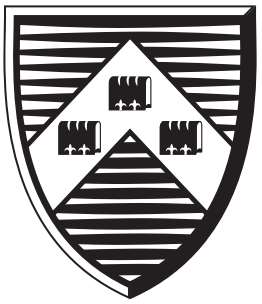

University of York

Masters Dissertation

Spin chains, hyper-fine spin interactions and magnetic spin coupling in quantum dots

Author:
John Children

Supervisor:
Dr. Irene D'Amico

A dissertation submitted in fulfilment of the requirements for the degree of Master of Physics

in the

Semiconductor Spintronics and Quantum Information group
Department of Physics

May 2013

Declaration of Authorship

I certify that any substantive contributions from the work of others are clearly identified and attributed, and that with these exceptions, this work is wholly my own. I also certify that to the best of my knowledge my work has not been used in the assessed work of others without attribution.

I confirm that I have read and understood the information concerning academic misconduct in the Undergraduate Handbook (Section 4).

Signed: _____

Name (Print): _____

Date: _____

Time (if past deadline): _____

"If you think you understand quantum mechanics, you don't understand quantum mechanics."

commonly attributed to Richard Feynman

UNIVERSITY OF YORK

Abstract

Faculty Name
Department of Physics

Master of Physics

Spin chains, hyper-fine spin interactions and magnetic spin coupling in quantum dots

by John Children

We investiage the properties of simple spin chains, the fermi contact hyperfine interaction and the comparative strength of external magnetic fields using a program built from theory. We find that there is an oscillation of probabilities of states due to spin interactions, which is found in both the spin chain and hyperfine model. When the magnetic field is introduced it is found that a probability density shift occurs. The timescale of interaction is verified as being related to the interaction constant for both hyperfine interaction and magnetic fields.

Acknowledgements

I would like to thank my project supervisor Dr D'Amico for perservering through my constant misunderstandings of quantum theory.

Contents

List of Figures

List of Tables

Abbreviations

TDSE **T**ime **D**ependant **S**chrodinger **E**quation

Physical Constants

Elementary Charge	e	=	$1.602\ 176\ 57 \times 10^{-19}\mathrm{C}$
Reduced Planck Constant	$\hbar$	=	$1.054\ 571\ 73 \times 10^{-34}\mathrm{J} \cdot \mathrm{s}$
Bohr Magneton	μ_B	=	$9.274\ 009\ 68 \times 10^{-24}\mathrm{J} \cdot \mathrm{T}^{-1}$
Nuclear Magneton	μ_N	=	$5.050\ 783\ 24 \times 10^{-27}\mathrm{J} \cdot \mathrm{T}^{-1}$
Electron g-factor	g	=	$-2.002\ 319\ 304\ 362\ 2$ (Dimensionless)

Symbols

C	Basis co-efficient	
D	Basis co-efficient of eigenvectors	
S	Spin	
t	Time	s
$\hat{H}$	Hamiltonian	J
J	Coupling Constant	J
A	Hyperfine Constant	J
B	Magnetic Field	T

ψ	Wavefunction	Probability Amplitude
ϕ	Basis vector	
φ	Eigenvector	
ε	Eigenvalue	
$\hat{\sigma}_x$	Pauli spin matrix x	
$\hat{\sigma}_y$	Pauli spin matrix y	
$\hat{\sigma}_z$	Pauli spin matrix z	
$\mathbb{1}$	2x2 Identity matrix z	

For my Father...

Chapter 1

Introduction

1.1 Quantum Information

Similarly to the way in which information is represented as a bit in classical computing, quantum information can be represented as a quantum bit or qubit [1]. The interesting and useful property of this qubit is that it can be utilised to represent a number of different states due to the superposition behaviour found in quantum mechanics. For example, if we have two quantum states $|0\rangle$ and $|1\rangle$ these states can be used to represent information and also form a superposition [**?**].

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1.1}$$

Where α and β are complex co-efficents of the basis states, which cannot be measured directly. However, we can measure the probability associated with the system being in one of the states $|0\rangle$ and $|1\rangle$ through the modulus squared of our co-efficients. The sum of these co-efficients $|\alpha|^2 + |\beta|^2$ must therefore be equal to one due to the nature of probability, which forms the normalisation condition. This superposition means that information can be represented as a vector in Hilbert space on a Bloch sphere as shown in figure (1.1).

Additionally, if we assemble several qubits consisting of similar $|0\rangle$ and $|1\rangle$ representations of state we can form new states from kronecker products because of this vector representation. Thus a state of two qubits will consist of a superposition four states with four complex co-effecients associated with each state. This

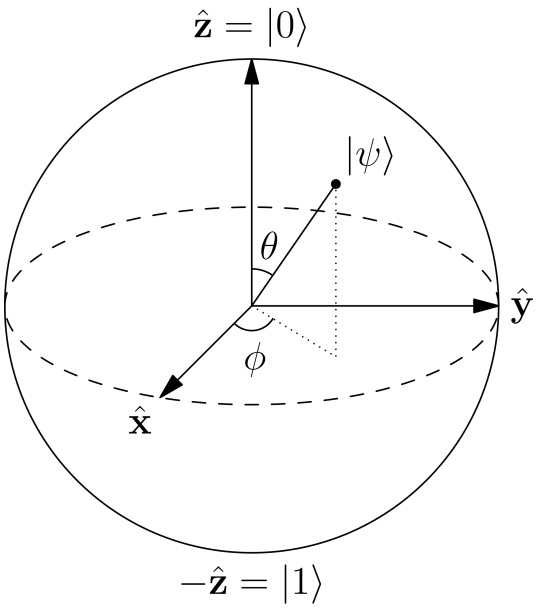

FIGURE 1.1: Bloch sphere representation of a quantum bit [2]

property means that for N qubits there will be 2^N possible complex values which can be used to represent information, which is a very desirable advantage over classical information [3].

However, chosing a system that can both represent and utilise these qubits is a difficult task which will be examined in the next subsection.

1.2 Physical Representation of Qubits

Quantum bits can be represented through a wide range of quantum systems systems ranging from ion-traps [4], optical implementations [5], nuclear magnetic resonance [6], all of which have their own individual strengths and weakness.

One possible implementation however has a great deal of promise in the form of the spin of an electron trapped in a semiconductor quantum dot as such a technology could utilise the advancements made in solid state physics that have greatly contributed to the current digital era [7].

1.3 Quantum Dots

A quantum dot is a portion of matter which can confine a charge quanta within a three dimensional box through electrostatic potentials [1]. Typically this kind of device is implemented with a semiconductor material which confines an electron into a particular area of space. The spin state or the energy level of the electron can then be used to form a qubit using two or more states and then be accessed through a range of methods [8]. For this project, the spin state of the electron was chosen as the basis for the qubits, as there are only two possible states due to the properties of the particle, and the semiconductor material chosen was silicon-29 as the nuclei also only have half-spin states.

This system was chosen because there is great potential for computational use of qubits in this fahsion, with a great gate system [9], and such a system has been shown to be initialisable into a state using ultra-fast optical pulses [10], fufilling a requirement of the DiVincenzo checklist [11].

1.4 Decoherence

Currently, the primary problem with all implementations of quantum information is the decoherence problem [7], which causes the information in a system to be obscured by interaction with its enviroment. Thus, decoherence forms part of the DiVincenzo checklist for a physical implementation of a quantum computer and as such is an area of high interest in quantum information research [11].

Because of this, a consideration of this project was the sources of decoherence for the electron qubit system as it was hoped that modeling the interactions responsible would lead to further insight and help identify ways to minimise this problem. For the quantum dot electron spin qubit, the primary source of decoherence will be the spin interaction between the electron and the nuclei due to fermi contact [12–14], thus this will be one of the primary focuses of the simulation.

Another possible interaction which could affect decoherence of the system is the coupling of spins with an external magnetic field, thus this interaction will also be considered.

Chapter 2

Methods

2.1 Simple Spin Chain

The first part of the project concerns a simple spin chain. This was used to investigate the basics of spin interaction in order to progress further. This would frame the way in which the interactions would occur in the later methods, investigate how the system's evolution in time could be modeled and the error associated with this modeling.

2.1.1 Concept

The simple spin chain consists of a number of identical theoretical spins, N, which are able to interact with only their nearest neighbour. The spins can have two possible states, up and down and each interaction is governed by a constant J, which is the specific exchange interaction co-efficient.

In order to find the state of the system at a later time, the time dependent schrodinger equation **TDSE** is employed [15].

$$i\hbar\frac{\partial |\psi\rangle}{\partial t} = \hat{H} |\psi\rangle \tag{2.1}$$

The hamiltonian for this system is:

$$\hat{H} = \sum_{i,j=1}^{N} J_{ij} S_i \cdot S_j \tag{2.2}$$

This hamiltonian is time-independant and so will be relatively easy to solve. In the case of the simple spin chain, spins can only interact with their neighbour. Hence:

$$\begin{aligned} S_k \cdot S_{k+1} &= 1 \qquad & k &= \{\, 1 \dots N-1 \,\} \\ S_k \cdot S_{k-1} &= 1 \qquad & k &= \{\, 2 \dots N \,\} \end{aligned} \tag{2.3}$$

And all other products are zero

2.1.2 Basis and Initial Conditions

The wavefunction of the system at a specific time can be expressed using a basis representing the possible states of the system ϕ_i and an associated complex coeffcient C_i representing the probability amplitude of each basis state.

$$|\psi\rangle = \sum_{i=1}^{N} C_i(t) \, |\phi_i\rangle \tag{2.4}$$

Choosing the basis for the system will determine the matrix formulation of the hamiltonian. For this simple spin chain a similarly simple basis was chosen consisting of N possible states, each with one spin up state and $N-1$ spin down states. This was chosen so that each basis can be represented in a vector form consisting of one 1 value and $N-1$ 0 values.

The initial conditions for the system can then be chosen through setting the initial values of $C(t)$ at t = 0, such that the system can be initialised in a particular state or a superposition of states.

For $N = 2$ the basis set consists of:

$$|01\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \, |10\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{2.5}$$

Thus the hamiltonian for N = 2 is:

$$\hat{H} = \begin{pmatrix} 0 & J_{21} \\ J_{12} & 0 \end{pmatrix} \tag{2.6}$$

And the basis set for N = 3 is:

$$|001\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}; |010\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}; |100\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{2.7}$$

and the hamiltonian for N = 3 is:

$$\hat{H} = \begin{pmatrix} 0 & J_{21} & 0 \\ J_{12} & 0 & J_{32} \\ 0 & J_{23} & 0 \end{pmatrix} \tag{2.8}$$

and so on for larger values of N.

For the simple spin chain, every value of J was set to be equal such that each interaction was of equal magnitude for simplicity. This value was set as 0.04 in the program as this value produced aesthetically pleasing results.

Additionally, a single value of J unique to each coupling was tested, as it was advised that this value particular would result in periodic oscillations in probability by the project supervisor. This value was given by:

$$J_{i,i+1} = J_0 \times \sqrt{i(N-i)} \tag{2.9}$$

Where i refers to an index in the hamiltonian similar to those displayed earlier and and J_0 is a constant.

2.1.3 Iterative integration of TDSE

In order to solve the Time Dependant Schrodinger Equation with our basis

$$i\hbar \sum_{i=1}^{N} \frac{\partial C_i(t)}{\partial t} |\phi_i\rangle = \sum_{i=1}^{N} C_i(t) \hat{H} |\phi_i\rangle \tag{2.10}$$

We multiply through by another basis state $bra\phi_j$

$$i\hbar \sum_{i=1}^{N} \frac{\partial C_i(t)}{\partial t} \langle \phi_j | \phi_i \rangle = \sum_{i=1}^{N} C_i(t) \langle \phi_j | \hat{H} | \phi_i \rangle \tag{2.11}$$

But $\langle \phi_j | \phi_i \rangle = \delta_{ij}$, which is the kronecker delta function, for which:

$$\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j, \end{cases} \tag{2.12}$$

Therefore we arrive at:

$$\hbar \frac{\partial C_j(t)}{\partial t} = \sum_{i=1}^{N} C_i(t) \hat{H}_{ij} \tag{2.13}$$

We can then employ a simple numeric approximation of partial differential

$$\frac{\partial C_i(t)}{\partial t} \approx \frac{C_j(t + \Delta t) - C_j}{\Delta t} \tag{2.14}$$

Thus we find our approximate equation of motion for the state j:

$$C_j(t + \Delta t) = C_j(t) + \frac{\Delta t}{i\hbar} \sum_{i=1}^{N} C_i(t) \hat{H}_{ij} \tag{2.15}$$

By iterating this equation over time, we can find the associated probability amplitudes C_j of the basis states.

2.1.4 Measuring Error

Due to the iterative nature of this method, there will be some error after a number of timesteps. This error is dependant both on the size of Δt and the machine precision being used in iteration. Therefore in order to minimise error a small value of Δt must be chosen as well as a good level of precision. There are some limits on Δt, which must be large enough for iterations of the equation of motion and so

In the program, $\frac{\Delta t}{i\hbar}$ was saved as a single variable which was set to be equal to $2\times10^{-2}i$ for the simple spin chain. The variable precision was set to IEEE floating double with 15 digits of precision in the mantissa and an exponent between the ranges of 300 and -300. As $J \times \frac{\Delta t}{\hbar} > 10^{-15}$, there should be no problems here and the program should iterate correctly.

The resultant error can be measured with $\sum_{i=1}^{N} |C_i|^2$ as for the normalised wave-function this value should be equal to 1 at all times. Any deviation from this value will be a result of error from these two sources and hence the effect of error on the results of the program can be measured by recording this value. As such this value was recorded to a file at each time step so that the error at each time step can be measured.

2.2 Hyperfine interaction

The next stage after the simple spin chain was to consider a basic physical system consisting of an electron and several nuclei. For physical data, silicon-29 was selected as it has a half spin, similar to the electron. Data for the physical constants in this interaction was taken from Klauser 's PhD Thesis [13].

2.2.1 Basis and Initial Conditions

For this case, the basis was defined using the individual spin state of each particle in the system. For a single qubit this was defined as:

$$|\uparrow\rangle = |0\rangle = \begin{pmatrix}1\\0\end{pmatrix}; |\downarrow\rangle = |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \tag{2.16}$$

For multiple qubits, the basis was defined through kronecker products of the qubit states. For example:

$$|\uparrow\downarrow\rangle = |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{2.17}$$

In order to make clear which spin in the ket is the electron, future states of this system will be written as $|\Uparrow;\uparrow\downarrow \ldots\rangle$, where the double arrow signifies the spin state of the electron. This basis will allow our hamiltonian to be composed of pauli spin matrices, which for this basis of spin 1/2 particles are defined as [**?**]:

$$\hat{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \hat{\sigma}_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}; \hat{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.18}$$

Additionally we will need to consider the two dimensional identity matrix $\mathbb{1}$ defined as:

$$\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.19}$$

The pauli spin matrices can be related to the spin vector of a particle in the solid state physics convention through

$$\vec{S} = \frac{1}{2}\vec{\sigma} = \frac{1}{2}(\sigma_x\vec{\imath} + \sigma_y\vec{\jmath} + \sigma_z\vec{k}) \tag{2.20}$$

This relation will allow the expression of spin in our hyperfine hamiltonian for our basis.

2.2.2 Hyperfine Hamiltonian

If we neglect the interactions of the nuclei with each other for simplicity, the hamiltonian of consists fermi contact of the electron with nuclear spins N in a periodic lattice defined as [13]:

$$\hat{H} = \vec{S} \cdot \vec{h}; \quad \vec{h} = \sum_{l=1}^{N} A_l \vec{I}_l; \quad A_l = A\nu_0 |F(\vec{r}_l)|^2 \tag{2.21}$$

Where the $\hat{S}$ is the electron spin state vector, $\vec{I_l}$ is the spin state vector of a particular nucleus l, ν_0 refers to the primitive lattice size, $|F(\vec{r_l})|^2$ to the modulus squared of the periodic Bloch function and hyperfine constant A is defined as:

$$A = \frac{4\mu_0}{3} g\mu_N \mu_B |u_0|^2 \tag{2.22}$$

Where $|u_0|^2$ is dependant on the nuclei and all other terms refer to physical constants defined in the list of physical constants.

Because the $\nu_0|F(\vec{r_l})|^2$ the term of A_l in (2.21) relates to the positioning of nuclei to the effective potential experienced by the nuclei, this can be simplified initially by assuming that all nuclei have the same magnitude of interaction with the electron. We can therefore assume that $A_l = A$ for all l. Our hamiltonian therefore becomes:

$$\hat{H} = \vec{S} \cdot \sum_{l=1}^{N} A\vec{I_l} \tag{2.23}$$

Using our relation for spin to the pauli matrices (2.20), we obtain a matrix representation for the hamiltonian in terms of our basis:

$$\hat{H} = \frac{1}{2}\vec{\sigma}_e \cdot \sum_{l=1}^{N} \frac{1}{2} A\vec{\sigma_l} = \frac{A}{4}(\vec{\sigma}_e \otimes \vec{\sigma}_1 \otimes^N \mathbb{1} + \mathbb{1} \otimes \vec{\sigma}_e \otimes \vec{\sigma}_2 \otimes^{N-1} \mathbb{1} + \ldots + \otimes^N \mathbb{1} \otimes \vec{\sigma}_e \otimes \vec{\sigma}_N) \tag{2.24}$$

For the case of N = 1

$$\hat{H} = \frac{A}{4}(\sigma_{xe} \otimes \sigma_{x1} + \sigma_{ye} \otimes \sigma_{y1} + \sigma_{ze} \otimes \sigma_{z1}) \tag{2.25}$$

$$\sigma_{xe} \otimes \sigma_{x1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{2.26}$$

$$\sigma_{ye} \otimes \sigma_{y1} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \tag{2.27}$$

$$\sigma_{ze} \otimes \sigma_{z1} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.28}$$

So:

$$\hat{H} = \frac{A}{4} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.29}$$

If we are to run the simulation for varying values of N, this procedure will need to be automated in the simulation depending on N.

2.2.3 Eigenvalues

As the pauli matrices are diagonalisable, so are their tensor products [1]. We can therefore find the eigenvalues of our hamiltonian through the diagonalisation of our matrix. This would be useful when deciding upon initial conditions for the system as if the initial conditions are equal to the eigenvalues there should be no change in the basis co-efficients with time, meaning that we should expect no change in basis state co-efficients upon iterating our equation with time.

2.2.4 Fidelity

One important measure of the effect of interactions with the time evolution of the system is through the use of fidelity. Fidelity is defined as:

$$Fidelity = |\langle initialstate|\psi\rangle|^2 \tag{2.30}$$

Fidelity is therefore a measure of the probability that a system will be found in it's initial state at a specific time, so by measuring fidelity in the simulation we will gain insight into how the system changes from its initial conditions.

2.2.5 Physical Parameters

Most of the parameters in the fermi contact hyperfine hamiltonian are physical constants, however the hyperfine constant itself is material dependant because of the $|u_0|^2$ term. Fortunately in [13] the value of A for silicon-29 is listed as $0.1\mu eV$ and this value was implemented into the program.

2.3 Magnetic Field

The next stage of the project concerned the interaction of the spins with an external magnetic field.

2.3.1 Magnetic Hamiltonian

In general the magnetic field coupling with a dipole has a hamiltonian of [15]:

$$\hat{H}_B = -\vec{\mu} \cdot \vec{B} \tag{2.31}$$

Where μ is the vector dipole moment of the particle and $\vec{B}$ is the magnetic field vector. This part will therefore have two componets, one with the electron coupling with the field H_{Be} and with the nuclei H_{BI}

$$\hat{H}_B = \hat{H}_{Be} + \hat{H}_{BI} = -\vec{\mu}_e \cdot \vec{B} - \sum_{i=1}^{N} \vec{\mu}_{Ii} \cdot \vec{B} \tag{2.32}$$

Considering the electron spin component and assuming $\vec{B}$ only in the z dimension we can use the z component of electron magnetic moment:

$$\mu_{ze} = -g_e \mu_B S_z \tag{2.33}$$

For which the $\vec{S}$ spin vector can be related to our pauli matrices as in (2.20). Hence for the electron:

$$\hat{H}_{Be} = \frac{1}{2} g_e \mu_B B_z \sigma_z \tag{2.34}$$

So the matrix form of the electron for our basis becomes:

$$\hat{H}_{Be} = \frac{1}{2} g_e \mu_B B_z \sigma_z \otimes^N \mathbb{1} \tag{2.35}$$

For N = 1

$$\hat{H}_{Be} = \frac{1}{2} g_e \mu_B B_z \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{2} g_e \mu_B B_z \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{2.36}$$

Similarly, considering the magnetic field coupling with the nuclei in the z dimension.

$$\mu_{zI} = -\mu_I S_z \tag{2.37}$$

So similarly for N = 1:

$$\hat{H}_{Be} = \frac{1}{2} \mu_I B_z \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \frac{1}{2} \mu_I B_z \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{2.38}$$

2.3.2 Physical Parameters

For the magnetic hamiltonian most parameters are universal physical constants, however the μI is material dependant. The value used for this parameter was the one given in Klauser's PhD thesis [13] for silicon-29, which is given as a ratio compared to the nuclear magneton and as such this value had to be converted to an energy unit if it was to be used in the program.

