even converge. A 'trick' that can be used to evaluate ground state configurations is to use imaginary time, $\tau = -it$. This is described in detail in Landau's book. This transformation has the effect of converting the Lagrangian in the original formula into a Hamiltonian which is evaluated at real time τ.

$$S = \int L dt = i \int H d\tau \tag{15.7}$$

The net result of these approximations is that the propagator becomes

$$G(x, -i\tau; x_0, 0) = \mathcal{A} \int e^{-\epsilon \mathcal{E}(x_j)} dx_1 ... dx_{N-1} \tag{15.8}$$

where there are N time points, separated by ϵ, and indexed by j so that $\tau = j\epsilon$. $\mathcal{E}$ is the summed energy along the path,

$$\mathcal{E} = \sum_{j=1}^{N} \left[\frac{m}{2} \left(\frac{x_j - x_{j-1}}{\epsilon} \right)^2 + V \left(\frac{x_j + x_{j-1}}{2} \right) \right] \tag{15.9}$$

and $\mathcal{A}$ is a normalisation factor. *In these equations, we have used the convention $\hbar = 1$ to make things simpler.* A path is described by the set of discrete values, x_j, and the propagator $G(x, -i\tau; x_0, 0)$ is an appropriately weighted sum over the all possible paths. This propagator allows us to determine the probability distribution of the particle position at "time" τ.

Your Milestone program will demonstrate that this works and gives the same answer as the conventional quantum mechanical approach. Notice that, in principle, equation 15.8 requires the evaluation for all possible paths, however we will examine a more effective way of proceeding. Using a small number of steps, your programme will examine a set of random paths linking x_0 and x. The result would be exact in the limit where the number of paths tends to infinity. Of course, it would be far too time consuming to evaluate this by computing all the paths, thus we have to rely on a method that ensures that the selected sequence of random paths preserves the probabilistic properties of the full sample of paths. More details are given below.

Once it is possible to compute the propagator, G, it is then straightforward to compute the ground-state probability distribution. This is given by the large τ limit of

$$|\Phi(x)|^2 = \frac{1}{Z} G(x, -i\tau; x, 0) \tag{15.10}$$

with $Z = \int G(x, -i\tau; x, 0) dx$ playing a role analogous to the partition function in statistical mechanics.

In order to make the program run quickly, we use the metropolis algorithm to make a Monte Carlo selection of paths taking into account their weights (a good explanation is given in Lepage 2005). A simple version is as follows:

1. pick an initial random path, ie., a set of x_j.
2. perturb one of the x_j by a small random number
3. Calculate the change in the action $\epsilon \Delta \mathcal{E}$
4. We retain the new path if
   (1) $\Delta \mathcal{E} < 0$ or
   (2) if not, we draw a uniform random number r between 0 and 1 and if the random number is less than $\exp(-\epsilon \Delta \mathcal{E})$ we keep the new path.
   (3) If neither of these conditions is met, we leave x_j unchanged.
5. Move on to the next path point in the lattice.
6. repeat this for a very large number of iterations.

Every time all the points on the lattice have been looped through, in each case testing whether or not a shift should be retained according to the rules outlined above, is called a thermalisation sweep. When a set number of thermalisation sweeps have been executed, properties of the lattice can by averaging over a random selection of paths. For example, you could store every 100th path generated by your program. Because of the way in which the set of paths has been generated, there is no need to weight the paths when you average.

15.4 Work Plan

1. **Design your program.** The first task is to design your program. Use pseudocode techniques to design a program that you will use to investigate Feynman path integrals. For the milestone problem, you should design your program so that it can generate a set of random paths, and then average these path, weighting each by $\exp(-\epsilon\mathcal{E}(x_j))$, to obtain quantities such as the ground-state wave function. Later, you can greatly improve the efficiency of your code using the Metropolis algorithm so that it picks paths with greater weight more frequently.

   As well as an outline of the code structure, you should include: (a) the names and descriptions of the functions you will use: what variables will they take as arguments and what will they return? (b) what data you will input to the program, how the program output will the results? (c) what data structures the program will use to store its internal data?

