# Adiabatic Gibbs State Preparation through Quantum Linear Algebra

Koreman, Tim

# Adiabatic Gibbs State Preparation through Quantum Linear Algebra

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
THEORETICAL PHYSICS


| | |
|---|---|
| Author : | Tim Koreman |
| Student ID : | s2418541 |
| Supervisor : | dr. V. Dunjko |
| In collaboration with: | C. Gyurik MSc. |
| | Y. Herasymenko MSc. |
| Second corrector : | dr. J. Tura Brugués |


Leiden, The Netherlands, July 29, 2021

# Adiabatic Gibbs State Preparation through Quantum Linear Algebra

**Tim Koreman**

Lorentz Institute for Theoretical Physics, Leiden University
P.O. Box 9500, 2300 RA Leiden, The Netherlands

July 29, 2021

## Abstract

In this work we present a novel quantum algorithm to sample from the Gibbs distribution of a classical system. The algorithm consists of directly preparing an encoding of the $+1$ eigenstate of a stochastic matrix $P$ via adiabatic state preparation. The key mechanism that is investigated is a state encoding such that measuring the state proportional to the Gibbs distribution at inverse temperature $\beta$ is equivalent to sampling from the Gibbs distribution at $2\beta$. We present analytical observations and numerical analysis of the time complexity of this algorithm and compare it to classical mixing times for a specific spin model and find at most a quadratic speedup under some assumptions. We also extend the numerical results and give more analytical observations for algorithms found in Orsucci (2019) [1] for a similar problem to higher spin numbers by using a symmetric subspace projection and find that the indication of an exponential speed up reported there does not seem present at higher spin numbers. Instead an indication of a polynomial speedup over classical mixing is present.

# Contents

Chapter 1

# Introduction

One of the fundamental objects in classical (computational) statistical physics is the Gibbs distribution* $\boldsymbol{\pi}_G(\beta)$. This distribution gives the probability to find a system, with a finite state space, in the state $x$ when the system is in thermal equilibrium at a temperature $T$ and is defined for a system with a Hamiltonian $H(x)$ as

$$\boldsymbol{\pi}_G(\beta)_x = \frac{\exp(-\beta H(x))}{Z(\beta)} \tag{1.1}$$

where $Z(\beta)$ is the partition function and, $\beta = 1/T$†. The Hamiltonian $H(x)$ is a function that gives an energy for each state $x$. Sampling from this distribution allows one to study most of the physics of a system in thermal equilibrium [2]. Sampling from the Gibbs distribution also has applications outside of computational statistical physics. In general, applications of Gibbs sampling are e.g. in statistical inference [3] and training machine learning models like (restricted) Boltzmann machines used throughout generative modelling [4].

Sampling from the Gibbs distribution is usually performed using a stochastic matrix $P$ such that $P\boldsymbol{\pi}_G(\beta) = \boldsymbol{\pi}_G(\beta)$. Here we want to look at the problem of sampling from $\boldsymbol{\pi}_G(\beta)$ given (sparse) access to the matrix $P$. By the Perron-Frobenius theorem [5] we are guaranteed that every irreducible stochastic matrix $P$ has an unique stationary distribution. Therefore we could find $\boldsymbol{\pi}_G(\beta)$ by finding the kernel of $\mathbb{1} - P$ or the $+1$ eigenvector of

*This distribution is also referred to as the thermal distribution or Boltzmann distribution; in this work we will use the term Gibbs distribution.

†We are working in units where $k_B = 1$.

$P$, however, classically finding eigenvectors has time complexity polynomial in the matrix size [6]. As the state spaces of the systems we are interested in typically have exponentially many states, this way of preparing the Gibbs distribution is not classically efficient. Classically, these sampling problems are commonly solved using Markov chain mixing. When $P$ is irreducible and aperiodic then for every probability vector $\boldsymbol{x}$ we are guaranteed $\lim_{l\to\infty} P^l \boldsymbol{x} = \boldsymbol{\pi}_G(\beta)$ [7]. Given this we can start in any initial state and if we update that state often enough using the probability rules given by $P$ we obtain a sample from the distribution $\boldsymbol{\pi}_G(\beta)$. How often we need to perform this updating to be within total variation distance $\epsilon$ of the stationary distribution is called the *mixing time* and is lower bounded by the inverse spectral gap of $P$. Sampling from this distribution is in general a computationally hard problem: for certain Hamiltonians it is related to problems which are known to be NP-Hard [8, 9]. For a general Hamiltonian the mixing time can be exponentially large, however, there are instances [10] where we know the mixing time increases logarithmically with state space dimension.

The connection between Gibbs sampling and NP-Hard problems implies that we do not expect quantum computers to be able to efficiently perform this sampling for a general Hamiltonian [11]. There has however been interest in finding quantum algorithms to perform this sampling [12–15]. Algorithms have been proposed that offer a polynomial speedup in the spectral gap [13, 15] of the stochastic matrix giving potential for a polynomial speedup over classical mixing. Preparing an encoding of the Gibbs distribution is also a step used in the initialisation of other algorithms [14, 16, 17]. These algorithms use complex constructions using e.g. quantum walk operators [14, 15] or complex annealing schedules [13]. The algorithm we propose is conceptually more straightforward.

In this work we look at systems that consist of a collection of classical spins. Each spin has two possible states (up or down), therefore the composite system has a number of possible states that increases exponentially with the number of spins. As we are interested in these exponentially large state spaces, directly finding the $+1$ eigenstate of $P$ is not classically efficient. There are quantum algorithms that solve some linear algebra problems logarithmically in the matrix dimension (e.g. the algorithm proposed by Harrow, Hassidim and, Lloyd (2009) [18] (HHL) for linear systems solving). This might be an indication that instances exist where quantum computers are able to efficiently directly prepare an encoding of the $+1$ eigenvector of $P$. Note that preparing such a state is not equivalent to

classically finding the vector. Measuring the state allows one to sample from the encoded distribution, but to construct the entire vector typically requires exponentially (in the number of spins) many samples. An advantage of being able to directly do this through linear algebra routines is that it allows to use techniques from numerical linear algebra to improve the time complexity. One such technique we look at is preconditioning. Preconditioning has been shown to enable exponential speedup for quantum linear systems solving over classical algorithms for some problems [19].

In this work we present a novel algorithm to directly prepare an encoding of the Gibbs distribution as a quantum state through an eigenstate preparation method which allows us to sample from the stationary distribution by measuring the prepared state. In particular, our objective is to prepare the state $|\boldsymbol{\pi_G}\rangle$ such that $(\mathbb{1} - P)\,|\boldsymbol{\pi_G}\rangle = 0$. For this to work the stationary distribution $\boldsymbol{\pi_G}$ has to be encoded as

$$|\boldsymbol{\pi_G}\rangle = \sum_i \frac{(\pi_G)_i}{||\boldsymbol{\pi_G}||_2} |i\rangle \ , \tag{1.2}$$

where the states $|i\rangle$ are the computational basis states. Many other quantum algorithms used for Gibbs sampling aim to produce a different encoding of $\boldsymbol{\pi_G}$ which is given by (again in the computational basis)

$$|\sqrt{\boldsymbol{\pi_G}}\rangle = \sum_i \sqrt{(\pi_G)_i}\,|i\rangle \ . \tag{1.3}$$

Measuring this encoded state allows to sample directly from $\boldsymbol{\pi_G}$. One key observation we make is that using the encoding of a vector $\boldsymbol{x}$ as a quantum state $|\boldsymbol{x}\rangle$ as in equation 1.2 has the consequence that measuring this state is equivalent to sampling from a different distribution which we call the *squared distribution* and is given by

$$(x^2)_i = \frac{x_i^2}{||x||_2^2} = \frac{x_i^2}{\sum_j x_j^2} \ . \tag{1.4}$$

For the Gibbs distribution with inverse temperature $\beta$ the squared distribution is equivalent to the Gibbs distribution at $2\beta$[‡]. At $2\beta$ the system is closer to the ground state of the system than at $\beta$. Since sampling from the

[‡]See section 3.1 for details.

ground state is (often exponentially) harder compared to sampling from the high temperature distribution, for many systems classical mixing has a time complexity transition at some critical inverse temperature $\beta_C$. The main question we will be looking into in this thesis is whether this temperature doubling allows for an exponential speedup in some circumstances. Preparing encodings of the Gibbs distribution like we propose here has applications outside of sampling problems. One of the necessary steps of many quantum machine learning algorithms is to prepare specific input states [20]. When these distributions are close to uniform this can be done efficiently. However, when some elements dominate this can be exponentially hard. Gibbs distributions are often sharply peaked below some critical temperature so efficient Gibbs sampling might have applications in preparing sharply peaked input states for other algorithms. Encoding a distribution as in equation 1.2 additionally has applications outside of preparations of Gibbs distributions. The squared distribution amplifies the biggest elements in the original distribution, which is comparable to methods used in machine learning.

There are many algorithms to prepare the desired state. The new algorithm we propose here prepares this encoded state using adiabatic state preparation [21]. Adiabatic state preparation is a quantum computing paradigm that uses time dependent evolution to prepare eigenstates of Hamiltonians that encode the problem one wants to solve. This paradigm was shown to be polynomially equivalent to standard gate-based quantum computing [22] and has been studied for a wide array of computational problems [23, 24]. We give analytical observations and numerically study this algorithm for one of the simplest model with such a phase transition, the uniform zero-field Curie-Weiss model. We compare our algorithm to the classical mixing time and to two other state preparation algorithms proposed for preparation of a similar state in Orsucci (2019) [1]. We add novel analytical analysis to two of the algorithms proposed therein and we extend the numerical analysis presented there to higher spin numbers by using a projection to the symmetric subspace (see section 2.5) of the transition matrix $P$.

The novel results in this work include the definition of a novel algorithm for Gibbs sampling using adiabatic state preparation. We optimise the dependence on the minimal adiabatic gap to a linear dependence for the uniform zero-field Curie-Weiss model under some assumptions. We further give numerical analysis of the scaling of the minimal adiabatic gap for this model. We also present novel analytical observations for the algorithms

presented in Orsucci (2019) [1]: we give an expression for the overlap between two encoded Gibbs distributions for the projection-based algorithm defined in Orsucci (2019) [1] and we give novel bounds on the condition number for the algorithm based on the so-called relaxation method. We also extend the numerical analysis done in [1] to higher spins number. For both the numerical analysis of our adiabatic algorithm and for the two algorithms defined by Orsucci (2019) [1] we use a novel closed form expression for the projection of the stochastic matrix of the uniform zero-field Curie-Weiss model onto the symmetric subspace.

This thesis is structured as follows: We start in chapter 2 by giving some background on (adiabatic) quantum computing, (quantum) complexity theory and how it relates to the problem we are trying to find an algorithm for, Markov chain mixing processes, the Curie-Weiss model and the symmetric subspace projection. In the same chapter we also define the used notation and give numerical results for the classical mixing time for our problem to be able to directly compare the time complexity of our quantum algorithms. In chapter 3 we define our algorithm and give a brief description of the other algorithms from Orsucci (2019) [1]. In the same chapter we also give numerical results for the time complexities of the three algorithms and offer analytical observations on all three algorithms. Finally in chapter 4 we offer an interpretation of the results and sketch further research directions.

Chapter 2

# Relevant Background & Notation

## 2.1 General Notation

For a $n \times n$ matrix $A$ there are $n$ eigenvalues $\lambda$ such that $Av = \lambda v$ for an eigenvector $v$. We write the $i$-th eigenvalue as $\lambda_i(A)$ where they are ordered descending and counted with multiplicity, i.e. $|\lambda_1(A)| \geq |\lambda_2(A)| \geq \ldots \geq |\lambda_n(A)|$.

For a $n \times n$ matrix $A$ there are $n$ singular values $\sigma$ such that $Av = \sigma u$ and $A^\dagger u = \sigma v$ for a left singular vector $u$ and a right singular vector $v$. The singular values of $A$ are denoted with $\sigma_i(A)$ where $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_n(A) \geq 0$.

When an matrix is Hermitian (i.e. when $A^\dagger = A$) the singular values of $A$ are the absolute values of the eigenvalues of $A$. In general the singular values of a matrix $A$ are given by the square root of the eigenvalues of $A^\dagger A$.

We define the singular value gap as $\zeta(A) = \sigma_{n-1}(A) - \sigma_n(A)$ and the spectral gap as $\delta(A) = |\lambda_1(A)| - |\lambda_2(A)|$. Note that when $P$ is an irreducible stochastic matrix (see section 2.3.1 for definitions) we are guaranteed $\lambda_1(P) = 1$ and so $\delta(P) = 1 - |\lambda_2(P)|$ and therefore $\delta(P) = \lambda_{n-1}(\mathbb{1} - P)$.

We denote vectors in bold by $\boldsymbol{x}$ and their individual elements by $x_i$. With $||\boldsymbol{x}||_p$ we denote the $L_p$ norm defined by

$$||\boldsymbol{x}||_p = \left( \sum_i |x_i|^p \right)^{1/p} , \tag{2.1}$$

where in this thesis we will be using the $L_1$ and $L_2$ norm. General quantum states are denoted by $|\psi\rangle$ and $|\boldsymbol{x}\rangle$ is used for the encoding of a vector $\boldsymbol{x}$ in a quantum state as

$$|\boldsymbol{x}\rangle = \sum_i \frac{x_i}{||\boldsymbol{x}||_2} |i\rangle \tag{2.2}$$

where $|i\rangle$ are the computational basis states. All quantum states in this work are in the computational basis.

We will be considering classical spin systems with $N$ spins with a state space of dimension $n = 2^N$.

## 2.2 Quantum Computing & Algorithmics

### 2.2.1 Adiabatic Quantum Computing

In quantum computing (QC) we use aspects of quantum mechanics to solve computational problems. A QC process is described using (normalized with respect to the $L_2$ norm) complex vectors in a Hilbert space $\mathcal{H}$ and unitary operators $\mathcal{H} \to \mathcal{H}$. In general these Hilbert spaces can be infinite dimensional however in this work we only look at finite dimensional spaces. We call these vectors states and denote them by $|\psi\rangle \in \mathcal{H}$. Since we are looking at finite dimensional states we can choose a basis $\{|\lambda_i\rangle\}$ and write every state in that space as

$$|\psi\rangle = \sum_i a_i |\lambda_i\rangle , \tag{2.3}$$

with $a_i$ a complex coefficient for each of the basis states. We call such a state a superposition of basis states. When we measure we do so with respect to a chosen basis and when we perform the measurement the system is projected to one of those basis vectors. When we have a system in a state as in equation 2.3 the probability of it being in $|\lambda_i\rangle$ after measurement is given by $|a_i|^2$. This relation is called the *Born rule*. All states are therefore

normalised with respect to the $L_2$ norm, i.e. such that $\sum_i |a_i|^2 = 1$. We consider quantum systems consisting of a collection of qubits. A single qubit is a two state system that has two basis states which we call up (denoted as $|0\rangle$) and down (denoted as $|1\rangle$). An $N$ qubit system is the $N$ fold tensor product of such two state systems. The exponentially increasing collection of basis states we write as $N$ bit strings, i.e. they are written as e.g. $|00101\rangle$. We call this basis the computational basis.

In the most common form of QC the operators used to perform the computations take the form of unitary gates and there the QC process is described by a circuit composed of those gates acting on some initial state. In this work we look at a different form of QC called adiabatic quantum computing (AQC) [25]. AQC has been shown to be polynomially equivalent to the standard gate based paradigm [22]. The time dynamics of a quantum system are governed by the Schrödinger equation* [26]

$$H(t)\,|\psi(t)\rangle = i\frac{d}{dt}\,|\psi(t)\rangle\ , \tag{2.4}$$

with the operator $H(t)$, which is the Hamiltonian, the operator that generates infinitesimal time translations for the system. The quantum adiabatic theorem [26] states that if we initialize a system close to an eigenstate of $H(0)$ and we apply the time evolution slow enough we remain close to an eigenstate of $H(t)$ at a time $t$. We can use this to solve computational problems by encoding the solution to a problem as the eigenstate of a problem Hamiltonian $H_p$. We prepare this eigenstate by starting from some initial Hamiltonian $H_I$ for which we can easily prepare the corresponding eigenstate and which does not commute with $H_P$. We then adiabatically evolve from $H_I$ to $H_P$. To do so we define an interpolating Hamiltonian

$$H(t) = f(t)H_I + g(t)H_P \tag{2.5}$$

with $f, g$ monotonic functions that satisfy $f(0) = g(T) = 1$ and $f(T) = g(0) = 0$. The time $T$ is chosen such that the evolution is slow enough to remain adiabatic. The time complexity of preparing a state via this algorithm is determined by how big $T$ has to be for the evolution to remain adiabatic. Here we look at the often used approximation of the adiabatic condition, in appendix A.1 we give a more rigorous version of the adiabatic condition. If $|E_i(t)\rangle$ denotes the eigenstate of $H(t)$ we want to re-

*In units where $\hbar = 1$.

main close to and $|\psi(t)\rangle$ the state of the system, both at time $t$, then we are guaranteed that at the final time $T$

$$|\langle E_i(T)|\psi(T)\rangle|^2 \geq 1 - \epsilon^2 \tag{2.6}$$

when

$$\frac{\max|\langle \frac{\mathrm{d}H(t)}{\mathrm{d}t}\rangle|}{\Delta_{\min}^2} \leq \epsilon \tag{2.7}$$

where $\Delta_{\min}$ is the smallest eigenvalue gap of the time dependent Hamiltonian $H(t)$ along the path between the state we want to remain in and the next excited state. $\max|\langle \frac{\mathrm{d}H(t)}{\mathrm{d}t}\rangle|$ denotes the maximal matrix element between the eigenstate which we want to remain close to and the next excited state.

