
NUMERICAL METHODS FOR SIMULATING SPIN CHAIN DYNAMICS

C. DOHERTY, M. GADIOUX, M. GRAHAM, C. J. IRVINE

DUBLIN INSTITUTE FOR ADVANCED STUDIES,
SCHOOL OF THEORETICAL PHYSICS

JULY 2020

Abstract

We employ several numerical techniques to study the dynamics of many-body quantum systems. We do Monte Carlo simulations of the two-dimensional Ising and ten-state Potts models and find critical temperatures T_C at ~ 2.5 and ~ 0.72 respectively, for a coupling constant $J = 1$, and with periodic boundary conditions. The systems change from a (mostly) ferromagnetic state to a nearly unmagnetised state very quickly as the temperature is increased across T_C . We also perform quantum control experiments using gradient descent on a qubit acted upon by a time-dependent magnetic field. We find that the maximum fidelity F is almost linearly dependent on the amount of time T given to the qubit to evolve, up to $T \sim 1.4$ where F plateaus at 1. In addition, we study a system coupled to a generalised quantum XY spin chain, which we write in terms of the fermionic creation and annihilation operators to exploit the Gaussian formalism and benefit from much faster computational times. We examine the decoherence of the system and find a critical point in its behaviour when the environment's magnetic field has the same magnitude as its coupling constants' magnitude, provided the system is pure initially and the system Hamiltonian commutes with the total Hamiltonian. Furthermore, we find that the decoherence depends on the environment's temperature only when the system Hamiltonian does not commute with the total Hamiltonian, but we cannot explain why. Finite-size effects are seen to be proportional to the chain size but inversely proportional to the magnitude of the environment's couplings. Finally, we apply quantum control to this generalised quantum XY model. We find that to maximise the purity of the system for long times, the magnetic field strength should be much larger on the system than on the environment.

Contents

1	Introduction	2
2	Monte Carlo Simulations	2
2.1	The Monte Carlo Approach	2
2.2	Markov Chain Monte Carlo (MCMC)	3
2.2.1	Hybrid Monte Carlo (HMC)	3
2.3	An Application of the Monte Carlo Method: The Ising Model	4
2.3.1	The Algorithm	5
2.3.2	Results & Data Analysis	5
2.3.3	The Potts Model	6
3	Machine Learning for Quantum Control	7
3.1	Machine Learning and its use for Physics	7
3.1.1	Supervised, unsupervised and reinforcement learning	7
3.1.2	Uses in Physics	8
3.2	Quantum Control	8
3.3	Determining magnetic field to control a qubit state	9
4	Fermionic Representation of Spin Chains	11
4.1	Introduction	11
4.2	Theoretical Framework	12
4.2.1	Model	12
4.2.2	Gaussian Fermionic Systems	13
4.2.3	Jordan-Wigner Transformation	13
4.3	Results: Decoherence of a Single-Spin System	14
4.3.1	Decoherence	14
4.3.2	Critical Points in Decoherence Behaviour	15
4.3.3	Temperature Dependence and Finite-Size Effects	17
4.3.4	Further Research	19
4.4	Machine learning for control of decoherence	19
5	Conclusion	20
6	Acknowledgements	21
A	Fermionic Representation	23
A.1	Schrödinger-type Equation for the Covariance Matrix	23
A.2	Derivation of Eq. 4.17	23
A.3	Matrices	24
A.4	Pure States	24

1 Introduction

It is well-known that quantum many-body problems are much more difficult to solve than classical ones (e.g. [1]). Therefore, it is usually very difficult or even impossible to obtain analytic solutions, and we are often forced to employ numerical and computational methods. Unfortunately, it is in the nature of quantum mechanics that the amount of information needed to describe a physical system increases exponentially with the number of particles [1]. As a result, brute-force computational methods quickly become unfeasible even with relatively low numbers of particles, and we must resort to smarter techniques.

Monte Carlo simulations and their variations are widely used in computational physics, and also have found their place in many-body quantum problems. Their great advantage is the ability to efficiently scan the phase-space of the problem at hand, while also identifying essential features such as potential wells. This method has been popular in quantum physics for several decades, and has been used to calculate properties of many-electron systems [2], to calculate the ground and some excited states of light nuclei [3] and to study fermionic systems [4]. Monte Carlo simulations have also been used to identify phase transitions, e.g. [5].

Machine learning as a topic has seen an explosive growth as its own field of research and as a tool to aid the research of the physical sciences over the last decade [6–8]. It lends itself particularly well to the field of quantum physics; demonstrating ability to find solutions to notoriously difficult systems in many-body physics [8] while also being particularly adept at helping us classify, analyse and control quantum systems [9, 10]. It is this latter approach that we focus on in this paper, motivated by previous success at controlling quantum states [10] to build a simpler, but no less effective tool of control via a different discipline of machine learning [11]. We look to the heights of applicability also, realising that the landscape today necessitates any simulation of quantum systems to push beyond what is conventionally available for classical simulation [12]. This development will allow for the application of machine learning to give us insight into the control of real quantum systems in use today.

Other methods consist in looking at systems from a specific class which can be efficiently computed. This is often done by restricting the Hamiltonian to a particular form where all of the information can be compressed into a quantity of lower computational complexity. A common example in the literature is to consider spin chains whose Hamiltonian can be written in quadratic terms of the fermionic creation and annihilation operators, as that allows the computational complexity to be reduced to a polynomial function of the system size, e.g. [13–15].

We focus on the application of numerical methods to quantum physics; in particular, we look at numerical models of spin chain dynamics. We use Monte Carlo and Hybrid Monte Carlo methods, in particular the Metropolis algorithm, to study the Ising and Potts models in Sec. 2. We use machine learning to investigate optimal strategies in quantum control in Sec. 3. Finally, we numerically model the decoherence of a spin-1/2 particle coupled to a quantum XY spin chain in Sec. 4.

2 Monte Carlo Simulations

2.1 The Monte Carlo Approach

The Monte Carlo Method is a computational algorithm that relies on repeated random sampling to obtain numerical results. It is an approximative method and can be used to evaluate complicated multi-dimensional integrals that would otherwise be impossible to solve analytically. It is an important model to consider in research as it is built from relatively simple concepts and can be applied to a wide range of problems, varying from numerical integration and optimisation, to system simulation for use in quantum spin chain analysis, as well as having applications in other areas such as statistics and finance.

There are several different Monte Carlo methods, but all of them follow the same basic principles: (i) the domain of possible inputs is established, (ii) inputs are randomly generated from a specified probability distribution over the domain, and (iii) the inputs are rejected or accepted dependent on a predetermined computation and the results are compiled.

Thus, down to the most basic concept, the Monte Carlo method is an algorithm for sampling from a probability distribution, using randomness to solve problems that would otherwise require more rigorous approaches. It employs the techniques of importance sampling, as it is a method of estimating properties of a particular distribution, while only using samples generated from a different distribution. The most general

form of this method uses randomly generated points taken from a uniform distribution, and can be used for something as simple as to estimate the value of pi. This method can also be applied to problems where it is necessary to average over a large number of different configurations, each configuration with the probability

$$P(x_i) = \frac{1}{Z} e^{-S(x_i)} \quad (2.1)$$

where Z is the partition function, and $S(x_i)$ is the action of the system, with a set number of n degrees of freedom associated with it, $i = 1, \dots, n$. The action, $S(x_i)$, describes the overall motion of the system and gives insight into how energy-costly each configuration is.

2.2 Markov Chain Monte Carlo (MCMC)

Taking this simple method one step further, for a more accurate model of the underlying probability distribution, a Markov chain can be produced with the same distribution as that being studied. A Markov chain is a random process describing a sequence of possible events in which the probability of each event is independent of the previous one. The Metropolis-Hastings algorithm was employed in this paper to achieve this effect — generating a Markov chain using a proposed density for new steps along with a test to control which moves are accepted. The first configuration, x_0 , is randomly generated, stored, and adjusted according to a predetermined step size, ϵ . The Metropolis test is then carried out to check whether this new move can be accepted according to the acceptance ratio, ΔS — the difference in the action between the i^{th} step and the $(i+1)^{\text{th}}$ step:

$$\Delta S = S(x_{i+1}) - S(x_i) \quad (2.2)$$

A uniform, random number t is generated, $t \in [0,1]$, and a decision is made depending on the following outcome:

- If $e^{-\Delta S} \leq t$, the new configuration is accepted and becomes the new initial value for the next step, $x_{i+1} = x_i$.
- If $e^{-\Delta S} > t$, the new configuration is rejected and replaced with the previous one.

After the number of iterations required, a Markov chain is obtained consisting of a list of all of the accepted configurations, and thus, a sample of the desired distribution can be obtained by recording the states in this chain, allowing for the easy computation of observables. The higher the number of iterations taken, the more accurately the resulting distribution will describe that of the target distribution. This MCMC method is particularly useful as it is a way to obtain a sequence of random samples from a probability distribution from which direct sampling could be otherwise difficult.

2.2.1 Hybrid Monte Carlo (HMC)

The Hybrid Monte Carlo method is a type of MCMC that also uses the Metropolis algorithm, however, unlike previous approaches it attempts to remove the limitations on the correlation length of the steps in the model by implementing Hamiltonian dynamics through a Gaussian-distributed momentum vector, according to the usual equations

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i} \quad (2.3)$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial x_i} \quad (2.4)$$

where p is the momentum vector, x is the position vector, and H is the Hamiltonian of the system defined by the equation

$$H(x, p) = U(x) + K(p) \quad (2.5)$$

where $U(x)$ is the potential energy, and $K(p) = \frac{p^2}{2}$ is the kinetic energy of the system [16].

Instead of allowing random steps to be taken, the system is given a momentum and evolved through the configuration space using a numerical integrator. This momentum is an independent, random variable

and so does not contribute to the end result. In our case, the Leapfrog Method was used to carry out this momentum evolution for proposed moves to new points in the space.

Like before, initial configurations are randomly generated for the initial position, x_0 , and in this case the initial momentum, p_0 . These values are stored and evolved using the Leapfrog integration method, defined by the equations

$$p_i \left(t + \frac{\epsilon}{2} \right) = p_i(t) - \frac{\epsilon}{2} \frac{\partial U}{\partial x_i} x_i(t) \quad (2.6)$$

$$x_i(t + \epsilon) = x_i(t) + \epsilon p_i \left(t + \frac{\epsilon}{2} \right) \quad (2.7)$$

$$p_i(t + \epsilon) = p_i \left(t + \frac{\epsilon}{2} \right) - \frac{\epsilon}{2} \frac{\partial U}{\partial x_i} x_i(t + \epsilon) \quad (2.8)$$

where ϵ is the size of each step [16]. As seen in the above equations, the initial momentum variables undergo one half step. Then, using the new values for the momentum variables, the position variables undergo one full step, and finally the momentum variables undergo another half step using the newly generated position variables. The Leapfrog integration method is only slightly higher in computational cost than the Euler method, but it's error term is third order, ensuring more accuracy from the simulation [16]. As can be seen in Fig. 1, the derivatives are calculated at the midpoint of each step. Thus, this method only requires a single additional step to be taken in order to move the momentum one half step ahead of the position.

Once the integration is complete, a test similar to that of the MCMC is carried out, except in this case, ΔH is determined by the difference in the Hamiltonian between the i^{th} step and the $(i + 1)^{\text{th}}$ step:

$$\Delta H = H(x_{i+1}, p_{i+1}) - H(x_i, p_i) \quad (2.9)$$

where the momentum samples are discarded after sampling.

As before, a uniform, random number t is generated, $t \in [0, 1]$, and a decision is made depending on the following outcome:

- If $e^{-\Delta H} \leq t$, the new configuration is accepted and becomes the new initial value for the next step, $x_{i+1} = x_i$.
- If $e^{-\Delta H} > t$, the new configuration is rejected and replaced with the previous one.

Compared to the randomness of the Metropolis algorithm, the HMC reduces the correlation between successive sample states by proposing moves that maintain a high probability of acceptance. This is a direct result of the energy-conservation properties that accompany the use of Hamiltonian dynamics while integrating. The proposed states move across the sample space in larger steps, and thus are more likely to be dissimilar to the previous states. This results a quicker rate of convergence to the target distribution, and the avoidance of the slow exploration of the space that results from the random nature of previous models [16]. Due to this reduced correlation, fewer Markov chain samples are necessary to approximate the integrals required.

2.3 An Application of the Monte Carlo Method: The Ising Model

The magnetic dipoles of atoms in materials exist usually with a random orientation. As a result of this, no overall macroscopic magnetic moment is achieved. However, in certain materials such as iron, cobalt, and nickel, a preferred alignment occurs among atomic spins, resulting in a magnetic moment. This phenomenon, known as ferromagnetism, occurs due to the system's attempt to control both energy minimisation and entropy maximisation, with temperature being the mediator between the two, as described by the Gibbs distribution [17].

The Ising Model is a mathematical representation of such ferromagnetic materials, providing an effective and accessible way to solve problems while allowing us to ignore the more complex aspects of the system like quantum spin and angular momentum. The model consists of dipole spins placed at regular points on a square lattice with dimensions L , restricted to be in either the up (+1) or down (-1) states. The range of interaction for each spin is restricted to immediately adjacent sites, and periodic boundary conditions are enforced on the lattice in order to maximise spin interaction at the edges.

