Classical and Quantum Simulation of Stoquastic Hamiltonian Systems

Maarten Stroeks
MSc. Thesis Applied Physics (Casimir Pre-PhD Track)
QuTech, Delft University of Technology

Supervisor: Prof. dr. B.M. Terhal

Dated: August 2020

Classical and Quantum Simulation of Stoquastic Hamiltonian Systems

by

Maarten Stroeks

in partial fulfilment of the requirements for the degree of

Master of Science
in Applied Physics

at Delft University of Technology

to be defended on Tuesday August 18, 2020 at 13.00h

Supervisor:	Prof. dr. B.M. Terhal
Thesis Committee:	Prof. dr. J.M. Thijssen
	Dr. M. Blaauboer

Acknowledgements

I hereby express my gratitude towards all people that have helped me with the completion of this thesis. I cannot stress enough the importance of the conversations and discussions that I have had with others about the contents of this work.

First and foremost, I thank Barbara for her guidance, patience and valuable advice. I am thankful for the freedom that you have given me throughout this project and the confidence that you have put in my ability. This past year has been incredibly enlightening, and the things that you have taught me and the discussions that we have had are an integral part of that. To be supervised by such an experienced, devoted and rigorously working scientist is very inspiring.

I also express my gratitude towards the people that have contributed to my Casimir track outside of this thesis: I sincerely thank Tom O'Brien and Miriam Blaauboer for the supervisions of my two short-rotation Casimir project, which directly and indirectly have had a positive effect on the work in this thesis. Many thanks also to Alessandro Ciani for useful discussions about and comments on my PhD proposal.

I additionally thank Jos Thijssen and Miriam Blaauboer for being part of my thesis committee.

Summary

Quantum systems are *in general* not efficiently simulatable by classical means. If one wishes to determine (some of) the eigenvalues of a Hamiltonian H that is associated with a quantum system, there are two favoured strategies: Quantum simulation and quantum Monte Carlo schemes. The former strategy uses an experimentally well-controllable quantum system that emulates the system of interest, in a digital (i.e. universal) or analog manner. The latter, albeit with a limited range of applicability, uses classical stochastic processes to efficiently obtain (often low-lying) eigenvalues of H. Quantum Monte Carlo methods may suffer from a sign problem when simulating fermionic or frustrated bosonic systems. This yields, for a given accuracy, a simulation time that scales exponentially in the system size and the inverse temperature.

Here, we consider *stoquastic* (sign-problem-free) Hamiltonians, which are conjectured to be efficiently simulatable by means of Monte Carlo methods. We prove that a Trotterized version of the imaginary-time (or euclidian-time) evolution signal $\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle$ (where $\tau \geq 0$) can be obtained up to an ϵ-additive error by classical means in time $poly(\epsilon^{-1}, n, L)$ (where L is a property of the Trotterization scheme), provided that H is stoquastic and sufficiently local. We additionally require the state $|\Phi\rangle$ to obey two conditions on which we shall elaborate later. We develop the Monte Carlo (MC) scheme that realizes the estimation of the signal $\mathcal{F}(\tau)$ and which does *not* depend on the Metropolis algorithm and its associated walker dynamics. We present several extensions of the Monte Carlo scheme for determining other quantities of interest, one of which is the partition function $Z(\beta) = Tr\big(e^{-\beta H}\big)$. We furthermore discuss the relation between the accuracy of these MC estimates and the extent to which the Hamiltonian H is frustrated. We consider, in addition, a quantum-phase-estimation (QPE) based quantum simulation scheme that determines the Trotterized version of the real-time evolution of state $|\Phi\rangle$, $\langle\Phi| e^{-itH} |\Phi\rangle$, up to an ϵ-additive error. We *inefficiently* implement the quantum simulation scheme on a classical computer. Using the spectral decomposition of H, one infers that the decay rates of the imaginary-time evolution (MC) signal and the oscillation frequencies of the real-time evolution (QPE) signal correspond to the eigenvalues of the stoquastic Hamiltonian H. We employ the Matrix Pencil Method (particularly noise-resilient) to extract (some of) the eigenvalues of H from the MC and QPE signals and investigate how its performance compares for these two signals. We use the transverse-field Ising chain as an archetypal stoquastic system to numerically study several aspects of the aforementioned MC and QPE methods. The Trotter error that infects the MC and QPE signals is numerically shown to be approximately equal in magnitude for both settings. We have established, analytically as well as numerically, that the (sampling) noise that is imposed on the QPE signal is *always* at least that of the MC signal. The (sampling) noise on the MC signal is shown, in addition, to vanish as $\tau \to 0$. Although the Matrix Pencil Method performs equally well in the noiseless-signal setting in extracting eigenvalues from the MC or QPE signals, we show numerically that in order to distinguish the ground-state and first-excited-state of the transverse-field Ising chain in the noisy-signal setting, the noise magnitude on the MC signal needs to be smaller than that for the QPE signal. Parts of the spectrum of the transverse-field Ising chain in its intermediately strong-coupled regime are reconstructed using the MC and QPE schemes, and we comment on the ability of these schemes in doing so.

Contents

I Introduction **7**

I.1 Local and Stoquastic Hamiltonians 9

I.2 Frustration and its implications 11

I.3 Preliminaries 12

I.4 Trotterization 16

I.4.1 Trotterization for Arbitrary-Time Evolution 16

I.4.2 Trotterization for Real-Time Evolution 18

I.4.3 Trotterizing Hamiltonian Dynamics (in Real- and Imaginary-Time) 20

I.4.4 Higher-Order Trotter Decompositions 21

II Quantum Simulation **24**

II.1 Digital Quantum Simulation 24

II.1.1 Initial-State Preparation & Hamiltonian Evolution 25

II.1.2 Measurement 26

II.2 Analog Quantum Simulation 26

II.3 Quantum Phase Estimation 26

III Signal Analysis **31**

III.1 Matrix-Pencil Method 32

IV Monte Carlo Scheme for Local and Stoquastic Hamiltonians **37**

IV.1 Efficiently determining single matrix elements of the Gibbs density matrix 37

IV.2 Considering superpositions of basis states 41

IV.3 Extension to Non-Hermitian Propagation Operators 43

IV.4 Short-Time Simulations 44

IV.5 Determining the Partition Function 45

IV.6 Overview 47

IV.7 Use of the Monte Carlo Scheme in a Quantum Computing Setting 48

IV.7.1 (Proposing) A Hybrid Algorithm for Estimating Eigenvalues of Stoquastic Hamiltonians 49

V Ising Chain in a Transverse Field **50**

V.1 Choice of the State $|\Phi\rangle$ 52

V.2 Monte Carlo and Quantum Phase Estimation Signals 54

V.3 Temporal Dependence of the Noise on the MC and QPE Signals 57
V.4 Estimates of the Spectrum of the Transverse-Field Ising Chain 58
V.5 Convergence of Eigenvalue Estimates as a Function of Sample Size $|\Sigma|$ 63

VI Conclusions and Outlook **64**

A Appendix: The Sign Problem **68**

B Appendix: Code **71**

I Introduction

The state of a quantum system evolves in time according to the time-dependent Schrödinger equation: $i\hbar\frac{\partial}{\partial t}|\Phi\rangle = H|\Phi\rangle$ (where $|\Phi\rangle$ represents the state of the system and H its Hamiltonian). If H were to be time-independent, then the system state at time t, $|\Phi(t)\rangle$, is given by $e^{-iHt/\hbar}|\Phi(0)\rangle$ – i.e. the initial state $|\Phi(0)\rangle$ is propagated by the propagation operator $e^{-iHt/\hbar}$. Suppose one wishes to determine (some of) the eigenvalues of H. Since direct diagonalization of H is in general computationally intractable by classical means (as will be argued below), other methods for finding its eigenvalues must be found. H is Hermitian, and therefore the associated propagation operator can be decomposed as follows in terms of its orthonormal set of eigenvectors $\{|\psi_j\rangle\}$ and real-valued eigenvalues $\{E_j\}$: $\sum_j e^{-iE_jt/\hbar}|\psi_j\rangle\langle\psi_j|$. If one is thus able to accurately and efficiently track $|\Phi(t)\rangle$ as a function of time, the eigenvalues of H can be estimated from the frequencies with which the elements of $|\Phi(t)\rangle$ oscillate in time t. Apart from intrinsic interest in the eigenvalues of H, having access to the eigenvalues allows one to calculate e.g. the partition function $Z = Tr\big(e^{-\beta H}\big)$, from which (other) physically relevant quantities can be obtained.