2. **Solve the Milestone Problem.** For the milestone problem, we will compute the ground-state wavefunction of a simple quantum mechanical system by direct evaluation of a number of random paths. Consider a lattice with $N = 7$ spatial points, spread over a complex time interval $T = 4$, with "time" step $\epsilon = T/N$.

   It is extremely useful to check this program by using a potential for which we know the analytic solution. A good starting point is the harmonic oscillator.

   $$V(x) = \frac{1}{2}x^2 \tag{15.11}$$

   Apply your program to a particle of mass, $m = 1$, and use units such that $\hbar = 1$. In these units, the classical trajectory has period 2π; we will consider the time interval $T = 3$. You will need to evaluate the summed energy along each path as in Eq. 15.9 and use this to approximate the multidimensional integral Eq. 15.8.

   You should then test your program against the result from the conventional approach to quantum mechanics to confirm that you recover the ground-state wave function

   $$|\phi(x)|^2 = \left(\frac{e^{-x^2/2}}{\pi^{1/4}}\right)^2 \tag{15.12}$$

   Note that the simple method described above will not fit exactly unless you increase the number of points in your integration by a large factor. You could investigate use of the python Vegas module to speed up integration.

   For the milestone your program must plot $|\phi(x)|^2$ in the range $x = 0$ to 2 (evaluated using the Schrodinger and path integral approaches) as a function of x and determine its value at $x = 1$.

3. **Extend your program** Now you have a working program that evaluates the propagator by direct integration of the path integral. This will be too slow/inaccurate to use in your research and you should extend the program to use the Metropolis algorithm. In this way, you can simply average quantities over the retained paths. A good starting point is to confirm that you can reproduce the ground-state wave function of the harmonic oscillator.

4. **Research with your Program** Now you have a working code, you can investigate Feynman path integrals in more detail. You should not look at all the points below, they are intended as suggestions for the starting point for your investigation. It is more important to investigate one point in depth than to cover a wide range of phenomena.

    - A particularly nice feature of the path integral approach is that you can investigate the transition from quantum mechanical behaviour as a function of the Planck constant (or equivalently the physical dimensions of the system). Compare the results you get from classical mechanics, conventional quantum mechanics and the path integral approach. You will find this discussed in Sakurai's book.
    - There are several tricks that we've use to derive the lattice solution to the path integral problem: the transformation to complex time, the apparent assumption in the Eq. 15.9 that the path is smooth. These and other interesting issues are discussed in Feynman & Hibbs for example. *Remember, however, that this is a computing project! You will need to compare analytic theory against the numerical method.*
    - Although the Metropolis method is simple, it is not particularly efficient or robust. Investigate how the accuracy of your results can be improved. The statistical uncertainties can be quantified by looking at the variation in accepted paths. Is it better to run a single Markov chain or to have multiple chains? More sophistical MCMC integration methods are discussed extensively in the literature. These issues are discussed in Lepage's article.
    - The approach we have followed allows us to extract the ground-state wave function. You can easily modify your program to investigate other potentials. Lepage 2005 discusses how you can investigate excited states. Indeed, once you have determined a set of paths from the Metropolis algorithm, you can compute expectation values for many other quantities.
    - The path integral approach can be applied to systems that are much more complex than the harmonic oscillator that we used in the Milestone problem. Investigate the use of more complicated Potentials (and Lagrangian). This flexibility makes Feynman path integrals the only practical approach to problems in QCD, the quantum mechanical description of the strong force. A simplified introduction is provided in section 3 of Lepage 2005, a more advanced text is Gupta 1998.

References

Feynman R. & Hibbs A. R., 1965 Quantum Mechanics and Path Integrals. (Use the emended edition from 2010)

Sakurai J. J. & Napolitano J., 2012, Modern Quantum Physics, Pearson

Creutz M., 1985, Quarks, Gluons and Lattices, Cambridge Univ. Press.