The Hamiltonian for a spin at a specific site on the lattice is described by the equation

$$H_i = -J \sum_{\langle i,j \rangle} S_i S_j \quad (2.10)$$

where $\langle i,j \rangle$ is the sum over the nearest-neighbours of i , S_i is the spin at site i , and J is the coupling constant. The nature of the interactions depends on the sign of this coupling constant. If J is positive, the system will favour parallel alignment and the material will be ferromagnetic. If J is negative, the system will favour anti-parallel alignment and the material will be anti-ferromagnetic. In our case, this coupling constant was taken to be $+1$.

The energy of the system depends on whether neighbouring spins in the lattice are in alignment or not, with spins that agree on alignment having a lower energy than those which don't. This model allows for the identification and analysis of a phase transition, as although the system tends to the lowest energy, it is also dependent on the temperature, allowing for more than one structure to be obtained.

The Ising model can often be difficult to evaluate numerically if there are many states in the system. Since two spin orientations are being considered, and there are N number of spins, there are 2^N different possible states. As N becomes large the computations become more tasking and time consuming, thus motivating the simulation of the Ising Model using Monte Carlo methods.

2.3.1 The Algorithm

The lattice is initialised to a random starting configuration, with the randomness having the benefit of using less computation time than trying to reach a homogeneous configuration [17]. A random location on the lattice is selected and a decision is made as to whether the energy change that would result after flipping that particular spin is less than zero. If this energy change is found to be less than zero, the spin is flipped at that site and the associated change in energy is recorded. However, if this energy change is found instead to be greater than zero, the metropolis test is carried out to decide whether to flip the spin — a random number is generated between 0 and 1 and then weighed against the Boltzmann probability factor, $e^{-\delta\beta H}$. If the random number is less than this probability, the spin is flipped, otherwise it is left unchanged in its original configuration.

This process is then repeated until the stopping criterion is met, iterating several times over the whole lattice. This algorithm produces a single set of observables at a specified temperature. In order to identify the phase transition that takes place, many temperatures must be considered, and the process repeated a number of times.

A problem that can occur after the initialization of the lattice in the beginning is that the configuration may not necessarily be in equilibrium and will require several few Monte Carlo steps to reach this equilibrated state. The results that are produced during this period are transient and can be disregarded. Therefore, in order to avoid this, the program is adjusted such that one thousand Monte Carlo steps are made prior to any data collection in order to ensure that this equilibrated state has been reached.

2.3.2 Results & Data Analysis

This method was employed and studied for lattices of size 2×2 , 5×5 , and 20×20 , with temperatures ranging from $T = 5$ to $T = 0.5$, with a step size of $\delta T = 0.1$. The expectation value of the energy was

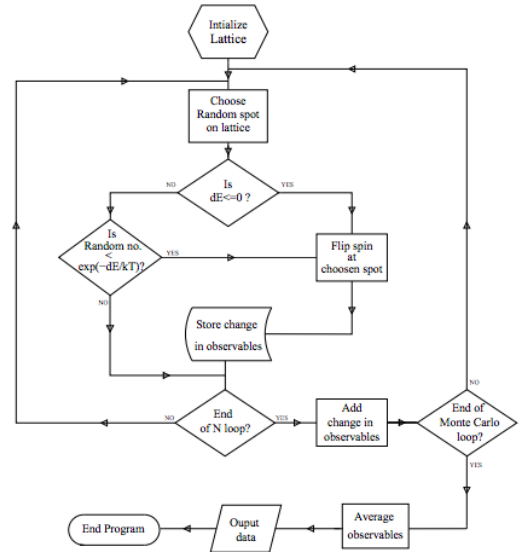


Fig. 1: The flow of logic behind the Metropolis Algorithm [17].

calculated using the equation

$$\langle E \rangle = \frac{1}{2} \left\langle \sum_i^N H_i \right\rangle \quad (2.11)$$

with the factor of a half being introduced to account for each spin being counted twice.

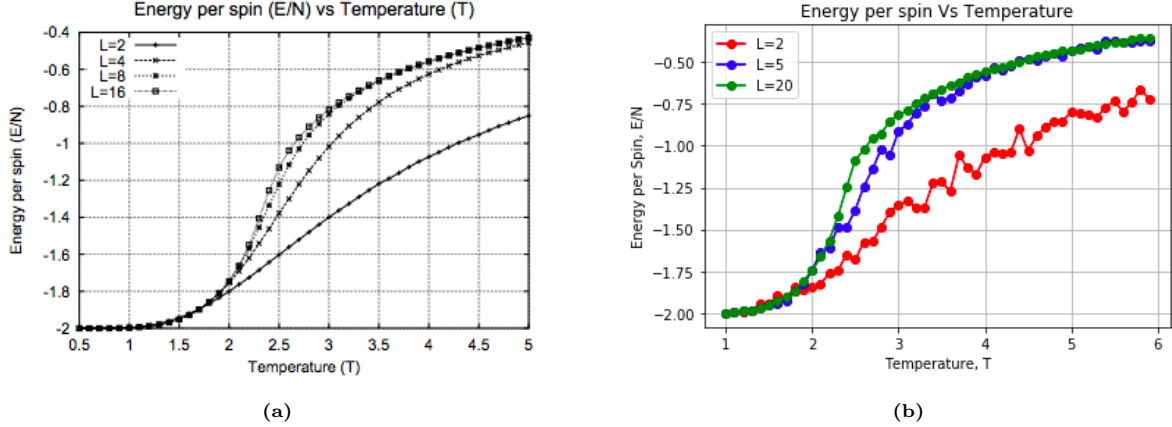


Fig. 2: Graph showing the Energy per spin Vs. the temperature of the system for varying lattice sizes, $L \times L$. Fig. (a) shows the expected values [17], while Fig. (b) shows the results obtained from this experiment.

Fig. 2 above shows the energy per spin as a function of the temperature of the system. Evidently, the curve of the graph becomes more pronounced as the lattice size is increased. This effect, however, becomes less noticeable as higher lattice sizes are considered, as seen from the lack of a definite marked difference between the 8×8 and 16×16 size lattices on the graph. The steep gradient in the larger lattices points towards a possible phase transition. At higher temperatures, the energy per spin is relatively high, keeping with the expectation of having a random configuration. At lower temperatures, the curve appears to stabilise at $\frac{E}{N} = -2J = -2$, giving the indication that the spins are all aligned in parallel [17]. Although our results for the $L = 20$ case nearly exactly mirror the expected curve, the plot obtained for the $L = 2$ case is not as smooth as anticipated. This could be due to the number of steps taken. Since this is the energy per spin, evidently, the more samples taken, the more smoothly the resulting curve will be.

2.3.3 The Potts Model

One of the limitations of the Ising Model is that it only allows for two spin states — spin up, and spin down. This model, however, can be generalised to the Potts Model, one that can function for systems with an arbitrary number of spin states. The Hamiltonian for such systems changes to

$$H_i = -J \sum_{\langle i,j \rangle} \delta_{i,j} \quad (2.12)$$

where $\delta_{i,j}$ is the Kronecker delta, equal to 1 if the spins at site i and j are in alignment, and equal to 0 otherwise.

Fig. 3 below shows the results obtained for a ten state Potts model with a lattice size of $L = 10$. The critical temperature occurs clearly at $T_C = 0.72$, which compares very well to the expected result of 0.7 [5]. A finite peak can be observed in the specific heat capacity at the transition point, as seen in Fig. 3b, agreeing with the literature [5]. These effects, however, are most notable as the dimensions of the lattice are taken to infinity. Fig. 4 below shows the fluctuations of the first order transition in the ten-state Potts Model that we produced in Fig. 3. The nature of this plot unambiguously confirms the presence of a first order transition at this critical temperature.

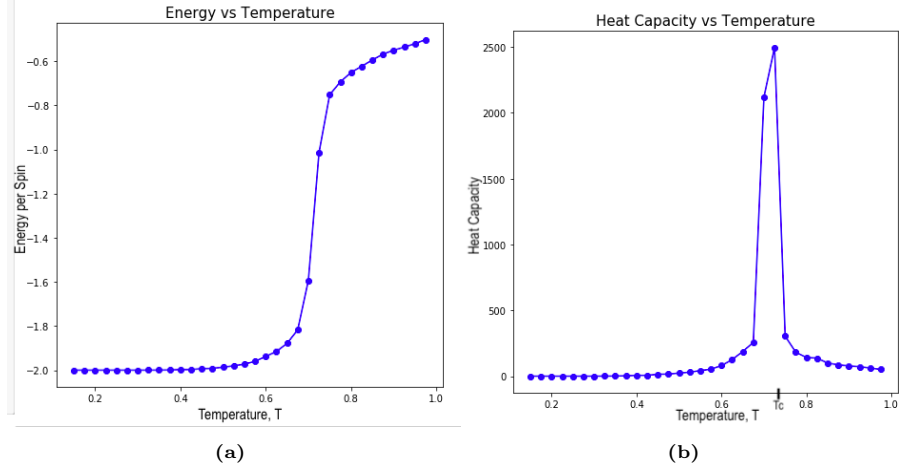


Fig. 3: Graph showing the Potts Model for a system with ten spin states and lattice of dimensions $L = 10$. Fig. (a) shows the Energy per spin Vs. the temperature, with a clear transition occurring at $T = 0.72$, while Fig. (b) shows the Heat capacity vs. the temperature, with a spike occurring at this critical temperature.

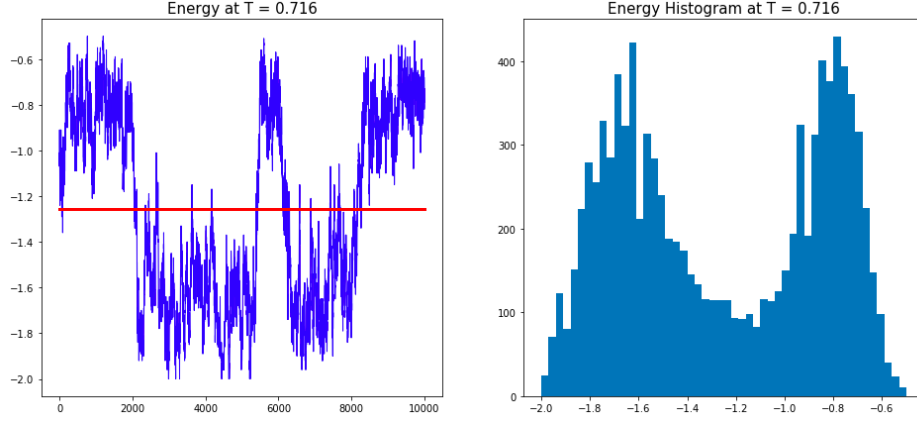


Fig. 4: Graphs showing the time dependence of the internal energy in the MC simulations on a 10×10 lattice.

3 Machine Learning for Quantum Control

3.1 Machine Learning and its use for Physics

‘Machine Learning’ is a very malleable word depending on the context in which it appears. It is worth spending some time clarifying what we mean when we use the term in this paper. For our purposes, machine learning refers to computer systems that have the ability to improve themselves without explicit programming. This has many uses, especially today in the world of Big Data — referring to extremely large data sets that a human would have no hope of analysing manually, but machine-learning algorithms are particularly useful analysing trends from. Big Data also often provides a large playground for the machine-learning algorithm to learn from and make accurate predictions on future data. This leads to an important distinction between the types of machine learning that are currently used.

3.1.1 Supervised, unsupervised and reinforcement learning

Machine-learning algorithms are typically split into three subsections depending on their execution and method by which they learn: supervised learning, unsupervised learning and reinforcement learning. While the boundaries between the three are not often rigid, it is worthwhile to define them and acknowledge the usage case for each type. We will treat the difference between the three at a qualitative level without

considering the nuances of how each type of learning functions mathematically.

Supervised learning is the most quintessential example of a machine-learning algorithm. It generates a predictive model based on data that it is ‘trained’ on — the key importance being that the data has a label so that the algorithm can ‘check’ if it was right and adjust its next guess accordingly. The image of a student using flash cards to test themselves on a topic and flipping over to see the answer is an apt one. Once trained, the algorithm is then fed new, unlabelled data of a similar type to its training data, and makes a prediction on what the unknown label for that data should be.

Unsupervised learning, naturally the converse of the previous, finds its use analysing unlabelled systems of data. It does not ‘train’ and instead classifies the data it is given to minimise the error in the predictive model it generates. Think of drawing a line through a cluster of data points minimises the distance from the line to all the points. Unsupervised learning is arguably the most useful to physical science, as it provides new insights to data without prior knowledge being a prerequisite.

Reinforcement learning is somewhat more specific than the other two types, and can often also incorporate the previous two types of learning. It involves the presence of an agent with a strategy for execution. A reward function tracks the performance of this strategy through clauses we have defined, rewarding the agent for actions we deem as useful while penalising for actions we deem as negative. The agent will repeat continuously to attempt to maximise its reward, in turn telling us the most optimal way to carry out a task or to analyse data.

Within all three types of learning, there is a very fundamental concept hiding under the hood, the cost function. This is a mathematical function that quantifies the accuracy of the algorithm’s predictions or the efficiency of its actions. It is what lies at the heart of all machine-learning algorithms, and its optimisation through time is what gives the algorithm its appearance of ‘learning’.

3.1.2 Uses in Physics

While machine learning is traditionally associated with more commercial uses, such as smart assistants being able to understand and reproduce language, it is finding increasing use in Physics.