We consider a Hamiltonian that consists of n variables (e.g. n sites), where the 'state' of each variable lies in a σ-dimensional vector space. Then the state $|\Phi\rangle$ of the whole system lives in a vector space of dimensionality σ^n and H is a $\sigma^n \times \sigma^n$ matrix. Suppose one wants to simulate the abovementioned dynamics of $|\Phi(t)\rangle$ on a classical computer: The state $|\Phi\rangle$ at each instance in time would take up an amount of storage that is exponential in the number of variables n and the exponentiation of H to explicitly obtain $e^{-iHt/\hbar}$ (to determine the time dynamics of $|\Phi\rangle$ exactly) requires exponentially large computational effort since H consists of an exponential number of elements. In general, therefore, the simulation of quantum system dynamics on a classical computer is inefficient ([1],[2]).

Exceptions to this rule are quantum Monte Carlo methods: With these techniques, one is able to estimate e.g. the partition function or thermal averages of operators (and consequently low-lying eigenvalues) of *subclasses* of Hamiltonians in $poly(n)$ time by classical means. These methods often make use of the correspondence between the time-dependent Schödinger equation (when transformed from real to imaginary time: $it \to \tau$) and the diffusion equation: Expressing the Schrödinger equation in terms of spatial coordinates and imaginary time (or euclidian time), one obtains $\frac{\partial}{\partial\tau}\Phi(\boldsymbol{x},\tau) = \Big(\frac{\hbar^2}{2m}\nabla^2 - V(\boldsymbol{x})\Big)\Phi(\boldsymbol{x},\tau)$, which reduces to the diffusion equation when $V(\boldsymbol{x}) = 0$ (a non-zero V is then compensated for at a later stage) ([3]). The Monte Carlo methods that explicitly make use of this correspondence are called Diffusion Monte Carlo methods (which is a particular group of projector Monte Carlo schemes) and are often used in quantum chemistry settings. The propagation operator corresponding to the time-dependent Schrödinger equation in imaginary time is $e^{-\tau H/\hbar}$. The dynamics of the state $|\Phi(\tau)\rangle$ is now not associated with a superposition of oscillating signals (the frequencies of which would be proportional to the eigenvalues of H), but with a superposition of decaying signals (where the decay rates are proportional to the eigenvalues of H). Since the signals corresponding to higher-lying eigenvalues are exponentially suppressed as a function of τ comparing to those associated with low-lying eigenvalues (which are the ones of primary interest in many settings), the latter can often be estimated relatively efficiently by classical means.

The efficiency of these quantum Monte Carlo methods when used to simulate fermionic or frustrated bosonic systems is generally limited by the *sign problem*: This causes the run-time of a Monte Carlo algorithm for calculating the expectation value of an observable (to a given precision) to scale exponentially in the system size n and inverse temperature β ([4]). This means that the quantum Monte Carlo method does not succeed in determining properties of the system in $poly(n)$ time by classical means.

To avoid the need for storage capacities and run-times that are exponentially large on classical computers, one can thus choose either of two methods: Simulate the system classically if its Hamiltonian H is fit to be efficiently simulated by a quantum Monte Carlo algorithm, or simulate the system by means of *quantum simulation*. The latter method would use an experimentally well-controllable quantum system of which the dynamics can be directly mapped to the dynamics of the state of the quantum system to be simulated, in either a digital (universal) or analog manner. By thus tracking the state of the simulator as a function of time, the time evolution of the state of the target system can be determined. The limitations of *quantum simulation* are essentially threefold: The simulator needs to evolve according to the propagation operator for a sufficiently large time interval to be able to determine e.g. the eigenvalues of the Hamiltonian, but the system will suffer from relaxation and decoherence due to interaction with its environment (which is undesired if the system to be simulated is isolated). In addition, the Hamiltonian to be simulated needs to be sufficiently local – i.e. $H = \sum_{i=1}^{N} H_i$, where $N \leq poly(n)$ and each H_i acts non-trivially on

at most k variables of the system, such that $k \ll n$ and k is possibly even independent of n. This avoids the need for exponentiation of the $\sigma^n \times \sigma^n$ Hamiltonian and instead requires the exponentiation of many $\sigma^k \times \sigma^k$ interaction terms. This technique is known as *Trotterization* (on which we shall elaborate in some detail later on in this introductory chapter). We note that most Hamiltonians encountered in physics are indeed local Hamiltonians. A third prominent limitation is the choice of the initial state $|\Phi\rangle$: The only eigenvalues of H that can be accurately estimated have an associated eigenstate with a *sufficiently* large overlap with $|\Phi\rangle$. $|\Phi\rangle$ should thus be chosen with care, and this choice does require prior knowledge about properties of Hamiltonian H. In addition, one should be able to prepare the n-qubit state $|\Phi\rangle$ one a qubit register by means of a circuit of $poly(n)$-depth.

A class of Hamiltonians that is generally fit for quantum Monte Carlo simulations is that of *stoquastic* Hamiltonians: A (real-valued) Hamiltonian is stoquastic (in a particular basis $\mathcal{B}$) if its off-diagonal elements are non-positive: $\langle x| H |y\rangle \leq 0$, for $x \neq y$ ($|x\rangle$ and $|y\rangle$ being elements of $\mathcal{B}$). As a consequence, its associated Gibbs density matrix $e^{-\tau H}$ is an element-wise non-negative matrix (for $\tau \in \mathbb{R}_+$) and this property makes it particularly suitable for Monte Carlo sampling since the simulations of these Hamiltonians cannot suffer from a sign problem ([5]). As is apparent from its definition, stoquasticity is a basis-dependent property – i.e. if a Hamiltonian is not stoquastic in a certain basis, one can try to find a basis transformation such that the Hamiltonian becomes (approximately) stoquastic in the new basis and run a Monte Carlo algorithm in this new basis. In [6], a systematic method is presented for obtaining a basis (if it exists) in which a Hamiltonian is (approximately) stoquastic and thereby for easing the sign problem.

In this report, we develop a quantum Monte Carlo method that approximately obtains (a subset of) the eigenvalues of stoquastic Hamiltonians by tracking the system dynamics in imaginary time, i.e. evolution according to the non-negative propagation operator $e^{-\tau H}$ (where $\tau \geq 0$). Specifically, we prove that the quantity $\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle$ (where $|\Phi\rangle$ is a state of generally $exp(n)$ complex-valued elements) can be obtained, with a Trotter error that is generally efficiently suppressible and an additional ϵ-additive error, in time $poly(\epsilon^{-1}, n, L)$ when $H = \sum_{i=1}^{N} H_i$ (where each H_i is stoquastic and acts non-trivially on at most $log(poly(n))$, but typically $\mathcal{O}(1)$, variables of the system). We additionally require that $|\Phi\rangle = \sum_x \Phi(x) |x\rangle$ (where $\{|x\rangle\}$ is the basis in which the MC scheme is ran) is such that the quantity $\frac{\Phi(y)}{\Phi(x)^*}$ can be efficiently obtained (for two given basis states $|x\rangle$ and $|y\rangle$) and a sample from $|\Phi(x)|^2$ can be efficiently drawn. L (generally proportional to N) is a quantity that depends on the Trotterization scheme and corresponds to the number of local imaginary-time propagation operators in the Trotterized version of $e^{-\tau H}$. Estimating $\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle$ can be efficiently done for $\epsilon \geq \frac{1}{poly(n)}$ and $L \leq poly(n)$. By determining $\mathcal{F}(\tau = \Delta\tau k)$ at $k \in \{0, 1, ..., K \leq poly(n)\}$, we effectively obtain a noisy and Trotterized version of the signal $g(k) = \sum_j e^{-\Delta\tau E_j k} |\langle\Phi|\psi_j\rangle|^2$. Obtaining (a subset of) the decay rates of this signal g as a function of k – which equal the eigenvalues of H – is done by means of the Matrix Pencil Method (which is particularly noise-resilient). The Monte Carlo samples that are required to determine $\mathcal{F}(\tau)$ are obtained stochastically using a method that is *not* based on the Metropolis algorithm (and its associated walker dynamics) and therefore circumvents potential difficulties regarding walker distribution convergence that might be encountered in e.g. diffusion MC simulations. We present several extensions of our MC scheme beyond the estimation of imaginary-time state evolution, such as the estimation of the partition function $Z = Tr\left(e^{-\beta H}\right)$.