Due to the nature of the scale of the simulation it was decided that the physical parameters should also be small in the program to prevent numerical error and excessive use of mantissas. As such the energy of the hamiltonian generated by the program is all in micro-electronvolt units and the time outputted by the program is in picoseconds. As such all other physical parameters are similarly adjusted in order to be used by the program.

2.3.3 Timescale of interaction

An important aspect of checking the accuracy of the results obtained was through use of a measure of the timescale over which the interations were expected to take place. This timescale was approximated as:

$$\Delta\tau = \frac{\hbar}{J} \tag{2.39}$$

Where J is the generic coupling of one state to another. So for example for the hyperfine hamiltonian this value would be:

$$\Delta\tau_{hf} = \frac{4\hbar}{A} \tag{2.40}$$

and for the magnetic-nuclei and magnetic-electron couplings:

$$\Delta\tau_{BI} = \frac{2\hbar}{\mu_I B_z}; \Delta\tau_{Be} = \frac{2\hbar}{g_e \mu_B B_z} \tag{2.41}$$

Thus for our simulated silicon-29 system we can expect these time-scale values to be:

$$\Delta\tau_{hf} = \frac{4 \times 4.135667516 \times 10^3 \mu eV \cdot ps}{0.1 \mu eV} \approx 1.6 \times 10^5 ps \tag{2.42}$$

$$\Delta\tau_{BI} = \frac{2 \times 4.135667516 \times 10^3 \mu J \cdot ps \times 1.60217657 \times 10^{-19}}{-0.5553 \times 5.05078324 \times 10^{-21} \mu J \dot{T}^{-1} \times B_z} \approx -\frac{5 \times 10^5}{B_z} ps \tag{2.43}$$

$$\Delta\tau_{Be} = \frac{2 \times 4.135667516 \times 10^3 \mu J \cdot ps \times 1.60217657 \times 10^{-19}}{-2.0023193043622 \times 9.27400968 \times 10^{-18} \mu J \dot{T}^{-1} \times B_z} \approx -\frac{0.75 \times 10^2}{B_z} ps \tag{2.44}$$

So at around $0.001T$ the hyperfine and magnetic electron interaction will be of approximately equal strength, while the hyperfine and magnetic nuclei interaction will be around equal strength at around $0.5T$. These values will be compared to results from the program to verify the self-consistency of the results.

2.4 One dimensional lattice

So far the distribution and location of the nuclei have not been considered and have instead been assumed to located where their interaction with the electron is of equal strength. However, this is non-physical so some improvement was made by considering a one dimensional periodic lattice. There were two more considerations that were made in this regard, the position of the spins and the strength of interaction as a result of their position.

2.4.1 Hyperfine interaction modulated in space

For the purposes of this model, it was decided that the hyperfine interaction strength should be a modelled as a Gaussian function as this was sufficiently similar to the probability distribution of an electron in a quantum dot. This Gaussian was centered at zero, where the electron's highest probability of occupation was assumed to be. The general form of the Gaussian function is:

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \tag{2.45}$$

Where a determines the height of the peak of the function, b determines the center of the function and c determines then width of the bell curve. Hence for the model interaction:

$$a = 1, b = 0, c = 1 \tag{2.46}$$

Thus the hyperfine interaction hamiltonian becomes:

$$\hat{H} = \frac{1}{2}\vec{\sigma}_e \otimes \sum_{l=1}^{N} \frac{1}{2} A\vec{\sigma}_l e^{-\frac{{x_i}^2}{2}} \tag{2.47}$$

2.4.2 Distribution of Spins

Several different distributions of spin were tested which were switched between within the program. Due to time contraints only two distributions were tested however on the two nuclei system, placing the nuclei at positions equal to the positive and negative standard deviation c and with the nuclei at half of these values.

2.5 Non-iterative Solution

Using the diagonalisation of the hamiltonian it is possible to find the state of the system at any time without needed to iterate in time steps. This involves solving the schrodinger equation exactly and then performing a basis rotation so that the state of the system can be found at any time.

2.5.1 General Solution to TDSE

From our wavefunction basis:

$$|\psi\rangle = \sum_{i=1}^{N} D_i(t)\,|\phi_i\rangle \tag{2.48}$$

If we use a basis comprised of the set of eigenvectors $\{\,\varphi_i, i = 1 \ldots N\,\}$, then by the property $\hat{H}\,|\varphi\rangle = \varepsilon_i\,|\phi_i\rangle$, the right hand side of the TDSE becomes:

$$\hat{H}\,|\psi\rangle = \sum_{i=1}^{N} D_i(t)\hat{H}\,|\psi\rangle = \sum_{i=1}^{N} D_i(t)\varepsilon_i\,|\phi_i\rangle \tag{2.49}$$

So the TDSE becomes:

$$i\hbar\frac{\partial}{\partial t}(\sum_{i=1}^{N} D_i\,|\varphi_i\rangle) = \sum_{i=1}^{N} D_i(t)\varepsilon_i\,|\varphi_i\rangle \tag{2.50}$$

If we introduce a time dependence operator $\hat{U}$ such that:

$$|\psi(t)\rangle = \hat{U}\,|\psi(t=0)\rangle \tag{2.51}$$

The TDSE is clearly solved if the value of U is:

$$\hat{U} = e^{\frac{-i\varepsilon_i t}{\hbar}} \tag{2.52}$$

So the wavefunction at any time can be found by:

$$|\psi\rangle = \sum_{i=1}^{N} D_i(t=0)e^{\frac{-i\varepsilon_i t}{\hbar}}\,|\phi_i(t=0)\rangle \tag{2.53}$$

Thus we have our wavefunction at any time as a function of t, which can be found easily from the eigenstates of our hamiltonian. The difficulty is that this eigenvector basis is different to our information basis, so in order to establish meaningful initial conditions we will need to perform a basis rotation.

2.5.2 Change of basis

In order to establish meaningful initial conditions we perform a basis rotation by first considering the two basis:

$$|\psi\rangle = \sum_{k=1}^{N} C_k(t)\,|\phi_k\rangle \tag{2.54}$$

$$|\psi\rangle = \sum_{i=1}^{N} D_i(t)\,|\varphi_i\rangle \tag{2.55}$$

So:

$$\sum_{i=1}^{N} D_i(t)\,|\varphi_i\rangle = \sum_{k=1}^{N} C_k(t)\,|\phi_k\rangle \tag{2.56}$$

Multiplying both sides by $\langle\varphi_j|$, the j-th eigenvector gives:

$$\langle\varphi_j|\sum_{i=1}^{N} D_i(t=0)|\varphi_i\rangle = \langle\varphi_j|\sum_{k=1}^{N} C_k(t=0)|\phi_k\rangle \tag{2.57}$$

The left hand side can be re-arranged such that:

$$\sum_{i=1}^{N} D_i(t=0)\,\langle\varphi_j|\varphi_i\rangle = \sum_{i=1}^{N} D_i(t=0)\delta_{ij} \tag{2.58}$$

And δ_{ij} is zero unless $i = j$, thus:

$$D_j(t=0) = \sum_{k=1}^{N} C_k(t=0)\,\langle\varphi_j|\phi_k\rangle \tag{2.59}$$

Thus we have a relation to rotate our initial conditions from one basis to another. This relation was implemented into the program such that the initial conditions could be easily set.

1	Initialise the basis
2	Generate the hamiltonian for the system
3	Find the eigenstates of the hamiltonian through diagonalisation
4	Solve the TDSE for a certain time
5	Find the fidelity of the state at that time
6	Determine and record the timescale of interaction

TABLE 2.1: Table of program specifications

2.6 Implementation

The simulation was implemented in fortran 90, details of which are listen in A. The basic specification for the program was that it needed to:

All code was self written, although some usage was made of common linear algebra libaries, however there were some unique implementation challenges that needed to be overcome. The following details some of the subroutines developed for this project.

2.6.1 Scalable Hamiltonians

For ease in running the simulations it was required to be able to alter the size of the hamiltonians with the number of nuclei. For this a subroutine which performed kronecker products on the pauli matrices as described in (2.24). Routines were developed for the computation of both magnetic and hyperfine hamiltonians from Pauli matrices which are detailed in Appendix B. Performing this task required a kronecker product subroutine which was standard across both hamiltonian subroutines, detailed in the next subsection.

2.6.2 Kronecker Product Routine

In order to perform Kronecker products for the computation of the hamiltonian a kronecker product subroutine was developed as a suitable version was not found in a library. The developed subroutine is listen in Appendix B. The routine performs double precision complex kronecker products for matrices of dimension $m \times n$ and $k \times l$ producing a matrix of dimension $mk \times nl$, which should work with two dimensional matrices of any size, making it sufficient for the application necessary.

2.6.3 Time scale of interaction routine

In order to consistently measure the timescale of interaction for the hyperfine and magnetic interactions. Doing this required the ability to measure the distance between maxima and minima in $|C_i|^2$ as well as distinguish between maxima and minima of the same size. This is performed by reading values from the final output file of $|C_i|^2$ and comparing values between timesteps to find maxima and minima. Values of these maxima and minima are then stored to be compared to other values and if a match is found then it is reported and the time between the points is recorded in the file. In this way we have a consistent measure of timescale to be used in analysis of the interactions. This routine can be found in Appendix B

Chapter 3

Results and Discussion

3.1 Spin Chain

For the simple spin chain with a basis of three states corresponding to one up spin in each state, we obtained the results displayed in figure (3.1). Here the probabilities of occupation for each state in the basis $|C_i|^2$ are plotted against the number of time steps performed by the program. As this simulation was unphysical it was decided to display time-steps instead of using the time based on the constants of the TDSE as the coupling constant J was arbitrary. The initial conditions in the graph are $C_1 = 1$ which was chosen as a good starting point for investigation.

Written explicitly the basis set for this system is:

$$|\psi\rangle = \sum_{i=1}^{N} C_i(t)\,|\phi_i\rangle = C_1\,|\uparrow\downarrow\downarrow\rangle + C_2\,|\downarrow\uparrow\downarrow\rangle + C_3\,|\downarrow\downarrow\uparrow\rangle \tag{3.1}$$

It was found that for this case the probabilities of occupation for each state oscillated with time and that these oscillations were periodic. The graph also shows that the most probable state will pass down the chain, almost like a wave, before being reflected at either end, resulting in the states corresponding to spin up at either end of the chain being most probable.

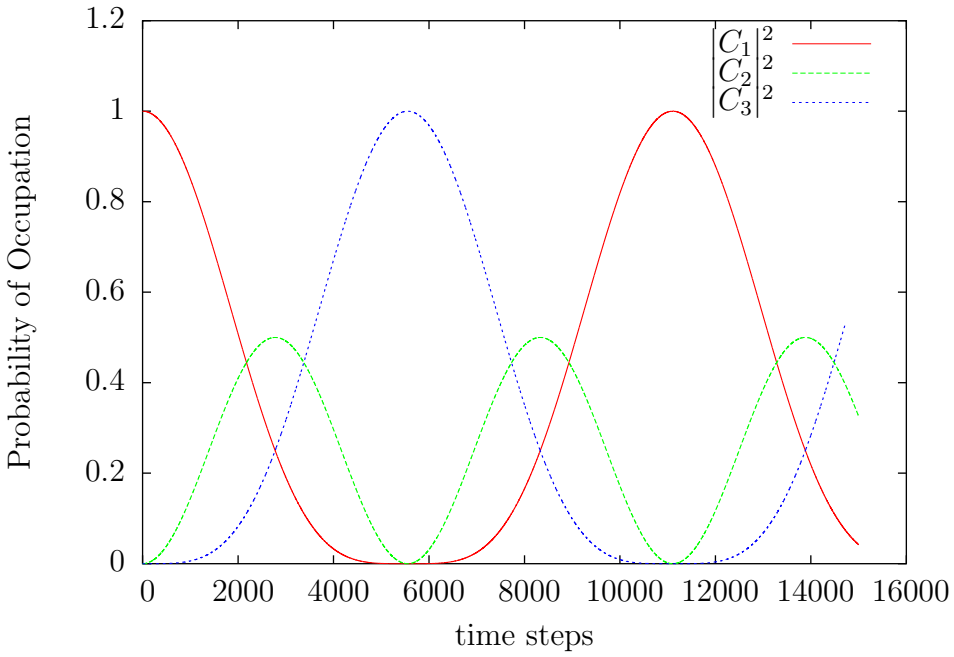

FIGURE 3.1: Graph of the probabilities of occupation for a 3 Spin Chain with initial condition $C_1 = 1$

3.1.1 Larger Spin Chains

Continuing the investigation to larger spin chains also yielded interesting results as displayed in (3.2)

For four spins, the same oscillatory behaviour was observed, however the oscillations did not seem periodic within the number of timesteps calculated by the program. Thus the program was run again with a larger number of time-steps as shown in figure (3.3).

On this larger scale it seemed that there was unlikely to be periodic behaviour, or that such behaviour would at least require a very long time period to be apparent. So a 5 spin chain was also investiaged for similarities as in figure (3.4).

As can be seen from figure (3.4), there is little periodicity, much as in figure (3.2). It therefore seems that for chains of over three spins there is no periodicity with the initial conditions $C_1(t = 0) = 1$ and a constant coupling constant J

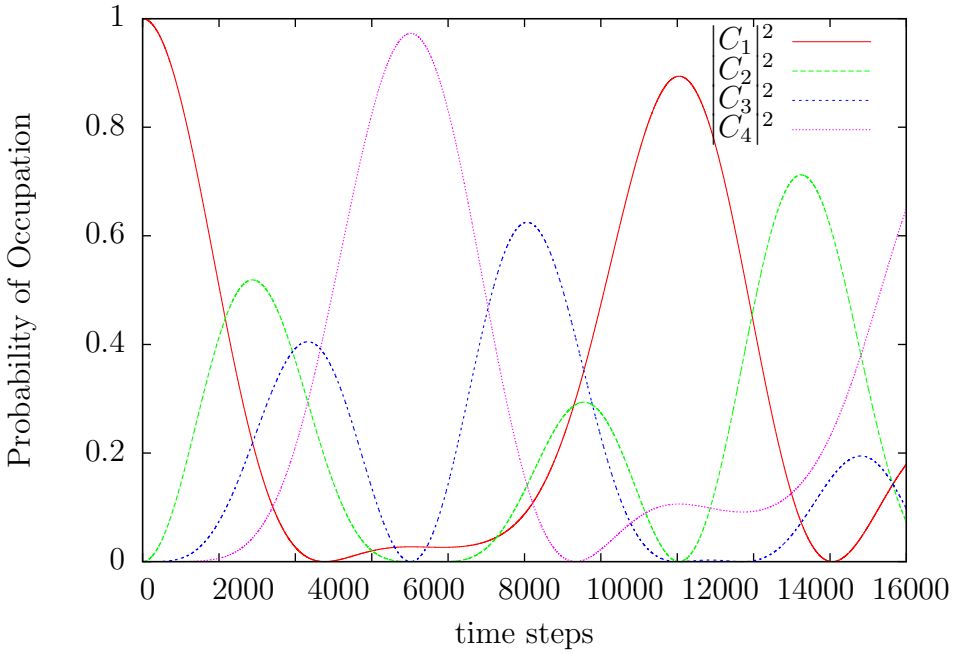

FIGURE 3.2: Graph of the probabilities of occupation for a 4 Spin Chain with initial condition $C_1 = 1$

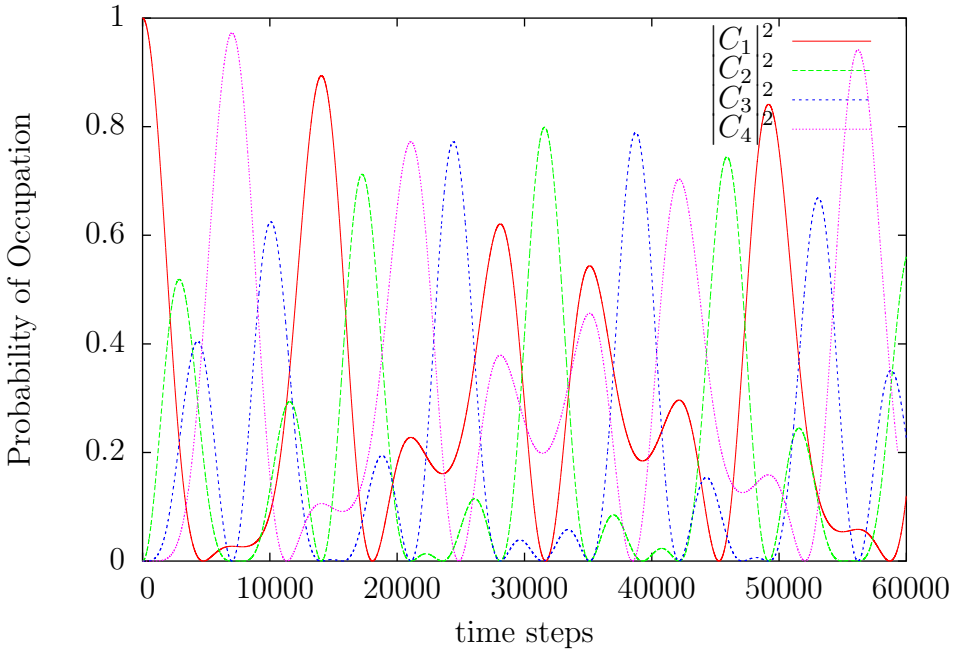

FIGURE 3.3: Graph of the probabilities of occupation for a 4 Spin Chain with initial condition $C_1 = 1$ for an extended number of time-steps

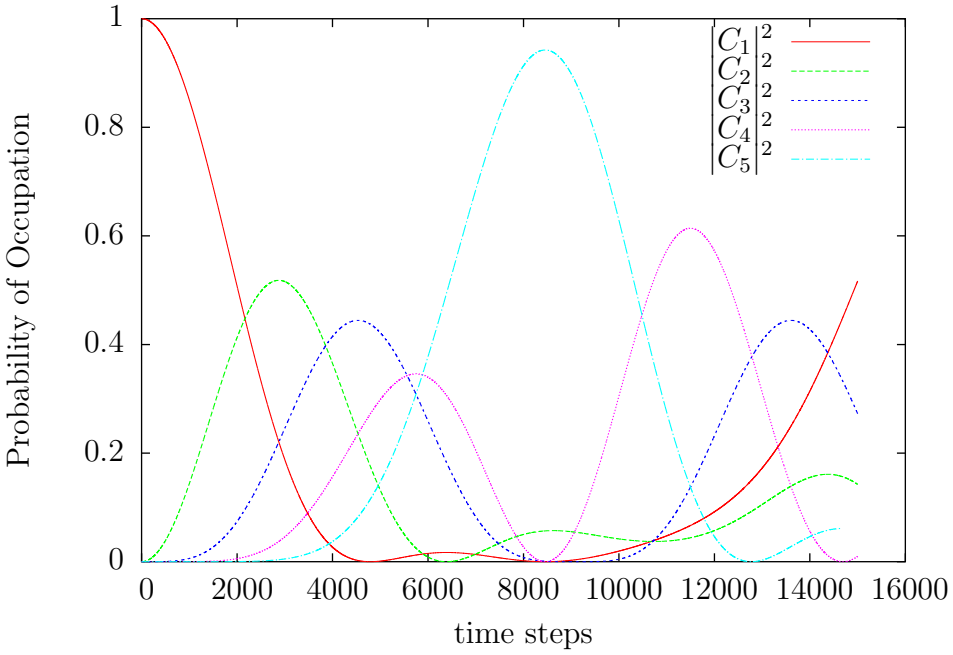

FIGURE 3.4: Graph of the probabilities of occupation for a 5 Spin Chain with initial condition $C_1 = 1$

3.1.2 Alternative Initial Conditions

In order to see if this was the case for all initial conditions, an alternative set of conditions $C_i(t = 0) = \frac{1}{\sqrt{N}}; i = 1, N$, where N is the number of spins, was chosen. For the 3 spin case the results are displayed in figure (3.5).