Astronomy has seen a large growth in the use of machine learning in recent years [6]. This is due to the large quantity of datasets being generated observing the cosmos — pushing astronomy into the regime of big data. The use of a machine-learning algorithm to classify a galaxy from an image or to determine the speed at which a body is moving from us based on its emission spectra can be massively useful in reducing the analysis needed to be performed by hand on these large astronomical datasets.

Medical Physics is an area that has also found a use for machine learning in a variety of areas [7]. These applications are wide, including automation of mundane tasks, amplifying the precision of treatments and assisting in diagnosis and planning of treatment. These are due to the ability of deep learning networks — a subsection of machine learning that utilises neural networks that take in an input and translate it to a higher-dimensional ‘hidden layer’ before producing an output — to accurately analyse images and categorise important features. This adds an extra fidelity to diagnosis stemming from medical imaging such as MRI.

However, one of the most fruitful areas of use of machine learning in physics, and the emphasis of this section of this paper, is the use of machine learning in quantum computation and information. This comes naturally, due to the hybrid nature much of quantum information shares with information science and computer science. The nature between quantum physics and machine learning is two-fold; the use of quantum computation to speed-up the ability of a machine-learning algorithm to train and/or increase its efficiency and effectiveness when in operation, and also the use of classical machine learning to analyse and classify quantum systems. We will focus on the latter.

3.2 Quantum Control

Machine learning promises many uses for quantum physics. One is the use of neural networks to analyse quantum many-body problems and have the network predict the ground-state of the composite quantum system [8] — a notoriously difficult task. Another use, directly influencing the field of quantum computation, is the use of machine learning to classify quantum states. This includes whether the state is entangled, mixed, decoherent etc. This promises to be advantageous in the building of larger quantum computers which are more resistant to their environment [9].

For us, the interest is on the area of quantum control. This — for our purposes — will involve finding the most optimal method to either transform a quantum state or to maintain a quantum state against noisy surroundings. The advantages of achieving this control are many; it may inform efficient design and control of quantum computers or maximise the information that can be gained from a system via quantum metrology for example. Machine learning is a valuable approach for quantum control. The machinery of quantum mechanics naturally give many ways to classify how ‘close’ a current state is to a target state — making for a perfect cost function for the machine-learning algorithm to optimise over. This may be an probability amplitude in Dirac Notation $|\langle\phi|\psi\rangle|^2$, quantifying the probability of measuring state $|\phi\rangle$ to be in state $|\psi\rangle$ and vice versa, or the trace of the density matrix squared in density operator notation $\text{tr}(\hat{\rho}^2)$, informing how decoherent the total system is. These two examples will be utilised and explored in this paper.

3.3 Determining magnetic field to control a qubit state

To consider the power of this machine learning approach to quantum control, we will consider a relevant problem. We consider a single 2-level spin system which is described via the Hamiltonian

$$\hat{H} = -\hat{\sigma}_z - h_x(t)\hat{\sigma}_x, \quad (3.1)$$

where $\hat{\sigma}_z$ and $\hat{\sigma}_x$ are the Pauli Operators represented by the conventional 2×2 matrices. $h_x(t)$ is an external magnetic field capable of changing the system state.

The problem is: given the system is initially in the groundstate configuration $|\Psi(0)\rangle$ with $h_x(0) = -2$, what time-dependent magnetic field profile brings the qubit as close as possible to a target state $|G(T)\rangle$, the groundstate of the Hamiltonian Eq. 3.1 when $h_x(T) = 2$ after a time T has passed? Using our previous discussions, it is easy to see why machine learning is a prime candidate for handling this task by introducing the **fidelity**;

$$F = |\langle G(T)|\Psi(T)\rangle|^2. \quad (3.2)$$

The fidelity serves as a perfect cost function for the machine-learning algorithm to optimise over: it is bounded between 0 and 1 and gives a succinct qualitative measure of how aligned two states are in our Hilbert Space. All that is left is to choose the method of machine learning. Reinforcement learning lends itself well to this problem, as you can visualise the algorithm taking the initial state through the evolution along the Bloch sphere [18] as close to the target state as it can before reevaluating the fidelity Eq. 3.2 and trying again. A diagram of this can be seen below in Fig. 5. Indeed, this approach has been executed with a great deal of success already [10].

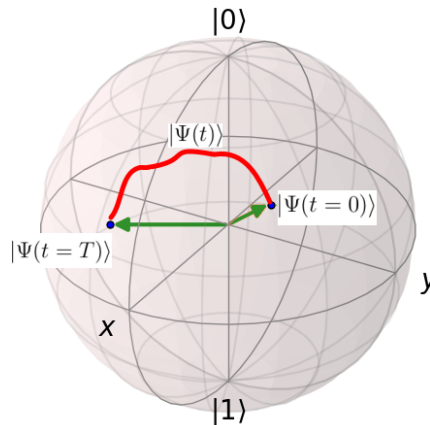


Fig. 5: An illustration of the vectors on the Bloch Sphere. We see the two green vectors as the initial state at $t = 0$ and the target state at $t = T$. The red is an optimal path the algorithm has found to map $|\Psi(t = 0)\rangle$ to $|\Psi(t = T)\rangle$ by performing gradient descent on the applied magnetic field $h_x(t)$.

The approach we take, however, is to treat this as a solely unsupervised learning problem. We optimise the fidelity Eq. 3.2 using Gradient Descent [11] (or rather — ascent in this case, since we’re maximising the

fidelity).

This was done by splitting $h_x(t)$ into a time lattice — A discretised array of values separated by a time dt . We will represent this $h_x(t_i)$ where t_i are i discrete times separated by dt . The rigour from mathematical analysis allows this as a fair numerical approximation as for $dt \rightarrow 0$, $h_x(t_i)$ tends back to a continuous function in time. This splitting allows us to treat each element of the $h_x(t_i)$ array as a feature that the final fidelity $F(T)$ is dependent on. Using gradient descent, we can then optimise each feature of $h_x(t_i)$ in time iteratively, in turn optimising the entire array. The power of this approach over reinforcement learning lies in the simplicity with which the gradient descent algorithm can be implemented. Taking that the fidelity $F(T)$ is implicitly dependent on each section of the $h_x(t_i)$ array, gradient descent improves $F(T)$ by iteratively shifting each $h_x(t_i)$ in the direction of positive gradient of $F(T)$ relative to that segment:

$$h_x(t_i) \rightarrow h_x(t_i) + \frac{\partial F(T)}{\partial h_x(t_i)} \times \text{LR}, \quad (3.3)$$

where LR represents the learning rate of the algorithm. This is a factor that determines the sensitivity of the algorithm's convergence to the local optimum. We will forgo discussion of the nuance of the learning rate for brevity here, but readers with machine-learning experience may identify 1 as quite a large rate — other rates were tested and this was found to be optimal. Concern of over-fitting to the model can be waived since it is not being used to make future predictions. The $h_x(t_i)$ array sections were separated by $dt = 0.01$ and the gradient descent was iterated 1000 times. Results of this approach are seen below in Fig. 6.

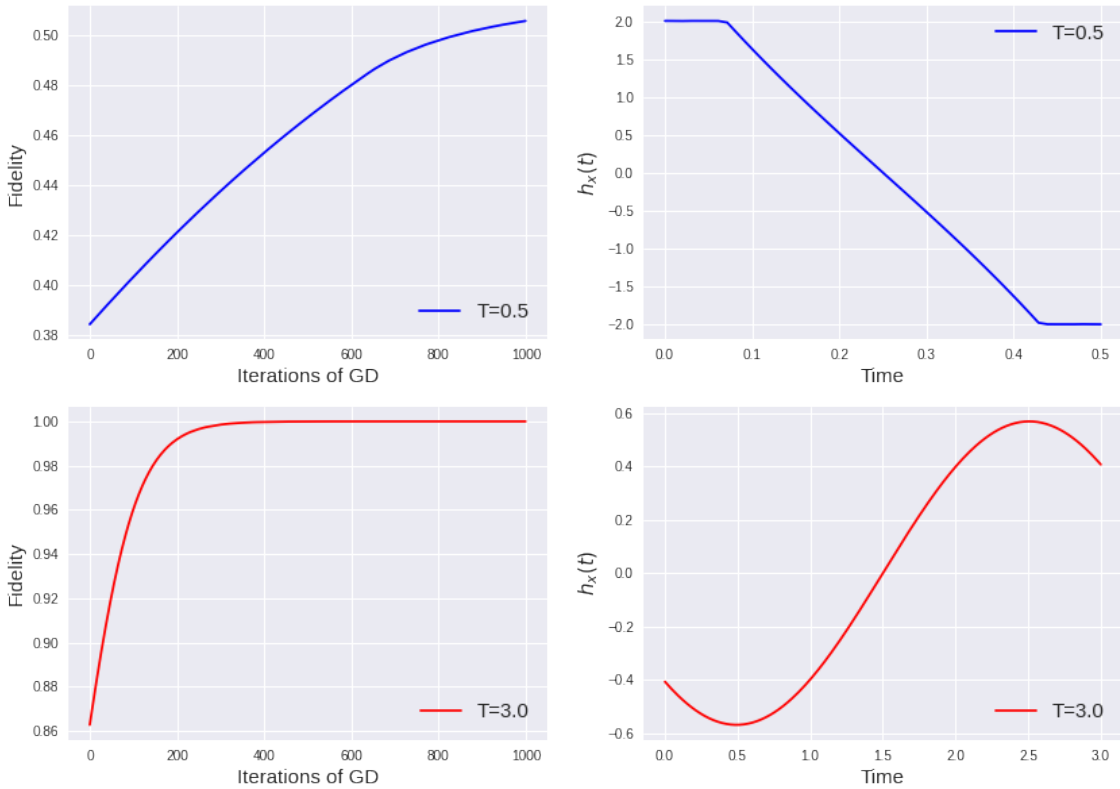


Fig. 6: On the left is a figure of total fidelity versus number of iterations of gradient descent learning executed. Includes learning for dynamics up to a total time of $T = 0.5$ and $T = 3.0$ for the system. On the right we see the $h_x(t)$ profile that the algorithm found after 1000 iterations again for $T = 0.5$ and $T = 3.0$.

As we would expect for gradient descent, we see the algorithm improves the fidelity quickly at first, and then begins to slow the improvements as it approaches the local maximum. The actual $h_x(t)$ profile generated for the maximum time $T = 0.5$ shows an immediate flip to $h = 2$ followed by a discontinuous drop to the $h = -2$ halfway along the time series. For a longer maximum time of $T = 3.0$ we see a much

smoother transition for $h_x(t)$, appearing sinusoidal in nature. We attribute this fact to the algorithm having to act much more rashly on a shorter timescale of $T = 0.5$ than $T = 3.0$ to attempt to achieve the maximum fidelity to the target state $|G(T)\rangle$.

An interesting behaviour observed is the dependence of the upper bound of the fidelity on the final time of the state evolution $t = T$. For the initial exploration with $t = 0.5$, we see that the maximum fidelity can not be achieved much higher than 0.5, but the model approaches perfect fidelity for $t = 3$. Fig. 7 shows an exploration of this relationship in more detail.

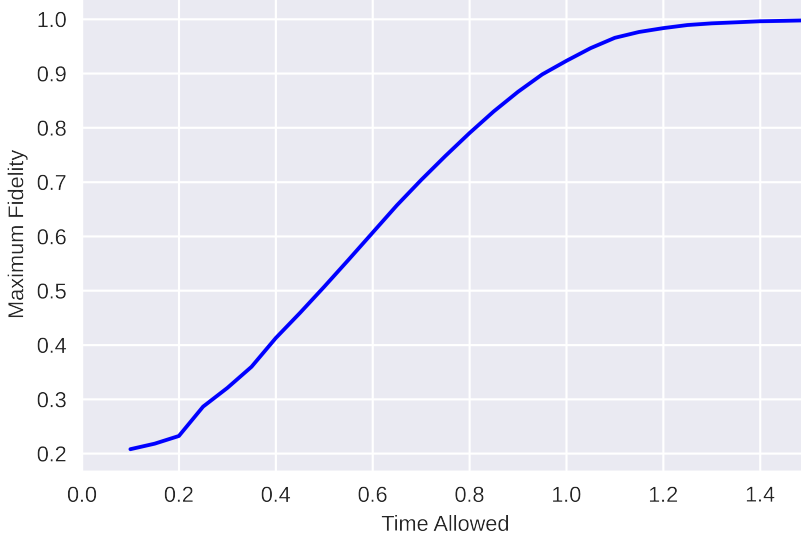


Fig. 7: A plot of maximum achievable fidelity for a given allowed time of evolution for the state. For $T \sim 1.4$, fidelity of 1.0 is achievable.

For these results, the algorithm did not run for set iterations and instead terminated after the improvements on fidelity were only being made on the scale of 10^{-3} . We clearly demonstrate a dependence on the maximum fidelity of the final state to the target state on the total time of evolution of the dynamics of the system that appears linear in nature. At around $T = 1.0$ the nature ceases to appear linear and plateaus into settling at fidelity of 1.0 at around $T = 1.4$. If that total evolution time is granted, the algorithm behaves exceptionally well, with much simpler code being implemented than in its reinforcement learning counterpart [10].