We consider a quantum-phase-estimation (QPE) based quantum simulation scheme (which is *inefficiently* implemented on a classical computer) that produces the signal $\langle\Phi| e^{-itH} |\Phi\rangle$. We effectively obtain a noisy and Trotterized version of the signal $\sum_j e^{-i\Delta t E_j k} |\langle\Phi|\psi_j\rangle|^2$ at $k \in \{0, 1, ..., K \leq poly(n)\}$. We then make use of the Matrix Pencil Method to obtain oscillation frequencies of this signal (which equal the eigenvalues of H). We study in particular whether this digital quantum simulation algorithm based on QPE outperforms the aforementioned Monte Carlo algorithm for finding eigenvalues of stoquastic Hamiltonians – i.e. Hamiltonians that do not suffer from a sign problem. Since quantum Monte Carlo schemes could in principle obtain useful properties in $poly(n)$ time in this case, one might wonder whether a QPE-based algorithm can provide additional benefits over the Monte Carlo scheme. In [7], the complexity of classically applying the QPE scheme to a stoquastic Hamiltonian to obtain its ground state is investigated by means of a projector MC scheme (that makes use of many MC walkers).

To investigate several aspects of the aforementioned quantum Monte Carlo method (and to make comparisons to the QPE-based algorithm), we numerically study the one-dimensional Ising chain in a transverse field in a *proof-of-principle* setting. This system is associated with the following Hamiltonian:

$$H = \sum_{i=1}^{N} H_i = -J\left(\sum_i \sigma_i^z \sigma_{i+1}^z + g \sum_i \sigma_i^x\right), \tag{1}$$

which is stoquastic in the standard basis for $g > 0$. The number of bond Hamiltonians (H_i) is equal to the number of variables of the system in this case ($N = n$) and they are all 2-local. This system, which has been extensively studied (in e.g. [8]), is exactly solvable and exhibits a second-order phase transition at $T = 0$ as a function of the dimensionless field variable at $g = g_c$. For $g \gg g_c$ (the paramagnetic phase), the ground state is given by $|+\rangle^{\otimes n}$ and as g crosses g_c from above, the system moves to the ferromagnetic phase and the Z_2 symmetry is broken. For $g \ll g_c$, the ground state is degenerate and given by $|0\rangle^{\otimes n}$ or $|1\rangle^{\otimes n}$. We numerically study, amongst several other aspects, the magnitude of the noise that is imposed on the MC and QPE signals, and the maximum allowed noise magnitude to recover – by means of the Matrix Pencil Method – the ground-state en first-excited-state eigenvalues of the transverse-field Ising chain (in the intermediately strong-coupled regime) from the MC and QPE signals.

In the remainder of this chapter, we will introduce concepts that are of central importance to the rest of this report such as *stoquastic Hamiltonians*, *frustration* and *Trotterization*. In Chapter II, we will elaborate further on the concept of *quantum simulation*, and specifically the use of quantum phase estimation for digital quantum simulation. Chapter III will be used to introduce the *Matrix Pencil Method* as a means to obtain eigenvalues of a Hamiltonian given the real-time or imaginary-time evolution of a state that evolves according to this Hamiltonian. In Chapter IV, an elaborate description is given of the Monte Carlo scheme that is used to efficiently track the imaginary-time dynamics of a state evolving according to a stoquastic Hamiltonian (and several of its extensions). In Chapter V, we discuss the *transverse-field Ising chain* and in particular compare the use of quantum phase estimation and our Monte Carlo scheme for determining its eigenvalues. Chapter VI is used to present the conclusions of this report and points of further study.

I.1 Local and Stoquastic Hamiltonians

A (real-valued) Hamiltonian H is stoquastic in a basis $\mathcal{B}$ if all its off-diagonal elements are non-positive: $\langle x| H |y\rangle \leq 0$, for $x \neq y$ (and states $|x\rangle$, $|y\rangle$ being elements of basis $\mathcal{B}$). Consequently, its associated Gibbs density matrix $e^{-\tau H}$ is element-wise non-negative (for $\tau \in \mathbb{R}_+$). This can be shown as follows:

$$G \equiv \tau \max_x \langle x| H |x\rangle \, I - \tau H. \tag{2}$$

G is by construction an element-wise non-negative matrix (due to stoquasticity of H and $\tau \geq 0$). Exponentiating on both sides of the previous equation yields:

$$e^{-\tau H} = e^{G} e^{-\tau \max_x \langle x|H|x\rangle I} = e^{-\tau \max_x \langle x|H|x\rangle} \sum_{j=0}^{\infty} \frac{G^j}{j!}, \tag{3}$$

and since the product of element-wise non-negative matrices equals an element-wise non-negative matrix itself, $e^{-\tau H}$ is an element-wise non-negative matrix.

This non-negativity property results in the absence of the sign problem when using Monte Carlo procedures to simulate the system ([5]), and therefore renders its classical simulation more likely to be computationally tractable. In Appendix A, we illustrate the appearance of the sign problem for a non-stoquastic Hamiltonian by means of an example. In this report, we are particularly interested in *local* and *stoquastic* Hamiltonians:

Definition I.1. ***Local Hamiltonians.*** *A Hamiltonian H associated with an n-variable system is local if it admits a decomposition into a set of Hermitian operators $\{H_i\}$ – i.e. $\sum_i^N H_i$ – such that each H_i acts non-trivially on at most $log(poly(n))$ (but typically $O(1)$) variables of the system.*

We denote the number of variables on which H_i acts non-trivially (i.e. its *locality*) by k. We note that although this locality k need not be the same for all i, we shall call a Hamiltonian H k-local if k is the largest locality of all H_i's in the decomposition of H. The number of terms N in the decomposition of H might, in principle, be superpolynomial in the system size n. In this work, however, we focus in particular on Hamiltonians for which $N \leq poly(n)$.

Definition I.2. ***Global and Piece-wise Stoquasticity.*** *A real-valued Hamiltonian H is globally stoquastic in a basis $\mathcal{B}$ if: $\langle x| H |y\rangle \leq 0$ for $x \neq y$ and where $|x\rangle$ and $|y\rangle$ are elements of $\mathcal{B}$. If H is a k-local Hamiltonian, then H is m-piece-wise stoquastic (where $m \geq k$) in basis $\mathcal{B}$ if it admits a decomposition into a set of m-local real-valued and Hermitian operators $\{\tilde{H}_i\}$ such that ($\forall i$): $\langle x| \tilde{H}_i |y\rangle \leq 0$ for $x \neq y$ and where $|x\rangle$ and $|y\rangle$ are elements of $\mathcal{B}$.*

We note that a Hamiltonian being globally stoquastic is equivalent to it being $(m = n)$-piece-wise stoquastic. Furthermore, any piece-wise stoquastic Hamiltonian is globally stoquastic as well, and a Hamiltonian being m-piece-wise stoquastic implies it being $(m + 1)$-piece-wise stoquastic as well. In [9], the distinction between global and piece-wise stoquasticity of a local Hamiltonian is discussed in more detail. In particular, the complexity of deciding whether a given Hamiltonian belongs to a particular stoquasticity class is investigated. In the remainder of this work, we shall call a k-local and m-piece-wise stoquastic Hamiltonian simply a local and piece-wise stoquastic Hamiltonian if $m \leq log(poly(n))$. To demonstrate the ubiquity of stoquastic Hamiltonians, we present some examples below.

Spin Hamiltonians: We consider a Heisenberg chain in a magnetic field, which is represented by the following Hamiltonian:

$$H = \sum_i H_i = \sum_i \Big(J_x \, \sigma_i^x \sigma_{i+1}^x + J_y \, \sigma_i^y \sigma_{i+1}^y + J_z \, \sigma_i^z \sigma_{i+1}^z + B_x \, \sigma_i^x + B_y \, \sigma_i^y + B_z \, \sigma_i^z \Big), \tag{4}$$

where J_α, $B_\alpha \in \mathbb{R}$ (for $\alpha \in \{x, y, z\}$). In the standard (local σ^z) basis, the bond Hamiltonian H_i is given by:

$$H_i = \begin{pmatrix} J_z + B_z & 0 & B_x - iB_y & J_x - J_y \\ 0 & -J_z + B_z & J_x + J_y & B_x - iB_y \\ B_x + iB_y & J_x + J_y & -J_z - B_z & 0 \\ J_x - J_y & B_x + iB_y & 0 & J_z - B_z \end{pmatrix}. \tag{5}$$

The Hamiltonian H in eq. 4 is thus piece-wise stoquastic in the standard basis in the following parameter regime:

$$J_x \leq |J_y|, \; \forall J_z, \; B_x \leq 0, \; B_y = 0, \; \forall B_z. \tag{6}$$

For other parameter regimes, one might find a basis transformation that transforms the Hamiltonian to a piece-wise stoquastic form when expressed in the new basis.