For the three spin chain, with these alternate initial conditions, similar periodic behaviour was observed. However it seemed that the most probable state after time evolution was C_2, which was likely the case due to the "wave" of probabilties traveling to either end of the chain and then reflecting so that their amplitude peaked in the center of the chain.

The same initial conditions were then computed for 4 spins in figure (3.6).

From figure (3.6), we can see similar behaviour to the case of 3 spins. However, as there are two states corresponding a central spin up particle, we can see that both states have equal probabilities. Interestingly, for these initial conditions it seems to be the case that the probabilities are periodic, unlike with the $C_1(t = 0) = 1$ initial condition. Similarly the case of 5 spins is displayed in figure (3.7).

In figure (3.7), the same behaviour is illustrated again. In this case however, there is one central spin up state and two spins between the boundaries. Hence the

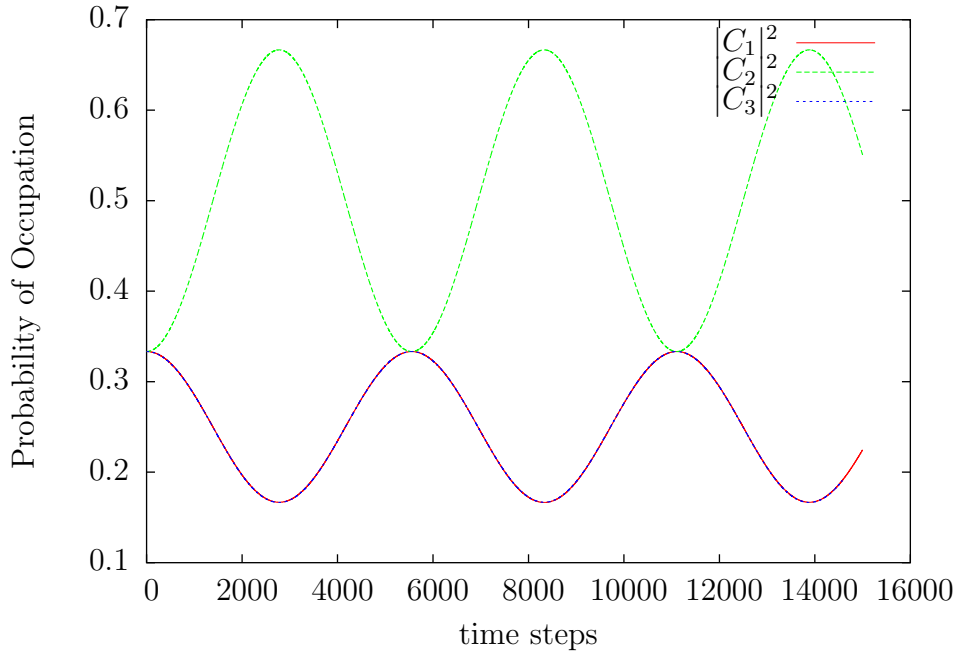

FIGURE 3.5: Graph of the probabilities of occupation for a 3 Spin Chain with initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

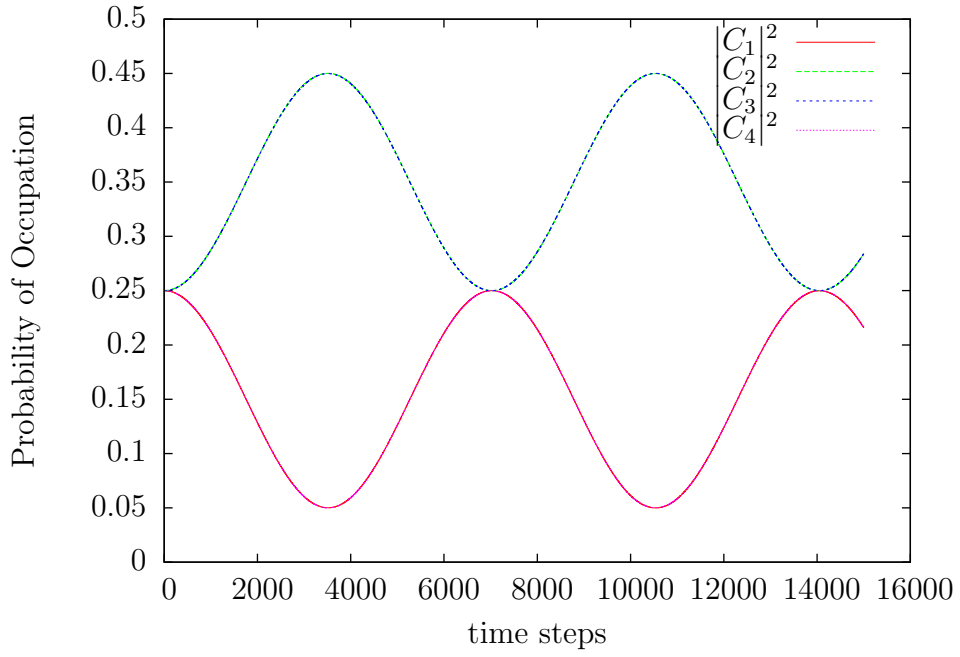

FIGURE 3.6: Graph of the probabilities of occupation for a 4 Spin Chain with initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

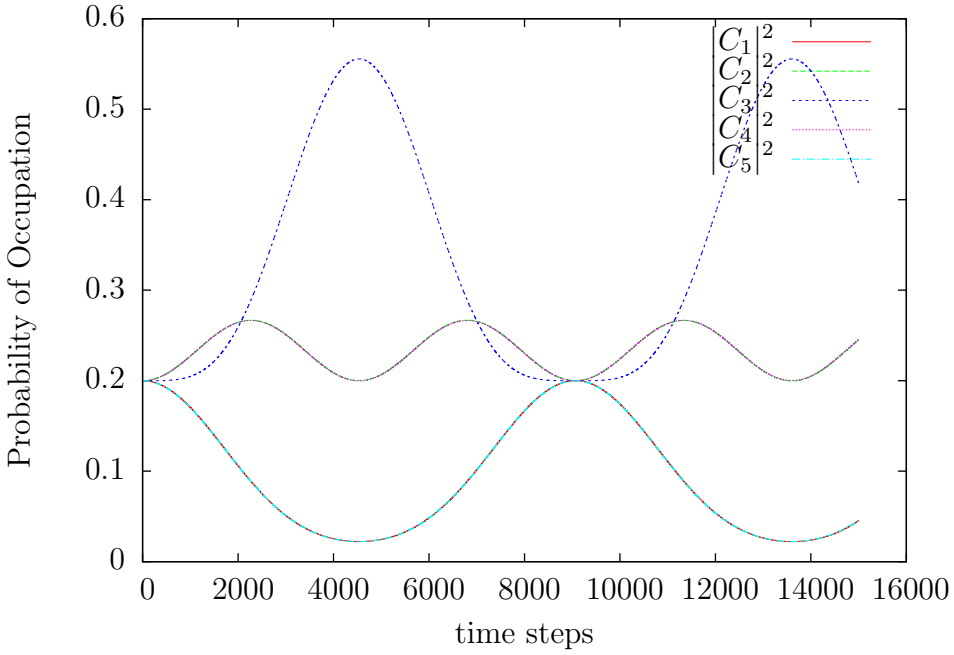

FIGURE 3.7: Graph of the probabilities of occupation for a 5 Spin Chain with initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

probability of the central spin being spin up has the highest peak and the ends of the system the lowest trough, while the other two states oscillate a minimal amount. This clearly illustrates this "reflective" behaviour as there appears to be nodes on the central spin up state and the ends, while the other two states have almost anti-node oscillations.

It seems reasonable that this trend would continue for odd and even spins, with the probability being highest for the central spins, being either a single spin or pair of spins.

3.1.3 Periodicity

In addition to these initial conditions, a coupling constant specific to each interaction $J_{i,i+1}$ was tested, given by equation (2.9) as it was thought that this would result in periodic behaviour for any set of initial conditions. This was unneccessary to test on the 3 spin case as that was already periodic so the four spin case was the first to be simulated as shown in figure (3.8), which is identical in input to figure (3.2), aside from this new coupling constant.

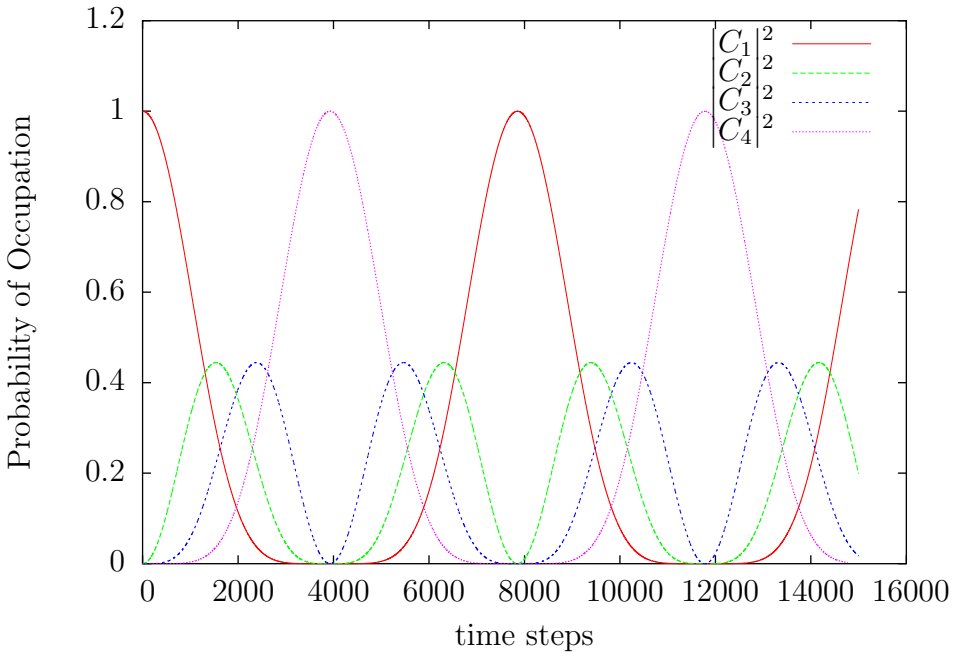

FIGURE 3.8: Graph of the probabilities of occupation for a 4 Spin Chain with a periodic coupling constant and initial condition $C_1 = 1$

As can be seen from figure (3.8), it is clear that this set of coupling constants does indeed cause there to be periodic behaviour in the oscillation of probabilities.

Similarly for 5 spins there is (3.9) which demonstrates the same behaviour.

Next, the combination of the $\{ C_i(t=0) = \frac{1}{\sqrt{N}}; i = 1, N \}$ initial conditions and the new coupling constants were tested, first with four spins in figure (3.10) and then with five in figure (3.11)

In figure (3.10), the same behaviour is observed as in figure (3.6), the only difference being period of the oscillations in probability. This shows that the new coupling constant has no perculiar or novel effect on the system if it is already periodic due to it's initial conditions

The same behaviour is illustrated in figure (3.11), where a similar behaviour is observed to the four spin case, with an equally similar change in period of the oscillations.

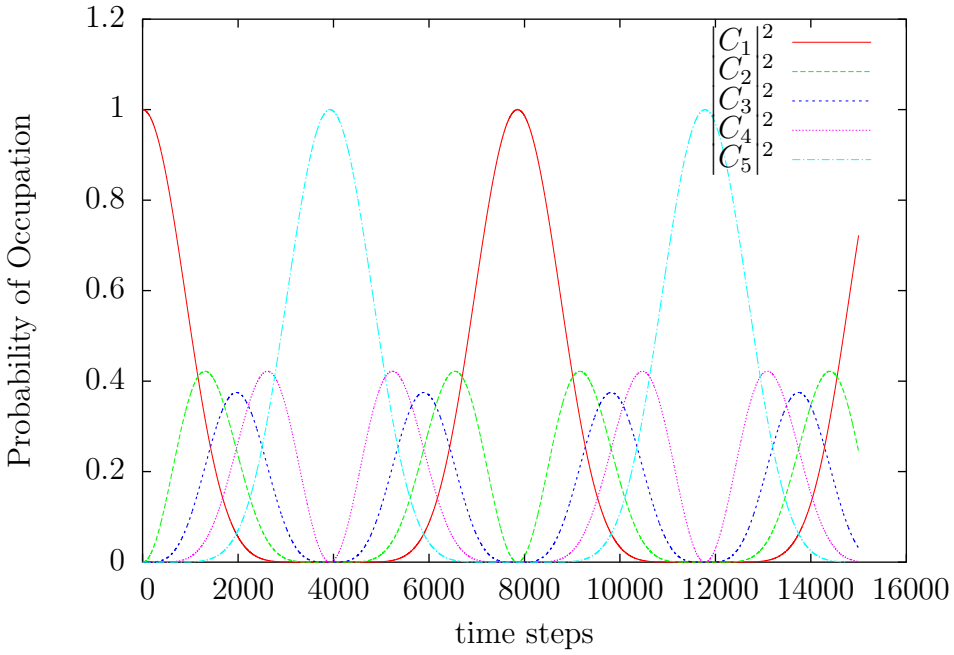

FIGURE 3.9: Graph of the probabilities of occupation for a 5 Spin Chain with a periodic coupling constant and initial condition $C_1 = 1$

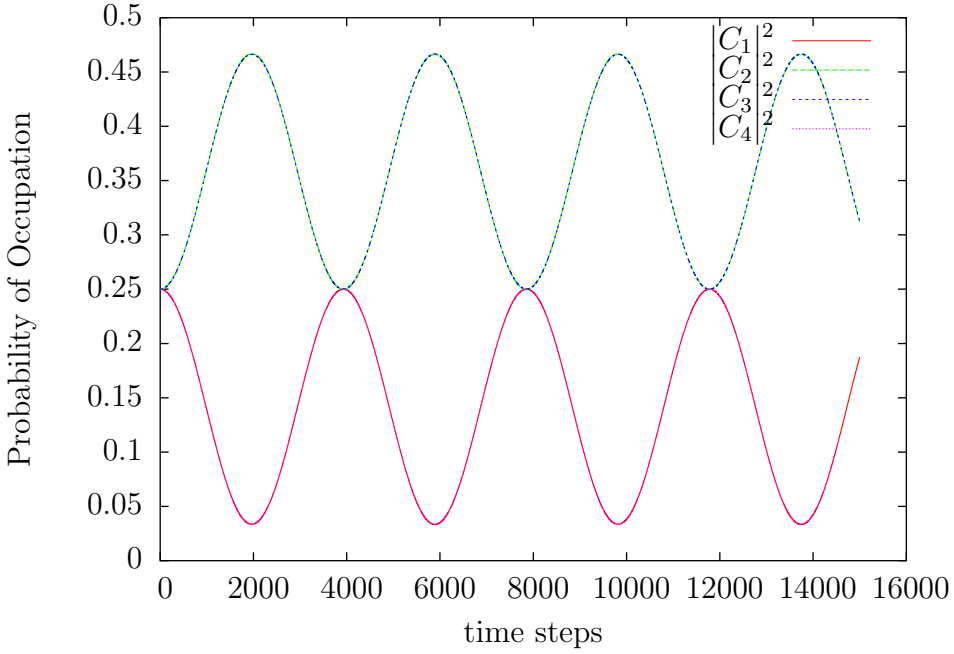

FIGURE 3.10: Graph of the probabilities of occupation for a 4 Spin Chain with a periodic coupling constant and initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

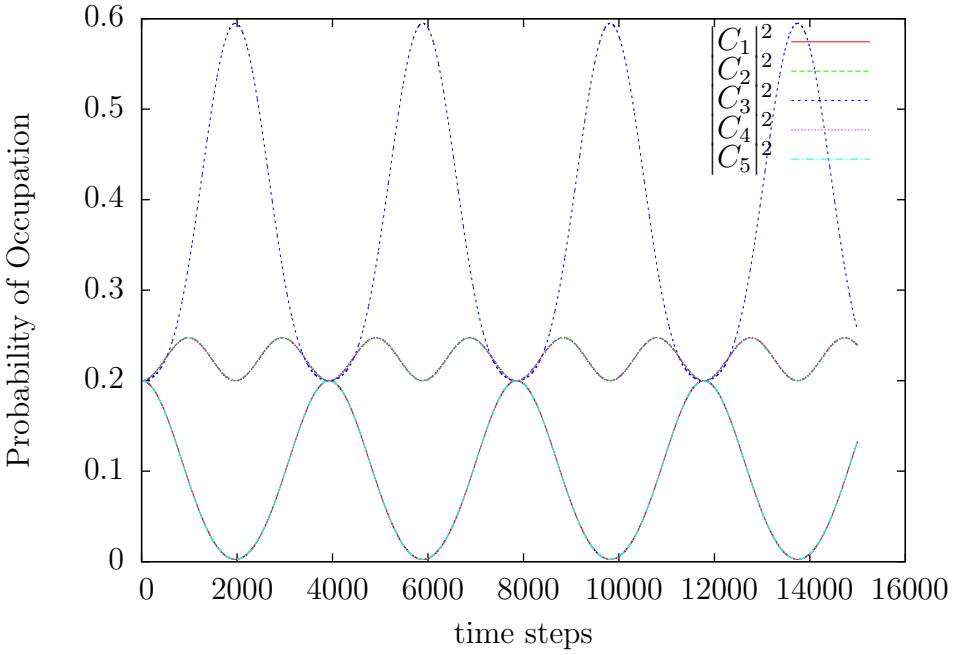

FIGURE 3.11: Graph of the probabilities of occupation for a 5 Spin Chain with a periodic coupling constant and initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

3.1.4 Error

For each of these simulations the error as measured by $\sum_{i=1}^{N} |C_i|^2$ was recorded at each time-step in order to make sure the simulation was not runing into difficulties. What follows is an analysis of the change in error over time and how it relates to the results.

In figure (3.12), it can be seen that the sum of probabilities always remains approximately 1 throughout the duration of the simulation, with a small amount of oscillation of around ± 0.0003, which will also be the absolute and relative error. These oscillations also have approximate the same period as figure (3.1), so this oscillation is likely to simply be characterisitic of the system. It is therefore concluded that the iterative method adopted is sufficiently accurate for the simple spin chain type system with the variables chosen.

A similar situation is observed in figure (3.13), where the same oscillation is observed. The only noticable difference between this system and the system in figure (3.12) is the period of oscillation, which is slightly longer, the absolute and relative error are around the same.