Lepage, P. G., 2005 'Lattice QCD for Novices'. http://arxiv.org/pdf/hep-lat/0506036v1.pdf.

Gupta R., 1998, Lecture notes from the Les Houche summer http://arxiv.org/abs/hep-lat/9807028. This is an advanced text.

Part V

Climate Physics

Chapter 16

Global heating and the spectra of greenhouse gases

16.1 Introduction

The aim of this project is to understand the spectral response of the Earth to both incoming and outgoing electromagnetic radiation. The energy balance of the Earth will be modelled using **Planck's radiation law** combined with **molecular spectra**. For a Sun temperature, $T_\odot = 5770$ K, a no atmosphere model predicts that the Earth's temperature is $T_\oplus \approx 255$ K, depending on the **albedo**. Adding an atmosphere raises the surface temperature to a new equilibrium. The goal of this project is to calculate this temperature—as far as possible from first principles.

16.2 Background theory

16.2.1 Planck's radiation law

The spectral radiance of a thermal emitter at temperature T is

$$B_\nu = \frac{2h\nu^3}{c^2}\frac{1}{\mathrm{e}^{h\nu/k_\mathrm{B}T}-1}\ , \tag{16.1}$$

where ν is the frequency. The factor of 2 is for two polarizations. The spectral radiance is the power per unit area per unit solid angle per unit frequency. The SI units are $\mathrm{Wm^{-2}sr^{-1}Hz^{-1}}$. To obtain the Sun's irradiance at the Earth we integrate over the solid angle of the Sun and weight the contributions using **Lambert's cosine law**:

$$I_{\nu,\oplus} = \left(\frac{R_\odot}{D}\right)^2 \int_0^{2\pi}\int_0^{\pi} \cos\theta\sin\theta \mathrm{d}\theta\mathrm{d}\phi\ B_{\nu,\odot}\ ,$$

where D is the Earth-Sun distance. Using $\int_0^{2\pi}\int_0^{\pi}\cos\theta\sin\theta\mathrm{d}\theta\mathrm{d}\phi = \pi$ we get

$$I_{\nu,\oplus} = \pi\left(\frac{R_\odot}{D}\right)^2\ B_{\nu,\odot}\ .$$

The irradiance has units of $\mathrm{Wm^{-2}Hz^{-1}}$. If we integrate over frequency we should get the **Solar constant** 1361 $\mathrm{Wm^{-2}}$. The incoming and outgoing radiation must be equal, i.e.

$$\int_0^\infty \mathrm{d}\nu I_{\nu,\oplus} = \alpha\ 4\pi \int_0^\infty \mathrm{d}\nu B_{\nu,\oplus}\ ,$$

where $B_{\nu,\oplus}$ is the spectral radiance of the Earth at $T_\oplus$, and $\alpha = 0.296 \pm 0.002$ is the average terrestial albedo. This equality determines Earth's surface temperature, $T_\oplus$, in a no atmosphere model.

When we add an atmosphere, greenhouse gases absorb a fraction of the outgoing radiation in particular spectral regions. The most important greenhouse gases include CO_2, CH_4, N_2O and CFCs. Other gases such as H_2O, O_3, H_2, and SO_2 also have resonances in the spectral regions of interest.

16.2.2 Basic molecular physics of greenhouse gases

All molecules interact with EM fields via **electronic, vibrational and rotational resonances**. Typically these resonance are in the frequency range 100s to 1000s THz, 1 to 100 THz and < 1 THz, respectively. Given the spectrum of the incoming and outgoing radiation, only the electronic and vibrational resonances play an important role in the greenhouse effect.

To understand details, we shall focus on the most prominant example, CO_2. The main electronic resonance from the ground state to the first excited state has a frequency of 1400 THz (corresponding to a wavelength of 220 nm) which is outside the range of both incoming and outgoing radition. CO_2 has three vibration modes, bend (around 20 THz), symmetric stretch (40 THz) and asymmetric stretch (70 THz), but due to symmetry, the symmetric stretch does couple strongly to EM fields. The **bending mode** resonance is close to the peak of the outgoing spectrum. The resonances associated with this mode have a rich structure involving individual transitions that may change either the vibrational or rotational excitation. To understand and be able to perform calculation based on the CO_2 spectra, we shall download spectral data from the HITRAN database [1].