Real-Space Hamiltonians: We consider a system of n particles in d spatial dimensions, the state of which is described by the vector $\boldsymbol{R} = (\boldsymbol{r_1}, \boldsymbol{r_2}, ..., \boldsymbol{r_n})$ (which lies in a (d^n)-dimensional vector space). The particles live in a potential $V(\boldsymbol{R})$, which itself depends on the state vector $\boldsymbol{R}$. The Hamiltonian associated with this system is the following:

$$H = -\frac{1}{2m} \nabla^2_{\boldsymbol{R}} + V(\boldsymbol{R}), \tag{7}$$

where $\nabla^2_{\boldsymbol{R}} = \nabla^2_{\boldsymbol{r}_1} + \nabla^2_{\boldsymbol{r}_2} + ... + \nabla^2_{\boldsymbol{r}_n}$. We consider the case of a general number of spatial dimensions d, such that the vector $\boldsymbol{R} = (\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_n)$ (where $\boldsymbol{r}_i = (x_i^1, x_i^2, ..., x_i^d)$) describes the state of the system. Next, we consider a discretized d-dimensional space in which the n particles can reside in small spatial patches of volume a^d. This allows for the expression of H in terms of the eigenbasis of the potential energy operator. This eigenbasis is given by the set of states of all available spatial particle configurations $\{|\boldsymbol{R}\rangle\}$, where $\boldsymbol{R} = (\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_n)$ and each $\boldsymbol{r}_i$ equals $q_i^{(1)} a \, \hat{\boldsymbol{x}}^1 + q_i^{(2)} a \, \hat{\boldsymbol{x}}^2 + ... + q_i^{(d)} a \, \hat{\boldsymbol{x}}^d$ (where $\{q_i^{(1)}, q_i^{(2)}, ..., q_i^{(d)}\}$ are integers and $\{\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, ..., \hat{\boldsymbol{x}}^d\}$ represent the Cartesian unit vectors). Each individual kinetic energy operator $-\frac{1}{2m} \nabla^2_{\boldsymbol{r}_i}$ can be rewritten as follows in terms of Cartesian coordinates:

$$-\frac{1}{2m} \nabla^2_{\boldsymbol{r}_i} = \sum_{\hat{\boldsymbol{q}} \, \in \, \{\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, ..., \hat{\boldsymbol{x}}^d\}} - \frac{|\boldsymbol{r}_i\rangle\langle \boldsymbol{r}_i + a\hat{\boldsymbol{q}}| + |\boldsymbol{r}_i\rangle\langle \boldsymbol{r}_i - a\hat{\boldsymbol{q}}| - 2\,|\boldsymbol{r}_i\rangle\langle \boldsymbol{r}_i|}{2m \, a^2}, \tag{8}$$

where each $-\frac{1}{2m} \nabla^2_{\boldsymbol{r}_i}$ operator acts on the i^{th} particle as a kinetic energy operator, and effectively as identity on the other $n - 1$ particles. The complete Hamiltonian H can thus be rewritten in the $\{|\boldsymbol{R}\rangle\}$ basis as follows (where we identify diagonal elements and off-diagonal elements):

$$H = \underbrace{-\frac{1}{2m \, a^2} \sum_{i=1}^{n} \sum_{\hat{\boldsymbol{q}} \, \in \, \{\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, ..., \hat{\boldsymbol{x}}^d\}} \Big(|\boldsymbol{r}_i\rangle\langle \boldsymbol{r}_i + a\hat{\boldsymbol{q}}| + |\boldsymbol{r}_i\rangle\langle \boldsymbol{r}_i - a\hat{\boldsymbol{q}}| \Big)}_{\text{off-diagonal terms}} + \underbrace{\frac{3}{m \, a^2} \sum_{i=1}^{n} |\boldsymbol{r}_i\rangle\langle \boldsymbol{r}_i| + V(\boldsymbol{R})}_{\text{diagonal terms}}, \tag{9}$$

where we note that $V(\boldsymbol{R})$ is obviously diagonal when expressed in its own eigenbasis. All off-diagonal elements of H are non-positive in the $\{|\boldsymbol{R}\rangle\}$ basis and we thus conclude that H is stoquastic in this basis. This would suggest that simulation of H by means of a Monte Carlo scheme might be computationally tractable. However, the simulation of a fermionic (e.g. electronic structure) systems – which is a particular case of the scenario described thus far – is known to be generally a computationally intractable task by classical means, in particular for $d > 1$.

The above paradox can be resolved as follows: The aforementioned Hamiltonian and its state space do not necessarily correspond to that of a fermionic system. In the formulation above, the spatial particle configurations are not restricted to lie in a particular subspace, while the spatial configurations associated to fermionic systems have to obey the symmetry postulate. One has to restrict the state space such that only those states are available to the fermionic system that are anti-symmetric with respect to pair permutations. This can, for instance, be realized by moving from the first-quantization setting as shown above to a second-quantization setting by introducing fermionic creation and annihilation operators $\{c^\dagger_{k,\sigma}, c_{k,\sigma}\}$ (where $k, \sigma \in \{1\uparrow, 1\downarrow, 2\uparrow, ..., \mathcal{V}\uparrow, \mathcal{V}\downarrow\}$ labels the fermionic levels and we have restricted the discussion to spin-$\frac{1}{2}$ particles), such as in the Fermi-Hubbard model:

$$H = -t \sum_{\langle k,l\rangle,\sigma} \left(c^\dagger_{k,\sigma} c_{l,\sigma} + c^\dagger_{l,\sigma} c_{k,\sigma}\right) + U \sum_k n_{k,\uparrow} n_{k,\downarrow} - \mu \sum_k \left(n_{k,\uparrow} + n_{k,\downarrow}\right), \tag{10}$$

where the sum over $\langle k, l\rangle$ is a sum over nearest-neighbour sites, $n_{k,\sigma}$ equals $c^\dagger_{k,\sigma} c_{k,\sigma}$, t is the hopping parameter, U the on-site interaction strength and μ the chemical potential. The label k labels a lattice site, just as $\{q^{(1)}, q^{(2)}, ..., q^{(d)}$ labels the available discretized positions of the particles in the discussion above. The anti-symmetry of the states of the system are now ensured by the anti-commutation relations of the fermionic creation and annihilation operators. One can examine the stoquasticity of the Fermi-Hubbard Hamiltonian by transforming the fermionic creation and annihilation operators to spin matrices by means of the Jordan-Wigner transformation. The Fermi-Hubbard can then be shown to be generally non-stoquastic (in the standard basis) for positive hopping parameters ($t > 0$).

One could propose to set up a Monte Carlo algorithm in the $\{|\boldsymbol{R}\rangle\}$ basis for the stoquastic Hamiltonian in eq. 9 and post-select on states that are anti-symmetric w.r.t pair permutations after a simulation run to effectively simulate a fermionic system. However, the subspace of the complete Hilbert space associated with eq. 9 that is anti-symmetric w.r.t pair permutations is arguably exponentially small. Therefore, post-selection on anti-symmetric states will generally *not* be a computationally tractable manner of simulating a fermionic system.

I.2 Frustration and its implications

As will become apparent in a later stage of this report, one of the factors that severely limits the ability to estimate eigenvalues or the partition function of a stoquastic Hamiltonian (using the Monte Carlo method developed in this report) in a computationally tractable way is *frustration* of Hamiltonian H. To gain insight into when a Hamiltonian is frustrated and if so, to what extent it limits the functionality of the Monte Carlo method, we introduce a number of useful concepts in this section.

We denote the set of n-variable local Hamiltonians H (defined in Definition I.1) by LH. The subset $C \subset LH$ corresponds to all H for which $\{H_i\}_{i=1}^N$ is a commuting set. The subset $S \subset LH$ is the set associated with the Hamiltonians that are stoquastic (piece-wise and/or globally). $D\ (\subset S \subset LH)$ is the subset of classical (diagonal) Hamiltonians. The subset of frustrated Hamiltonians is denoted by $F \subset LH$, and its complement is given by the subset of frustration-free Hamiltonians:

Definition I.3. ***Frustration-Freeness.*** *Suppose the local Hamiltonian $H = \sum_i^N H_i$ represents an n-variable system (i.e. $H \in LH$). Furthermore, suppose $H|\psi_0\rangle = E_0|\psi_0\rangle$ (where E_0 is the ground state energy of the n-variable system) and $H_i|\psi_0^i\rangle = E_0^i|\psi_0^i\rangle$ (where E_0^i is the ground state energy of the subsystem labeled by i). Then Hamiltonian H is frustration-free if ($\forall i$): $H_i|\psi_0\rangle = E_0^i|\psi_0\rangle$ – i.e. the ground state of the n-variable system $|\psi_0\rangle$ simultaneously minimizes the energy of each individual H_i.*

We note that if Hamiltonian $H = \sum_i^N H_i\ (\in LH)$ consists of a commuting set of terms $\{H_i\}$, this does not necessarily imply frustration-freeness: Although the ground state is necessarily an eigenstate of

each H_i, it might not be the lowest energy eigenstate of each individual H_i. Therefore, frustration might still occur. Stoquastic, or even classical (diagonal), Hamiltonians in LH can be frustrated as well. This is schematically depicted in Figure 1.