### 2.2.2 Complexity Theory

There exists a large zoo of complexity classes that are used to describe and categorize problems both in classical and quantum computing. See e.g. [27] for an overview of classical complexity classes and [28] for an overview of quantum complexity classes. Here we give a background of the complexity of the problem we are trying to find an algorithm for in this work.

Two of the main classical complexity classes are P and NP. Those classes are defined in terms of decision problems which are problems that allow only two possible answers, either YES or NO. A decision problem is in P if it can be solved in polynomial time and in NP if there exists an algorithm to verify a proof in polynomial time by a deterministic Turing machine. The ability to solve the problem in polynomial time implies that we can also verify an input in polynomial time, and as such we find that P is a subset of NP. A class of problems that are conjectured[†] to be strictly harder than those in P are those which are NP-Hard. NP-Hard problems are those to which every problem in NP can be reduced to in polynomial time, i.e. a solution of every problem $A \in$ NP can be mapped to a solution $B$ of a specific NP-Hard problem in polynomial time.

[†]This conjecture holds when P $\neq$ NP.

One of the problems in NP-Hard is graph partitioning. This problem is stated as: *"given a graph G with an even number of vertices and an integer k, is there a partition in two subsets of equal size such that there are at most k edges between the subsets?"* This problem can be connected to statistical physics. It can be shown that graph partitioning is equivalent to minimizing the energy of a formulation of an Ising model [9]. When looking at low enough temperatures Gibbs sampling allows one to find the lowest energy spin configurations. From those configurations the ground state energy can easily be determined therefore classical Gibbs sampling has to be hard in general. It is widely believed that quantum computers will not be able to solve worst-case instances of NP-Hard problems efficiently. Therefore we also do not expect a quantum computer to be able to efficiently perform Gibbs sampling in the general case. This does however not exclude the possibility of the ability of quantum computers to solve specific instances of NP-Hard problems for which classical algorithms fail. There is also the possibility of quantum algorithms offering a polynomial speedup over the fastest classical algorithm for an NP-Hard problem.

When analysing algorithms we are often mainly interested in how they scale for increasing problem sizes. In order to describe this we define the big-O notation. This notation denotes that a function $f(x)$ scales at most as fast as some other function $g(x)$. We write $f(x) = \mathcal{O}(g(x))$ if there exists a positive constant such that $f(x) < Cg(x)$ when $x$ is bigger than some constant $x_0$. A similar notation exists for the asymptotic lower bound which is denoted by big-$\Omega$. When an asymptotic scaling involves terms with $\ln x$ those contribute very little to the overall scaling so those can often be ignored. For algorithms that scale in such a way we use the soft O-notation which we write as $\tilde{O}$ to suppress the logarithmic factors.

### 2.2.3 (Quantum) Linear System Solving

One problem where QC offers potential for an exponential speedup over the fastest classical algorithm is solving linear systems of equations. That is, given a non-singular $n \times n$ matrix $A$ and vector $\boldsymbol{b}$ find the vector $\boldsymbol{x}$ such that $A\boldsymbol{x} = \boldsymbol{b}$. The fastest classical algorithm to find $\boldsymbol{x}$ has a time complexity that scales polynomially in $n$ [29]. The related quantum version of this problem asks to prepare a state proportional to $A^{-1}|b\rangle = |x\rangle$. The main example of an algorithm that solves this problem was proposed by Harrow, Hassidim, and, Lloyd (2009) [18] (HHL). The HHL algorithm has a time complexity that scales polynomially with $\kappa \log n$, where $\kappa(A)$ is the

condition number of the matrix $A$. The condition number of a matrix $A$ is defined as

$$\kappa(A) = ||A||\ ||A^{-1}|| \tag{2.8}$$

where $||A||$ is any consistent norm[‡]. For numerical analysis we here choose the $||\cdot||_2$ norm such that

$$\kappa(A) = ||A||_2\ ||A^{-1}||_2 = \frac{\sigma_1(A)}{\sigma_n(A)} \tag{2.9}$$

So when $A$ is a matrix with constant condition number this algorithm offers an exponential speed up over the fastest classical algorithm. The complexity of the HHL algorithm was later optimised to quasilinear in $\kappa(A)$ and $\ln n$ (e.g. in [30]) giving a time complexity of $\tilde{O}(\kappa(A))$. Note that the quantum version of the problem solves a different problem than the classical algorithm. When one prepares the state $|x\rangle$ one can use measurements to either sample from the encoded vector or determine some other related quantities. To reconstruct the complete distribution requires roughly $n$ samples losing the potentially exponential speedup.

Since solving systems of linear equations is ubiquitous in many scientific fields there has been a significant amount of work done to improve the time complexity of solving these problems on a quantum computer [31–33]. One of those methods we look at here is preconditioning. The goal of preconditioning is to multiply the matrix $A$ by some other matrix that lowers the condition number of the product. Here we look at multiplicative left-preconditioning where instead of solving the system $Ax = b$ we want to solve $PAx = Pb$ where $P$ is called the preconditioner. This preconditioner is chosen to be such that $\kappa(PA) \ll \kappa(A)$ such that the algorithm can more efficiently find the solution for that problem. The choice that would improve the condition number the most is to choose $P = A^{-1}$, however, finding the inverse of the matrix is equivalent to solving the linear system itself so that option is inefficient. There are multiple algorithms to find preconditioners for the classical problem and recently there has been interest in optimising the quantum version of the problem using preconditioning [19, 34]. One algorithm to find a preconditioning matrix is the SParse Approximate Inverse (SPAI) algorithm [35]. The SPAI algorithm is used to find the best approximation of the inverse of $A$ satisfying an *a priori* chosen sparsity pattern.

[‡]A matrix norm is called (self) consistent when it satisfies $||AB|| \leq ||A||\ ||B||$.

## 2.3 Markov chains and Gibbs Samplers

### 2.3.1 Markov Chains and Classical Mixing

Here a Markov chain (MC) process is a discrete-time stochastic process over a finite state space where the probability to transition from state to state is only dependent on the current state of the system. We describe such a process by a stochastic matrix $P$ where here we use left stochastic matrices[§] which are matrices with all elements entry-wise non-negative and all columns summing to unity. The element $P_{ji}$ of such a matrix gives the probability of going from state $i$ to state $j$. When we start the system in some state then each time step the system probabilistically goes to some other state. Therefore after some time has passed there is a distribution over all possible states the system can be in at that point. We write that distribution as a probability column vector $\boldsymbol{x}$, which is a non-negative real vector where all elements sum to unity. When at some point the system has a state distribution given by $\boldsymbol{x}$ the distribution at the next step is given by $\boldsymbol{x}' = P\boldsymbol{x}$. The element $x_i$ denotes the probability to find the system in state $i$. A special state distribution is the stationary distribution $\boldsymbol{\pi}$ which is the state distribution $\boldsymbol{\pi}$ such that $P\boldsymbol{\pi} = \boldsymbol{\pi}$. Since for every column stochastic matrix the transpose $P^T$ has the uniform vector as its $+1$ eigenvector. This guarantees that every stochastic matrix $P$ an $+1$ eigenvector.

For our purposes we need some extra conditions for the MC. We define that a MC is irreducible if we can reach every state from every other state in a finite number of steps with a non-zero probability. For each irreducible MC we are guaranteed by the Perron-Frobenius theorem [5] that there exists an unique stationary distribution [7]. A state of a MC is periodic if the process can only return to that state in a number of steps which is a multiple of some integer larger than one. When all states are aperiodic we call the MC aperiodic. When an MC is irreducible the existence of a single aperiodic state implies that all states are aperiodic. When a MC is both irreducible and aperiodic we are guaranteed that from every initial state $\boldsymbol{x}$ we converge to the unique stationary distribution. This process is called mixing, i.e.. this guarantees that

$$\lim_{l\to\infty} P^l \boldsymbol{x} = \boldsymbol{\pi} \tag{2.10}$$

[§]We use left stochastic matrices here since that mirrors how operators are applied to states in quantum computing. In mathematics right stochastic matrices, where each row sums to unity and elements $P_{ij}$ denote the transition probability from state $i$ to state $j$, are more common.

for any probability vector $\boldsymbol{x}$ and $\boldsymbol{\pi}$ the stationary distribution of $P$. In physics literature the term Ergodic MC is widely used, a process is called ergodic if all statistical properties can be inferred from a sufficiently long sample of the process. A MC process is ergodic if it is irreducible and has at least one aperiodic state.

An MC is called time-reversible when for its stationary distribution the detailed balance condition holds, i.e.

$$\boldsymbol{\pi}_j P_{ij} = \boldsymbol{\pi}_i P_{ji} \,. \tag{2.11}$$

This guarantees that, when the chain has reached the stationary distribution, the transition $i \to j$ is as probable as the transition $j \to i$. This guarantees the chain is invariant under time reversal. The number of steps needed to converge to within distance $\epsilon$ of the stationary distribution in a mixing process is called the mixing time $t_{\text{mix}}(\epsilon)$. This distance is defined with respect to the total variation distance defined as $d(\boldsymbol{x}, \boldsymbol{x}') = 1/2 \sum_i |x_i - x_i'|$, we further define $d(k) = \max_{\boldsymbol{x}} d(P^k \boldsymbol{x}, \boldsymbol{\pi})$ as the maximum distance between any initial state $\boldsymbol{x}$ and the stationary distribution $\boldsymbol{\pi}$ after $k$ steps. Then $t_{\text{mix}}(\epsilon)$ is the smallest time such that $d(t_{\text{mix}}(\epsilon)) \leq \epsilon$. For aperiodic, reversible and irreducible MCs this mixing time is bounded by [7]

$$\left(\frac{1}{\delta(P)} - 1\right) \log\left(\frac{1}{2\epsilon}\right) \leq t_{\text{mix}}(\epsilon) \leq \frac{1}{\delta(P)} \log\left(\frac{1}{\epsilon \boldsymbol{\pi}_{\min}}\right) \tag{2.12}$$

with $\delta(P)$ the spectral gap, the gap between the highest and second highest eigenvalue, of the transition matrix $P$ and $\boldsymbol{\pi}_{\min}$ the smallest element of the stationary distribution. When $\delta(P) \ll 1$ this gives a tight bound on the mixing time and $t_{\text{mix}} = \tilde{O}(1/\delta(P))$.

### 2.3.2 Gibbs Samplers

One family of stochastic matrices which we are interested in here are those where the stationary distribution has the form given by the Gibbs distribution

$$\boldsymbol{\pi}_G(\beta)_x = \frac{\exp(-\beta H(x))}{Z(\beta)} \tag{2.13}$$

where $\beta \in [0, +\infty)$ is a parameter often called the inverse temperature, $H(x)$ is a function that gives a scalar value for every state $x$ and $Z(\beta)$ is

the normalisation factor called the partition function. The function $H(x)$ is called the Hamiltonian and the values it gives are called the energy of a state $x$. We call this family of stochastic matrices for any $H(x)$ *Gibbs samplers*[¶] which we will denote by $P_G(\beta)$.

Here we look at systems that consist of classical spins which can either be $+1$ or $-1$. A state of the system is described by a specific configurations of those spins which we denote by $\sigma$ with $\sigma_i$ denoting a specific configuration $i$ (not to be confused with the $i$-th element of a vector). Here the element $P_G(\beta)_{ji}$ gives the probability to go from spin configuration $\sigma_i$ to spin configuration $\sigma_j$. These transition matrices are therefore given in the basis consisting of all possible configurations $\{\sigma\}$. We only look at Markov chains where we change at most one of those spins in a state transition. We consider two stochastic matrices that satisfy those constraints. For Glauber dynamics the transition matrix is given by [7]

$$P_G(\beta)_{ji} = \frac{1}{N}\frac{\exp(-\beta H(\sigma_j))}{\exp(-\beta H(\sigma_i)) + \exp(-\beta H(\sigma_j))} \tag{2.14}$$

when $i, j$ differ by only one spin, $P_{ii} = 1 - \sum_j P_{ji}$ and $P_{ji} = 0$ otherwise. Here $N$ is the total number of spins in the system. The other transition matrix we consider is from the Metropolis-Hastings algorithm which is given by [6]

$$P_G(\beta)_{ji} = \frac{1}{N}\min(1, \exp(-\beta(H(\sigma_j) - H(\sigma_i)))) \tag{2.15}$$

when $i, j$ differ by only one spin, $P_{ii} = 1 - \sum_j P_{ji}$ and $P_{ji} = 0$ otherwise. Both of these stochastic matrices have the Gibbs distribution as their unique stationary distribution and are aperiodic, irreducible and time-reversible.

## 2.4 Curie-Weiss Model

### 2.4.1 Model background

We want to consider a model where there are inverse temperatures $\beta$ such that the mixing time at $2\beta$ is exponentially larger. One of the simplest ex-

[¶]Note that Gibbs sampler is also used for a specific Monte Carlo sampling algorithm, here we use it to denote the family of stochastic matrices with the Gibbs distribution as the stationary distribution.

amples of such a model is the Curie-Weiss model. Here we briefly define the terms we need, for a more detailed overview see e.g. [36]. The Curie-Weiss model is the Ising model on the complete graph with the Hamiltonian for a spin configuration $\sigma$ given by

$$H(\sigma) = -\frac{J}{N} \sum_{1 \leq i < j \leq N} \sigma_i \sigma_j - \mu B \sum_{i=1}^{N} \sigma_i \tag{2.16}$$

with $J$ the coupling strength between the spins, $\mu$ the magnetic moment of each spin and $B$ the magnitude of the external field in the direction of the spins. We look at the simplest case and set $J = 1$ and $B = 0$ which we call the uniform zero-field Curie-Weiss model. Since the model is defined on the complete graph, the model is invariant under exchanges of the spins so we can write the state of the model as a bit string where a 0 denotes spin up and 1 spin down. Since we can freely exchange spins the Hamming weight of a state, which is defined as the number of 1s in a bit string, fully describes the macroscopic state of the system.

The absolute magnetisation $|M|$ of a spin configuration is defined as the absolute value of the sum over all spins. This model has a phase transition at the critical inverse temperature $\beta_C = 1$ (see figure 2.1). Below $\beta_C$ the Gibbs distribution is close to uniform, giving an average absolute magnetisation close to zero. Above $\beta_C$ the Gibbs distribution goes towards a bimodal distribution with the two lowest energy spin configurations as the modes of the Gibbs distribution. Those configurations are all spins up or all spin down giving an average magnetisation above $\beta_C$ that goes to one. We call those states the ground states of the system.

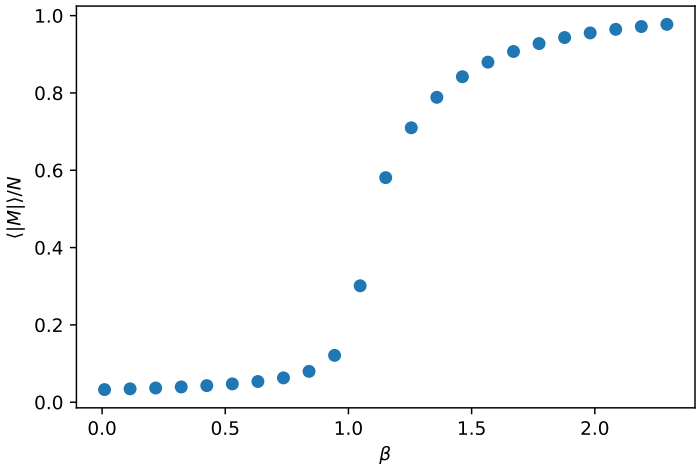


***Figure 2.1:*** *Average absolute value of the magnetisation per spin of the Gibbs distribution for the uniform zero-field Curie-Weiss model at a range of inverse temperatures $\beta$ for 600 spins.*

This model also has a phase transition in the classical mixing time at the same critical inverse temperature $\beta_C$. In [10] the following theorem is proven.