These results for quantum control from low complexity machine-learning algorithms are promising. However, we have so far only applied them to a single 2-spin qubit. For applicability to practical quantum information science, we must generalise to larger systems of interacting qubits. While we could easily generalise the current Hamiltonian Eq. 3.1 to include more qubits, we could only simulate a maximum of ~ 10 qubits on a classical computer due to the exponential dependence of the dimension of the Hilbert space on the number of qubits interacting in the system. At the time of writing, IBM’s Q quantum computer has 53 interacting qubits [12]. For insights gained from our simulations to be of use, we must push to system sizes comparable or beyond this. Thus, we embark on means to simulate large systems of interacting qubits on a classical computer in Sec. 4.

4 Fermionic Representation of Spin Chains

4.1 Introduction

We use a Jordan-Wigner transformation and Gaussian formalism to efficiently model the decoherence of a spin-1/2 particle due to interaction with a spin chain. The decoherence of a quantum system, \mathcal{S} , due

to interactions with an environment, \mathcal{E} , is an unavoidable occurrence in quantum computing. Modelling such systems is a useful tool to help our understanding, but brute-force approaches, where one simulates the composite \mathcal{SE} -system in its entirety, can be computationally very expensive, even for a small system of interest \mathcal{S} . Alternative methods employ approximations (weak coupling approximation, Born-Markov approximation, etc.) to reduce the degrees of freedom of the composite system, but usually cannot model strong coupling between the system and environment.

We focus on a model that can be simulated efficiently with no loss of information or accuracy. The quantum XY spin chain can be mapped to a free fermion model via the Jordan-Wigner transformation. In what follows, we generalise the quantum XY model by introducing an arbitrary Hamiltonian for one of the spins in the chain. At first glance, this addition introduces linear fermion terms, meaning we cannot immediately make use of the Gaussian formalism to model the system efficiently. However, the linear terms can be accounted for by the introduction of a single ancillary fermion, a method due to Colpa [19], allowing us to take advantage of the Gaussian approach. This is similar to the work of Rossini *et al.* [13]. The magnetic field in their paper is purely transverse, whereas our method allows us to include components in the x - and y -directions. The cost of this is that we are limited in the type of system-environment coupling we can choose.

In a model with N spins, the Gaussian method allows us to model the system using matrices of size $\mathcal{O}(N)$, as opposed to a brute force approach, with matrices of size $\mathcal{O}(2^N)$. In the brute force approach, the computation time became excessive even for $N < 20$, whereas the Gaussian method allowed us to simulate spin chains of size $N = 300$ without much difficulty. We then use this framework to investigate the behaviour of our system in different situations, including critical points and temperature dependence of its decoherence, along with finite-size effects.

4.2 Theoretical Framework

4.2.1 Model

We consider a open chain of N spin-1/2 particles. Our system, \mathcal{S} , is a single spin-1/2 particle in a magnetic field $\boldsymbol{\omega}_1 = (\omega_1^x, \omega_1^y, \omega_1^z)$, at the first site in the chain. The other spins in the chain function as our environment, \mathcal{E} . Its dynamics are governed by nearest neighbour coupling, the strength of which depends on the constants λ_j^\pm , $j \neq 1$. Further, each spin is affected by an external transverse magnetic field with strength ω_j^z , $j \neq 1$, which may be different for each spin. The system-environment interaction, $\hat{V}_{\mathcal{SE}}$, is governed by the (complex) coupling strengths λ_1^\pm and λ_1^z . The Hamiltonian of the composite system is then

$$\hat{H} = \hat{H}_{\mathcal{S}} + \hat{H}_{\mathcal{E}} + \hat{V}_{\mathcal{SE}}, \quad (4.1)$$

where

$$\hat{H}_{\mathcal{S}} = \boldsymbol{\omega}_1 \cdot \hat{\boldsymbol{\sigma}}_1, \quad (4.2)$$

$$\hat{H}_{\mathcal{E}} = \sum_{n=2}^N \omega_n^z \hat{\sigma}_n^z + \sum_{n=2}^{N-1} [(\lambda_n^- \hat{\sigma}_n^- + \lambda_n^+ \hat{\sigma}_n^+) \hat{\sigma}_{n+1}^+ + \text{h.c.}], \quad (4.3)$$

$$\hat{V}_{\mathcal{SE}} = (\lambda_1^- \hat{\sigma}_1^- + \lambda_1^+ \hat{\sigma}_1^+ + \lambda_1^z \hat{\sigma}_1^z) \hat{\sigma}_2^+ + \text{h.c.} \quad (4.4)$$

The initial state of the system is given by

$$\hat{\rho}(0) = \hat{\rho}_{\mathcal{S}}(0) \otimes \hat{\rho}_{\mathcal{E}}^{\text{th}}(\beta), \quad (4.5)$$

where $\hat{\rho}_{\mathcal{S}}(0) = \frac{1}{2} (\mathbb{I}_2 + \mathbf{n} \cdot \hat{\boldsymbol{\sigma}}_1)$ is an arbitrary mixed state of the system \mathcal{S} with Bloch vector \mathbf{n} , and $\hat{\rho}_{\mathcal{E}}^{\text{th}}(\beta) = \exp(-\beta \hat{H}_{\mathcal{E}}) / \mathcal{Z}$ is the thermal state with inverse temperature β and partition function $\mathcal{Z} = \text{tr} [\exp(-\beta \hat{H}_{\mathcal{E}})]$. The state of the system evolves according to the Schrödinger equation

$$\frac{d}{dt} \hat{\rho}(t) = -i[\hat{H}, \hat{\rho}(t)]. \quad (4.6)$$

Solving this differential equation numerically requires us to work with matrices of size $2^N \times 2^N$, the size of which quickly becomes prohibitive, but the Gaussian approach provides an alternate, and far more efficient, solution.

4.2.2 Gaussian Fermionic Systems

Consider a system of N fermions with creation and annihilation operators, \hat{a}_n^\dagger and \hat{a}_n , for $n = 1, \dots, N$, satisfying the canonical commutation relations $\{\hat{a}_n, \hat{a}_m^\dagger\} = \delta_{nm}$, $\{\hat{a}_n, \hat{a}_m\} = 0$. We define the vector

$$\boldsymbol{\alpha} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_N \\ \hat{a}_1^\dagger \\ \vdots \\ \hat{a}_N^\dagger \end{bmatrix}. \quad (4.7)$$

Suppose we have a system Hamiltonian which is quadratic in the creation and annihilation operators,

$$\hat{H} = \sum_{n,m=1}^{2N} c_{nm} \hat{\alpha}_n^\dagger \hat{\alpha}_m = \frac{1}{2} \boldsymbol{\alpha}^\dagger \mathbf{H} \boldsymbol{\alpha}. \quad (4.8)$$

It can be shown, via the commutation relations, that \mathbf{H} can be chosen to be of the form

$$\mathbf{H} = \begin{pmatrix} \boldsymbol{\mu} & -\boldsymbol{\nu}^* \\ \boldsymbol{\nu} & -\boldsymbol{\mu}^* \end{pmatrix}, \quad (4.9)$$

where $\boldsymbol{\mu}$ is Hermitian and $\boldsymbol{\nu}$ is antisymmetric. Given a state of the fermionic system $\hat{\rho}$, we define its covariance matrix, $\boldsymbol{\Lambda}$, by

$$\Lambda_{nm} = \text{tr}(\hat{\rho}[\hat{\alpha}_m^\dagger, \hat{\alpha}_n]), \quad (4.10)$$

which evolves according to a Schrödinger-type equation (see Appendix A.1 for the derivation):

$$\frac{d}{dt} \boldsymbol{\Lambda} = -i[\mathbf{H}, \boldsymbol{\Lambda}]. \quad (4.11)$$

A Gaussian state is a state that can be written in the form

$$\hat{\rho} = \frac{1}{\mathcal{Z}} \exp(-\hat{A}/2) = \frac{1}{\mathcal{Z}} \exp\left(-\frac{1}{2} \boldsymbol{\alpha}^\dagger \mathbf{A} \boldsymbol{\alpha}\right), \quad (4.12)$$

where \hat{A} is quadratic in the creation and annihilation operators, and $\mathcal{Z} = \text{tr}[\exp(-\hat{A}/2)]$ is a normalising factor. As in the case of the Hamiltonian, we can choose \mathbf{A} to be of the form shown in Eq. 4.9. A Gaussian state is completely characterised by its covariance matrix $\boldsymbol{\Lambda}$ via the relation $\boldsymbol{\Lambda} = -\tanh\left(\frac{\mathbf{A}}{2}\right)$ [20].

The great advantage of using the Gaussian formalism is that the size of the covariance matrix is $2N \times 2N$, much smaller than the $2^N \times 2^N$ density matrix $\hat{\rho}$, allowing us to proceed numerically using the much smaller covariance matrix while losing no information.

4.2.3 Jordan-Wigner Transformation

In order to employ the Gaussian formalism we need to represent our spin-chain as a system of fermions. This is achieved through a Jordan-Wigner transformation. We define the operators, for $1 \leq n \leq N$,

$$\hat{a}_n = \left(\bigotimes_{m=1}^{n-1} \hat{\sigma}_m^z \right) \hat{\sigma}_n^+, \quad (4.13)$$

which are seen to satisfy the canonical commutation relations. The Hamiltonian can now be expressed as the sum of

$$\hat{H}_{\mathcal{S}} = (\omega_1^x + i\omega_1^y) \hat{a}_1^\dagger + (\omega_1^x - i\omega_1^y) \hat{a}_1 + \omega_1^z (\hat{a}_1 \hat{a}_1^\dagger - \hat{a}_1^\dagger \hat{a}_1) \quad (4.14)$$

$$\hat{H}_{\mathcal{E}} = \sum_{n=2}^N \omega_n^z (\hat{a}_n \hat{a}_n^\dagger - \hat{a}_n^\dagger \hat{a}_n) + \sum_{n=2}^{N-1} [(\lambda_n^- \hat{a}_n^\dagger - \lambda_n^+ \hat{a}_n) \hat{a}_{n+1} + \text{h.c.}] \quad (4.15)$$

$$\hat{V}_{\mathcal{SE}} = (\lambda_1^- \hat{a}_1^\dagger - \lambda_1^+ \hat{a}_1) \hat{a}_2 + \lambda_1^z \hat{a}_2 + \text{h.c.} \quad (4.16)$$

If the initial state $\hat{\rho}_S(0)$ is not pure, or maximally mixed, i.e. $0 < |\mathbf{n}| < 1$, it becomes (see Appendix A.2 for the derivation):

$$\hat{\rho}(0) = \frac{\sqrt{1 - |\mathbf{n}|^2}}{2\mathcal{Z}} \exp \left\{ \frac{\tanh^{-1} |\mathbf{n}|}{|\mathbf{n}|} \left(n^z \hat{a}_1 \hat{a}_1^\dagger - n^z \hat{a}_1^\dagger \hat{a}_1 + n^- \hat{a}_1 + n^+ \hat{a}_1^\dagger \right) - \beta \hat{H}_\mathcal{E} \right\}, \quad (4.17)$$

where $n^\pm = n_x \pm i n_y$, and \mathcal{Z} is a new normalisation factor. The case of pure states is handled by a limiting process described in Appendix A.4. The $|\mathbf{n}| = 0$ can be handled in the same way, but does not arise in our results.

This Hamiltonian includes terms linear in the fermionic operators, and the state is not Gaussian, so the Gaussian formalism cannot immediately be used. To account for this, following the method in [19], we introduce a fictitious fermion at site $n = N + 1$, which is not coupled to the rest of the chain. By the Jordan-Wigner transformation, the fermionic operators corresponding to this spin are given by $\hat{a}_{N+1} = \left(\bigotimes_{m=1}^N \hat{\sigma}_m^z \right) \hat{\sigma}_{N+1}^+$. We map our system to the new via the unitary transformation $U = (\hat{a}_{N+1} \hat{a}_{N+1}^\dagger - \hat{a}_{N+1}^\dagger \hat{a}_{N+1} + \hat{a}_{N+1}^\dagger + \hat{a}_{N+1})/\sqrt{2}$. Linear fermion terms transform as

$$\hat{U}^\dagger \hat{a}_n \hat{U} = \hat{a}_n (\hat{a}_{N+1} - \hat{a}_{N+1}^\dagger), \quad n \neq N + 1 \quad (4.18)$$

while quadratic fermion terms are unchanged. In this new basis we have the Hamiltonian

$$\hat{U}^\dagger \hat{H}_S \hat{U} = (\omega_1^x + i\omega_1^y) \hat{a}_1^\dagger (\hat{a}_{N+1} - \hat{a}_{N+1}^\dagger) + (\omega_1^x - i\omega_1^y) \hat{a}_1 (\hat{a}_{N+1} - \hat{a}_{N+1}^\dagger) + \omega_1^z (\hat{a}_1 \hat{a}_1^\dagger - \hat{a}_1^\dagger \hat{a}_1) \quad (4.19)$$

$$\hat{U}^\dagger \hat{H}_\mathcal{E} \hat{U} = \sum_{n=2}^N \omega_n^z (\hat{a}_n \hat{a}_n^\dagger - \hat{a}_n^\dagger \hat{a}_n) + \sum_{n=2}^{N-1} [(\lambda_n^- \hat{a}_n^\dagger - \lambda_n^+ \hat{a}_n) \hat{a}_{n+1} + \text{h.c.}] \quad (4.20)$$

$$\hat{U}^\dagger V_{S\mathcal{E}} \hat{U} = (\lambda_1^- \hat{a}_1^\dagger - \lambda_1^+ \hat{a}_1 + \lambda_1^z \hat{a}_{N+1}^\dagger - \lambda_1^z \hat{a}_{N+1}) \hat{a}_2 + \text{h.c.}, \quad (4.21)$$

and the Gaussian state, $\hat{U}^\dagger \hat{\rho}(0) \hat{U}$, is

$$\frac{\sqrt{1 - |\mathbf{n}|^2}}{2\mathcal{Z}} \exp \left\{ \frac{\tanh^{-1} |\mathbf{n}|}{|\mathbf{n}|} \left(n^z \hat{a}_1 \hat{a}_1^\dagger - n^z \hat{a}_1^\dagger \hat{a}_1 + [n^- \hat{a}_1 (\hat{a}_{N+1} - \hat{a}_{N+1}^\dagger) + \text{h.c.}] \right) - \beta \hat{H}_\mathcal{E} \right\}. \quad (4.22)$$

We can now apply the Gaussian formalism to the $N + 1$ fermion system, writing $H = \frac{1}{2} \boldsymbol{\alpha}^\dagger \mathbf{H} \boldsymbol{\alpha}$ and $A = \frac{1}{\mathcal{Z}} \exp(-\frac{1}{2} \boldsymbol{\alpha}^\dagger \mathbf{A} \boldsymbol{\alpha})$, where \mathbf{H} and \mathbf{A} are $2(N + 1) \times 2(N + 1)$ matrices. The covariance matrix $\boldsymbol{\Lambda} = -\tanh(\frac{\mathbf{A}}{2})$ evolves by Eq. 4.11.