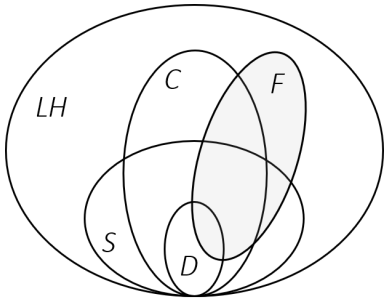


Figure 1: The set of frustrated local Hamiltonians (F) is not restricted to the class for which $\{H_i\}$ are non-commuting. Frustration of H can occur if $\{H_i\}$ is a commuting set as well (C). Additionally, frustration of Hamiltonian H can occur if it is stoquastic (S), and even if it is classical (D).

It is clear from Definition I.3 that if one were to systematically set the smallest eigenvalue of each H_i to zero, then the smallest eigenvalue of $H = \sum_i^N H_i$ is larger then zero if H is frustrated: Frustration generally causes the eigenspectrum of H to be shifted upwards by an amount that we shall denote by Λ.

We develop in this report a quantum Monte Carlo method that efficiently obtains the signal $\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle$ (where $\tau \geq 0$) and the partition function $Z(\beta) = Tr\big(e^{-\beta H}\big)$ (where $\beta \geq 0$) up to an ϵ-additive error (provided that $H \subset LH$ and $H \subset S$). This Monte Carlo scheme requires setting the smallest eigenvalues of each H_i to zero (and thereby setting all eigenvalues to be non-negative). For the scheme to indeed run efficiently, the desired accuracy ϵ should be at least of magnitude $1/poly(n)$. Therefore, since the error is additive, accurate estimates of the aforementioned quantities can only be obtained if they too are at least of magnitude $1/poly(n)$. This directly relates to the degree to which H is allowed to be frustrated. The lower bounds for $\mathcal{F}(\tau)$ and $Z(\beta)$ in terms of the frustration Λ can be obtained as follows (using the spectral decomposition of H in terms of its eigenvalues $\{E_j\}$ and eigenstates $\{|\psi_j\rangle\}$):

$$\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle = \sum_j e^{-\tau E_j} |\langle\Phi|\psi_j\rangle|^2 \geq e^{-\tau\Lambda} |\langle\Phi|\psi_0\rangle|^2, \tag{11a}$$

$$Z(\beta) = Tr\big(e^{-\beta H}\big) = \sum_j e^{-\beta E_j} \geq e^{-\beta\Lambda}. \tag{11b}$$

We conclude that $\mathcal{F}(\tau)$ and $Z(\beta)$ are of at least $1/poly(n)$ size if $\tau\Lambda \leq log\big(poly(n)\big)$ and $|\langle\Phi|\psi_0\rangle|^2 \geq 1/poly(n)$, and $\beta\Lambda \leq log\big(poly(n)\big)$, respectively. In other words, given a particular scaling of Λ as a function of the system size n, an accurate estimate can be obtained of $\mathcal{F}(\tau)$ for a limited τ interval and of $Z(\beta)$ only above certain system temperatures.

In a quantum simulation setting (specifically a quantum-phase-estimation setting), the signal of interest takes the form $\mathcal{F}(it) = \langle\Phi| e^{-itH} |\Phi\rangle$. A non-zero Λ does not impose the difficulty of the signal strength being smaller than $1/poly(n)$ in that case. It does, however, offset the frequencies (by an amount Λ) with which the signal oscillates in time t. Since higher frequencies are generally more difficult to obtain by means of sampling of the signal than lower frequencies, frustration might impose difficulties in obtaining the eigenvalues of H from the signal $\mathcal{F}(it)$.

I.3 Preliminaries

In this section we introduce several definitions that have not been introduced up to this point in the report. In addition, we state several lemma's which will be useful throughout its remainder.

Definition I.4. ***Induced Matrix Norm.*** *Let $A \in \mathbb{R}^{m\times n}$. An induced matrix norm is a matrix norm $||\cdot||_{a,b} : \mathbb{R}^{m\times n} \to \mathbb{R}$ defined as:*

$$||A||_{a,b} = \max_{\boldsymbol{x}} ||A\boldsymbol{x}||_a, \text{ such that } ||\boldsymbol{x}||_b \leq 1, \tag{12}$$

where $||\cdot||_a$ is a vector norm on $\mathbb{R}^m$ and $||\cdot||_b$ is a vector norm on $\mathbb{R}^n$. When the same vector norm is used on both $\mathbb{R}^m$ and $\mathbb{R}^n$, we write $||A||_c = \max_{\boldsymbol{x}} ||A\boldsymbol{x}||_c$, such that $||\boldsymbol{x}||_c \leq 1$.

Properties of the *Induced Matrix Norm*:

1. $||A|| \geq 0$, where the inequality holds iff A is the zero matrix (*non-negativity*).
2. $||\alpha A|| = |\alpha| \, ||A||$, for $\alpha \in \mathbb{R}$ (*homogeneity*).
3. $||A + B|| \leq ||A|| + ||B||$ (*triangle inequality*).
4. $||A\boldsymbol{x}|| \leq ||A|| \, ||\boldsymbol{x}||$.

   Proof. Suppose $||A\boldsymbol{x}|| > ||A|| \, ||\boldsymbol{x}||$, then $||A\frac{\boldsymbol{x}}{||\boldsymbol{x}||}|| > ||A||$ (we note that $\frac{\boldsymbol{x}}{||\boldsymbol{x}||}$ is a vector of norm 1). This contradicts the definition of the induced matrix norm $||A||$ and thus the supposition is incorrect. Therefore, $||A\boldsymbol{x}|| \leq ||A|| \, ||\boldsymbol{x}||$. ■
5. $||AB|| \leq ||A|| \, ||B||$ (*sub-multiplicativity*).

   Proof. $||AB|| = \max_{||x||\leq 1} ||AB\boldsymbol{x}|| \leq \max_{||x||\leq 1} ||A|| \, ||B\boldsymbol{x}|| = ||A|| \, \max_{||x||\leq 1} ||B\boldsymbol{x}|| = ||A|| \, ||B||$. ■

Throughout this report, whenever we speak of a matrix norm, we refer to the $p = 2$ - *Induced Matrix Norm* defined as follows:

$$||A|| = \sqrt{\lambda_{max}\big(A^\dagger A\big)}, \tag{13}$$

where λ_{max} denotes the largest eigenvalue.

Lemma I.1. ***Markov's Inequality.*** *Suppose X is a non-negative random variable and $\mu = \mathbb{E}(X)$ is its expectation value, then for $a > 0$:*

$$Pr\big(X \geq a\big) \leq \frac{\mu}{a}. \tag{14}$$

Proof. For all $a > 0$, the following holds: $\mu = \mathbb{E}(X) = \sum_\alpha \alpha \, Pr(X = \alpha) \geq \sum_{\alpha \geq a} \alpha \, Pr(X = \alpha) \geq a \sum_{\alpha \geq a} Pr(X = \alpha) = a \, Pr(X \geq a)$. And therefore $Pr\big(X \geq a\big) \leq \frac{\mu}{a}$ holds. ■

Lemma I.2. ***Chebyshev's Inequality.*** *Suppose X is a random variable with expectation value $\mu = \mathbb{E}(X)$ and non-zero variance $\sigma^2 = Var(X)$, then for $a > 0$:*

$$Pr\Big(\big|X - \mu\big| \geq a\,\sigma\Big) \leq \frac{1}{a^2}. \tag{15}$$

Proof. Define the non-negative random variable $Y = (X - \mu)^2$. The expectation value of Y equals the variance of X: $\mathbb{E}(Y) = Var(X)$. Applying Lemma I.1 to random variable Y yields: $Pr\big(Y \geq b^2\big) = Pr\big(|X - \mu| \geq b\big) \leq \frac{\mathbb{E}(Y)}{b^2} = \frac{Var(X)}{b^2}$, for $b > 0$. Taking $b \equiv a\sigma$ (and thus $a > 0$) then gives $Pr\big(|X - \mu| \geq a\sigma\big) \leq \frac{1}{a^2}$. ■

Lemma I.2 states that the probability that a realization of random variable X deviates from its mean by more than a times its standard deviation is not larger than $\frac{1}{a^2}$. This inequality will prove to be very useful for determining the accuracy of the estimates for our quantities of interest (which are obtained through a quantum simulation procedure or a Monte Carlo procedure).