16.2.3 How to obtain spectral data from HITRAN

Go to HITRAN line-by-line data https://hitran.org/lbl/https://hitran.org/lbl/ and follows the following steps (to download the data you will need to register):

1. Select ID2, i.e., CO_2
2. Now you get all the isotopes of CO_2 but as over 98% is $^{12}C^{16}O_2$ we can unselect all and reselect only the most abundance one, ID1.
3. Next, we get to select wavelength range. To reduce the amount of data, we can focus on a particular region of interest that overlaps with important bending mode. Enter ν_{min}, and ν_{max} equal to 570 $\mathrm{cm^{-1}}$ and 770 $\mathrm{cm^{-1}}$, respectively.
4. Select ouptut format. Click on Create New Output format. From the Available parameters list on the right, Select ν, and S. This gives us the frequencies and strength of each resonance. We also want to add γ_{air} as this tells us about the **pressure broadening** broadening of the line as a function of the air pressure. **Question**: Why don't we need γ_{self}?
5. There are a few other parameters to set:

- Field separator: [comma]
- Line Endings: Unix / Linux / Mac OS X (LF)
- Header line (not necessary)

6. Now you can download the data and save it as say 'CO2Min.csv'.

7. To test your download, write a small python code and use e.g. pandas to view.

```
import pandas as pd
HITRAN_data = pd.read_csv('./CO2Min.csv', usecols=[0,1,2],header=0)
print(HITRAN_data.head(6))
```

   It should display the first six rows.

8. The parameters and units are explained here https://hitran.org/docs/definitions-and-units/

16.2.4 Atmospheric transmission

Absorption removes a fraction of incident radition equal to the effective area or corss-section of the absorbers. We can write this as the intensity at a particular frequency ν and altitude $z + \mathrm{d}z$ is given by

$$\mathcal{I}(\nu, z + \mathrm{d}z) = [1 - N(z)\sigma(\nu, z)\mathrm{d}z]\, \mathcal{I}(\nu, z) \ , \tag{16.2}$$

where $N(z)$ is the number density of absorbers at a function of altitude, z, and $\sigma(\nu, z)$ is the **absorption cross-section**. We have written this a function of frequency, ν and altitude z but will (at least initially) make the approximation that it is independent of altitude. If both the number density and absorption cross section are independent of the height, z, then integration gives

$$\mathcal{I} = \mathcal{I}_0 \mathrm{e}^{-N\sigma z} \ . \tag{16.3}$$

This is known as the **Bougier-Lambert-Beer law** and describe a simple exponential decay of the outgoing intensity. The quantity $\mathrm{OD} = N\sigma z$ is known as the **optical depth**. For the bending mode resonance of the CO_2 molecule at a density of a few hundred ppm the optical depth is of order 10, i.e. the atmosphere is *optically thick* in this frequency range.

16.2.5 Absorption cross-section

The absorption cross-section contains all the information about the resonance frequencies and the lineshape. For each resonance, pressure broadening give us a **Lorentzian lineshape** of the form, see eqn (6) in Ref. [2],

$$\sigma(\nu, z) = \frac{S}{\pi} \frac{\Gamma(z)}{1 + (\nu - \nu_0)^2/\Gamma(z)^2} \ , \tag{16.4}$$

where $\Gamma(z) = N(z)\gamma_{\mathrm{air}}$ is the linewith and ν_0 is the resonance frequency. In practice, ν_0 may also depend on the altitude z due to pressure shifts, but we are going to neglect those. This Lorentzian is repeated for each row in our Hitran data file. We shall plot the sum of all line in Part I of the Milestone project. To go from the cross section data to transmission through the atmosphere, we need to insert the cross section into the BLB law, eqn (16.3). This results in a plot something like Fig. 16.1. As the spectrum is relatively complex, Wilson and Gea Banacloche [2] approximate the many-line spectrum with a single-effective-cross-section. You might want to do this first to get your code working before trying the full spectrum.