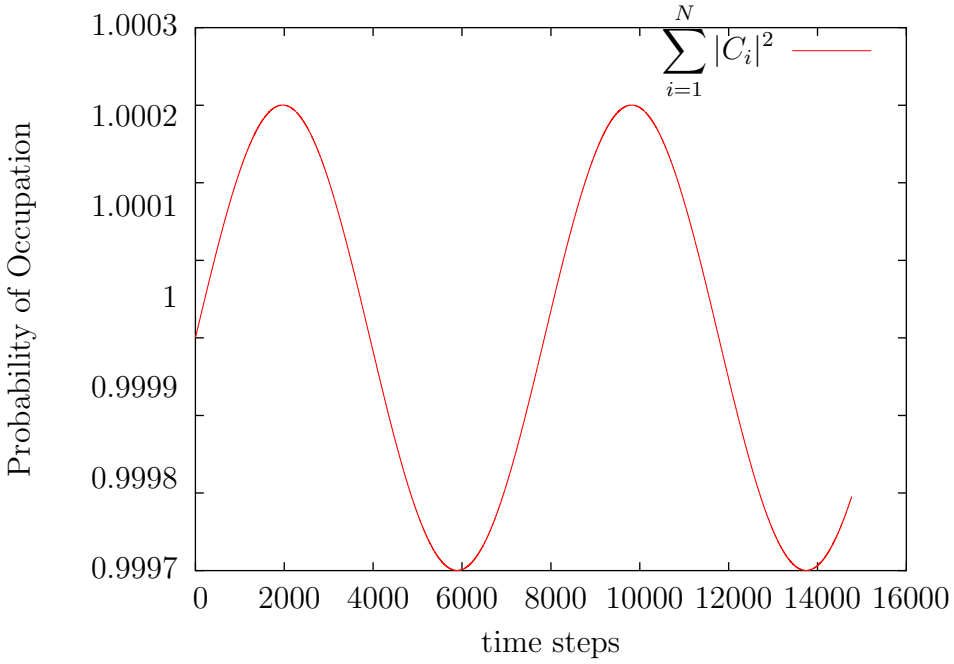

FIGURE 3.12: Graph of the sum of the probabilites of occupation of states for the entire basis set with 3 spins and initial condition $C_1 = 1$

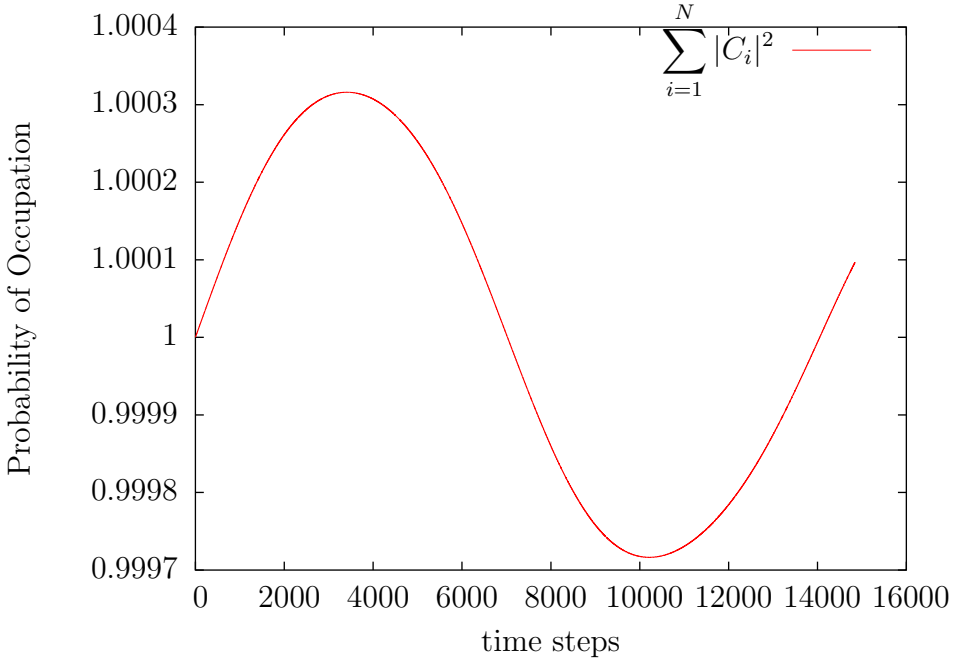

FIGURE 3.13: Graph of the sum of the probabilites of occupation of states for the entire basis set with 4 spins and initial condition $C_1 = 1$

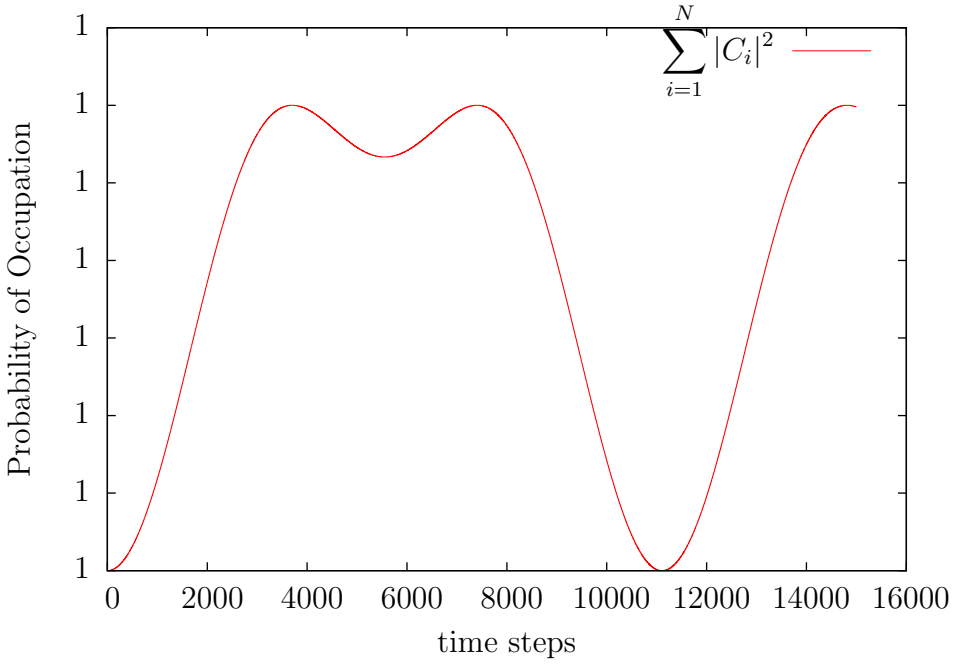

FIGURE 3.14: Graph of the sum of the probabilites of occupation of states for the entire basis set with 3 spins with initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

Figure (3.14) shows the sum of probabilities with the $C_i(t = 0) = \frac{1}{\sqrt{N}}; i = 1, N$ initial conditions and 3 spins. From this graph it seems that the nature of the oscillation of the sum of the probabilities is specific to the system as suspected. There is a slight problem with the axes displayed on this graph as all of the values on the y axis are displayed as one. This is due to the rounding of the values perfomed by gnuplot and as such the values on the axis are less than 0.0001, so it the error of the simulation for the system is minimal.

Figure (3.15) is similar to the three spin case with the alternative initial conditions, however the time frame for this graph was too short as it was made as a pair from the simulation run for the graph in figure (3.6). It would be interesting to run the simulation over a longer period of time to see if the sum of probabilities is periodic in the same way as the basis co-efficients.

The five spin case in figure (3.16) is also similar to the four spin case in that it cannot be stated with certainty whether it is periodic. However, the same low amount of error is observed.

For the coupling constant that resulted in periodic results, displayed in figure (3.17), it was found that the error oscillated in a similar way to the single value

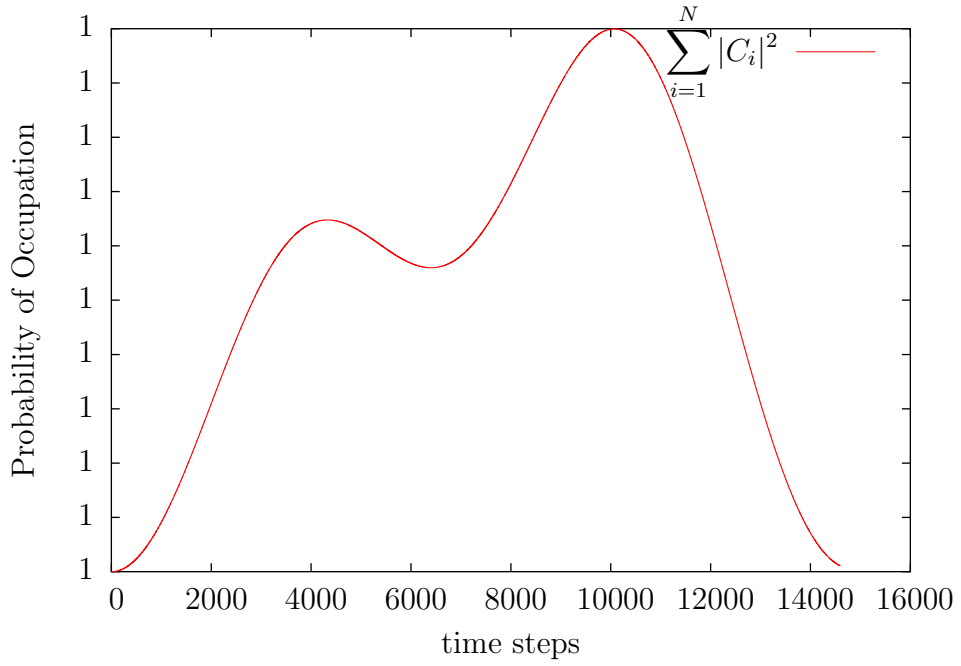

FIGURE 3.15: Graph of the sum of the probabilites of occupation of states for the entire basis set with 4 spins with initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

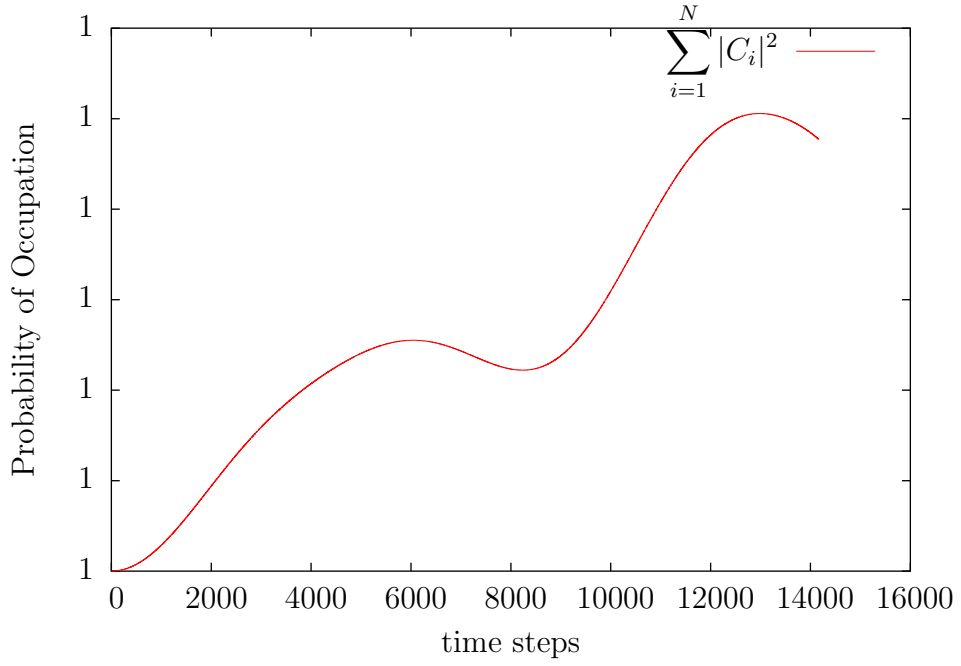

FIGURE 3.16: Graph of the sum of the probabilites of occupation of states for the entire basis set with 5 spins with initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

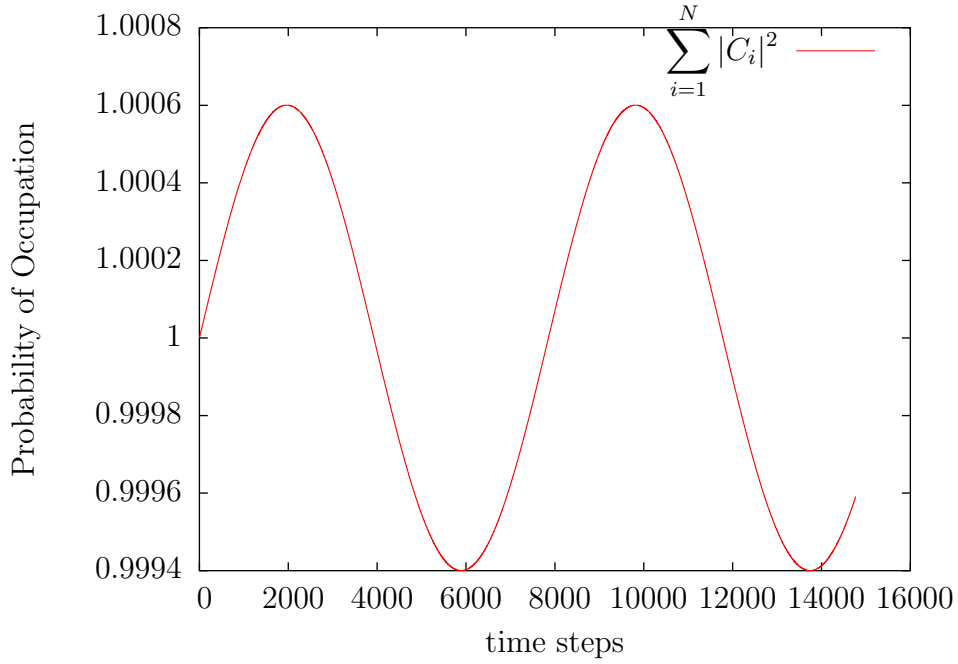

FIGURE 3.17: Graph of the sum of the probabilites of occupation of states for the entire basis set with 4 spins with a periodic coupling constant and initial condition $C_1 = 1$

coupling constant shown in figure (3.13), which indicates that the oscillation in error may be more due to initial conditions than the system itself.

When the initial conditions and periodic coupling constants were combined, as in figures (3.18) and (3.19), it was found that the oscillation of error was both small and periodic.

From the figures in this subsection, it seems quite likely that the results from the spin chain program were sufficiently accurate for the results of the program to be accepted assuming any error came from a lack of numerical precision and not underlying theory. From here we progress onwards to representations of pseudo-physical systems.

3.2 Hyperfine

The first set of results from the hyperfine hamiltonian program and displayed in figures (3.20 - 3.23). These initial results were for the very simple one electron one nuclei system so they are very basic, but provide some important insight into the system. For this system the complete basis set is written:

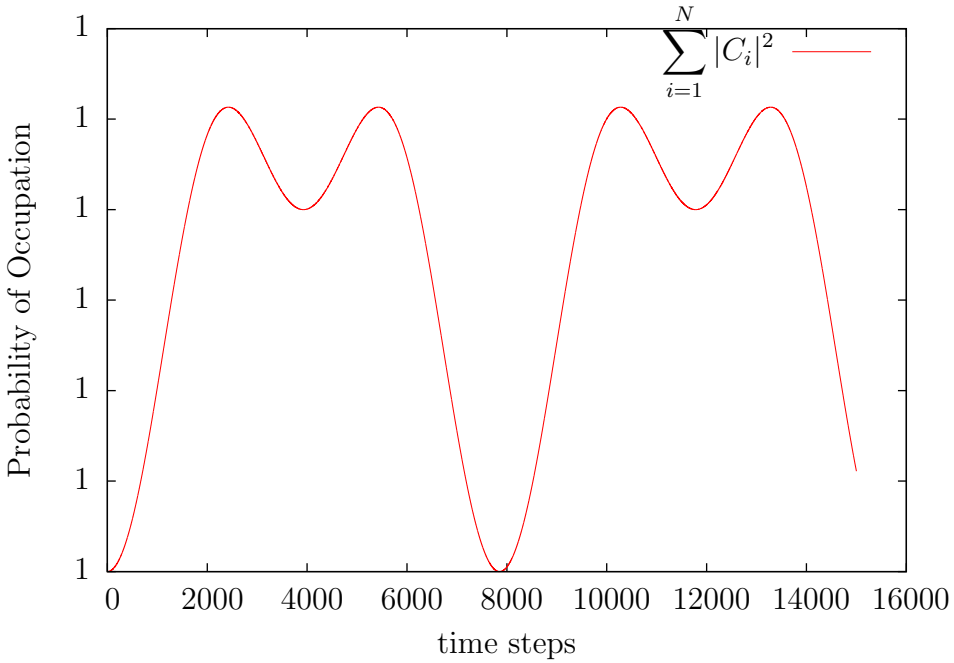

FIGURE 3.18: Graph of the sum of the probabilites of occupation of states for the entire basis set with 4 spins with a periodic coupling constant and initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

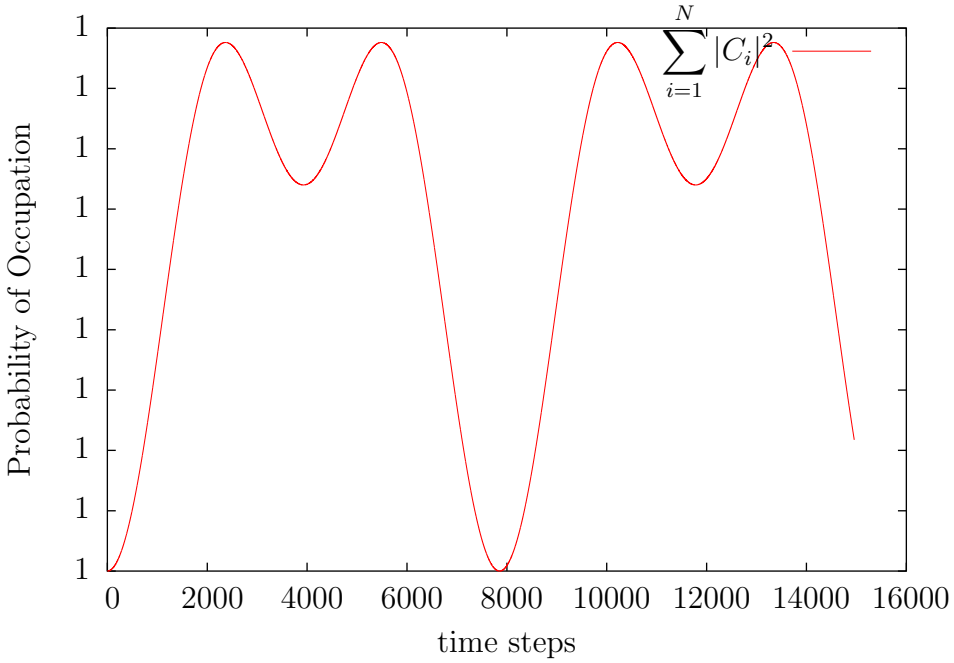

FIGURE 3.19: Graph of the sum of the probabilites of occupation of states for the entire basis set with 5 spins with a periodic coupling constant and initial conditions $C_i = \frac{1}{\sqrt{N}}; i = 1, N$

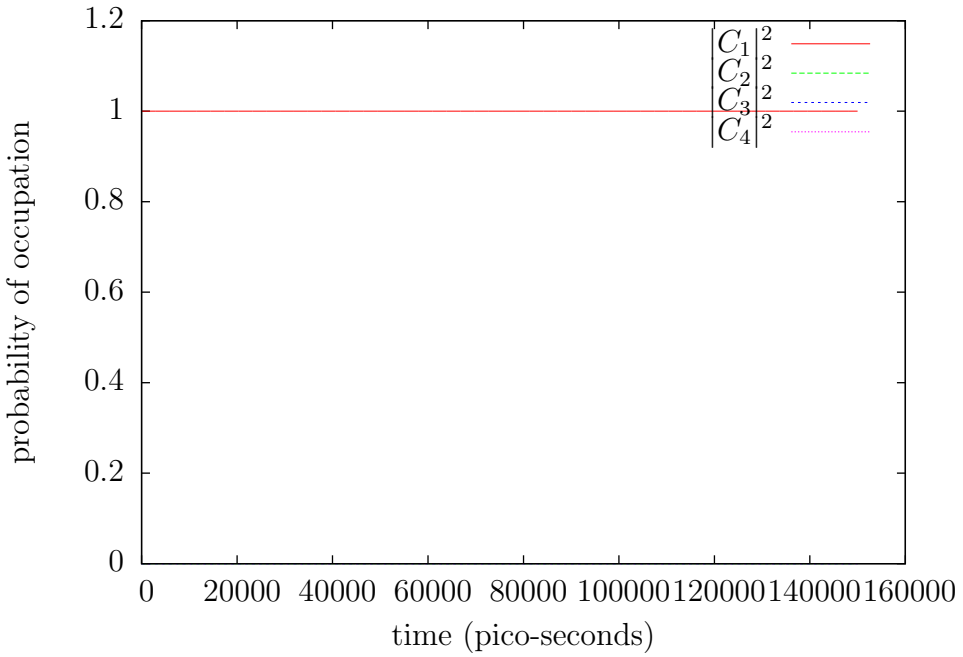

FIGURE 3.20: Graph the time evolution of the probabilities of the electron hyperfine system with 1 nuclei and initial condition $C_1 = 1$

$$|\psi\rangle = \sum_{i=1}^{N} C_i(t)\,|\phi_i\rangle = C_1\,|\Uparrow;\uparrow\rangle + C_2\,|\Uparrow;\downarrow\rangle + C_3\,|\Downarrow;\uparrow\rangle + C_4\,|\Downarrow;\downarrow\rangle \tag{3.2}$$

In figure (3.20), the system remains in a single state throughout time, indicating that the $|\Uparrow;\uparrow\rangle$ state seems to be an eigenvalue of the system. This seems reasonable as the system is fully polarised and so their can be no exchange of spins between the particles.

In figures (3.21) and (3.22), we find the same periodic oscillation of probabilities as those found in the spin chain between the $|\Uparrow;\downarrow\rangle$ and $|\Downarrow;\uparrow\rangle$ which seems to be reasonable.

Finally, figure (3.23) displays the last pure state initial condition. Similar to the results in (3.20), this is periodic and thus the $|\Downarrow;\downarrow\rangle$ state is likely to also be an eigenstate. This was then investigated further with the eigenvalue subroutine in the program.