**Theorem 2.1** *Let $\alpha(N) > 0$, then:*

- *For $\beta = 1 - \alpha$ the spectral gap is given by $(1 + O(1))\alpha/N$ where the $O(1)$ term tends to 0 as $N \to \infty$.*
- *When $\alpha = O(1/\sqrt{N})$ the spectral gap is $O\left(N^{-3/2}\right)$.*
- *For $\beta = 1 + \alpha$ the spectral gap is*

$$O\left(\frac{\alpha}{N}\exp\left[-\frac{N}{2}\int_0^{\xi}\log\left(\frac{1+g(x)}{1-g(x)}\right)\,\mathrm{d}x\right]\right) \tag{2.17}$$

*with $g(x) = (\tanh(\beta x) - x)/(1 - x\tanh(\beta x))$ and $\xi$ the unique positive root of $g(x)$.*

From this theorem we see that the inverse spectral gap below $\beta_C$ increases linearly with $N$ and above $\beta_C$ increases exponentially resulting in a sharp transition in the mixing time for this model.

### 2.4.2 Numerical Mixing Time Bounds

To be able directly compare the time complexity of the quantum algorithm (which we will define later in chapter 3) to the classical time complexity, we look numerically at the mixing time bounds for the stochastic matrices we defined in section 2.3.2 for the uniform zero-field Curie-Weiss model. From equation 2.12 we know the mixing time is bounded by $1/\delta$ (suppressing logarithmic factors). We first look at the stochastic matrix corresponding with Glauber dynamics for the uniform zero-field Curie-Weiss model at three inverse temperatures: $\beta = 0.7$, $\beta = 1.0$, and $\beta = 1.4$. These inverse temperature were chosen to be below, at and above the phase transition point. In figure 2.2 we see these results. These results support the analytical results given in theorem 2.1. Note that for two of the chosen inverse temperatures sampling at $\beta$ is linearly easy and sampling at $2\beta$ exponentially hard, making this an interesting case to look at with regards to the time complexity of our quantum algorithm.

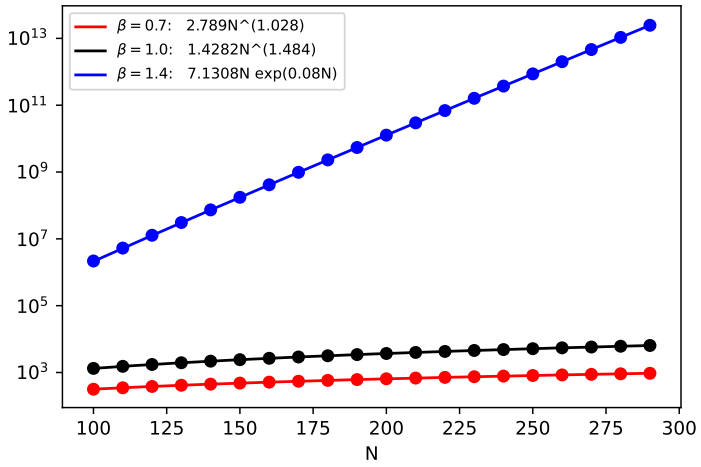


***Figure 2.2:*** *Inverse spectral gap of the stochastic matrix corresponding with Glauber Dynamics for the uniform zero-field Curie-Weiss model at three different inverse temperatures (see legend) for a range of $N$. Points indicate simulated values and lines indicate the best fitted functions as predicted by [10] (see legend). Note that below the critical inverse temperature $\beta = 1$ the inverse spectral gap scales linearly and exponentially above.*

We also want to look at the same results for the stochastic matrix corresponding with the Metropolis-Hastings algorithm (see figure 2.3). We observe the same scaling behaviour.

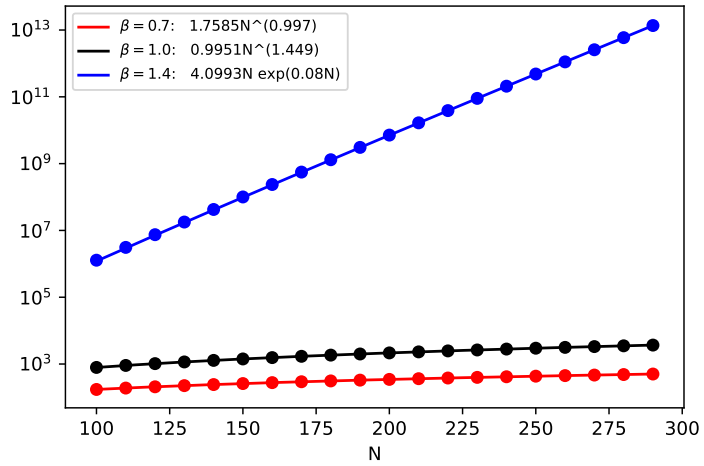


***Figure 2.3:*** *Inverse spectral gap of the stochastic matrix corresponding with the Metropolis-Hastings algorithm for the uniform zero-field Curie-Weiss model at three different inverse temperatures (see legend) for a range of $N$. Points indicate simulated values, and lines indicate the best fitted functions as predicted by [10] (see legend). Note that below the critical inverse temperature $\beta = 1$ the inverse spectral gap scales linearly and exponentially above.*

## 2.5 Symmetric Subspace Projection

Here we give a brief description of how we used symmetric subspace projection to reduce the dimensionality of our numerical problem. See [37] for an overview of the relevant group- and representation theory and [38] for a more in depth discussion on symmetric subspace projection in quantum information. For a digression on the representation theory of the symmetric group see [39].

To study the time complexities of our classical and quantum algorithms we need to find the eigenvalues and singular values of the matrices $P_G(\beta)$. For a system with $N$ spins those matrices have dimension $2^N \times 2^N$ which makes directly finding their eigenvalues and singular values numerically tractable in a reasonable time frame for up to $N \sim 20$. The matrices we look at are in the basis given by all $2^N$ possible configurations of $N$ classical spins. Since our model is defined on the complete graph the macroscopic physics of the model is the same if we exchange any two spins, i.e. our model is invariant under all permutations of spins. Due to this invariance with respect to the symmetric group, all spin configurations with

the same Hamming weight represent the same macroscopic state of the system. When we represent a spin configuration as a bit string with 0 denoting spin up and 1 denoting spin down the configurations $|110\rangle$, $|101\rangle$, and $|011\rangle$ represent the same macroscopic state. As such our exponentially large matrices contain information on the configuration of specific micro states which is irrelevant for the macro state of the system. We can use this to reduce the dimension of the matrix we need to look at by projecting it onto the symmetric subspace which is the space spanned by the uniform combination of states with the same number of spins up or down. For example for $N = 3$, this is the subspace spanned by the spin configurations $|000\rangle, 1/\sqrt{3}(|001\rangle + |010\rangle + |100\rangle), 1/\sqrt{3}(|110\rangle + |101\rangle + |011\rangle)$ and $|111\rangle$. This reduces the matrix from dimension $2^N \times 2^N$ to $(N+1) \times (N+1)$. For our numerical analysis we need to find the second smallest eigenvalues and singular values of $\mathbb{1} - P_G(\beta)$ for the uniform zero-field Curie-Weiss model. Since we conjecture that the physics of the system is described by the part of the matrix that is projected onto the fully symmetric subspace we expect that those relevant eigenvalues and singular values can be found by looking at the projection. To numerically investigate whether the eigenvalues and singular values we are interested in are given by the symmetric subspace projection we look at the relative difference between the full transition matrix and the projection onto the subspace. In figure 2.4 we see relative difference between the second smallest eigenvalues and second smallest singular values for the transition matrix for Glauber dynamics for the uniform zero-field Curie-Weiss model. These are for $N$ where we can directly compare the results. The difference is given in units of a fraction of the full matrix values. The difference is in the order of magnitude of the precision the numerical eigenvalue algorithms we used (ARPACK & LAPACK) work to.

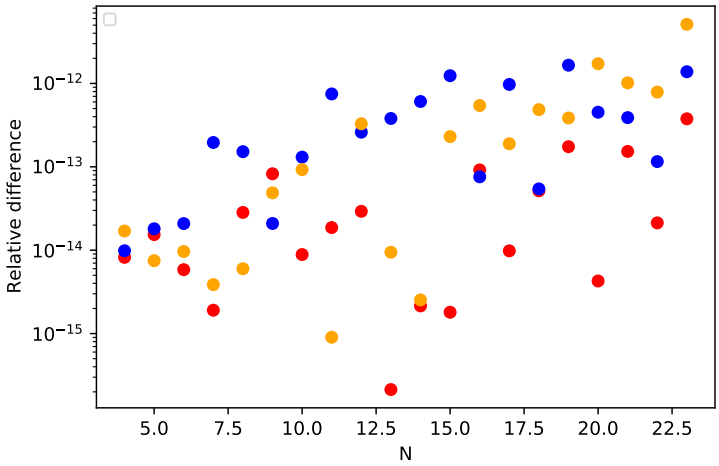


***Figure 2.4:*** *Relative difference between the full transition matrix for Glauber dynamics for the uniform zero-field Curie-Weiss model. Red and blue dots denote the second smallest eigenvalue at $\beta = 0.7$ and $\beta = 1.4$ respectively and orange dots the second smallest singular values at $\beta = 0.7$.*

If we can find a closed expression for this symmetric subspace of the transition matrices we are interested in we can numerically analyse the eigenvalues and singular values for far higher spin numbers.

**Theorem 2.2** *Let $H_{lk}(\beta)$ denote the probability for a single spin flip Markov chain at inverse temperature $\beta$ on the complete graph to go from a state with Hamming weight $k$ to a state with Hamming weight $l$. Then the projection $S(\beta)$ of the Gibbs sampler $P_G(\beta)$ for such a N-spin model onto the symmetric subspace is given by*

$$S_{(k+1),k}(\beta) = \frac{N-k}{N}\sqrt{\frac{\binom{N}{k}}{\binom{N}{k+1}}}H_{(k+1),k}(\beta) = \frac{N-k}{N}\sqrt{\frac{k+1}{N-k}}H_{(k+1),k}(\beta) \tag{2.18a}$$

$$S_{(k-1),k}(\beta) = \frac{k}{N}\sqrt{\frac{\binom{N}{k}}{\binom{N}{k-1}}}H_{(k-1),k}(\beta) = \frac{k}{N}\sqrt{\frac{N-k+1}{k}}H_{(k-1),k}(\beta) \tag{2.18b}$$

$$S_{k,k}(\beta) = 1 - S_{(k+1),k}(\beta) - S_{(k-1),k}(\beta) \tag{2.18c}$$

*with the other elements equal to zero.*

This can be proven by directly writing out the transition matrices and the relevant basis transformation.

## 2.6 Numerical Implementation

For the numerical analysis performed for this thesis the eigenvalues and singular values of large tridiagonal matrices needed to be found. This was done with SciPy [40] and NumPy [41] in Python [42]. For the eigenvalues the matrices were transformed to a basis in which they are symmetric and then the SCIPY.LINALG.EIGH_TRIDIAGONAL routine for the eigenvalues of symmetric tridiagonal matrices was used to find the eigenvalues. This routine uses calls to LAPACK for the eigenvalue finding. For the singular values the routine SCIPY.SPARSE.LINALG.SVDS for singular values of sparse matrices was used. This routine uses calls to ARPACK for the singular value finding.

The plots seen in this thesis are made using MatPlotLib [43] in Python [42].

Chapter 3

# Quantum Linear Algebra for Gibbs Sampling

## 3.1 Quantum Linear Algebra

In section 2.3.1 we defined the stationary distribution of a Markov Chain $P$ as the distribution $\boldsymbol{\pi}$ such that $P\boldsymbol{\pi} = \boldsymbol{\pi}$. We can reformulate the preparation of the stationary distribution as finding the kernel of $\mathbb{1} - P$. The stationary distribution is given as a probability vector meaning it is normalised to unity with respect to the $L_1$ norm. Since any scalar multiplied with this vector is also in the kernel the same vector normalised to unity with respect to the $L_2$ norm is also in the kernel of $P$. As such we can find an encoding of the stationary distribution by preparing the state $|\boldsymbol{\pi}\rangle$ such that $(\mathbb{1} - P)|\boldsymbol{\pi}\rangle = 0$. This state is given by

$$|\boldsymbol{\pi}\rangle = \sum_i \frac{\boldsymbol{\pi}_i}{||\boldsymbol{\pi}||_2} |i\rangle \tag{3.1}$$

where $\boldsymbol{\pi}$ is the stationary distribution. If we prepared such a state we can sample from the encoded distribution by measuring the state $|\boldsymbol{\pi}\rangle$. However because of the Born rule measuring this is equivalent to sampling from the distribution

$$(\boldsymbol{\pi}^2)_i = \frac{\pi_i^2}{||\boldsymbol{\pi}||_2^2} \tag{3.2}$$

which we call the *squared distribution*. For the Gibbs distribution we defined in section 2.3.2 measuring the state $|\boldsymbol{\pi}_G(\beta)\rangle$ is equivalent to sampling

from

$$(\boldsymbol{\pi}_G^2(\beta))_i = \frac{(\pi_G(\beta))_i^2}{||\boldsymbol{\pi}_G(\beta)||_2^2} = \frac{\exp(-\beta H(i))^2}{Z(\beta)^2} \frac{1}{\sum_x \frac{\exp(-\beta H(x))^2}{Z(\beta)^2}} \tag{3.3}$$

$$= \frac{\exp(-2\beta H(i))}{\sum_x \exp(-2\beta H(x))} = \frac{\exp(-2\beta H(i))}{Z(2\beta)} \tag{3.4}$$

which is exactly the Gibbs distribution at $2\beta$. So preparing and measuring the stationary state of a Gibbs sampler with parameter $\beta$ in this manner is equivalent to classically sampling from the same sampler at $2\beta$. In section 2.4.2 we observed that the model we look at here has a time complexity phase transition where classically sampling below $\beta = 1$ is exponentially easier than sampling above $\beta = 1$. This might offer potential for a speedup by preparing a quantum state with $\beta$ in the classically easy regime such that it allows us to sample at $2\beta$ in the classically hard regime. This is visualised in figure 3.1.

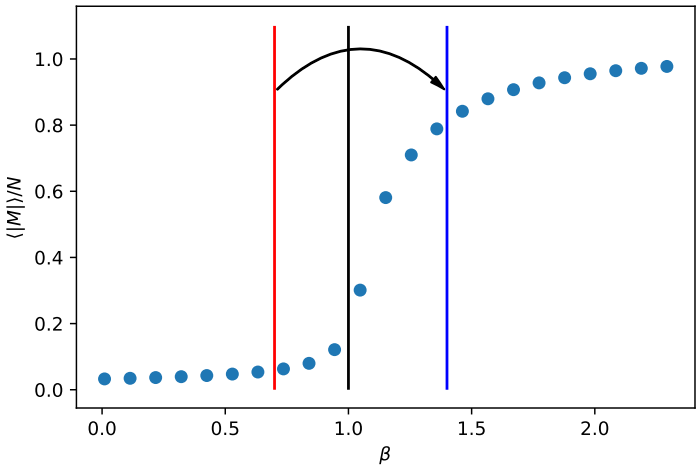


***Figure 3.1:*** *Visualisation of the $\beta$ doubling. The dots indicate the average absolute value of the magnetisation per spin of the Gibbs distribution for the uniform zero-field Curie-Weiss model at a range of inverse temperatures $\beta$ for 600 spins. The black, red and, blue lines indicate $\beta = 1.0$, $\beta = 0.7$ and, $\beta = 1.4$ respectively. The plot visualises that preparing an encoding at some temperature below the phase transition allows to sample from above the phase transition.*

One observation we make, when we promote the vector $\boldsymbol{\pi}_G(\beta)$ to the state

$|\boldsymbol{\pi}_G(\beta)\rangle$ we introduce a normalisation factor $1/||\boldsymbol{\pi}_G(\beta)||_2$. Writing this factor out gives

$$||\boldsymbol{\pi}_G(\beta)||_2 = \left(\sum_i |(\pi_G(\beta)_i|^2\right)^{1/2} \tag{3.5}$$

$$= \left(\sum_i \frac{\exp(-\beta H(i))^2}{Z(\beta)^2}\right)^{1/2} \tag{3.6}$$

$$= \frac{1}{Z(\beta)}\left(\sum_x \exp(-2\beta H(x))\right)^{1/2} \tag{3.7}$$

$$= \frac{\sqrt{Z(2\beta)}}{Z(\beta)}\,. \tag{3.8}$$

and therefore $1/||\boldsymbol{\pi}_G(\beta)||_2 = \frac{Z(\beta)}{\sqrt{Z(2\beta)}}$. So when we look at preparing $|\boldsymbol{\pi}_G(\beta)\rangle$ we are implicitly including a normalisation factor that depends on the partition function at $2\beta$. We know that at the critical inverse temperature the partition function has a transition from magnitude $\sim 2^N$ to magnitude $\sim e^N$. So when $2\beta$ is above the critical inverse temperature even when we would sum over all $2^N$ elements in the basis the pre-factor would be similar to $(2/e)^N$ which is exponentially decreasing. If this conjuncture holds we would expect there to be a complexity transition occurring for the preparation of $|\boldsymbol{\pi}_G(\beta)\rangle$ at exactly half the inverse temperature where the complexity transition occurs for the preparation of $\boldsymbol{\pi}_G(\beta)$. This would imply we would not expect to find a possibility of an exponential speedup by using this encoding for the Gibbs distribution.