The state of our system \mathcal{S} is fully characterised by the expectation values of $\hat{\sigma}_1^z = \hat{a}_1 \hat{a}_1^\dagger - \hat{a}_1^\dagger \hat{a}_1$ and $\hat{\sigma}_1^+ = \hat{a}_1$. In the new basis, $\hat{\sigma}_1^+$ is given by $\hat{a}_1 (\hat{a}_{N+1} - \hat{a}_{N+1}^\dagger)$ while $\hat{\sigma}_1^z$ is unchanged. These expectation values can be found from $\boldsymbol{\Lambda}$ by

$$\langle \hat{\sigma}_1^+ \rangle = \frac{1}{2} (\Lambda_{N+1, N+2} - \Lambda_{2(N+1), N+2}) \quad (4.23)$$

$$\langle \hat{\sigma}_1^z \rangle = \Lambda_{N+2, N+2}. \quad (4.24)$$

In conclusion, the Jordan-Wigner transformation and Gaussian formalism allows us to simulate a quantum XY spin model, with an arbitrary Hamiltonian acting on the first spin, in an efficient way. We simply determine the matrix \mathbf{H} and the initial covariance matrix $\boldsymbol{\Lambda}(0)$, either from \mathbf{A} or as a limit (see Appendices A.3 and A.4). We then evolve $\boldsymbol{\Lambda}(t)$ according to Eq. 4.11, from which we can determine the state of \mathcal{S} at any time, allowing us to efficiently and accurately study the behaviour and decoherence of \mathcal{S} in a wide range of situations.

4.3 Results: Decoherence of a Single-Spin System

4.3.1 Decoherence

Decoherence refers to the irreversible disappearance of quantum coherence from a system, which is the source of many quantum phenomena. It is due to the (unavoidable) interaction of the system with its

environment, which results in the system being entangled with it. Decoherence is frequently associated with the transmission of information from the system to the environment, as well as the emergence of classical behaviour from quantum systems [21].

Several quantities describing decoherence exist in the literature; in this paper we use two different measures: the Loschmidt echo \mathcal{L} and the purity \mathcal{P} . If the system is composed of a single qubit, initially pure and coupled to an environment which is in the ground state, then the off-diagonal elements ρ_{eg} of the system's density matrix decay according to

$$\rho_{eg}(t) = \rho_{eg}(0)e^{i\omega_1^z t}D(t) \quad (4.25)$$

where ω_1^z is a factor related to the system Hamiltonian and $D(t)$ is the decoherence factor [13]. In this case the diagonal elements are independent of time. Here we have assumed that the system Hamiltonian commutes with both the interaction Hamiltonian and the environment Hamiltonian. The Loschmidt echo is then given by

$$\mathcal{L}(t) = |D(t)|^2 = \left| \frac{\rho_{eg}(t)}{\rho_{eg}(0)} \right|^2. \quad (4.26)$$

In this work we assume that the system is initially in the state $\hat{\rho}_S(0) = \frac{1}{2}(\mathbb{I}_2 + \mathbf{n} \cdot \hat{\boldsymbol{\sigma}}_1)$ with $|\mathbf{n}| \leq 1$. Straightforward algebra shows that the purity is then given by

$$\mathcal{P}(t) = \text{tr } \hat{\rho}_S^2 = \frac{1}{2}(1 + |\mathbf{n}|). \quad (4.27)$$

The Loschmidt echo ranges from 0 (maximally decohered) to 1 (fully coherent), while the purity ranges from 0.5 (maximally decohered) to 1 (fully coherent). Both measures can be used to characterise the same quantity, since there exists a strictly increasing function relating them. In other words, if both Eqs. 4.26 and 4.27 are valid, and if t_1 and t_2 are any two positive times, then [15]

$$\mathcal{L}(t_1) > \mathcal{L}(t_2) \iff \mathcal{P}(t_1) > \mathcal{P}(t_2). \quad (4.28)$$

In this paper we mostly focus on pure dephasing (see especially Sec. 4.3.2). In this regime both the Loschmidt echo and the purity are easy to compute from the covariance matrix, and thus from Eq. 4.28 observations that apply to one measure immediately apply to the other. We also look at other forms of decoherence, notably in Sec. 4.3.3, where the Loschmidt echo is more difficult to compute, and thus the purity is used instead. In this work, we use the purity as our measure of decoherence except when otherwise indicated.

4.3.2 Critical Points in Decoherence Behaviour

We firstly study our spin chain (described in Sec. 4.2.1) with the parameters selected to match the XY model from [13] (Rossini *et al.*, 2007) as closely as possible. Thus, we have $\omega_1^z = -1$, $\omega_1^x = \omega_1^y = 0$, $\lambda^z = 0.25$, $\lambda_1^+ = \lambda_1^- = 0$ and $\lambda_n^+ = \lambda_n^- = -1$ ($n \neq 1$), with w_n^z ($n \neq 1$) being the independent variable. With these parameters, only two differences remain between our spin chain and Rossini *et al.*'s. Firstly, our system is coupled to the environment via $\hat{\sigma}_2^\pm$, whereas theirs is coupled via $\hat{\sigma}_2^z$. Secondly, we use open boundary conditions, whereas they use closed boundary conditions. Here we use the same measure of decoherence as Rossini *et al.*, i.e. the Loschmidt echo.

Our results are quite similar to Rossini *et al.*'s; in particular, the shape of the decoherence curves are very similar (Fig. 8). When the independent variable (λ in [13], ω_n^z in this work) is below 1 in magnitude, the decoherence has a sinusoidal behaviour, with the average value, frequency and amplitude decreasing as $|\omega_n^z| \rightarrow 1$. When $|\omega_n^z| > 1$, the decoherence has a decaying sinusoidal curve, with a rate of decay proportional to $|\omega_n^z|$. These observations suggest a critical point at $|\omega_n^z| = 1$, which is also the conclusion made in [13]. This hypothesis is supported by the analysis of both the minimum and long-time Loschmidt echo, which show non-smooth behaviour at the critical point (Fig. 9). Furthermore, the long-time Loschmidt echo is smallest at the critical point, which agrees with results from [13, 15]. However, when fitting a Gaussian decay to the short-time Loschmidt echo behaviour,

$$\mathcal{L}(t) \sim e^{-\alpha t^2}, \quad (4.29)$$

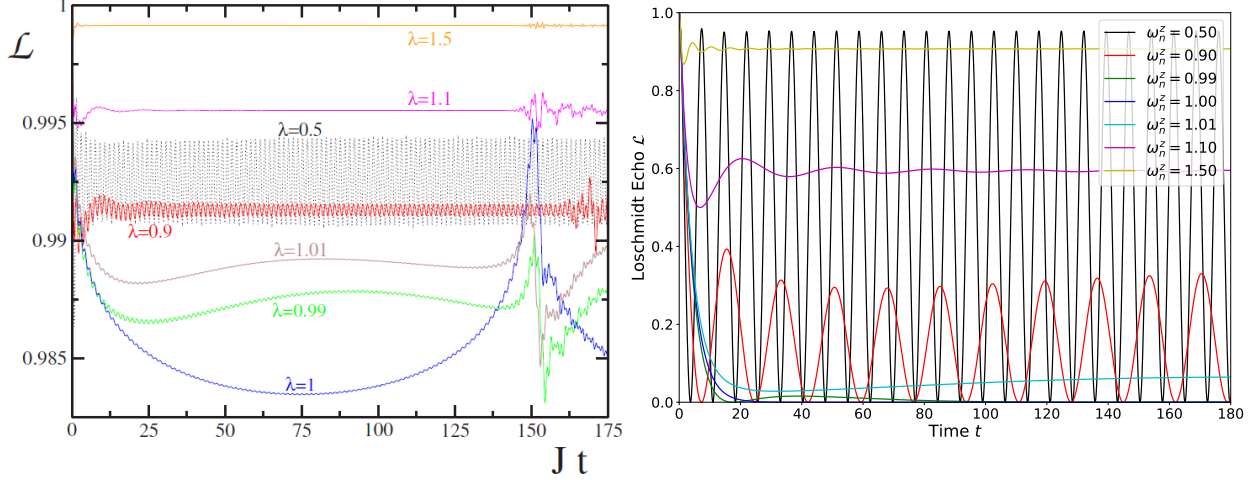


Fig. 8: **Left:** Loschmidt echo $\mathcal{L}(t)$ for Rossini *et al.*'s single-link Ising model [13]. **Right:** Loschmidt echo for our model with parameters selected to match Rossini *et al.*'s as much as possible: $\omega_1^z = -1$, $\lambda_1^z = 0.25$, $\lambda_n^+ = \lambda_n^- = -1$ ($n \neq 1$), $\lambda_1^+ = \lambda_1^- = \omega_1^x = \omega_1^y = 0$. The initial state is a random pure state, $\beta = 1000$ and the chain size is $N = 300$.

the parameter α does not show a stark change in behaviour at the critical point, in contrast to [13]. There are also two other major differences: firstly, \mathcal{L} varies from 0 to 1 in our work but from ~ 0.983 to 1 in [13]; and secondly, finite size effects appear after $t = 300$ in our work but only after $t = 150$ in [13]. We believe that the differences in the range of \mathcal{L} are due to the different types of couplings between the system and the environment. It is likely that the differences in the finite-size effects are a result of the type of boundary conditions (open versus closed) — in particular, see Sec. 4.3.3.

In addition to analysing this example, we have explored the parameter space of the model to look for other interesting phenomena, and to try to understand the effect of the different parameters on the decoherence.

- We found that the decoherence is independent of ω_1^z , including the critical point and the behaviour of the Loschmidt echo and purity above and below the critical point. In particular, when $\omega_n^z = 0$ ($n \neq 1$) and $\omega_1^x = \omega_1^y = \lambda_1^+ = \lambda_1^- = 0$, the decoherence obeys a perfect sinusoidal shape that is independent of

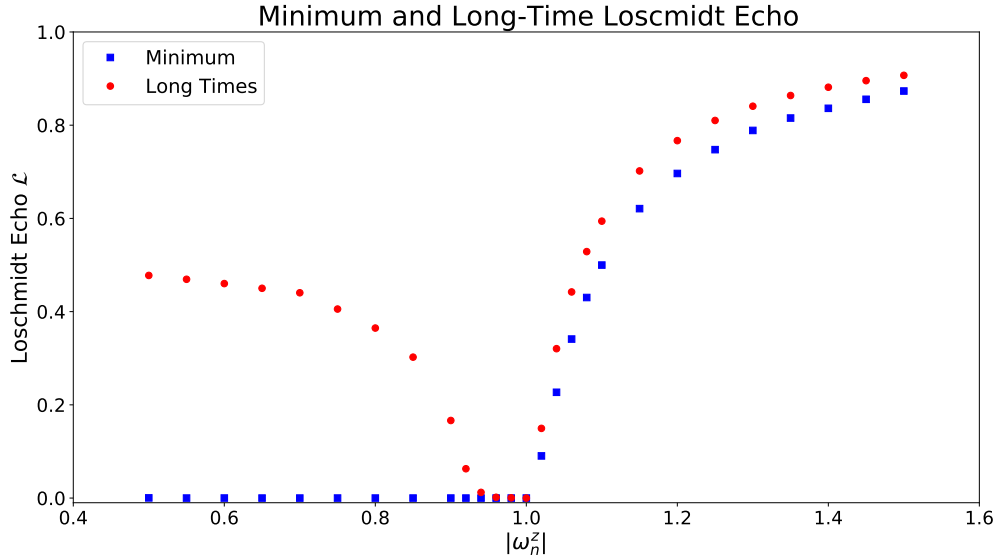


Fig. 9: Minimum and long-time Loschmidt echo $\mathcal{L}(t)$ showing a critical point at $|\omega_n^z| = 1$ with the following parameters: $\omega_1^z = -1$, $\lambda_1^z = 0.25$, $\lambda_n^+ = \lambda_n^- = -1$ ($n \neq 1$), $\lambda_1^+ = \lambda_1^- = \omega_1^x = \omega_1^y = 0$. The initial state is a random pure state, $\beta = 1000$ and the chain size is $N = 300$. For the long-time \mathcal{L} for oscillating curves, the average is plotted.