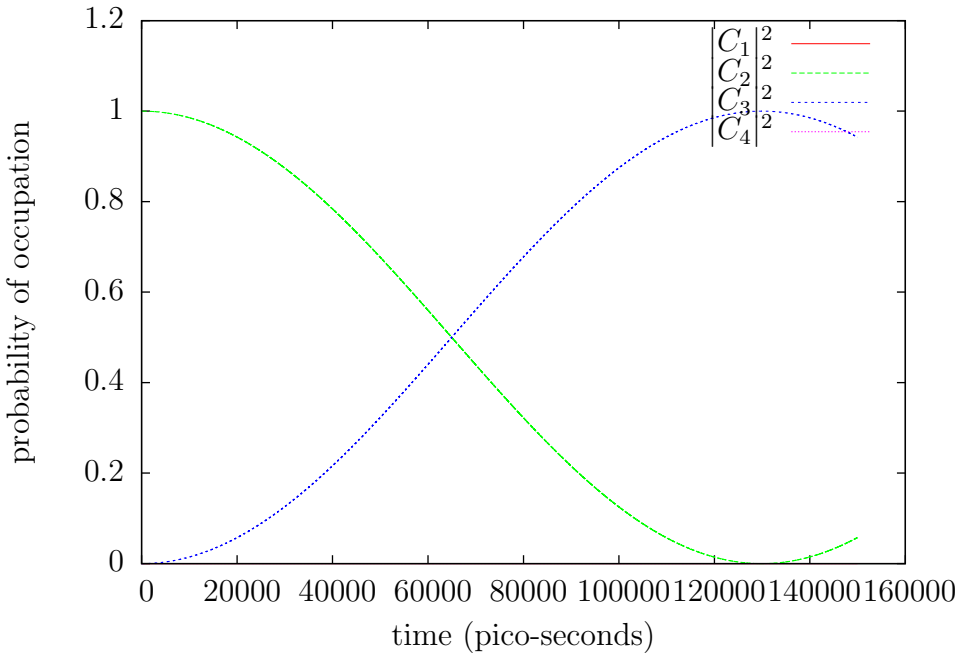

FIGURE 3.21: Graph the time evolution of the probabilities of the electron hyperfine system with 1 nuclei and initial condition $C_2 = 1$

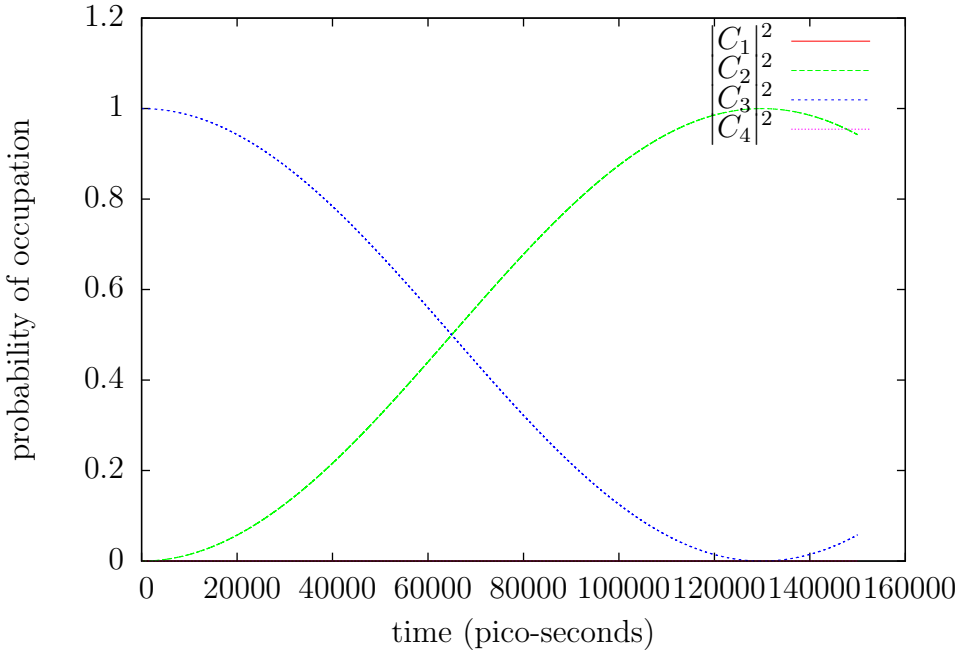

FIGURE 3.22: Graph the time evolution of the probabilities of the electron hyperfine system with 1 nuclei and initial condition $C_3 = 1$

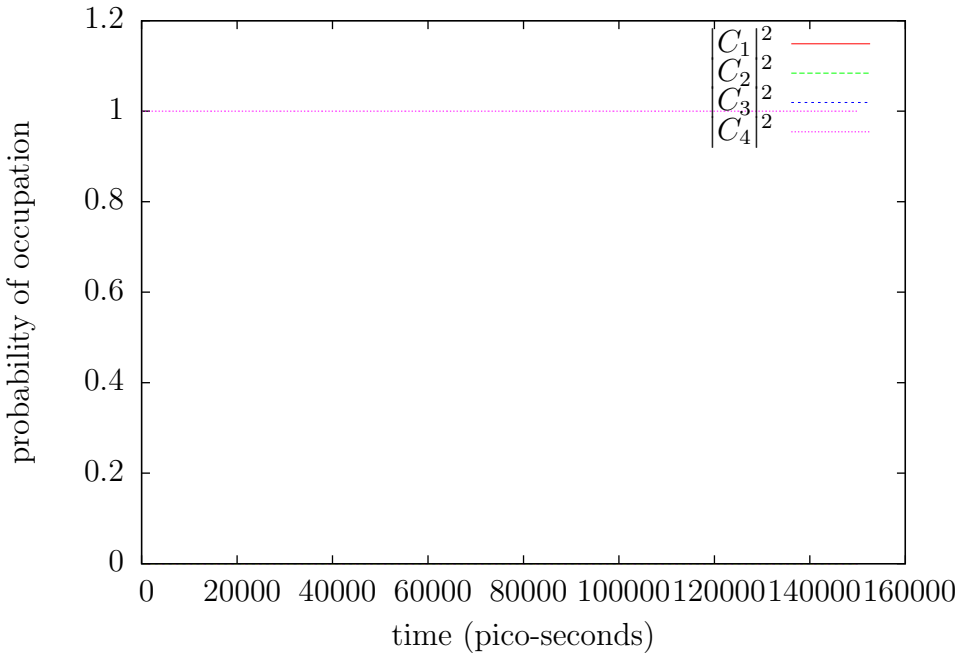

FIGURE 3.23: Graph the time evolution of the probabilities of the electron hyperfine system with 1 nuclei and initial condition $C_4 = 1$

Eigenvalues	0.025	-0.075	0.025	0.025
Eigenvectors	0	0	1	0
	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0	0
	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	0	0
	0	0	0	1

TABLE 3.1: Eigenvalues of the electron hyperfine hamiltonian with 1 nuclei

3.2.1 Eigenstates

The eigenvalues yielded by the program for the single nucleus system are displayed in table (3.1).

Based on these eigenvalues, the results from system with the $C_1 = 1$ and $C_4 = 1$ seem to match expectations as the associated states form the 3rd and 4th eigenvectors of the hamiltonian. In order to examine this further the initial conditions $C_2 = \frac{1}{\sqrt{2}}; C_3 = \frac{1}{\sqrt{2}}$ where also examined in order to show that these conditions also resulted in stationary states and the results of this are displayed in figure (3.24), also agreeing with expectations.

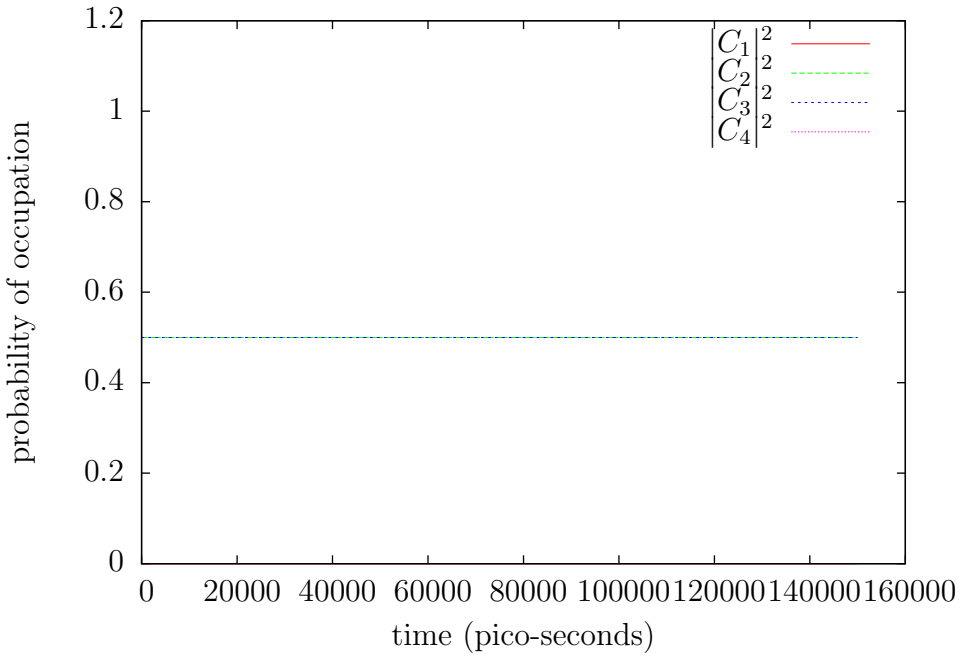

FIGURE 3.24: Graph the time evolution of the probabilities of the electron hyperfine system with 1 nuclei and initial conditions $C_2 = \frac{1}{\sqrt{2}}; C_3 = \frac{1}{\sqrt{2}}$

3.2.2 Number of Spins

Extending the system to 2 nuclei, we find a few more interesting properties of the system. In figure (3.25), much as in the 1 nuclei case, we find that the initial condition of $C_1 = 1$ where all the spins are aligned is a stationary state and this result is also confirmed by the eigenvalue routine.

Continuing to other initial conditions, we find that in the case of the initial conditions $C_2 = 1$ and $C_3 = 1$ that the only states which become probable after time evolution are those with an identical number of up and down spins to those in the initial conditions. For example, when initialising the system in the state $|\Uparrow;\uparrow\downarrow\rangle$, the only possible states the system can be found in upon measurement are $|\Uparrow;\uparrow\downarrow\rangle$, $|\Uparrow;\downarrow\uparrow\rangle$ and $|\Downarrow;\uparrow\uparrow\rangle$. Hypothetically, this property could be used to reduce the size of the matrices involved in the simulation and is very similar to the simple spin chain discussed earlier, however as an external magnetic field is also used in the program this is not practical for this simulation.

Finally, there is an interesting result in figure (3.28) as states $|\Downarrow;\uparrow\downarrow\rangle$ and $|\Downarrow;\downarrow\uparrow\rangle$) both have equal probability at all times. This is because the system is initialised in the state $|\Uparrow;\downarrow\downarrow\rangle$, so the spin up state can "pass" to either of the two nuclei as only the fermi contact is considered. This causes the same probability density

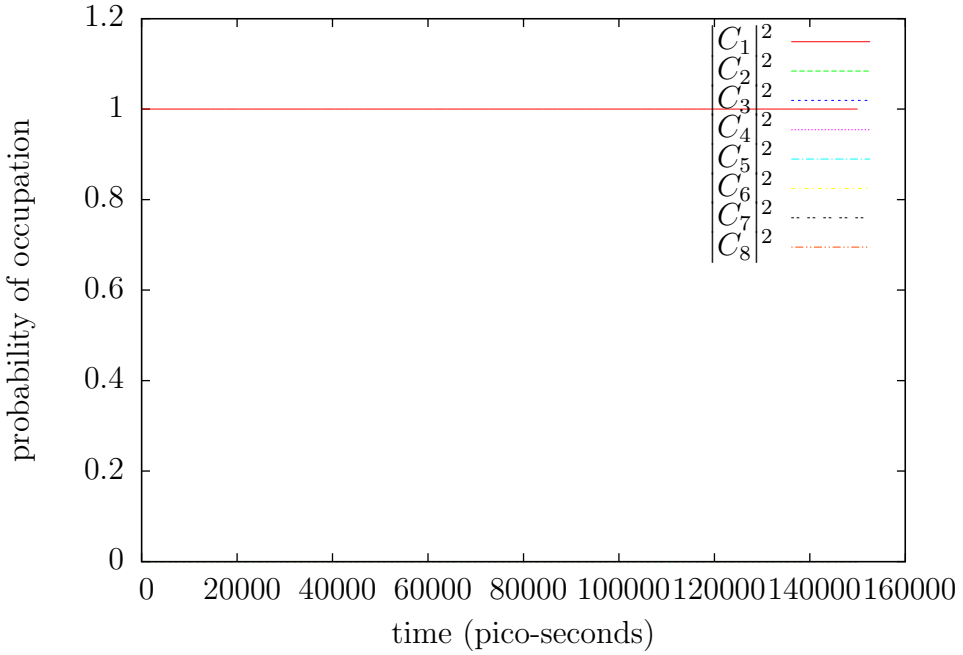

FIGURE 3.25: Graph the time evolution of the probabilities of the electron hyperfine system with 2 nuclei and initial condition $C_1 = 1$

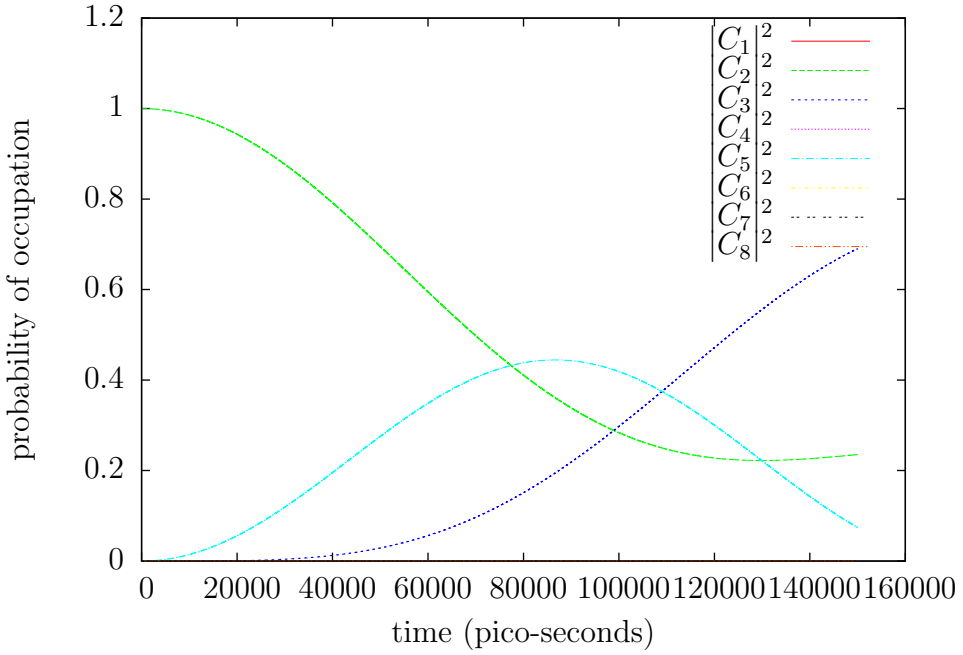

FIGURE 3.26: Graph the time evolution of the probabilities of the electron hyperfine system with 2 nuclei and initial condition $C_2 = 1$

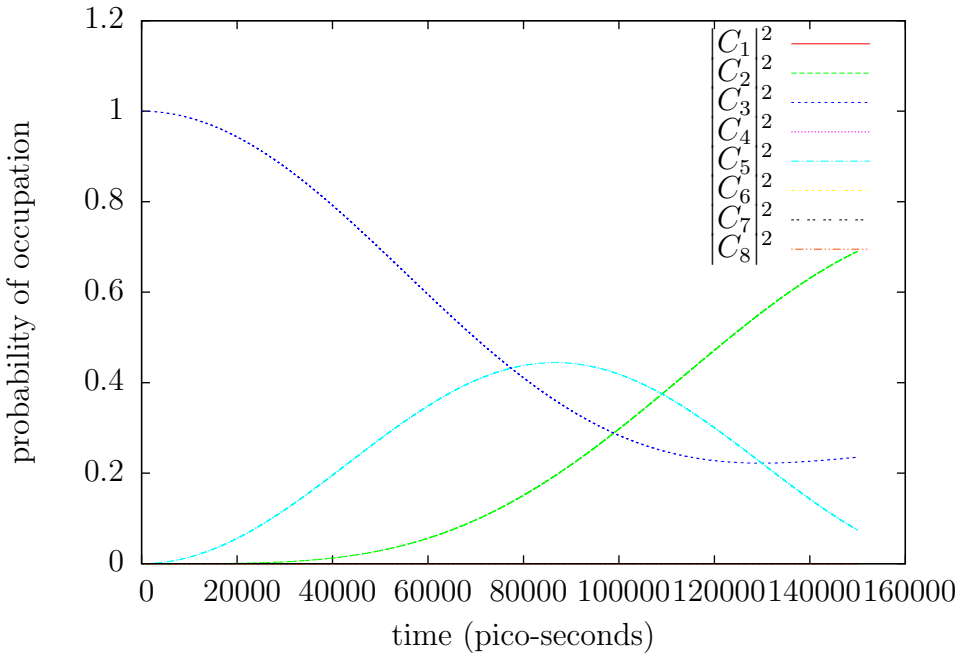

FIGURE 3.27: Graph the time evolution of the probabilities of the electron hyperfine system with 2 nuclei and initial condition $C_3 = 1$

reflective properties to occur as seen in the spin chain with a central up spin where we find nodes and anti-nodes in the probability.

These results are symmetric, with the behaviour of C_4 being identical to that of C_5 and the behaviour of C_1 being identical to that of C_8, which is to be expected as there both nuclei interact equally with the electron as the modulation of the interaction due to the location of the nuclei is not part of these results.

3.2.3 Fidelity

For the hyperfine interaction, fidelity was found to be somewhat similar to the time evolution of the probabilities for pure states. This behaviour is illustrated in figures (3.29) and (3.30). This seems reasonable as the inner product of the basis vectors corresponding to the states with the total wavefunction would result in the "selection" of value in the row corresponding to the unity value in the basis vector. This would result in the basis co-efficient being selected and then the modulus squared of it's value being found, resulting in the probability of that basis state. This method might be more interesting for mixed states, but it when the magnetic field is introduce the effect on the fidelity is interesting.