## 3.2 Adiabatic State Preparation

### 3.2.1 Set-up

To be able to sample using preparation of $|\boldsymbol{\pi}_G(\beta)\rangle$ we need an algorithm to prepare such a state given sparse access to the stochastic matrix $P_G(\beta)$. Here we propose to do this via adiabatic state preparation. As discussed in section 2.2.1, we need to define a problem Hamiltonian $H_P$, an initial Hamiltonian $H_I$, and an interpolation function. We found that we can prepare $|\boldsymbol{\pi}_G(\beta)\rangle$ by preparing the state such that $(\mathbb{1} - P_G(\beta))\,|\boldsymbol{x}\rangle = 0$. As we need a Hermitian matrix, we use an ancilla qubit to encode our matrix in a Hamiltonian as

$$H(\beta)_P = \begin{pmatrix} 0 & \mathbb{1} - P_G(\beta) \\ \mathbb{1} - P_G(\beta)^T & 0 \end{pmatrix} . \tag{3.9}$$

This Hermitian operator has the stationary distribution of $P$ encoded as $\begin{pmatrix} 0 & \pi_G(\beta) \end{pmatrix}^T$ as one of its two zero eigenvalue vectors*.

In the limit where $\beta \to 0$ the Gibbs distribution goes towards the uniform distribution. The encoding of this uniform distribution (uniform superposition) is easy to prepare, so we can use the same Hamiltonian at low $\beta$ as the initial Hamiltonian. This gives the adiabatic Hamiltonian

$$H(t) = f(t)H_s + g(t)H_p \tag{3.10}$$

$$= \begin{pmatrix} 0 & A(t) \\ A^T(t) & 0 \end{pmatrix} , \tag{3.11}$$

where we define

$$A(t) = f(t)(\mathbb{1} - P_G(0)) + g(t)(\mathbb{1} - P_G(\beta)) \tag{3.12}$$

with $f, g$ the still to be defined interpolating functions. In section 2.2.1 we saw that the time complexity of the adiabatic algorithm is dominated by the eigenvalue gap between the zero eigenvalue state and the next excited state.

To consider the time complexity of the adiabatic algorithm we need to know the eigenvalues of matrices of the form

$$M = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} , \tag{3.13}$$

where we notice that the square of these matrices is given by

$$M^2 = \begin{pmatrix} AA^T & 0 \\ 0 & A^TA \end{pmatrix} , \tag{3.14}$$

*The other zero eigenvalue state is the $+1$ eigenvector of $P^\dagger$, denoted by $U$ encoded as $\begin{pmatrix} U & 0 \end{pmatrix}^T$. This state has zero overlap with the desired state so the transition probability between the two options is zero. Therefore by initialising in the correct state we will end up in the correct state.

The eigenvalues of $M^2$ are given by the eigenvalues of $AA^T$ which implies that the absolute values of the eigenvalues of $M$ are given by the square root of the eigenvalues of $AA^T$. These are exactly the singular values of $A$, so the absolute values of the eigenvalues of $M$ are given by the singular values of $A$. We are interested in the gap between the zero energy state and the next excited state, this eigenvalue gap is exactly given by $\zeta(\mathbb{1} - P_G(\beta))$. We use here the notation we defined in section 2.1.

Therefore the eigenvalue gap we are interested in of $H(t)$ is given by $\zeta(f(t)(\mathbb{1} - P_G(0)) + g(t)(\mathbb{1} - P_G(\beta)))$. When we have two irreducible and aperiodic Markov chains $P, Q$ then $aP + bQ$ is also a irreducible and aperiodic Markov Chain if $a + b = 1$, and $a, b$ are real and non-negative[†]. So when we restrict the interpolating functions such that $f(t) + g(t) = 1\ \forall t$ then $f(t)(\mathbb{1} - P_G(0)) + g(t)(\mathbb{1} - P_G(\beta))$ is a stochastic matrix for all $t$. This implies $\lambda_n((t)(\mathbb{1} - P_G(0)) + g(t)(\mathbb{1} - P_G(\beta))) = 0\ \forall t$. Following Weyl's inequality[‡] [44] this also implies $\sigma_n((t)(\mathbb{1} - P_G(0)) + g(t)(\mathbb{1} - P_G(\beta))) = 0\ \forall t$. So to find the minimum eigenvalue gap along the path for $H(t)$ we need to find the smallest value of $\sigma_{n-1}(f(t)(\mathbb{1} - P_G(0)) + g(t)(\mathbb{1} - P_G(\beta)))$ for $0 \leq t \leq T$.

**Theorem 3.1** *The singular value gap of* $(1 - g(t))(\mathbb{1} - A) + g(t)(\mathbb{1} - B)$*, with* $A, B$ *irreducible and aperiodic symmetric column stochastic matrices such that* $\xi(\mathbb{1} - A)$ *is exponentially larger compared to* $\xi(\mathbb{1} - B)$*, is smallest at* $t = T$*.*

*Proof* For a symmetric matrix $A$ we have $\lambda_i(A) = \sigma_i(A)\ \forall i$. If $A, B$ are both symmetric $A + B$ is symmetric. So for symmetric $A, B$ finding the singular value gap of $(1 - g(t))(\mathbb{1} - A) + g(t)(\mathbb{1} - B)$ is equivalent to finding the gap between $\lambda_n((1 - g(t))(\mathbb{1} - A) + g(t)(\mathbb{1} - B))$ and $\lambda_{n-1}((1 - g(t))(\mathbb{1} - A) + g(t)(\mathbb{1} - B))$. Since $A, B$ are irreducible and aperiodic stochastic matrices, $(1 - g(t))A + g(t)B$ is a irreducible and aperiodic stochastic matrix, and therefore $\lambda_n((1 - g(t))(\mathbb{1} - A) + g(t)(\mathbb{1} - B)) = 0$. From Horn's inequalities [45] we get

[†]The defining characteristic for a column stochastic matrix is that all columns sum to unity and thus all columns of $P + Q$ sum to two. When $a + b = 1$, and $a, b$ are real we therefore have that every column of $aP + bQ$ sums to unity making it a column stochastic matrix.

[‡]Weyl's inequality states that $\prod_{i=1}^{k} |\lambda_i(A)| \leq \prod_{i=1}^{n} \sigma_i(A)$ with equality only when $k = n$. When $\lambda_n(A) = 0$ the product of all singular values has to be zero. This implies that the lowest singular value of $A$ also has to be equal to zero.

$$\lambda_i((1-g(t))(\mathbb{1}-A)) + \lambda_j(g(t)(\mathbb{1}-B)) \leq$$
$$\leq \lambda_k((1-g(t))(\mathbb{1}-A) + g(t)(\mathbb{1}-B)) \text{ if } \quad i+j = n+k \tag{3.15}$$

which we can write as

$$g(t)\lambda_{n-1}(\mathbb{1}-B) \leq \lambda_{n-1}((1-g(t))(\mathbb{1}-A) + g(t)(\mathbb{1}-B)) \tag{3.16}$$

and

$$(1-g(t))\lambda_{n-1}(\mathbb{1}-A) \leq \lambda_{n-1}((1-g(t))(\mathbb{1}-A) + g(t)(\mathbb{1}-B)) \tag{3.17}$$

which gives two monotonic lines as lower bounds, one increasing and one decreasing. From this we see that $\lambda_{n-1}((1-g(t))(\mathbb{1}-A) + g(t)(\mathbb{1}-B))$ is lowest at the intersection point of those two lines at

$$g(t) = \frac{\lambda_{n-1}(\mathbb{1}-A)}{\lambda_{n-1}(\mathbb{1}-A) + \lambda_{n-1}(\mathbb{1}-B)} \,. \tag{3.18}$$

Plugging this point into the lines gives a lower bound of

$$\frac{\lambda_{n-1}(\mathbb{1}-A)\lambda_{n-1}(\mathbb{1}-B)}{\lambda_{n-1}(\mathbb{1}-A) + \lambda_{n-1}(\mathbb{1}-B)} \,. \tag{3.19}$$

Now when $N$ large we have $\lambda_{n-1}(\mathbb{1}-A) \gg \lambda_{n-1}(\mathbb{1}-B)$ which results in a lower bound for large $N$ that tends towards $\lambda_{n-1}(\mathbb{1}-B)$ which concludes the proof. ■

To make a similar statement for general, non-symmetric, $A, B$ we would need lower bounds for the singular values of sums of matrices which is a challenging question even if the singular values of both matrices are known. However from numerical results we conjecture that a similar result holds for the problem we are looking at (see figure 3.2) and for a collection of other stochastic matrices (see appendix B). If we assume that the gap is indeed smallest at $t = T$ we can drastically reduce the number of numerical computations that are needed to analyse the time complexity of the adiabatic algorithm here.

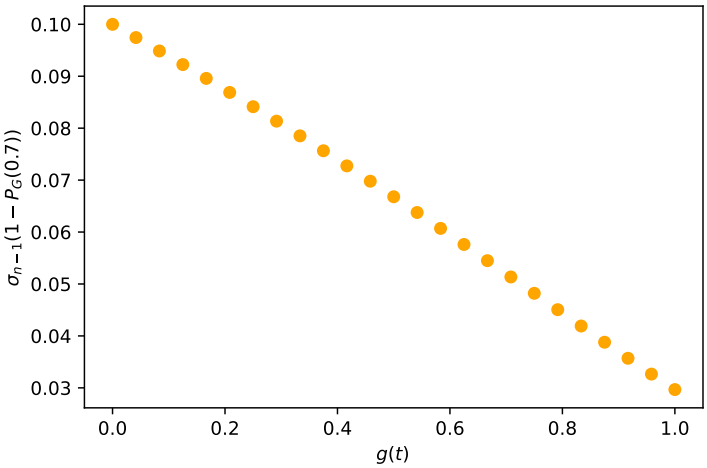


***Figure 3.2:*** *The second lowest singular values of* $(1-g(t))P_G(0)+g(t)P_G(\beta)$ *as a function of* $g(t)$*. Here* $N=10$ *and* $\beta=0.7$ *for the stochastic matrix corresponding with Glauber dynamics for the uniform zero-field Curie-Weiss model.*

In general the adiabatic algorithm has a quadratic (or higher power) dependence on the inverse minimal gap along the path. For many applications of the adiabatic algorithm the minimum gap is at an unknown point somewhere along the path between $t=0$ and $t=T$. This makes it hard to optimize the interpolating function $g(t)$ to improve the dependence on the gap in the time complexity. Roland & Cerf (2002) [46] showed that when an analytic expression for the gap along the path is available the time complexity can be optimised to linear in the gap. When we observe figure 3.2 we see that for our case the gap along the path seems to be a (linear) line. If we assume the gap along the path has this form at every $N$ we can use the same argument as used in Roland & Cerf (2002) to optimise the dependence of the eigenvalue gap of the time complexity of the algorithm. This derivation uses the approximate adiabatic condition defined in section 2.2.1. To see that this also holds for the more rigorous adiabatic condition see appendix A.1. When we look at figure 3.2 we see that the energy levels are closer together for $t$ close to $T$. We can therefore speed up the adiabatic evolution at lower $t$ and only slow down when we get close to $t=T$. To find what function to use for $g(t)$ we want to impose the adiabatic condition locally. To do this we want to solve the a differential equation of the form

$$\frac{\mathrm{d}g(t)}{\mathrm{d}t} = \epsilon\Delta_{\min}^2 = \epsilon[A - (A-B)g(t)]^2 \tag{3.20}$$

which we can integrate (and use that $g(0) = 0$) to find

$$g(t) = \frac{A^2\epsilon t}{A\epsilon t(A-B)+1} \tag{3.21}$$

To find the adiabatic time $T$ we solve $g(T) = 1$ to find

$$T = \frac{1}{\epsilon}\left[\frac{1}{AB}\right] . \tag{3.22}$$

For our problem $A = \sigma_{n-1}(\mathbb{1} - P(0))$ and $B = \sigma_{n-1}(\mathbb{1} - P(\beta))$. We are interested in the case where $\sigma_{n-1}(\mathbb{1} - P(0)) \gg \sigma_{n-1}(\mathbb{1} - P(\beta))$ for large $N$. So for large $N$ the adiabatic time goes with $1/\sigma_{n-1}(\mathbb{1} - P(\beta))$, i.e. linearly in the inverse smallest gap along the path.

### 3.2.2 Curie-Weiss model Numerical Results

We start by considering the singular value gap of $\mathbb{1} - P_G(\beta)$ for the stochastic matrix corresponding with Glauber dynamics for the uniform zero field Curie-Weiss model at $\beta = 0.7$. For this model we earlier found in section 2.4 there exists a classical mixing time transition at the critical inverse temperature $\beta_C = 1$. In figure 3.3 we compare those to the spectral gap of the stochastic matrix at $\beta = 0.7$ and $\beta = 1.4$. Keeping in mind that the quantum algorithm at $\beta = 0.7$ samples from the same distribution as the classical algorithm at $\beta = 1.4$, we see that our quantum algorithm seems to scale quadratically faster than the comparable classical mixing time.

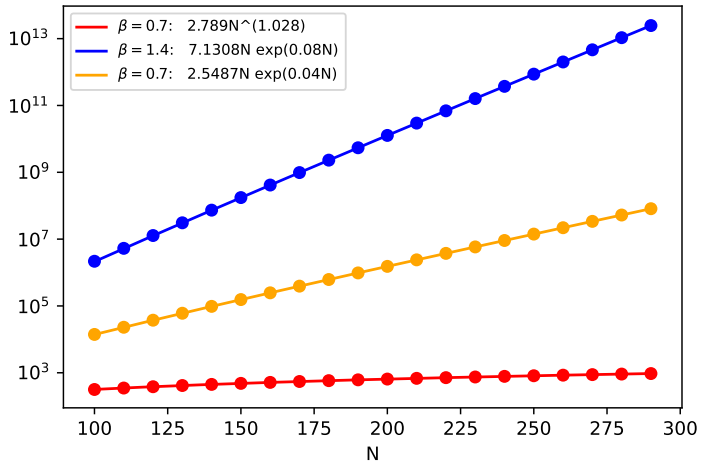


***Figure 3.3:*** *Inverse spectral gap of $P_G(\beta)$ (red and blue) and the inverse singular value gap of $\mathbb{1} - P_G(\beta)$ (orange) for the stochastic matrix corresponding with Glauber dynamics for the zero-field Curie-Weiss model at the temperature in the caption. Points indicate numerical results and lines fitted functions as given in the legend.*

One interesting aspect we can see from this plot is that the relevant inverse eigenvalue at $\beta = 0.7$ scales almost linearly in $N$ while the relevant inverse singular value at the same value for $\beta$ scales exponentially. We can look whether and at what value of $\beta$ the singular value has a complexity transition. This is shown in figure 3.4. The transition point for the singular value seems to be at $\beta = 0.5$ which is exactly half of the transition $\beta$ for the mixing time. This might be an indication that there is indeed an underlying connection between quantum complexity at $\beta$ and the classical complexity at $2\beta$. To see whether this is an effect specific to our choice of Gibbs sampler we look at the stochastic matrix corresponding with the Metropolis-Hastings sampler as well. In figure 3.5 we see the singular value gap and eigenvalue gaps for the stochastic matrix corresponding with Metropolis-Hastings for the uniform zero-field Curie-Weiss model at the same inverse temperatures. We observe the same behaviour as found for the stochastic matrix corresponding with Glauber dynamics. In figure 3.6 we again look where the complexity transition occurs. It again seems to occur where $2\beta$ crosses the critical inverse temperature.