ω_1^z and whose frequency is directly proportional to λ^z .

- At first sight the critical point seems to be independent of λ^z , even when this parameter is imaginary. λ^z only appears to modify the horizontal and vertical scales of the decoherence curves when real, and when imaginary also adds small oscillations to these curves.
- The location of the critical point is highly dependent on the values of λ_n^+ and λ_n^- for $n \neq 1$. In this work, we have only studied the case where all λ_n^+ ($n \neq 1$) are the same and all λ_n^- ($n \neq 1$) are the same (but λ_n^+ and λ_n^- can be different). Our runs suggest that the critical point is always at the largest magnitude of λ_n^+ or λ_n^- , even when one or both of these take imaginary values.
- In each case, there is a stark change in the behaviour of the long-time purity (and sometimes also the minimum purity) at the critical point, in a similar way to Fig. 9. The long-time purity is always smallest at the critical point, in agreement with results in [15].

We note that the above sets of parameters all make the Hamiltonians $\hat{H}_{\mathcal{E}}$ and $\hat{V}_{S\mathcal{E}}$ commute with the system Hamiltonian \hat{H}_S . We have also run simulations for parameters which make the Hamiltonians non-commuting, but in these cases the behaviour of the decoherence is messy and it is difficult to obtain any useful information.

We emphasise that the above conclusions are made based on relatively few tests, due to time constraints and the sheer size of the parameter space that there is to explore. Thus, one should be careful before assuming that these conclusions hold in the general case. Furthermore, it is quite likely that there exist very interesting behaviours of the decoherence for certain combinations of parameters that are not mentioned in this paper.

4.3.3 Temperature Dependence and Finite-Size Effects

We study also the temperature dependence of the behaviour of the decoherence. In all simulations in which we found the purity to depend on the inverse temperature β , we saw the dependence virtually disappear for $\beta > 30$. This is likely because the temperature is $T = \frac{1}{\beta}$ (in natural units), so large values of β are equivalent to small temperatures T , thus leading to only small changes in the behaviour of the spin

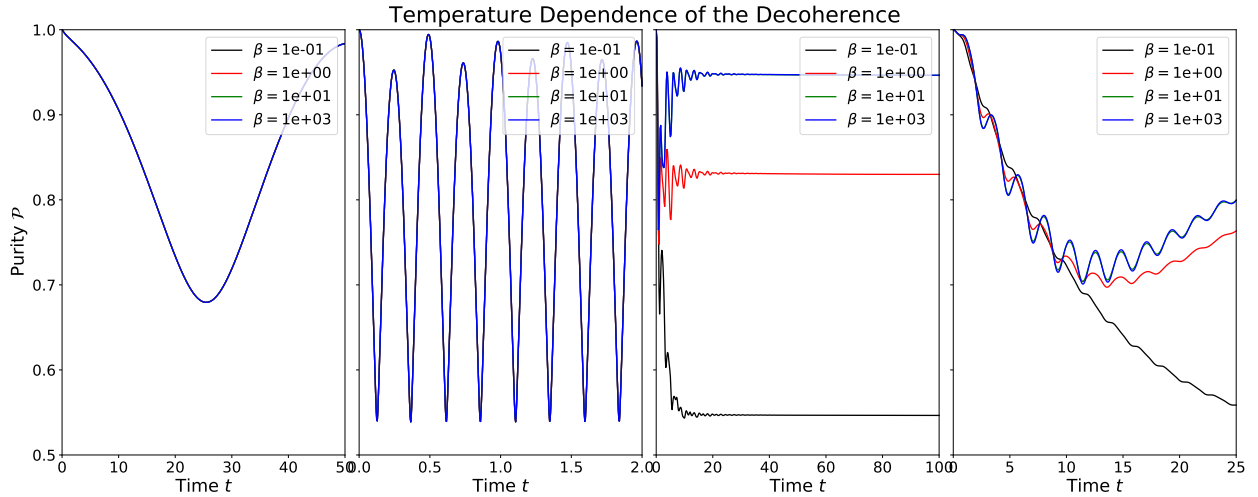


Fig. 10: Purity plotted for four different sets of parameters at $\beta = 0.1$ (black), 1 (red), 10 (green) and 1000 (blue). A missing colour means that the decoherence is the same at higher β . \hat{H}_S commutes with $\hat{H}_{\mathcal{E}}$ and $\hat{V}_{S\mathcal{E}}$ in the first two subfigures, but not in the last two. In all cases, the spin chain is of size N and the initial state is a random pure state. Parameters:

Far Left: $\omega_1^z = 1$, $\omega_1^x = 0$, $\omega_1^y = 0$, $\lambda_1^z = 0.25$, $\lambda_1^+ = 0$, $\lambda_1^- = 0$, $\omega_n^z = -3.9$, $\lambda_n^+ = -i$, $\lambda_n^- = -4$ ($n \neq 1$).

Middle Left: $\omega_1^z = 1$, $\omega_1^x = 0.5$, $\omega_1^y = 0.5$, $\lambda_1^z = 1 + 5i$, $\lambda_1^+ = 3 + 2i$, $\lambda_1^- = -2 + 3i$, $\omega_n^z = -1$, $\lambda_n^+ = -1$, $\lambda_n^- = -1$ ($n \neq 1$).

Middle Right: $\omega_1^z = 1$, $\omega_1^x = 0$, $\omega_1^y = 0$, $\lambda_1^z = 0.25$, $\lambda_1^+ = -1$, $\lambda_1^- = 1$, $\omega_n^z = -1$, $\lambda_n^+ = -1$, $\lambda_n^- = -1$ ($n \neq 1$).

Far Right: $\omega_1^z = 1$, $\omega_1^x = -1$, $\omega_1^y = 0$, $\lambda_1^z = 0.25$, $\lambda_1^+ = 0$, $\lambda_1^- = 0$, $\omega_n^z = -1$, $\lambda_n^+ = -1$, $\lambda_n^- = -1$ ($n \neq 1$).

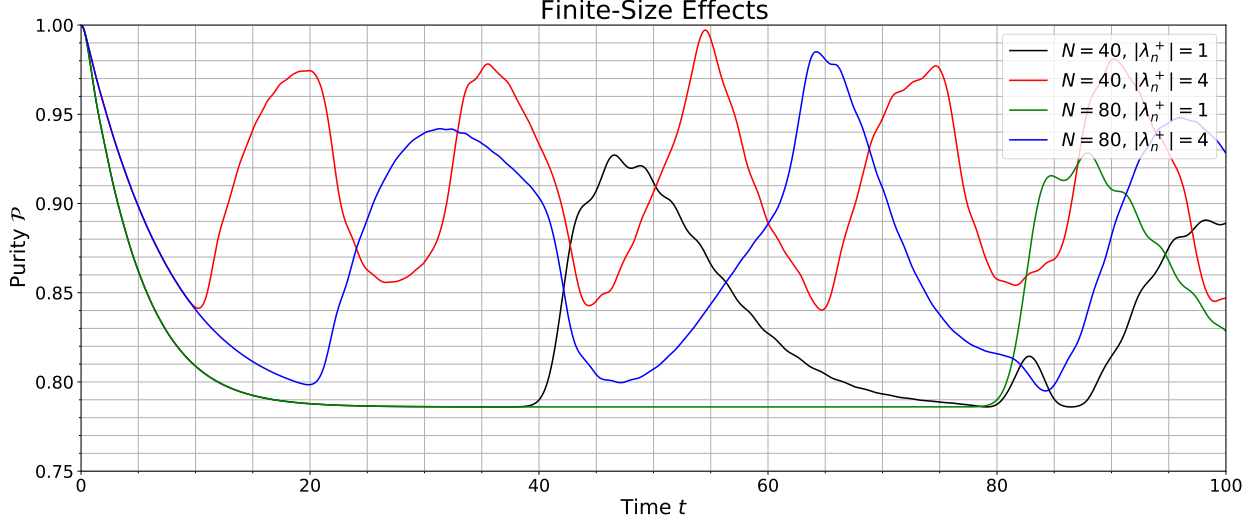


Fig. 11: Finite-size effects in the decoherence. Notice that for identical λ_n^+ ($n \neq 1$), the purity is identical until finite-size effects appear at $t^* \propto N$. The initial state is a random pure state; β is set to 1000. The other parameters are: $\omega_1^z = 1$, $\omega_1^x = 0$, $\omega_1^y = 0$, $\lambda_1^z = 0.25$, $\lambda_1^+ = 0$, $\lambda_1^- = 0$, $\omega_n^z = -1$, $\lambda_n^- = -1$ ($n \neq 1$).

chain. When temperature dependence is observed, the purity is seen to be lower for higher T (lower β), which is the result one would expect from thermodynamical arguments (Fig. 10).

Moreover, we find that temperature dependence only exists when the system Hamiltonian \hat{H}_S does not commute with both the interaction Hamiltonian \hat{V}_{SE} and the environment Hamiltonian \hat{H}_E . It is clear from Eqs. 4.2 and 4.3 that \hat{H}_S always commutes with \hat{H}_E in our model, so the only constraint is that $[\hat{H}_S, \hat{V}_{SE}] = 0$. In Fig. 10 we plot the purity of the system at various temperatures for four different sets of parameters: two commuting cases and two non-commuting cases. A consequence of this result is that in order to use a single-qubit system to probe the temperature of its environment, as in [22], one needs to use an interaction Hamiltonian that does not commute with the system Hamiltonian. However, we have seen from several simulations that the decoherence in this case follows a pattern that is difficult to predict, so a careful choice of the parameters will be required to achieve accurate results with such a spin chain. We note, however, that there exist other models (i.e. other types of couplings) where the system Hamiltonian commutes with both the interaction Hamiltonian and the environment Hamiltonian but the decoherence is still temperature dependent.

The number of spins N in the environment is also an important quantity to consider. We found that, not surprisingly, the decoherence of the system as a function of time is independent of N , up until finite-size effects start influencing the chain. The time t^* at which these effects become apparent are directly proportional to N , as expected, but also depend on the magnitudes of λ_n^+ and λ_n^- , as shown in Fig. 11. Our results suggest that t^* is inversely proportional to the larger of the magnitudes of λ_n^+ and λ_n^- ,

$$t^* \approx \frac{N}{\max\{|\lambda_n^+|, |\lambda_n^-|\}}, \quad (4.30)$$

although we have only considered the cases where all λ_n^+ ($n \neq 1$) are equal and all λ_n^- ($n \neq 1$) are equal. Thus, one could view the magnitudes of λ_n^+ and λ_n^- as speeds of transmission of information about the system. At the end of the spin chain, this information would ‘bounce’ back and interfere with previous spins, causing finite-size effects. We note also that we use open boundary conditions; if we had used closed boundary conditions, t^* would have been halved, as the information would have travelled in two directions and interfered halfway. This is seen in the results of [13].

4.3.4 Further Research

One of the major steps left to do is to continue exploring the parameter space for our spin chain and to better understand the influence of each term in the model on the decoherence of the system. In this paper we have modified the parameters ‘in bulk’, i.e. all ω_n^z are given the same value (except for $n = 1$), and similarly for λ_n^+ and λ_n^- . It would be interesting to study the case where these parameters are adjusted individually, for example by adding a random perturbation for each n , or by using non-constant patterns like alternating signs or linearly increasing values. A better understanding of the model may also help to predict critical points and even to optimise initial conditions for certain machine-learning algorithms, which, given the size of the parameter space, could greatly reduce run times.

Furthermore, it is not known why the decoherence is temperature independent when the system Hamiltonian commutes with the environment and interaction Hamiltonians. As this equivalence is not true for general spin chains, it would be interesting to pin down from which characteristics of the model this relation originates. We note that we have only looked at initially pure states in this paper ($|\mathbf{n}| = 1$). Our model is also perfectly applicable to initially mixed states, and investigating the decoherence of such states could also be a very interesting problem. Moreover, there are several possible applications of the study of the decoherence of a single-spin system, including quantum thermometry [22], remote parameter estimation [23] and quantum control (Sec. 4.4).