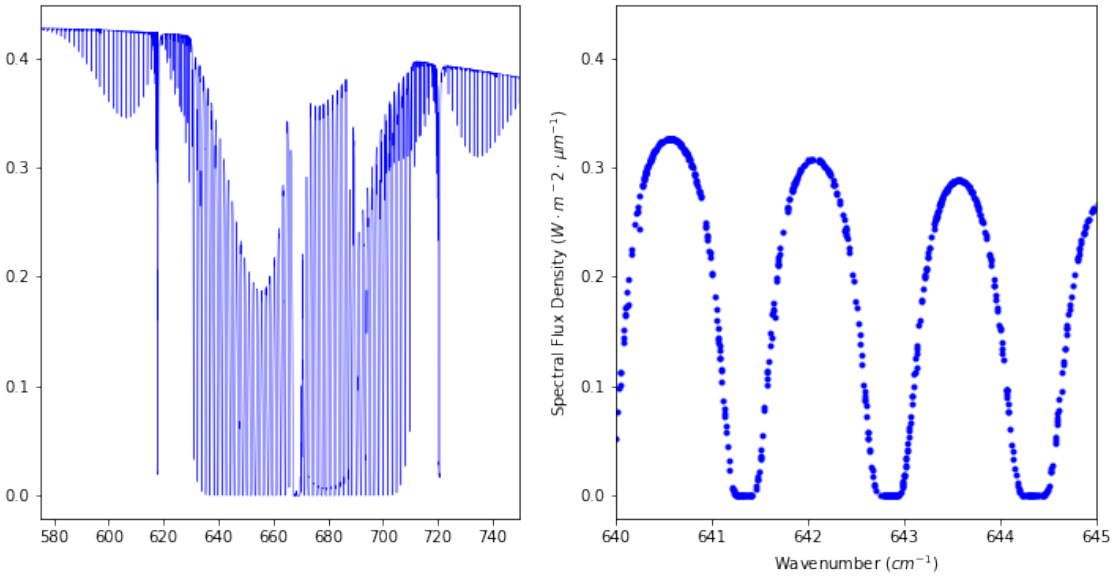


Figure 16.1: Transmission through the atmosphere in the region of the CO_2 bending mode. If we zoom in (right) we can see the Lorentzian lineshape of the individual resonances.

16.2.6 Radiation transfer model

The next question is how the absorption of outgoing radiation by greenhouse gases affects the temperature at the surface of the Earth. The effect is quite subtle. Basically, the atmosphere is optically thick in the region of the CO_2 bending mode, so radiation in this frequency range cannot escape until we reach the stratosphere. As the stratosphere is cooler, the trapped energy that is re-radiated at high altitude is less than it would have been, and therefore to restore the energy balance, we need to increase the surface temperature. When we calculate the spectrum for the milestone, we shall see that effectively a notch is removed from the blackbody spectrum, see e.g. Fig. 16.2, and to restore the energy balance (the area under the curve) we need to increase the surface temperature—that is global heating!

To model this effect, we need to say something about how radiation propagates upwards through the atmosphere. This is known as **radiation transfer**. The rate of change of radiation propagating upwards at a frequency ν is given by the **Schwarzschild equation** or radiation transfer equation,

$$\frac{\mathrm{d}I(\nu,z)^+}{\mathrm{d}z} = -N\sigma(\nu)\left\{I(\nu,z)^+ - B\left[\nu,T(z)\right]\right\} , \tag{16.5}$$

where $B\left[\nu,T(z)\right]$ is spectral radiance given by Planck's formula, eqn (16.1). The first term in this equation just says at each altitude z we remove some energy in particular spectral regions due to absorption. The second term says that this energy is redistributed to the surrounding gas as heat, and then re-radiated as blackbody radiation at the local temperature for that altitude. We could either solve this using numerical integration, or we can make an approximation, see eqn (22) in Section V of Ref. [2]. The approximate predicts that the outgoing radiation at the 'top' of the atmosphere is