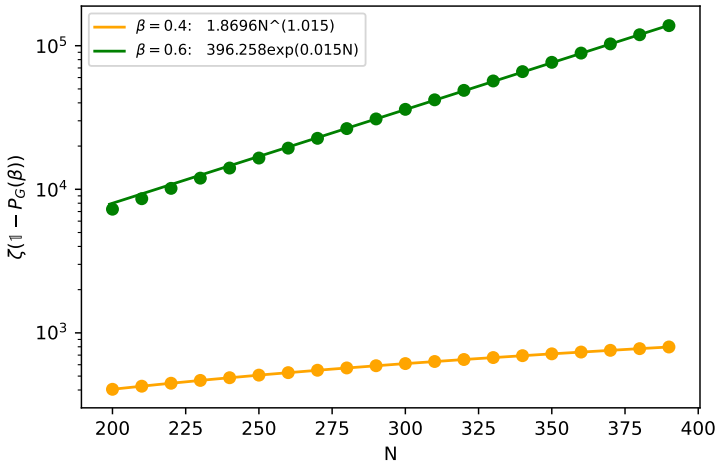


***Figure 3.4:*** *The inverse singular value gap of $\mathbb{1} - P_G(\beta)$ for the stochastic matrix corresponding with Glauber dynamics for the zero-field Curie-Weiss model at the temperature in the caption. Points indicate numerical results and lines fitted functions as given in the caption. Note the complexity phase transition somewhere between these two inverse temperatures.*

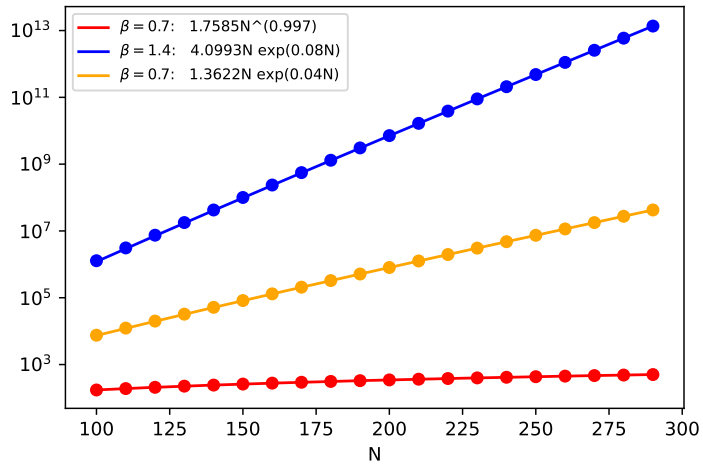


***Figure 3.5:*** *Inverse spectral gap of $P_G(\beta)$ (red and blue) and the inverse singular value gap of $\mathbb{1} - P_G(\beta)$ (orange) for the stochastic matrix corresponding with the Metropolis-Hastings algorithm for the zero-field Curie-Weiss model at the temperature in the legend. Points indicate numerical results and lines fitted functions as given in the legend.*

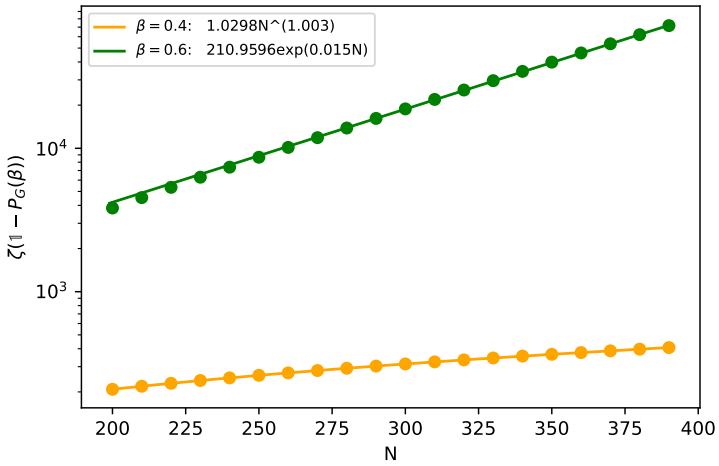


***Figure 3.6:*** *The inverse singular value gap of $\mathbb{1} - P_G(\beta)$ for the stochastic matrix corresponding with the Metropolis-Hastings algorithm for the zero-field Curie-Weiss model at the temperature in the legend. Points indicate numerical results and lines fitted functions as given in the caption. Note the complexity phase transition somewhere between these two inverse temperatures.*

### 3.2.3 Ising model on Regular Graphs

So far we considered the Ising model defined on the complete graph. To extend the results we got so far we want to look at the Ising model on different underlying graphs. When we consider the Ising model on a different graph we can no longer use the symmetric subspace projection to simplify the problem for higher spin numbers. For the low spin numbers we can calculate directly, we expect that finite size effects will distort the scaling of the time complexity. One aspect we can however look at for lower spin numbers is whether instances which are hard for the quantum algorithm are also instances which are hard for classical mixing. For this we need to look at a graph ensemble with multiple possible graphs at a given $N$. We choose to look at the Ising model on $k$-regular graphs. These $k$-regular graphs are graphs where each vertex has exactly $k$ edges. There are multiple possible $k$-regular graphs for a given number of spins so $k$-regular graphs are a graph ensemble. Here we want to look at the relative instance hardness for the classical and quantum algorithm.

For the Ising model on a $k$-regular graph we have a mixing time transition occurring at the critical inverse temperature $\beta_C$ such that $(k-1)\tanh\beta_C =$

1 [47]. Here we look at instances of 4-regular graphs and compare the singular value gap of $P_G(\beta)$ and the spectral gap of $P_G(2\beta)$. In figure 3.7 we see that the instances that are hard for classical mixing are also hard for our quantum algorithm.

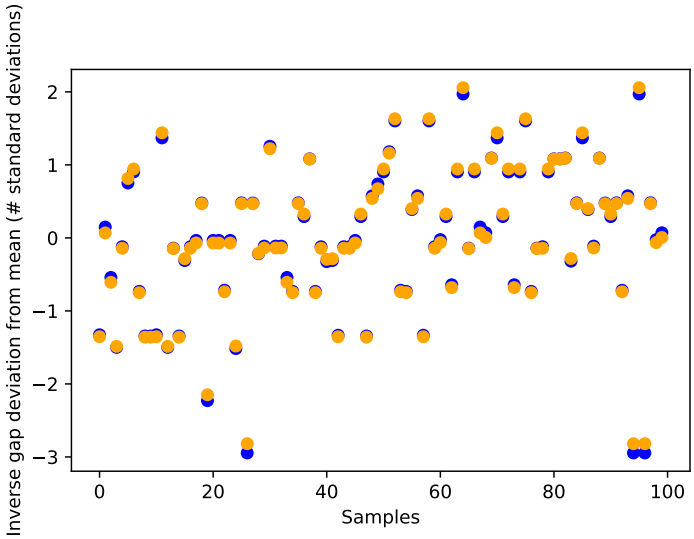


***Figure 3.7:*** *Deviation from the mean of the inverse spectral gap for $\beta = 0.5$ (blue) and inverse singular gap for $\beta = 0.25$ (orange) in units of standard deviations for the stochastic matrix corresponding with Glauber dynamics of an instance of the Ising model on 4-regular graphs. Each dot shows the result for 1 randomly sampled $N = 10$ 4 regular graph.*

In figure 3.7 the sampled graphs are unordered. We can see some more structure by ordering the sampled graphs by some parameter. Here we choose to order the graphs by their average local clustering coefficient which is defined as how connected the neighbours of a node are on average. In figure 3.8 we see the results for both gaps. We see that for both the classical and the quantum algorithm, the time complexity seems to have the same connection to how clustered the graph is giving further evidence to some connection between the hardness of preparing the quantum state at $\beta$ and classical sampling at $2\beta$.

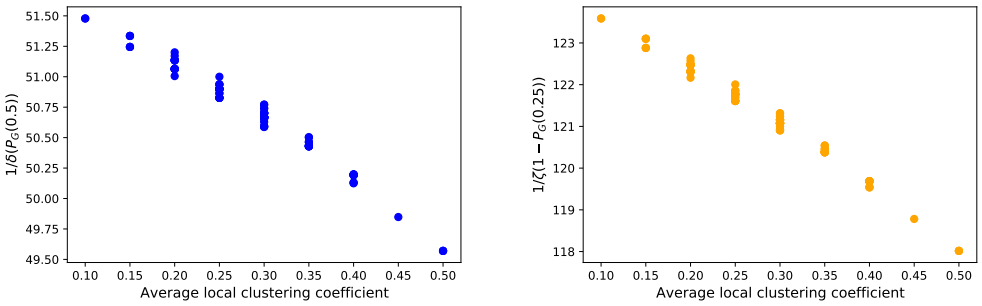


***Figure 3.8:*** *Inverse spectral gap for $\beta = 0.5$ (left panel) and inverse singular gap for $\beta = 0.25$ (right panel) against the average local clustering coefficient for the stochastic matrix corresponding with Glauber dynamics of an instance of the Ising model on 4-regular graphs. Each dot shows the result for 1 randomly sampled $N = 10$ 4 regular graph.*

## 3.3 Other State Preparation Methods

To see whether the observed behaviour is an effect of our specific algorithm or indicative of a more general result we look at other algorithms. The problem of preparing (an encoding of) the state $|\pi_G(\beta)\rangle$ was studied earlier by Orsucci (2019) [1]. Therein other algorithms to prepare a similar state were considered. Here we consider two of the proposed algorithms and use the symmetric subspace projection to extend the numerical analysis done in that work for higher spin numbers. For both these algorithms an indication of an exponential speed up over classical mixing was reported in Orsucci (2019) [1] when looking at numerical results for $N_{\max} = 20$.

### 3.3.1 Projection Methods

Another way to prepare $|\pi_G(\beta)\rangle$ is to directly project a state to the kernel of $\mathbb{1} - P_G(\beta)$. For this algorithm we assume we have an efficient $\alpha$-block-encoding [48] of $\mathbb{1} - P_G(\beta)$ and use qubitization [48] to approximate a projection on the kernel of $\mathbb{1} - P_G(\beta)$ [49]. In Orsucci (2019) [1] it is derived that the time complexity of this algorithm is given by

$$\tilde{O}(\alpha\zeta(\mathbb{1} - P_G(\beta))^{-1}|\langle\psi|\pi_G(\beta)\rangle|^{-1}) \tag{3.23}$$

with $|\psi\rangle$ an initial state we choose and $\alpha$ the normalisation parameter for the block encoding. The contribution from the inverse singular value gap

$\zeta(\mathbb{1} - P_G(\beta))^{-1}$ was already looked at for the adiabatic algorithm, those results can be found in section 3.2.2. In Orsucci (2019) [1] it was assumed we have some $|\psi\rangle$ with constant overlap with the state we want, we will here extend the analysis on this contribution to the time complexity. One possible advantage of looking at this contribution is that it is independent of the choice of Gibbs sampling matrix, i.e. when this contribution also scales exponentially in $N$ it might offer evidence for a broader effect. For this contribution we need to choose an initial state $|\psi\rangle$. Our goal is to sample in the high inverse temperature regime where $|\boldsymbol{\pi}_G(\beta)\rangle$ becomes a multi-modal distribution, where the modes are the ground states. Finding those ground states is at least as hard a problem as sampling so the closest distribution we can start at is $|\boldsymbol{\pi}_G(\beta)\rangle$ with $\beta$ in the easy regime, i.e. $\beta < 0.5$. To see how this overlap behaves we look at the overlap of two encoded Gibbs distribution at two different inverse temperature $\beta$,$\beta'$

$$\langle \boldsymbol{\pi}_G(\beta)|\boldsymbol{\pi}_G(\beta')\rangle = ((\sum_i \langle i| \frac{(\pi_G(\beta))_i}{||\boldsymbol{\pi}_G(\beta)||_2})(\sum_j \frac{(\pi_G(\beta'))_j}{||\boldsymbol{\pi}_G(\beta')||_2}|i\rangle) \tag{3.24}$$

$$= \sum_{i,j} \frac{(\pi_G(\beta))_i(\pi_G(\beta'))_j}{||\boldsymbol{\pi}_G(\beta)||_2||\boldsymbol{\pi}_G(\beta')||_2}\langle i|j\rangle \tag{3.25}$$

$$= \sum_i \frac{(\pi_G(\beta))_i(\pi_G(\beta'))_i}{||\boldsymbol{\pi}_G(\beta)||_2||\boldsymbol{\pi}_G(\beta')||_2} \tag{3.26}$$

$$= \sum_i \frac{1}{Z(\beta)Z(\beta')}\frac{\exp(-\beta H_i)\exp(-\beta' H_i)}{||\boldsymbol{\pi}_G(\beta)||_2||\boldsymbol{\pi}_G(\beta')||_2} \tag{3.27}$$

$$= \frac{Z(\beta+\beta')}{Z(\beta)Z(\beta')}\frac{1}{||\boldsymbol{\pi}_G(\beta)||_2||\boldsymbol{\pi}_G(\beta')||_2}\,. \tag{3.28}$$

We earlier found that the $L_2$ norm of $\boldsymbol{\pi}_G(\beta)$ is given by

$$||\boldsymbol{\pi}_G(\beta)||_2 = \frac{\sqrt{Z(2\beta)}}{Z(\beta)}\,. \tag{3.29}$$

Plugging this into our expression for $\langle \boldsymbol{\pi}_G(\beta)|\boldsymbol{\pi}_G(\beta')\rangle$ gives

$$\langle \boldsymbol{\pi}_G(\beta)|\boldsymbol{\pi}_G(\beta')\rangle = \frac{Z(\beta+\beta')}{Z(\beta)Z(\beta')}\frac{1}{||\boldsymbol{\pi}_G(\beta)||_2||\boldsymbol{\pi}_G(\beta')||_2} \tag{3.30}$$

$$= \frac{Z(\beta+\beta')}{\sqrt{Z(2\beta)Z(2\beta')}}\,. \tag{3.31}$$

When $\beta < \beta_C$ we have $Z(\beta) \sim 2^N$ and when $\beta > \beta_C$ we have $Z(\beta) \sim e^N$. When we choose $\beta$,$\beta'$ both below $\beta_C$ we expect this overlap to decrease relatively slowly while when we choose $\beta$ below $\beta_C$ but $\beta'$ above $\beta_C$ the overlap will decrease exponentially fast. To see whether this result holds for our model we want to numerically look at this overlap. Due to the permutation invariance of our system we can find the partition number by a weighted sum over the energy contribution for each Hamming weight state. We start by looking at the inverse overlap for the Gibbs distribution for $\beta = 0.7$ and $\beta' = 0.4$ for the uniform zero-field Curie-Weiss model (figure 3.9). In that figure we see that this overlap seems to decrease exponentially fast for these choices of $\beta$,$\beta'$. When we combine these results with those found in figure 3.3 and figure 3.5 we see potential for at most a polynomial speed up over the relevant classical mixing results in those same two figures.

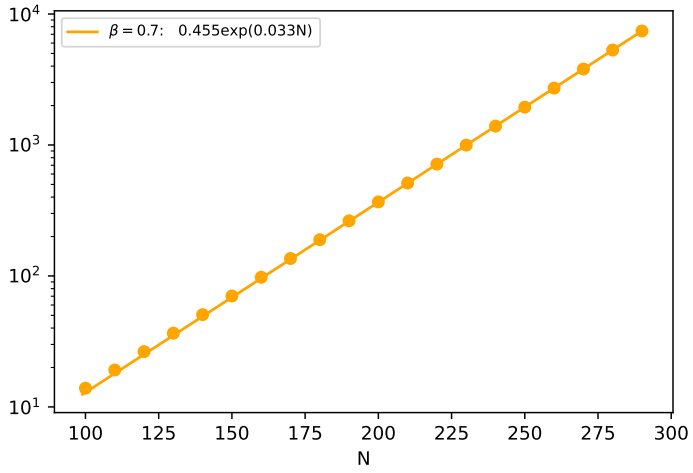


***Figure 3.9:*** *The inverse overlap between $|\pi_G(0.4)\rangle$ and $|\pi_G(0.7)\rangle$. Point indicate numerical results and lines the fitted functions given in the legend.*

We again want to know where this transition occurs, see figure 3.10. We see that the transition again seems to occur where $2\beta$ is above the critical inverse temperature.

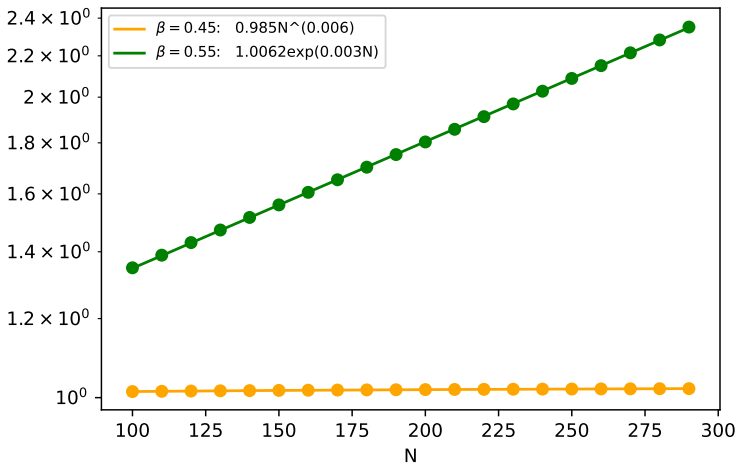


***Figure 3.10:*** *The inverse overlap between $|\pi_G(0.4)\rangle$ and $|\pi_G(\beta)\rangle$ for $\beta$ given in the caption. Point indicate numerical results and lines the fitted functions given in the caption. Note there seems to be a complexity transition in between those two inverse temperatures.*

### 3.3.2 Relaxation Method

The algorithms shown so far aim to prepare the state $|\pi_G(\beta)\rangle$ by finding the kernel of $\mathbb{1} - P_G(\beta)$. In Orsucci (2019) [1] it was shown that we can instead solve the linear system problem given by

$$(\mathbb{1} - \mu P_G(\beta))\boldsymbol{x} = \boldsymbol{b} \tag{3.32}$$

with $\mu \in (0,1)$ a free parameter and $\boldsymbol{b} \in \mathbb{C}^n$ an arbitrary vector. Since $(\mathbb{1} - \mu P_G(\beta))$ is a non-singular matrix we can write $\boldsymbol{x} = (\mathbb{1} - \mu P_G(\beta))^{-1}\boldsymbol{b}$. We now use the geometric series to write

$$\boldsymbol{x} = (\mathbb{1} - \mu P_G(\beta))^{-1}\boldsymbol{b} = \sum_k \mu^k P^k \boldsymbol{b}\,. \tag{3.33}$$

In Orsucci (2019) [1] it is shown that when we choose $\mu = 1 - 1/(\gamma t_{\text{mix}}(\epsilon))$ the solution $\boldsymbol{x}$ is within distance $\epsilon + 1/\gamma$ of the stationary distribution $\pi_G(\beta)$ for an arbitrary choice of $\boldsymbol{b}$.