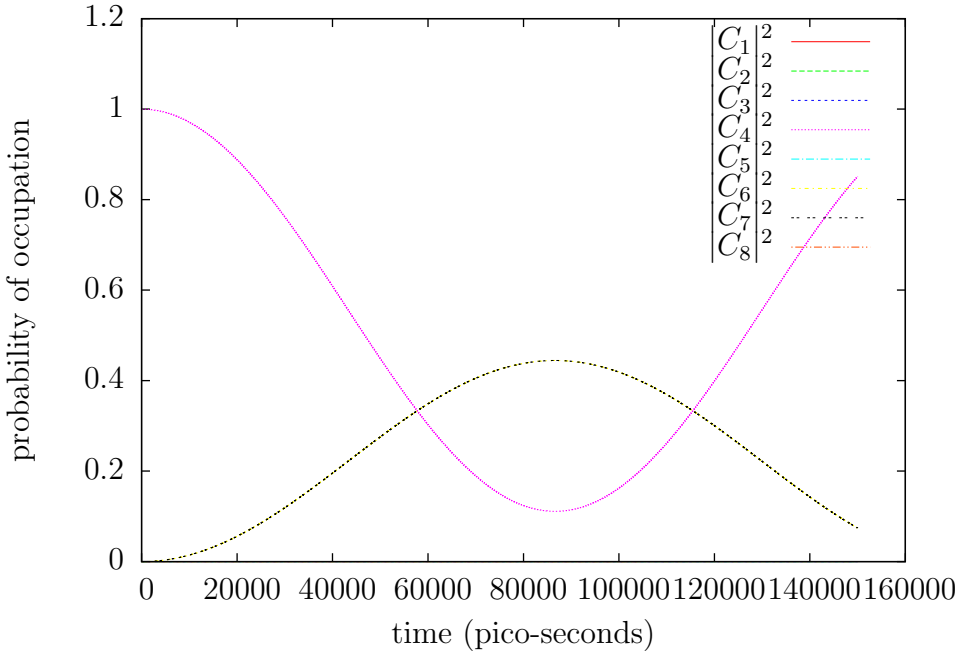

FIGURE 3.28: Graph the time evolution of the probabilities of the electron hyperfine system with 2 nuclei and initial condition $C_4 = 1$

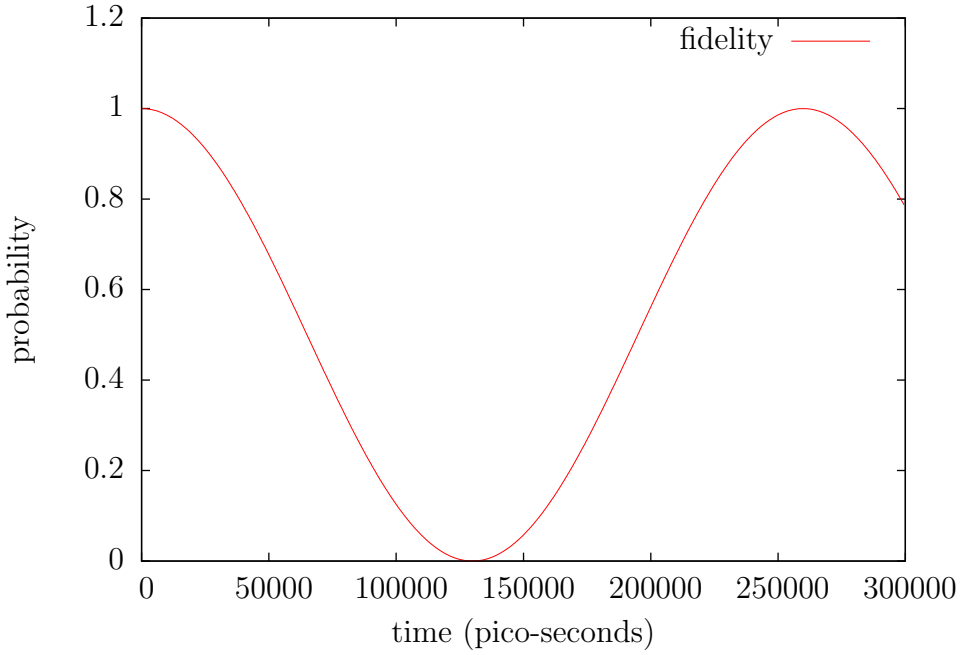

FIGURE 3.29: Graph of the fidelity of the electron hyperfine system with 1 nuclei and initial condition $C_2 = 1$

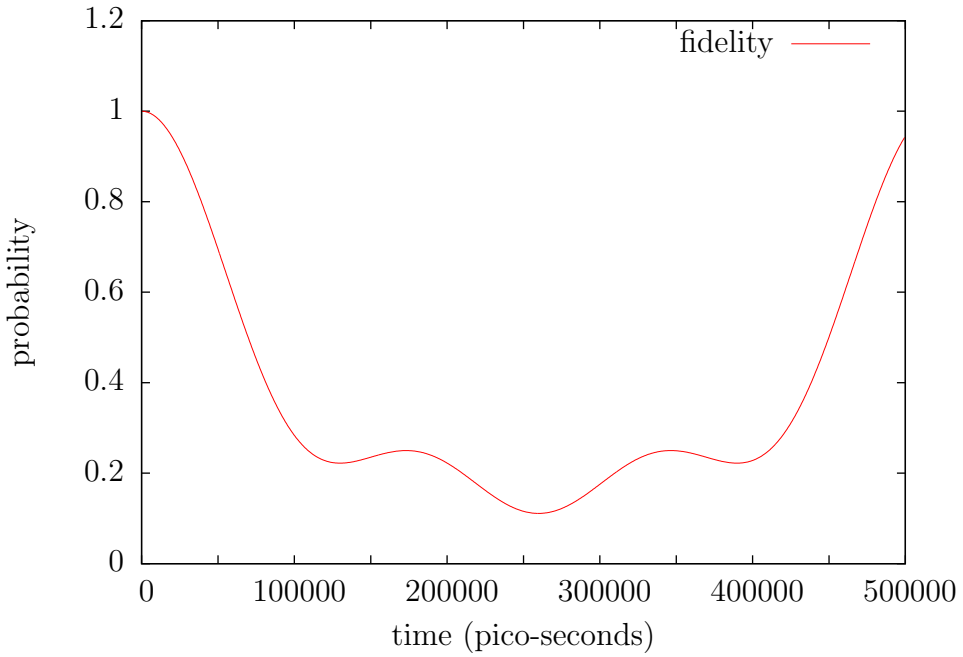

FIGURE 3.30: Graph of the fidelity of the electron hyperfine system with 2 nuclei and initial condition $C_2 = 1$

2	129925.90193604678
3	129925.90193604678

TABLE 3.2: Table of the output of the timescale routine for 1 nuclei

3.2.4 Timescales of interaction

From the timescale subroutine the results in table (3.2) were obtained from 1 nuclei case.

These values are sufficently close to the value predicted by equation (2.40) for self consistency to be likely.

3.2.5 Spin distribution

In figures (3.31) and (3.32), we observe the effects of the modulations of the electron on the system. For the first modulation, with the nuclei located at the standard deviation of the gaussian, we find that there is little change to system other than a slow timescale of interaction due to the effective hyperfine interaction being around half that of the unmodulated case.

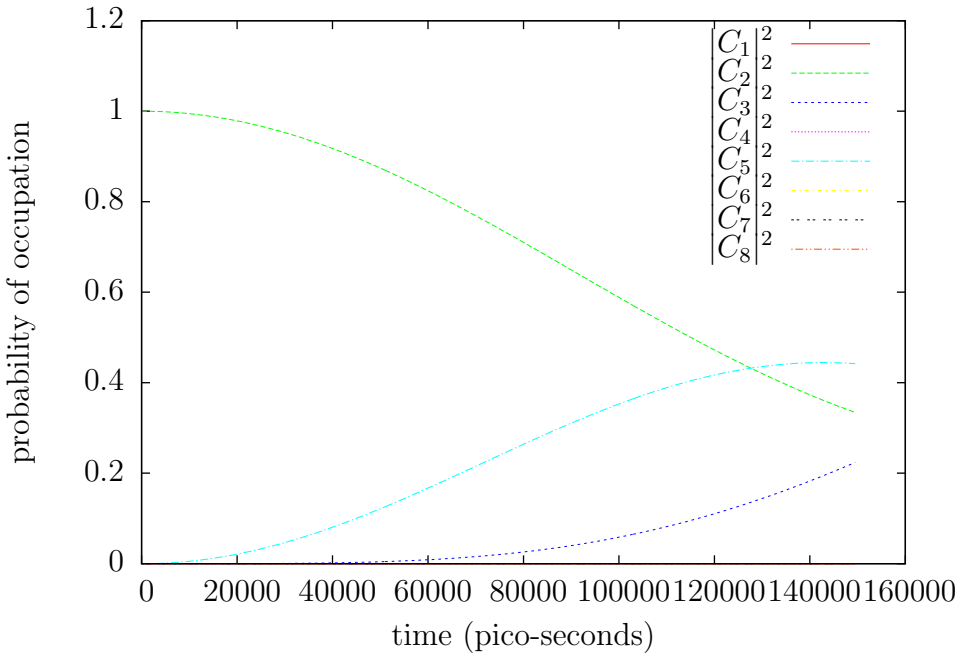

FIGURE 3.31: Graph of the time evolution of the electron hyperfine system with 1 nuclei and initial condition $C_2 = 1$ under the first modulation

However, for the second modulation in figure (3.32) it was found that having the nuclei at positions smaller than the standard deviation resulted in other spin states than the three previously found to be associated with the system. This is almost certainly an error though as the sum of probabilities is above one, so future distributions of spin will need to be examined more carefully.

3.3 Magnetic

Next the external magnetic field was introduced to the simulation and various strengths and orientations were tested on the system.

Figures (3.33) and (3.34) illustrate the effect of a 0.001 telsa field in the positive and negative z axis, with the positive z direction defined as being in the same direction as the up spin vector. The effect of this field appears to be identical in both directions however, which was odd because it was expected that effect of the magnetic field would be around 3 orders of magntiude larger for the electron than the nuclei, so it seems intuitive that the vector of the spin of the electron being in the same direction as the magnetic field would be preferable.

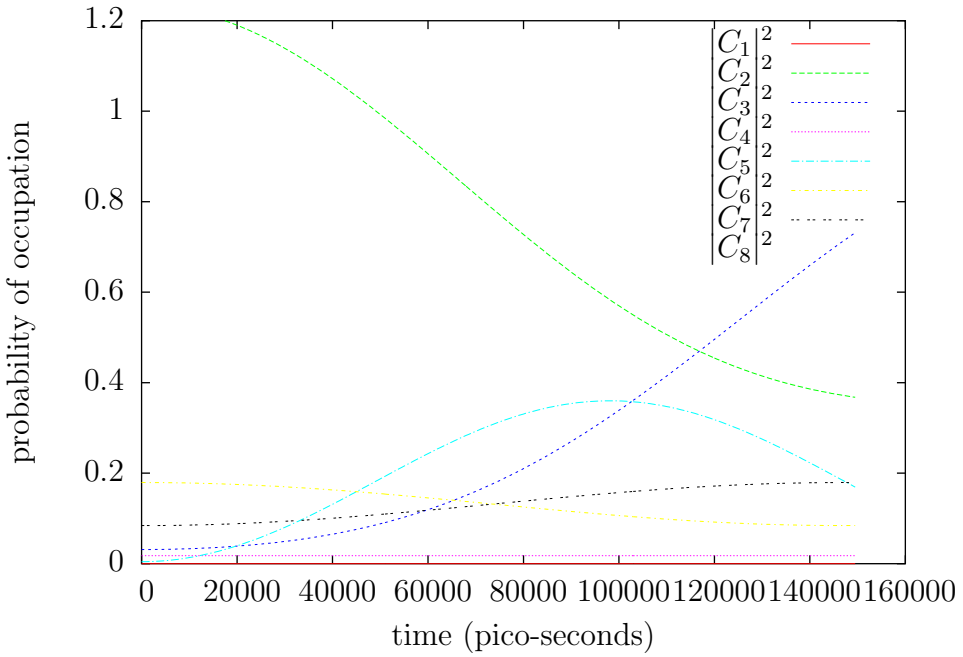

FIGURE 3.32: Graph of the time evolution of the electron hyperfine system with 2 nuclei and initial condition $C_2 = 1$ under the second modulation

However, it instead appears that the magnetic field simply reduces the likelyhood of the hyperfine interaction in the one nuclei case, causing the spin state of the system to be more likely to remain in the state it is initialised in.

However, with the same field in the two nuclei system the results become far more interesting as shown in figures (3.35) and (3.36). Here we find that the direction of the magnetic field causes the probability density to shift compared to the non-magnetic case show in figure (3.26) similar to the probability density shift of the spin half particle in a harmonic oscillator potential in Alistair Rae's *Quantum Mechanics* [15]. This shift appears to process forwards or backwards in time depending on the orientation of the magnetic field, with the positive and negative fields corresponding respectively.

It was also thought that when three spins were considered there would be a preference for states which contained a majority of spins that aligned with the magnetic field. However, no evidence for this behaviour was found in the results obtained.

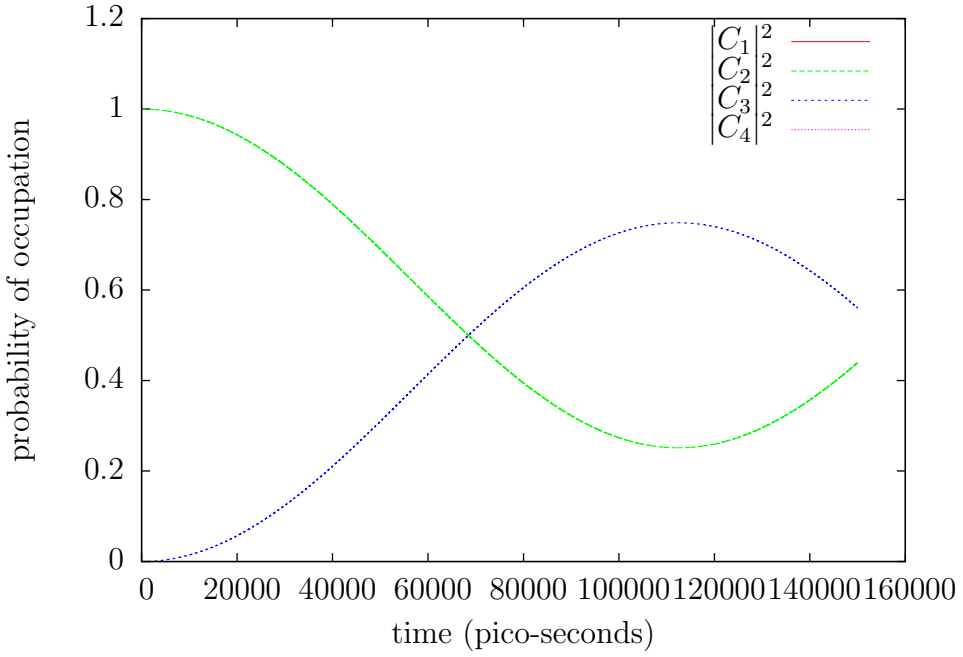

FIGURE 3.33: Graph the time evolution of the probabilities of the electron hyperfine system with 1 nuclei, initial condition $C_2 = 1$ and a $0.001T$ magnetic field

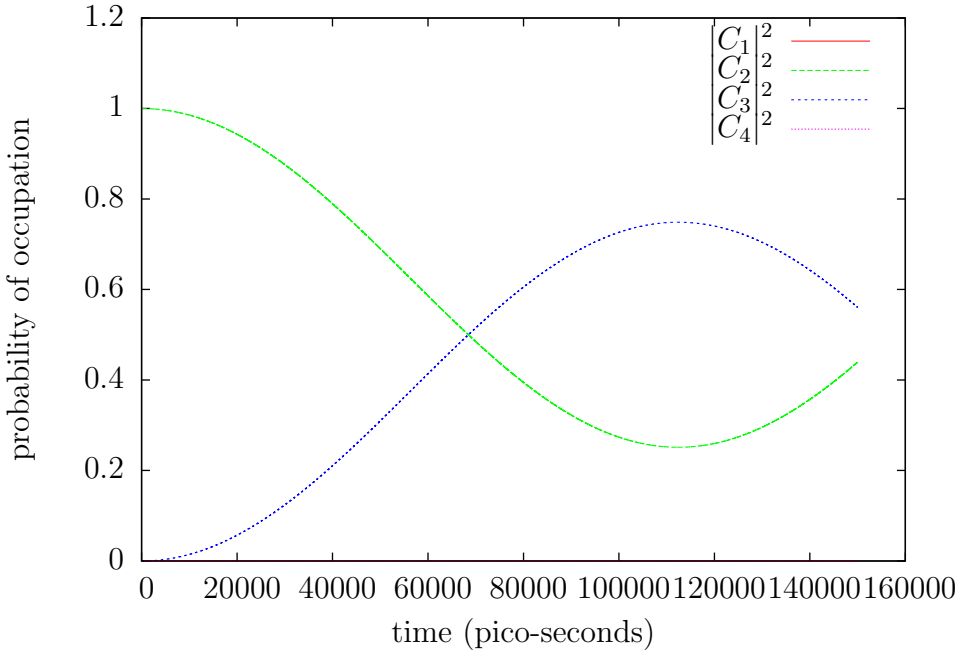

FIGURE 3.34: Graph the time evolution of the probabilities of the electron hyperfine system with 1 nuclei, initial condition $C_2 = 1$ and a $-0.001T$ magnetic field

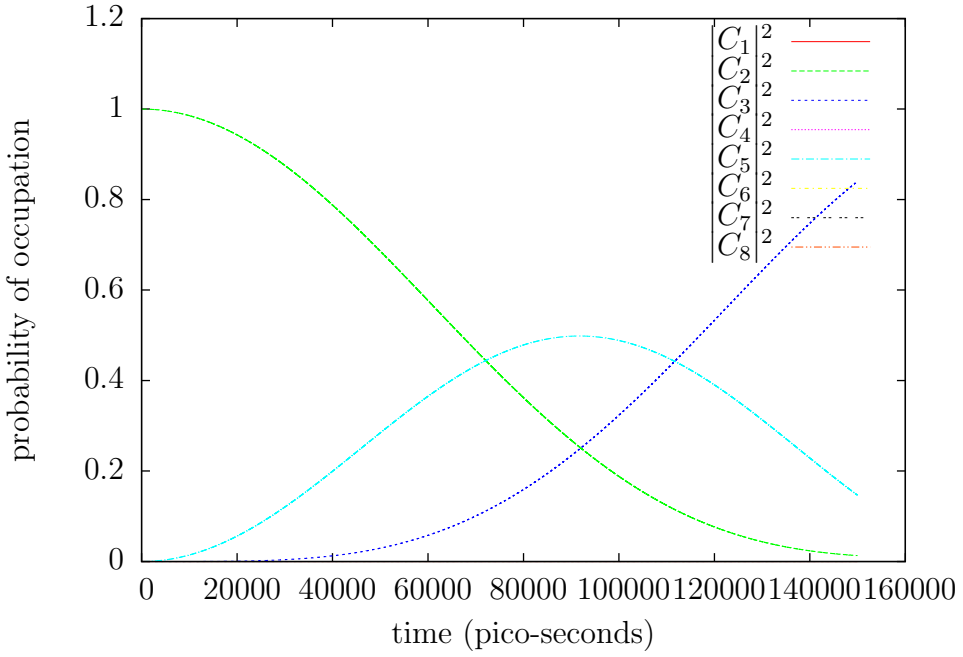

FIGURE 3.35: Graph the time evolution of the probabilities of the electron hyperfine system with 2 nuclei, initial condition $C_2 = 1$ and a $0.001T$ magnetic field

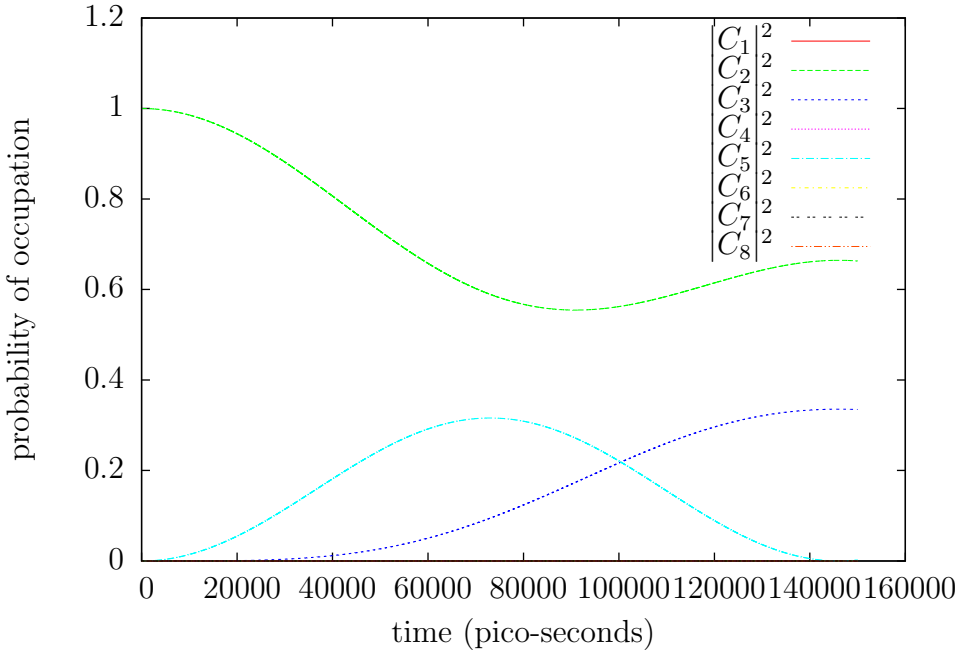

FIGURE 3.36: Graph the time evolution of the probabilities of the electron hyperfine system with 2 nuclei, initial condition $C_2 = 1$ and a $-0.001T$ magnetic field

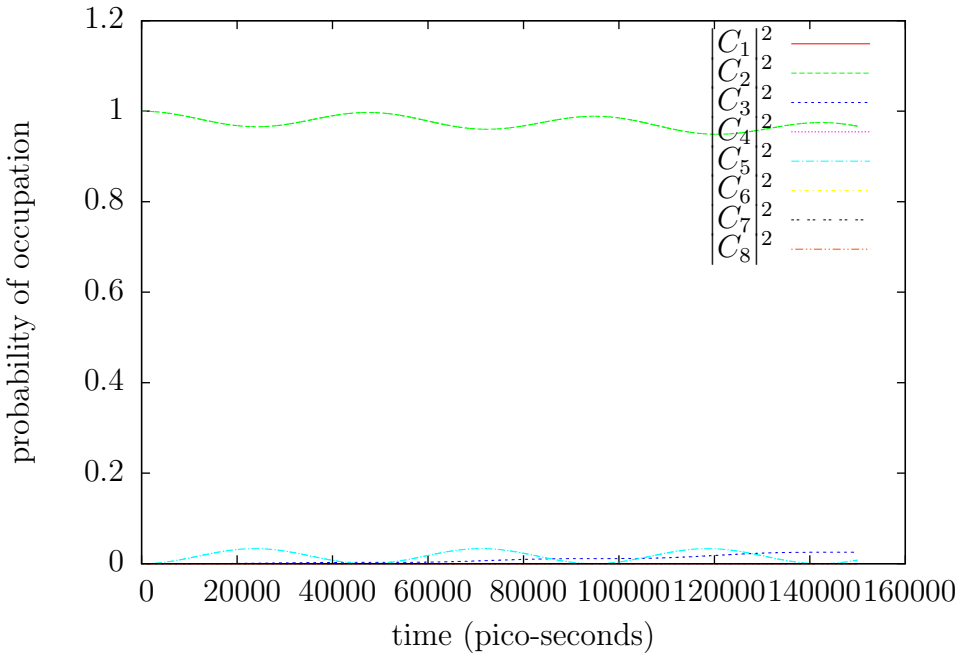

FIGURE 3.37: Graph the time evolution of the probabilities of the electron hyperfine system with 2 nuclei, initial condition $C_2 = 1$ and a $0.01T$ magnetic field

3.3.1 Magnetic field strength

Next several larger magnetic field strengths were tested as it was expected that the magnetic field interaction would start to dominate the system at around $0.01T$ and this was indeed found to be the case as shown in figures (3.37) and (3.38). This dominance allowed us to attempt to measure the timescale for the magnetic field coupling and check it against expectation.