$$I(\nu)^+ = B(\nu,T_\mathrm{s})\mathrm{e}^{-\mathrm{OD_s}(\nu)} + B(\nu,T_\mathrm{trop})\left(1-\mathrm{e}^{-\mathrm{OD_s}(\nu)}\right) , \tag{16.6}$$

where $\mathrm{OD_s}(\nu) = N_0\sigma(\nu)z_0$ is the optical depth from the surface to outer space, N_0 is the number density of greenhouse gas molecules, $z_0 = M_{\mathrm{CO2}}g/k_{\mathrm{B}}\mathrm{T}_s$ is their effective column height assuming an isothermal troposphere, T_{s} and T_{trop} are temperatures at the surface and top of the troposphere, respectively. For T_{trop} we follow Ref. [2] and use

$$T_{\mathrm{trop}} = T_{\mathrm{s}} - \Gamma_{\mathrm{LR}} z_0 \log(1-\eta)$$

where $\Gamma_{\mathrm{LR}} = 0.00649$ Km^{-1} and $\eta = 0.75$.

16.3 Work plan

16.3.1 Design of your program

The milestone has a number of sub-components. You will need code to model the following:

1. A function to generate the Planck spectrum for any temperature.
2. A code to plot the CO_2 spectra and explore an effective lineshape like the one used in Ref. [2].
3. A code to plot the solution of the **Schwarzschild equation**, see Ref. [2]. We shall use the 'single-slab isothermal troposphere' model, eqn (22) in Section V of Ref. [2]. Similar models can also be found in Chapter 4 of Ref. [3].

These three components can be stand-alone codes or all integrated into a single notebook.

16.3.2 Solve the milestone problem

The milestone will help us build the basic structure of our energy balance model that can be applied to a range of problems outlined in the Extension. The theoretical basis for your milestone code is outlined in Ref. [2]. We shall split the milestone into three deliverables:

1. **Energy balance without CO_2**: This involves the following steps.
   - Plot the incoming and outgoing radiation using the Planck spectrum on the same graph. Make a 4-panel plot, where columns one and two are plotted versus frequency and wavelength, respectively, and rows one and two use linear and log-log scaling, respectively. This is a useful exercise to check there all your parameters and units are correct.
   - Download the E490 Solar reference spectrum
     https : //www.nrel.gov/grid/solar − resource/spectra − astm − e490.html
     and add this to your plots. If you units are correct then there will be approximate agreement between your equations and the experimental data.
   - Sum the area under the curve for the incoming Planck prediction, incoming E490 data, and outgoing Planck prediction. Quote the values in your figure capture.
   - Set the albedo parameter, then adjust the Earth's temperature to achieve an energy balance. What is the predicted temperature?
2. **Plotting and approximating the CO_2 absorption spectrum**.

- Now use the HITRAN CO_2 data to make a plot of the CO_2 spectrum. **Questions**: Can you explain the main features in the spectra? What are the P, Q and R branch? What is the effect of temperature on the evelope of the spectra? Why is the Q branch asymmetric?
- Plot both a stick spectrum and a high resolution plot showing the Lorentzian lineshapes.
- Fit a function to approximate the spectrum, similar to eqn (8) in Ref. [2]. **Question**: How does this plot change with altitude?

3. **Energy balance with CO_2**:

- Following Section V of Ref. [2] we solve the Schwarzschold equation, eqn (16.5), using the 'single-slab isothermal troposphere' approximation, eqn (16.6). This allows us to replot the outgoing irradiance with any desired partial pressure of CO_2. If you use an approximate fit to the CO_2 spectrum you should get a plot like the black line in Fig. 3 of Ref. [2]. If you include the full spectrum from HITRAN your plot should look a bit like Fig. 16.2. Make a plot for say 350 and 450 ppm CO_2. **Question**: In what regions of the spectrum are the effects of increasing CO_2 most important?