We showed in section 2.2.3 that the time complexity of preparing a state that is proportional to the solution of a linear system of equations on a

quantum computer is quasilinear in the condition number $\kappa$ and $\log n$. We now want to introduce new bounds on the condition number of $\mathbb{1} - \mu P$. We write the condition number as

$$\kappa(\mathbb{1} - \mu P) = ||\mathbb{1} - \mu P||_2 \, ||A(\mathbb{1} - \mu P)^{-1}||_2 \, . \tag{3.34}$$

We now use the following inequality (which is derived from Hölder's inequality [50])

$$||A||_2 \leq \sqrt{||A||_1 \, ||A||_\infty} \tag{3.35}$$

to write

$$||\mathbb{1} - \mu P||_2 \leq \sqrt{||\mathbb{1} - \mu P||_1 \, ||\mathbb{1} - \mu P||_\infty} \, . \tag{3.36}$$

The $||\cdot||_1$ is given by the maximum absolute column sum of the matrix and $||\cdot||_\infty$ by the maximum absolute row sum. $\mathbb{1} - P$ has all columns absolute summing at most two so $\mathbb{1} - \mu P$ with $\mu \in (0, 1)$ has a maximum column sum smaller or equal to two. Since we are looking at time-reversible single spin flip Markov chains the maximum absolute row sum of $\mathbb{1} - \mu P$ is $O(N)$. Hence we have

$$||A||_2 \leq O(\sqrt{N}) \, . \tag{3.37}$$

We know that there have to be instances where $\kappa(\mathbb{1} - \mu P)$ increases exponentially so this contribution has to be from the $||A(\mathbb{1} - \mu P)^{-1}||_2$ term. We can again use the geometric series to write $(\mathbb{1} - \mu P)^{-1} = \sum_{l=0}^{\infty} \mu^l P^l$ and

$$||(\mathbb{1} - \mu P)^{-1}||_2 = ||\sum_{l=0}^{\infty} \mu^l P^l||_2 \, . \tag{3.38}$$

When $l \geq t_{mix}$ we have that $d(P^l \boldsymbol{b}, \boldsymbol{\pi}) \leq \epsilon$ for every stochastic matrix $\boldsymbol{b}$ hence for $l \geq t_{mix}$ we get that $P$ is a matrix where every column is approximately equal to the stationary distribution $\boldsymbol{\pi}$ (see appendix A.3). We define this matrix as $Q$ and therefore write for $l \geq t_{mix}$ that $P^l = Q$. Using this we write

$$\sum_{l=0}^{\infty} \mu^l P^l = \sum_{0\leq l<t_{\text{mix}}} \mu^l P^l + \sum_{t_{\text{mix}}\leq l<\infty} \mu^l P^l \tag{3.39}$$

$$= \sum_{0\leq l<t_{\text{mix}}} \mu^l P^l + Q \sum_{t_{\text{mix}}\leq l<\infty} \mu^l < \sum_{0\leq l<t_{\text{mix}}} \mu^l P^l + Q\gamma t_{\text{mix}} \tag{3.40}$$

using that $\mu < 1$ so $\sum_{t_{\text{mix}}\leq k} \mu^k < \sum_k \mu^k = 1/(1-\mu) = \gamma t_{\text{mix}}$ in the last step. We thus write

$$||\sum_{l=0}^{\infty} \mu^l P^l||_2 < ||\sum_{0\leq l<t_{\text{mix}}} \mu^l P^l||_2 + \gamma t_{\text{mix}}||Q||_2 \,. \tag{3.41}$$

The $||\cdot||_2$ norm of a matrix is given by the highest singular value of that matrix. When $\beta$ below the critical inverse temperature for high $N$ the stationary distribution is close to uniform and therefore the matrix $Q$ is close to uniform as well. Here the matrix is symmetric so the 2-norm is given by the largest eigenvalue of $Q$. For the uniform stochastic matrix[§] the largest eigenvalue is equal to one. When $\beta$ above the critical inverse temperature the stationary distribution for large $N$ goes towards a sharply peaked multi modal distribution. In general the largest singular value of a matrix $A$ is given by the square root of the largest eigenvalue of $A^T A$. Here $Q$ is given by vectors with all elements close to zero except at the ground states. Hence $Q^T Q$ tends towards a diagonal matrix where the only non-zero elements are those corresponding to the ground states which have magnitude $\sim 2^N$. This might explain why for $\beta$ above the critical inverse temperature the condition number has to increase exponentially fast.

To investigate these bounds and to compare the time complexity of this algorithm with our adiabatic preparation algorithm we numerically look this at condition number. With the right choice of operator norm the condition number is given by $\sigma_1(\mathbb{1} - \mu P_G(\beta))/\sigma_n(\mathbb{1} - \mu P_G(\beta))$. We start by looking at the stochastic matrix corresponding with Glauber dynamics for the uniform zero-field Curie Weiss model at $\beta = 0.7$ in figure 3.11. We see behaviour that is very similar to what we saw in section 3.2.2 for the adiabatic based algorithm and since the QLSP runtime is quasilinear in $\kappa$ we see a quadratic speed up over the relevant classical mixing results. We again wish to see where the transition occurs. In figure 3.12 we see that this transition again seems to occur where $2\beta$ is above the critical inverse temperature.

[§]Stochastic matrix where every element is given by $1/n$, where $n$ is the number of states in the Markov chain.

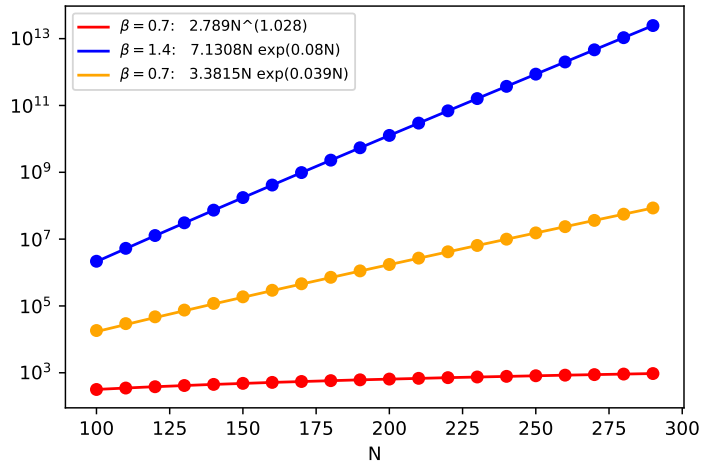


***Figure 3.11:*** *Inverse spectral gap of $P_G(\beta)$ (red and blue) and the condition number of $\mathbb{1} - \mu P_G(\beta)$ (orange) for the stochastic matrix corresponding with Glauber dynamics for the zero-field Curie-Weiss model at the temperature in the caption. Points indicate numerical results and lines fitted functions as given in the legend.*

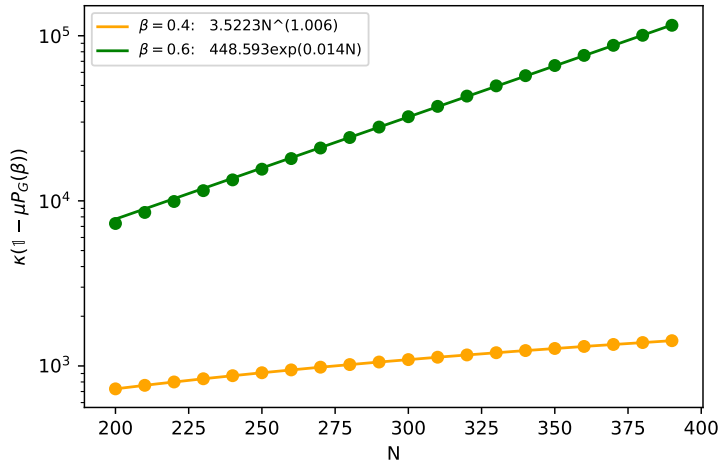


***Figure 3.12:*** *The condition number of $\mathbb{1} - \mu P_G(\beta)$ for the stochastic matrix corresponding with Glauber dynamics for the zero-field Curie-Weiss model at the temperature in the caption. Points indicate numerical results and lines fitted functions as given in the legend. Note the complexity phase transition somewhere between these two inverse temperatures.*

In figure 3.13 and 3.14 these results are repeated for the stochastic matrix

corresponding with the Metropolis-Hastings algorithm giving the same transition inverse temperature and quadratic speed up.

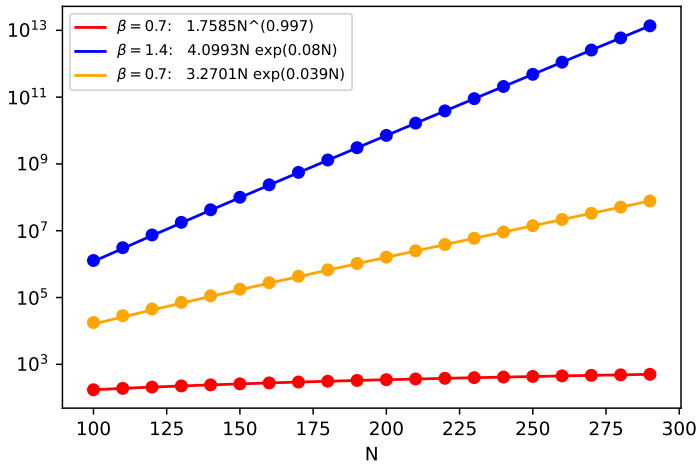


***Figure 3.13:*** *Inverse spectral gap of $P_G(\beta)$ (red and blue) and the condition number of $\mathbb{1} - \mu P_G(\beta)$ (orange) for the stochastic matrix corresponding with the Metropolis-Hastings algorithm for the zero-field Curie-Weiss model at the temperature in the caption. Points indicate numerical results and lines fitted functions as given in the legend.*

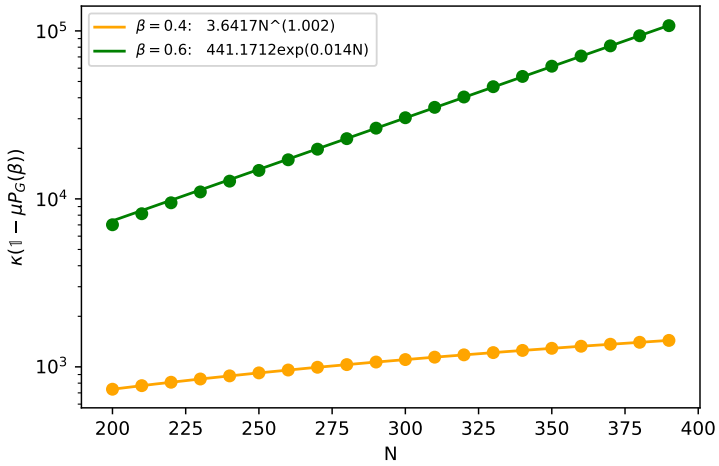


***Figure 3.14:*** *The condition number of $\mathbb{1} - \mu P_G(\beta)$ for the stochastic matrix corresponding with the Metropolis-Hastings algorithm for the zero-field Curie-Weiss model at the temperature in the caption. Points indicate numerical results and lines fitted functions as given in the legend. Note the complexity phase transition somewhere between these two inverse temperatures.*

As mentioned in section 2.2.3 we can use preconditioning to lower the condition number of the matrix. The matrix we are looking at here numerically seems to be close to diagonally dominant. To use the SParse Approximate Inverse (SPAI) algorithm we defined in section 2.2.3 we need to choose a sparsity pattern. For diagonally dominant matrices a banded matrix sparsity pattern is often used [35]. Since applying this preconditioning breaks the permutation invariance we can only numerically look at the condition number from the full matrix. As a baseline we first look at the condition number at $\beta = 0.7$ for stochastic matrix corresponding with Glauber dynamics for the uniform zero-field Curie-Weiss model without any preconditioning applied (see figure 3.15).

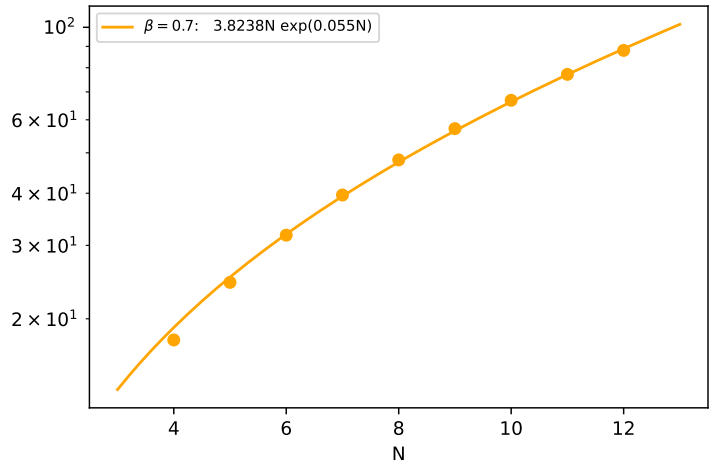


***Figure 3.15:*** *Condition number of $1 - \mu P_G(\beta)$ for the stochastic matrix corresponding with Glauber dynamics for the zero-field Curie Weiss model without any preconditioning at $\beta = 0.7$. Points indicate numerical results and lines fitted functions as given in the legend.*

We then apply the SPAI using the banded matrix sparsity pattern for an increasing number of bands (see figures 3.16 and 3.17). Each of the applied preconditioners lowers the condition number of the matrix as expected. However the preconditioning seems to have the effect that, while for each instance the condition number decreases, the scaling seems to become more steep. This could potentially be understood since the applied sparsity pattern has a fixed number of bands the pattern becomes more sparse for increasing $N$ so the SPAI works better at lower $N$ giving rise to a steeper scaling.

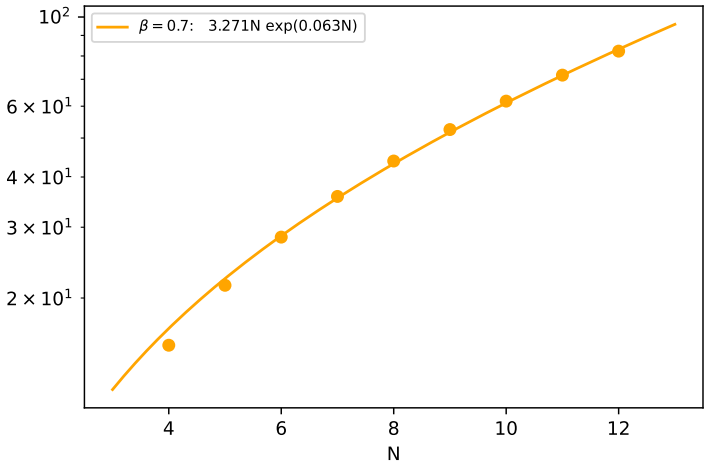


***Figure 3.16:*** *Condition number of $1 - \mu P_G(\beta)$ for the stochastic matrix corresponding with Glauber dynamics for the zero-field Curie Weiss model with a 2 band SPAI applied at $\beta = 0.7$. Points indicate numerical results and lines fitted functions as given in the legend.*

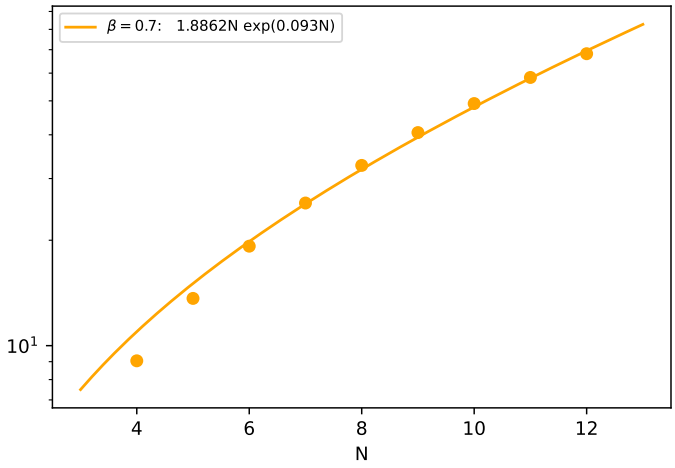


***Figure 3.17:*** *Condition number of $1 - \mu P_G(\beta)$ for the stochastic matrix corresponding with Glauber dynamics for the zero-field Curie Weiss model with a 5 band SPAI applied at $\beta = 0.7$. Points indicate numerical results and lines fitted functions as given in the legend.*

One potential improvement could therefore be to let the number of bands scale linearly with increasing $N$. Since the matrix dimension scales ex-

ponentially in $N$ the linearly increasing number of bands still results in a sparse sparsity pattern. In figure 3.18 we see that this preconditioner has the a similar effect as the fixed band number preconditioners.