Definition I.5. ***(Ir)reducible Matrix.*** *Suppose A is an $n \times n$ matrix with entries $a_{i,j}$. Matrix A is reducible if a partition of its index set $\{1, 2, ..., n\}$ into non-empty and disjoint sets S_1 and S_2 exists such that $a_{i,j} = 0$ for $i \in S_1$ and $j \in S_2$. If such a partition does not exist, matrix A is irreducible.*

Definition I.6. ***Variance of a Complex Random Variable.*** *Suppose $Z = X + iY$ (where X, $Y \in \mathbb{R}$) represents a complex random variable. The variance of Z is defined as follows:*

$$Var(Z) \equiv \mathbb{E}\big(|Z|^2\big) - |\mathbb{E}\big(Z\big)|^2 = Var(X) + Var(Y). \tag{16}$$

Definition I.7. ***Computational Tractability.*** *If an algorithm takes computational resources that scale polynomially in the system size, it is said to be computationally tractable (or efficient). If this scaling is super-polynomial (typically exponential), it is considered computationally intractable (or inefficient).*

(Markov Chain) Monte Carlo: Suppose one wishes to calculate an expectation value of some observable $\langle \mathcal{O} \rangle$ (e.g. a thermal average of some observable). This expectation value is defined as:

$$\langle \mathcal{O} \rangle = \sum_{x \in \Omega} Pr\big(\mathcal{O} = O(x)\big)\, O(x), \tag{17}$$

where $O(x)$ denotes the realization of $\mathcal{O}$ at x and x lies in a sample space Ω. Instead of explicitly calculating $\langle \mathcal{O} \rangle$, one can sample the expectation value $\langle \mathcal{O} \rangle$ by generating a set of independent samples which are distributed according to $Pr(O(x))$ and calculate the associated sample average of $\mathcal{O}$. Using Lemma I.2, one can show that the sample average $\langle \tilde{\mathcal{O}} \rangle$ estimates the true expectation value $\langle \mathcal{O} \rangle$ to within an ϵ-additive statistical error with probability $1 - \frac{Var(\mathcal{O})}{\mathcal{N}}$ (where $Var(\mathcal{O})$ denotes the variance of observable $\mathcal{O}$)[1], provided that the size of the sample set over which the sample average is calculated is $\mathcal{N}\epsilon^{-2}$. This will be used on numerous occasions throughout this work. This property is particularly useful in instances where one wishes to calculate an expectation value of e.g. an operator associated to a physical system, where the sample space Ω is exponentially large as a function of the system size. This would yield the direct evaluation of $\langle \mathcal{O} \rangle$ in eq. 17 computationally intractable and hence one wishes to indeed estimate $\langle \mathcal{O} \rangle$ with a sample average using a number of samples that is a polynomial as a function of the system size. The question is now how one generates a polynomial number of samples (distributed according to the probability distribution of interest) in an amount of time that is polynomial in the system size as well. A strategy that is often employed is *Markov Chain Monte Carlo* (MCMC), a particular and favoured variant of which is the *Metropolis algorithm* ([10]). In the remainder of this section, we introduce several concepts related to Markov chains and the Metropolis algorithm (primarily based on [11] and [12]). We note that the sampling process in the Monte Carlo scheme for piece-wise stoquastic Hamiltonians that we develop in Chapter IV differs from that of the MCMC schemes presented in this section, and does *not* make explicit use of the Metropolis algorithm.

Markov chains belong to a particular class of correlated chains, and to illustrate how they differ from uncorrelated chains, we first define the latter and afterwards define a Markov chain.

Definition I.8. ***Uncorrelated Chain.*** *Suppose $X_1, X_2, ..., X_P$ are P random variables. The realizations of each X_i lie in a sample space Ω. The chain $X_1, X_2, ..., X_P$ is uncorrelated if the probability of occurrence of a particular sequence of realizations of each of the P random variables is statistically uncorrelated ($P_1(X)$ denotes the independent probability of occurrence for a single X):*

$$Pr_P(X_1, X_2, ..., X_P) = Pr_1(X_1) Pr_1(X_2) ... Pr_1(X_P). \tag{18}$$

Definition I.9. ***Markov Chain.*** *Suppose $X_1, X_2, ..., X_P$ are P random variables. The realizations of each X_i lie in a sample space Ω. The chain $X_1, X_2, ..., X_P$ is a Markov Chain if the probability of occurrence of a particular sequence of realizations of each of the P random variables is as follows (defined in terms of a transition probability $T(X \to X')$ for having a realization X' succeed a realization X):*

$$Pr_P(X_1, X_2, ..., X_P) = Pr_1(X_1) T(X_1 \to X_2) T(X_2 \to X_3) ... T(X_{P-1} \to X_P), \tag{19}$$

where $\sum_{X'} T(X \to X') = 1$. The probability of a realization X' succeeding a realization X in a Markov chain thus depends on X, but not on the realizations preceding X in the chain.

For Markov chains to be practically useful in the context of Markov Chain Monte Carlo, they need to satisfy a number of conditions. We now define these conditions on Markov chains and afterwards discuss the Metropolis algorithm as a particular case of Markov Chain Monte Carlo methods.

Definition I.10. ***Time-Homogeneous Markov Chain.*** *A Markov chain $X_1, X_2, ..., X_P$ (where the realizations of each X_i lie in Ω) is time-homogeneous if, for $X, X' \in \Omega$ and $i \in \{1, 2, ..., P\}$, we have:*

$$Pr\big(X_{i+1} = X' \mid X_i = X\big) = T(X \to X'), \quad \forall i. \tag{20}$$

Definition I.11. ***Irreducible Markov Chain.*** *A Markov chain $X_1, X_2, ..., X_P$ (where the realizations of each X_i lie in Ω) is irreducible if $\forall X, X' \in \Omega$, $\exists\, t \geq 0$ such that:*

$$Pr\big(X_t = X' \mid X_0 = X\big) > 0. \tag{21}$$

[1] If $Var(\mathcal{O})$ is bounded, then the sample average $\langle \tilde{O} \rangle$ will equal the true expectation value $\langle \mathcal{O} \rangle$ as the size of the sample set goes to infinity.

Definition I.12. ***Aperiodic Markov Chain.*** *An irreducible Markov chain $X_1, X_2, ..., X_P$ (where the realizations of each X_i lie in Ω) is aperiodic if $\forall X \in \Omega$, we have:*

$$gcd\left\{t:\ Pr\big(X_t = X \mid X_0 = X\big) > 0\right\} = 1. \tag{22}$$

In other words, the probability that realizations along a Markov chain will return to the realization at $t = 0$ in a periodic manner equals zero for an aperiodic Markov chain.

Definition I.13. ***Stationary Distribution.*** *A distribution π on Ω with respect to the transition probability matrix T (defined by $\langle X|\, T\, |X'\rangle \equiv T(X \to X')$, $\forall X, X' \in \Omega$) is a stationary distribution if:*

$$\boldsymbol{\pi}\, T = \boldsymbol{\pi}, \tag{23}$$

where $\boldsymbol{\pi}$ is defined to be a vector containing as elements $\pi(X)$ (for all $X \in \Omega$). In other words, the distribution π gets propagated by the transition probability matrix T to itself.