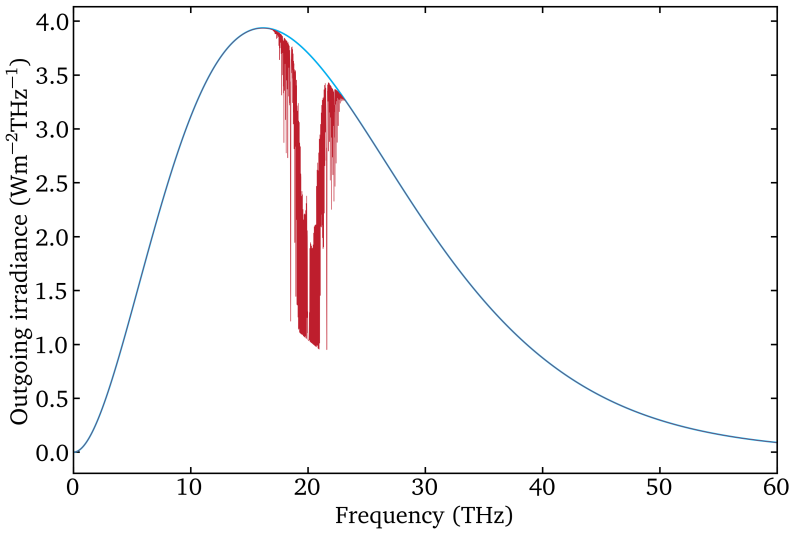


Figure 16.2: Calculated using 2-layer model and full CO_2 spectrum.

- Using the CO_2 modified spectrum, repeat our energy balance calculation. What is the expected surface temperature now?
- **Questions**: The output using a more sophisticated online simulator is shown Fig. 16.3. How do your results compare? Can you explain the spike in the centre of the CO_2 absorption feature in their model?

16.3.3 Extensions

Possible extensions might focus on a more detailed model of the role of CO_2 or applying related ideas to other molecules. Or other topics that build on our spectral model are encouraged.