4.4 Machine learning for control of decoherence

Now that a framework that can simulate many hundreds of interacting qubits on a classical computer has been achieved, we turn back to the application of quantum control. Given the power of the previous demonstration of Sec. 3.3, a natural question we tackled next was: can we use machine learning to find an optimal initial magnetic field for the Fermionic Hamiltonian representation such that the decoherence of the system is minimised with respect to its environment? Indeed, there are a few approaches we could take this in, each with practical application. We have opted to use $\text{tr}(\hat{\rho}^2)$ as a measure of decoherence of the system and hence an inverse ‘cost function’ for the machine learning algorithm to attempt to maximise. Similarly to Sec. 3.3, we will build the algorithm to obtain as close as possible to $\text{tr}(\hat{\rho}^2) = 1$, corresponding to a completely pure, coherent state. The magnetic field parameters ω_n^z , ω_n^+ and ω_n^- were fed to the algorithm as features for the gradient descent to optimise coherence. $\text{tr}(\hat{\rho}^2)$ is found at each time-step as in Eq. 4.27. This approach allows us to easily apply gradient descent to the framework that has already been developed in this section. This static approach we took of setting the initial magnetic field as fixed and allowing the system to evolve left an open question of how best to use $\text{tr}(\hat{\rho}^2)$ as a cost function. Evaluating it only at the end state was considered, as was taking an unweighted average of $\text{tr}(\hat{\rho}^2)$ at all time steps. We identified both to have shortcomings however, as the former would encourage the algorithm to only seek a pure state at the final time-step and potentially allow the state to be completely mixed for the entirety of the time prior. The latter does not convey the information that having a more pure state at a later time is more valuable than at an earlier time. To compensate for this, the cost function was taken to be a weighted average of $\text{tr}(\hat{\rho}^2)$ at all points in time:

$$\text{Cost} = \frac{\sum_i w_i \text{tr}(\hat{\rho}_i^2)}{\sum_i w_i}, \quad (4.31)$$

where w_i are the weights given at each time-step i . Indeed, this is a discrete sum only due to the nature of the numerical modelling we are carrying out. If we were to consider the case of continuous time-steps, this would become an integral equation:

$$\text{Cost} = \frac{\int w(t) \text{tr}[\hat{\rho}^2(t)] dt}{\int w(t) dt}. \quad (4.32)$$

The choice of the weights w_i is a topic of contention here. Indeed, it depends how important one considers purity at a later time-step to be compared to an earlier time-step. For this paper, we have chosen the w_i to be a monotonically increasing array from 1 to 2, i.e. valuing the $\text{tr}(\hat{\rho}^2)$ at the end of evolution twice as much as at the beginning.

Here, we have chosen all coupling $\lambda_n^\pm = \lambda_1^z = 1$ to allow free propagation of information through the chain. The initial Bloch vector was set to a pure state of $(n_x, n_y, n_z) = (1, 0, 0)$. The magnetic field terms

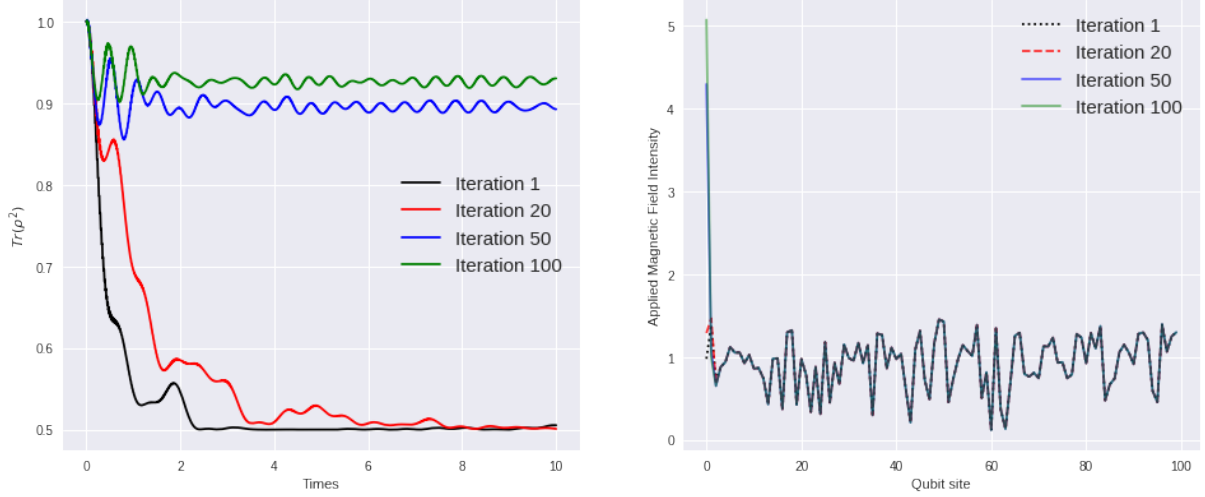


Fig. 12: On the left is a plot of $\text{tr}(\hat{\rho}^2)$ against time for multiple different iterations of the gradient descent algorithm at temperature $T=0.01\text{K}$. To the right is a plot of the intensity of the magnetic field profile at each qubit site for multiple iterations of the gradient descent algorithm.

ω_n were randomised initially and then fed to the gradient descent algorithm as features to optimise $\text{tr}(\hat{\rho}^2)$. Below are results for temperature $T = 0.01\text{ K}$ and a learning rate of 10 in Fig. 12.

From Fig. 12, we can see a sizeable improvement on the purity of the state that the gradient descent algorithm has managed to achieve. The plot on the right reveals that the algorithm has maintained the magnetic field intensity on the environment qubits and continuously increased the magnetic field strength on the $n = 1$ system qubit to maintain purity.

These results are reasonable and in agreement with the discussions of the Fermionic representation in previous parts of this paper. This framework may easily be adapted and applied in new, exciting directions. It may be used in a higher-temperature environment to search for a configuration that preserves the decay of purity and coherence of the system — possibly having application in practical quantum information science. Additionally, one could take a different approach of allowing the algorithm to change the magnetic field strength in time, essentially changing the time-independent Hamiltonian to a time-dependent one. This may give more flexibility in how purity could be maintained in the state.

Additionally, it seems prevalent to state that reinforcement learning is likely a more natural candidate for the outcome of this problem than in Sec. 3.3. This is not something that time allowed us to explore, but it would be possible to build a reward function for the reinforcement algorithm that rewards more heavily for purity maintained at a later time as we have done here with our weighted cost function, but while also allowing for more nuance. It would be possible to incentivise desirable criteria such as rewarding more heavily the longer the algorithm maintains a state of (semi) purity without breaking into a mixed state.

5 Conclusion

In this paper we looked at three different numerical techniques for solving many-body quantum problems. We focused in particular on simulating spin chain dynamics for several different configurations, including the two-dimensional Ising and Potts models and a generalised form of the one-dimensional quantum XY model.

We found from Monte Carlo simulations that the two-dimensional Ising model has a critical point at a temperature $T_C \sim 2.5$ when the coupling constant $J = 1$. The average energy per spin is -2 at very low temperatures ($T < 1$), and increases with temperature at a rate that grows with the size of the lattice. At $T = 6$, we found the average energy per spin to be approximately -0.35 for a lattice length $L = 20$. These observations are all consistent with the expected values [17], and demonstrate that Monte Carlo algorithms are usable to solve problems such as the Ising model. We emphasise that deviations from the expected

average-energy curve are larger for smaller lattice lengths L ; however as we are usually interested in the large L limit, this rarely affects the results. We also applied Monte Carlo methods to the Potts model with 10 degrees of freedom, where we found a critical temperature of $T_C \approx 0.72$. The critical point is clearly indicated by a very large spike in the heat capacity at T_C . These results also agree with the literature, e.g. [5].

We used a gradient-descent type of machine-learning algorithm to study the quantum control of a qubit by a time-dependent magnetic field. We found that using the fidelity as our cost function led to excellent results within a relatively small number of iterations (1000). Moreover, we observed a strong correlation between the amount of time T given to the system to evolve and the maximum fidelity reached. In particular, we found an approximately linear relation up to $T \sim 1$, where the maximum fidelity is almost 1. Beyond, the maximum fidelity increases more slowly up to $T \sim 1.4$, where it equals 1.

We generalised the one-dimensional quantum XY model by adding an arbitrary Hamiltonian to the first spin, which we considered as our system. We applied the Jordan-Wigner transformations, as well as a method due to Colpa [19], to use the Gaussian formalism with the fermionic creation and annihilation operators. This allowed us to study chains with hundreds of spins.

We found that different types of system-environment couplings can greatly increase or decrease the decoherence of the system. We compared our results to those in [13] with identical parameters but with a different system-environment coupling ($\hat{\sigma}_2^\pm$ instead of $\hat{\sigma}_2^z$) and with open (instead of closed) boundary conditions. We found that the different coupling decreased the Loschmidt echo \mathcal{L} from a range of $\sim 0.983 - 1$ in [13] to a range of $0 - 1$ for our model. However, the curves of the Loschmidt echo remain very similar in shape, and there exists a critical point at the same values of the parameters, in both cases characterised by non-smooth behaviour in the minimum and long-time \mathcal{L} at this point. Importantly, we found that the critical point depends strongly on the coupling constants λ_n^\pm , when the initial state is pure.

We also studied the dependence of the behaviour of the decoherence (measured by the purity) on the environment's temperature and on finite-size effects. We observed that the decoherence is temperature independent whenever the system Hamiltonian commutes with both the environment and interaction Hamiltonians. Unfortunately we do not know the reason for this equivalence; however we know that it is not true of all spin chains and that it must arise from some of the properties of our model. We also noticed that finite-size effects start becoming visible at a time that is proportional to the chain size and inversely proportional to the magnitude of the environment couplings. For chains with closed boundary conditions, this time is halved compared with open boundary conditions.

We applied machine-learning techniques to study quantum control for our generalised quantum XY model. We found after 100 iterations of a gradient-descent algorithm that the highest long-time purity is achieved when the magnetic field of the system is increased to a much larger value than for the environment, which is an expected result. In fact, it seems that the value of the magnetic field on the environment has little effect on the system's purity. Despite good initial results, we note that reinforcement learning is likely to be better suited to this problem than gradient descent, as it would allow for more complex cost functions.

We emphasise that there are still many open questions regarding our generalisation of the one-dimensional quantum XY model. In particular, we may have missed some very interesting decoherence behaviour for a certain set of parameters. Moreover, we have only looked at initially pure states, but the model also supports the system to be initially mixed. As mentioned previously, we also do not know why the decoherence is independent of temperature when the system Hamiltonian commutes with the total Hamiltonian.

6 Acknowledgements

This research was undertaken by its authors during a summer research studentship at the School of Theoretical Physics, Dublin Institute for Advanced Studies (DIAS). We would like to thank DIAS for giving us this fantastic opportunity, in particular all the staff at DIAS who helped to organise the studentship, everyone who gave us many interesting insights into their fields of research, and our supervisors Luuk Coopmans, Shane Dooley and Samuel Kováčik, the last of whom, along with Denjoe O'Connor, coordinated the studentship.

References

- [1] A. Scott. Computational complexity: Why quantum chemistry is hard. *Nature Physics*, 5(10):707–708, Oct 2009.
- [2] Foulkes, W.M.C., Mitas, L., Needs, R.J. and Rajagopal, G. Quantum monte carlo simulations of solids. *Rev. Mod. Phys.*, 73(1):33–83, Jan 2001.
- [3] Pudliner, B.S., Pandharipande, V.R., Carlson, J., Pieper, S.C. and Wiringa, R.B. Quantum Monte Carlo calculations of nuclei with $A < 7$. *Phys. Rev. C*, 56(4):1720–1750, Oct 1997.
- [4] Rubtsov, A.N., Savkin, V.V. and Lichtenstein, A.I. Continuous-time quantum monte carlo method for fermions. *Phys. Rev. B*, 72(3):035122, Jul 2005.
- [5] Challa, M.S.S., Landau, D.P. and Binder, K. Finite-size effects at temperature-driven first-order transitions. *Phys. Rev. B*, 34(3):1841–1852, Aug 1986.
- [6] D. Baron. Machine learning in astronomy: a practical overview. *arXiv: Instrumentation and Methods for Astrophysics*, 2019.
- [7] El Naqa, I. and Das, S. The role of machine and deep learning in modern medical physics. *Medical Physics*, 47(5):e125–e126, 2020.
- [8] M.R. Hush. Machine learning for quantum physics. *Science*, 355(6325):580–580, 2017.
- [9] Harney, C., Pirandola S., Ferraro, A. and Paternostro, M. Entanglement classification via neural network quantum states. *arXiv: Quantum Physics*, 2019.
- [10] Bukov, M., Day, A.G.R., Sels, D., Weinberg, P., Polkovnikov, A. and Mehta, P. Reinforcement learning in different phases of quantum control. *Physical Review X*, 8:031086, 2018.
- [11] S. Ruder. An overview of gradient descent optimization algorithms. *ArXiv*, abs/1609.04747, 2016.
- [12] M. Giles. Ibm’s new 53-qubit quantum computer is the most powerful machine you can use. <https://www.technologyreview.com/2019/09/18/132956/ibms-new-53-qubit-quantum-computer-is-the-most-powerful-machine-you-can-use/>. Accessed: 28/07/2020.
- [13] Rossini, D., Calarco, T., Giovannetti, V., Montangero, S. and Fazio, R. Decoherence induced by interacting quantum spin baths. *Physical Review A*, 75(3), Mar 2007.
- [14] Damski, B., Quan, H.T. and Zurek, W.H. Critical dynamics of decoherence. *Phys. Rev. A*, 83(6):062104, Jun 2011.
- [15] Quan, H.T., Song, Z., Liu, X.F., Zanardi, P. and Sun, C.P. Decay of loschmidt echo enhanced by quantum criticality. *Phys. Rev. Lett.*, 96(14):140604, Apr 2006.
- [16] R.M. Neal. Mcmc using hamiltonian dynamics. *The Handbook of Markov Chain Monte Carlo*, 2011.
- [17] J. Kotze. Introduction to monte carlo methods for an ising model of a ferromagnet. *The Handbook of Markov Chain Monte Carlo*, 2008.
- [18] F. Bloch. Nuclear induction. *Phys. Rev.*, 70:460–474, Oct 1946.
- [19] J.H.P. Colpa. Diagonalisation of the quadratic fermion hamiltonian with a linear part. *Journal of Physics A: Mathematical and General*, 12(4):469, 1979.
- [20] Kraus, C.V., Wolf, M.M., Cirac, J.I. and Giedke, G. Pairing in fermionic systems: A quantum-information perspective. *Phys. Rev. A*, 79:012306, Jan 2009.
- [21] W.H. Zurek. Decoherence and the transition from quantum to classical. *Physics Today*, 44(10):36, 1991.