Definition I.14. ***Ergodic Markov Chain.*** *A Markov chain $X_1, X_2, ..., X_P$ (where the realizations of each X_i lie in Ω) is ergodic if it is time-homogeneous, irreducible and aperiodic. If a Markov chain with stationary distribution π is ergodic, then ($\forall X_{init}, X \in \Omega$):*

$$Pr\big(X_t = X \mid X_0 = X_{init}\big) \to \pi(X)\,, \quad \text{as } t \to \infty. \tag{24}$$

The Metropolis algorithm consists of generating an ergodic Markov chain with a desired stationary distribution π. To that end, one must find a transition probability distribution $T(X \to X')$ ($\forall X, X' \in \Omega$) which leads to the stationary distribution π. We define $\pi(X, t)$ (where t denotes the MC step), which becomes independent of t as $t \to \infty$ for an ergodic Markov chain. The change of $\pi(X, t)$ to $\pi(X, t+1)$ is governed by the following master equation:

$$\pi(X, t+1) - \pi(X, t) = -\sum_{X'} T(X \to X')\, \pi(X, t) + \sum_{X'} T(X' \to X)\, \pi(X', t). \tag{25}$$

The stationary distribution satisfies $\pi(X, t) = \pi(X, t+1)$ and therefore we have:

$$\sum X' T(X \to X')\, \pi(X, t) = \sum_{X'} T(X' \to X)\, \pi(X', t). \tag{26}$$

A particular solution (the *detailed balance solution*) to this equation is:

$$T(X \to X')\, \pi(X) = T(X' \to X)\, \pi(X'), \quad \forall X, X' \in \Omega. \tag{27}$$

To cast the detailed balance solution in a form that is used for the Metropolis algorithm, we rewrite the transition probability as $T(X \to X') = r_{XX'} p_{XX'}$ (where $r_{XX'} = r_{X'X}$ and $r_{XX'}$ is non-negative ($\forall X, X' \in \Omega$), and $\sum_{X'} r_{XX'} = 1$). We note that $p_{XX'} \in [0, 1]$. The detailed balance solution now gives the following equation in terms of p:

$$\frac{p_{XX'}}{p_{X'X}} = \frac{\pi(X')}{\pi(X)}. \tag{28}$$

In the Metropolis algorithm, the quantities $r_{XX'}$ and $p_{XX'}$ respectively correspond to a trial step probability and a step acceptance probability.

Definition I.15. ***Metropolis Algorithm.*** *Given a state X, one proposes a new state X' with a probability $r_{XX'}$.*

- *The proposed step $X \to X'$ is accepted with probability $p_{XX'} = 1$ for $\pi(X') \geq \pi(X)$ and with probability $p_{XX'} = \pi(X')/\pi(X)$ if $\pi(X') < \pi(X)$ (satisfying eq. 28).*
- *If the proposed step is not accepted, it is rejected and the system remains in X.*
- *If the proposed step is accepted, then X' replaces X.*

This process is repeated for many Monte Carlo steps until arrival at the stationary distribution π.

One can now calculate the sample average of a quantity of interest over some sample set, whose samples are distributed according to the distribution of interest π and are generated by means of the Metropolis algorithm. One obtains such a sample set by, after arrival at the stationary distribution π, taking every X that is separated from the other X's in the set by a number of MC steps that is at least the correlation time (such that the samples in the sample set are statistically independent). For the evaluation of the transition probability, one does *not* require knowledge of the normalization factor of the distribution of interest π. We note that the number of MC steps equals the *total* number of trials, not only the successful trials.

Definition I.14 ensures that after an (in principle) infinite number of Monte Carlo steps, one arrives at the stationary distribution π provided that the Markov chain is ergodic. If one tries to calculate a sample average of some quantity of interest associated with a physical system, the number of MC steps after which convergence to the stationary distribution is ensured might in practice be superpolynomial as a function of the system size (this number of steps depends on the spectral gap of the transition probability matrix of the Markov chain), yielding the calculation computationally intractable. This is a prominent factor limiting the computational tractability of Markov Chain Monte Carlo schemes.

I.4 Trotterization

To obtain an explicit expression for the propagation operator of a quantum system consisting of n variables (which each can attain one of σ discrete values), one needs to exponentiate a $\sigma^n \times \sigma^n$ matrix – which is generally intractable in terms of computational effort. To avoid this intractable exponentiation, one decomposes the propagation operator into a repeated (M times) string of local propagation operators, where the computation of these local propagation operators is tractable since they correspond to interactions of locality $k = log(poly(n))$. The convergence of this decomposition as a function of M is guaranteed by the Lie-Trotter formula. We note furthermore that even if Hamiltonian H is sparse, its associated propagation operator might *not* be sparse and therefore even the storage of the propagation operator itself is inefficient in general. Since the concept of Trotterization is of such central importance for the remainder of this report, we will discuss it in some detail.

Suppose matrices A and B are Hermitian (and therefore also square) and of finite dimension. We consider the decomposition of the operator e^{A+B} into the product $e^A e^B$, where generally $[A, B] \neq 0$. The Lie-Trotter formula relates these operators as follows ([2],[13]):

$$e^{-it(A+B)} = \lim_{M\to\infty} \left(e^{-itA/M} e^{-itB/M}\right)^M, \tag{29}$$

where $t \in \mathbb{R}_+$ and we have introduced the Trotter variable M. This result is useful in the setting of real-time evolution of a system – i.e. the decomposition of the propagation unitary of the Schrödinger equation (e^{-iHt}). For imaginary-time evolution (which is encountered in quantum Monte Carlo settings), the following holds:

$$e^{-\tau(A+B)} = \lim_{M\to\infty} \left(e^{-\tau A/M} e^{-\tau B/M}\right)^M, \tag{30}$$

where $\tau \in \mathbb{R}_+$.

In the remainder of this section, we consider the decomposition of $e^{-t(A+B)}$ into $\left(e^{-tA/M} e^{-tB/M}\right)^M$, where $t \in \mathbb{R}_+$ and A and B are Hermitian in the imaginary-time setting, but are anti-Hermitian in the real-time setting. Specifically, we investigate in this section the error associated with this decomposition for finite choices of M in the imaginary- and real-time settings.

I.4.1 Trotterization for Arbitrary-Time Evolution

We define $S_M \equiv e^{-t(A+B)/M}$ and $T_M \equiv e^{-tA/M} e^{-tB/M}$, and evaluate the expression $S_M - T_M$ to find an upper bound on the matrix norm $||S_M - T_M||$. By definition, $e^{-t(A+B)/M}$ and $e^{-tA/M} e^{-tB/M}$ can be

expanded as follows:

$$S_M = \sum_{j=0}^{\infty} \frac{(-t)^j (A+B)^j}{j! M^j} \quad ; \quad T_M = \sum_{j=0}^{\infty} \sum_{l=0}^{\infty} \frac{(-t)^j A^j}{j! M^j} \frac{(-t)^l B^l}{l! M^l} = \sum_{j=0}^{\infty} \frac{(-t)^j}{j! M^j} \sum_{l=0}^{j} \binom{j}{l} A^l B^{j-l}. \tag{31}$$

$$S_M - T_M = \sum_{j=0}^{\infty} \frac{(-t)^j}{j! M^j} \Big((A+B)^j - \sum_{l=0}^{j} \binom{j}{l} A^l B^{j-l} \Big). \tag{32}$$

The terms for which $j = 0, 1$ vanish in the expression for $S_M - T_M$ (for any $[A, B]$). We therefore write:

$$S_M - T_M = \sum_{j=2}^{\infty} \frac{(-t)^j}{j! M^j} \Big(\underbrace{(A+B)^j}_{\alpha} - \underbrace{\sum_{l=0}^{j} \binom{j}{l} A^l B^{j-l}}_{\beta} \Big), \tag{33}$$

where we have identified quantities α and β. We note that α corresponds to the sum of all products (with j terms) of A and/or B matrices. β is the sum of all products (with j terms) of A and/or B matrices, with the constraint that all A's precede the B's. It will be convenient to rewrite the terms α and β as follows:

$$\alpha = \sum_{x \in \{0,1\}^j} \prod_{r=1}^{j} \begin{Bmatrix} A, & \text{if } x_r = 0 \\ B, & \text{if } x_r = 1 \end{Bmatrix} \quad ; \quad \beta = \sum_{x \in \{0,1\}^j} A^{j-|x|} B^{|x|}, \tag{34}$$

where x is a bit-string of j bits and $|x|$ denotes the Hamming weight of string x. We now bound $||S_M - T_M||$ by using the triangle inequality for the matrix norm $||\cdot||$:

$$||S_M - T_M|| \leq \sum_{j=2}^{\infty} \frac{t^j}{j! M^j} \, 2^j \sup_{x \in \{0,1\}^j} \underbrace{\left\| \prod_{r=1}^{j} \begin{Bmatrix} A, & \text{if } x_r = 0 \\ B, & \text{if } x_r = 1 \end{Bmatrix} - A^{j-|x|} B^{|x|} \right\|}_{K(x)}. \tag{35}$$

The term in $||\cdot||$'s (denoted by $K(x)$) attains its supremum value as a function of x for:

$$x_{sup} = \begin{cases} \{1\}^{j/2} \{0\}^{j/2}, & \text{for } j \text{ even}, \\ \{1\}^{(j-1)/2} \{0\}^{(j+1)/2}, & \text{for } j \text{ odd and } ||A|| > ||B||, \\ \{1\}^{(j+1)/2} \{0\}^{(j-1)/2}, & \text{for } j \text{ odd and } ||B|| > ||A||. \end{cases} \tag{36}$$