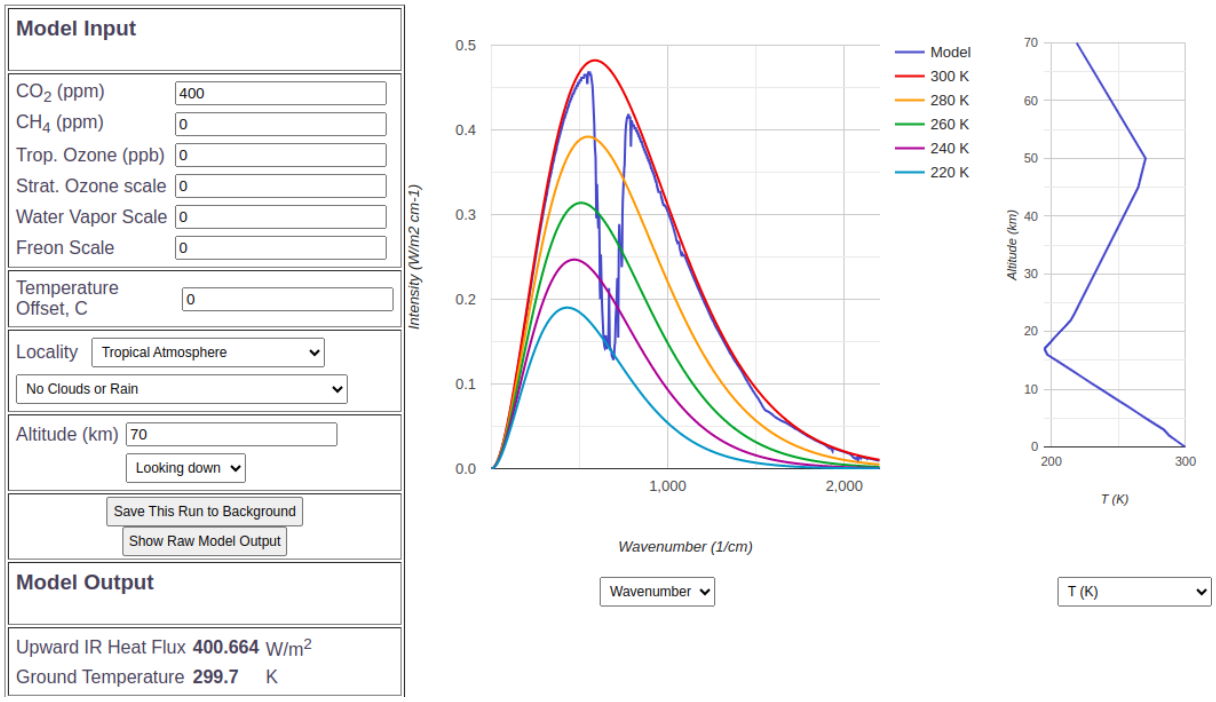


Figure 16.3: Outgoing radiation prediction for 400 ppm CO_2, from https://climatemodels.uchicago.edu/modtran/.

- CO_2: Possible directions might include extending the model by replacing the approximate spectrum with the actual spectrum, and including a more exact model of the tropopause and stratosphere. How does your calculated spectra compare with data, e.g. [4]? Can you reproduce the near linear relationship between surface temperature and total CO_2 emissions that we see in the IPCC reports.

- CH_4: it is often argued that methane [5] is two orders of magnitude more potent than CO_2. Can we understand this factor from the CH_4 absorption spectrum?

- H_2O: Water is not classed as a greenhouse gas but does play a role. Investigate the effect of water on the emission spectrum and attempt to build a simple model to illustrate that despite absorbing the infra-red there is no overall effect on surface temperature [6].

- SO_2 [7]: Stratospheric injection of SO_2 has been proposed as a way to geo-engineer a desired climate. However, the energy balance for this process is complicated by chemistry—SO_2 first reacts with O_2 to produce SO_3 and then with water to produce to form H_2SO_4. Is there an overall cooling or heating effect related to the spectra of these constituents?

- Other gases that could be explored. It has been argued that H_2 is also a greenhouse gas [8]. What does the spectrum tell us? We know that ozone has absorptions that affect both the incoming and outgoing spectrum. Use spectral data to assess its overall role?

References

1. *HITRAN online Line-by-Line Search;* https://hitran.org/lbl/

2. *Simple model to estimate the contribution of atmospheric CO2 to the Earth's greenhouse effect, DJ Wilson and J Gea-Banacloche, Am. J. Phys.* **80**, *306 (2012);*
https://aip.scitation.org/doi/10.1063/1.3624527

3. *Principles of Planetary Climate, RT Pierrehumbert (CUP Cambridge 2015);*
https://www-cambridge-org.ezphost.dur.ac.uk/highereducation/books/principles-of-planeta

4. *Using IASI to simulate the total spectrum of outgoing long-wave radiances , EC Turner, H-T Lee, and SFB Tett, Atmos. Chem. Phys.* **15**, *6561 (2015).*
https://acp.copernicus.org/articles/15/6561/2015/

5. *Advancing Scientific Understanding of the Global Methane Budget in Support of the Paris Agreement, AL Ganesan et al, Global Biogeochemical Cycles* **33**, *1475 (2019);*
https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GB006065

6. *Earth's outgoing longwave radiation linear due to H_2O greenhouse effect, DDB Koll and TW Cronin, PNAS,* **115**, *10293 (2018);*
https://www.pnas.org/doi/full/10.1073/pnas.1809868115

7. *Anthropogenic sulfur dioxide emissions: 1850–2005, SJ Smith et al, Atmos. Chem. Phys.* **11**, *1101 (2011);*
https://acp.copernicus.org/articles/11/1101/2011/

8. *Climate consequences of hydrogen emissions, IB Ocko and SP Hamburg, Atmos. Chem. Phys.* **22**, *9349 (2022);*
https://doi.org/10.5194/acp-22-9349-2022