3.3.2 Timescales of interaction

Table (3.3) displays the output of the timescale routine of the 2 nuclei system with a $0.01T$ magnetic field. Unlike the hyperfine results from table (3.2), there is a wider range of values due to the larger number of nuclei and the interplay of multiple interactions. However, it was found that the typical length of the timescale of interaction was around 4×10^5, agreeing the predictions of equation (2.44).

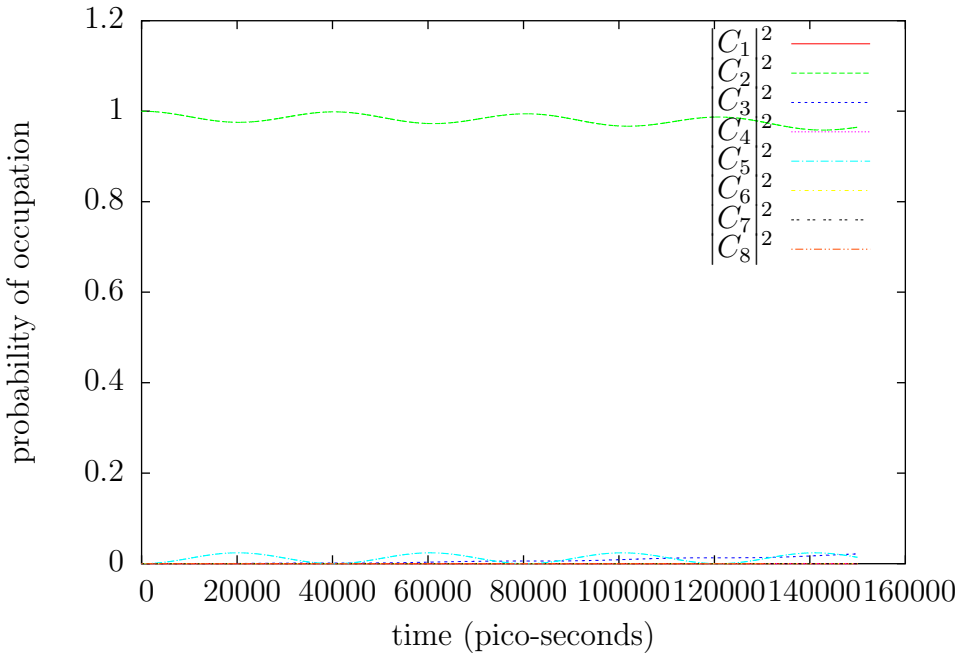

FIGURE 3.38: Graph the time evolution of the probabilities of the electron hyperfine system with 2 nuclei, initial condition $C_2 = 1$ and a $-0.01T$ magnetic field

3.3.3 Effect on Fidelity

Similarly to the non-magnetic case, for pure initial conditions the fidelity is simply the probability of the initial basis state, however the effect of the magnetic field on this probability is of interest. The same shift of probability density occurs in the fidelity as seen in figure (3.30) which is to be expected, but when the magnetic field starts to become an order of magnitude larger than hyperfine interaction we find that the fidelity is larger for a longer period of time, thus it seems that the magnetic field increases the fidelity of the system.

3.4 Non-iterative

The non-iterative method was successfully implemented into the program, however not much use was made of it due to it's late inclusion. The method allowed for much faster computation of the time evolution of the system with far less error and it was verified that the output matched that of the iterative method through comparison of results. Figure (3.41) shows how the routine was used to generate a coarse version of the results by skipping timesteps that would have to be calculated

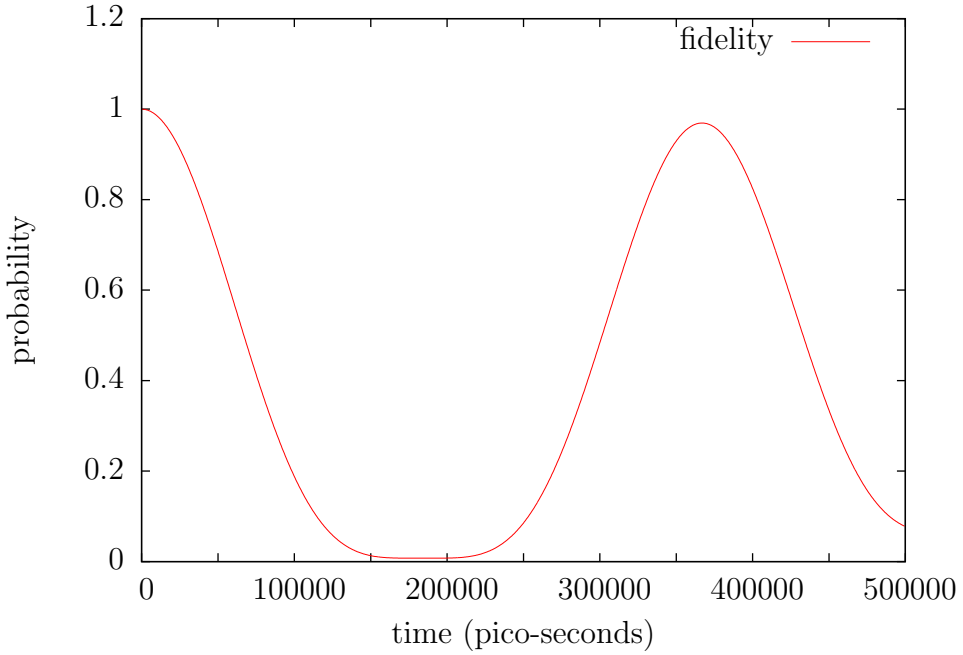

FIGURE 3.39: Graph of the fidelity of the electron hyperfine system with 1 nuclei and initial condition $C_2 = 1$ a $0.001T$ magnetic field

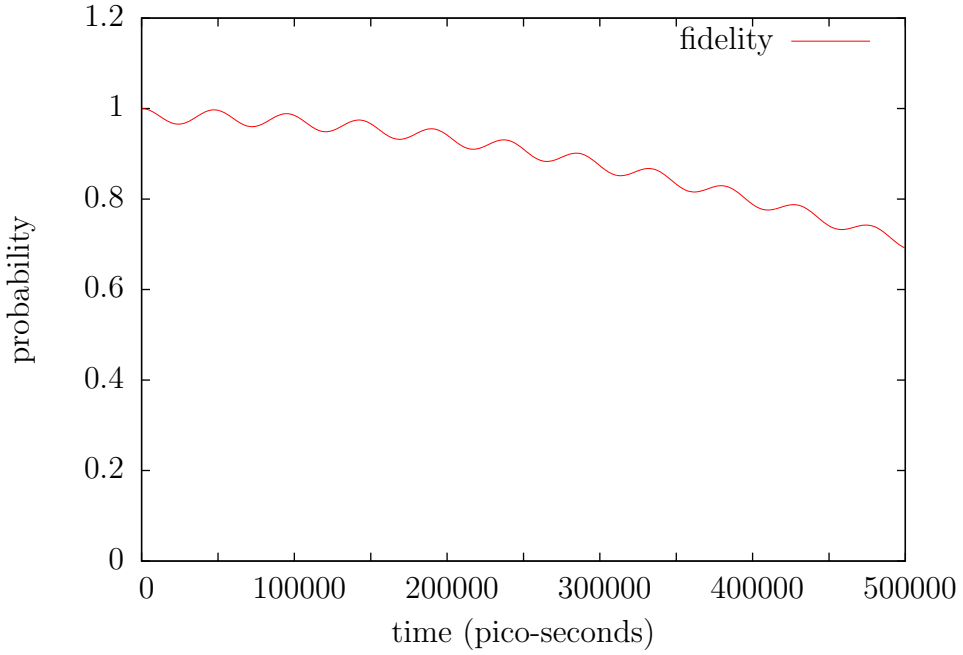

FIGURE 3.40: Graph of the fidelity of the electron hyperfine system with 2 nuclei and initial condition $C_2 = 1$ and a $0.01T$ magnetic field

2	6.9000001028180122
5	20142.800300151110
2	20384.200303748250
1	37806.157716045454
1	40284.000600278378
2	40285.400600299239
3	40293.700600422919
5	40285.300600297749
2	40781.300607688725
1	37803.600563317537
1	40286.400600314140
5	40285.500600300729
3	40279.600600212812
2	40285.500600300729
3	40776.100607611239
5	40285.300600297749
2	40781.400607690215
1	37803.600563317537
3	40279.800600215793
5	40285.300600297749
1	40286.200600311160
2	40285.400600299239
3	40776.000607609749
5	40285.200600296259
2	40781.400607690215

TABLE 3.3: Table of output values for timescale routine for 2 nuclei system with initial condition $C_2 = 1$ and a $0.01T$ magnetic field

by the iterative method. This method proved to be both more accurate and faster than the iterative method and would certainly be used in any future attempts at simulation of this system.

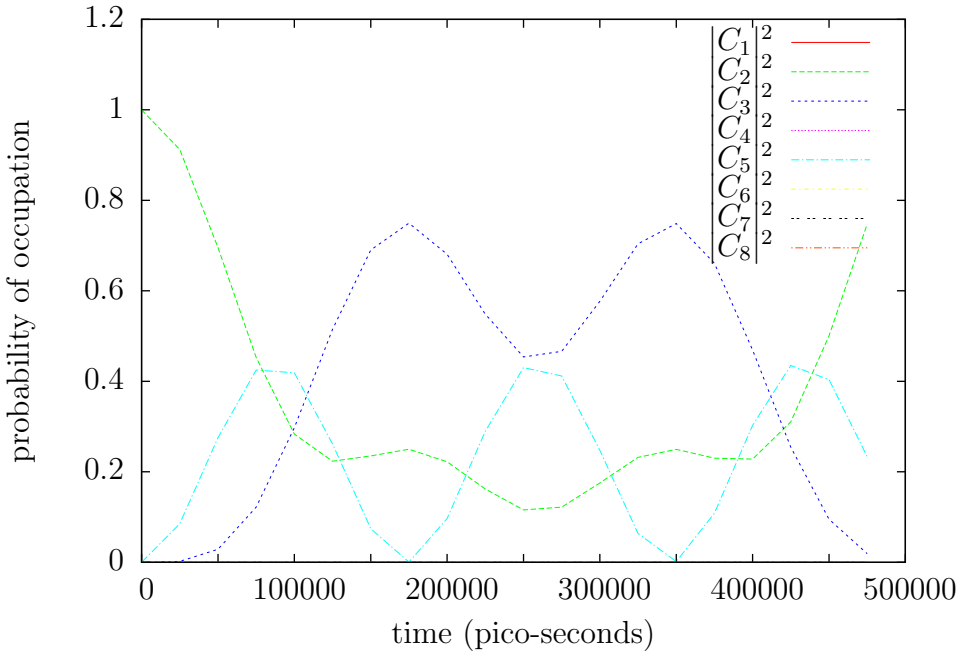

FIGURE 3.41: Coarse, non-iterative graph the time evolution of the probabilities of the electron hyperfine system with 2 nuclei and initial condition $C_2 = 1$

Chapter 4

Conclusions

4.1 Spin Chain

The spin chain was found to exhibit an oscillation of probabilities through time evolution due to the coupling of the spins. This oscillation was found to always be periodic when the system consisted of only three spins and to only be periodic for larger systems when certain conditions about the coupling constant or the intial conditions were met. Systems without these conditions were chaotic for the time period investigated, although some periodic behaviour is still hypothetically possible on a larger timescale than the time period simulated.

Error in the program was minimal, and seemed to be mostly related to the periodic behaviour of the probabilities, validating the iterative approach taken in terms of accuracy.

4.2 Hyperfine Interaction

The fermi contact hyperfine interaction between an electron and nuclei was found to be similarly periodic to the case of the simple spin chain, with probabilities only oscillating between basis states that shared an identical number of spins. This interaction occured on a similar timescale to the one predicted and the error due to the method of iteration was minimal.

The key difference between the hyperfine interaction model and the spin chain was the presence of a stationary state in the information basis of the hyperfine system. In the simple spin chain, no states were stationary, but in the hyperfine model at least two basis states were always eigenvectors. This is probably because the hyperfine basis set allowed all of the spins to be in the same state.

The modulation of the electron seemed to work when the nuclei were distributed on the positive and negative standard deviations of the gaussian, causing the timescale of interaction to increase as would be expected. However, this broke down for nuclei within the standard deviation of the gaussian for unknown reasons.

4.3 Magnetic Field

When the magnetic field was introduced it was found that a probability density shift occured in a direction in time depending on the orientation of the magnetic field. It was also found that the magnetic field seemed to make the initial state of the system more likely to be preserved if this state was a pure state as demonstrated through the measurement of fidelity and it would have been interesting to check if this was the case for mixed states.

The magnetic field was also found to operate on a timescale similar to that which was expected and also started to dominate when the field became stronger than the hyperfine interaction.

4.4 Decoherence

In terms of decoherence, the investigation into this concept beyond looking at the fidelity in the project was minimal. However, systems that did oscillate in time were certainly displaying dephasing behaviour and it would have been interesting to attempt to measure this more concisely and compare it to the values of typical decoherence time for quantum dots in the literature.

It would have been interesting to investigate this relationship in terms of both the number of nuclei in the system and the magnitude of the hyperfine coupling constant. Unfortunately this did not occur due to time constraints.

4.5 Further Directions

Some further directions this project could have been taken in would be to include the hyperfine interactions between the nuclei as for larger numbers of nuclei, only considering the fermi contact hyperfine interaction is less accurate.

Similarly, it would have been interesting to investigate the interactions with the nuclei of two electrons modulated so that they also interacted with each other. This approach would bring us closer to simulation of a 2-qubit gate, but this would require a great deal of work.

It would also be interesting to simulate the current system for larger time periods and with a greater number of nuclei, but this would require a greater amount of computational resources and a way of collating the results from the large amounts of data generated.

Appendix A

Resources Used

The code was compiled using several flags displayed in table (A.2) using a make file. The program was structured into modules such that it could be recompiled depending upon the needs of the simulation and to maximise performance. The program was run on the computer with specifications listed in table (A.1).

OpenMP was employed in the summations for each loop to make use of the laptop's dual core processor.

LAPACK was used in the diagonalisation of matrices to find the eigenvalues of the hamiltonians.

Data was plotted using gnuplot with a plotting script written for each basis set.

Processor	2x Intel(R) Pentium(R) CPU B980 @ 2.40GHz
Memory	3942MB
Operating System	Linux Mint 13 Maya

Table A.1: Computer used in simulation

Library	Compiler flag
BLAS	-lblas
LAPACK	-llapack
OpenMP	-fopenmp

Table A.2: Compiler flags used

Appendix B

Code

```
program simple_spinchain
implicit none

integer, parameter :: dp=selected_real_kind(15,300)

!Set up integers
integer :: i, j !loop integers
integer :: t !time integers

!Set up complex values
complex(kind=dp) :: sum_C_H
real(kind=dp) :: Hij, sum_C_abs_sqrd
real(kind=dp), dimension(:,:), Allocatable :: H, basis
real(kind=dp), dimension(:), Allocatable :: Hi
complex(kind=dp), dimension(:), Allocatable :: C

!define variables
integer :: N = 5
integer :: t_max = 10000
real(kind=dp) :: A = 0.04_dp
complex(kind=dp) :: k= (0.0_dp, 1.0E-2_dp)

!Open output file
open(100, file="C_output.dat")
open(200, file="sum_C_v_time.dat")

!allocate arrays
Allocate(C(1:N))
Allocate(H(1:N, 1:N))
```

```
Allocate(basis(1:N, 1:N))
Allocate(Hi(1:N))

H( 1, 2) = A*sqrt(real(1*(N-1)))
H( 2, 3) = A*sqrt(real(2*(N-2)))
H( 2, 1) = A*sqrt(real(1*(N-1)))
H( 3, 2) = A*sqrt(real(2*(N-2)))
H( 4, 3) = A*sqrt(real(3*(N-3)))
H( 3, 4) = A*sqrt(real(3*(N-3)))
H( 5, 4) = A*sqrt(real(4*(N-4)))
H( 4, 5) = A*sqrt(real(4*(N-4)))


  !Set up initial values - no normalisation required
  C(1) = 1.0_dp

do i=1, N
    !C(i) = 1.0_dp / SQRT(REAL(N,kind=dp))
end do

!Set up basis values
basis(1, 1:5) = (/1.0_dp,0.0_dp,0.0_dp,0.0_dp,0.0_dp/)
basis(2, 1:5) = (/0.0_dp,1.0_dp,0.0_dp,0.0_dp,0.0_dp/)
basis(3, 1:5) = (/0.0_dp,0.0_dp,1.0_dp,0.0_dp,0.0_dp/)
basis(4, 1:5) = (/0.0_dp,0.0_dp,0.0_dp,1.0_dp,0.0_dp/)
basis(5, 1:5) = (/0.0_dp,0.0_dp,0.0_dp,0.0_dp,1.0_dp/)

!compute
!iterate over time from t=1 to t_max
do t=1, t_max

  !iterate j from 1 to N
  do j=1,N

    sum_C_H = 0.0_dp

    !iterate i from 1 to N
    do i=1,N

      !call for complex matrix vector multiply
      Hi = MATMUL(H, basis(i, 1:N))

      !call for complex dot product
      Hij = DOT_PRODUCT(basis(j, 1:N), Hi)
```

```
      sum_C_H = sum_C_H + C(i) * Hij

    end do

      !Find C(j)
      C(j) = C(j) - k*sum_C_H

  end do

  !write to output file
  write(100, *) (abs(C(i))**2.0_dp, i = 1, N), t
  sum_C_abs_sqrd = 0.0_dp
  do i=1, N
    sum_C_abs_sqrd = sum_C_abs_sqrd + abs(C(i))**2.0_dp
  end do
  write(200, *) sum_C_abs_sqrd, t

end do

close(100)
close(200)

!deallocate arrays
deallocate(C)
deallocate(basis)
deallocate(H)
deallocate(Hi)



end program simple_spinchain
```

Listing B.1: Spin Chain Program

```
!*****************************************
!
!Performs Kronecker product of two matrices A and B producing
    matrix P
!n and m correspond to the dimensions of A and x and y
   correspond to the dimensions of B
!dimensions of P are assumed to be n*x and m*y
!
```

```
!*****************************************
subroutine kronecker_product(A, B, P, n, m, x, y)
  implicit none
  integer, parameter :: dp=selected_real_kind(15,300)
  integer :: i, j, k, l
  integer :: n, m, x, y
  complex(kind=dp), dimension(n, m) :: A
  complex(kind=dp), dimension(x, y) :: B
  complex(kind=dp), dimension(x*n, y*m) :: P

  do i = 1, n
    do j = 1, m

      do k = 1, x
  do l = 1, y

    P((i-1)*x + k, (j-1)*y + l) = A(i, j) * B(k, l)

  end do
      end do

    end do
  end do

  return

end subroutine kronecker_product
```