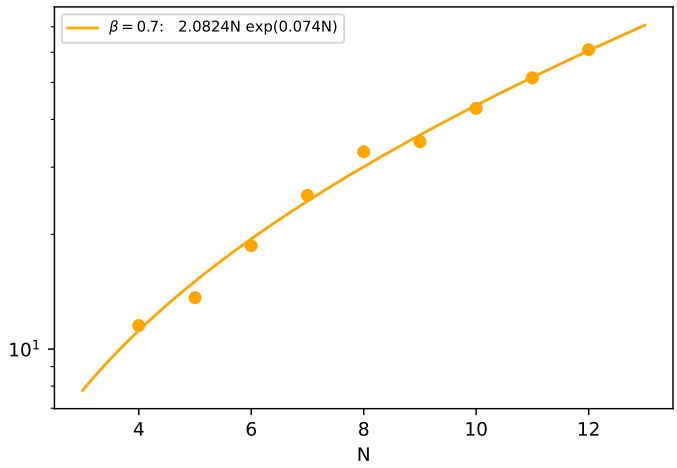


***Figure 3.18:*** *Condition number of $1 - \mu P_G(\beta)$ for the stochastic matrix corresponding with Glauber dynamics for the zero-field Curie Weiss model with a $N$ band SPAI applied at $\beta = 0.7$. Points indicate numerical results and lines fitted functions as given in the legend.*

Chapter 4

# Discussion & Conclusions

## 4.1 Discussion

We proposed an algorithm that prepares the stationary distribution of a stochastic matrix $P$ by using the adiabatic algorithm to prepare a state proportional to the 0 eigenstate of $\mathbb{1} - P$. We noticed that when we prepare the quantum state proportional to this stationary distribution measuring the state corresponds to sampling from a different distribution we called the squared distribution. For stochastic matrices where the stationary distribution is given by the Gibbs distribution this squared distribution is equivalent to the Gibbs distribution at twice the inverse temperature. For many stochastic matrices with the Gibbs distribution sampling at twice the inverse temperatures is in some conditions exponentially harder. We explored whether this might allow for a potential exponential speedup by preparing the quantum state proportional to the stationary distribution at inverse temperature $\beta$ and measuring that state to sample from the stationary distribution at inverse temperature $2\beta$.

We offered an analytical argument why this exponential speedup might not be possible due to the presence of the partition function at $2\beta$ in the normalisation constant. If this argument holds there should be an complexity transition for the quantum algorithm at half the inverse temperature at which the classical mixing time has a complexity transition. We numerically investigated this argument by looking at two stochastic matrices that have the Gibbs distribution as the stationary distribution for the uniform zero-field Curie-Weiss model. For both matrices we found that there exists a complexity transition for the quantum algorithm at halve the classical critical temperature giving evidence that our argument could

be valid. A question that remains is what mechanism in the specific algorithm causes this transition to happen at that point. The classical mixing time is bounded by the second smallest eigenvalue of $\mathbb{1} - P_G(\beta)$ and the time complexity of the adiabatic state preparation algorithm for this state is bounded by the second lowest singular value of $\mathbb{1} - P_G(\beta)$. A link between the quantum complexity at $2\beta$ and the classical complexity at $\beta$ would therefore imply a link between the singular values of $\mathbb{1} - P_G(\beta)$ and the eigenvalues of $\mathbb{1} - P_G(2\beta)$. The singular values of a matrix $A$ are given by the square root of the eigenvalues of the matrix $AA^\dagger$. To analyse the possible connection between the singular values at $\beta$ and the eigenvalues at $2\beta$ we therefore need to look at eigenvalues of $\mathbb{1} - P_G(\beta) - (P_G(\beta))^T + P_G(\beta)(P_G(\beta))^T$ which is a challenging problem. We know that when $P$ symmetric the eigenvalues are equal to the singular values so the link between the the singular values at $\beta$ and the singular values at $2\beta$ can not exist in general. For the Gibbs samplers we looked at we always observe the second lowest singular value to be smaller than the second lowest eigenvalue, however, when looking at more general stochastic matrices this does not always have to be the case. These observations make it unlikely that there exists a general rule for this connection.

From numerical results we noticed that the adiabatic gap along the path for this problem seems to be close to linearly decreasing as a function of the interpolating function. When we assume this linear relation holds in general we can optimise algorithm and find a quadratic speedup for our algorithm over classical mixing in some circumstances. However unless an analytic proof is found for this linearly decreasing adiabatic gap it is unclear whether this assumption holds for higher values of $N$. This quadratic speedup over classical mixing is however for a model for which there exists faster classical algorithms to sample from the Gibbs distribution than classical mixing. The method we used to analyse our algorithm with time complexity increasing polynomially with $N$ time can also be used to find the Gibbs distribution for this model polynomially with $N$ for every temperature. To show that our algorithm offers a speedup over the fastest classical algorithm we would need to look at a different model. However, the models for which there is no faster algorithm than classical mixing are also the models we can only analyse for low spin numbers where the scaling is distorted by finite-size effect.

When looking at the family of regular graphs we noticed that the instances that are classically hard also seem to be the instances which are hard for our quantum algorithm. For both the classical and quantum algorithm

the hardness seems to scale inversely proportional to how clustered the graphs are. Giving again evidence to a link between the quantum algorithm at $\beta$ and the classical mixing at $2\beta$.

We also looked at two algorithms proposed by Orsucci (2019) [1]. For these algorithms we extended the numerical analysis from $N_{\max} = 20$ to $N_{\max} = 290$ and found the indication of an exponential speed up presented in Orsucci (2019) [1] to not be present at higher $N$. For those two algorithms we do see evidence for a polynomial speed up for this model over classical mixing. These algorithms also prepare the same encoding of $|\pi_G(\beta)\rangle$ as our adiabatic based algorithm and therefore also allow to sample from the distribution at $2\beta$. For these algorithms we also see a time complexity transition at the point where the quantum algorithms would allow to sample from the classically hard regime. For all three algorithms for preparation of $|\pi_G(\beta)\rangle$ there seems to be the same time complexity transition at the point where measuring $|\pi_G(\beta)\rangle$ would allow to sample from the classically hard regime. Presence of this behaviour for all three quite different algorithms further points to an underlying connection between sampling from $|\pi_G(\beta)\rangle$ and classically sampling from $\pi_G(2\beta)$.

In section 3.2.1 we offered an argument for why an exponential speedup using this state encoding is unlikely, however, this does not disprove the existence of an algorithm that does offer an exponential speedup. We looked at a straightforward preconditioning scheme to see if that offers potential for a further speedup however that is not present in our results.

## 4.2 Outlook

The main open question is whether the quadratic speedup over classical mixing by preparing this state encoding found here for the uniform zero-field Curie-Weiss model extends to Gibbs sampling for other classical systems. To explore this numerical analysis could be performed for more models. A difficulty is that for many systems the scaling of the mixing time at low dimensions of the system size is dominated by finite-size effects. For the uniform zero-field Curie-Weiss model we circumvented this by using the symmetric subspace projection, however, for most systems that is not possible. To make a more general (analytical) statement on the possibility of a general quadratic speedup for Gibbs sampling we would need to find an expression for the second lowest singular value or condition number of $\mathbb{1} - P_G(\beta)$. This is a challenging problem even if all the

singular values of $P_G(\beta)$ are known.

We proposed to use the state encoding for sampling from Gibbs distributions. The main advantage of applying this to Gibbs distributions is that there we have an interpretation of the squared distribution. The algorithms proposed here could also be used to sample from the stationary distribution of more general stochastic matrices, however, there we have no interpretation for the squared distribution making comparison to classical algorithms difficult.

In this work we looked at applications of the encoding we defined for Gibbs sampling. There are other places where this encoding could be relevant. When we have a distribution with one element given by $1/\sqrt{2^N}$ and uniform elsewhere the squared version of this distribution gives the marked element with constant probability $1/2$ (see appendix A.2 for more details). Here the original distribution offers no information on the marked element in the large $N$ limit where the squared distribution gives the marked element with constant probability. This could potentially be used in relation to oracular problems.

## 4.3 Conclusions

We introduced a novel quantum algorithm to prepare an encoding of the Gibbs distribution through adiabatic state preparation of an encoding of the $+1$ eigenvector of a stochastic matrix $P$. We found that measuring this encoding of the Gibbs distribution at inverse temperature $\beta$ to be equivalent to sampling from the Gibbs distribution at $2\beta$. We found that adiabatic preparation of this encoding of the $+1$ eigenstate to be quadratically faster compared to classical mixing for the uniform zero-field Curie-Weiss model. We also looked at two algorithms from Orsucci (2019) [1] and extended the analysis and numerical results therein. We found that the indications of an exponential speedup presented there to not be present at higher $N$ however an indication of a polynomial speedup over classical mixing remains. We conjectured that the presence of a normalisation factor of $1/Z(2\beta)$ might fundamentally limit the potential for speedup for algorithms based on this state encoding.

# Bibliography


[1] Davide Orsucci. *Quantum Computation Models and Their Application*. PhD thesis, University of Innsbruck, June 2019.

[2] Jos Thijssen. *Computational Physics*. Cambridge University Press, 2nd edition, 2007.

[3] Alan E. Gelfand. Gibbs sampling. *Jour. of the Amer. Stat. Assc.*, 95:1300–1304, 2000.

[4] Asja Fischer and Christian Igel. Training restricted boltzmann machines: An introduction. *Pattern Recog.*, 47(1):25–39, 2014.

[5] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer-Verlag, 2nd edition, 2006.

[6] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3 edition.

[7] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[8] Francisco Barahona. On the computational complexity of ising spin glass models. *J. Phys. A: Math. Gen*, 15:3241–3253, 1982.

[9] Andrew Lucas. Ising formulations of many np problems. *Front. Phys.*, 12, 2014.

[10] Jian Ding, Eval Lubetzky, and Yuval Peres. The mixing time evolution of glauber dynamics for the mean-field ising model. *Comm. in Math. Phys.*, 289:725–764, 2009.

[11] Scott Aaronson. P $\stackrel{?}{=}$ NP. 2017. Found on `https://www.scottaaronson.com/papers/pnp.pdf`.

[12] Anirban Narayan Chowdhury and Rolando D. Somma. Quantum algorithms for gibbs sampling and hitting-time estimation. *Quan. Inf. & Comp.*, 17, 2017.

[13] Rolando D. Somma, S. Boixo, H. Barnum, and E. Knill. Quantum simulations of classical annealing processes. *Phys. Rev. Lett.*, 101(13), 2008.

[14] Shantanav Chakraborty, Kyle Luh, and Jérémie Roland. Analog quantum algorithms for the mixing of markov chains. *Phys. Rev. A*, 102, 2020.

[15] Peter C. Richter. Almost uniform sampling via quantum walks. *New Jour. of Phys.*, 9, 2007.

[16] Fernando G. S. L. Brandão, Amier Kalev, Tongyang Li, Cendric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. Quantum sdp solvers: Large speed-ups, optimality, and applications to quantum learning. 2019. arXiv:1710.02581.

[17] Joran van Apeldoorn, Adrás Gilyén, Sander Gribling, and Ronald de Wolf. Quantum sdp-solvers: Better upper and lower bounds. *Quantum*, 4, 2020.

[18] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103(15), 2009.

[19] B.D. Clader amd B.C. Jacobs and C.R. Sprouse. Preconditioned quantum linear system algorithm. *Phys. Rev. Lett.*, 110, 2013.

[20] Scott Aaronson. Read the fine print. *Nat. Phys.*, 11:291–293, 2015.

[21] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Rev. Mod. Phys.*, 90, 2018.

[22] Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landua, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Review*, 50(4):755–787, 2008.

[23] Davide Pastorello, Enrico Blanzieri, and Valter Cavecchia. Learning adiabatic quantum algorithms over optimization problems. *Quan. Mach. Int.*, 3, 2021.

[24] Junyu Liu and Yuan Xin. Quantum simulation of quantum field theories as quantum chemistry. *Jour. of H. En. Phys.*, 2020.

[25] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. 2000. arXiv:quant-ph/0001106.

[26] J. J. Sakurai and Jim Napolitano. *Modern Quantum Mechanics*. Cambridge University Press, 2 edition, 2017.

[27] Daniel P. Bovet and Pierluigi Crescenzi. *Introduction to the Theory of Complexity*. Prentice Hall, 1994.

[28] John Watrous. *Quantum Computational Complexity*, pages 7174–7201. Springer New York, New York, NY, 2009.

[29] Marko D. Petković and Predrag S. Stanimirović. Generalized matrix inversion is not harder than matrix multiplication. *Jour. of Comp. and Appl. Math.*, 230:270–282, 2009.

[30] Yiğit Subaşi, Rolando D. Somma, and Davide Orsucci. Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Phys. Rev. Lett.*, 122, 2019.

[31] Carlos Bravo-Prieto, Ryan LaRose, M. Cerezo, Yiğit Subaşi, Lukasz Cincio, and Patrick J. Coles. Variational quantum linear solver. 2020. arXiv:1909.05820.

[32] A. M. Childs, R. Kothari, and R. D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM J. Comp.*, pages 1920–1950, 2017.

[33] L. Wossnig, Z. Zhao, and A. Prakash. Quantum linear system algorithm for dense matrices. *Phys. Rev. Lett.*, 5, 2018.

[34] Changpeng Shao and Hua Xiang. Quantum circulant preconditioner for a linear system of equations. *Phys. Rev. A*, 98, 2018.

[35] Edmond Chow. A priori sparsity patterns for parallel sparse approximate inverse preconditioners. *SIAM J. Sci. Comput.*, 21(5):1804–1822, 2006.

[36] M. Kochmański, T. Paszkiewicz, and S. Wolski. Curie-weiss magnet - a simple model of phase transition. *Eur. J. Phys.*, 34, 2013.

[37] E. Stiefel and A. Fässler. *Group Theoretical Methods and Their Applications*. Birkhäuser Basel, 1992.

[38] Aram W. Harrow. The church of the symmetric subspace. 2013. arXiv:1308.6595.

[39] Koenraad M. R. Audenaert. A digest on representation theory of the symmetric group. Found on `http://personal.rhul.ac.uk/usah/080/QITNotes_files/Irreps_v06.pdf`.

[40] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, Jarrod K. Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[41] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[42] Guido Van Rossum and Fred L. Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[43] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[44] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[45] William Fulton. Eigenvalues, invariant factors, highest weights, and schubert calculus. *Bull. Amer. Math. Soc.*, 37:209–249, 2000.

[46] Jérémie Roland and Nicolas J. Ceff. Quantum search by local adiabatic evolution. *Phys. Rev. A*, 65, 2002.

[47] Elchanan Mossel and Allan Sly. Exact thresholds for ising-gibbs samplers on general graphs. *Ann. Probab.*, 41:294–328.

[48] Guang Hao Low and Isaac L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019.

[49] András Gilyén, Yuan Su, Gunang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. *Proc. of the Ann. ACM Symp. on Th. of Comp. - STOC '19*, pages 193–204, 2019.

[50] Denis Serre. *Graduate Texts in Mathematics: Matrices, Theory and Applications*. Springer-Verlag, 2010.

[51] Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler. Bounds for the adiabatic approximation with applications to quantum computation. *Jour. of Math. Phys.*, 48, 2007.

Appendix A

# Extended Mathematical Arguments

## A.1 Extended Optimisation Adiabatic Condition

In the main work we use the approximate adiabatic condition to optimize the time complexity of the adiabatic algorithm to a linear dependence on the smallest inverse gap along the path. Here we will show this result also holds when considering a more rigorous version of the adiabatic condition. We will be adapting the work done in Jansen, Seiler & Ruskai (2007) [51] to our algorithm.