- [22] Razavian, S., Benedetti, C., Bina, M., Akbari-Kourbolagh, Y. and Paris, M.G.A. Quantum thermometry by single-qubit dephasing. *Eur. Phys. J. Plus*, 134(284), 2019.
- [23] Kiukas, J., Yuasa, K. and Burgarth, D. Remote parameter estimation in a quantum spin chain enhanced by local control. *Phys. Rev. A*, 95(5):052132, May 2017.

A Fermionic Representation

A.1 Schrödinger-type Equation for the Covariance Matrix

A derivation of Eq. 4.11.

$$\frac{d}{dt}[\Lambda]_{jk} = \text{tr} \left([\hat{\alpha}_k^\dagger, \hat{\alpha}_j] \frac{d}{dt} \hat{\rho}(t) \right) = -\frac{i}{2} \sum_{n,m} H_{nm} \text{tr} \left([[\hat{\alpha}_k^\dagger, \hat{\alpha}_j], \hat{\alpha}_n^\dagger \hat{\alpha}_m] \hat{\rho}(t) \right) \quad (\text{A.1})$$

$$= i \sum_{n,m} H_{nm} \text{tr} \left[\left(-\delta_{m+N,j} \hat{\alpha}_n^\dagger \hat{\alpha}_k^\dagger + \delta_{mk} \hat{\alpha}_n^\dagger \hat{\alpha}_j - \delta_{nj} \hat{\alpha}_k^\dagger \hat{\alpha}_m + \delta_{n+N,k} \hat{\alpha}_j \hat{\alpha}_m \right) \hat{\rho} \right] \quad (\text{A.2})$$

$$= i \left(\sum_{n=1}^{2N} H_{nk} \Lambda_{jn} - \sum_{n=1}^{2N} H_{jn} \Lambda_{nk} \right) \quad (\text{A.3})$$

$$= -i[\mathbf{H}, \Lambda]_{jk}. \quad (\text{A.4})$$

A.2 Derivation of Eq. 4.17

We would like to express our initial state $\hat{\rho}(0)$ in terms of fermionic operators, and as an exponential, in order to bring it closer to the Gaussian form. This method is valid only when $0 < |\mathbf{n}| < 1$, which we assume. The eigenvalues of $\hat{\rho}_S(0) = \frac{1}{2}(\mathbb{I}_2 + \mathbf{n} \cdot \hat{\boldsymbol{\sigma}}_1)$ are $\frac{1}{2}(1 \pm |\mathbf{n}|) = \exp\{\ln[\frac{1}{2}(1 \pm |\mathbf{n}|)]\}$, with eigenstates (in density matrix form) $\frac{1}{2}(\mathbb{I}_2 \pm \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \hat{\boldsymbol{\sigma}}_1)$. We have:

$$\hat{\rho}_S(0) = \exp \left\{ \ln \left[\frac{1}{2} (1 + |\mathbf{n}|) \right] \right\} \frac{1}{2} \left(\mathbb{I}_2 + \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \hat{\boldsymbol{\sigma}}_1 \right) + \exp \left\{ \ln \left[\frac{1}{2} (1 - |\mathbf{n}|) \right] \right\} \frac{1}{2} \left(\mathbb{I}_2 - \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \hat{\boldsymbol{\sigma}}_1 \right) \quad (\text{A.5})$$

$$= \exp \left\{ \ln \left[\frac{1}{2} (1 + |\mathbf{n}|) \right] \right\} \frac{1}{2} \left(\mathbb{I}_2 + \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \hat{\boldsymbol{\sigma}}_1 \right) \cdot \exp \left\{ \ln \left[\frac{1}{2} (1 - |\mathbf{n}|) \right] \right\} \frac{1}{2} \left(\mathbb{I}_2 - \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \hat{\boldsymbol{\sigma}}_1 \right) \quad (\text{A.6})$$

$$= \exp \left\{ \ln \left[\frac{1}{2} (1 + |\mathbf{n}|) \right] \right\} \frac{1}{2} \left(\mathbb{I}_2 + \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \hat{\boldsymbol{\sigma}}_1 \right) + \ln \left[\frac{1}{2} (1 - |\mathbf{n}|) \right] \frac{1}{2} \left(\mathbb{I}_2 - \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \hat{\boldsymbol{\sigma}}_1 \right) \quad (\text{A.7})$$

$$= \exp \left\{ \frac{1}{2} \ln \left[\frac{1}{4} (1 - |\mathbf{n}|^2) \right] \right\} \mathbb{I}_2 + \frac{1}{2} \ln \left[\frac{1 + |\mathbf{n}|}{1 - |\mathbf{n}|} \right] \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \hat{\boldsymbol{\sigma}}_1 \quad (\text{A.8})$$

$$= \sqrt{\frac{1}{4} (1 - |\mathbf{n}|^2)} \exp \left\{ \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \hat{\boldsymbol{\sigma}}_1 \tanh^{-1} |\mathbf{n}| \right\}, \quad (\text{A.9})$$

where in the last equality we've used the identity $\tanh^{-1} |\mathbf{n}| = \frac{1}{2} \ln \left(\frac{1+|\mathbf{n}|}{1-|\mathbf{n}|} \right)$. Applying the Jordan-Wigner transformation:

$$\hat{\rho}_S(0) = \sqrt{\frac{1}{4} (1 - |\mathbf{n}|^2)} \exp \left\{ \frac{\tanh^{-1} |\mathbf{n}|}{|\mathbf{n}|} \left(n^z \hat{a}_1 \hat{a}_1^\dagger - n^z \hat{a}_1^\dagger \hat{a}_1 + n^- \hat{a}_1 + n^+ \hat{a}_1^\dagger \right) \right\}. \quad (\text{A.10})$$

Finally, since the exponent of Eq. A.10 and the exponent of $\hat{\rho}_\mathcal{E}^{\text{th}} = \frac{1}{\mathcal{Z}} \exp\{-\beta \hat{H}_\mathcal{E}\}$ commute, their product is given by Eq. 4.17.

A.3 Matrices

We present here the matrices \mathbf{H} and \mathbf{A} . Each has been divided into $(N+1) \times (N+1)$ blocks, with the rows and columns corresponding to the “ghost” fermion also indicated. Setting $\omega_1^\pm = \omega_1^x \pm i\omega_1^y$ and $K = \frac{\tanh^{-1}|\mathbf{n}|}{|\mathbf{n}|}$, we have

$$\mathbf{H} = \begin{bmatrix} -2\omega_1^z & \lambda_1^- & 0 & \cdots & 0 & \omega_1^+ & 0 & \lambda_1^{+*} & 0 & \cdots & 0 & -\omega_1^+ \\ \lambda_1^{-*} & -2\omega_2^z & \lambda_2^- & \cdots & 0 & \lambda_1^{z*} & -\lambda_1^{+*} & 0 & \lambda_2^{+*} & \cdots & 0 & -\lambda_1^{z*} \\ 0 & \lambda_2^{-*} & -2\omega_3^z & \cdots & 0 & 0 & 0 & -\lambda_2^{+*} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2\omega_N^z & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \hline \omega_1^- & \lambda_1^z & 0 & \cdots & 0 & 0 & \omega_1^+ & \lambda_1^{z*} & 0 & \cdots & 0 & 0 \\ 0 & -\lambda_1^+ & 0 & \cdots & 0 & \omega_1^- & 2\omega_1^z & -\lambda_1^{-*} & 0 & \cdots & 0 & -\omega_1^- \\ \lambda_1^+ & 0 & -\lambda_2^+ & \cdots & 0 & \lambda_1^z & -\lambda_1^- & 2\omega_2^z & -\lambda_2^{-*} & \cdots & 0 & -\lambda_1^z \\ 0 & \lambda_2^+ & 0 & \cdots & 0 & 0 & 0 & -\lambda_2^- & 2\omega_3^z & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 2\omega_N^z & 0 \\ \hline -\omega_1^- & -\lambda_1^z & 0 & \cdots & 0 & 0 & -\omega_1^+ & -\lambda_1^{z*} & 0 & \cdots & 0 & 0 \end{bmatrix} \quad (\text{A.11})$$

$$\mathbf{A} = \begin{bmatrix} 2Kn^z & 0 & 0 & \cdots & 0 & -Kn^+ & 0 & 0 & 0 & \cdots & 0 & Kn^+ \\ 0 & -2\beta\omega_2^z & \beta\lambda_2^- & \cdots & 0 & 0 & 0 & 0 & \beta\lambda_2^{+*} & \cdots & 0 & 0 \\ 0 & \beta\lambda_2^{-*} & -2\beta\omega_3^z & \cdots & 0 & 0 & 0 & -\beta\lambda_2^{+*} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2\beta\omega_N^z & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \hline -Kn^- & 0 & 0 & \cdots & 0 & 0 & -Kn^+ & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -Kn^- & -2Kn^z & 0 & 0 & \cdots & 0 & Kn^- \\ 0 & 0 & -\beta\lambda_2^+ & \cdots & 0 & 0 & 0 & 2\beta\omega_2^z & -\beta\lambda_2^{-*} & \cdots & 0 & 0 \\ 0 & \beta\lambda_2^+ & 0 & \cdots & 0 & 0 & 0 & -\beta\lambda_2^- & 2\beta\omega_3^z & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 2\beta\omega_N^z & 0 \\ \hline Kn^- & 0 & 0 & \cdots & 0 & 0 & Kn^+ & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \quad (\text{A.12})$$

A.4 Pure States

If our initial state $\hat{\rho}_S(0)$ is pure, i.e. its Bloch vector \mathbf{n} satisfies $|\mathbf{n}| = 1$, the state is not Gaussian, and cannot be written in the form given in Eq. 4.12, but this can easily be resolved. Fixing $\hat{\rho}_E$, the entries of $\mathbf{\Lambda}$ are continuous functions of $\hat{\rho}_S$. Therefore we consider our pure state, $\hat{\rho}_S$, as a limit of mixed states, $\hat{\rho}_S^{(j)}$. Each of these states can, by assumption, be written in the form

$$\hat{\rho}^{(j)} = \hat{\rho}_S^{(j)} \otimes \hat{\rho}_E = \frac{1}{Z} \exp\left(-\frac{1}{2}\boldsymbol{\alpha}^\dagger \mathbf{A}^{(j)} \boldsymbol{\alpha}\right), \quad (\text{A.13})$$

as in Eq. 4.12. For each state $\hat{\rho}^{(j)}$ we consider the corresponding covariance matrix given by $\mathbf{\Lambda}^{(j)} = -\tanh(\mathbf{A}^{(j)}/2)$. As the construction of $\mathbf{\Lambda}$ from a given $\hat{\rho}$ is continuous, we have $\mathbf{\Lambda} = \lim_{j \rightarrow \infty} \mathbf{\Lambda}^{(j)}$.

For any mixed state $\hat{\rho}_S$ with Bloch vector \mathbf{n} , we see that A is (after a permutation) block diagonal (see Appendix A.3); specifically rows and columns 1, $N+1$, $N+2$ and $2(N+1)$ form a 4×4 matrix separate from the other $2N-2$ rows and columns. We can calculate the tanh of each part of the matrix separately. We can write this 4×4 matrix as

$$A_S = \frac{\tanh^{-1}|\mathbf{n}|}{|\mathbf{n}|} \begin{bmatrix} 2n^z & -n^+ & 0 & n^+ \\ -n^- & 0 & -n^+ & 0 \\ 0 & -n^- & -2n^z & n^- \\ n^- & 0 & n^+ & 0 \end{bmatrix}. \quad (\text{A.14})$$

We then calculate $\Lambda_{\mathcal{S}} = -\tanh \frac{A_{\mathcal{S}}}{2}$ analytically, obtaining

$$\Lambda_{\mathcal{S}} = \begin{bmatrix} -n^z & \frac{n^+}{2} & 0 & -\frac{n^+}{2} \\ \frac{n^-}{2} & 0 & \frac{n^+}{2} & 0 \\ 0 & \frac{n^-}{2} & n^z & -\frac{n^-}{2} \\ -\frac{n^-}{2} & 0 & -\frac{n^+}{2} & 0 \end{bmatrix}. \quad (\text{A.15})$$

It follows from continuity that this expression for $\Lambda_{\mathcal{S}}$ holds in the limit $\hat{\rho}^{(j)} \rightarrow \hat{\rho}$. The remaining entries of the matrix are then given by $-\tanh(\beta\hat{H}_{\mathcal{E}}/2)$, which is independent of $\hat{\rho}_{\mathcal{S}}^{(j)}$. In conclusion, we can still determine $\Lambda(0)$. Although our state is not Gaussian, Eq. 4.11 still holds, and the state of our system \mathcal{S} at time t can nonetheless be extracted from $\Lambda(t)$.