Listing B.2: Kronecker Product Subroutine

```
  !*****************************************
  !Generates a hamiltonian for the hyperfine interaction from
    pauli matrices
  !*****************************************
  subroutine hyperfine_hamiltonian
    implicit none
    integer :: i, j, k
    complex(kind=dp), dimension(3,2,2) :: pauli
    real(kind=dp), dimension(2,2) :: identity
    integer, dimension(1:N,1:N) :: diag_matrix
    complex(kind=dp), dimension(:,:), Allocatable :: A
    complex(kind=dp), dimension(2,2) :: B
    complex(kind=dp), dimension(:,:), Allocatable :: P
```

```
    pauli(1, 1, 2) = 1.0_dp   !Pauli x
    pauli(1, 1, 1) = 0.0_dp   !Pauli x
    pauli(1, 2, 2) = 0.0_dp   !Pauli x
    pauli(1, 2, 1) = 1.0_dp   !Pauli x

    pauli(2, 1, 1) = 1.0_dp   !Pauli z
    pauli(2, 1, 2) = 0.0_dp   !Pauli z
    pauli(2, 2, 1) = 0.0_dp   !Pauli z
    pauli(2, 2, 2) = -1.0_dp    !Pauli z

    pauli(3, 1, 2) = (0.0_dp, -1.0_dp) !Pauli y
    pauli(3, 1, 1) = 0.0_dp   !Pauli y
    pauli(3, 2, 2) = 0.0_dp   !Pauli y
    pauli(3, 2, 1) = (0.0_dp, 1.0_dp)  !Pauli y

    identity(1, 1) = 1.0_dp   !Identity matrix
    identity(2, 2) = 1.0_dp   !Identity matrix
    identity(1, 2) = 0.0_dp   !Identity matrix
    identity(2, 1) = 0.0_dp   !Identity matrix

    !Creates a diagonal matrix for permuations of spin
   matrices
    do i = 1, N
      diag_matrix(i, i) = 1
    end do

    call generate_positions(1)

    !k signifies each of the cartesian dimensions
    !x = 1
    !z = 2
    !y = 3
    do k = 1, 3

      !Loops over number of possible combinations of identity
    and pauli matrices
      do i = 1, N

  !Allocates A input matrix for kronecker_product first
   iteration
  Allocate(A(1:2, 1:2))
  if (ierr/=0) stop 'Error in allocating matrix initial A'
```

```
!Allocate matrix result for kronecker product so it set as
 first pauli matrix
!and then can be deallocated in loop
Allocate(P(1:2, 1:2))
if (ierr/=0) stop 'Error in allocating matrix initial P'

!Sets up the first A input matrix as the k dimension pauli
 matrix
P = 0.5 * pauli(k, 1:2, 1:2)

!Loops over each matrix to be multiplied for this iteration
!Each j is a possible interaction pair
do j = 1, N

  !Sets previous product as next input
  A = P

  !Deallocates product matrix so it can be reshaped
  Deallocate(P)
  if (ierr/=0) stop 'Error in deallocating matrix P in loop
 '

  !Finds the next matrix product based on diagonlised
 matrix
  if(diag_matrix(i, j) == 1) then
    !Sets the B input matrix as the kth dimensional pauli
 matrix
    B(1:2, 1:2) = 0.5 * pauli(k, 1:2, 1:2) * exp(- ((
 x_position(i))**2) / 2)
  else
    !Sets the B input matrix as the identity matrix (non-
 interaction)
    B(1:2, 1:2) = identity(1:2, 1:2)
  end if

  !Allocates P matrix to the size of size of the next
 k_product matrix
  Allocate(P(1:2**(j+1), 1:2**(j+1)))

  !Calls the tensor product for A and B producing P
  !Dimensions of A are determined by place in j loop
  call kronecker_product(A, B, P, 2**j, 2**j, 2, 2)

  !Deallocates input matrix A to prepare for next loop
```

```
    Deallocate(A)
    if (ierr/=0) stop 'Error in deallocating matrix A in loop
   '

    !Allocates input matrix A in next size to prepare for
   next loop
    Allocate(A(1:2**(j+1), 1:2**(j+1)))
    if (ierr/=0) stop 'Error in allocating matrix A in loop'

  end do

  !Deallocates A for this interaction
  Deallocate(A)
  if (ierr/=0) stop 'Error in deallocating matrix A at end of
    loop'

  !Adds P to hamiltonian in matrix at each loop
  !Allows for additional hamiltonian elements to be added
   later
  H(1:QN,1:QN) = H(1:QN,1:QN) + P(1:QN, 1:QN)

  Deallocate(P)
  if (ierr/=0) stop 'Error in deallocating matrix P at end'

      end do
    end do

    !Multiplies all hamiltonian elements by hyperfine
   interaction constant
    H = interaction_constant * H

    !Writes Hamiltonian to file for debugging
    open(400, file="hamiltonian.dat")
    do i = 1, QN
      write(400, *) (H(i, j), j = 1, QN)
    end do
    close(400)


    return

  end subroutine hyperfine_hamiltonian
```

Listing B.3: Hyperfine Hamiltonian Subroutine

```
!*****************************************
!generates a magnetic hamiltonian from pauli matrices for a
  system of size N
!*****************************************
subroutine magnetic_hamiltonian
  implicit none
  integer :: i, j
  complex(kind=dp), dimension(2,2) :: pauli
  real(kind=dp), dimension(2,2) :: identity
  integer, dimension(1:N+1,1:N+1) :: diag_matrix
  complex(kind=dp), dimension(:,:), Allocatable :: A
  complex(kind=dp), dimension(2,2) :: B
  complex(kind=dp), dimension(:,:), Allocatable :: P

  pauli(1, 1) = 1.0_dp    !Pauli z
  pauli(1, 2) = 0.0_dp    !Pauli z
  pauli(2, 1) = 0.0_dp    !Pauli z
  pauli(2, 2) = -1.0_dp   !Pauli z

  identity(1, 1) = 1.0_dp   !Identity matrix
  identity(2, 2) = 1.0_dp   !Identity matrix
  identity(1, 2) = 0.0_dp   !Identity matrix
  identity(2, 1) = 0.0_dp   !Identity matrix

  !Creates a diagonal matrix for permuations of spin
 matrices
  do i = 1, N+1
    diag_matrix(i, i) = 1
  end do

    !Loops over number of possible combinations of identity
  and pauli matrices
    do i = 1, N + 1

!Allocates A input matrix for kronecker_product first
 iteration
Allocate(A(1:2, 1:2))
if (ierr/=0) stop 'Error in allocating matrix initial A'

!Allocate matrix result for kronecker product so it set as
 first pauli matrix
!and then can be deallocated in loop
Allocate(P(1:2, 1:2))
```

```
  if (ierr/=0) stop 'Error in allocating matrix initial P'

!Finds the next matrix product based on diagonlised matrix
  if(diag_matrix(i, 1) == 1) then
    !Sets the B input matrix as the kth dimensional pauli
 matrix
    P(1:2, 1:2) = 0.5_dp * ((bohr_magneton * g_electron)/
 electron_charge) * pauli(1:2, 1:2) * 1E6_dp
  else
    !Sets the B input matrix as the identity matrix (non-
 interaction)
    P(1:2, 1:2) = identity(1:2, 1:2)
  end if

!Loops over each matrix to be multiplied for this iteration
!Each j is a possible interaction pair
do j = 2, N + 1

  !Sets previous product as next input
  A = P

  !Deallocates product matrix so it can be reshaped
  Deallocate(P)
  if (ierr/=0) stop 'Error in deallocating matrix P in loop
 '

  !Finds the next matrix product based on diagonlised
 matrix
  if(diag_matrix(i, j) == 1) then
    !Sets the B input matrix as the kth dimensional pauli
 matrix
    B(1:2, 1:2) = 0.5_dp * ((n_mag_moment *
 nuclear_magneton)/electron_charge) * pauli(1:2, 1:2) * 1
 E6_dp
  else
    !Sets the B input matrix as the identity matrix (non-
 interaction)
    B(1:2, 1:2) = identity(1:2, 1:2)
  end if

  !Allocates P matrix to the size of size of the next
 k_product matrix
  Allocate(P(1:2**(j+1), 1:2**(j+1)))
```

```
    !Calls the tensor product for A and B producing P
    !Dimensions of A are determined by place in j loop
    call kronecker_product(A, B, P, 2**j, 2**j, 2, 2)

    !Deallocates input matrix A to prepare for next loop
    Deallocate(A)
    if (ierr/=0) stop 'Error in deallocating matrix A in loop
   '

    !Allocates input matrix A in next size to prepare for
   next loop
    Allocate(A(1:2**(j+1), 1:2**(j+1)))
    if (ierr/=0) stop 'Error in allocating matrix A in loop'

  end do

  !Deallocates A for this interaction
  Deallocate(A)
  if (ierr/=0) stop 'Error in deallocating matrix A at end of
    loop'

  !Adds P to hamiltonian in matrix at each loop
  !Allows for additional hamiltonian elements to be added
   later
  H(1:QN,1:QN) = H(1:QN,1:QN) + mag_field * P(1:QN, 1:QN)

  Deallocate(P)
  if (ierr/=0) stop 'Error in deallocating matrix P at end'

      end do


    !Writes Hamiltonian to file for debugging
    open(400, file="hamiltonian.dat")
    do i = 1, QN
      write(400, *) (H(i, j), j = 1, QN)
    end do
    close(400)

    return

  end subroutine magnetic_hamiltonian
```

Listing B.4: Magnetic Hamiltonian Subroutine

```
subroutine intergrate_TDSE
  implicit none

  !Library header for OpenMP
  !Required for single memory multiple processor operations
  !Requires compiler flag -fopenmp in gnu compilers to
 enable
  include 'omp_lib.h'

  !subroutine variables
  integer :: i, j                                    !Loop
  integers
  complex(kind=dp) :: sum_C_H                        !Sum
 of Complex values multipled by Hij


  !iterate j from 1 to QN
  do j=1,QN

    !Set summation to zero to initial addition in next loop
    sum_C_H = 0.0_dp

    !$OMP parallel do private(i) shared(C, H, j) reduction
 (+:sum_C_H)
    !iterate i from 1 to QN
    !With OpenMP enabled this loop is performed on each
 thread seperately
    !Values of sum_C_H are added together from each thread
 at end of loop
    do i=1,QN

!Adds product of C(i) and Hij to summation
!This is processed individually on each thread then summed
 at end
!if openMP is enabled (-fopenmp)
sum_C_H = sum_C_H + C(i) * H(j, i)


    end do
    !$OMP end parallel do
    !ends the parallel do loop for openMP

    !print *, i, j, sum_C_H
```

```
    !Find C(j) by multiplying TDSE_constant with sum_C_H
    C(j) = C(j) - TDSE_constant * sum_C_H


  end do



end subroutine intergrate_TDSE
```

Listing B.5: Iterative Time Evolution Subroutine

```
subroutine find_wavefunction(time, D)
implicit none
real(kind=dp) :: time          !time double precision value
 fed from program in loop
integer :: i, j          !loop variables
complex(kind=dp), dimension(1:QN) :: D

!sets C to zero for summation
C(:) = 0.0_dp

!loops over each eigenvector and basis element to find
 complete wavefunction values at specified time
do i=1, QN
  do j=1, QN
    C(j) = C(j) + D(i) * exp((eigenvalues(i) * cmplx(0.0_dp
 , time, kind=dp))/hbar) * eigenvectors(j, i)
  end do
end do


end subroutine find_wavefunction

subroutine rotate_basis(D)
implicit none
integer :: i, j          !loop variables
complex(kind=dp), dimension(1:QN) :: initial_coeffs !
 complex initial coefficients of eigenvectors
complex(kind=dp), dimension(1:QN) :: D

!manually set up co-efficients for testing
```

```
  initial_coeffs(:) = 0.0_dp
  initial_coeffs(2) = 1.0_dp

  D(:) = 0.0_dp

  do i = 1, QN
    do j = 1, QN
      D(i) = D(i) + initial_coeffs(j) * conjg(eigenvectors(j,
    i))
    end do
  end do


  end subroutine rotate_basis
```

Listing B.6: Non-Iterative Time Evolution Subroutine

```
!*******************************************
! Uses LAPACK to find eigenstates of passed matrix using
   DGEEV functions
! DGEEV finds the eigenvectors and eigenvalues of general
   matrices
!
! Matrix is assumed to be size NxN with N passed into
   subroutine
!
!*******************************************
subroutine find_eigenstates(A, N, eigenvalues, eigenvectors)
  implicit none

  integer, parameter :: dp=selected_real_kind(15,300)

  integer :: i, j          !Loop integers
  integer :: N
  integer, parameter :: LIWORK = 256      !Size of the work
   array
  integer :: ierr

  !Variables for DGEEV function
  real(kind=dp), dimension(1:N) :: WR     !Work (real)
  real(kind=dp), dimension(1:N) :: WI     !Work (imaginary)
  real(kind=dp), dimension(1:N, 1:N) :: A   !Input matrix
  real(kind=dp), dimension(1:N,1:N) :: VL   !Left vector(
   unused)
```

```
real(kind=dp), dimension(1:N,1:N) :: VR   !Right vector (
 calculated)
integer, dimension(1:LIWORK) :: WORK      !Work array used
 in calculating

!Output variables
complex(kind=dp), dimension(1:N, 1:N) :: eigenvectors
complex(kind=dp), dimension(1:N)      :: eigenvalues



!calls the DGEEV function to calculate eigenstates for
 matrix A.
!
! A(i, j) * V(i) = W(i) * V(i)
!
! Where V is the eigenvector and W is the eigenvalue
!
call DGEEV( 'N', 'V', N, A, N, WR, WI, VL, 1, VR, N, WORK,
 LIWORK, ierr)

!Reports errors if DGEEV fails to process using intrinsic
 error codes in argument
!          = 0:  successful exit
!          < 0:  if INFO = -i, the i-th argument had an
 illegal value.
!          > 0:  if INFO = i, the QR algorithm failed to
 compute all the
!                eigenvalues, and no eigenvectors have been
  computed;
!                elements i+1:N of WR and WI contain
 eigenvalues which
!                have converged.
if (ierr/=0) then
  print *, 'Error in solving eigenstates of hamiltonian'
if (ierr<0) then
  print *, 'value', ierr, 'had illegal value'
else
  print *, 'dgeev failed to converge'
end if
  stop
end if
```

```
!Open file for saving of eigenstates
open(600, file='eigenstates.dat', iostat=ierr)

!Writes to file if there is no error in opening file
if (ierr==0) then
  !Loops over each eigenvector and eigenvalue pair
do j=1, N
!Writes only real to file if there is no imaginary part to
 eigenvalue
if(WI(j)==0.0_dp) then
  write(600,*) 'Eigenvalue', j, '=', WR(j)
  !Copies eigenvalue to output variable
  eigenvalues(j) = cmplx(WR(j), 0.0_dp, kind=dp)
!Writes both complex and real parts otherwise
else
  write(600,*) 'Eigenvalue', j, '=', WR(j), '+', WI(j), 'i'
  !Copies eigenvalue to output variable
  eigenvalues(j) = cmplx(WR(j), WI(j), kind=dp)
end if

write(600,*) !Blank line
write(600,*) 'Eigenvector', j,  '=' !formatting
write(600,*) !Blank line

!Writes real parts of eigenvectors if eigenvalue is real
if(WI(j)==0.0_dp) then
  do i=1, N
    write(600, *) VR(i, j)
    !Copies eigenvector to output variable
    eigenvectors(i, j) = cmplx(VR(i, j), 0.0_dp, kind=dp)
  end do
!Writes eigenvalue pair to file if eigenvalue is complex
 and greater than zero
elseif(WI(j)>=0.0_dp) then
  do i=1, N
    write(600,*) VR(i, j), '+', VR(i, j+1), 'i'
    !Copies eigenvector to output variable
    eigenvectors(i, j) = cmplx(VR(i, j), VR(i, j + 1), kind
 =dp)
  end do
!Writes eigenvalue pair to file if eigenvalue is complex
 and less than zero
!typically second part of pair
else
```

```
    do i=1, N
      write(600,*) VR(i, j-1), '+', VR(i, j), 'i'
100   !Copies eigenvector to output variables
      eigenvectors(i, j) = cmplx(VR(i, j - 1), VR(i, j), kind
    =dp)
    end do
   end if

105 write(600,*) !Blank line

      end do
    else
      !If error opening file, aborts program and prints error
    message
110   stop 'Error in opening file eigenstates.dat'
    end if

    !Closes file and reports error if failure
    close(600, iostat=ierr)
115 if (ierr/=0) stop 'Error in closing file eigenstates.dat'

end subroutine find_eigenstates
```

Listing B.7: Eigenstate Subroutine. Uses LAPACK routine DGEEV [16]

Bibliography

[1] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, Cambridge; New York, 2000. ISBN 0521632358 9780521632355 0521635039 9780521635035.

[2] File:Bloch sphere.svg. URL `http://en.wikipedia.org/wiki/File:Bloch_Sphere.svg`.

[3] CarnegieMellonU. Quantum computing and the limits of the efficiently computable - 2011 buhl lecture, June 2011. URL `http://www.youtube.com/watch?v=8bLXHvH9s1A&feature=youtube_gdata_player`.

[4] G. J. Milburn, S. Schneider, and D. F. V. James. Ion trap quantum computing with warm ions. *Fortschritte der Physik*, 48(9-11): 801810, 2000. URL `http://iontrap.umd.edu/publications/archive/JamesSchneiderMilburn2000.pdf`.

[5] J. L. O'Brien, G. J. Pryde, A. G. White, T. C. Ralph, and D. Branning. Demonstration of an all-optical quantum controlled-NOT gate. *Nature*, 426(6964):264–267, November 2003. ISSN 0028-0836. doi: 10.1038/nature02054. URL `http://www.nature.com/nature/journal/v426/n6964/abs/nature02054.html`.

[6] B. E. Kane. A silicon-based nuclear spin quantum computer. *Nature*, 393(6681):133138, 1998. URL `http://fy.chalmers.se/~delsing/QI/Kane-Nature-98.pdf`.

[7] D. P. DiVincenzo. Quantum computation. *Science*, 270(5234):255261, 1995. URL `http://lib.semi.ac.cn:8080/tsh/dzzy/wsqk/science/vol270/270-255.pdf`.

[8] Mang Feng, Irene DAmico, Paolo Zanardi, and Fausto Rossi. Spin-based quantum-information processing with semiconductor quantum dots

and cavity QED. *Physical Review A*, 67(1):014306, January 2003. doi: 10.1103/PhysRevA.67.014306. URL http://link.aps.org/doi/10.1103/PhysRevA.67.014306.

[9] Daniel Loss and David P. DiVincenzo. Quantum computation with quantum dots. *Physical Review A*, 57(1):120–126, January 1998. doi: 10.1103/PhysRevA.57.120. URL http://link.aps.org/doi/10.1103/PhysRevA.57.120.

[10] David Press, Thaddeus D. Ladd, Bingyang Zhang, and Yoshihisa Yamamoto. Complete quantum control of a single quantum dot spin using ultrafast optical pulses. *Nature*, 456(7219):218–221, November 2008. ISSN 0028-0836. doi: 10.1038/nature07530. URL http://www.nature.com/nature/journal/v456/n7219/full/nature07530.html.

[11] David P. DiVincenzo and IBM. The physical implementation of quantum computation. *arXiv:quant-ph/0002077*, February 2000. URL http://arxiv.org/abs/quant-ph/0002077.

[12] Jan Fischer, Mircea Trif, W.A. Coish, and Daniel Loss. Spin interactions, relaxation and decoherence in quantum dots. *Solid State Communications*, 149(3536):1443–1450, September 2009. ISSN 0038-1098. doi: 10.1016/j.ssc.2009.04.033. URL http://www.sciencedirect.com/science/article/pii/S0038109809002610.

[13] Daniel Joseph Klauser. *Hyperfine interaction and spin decoherence in quantum dots*. PhD thesis, University of Basel, 2008. URL http://edoc.unibas.ch/761/.

[14] Alexander Khaetskii, Daniel Loss, and Leonid Glazman. Electron spin decoherence in quantum dots due to interaction with nuclei. *Physical Review Letters*, 88(18), April 2002. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.88.186802. URL http://link.aps.org/doi/10.1103/PhysRevLett.88.186802.

[15] Alastair I. M Rae. *Quantum mechanics*. Taylor & Francis, New York, 2008. ISBN 9781584889700 1584889705.

[16] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen.

LAPACK Users' Guide. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8 (paperback).