In [51] it is shown that the adiabatic condition

$$| \langle E_i(T) | \psi(T) \rangle |^2 \geq 1 - \epsilon(T)^2 \tag{A.1}$$

holds when $\epsilon(T)$ satisfies

$$\epsilon(T) \leq \frac{1}{T} \frac{m||\dot{H}||}{\Delta^2}\bigg|_{\text{u.b.}} + \frac{1}{T} \int_0^T \left( \frac{m||\ddot{H}||}{\Delta^2} + 7m\sqrt{m} \frac{||\dot{H}||^2}{\Delta^3} \right) \mathrm{d}t \tag{A.2}$$

with $\Delta$ the eigenvalue gap at a point in the adiabatic evolution and $m$ the degeneracy of the eigenstate of $H(t)$ we want to remain close to. We use the notation $f|_{\text{u.b.}} = f(0) + f(1)$. Here we have a Hamiltonian

$$H(t) = (1 - g(t))H_0 + g(t)H_1 \tag{A.3}$$

and we assumed that the gap along the path is of the form

$$\Delta(g(s)) = A - (A - B)g(s)\,, \tag{A.4}$$

where for our problem $A$ is second lowest singular value of $\mathbb{1} - P_G(0)$ and $B$ the second lowest singular value of $\mathbb{1} - P_G(\beta)$ for some $\beta$. Now we want to find a function $f$ that is the solution of

$$g(0) = 0 \tag{A.5}$$
$$\dot{g}(t) = k\Delta^p(g(t)) \tag{A.6}$$
$$k = \int_0^T \Delta^{-p}(t)\mathrm{d}t \tag{A.7}$$

with $p$ a parameter we want to optimise. From this we get

$$\dot{H}(s) = k\Delta^p(g(t))(H_1 - H_0) \tag{A.8}$$
$$\ddot{H}(s) = k^2\Delta^{2p-1}(g(t))\dot{\Delta}(f(s))(H_1 - H_0) \tag{A.9}$$

Now we want to plug this all into equation A.10 and see $||H_1 - H_0|| \leq 2$ to write

$$\epsilon(T) \leq \frac{1}{T}\frac{m||\dot{H}||}{\Delta^2}\bigg|_{\text{u.b.}} + \frac{1}{T}\int_0^T \left(\frac{m||\ddot{H}||}{\Delta^2} + 7m\sqrt{m}\frac{||\dot{H}||^2}{\Delta^3}\right) \tag{A.10}$$
$$\leq \frac{2k}{T}\left(\frac{\Delta^p}{\Delta^2}\bigg|_{\text{u.b.}} + \int_0^T \left[\frac{2k\Delta^{2p-1}(g(t))|\dot{\Delta}(g(t))|}{\Delta^2(g(t))} + \frac{28k\Delta^{2p}(g(t))}{\Delta^3(g(t))}\right]\right) \tag{A.11}$$
$$= \frac{2k}{T}\left(\Delta^{p-2}\Big|_{\text{u.b.}} + \int_0^T \left[2k\Delta^{2p-3}(g(t))|\dot{\Delta}(g(t))| + 28k\Delta^{2p-3}(g(t))\right]\right) \tag{A.12}$$
$$= \frac{2k}{T}\left(\Delta^{p-2}(0) + \Delta^{p-2}(1) + \int_0^1 \left[2\Delta^{p-3}(u)|\dot{\Delta}(u)| + 28\Delta^{p-3}(u)\right]\right) \tag{A.13}$$

where we used the change of integration variables $u = g(t)$ in the last line. For our problem we are interested in cases where $\Delta(1) \ll \Delta(0)$ so for $1 < p < 2$ we get $\Delta^{p-2}(0) + \Delta^{p-2}(1) = O(\Delta^{p-2}(1)) = O(\Delta_{\min}^{p-2})$. We also observe that

$$\int \Delta^{-p}(u)\mathrm{d}u = \int (\Delta(0) - (\Delta(0) - \Delta(1))u)^{-p}\mathrm{d}u \tag{A.14}$$

$$= \frac{\Delta(u)^{1-p}}{(p-1)(\Delta(0) - \Delta(1))} + c \tag{A.15}$$

with $c$ the integration constant. We therefore get

$$\int_0^1 \Delta^{-p}(u)\mathrm{d}u = \frac{\Delta(1)^{1-p} - \Delta(0)^{1-p}}{(p-1)(\Delta(0) - \Delta(1))} \tag{A.16}$$

Now when $p > 1$ we have $\Delta(1)^{1-p} \gg \Delta(0)^{1-p}$ so $\int_0^1 \Delta^{-p}(u)\mathrm{d}u = O(\Delta_{\min}^{1-p}) = k$. Since $\Delta(s)$ is strictly deceasing we can also show that $\int_0^1 \Delta^{p-3}(u)|\dot{\Delta}(u)| = O(\Delta_{\min}^{p-2})$. Combining this gives

$$\epsilon(T) \leq O(T^{-1}\Delta_{\min}^{1-p}\Delta_{\min}^{p-2}) = O(T^{-1}\Delta_{\min}^{-1}) \tag{A.17}$$

So the adiabatic time is $O(\epsilon^{-1}\Delta_{\min}^{-1})$ which confirms the result we got using the approximate adiabatic condition.

## A.2 Marked Distribution Squared

In this thesis we looked at sampling from the squared distribution where the original distribution is the Gibbs distribution. There are other circumstances where the distribution doubling could be useful. We could look at sampling from the distribution with $2^N$ elements and one marked element $s$ with the other elements uniform, i.e. the distribution with elements given by

$$\pi_s = \frac{1}{\sqrt{2^N}} \qquad \pi_{\text{other}} = \frac{1 - \frac{1}{\sqrt{2^N}}}{2^n - 1}\,.$$

The marked element of the squared distribution is then given by

$$(\pi^2)_s = \frac{\frac{1}{2^N}}{\frac{1}{2^N} + \sum_{i \neq s} \frac{(1 - \frac{1}{\sqrt{2^N}})^2}{(2^N - 1)^2}} = \frac{2^N - 1}{2^N - 1 + (\sqrt{2^N} - 1)^2}\,.$$

In the $N \to \infty$ limit this goes to

$$\lim_{N\to\infty} \frac{2^N - 1}{2^N - 1 + (\sqrt{2^N} - 1)^2} = \lim_{N\to\infty} \frac{\frac{\mathrm{d}}{\mathrm{d}N} 2^N}{\frac{\mathrm{d}}{\mathrm{d}N} 2^N + \frac{2(\sqrt{2^N}-1)}{2\sqrt{2^N}} \frac{\mathrm{d}}{\mathrm{d}N} 2^N} .$$

For $N \to \infty$ this goes to $\frac{1}{2}$, so here $\lim_{N\to\infty}(\pi^2)_s = \frac{1}{2}$. So in this context the original distribution gives the marked element with exponentially decreasing probability where the squared distribution gives the marked element with constant probability.

## A.3 Expression for $P^{t_{\text{mix}}}$

In the main text we claimed that $P^k \approx Q$ with $P$ an irreducible and aperiodic column stochastic matrix and $k \geq t_{\text{mix}}$ with $t_{\text{mix}}$ the mixing time of the stochastic matrix. Here $Q$ was defined as a matrix where every column is equal to the stationary distribution of $P$.

When $P$ is irreducible and aperiodic we are guaranteed to converge to within total variation distance $\epsilon$ of the stationary distribution $\boldsymbol{\pi}$

$$d(P^k \boldsymbol{b}, \boldsymbol{\pi}) \leq \epsilon \tag{A.18}$$

for any probability vector $\boldsymbol{b}$ when $k \geq t_{\text{mix}}$. So the matrix $P^k \boldsymbol{b}$ can be approximated by the matrix that maps every probability vector to the stationary distribution. We want to find the matrix $Q$ such that

$$Q\boldsymbol{b} = \boldsymbol{\pi} . \tag{A.19}$$

Writing elements of $Q$ as $q_{i,j}$ gives

$$\begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,n} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n,1} & q_{n,2} & \cdots & q_{n,n} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{pmatrix} , \tag{A.20}$$

which can be written as

$$\begin{pmatrix} q_{1,1}b_1 + q_{1,2}b_2 + \ldots + q_{1,n}b_n \\ q_{2,1}b_1 + q_{2,2}b_2 + \ldots + q_{2,n}b_n \\ \vdots \\ q_{n,1}b_1 + q_{n,2}b_2 + \ldots + q_{n,n}b_n \end{pmatrix} = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{pmatrix} . \tag{A.21}$$

This should hold for any stochastic matrix $\boldsymbol{b}$. This can be solved by setting every column of $Q$ equal to $\boldsymbol{\pi}$ which gives

$$\begin{pmatrix} \pi_1 b_1 + \pi_1 b_2 + \ldots + \pi_1 b_n \\ \pi_2 b_1 + \pi_2 b_2 + \ldots + \pi_2 b_n \\ \vdots \\ \pi_n b_1 + \pi_n b_2 + \ldots + \pi_n b_n \end{pmatrix} = \begin{pmatrix} \pi_1 (b_1 + b_2 + \ldots + b_n) \\ \pi_2 (b_1 + b_2 + \ldots + b_n) \\ \vdots \\ \pi_n (b_1 + b_2 + \ldots + b_n) \end{pmatrix} = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{pmatrix} , \tag{A.22}$$

which is what we wanted to show.

Appendix B

# Other Examples Adiabatic Evolution Stochastic Matrices

In the main text we proved theorem 3.1 and showed numerical results that give an indication that a similar result holds for the stochastic matrix corresponding with Glauber dynamics for the uniform zero-field Curie Weiss model. In this appendix we give some other instances where numerical results give an indication of the possibility that theorem 3.1 holds for more stochastic matrices. We first consider the stochastic matrix corresponding to the Metropolis-Hastings algorithm for the uniform zero-field Curie Weiss model. In figure B.1 we see the gap along the path again seems to be minimal at the end point.

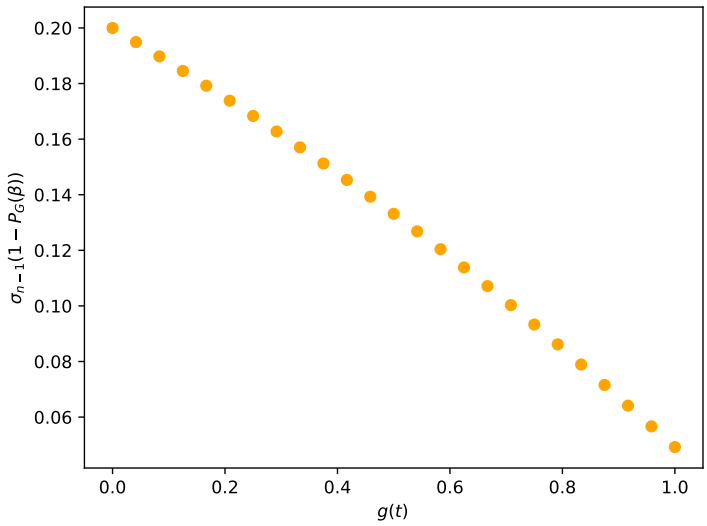


***Figure B.1:*** *The second lowest singular values of* $(1-g(t))P_G(0)+g(t)P_G(\beta)$ *as a function of* $g(t)$*. Here* $N=10$ *and* $\beta=0.7$ *for the stochastic matrix corresponding with Glauber dynamics for the uniform zero-field Curie-Weiss model.*

We can also consider the stochastic matrix corresponding with the Ising model on an instance of the 4-regular graphs as discussed in section 3.2.3. In figure B.2 we see that there the gap along the path also seems to be minimal at the right most end point.

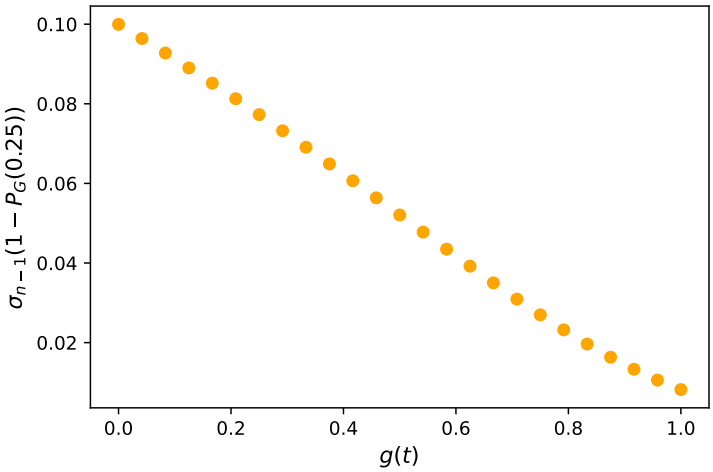


***Figure B.2:*** *The second lowest singular values of* $(1-g(t))(\mathbb{1}-P_G(0))+g(t)(\mathbb{1}-P_G(\beta))$ *as a function of* $g(t)$*. Here* $N=10$ *and* $\beta=0.7$ *for the stochastic matrix corresponding with Glauber dynamics for the uniform zero-field Ising model on an instance of a 10 spin* 4*-regular graph.*

To consider an Ising like model with different behaviour we look at Metropolis-Hastings for the Random Energy Model. In this $N$-spin model each of the $2^N$ possible states has a energy that is randomly chosen from a Gaussian distribution with the mean at zero and the variance given by $N/2$. The evolution for an instance of this model is given in figure B.3, we see again that the smallest gap is at the end of the evolution.

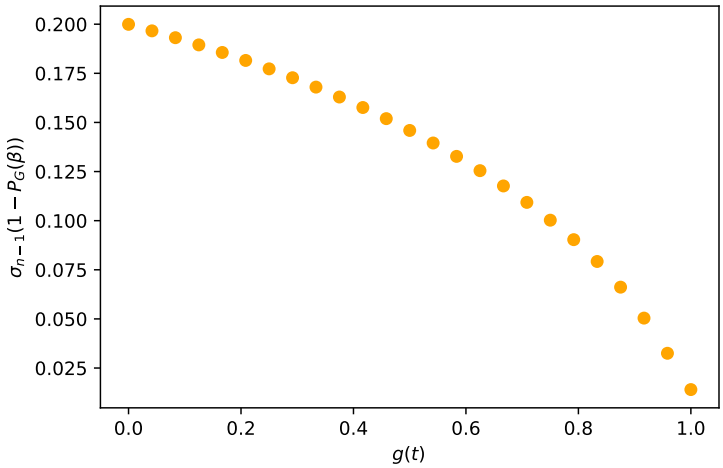


***Figure B.3:*** *The second lowest singular values of* $(1-g(t))(\mathbb{1}-P_G(0))+g(t)(\mathbb{1}-P_G(\beta))$ *as a function of* $g(t)$*. Here* $N=10$ *and* $\beta=0.25$ *for the stochastic matrix corresponding with Glauber dynamics for the Random Energy model.*

To consider matrices other than Gibbs samplers we need to define a different initial matrix for which we can easily prepare the encoding of the stationary distribution. We do this by defining the uniform stochastic matrix $U$ which has the uniform distribution as the stationary distribution. We now consider the interpolation from $U$ to a maximally dense (see figure B.4) and maximally sparse (see figure B.5) random stochastic matrix. Where we define the maximally dense random stochastic matrix as a stochastic matrix where every element is non-zero and chosen randomly from a distribution and the maximally spares random stochastic matrix as a matrix where every column has one element uniformly randomly chosen and set to one. We see that for both the gap along the path also seems to be minimal at the right most end point.

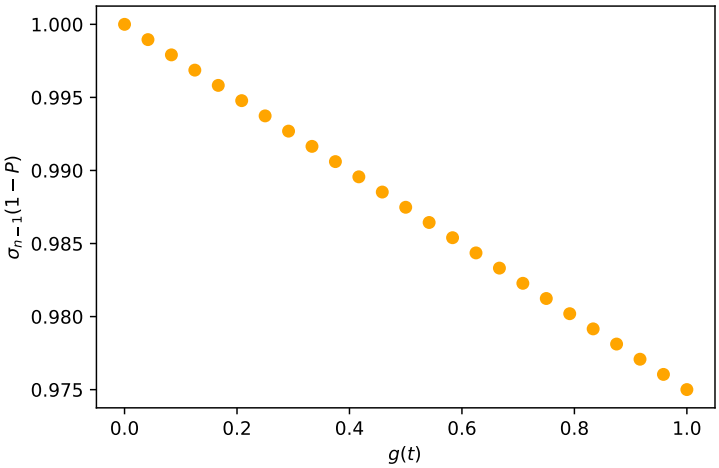


***Figure B.4:*** *The second lowest singular values of* $(1-g(t))(\mathbb{1}-U)+g(t)(\mathbb{1}-P)$ *as a function of* $g(t)$*. Here* $P$ *is a maximally dense random stochastic matrix of dimension* $n=2^{10}$*.*

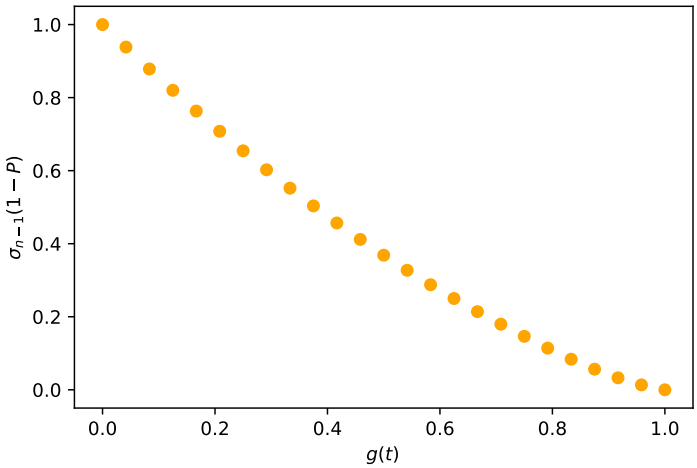


***Figure B.5:*** *The second lowest singular values of* $(1-g(t))(\mathbb{1}-U)+g(t)(\mathbb{1}-P)$ *as a function of* $g(t)$*. Here* $P$ *is a maximally sparse random stochastic matrix of dimension* $n=2^{10}$*.*