$K(x_{sup}) = \sup_{x \in \{0,1\}^j} K(x)$ can itself be upper bounded as follows:

$$K(x_{sup}) \leq \begin{cases} \frac{j^2}{4} ||\,[A,B]\,|| \, ||A||^{j/2-1} \, ||B||^{j/2-1}, & \text{for } j \text{ even}, \\ \frac{j^2-1}{4} ||\,[A,B]\,|| \, ||A||^{(j-1)/2} \, ||B||^{(j-3)/2}, & \text{for } j \text{ odd and } ||A|| > ||B||, \\ \frac{j^2-1}{4} ||\,[A,B]\,|| \, ||A||^{(j-3)/2} \, ||B||^{(j-1)/2}, & \text{for } j \text{ odd and } ||B|| > ||A||. \end{cases} \tag{37}$$

From this we conclude that for general $j \in \mathbb{N}$, $K(x_{sup}) \leq \frac{j^2}{4} ||\,[A,B]\,|| \Big(||A|| + ||B|| \Big)^{j-2}$. The inequality in eq. 35 now reduces to:

$$||S_M - T_M|| \leq \sum_{j=2}^{\infty} \frac{t^j}{j! M^j} \, 2^j \, \frac{j^2}{4} ||\,[A,B]\,|| \Big(||A|| + ||B|| \Big)^{j-2}. \tag{38}$$

So that:

$$\frac{M^2}{t^2} ||S_M - T_M|| \leq \sum_{j=2}^{\infty} \left(\frac{t \left(||A|| + ||B|| \right)}{M} \right)^{j-2} \frac{2^{j-2} \, j^2}{j!} ||\,[A,B]\,||. \tag{39}$$

This expression reduces to (if no further assumptions about the matrix norms $||A||$ and $||B||$ are imposed):

$$||S_M - T_M|| \leq ||\,[A,B]\,|| \left(e^{\frac{2 \left(||A|| + ||B|| \right) t}{M}} \left(\frac{t^2}{M^2} + \frac{t}{2 \left(||A|| + ||B|| \right) M} \right) - \frac{t}{2 \left(||A|| + ||B|| \right) M} \right). \tag{40}$$

Clearly, if $[A,B]=0$, then $S_M = T_M$. If one assumes that $\frac{t\left(||A||+||B||\right)}{M} \leq 1$, then:

$$\begin{aligned}\frac{M^2}{t^2}||S_M - T_M|| &\leq \sum_{j=2}^{\infty} \frac{2^{j-2}j^2}{j!} \,||\,[A,B]\,|| \\ &\leq \sum_{j=0}^{\infty} \frac{2^{j-2}j^2}{j!} \,||\,[A,B]\,|| \\ &= \frac{3e^2}{2} \,||\,[A,B]\,||.\end{aligned} \tag{41}$$

We thus find, provided that $\frac{t\left(||A||+||B||\right)}{M} \leq 1$, that the absolute error of approximating $e^{-t(A+B)/M}$ by $e^{-tA/M}e^{-tB/M}$ is as follows:

$$\boxed{||S_M - T_M|| \leq \frac{3e^2}{2} \,||\,[A,B]\,||\, \frac{t^2}{M^2}.} \tag{42}$$

So in the parameter regime for which $\frac{t\left(||A||+||B||\right)}{M} \leq 1$ the bound on the absolute error $||S_M - T_M||$ scales as $\frac{t^2}{M^2}$, instead of exponentially in $\frac{t}{M}$ (which holds in a general parameter regime).

Next, we consider the error of approximating $e^{-t(A+B)} = \left(S_M\right)^M$ by $\left(e^{-tA/M}e^{-tB/M}\right)^M = \left(T_M\right)^M$. We determine a bound on $\Big|\Big|\left(S_M\right)^M - \left(T_M\right)^M\Big|\Big|$ in terms of the bound on $\big|\big|S_M - T_M\big|\big|$ from eq. 42:

$$\begin{aligned}\Big|\Big|\left(S_M\right)^M - \left(T_M\right)^M\Big|\Big| =& \Big|\Big|\sum_{j=1}^{M} \left(S_M\right)^{j-1}\left(S_M - T_M\right)\left(T_M\right)^{M-j}\Big|\Big| \\ \leq& \sum_{j=1}^{M}\Big|\Big|\left(S_M\right)^{j-1}\left(S_M - T_M\right)\left(T_M\right)^{M-j}\Big|\Big| \\ \leq& \sum_{j=1}^{M}\Big|\Big|\left(S_M\right)^{j-1}\Big|\Big|\,\Big|\Big|\left(S_M - T_M\right)\Big|\Big|\,\Big|\Big|\left(T_M\right)^{M-j}\Big|\Big| \\ \leq& \sum_{j=1}^{M}\Big|\Big|\left(S_M - T_M\right)\Big|\Big| \\ =& M\,\Big|\Big|\left(S_M - T_M\right)\Big|\Big|,\end{aligned} \tag{43}$$

where the last inequality holds provided that $||S_M|| \leq 1$ and $||T_M|| \leq 1$. We conclude that if $||S_M|| \leq 1$, $||T_M|| \leq 1$ and $\frac{t\left(||A||+||B||\right)}{M} \leq 1$, then:

$$\boxed{\Big|\Big|\left(S_M\right)^M - \left(T_M\right)^M\Big|\Big| \leq \frac{3e^2}{2} \,||\,[A,B]\,||\, \frac{t^2}{M}.} \tag{44}$$

I.4.2 Trotterization for Real-Time Evolution

We now consider the particular case of trotterization for real-time evolution – i.e. the case for which the operators A and B are anti-Hermitian. The operators $S_M \equiv e^{-t(A+B)/M}$ and $T_M \equiv e^{-tA/M}e^{-tB/M}$ are now unitaries (since exponentials of anti-Hermitian matrices are unitary) and therefore have unit norm: $||S_M|| = ||T_M|| = 1$. This property allows for a slightly different analysis ([14]) in comparison to that of the previous section, and results in an upper bound on the absolute Trotter error that scales as t^2/M, regardless of the parameter regime.

By definition, a unitary operator U has the property $U^\dagger = U^{-1}$. In addition, multiplying an operator by a unitary does not change its matrix norm. We define operator $F_M \equiv T_M^\dagger S_M - I$, whose norm equals

the quantity of interest $||S_M - T_M||$:

$$||F_M|| = ||T_M^\dagger S_M - I|| = ||S_M - T_M||. \tag{45}$$

The derivative of operator F_M with respect to $\frac{t}{M}$ is given by:

$$\frac{\partial F_M}{\partial(t/M)} = e^{tA/M} \underbrace{\left(A\, e^{tB/M} - e^{tB/M} A\right)}_{[A,\, e^{tB/M}]} e^{-t(A+B)/M}. \tag{46}$$

The commutator $[A,\, e^{tB/M}]$ can be conveniently expressed in terms of the commutator $[A, B]$ as follows:

$$\frac{\partial}{\partial(t/M)}\left(e^{-tB/M}[A,\, e^{tB/M}]\right) = -e^{-tB/M}[A, B]\, e^{tB/M}. \tag{47}$$

Integrating this expression over a time interval $\frac{t}{M}$ (a single time-slice) yields:

$$e^{-tB/M}[A,\, e^{tB/M}] = -\int_0^{t/M} e^{-sB}[A, B]\, e^{sB} ds. \tag{48}$$

The explicit expression for $[A,\, e^{tB/M}]$ is now:

$$[A,\, e^{tB/M}] = -e^{tB/M} \int_0^{t/M} e^{-sB}[A, B]\, e^{sB} ds. \tag{49}$$

Using this expression for $[A,\, e^{tB/M}]$, we find for $\frac{\partial F_M}{\partial(t/M)}$:

$$\frac{\partial F_M}{\partial(t/M)} = -e^{tA/M} e^{tB/M} \int_0^{t/M} e^{-sB}[A, B]\, e^{sB} ds\, e^{-t(A+B)/M}. \tag{50}$$

Integrating once again over a single time-slice yields:

$$F_M = -\int_0^{t/M} \left(e^{rA} e^{rB} \int_0^r e^{-sB}[A, B] e^{sB} ds\, e^{-r(A+B)}\right) dr + \underbrace{F_M(t/M = 0)}_{=0}, \tag{51}$$

where r and s are real parameters. We now bound the matrix norm $||S_M - T_M||$:

$$\begin{aligned}
||S_M - T_M|| = ||F_M|| &= \left|\left|\int_0^{t/M} \left(e^{rA} e^{rB} \int_0^r e^{-sB}[A, B] e^{sB} ds\, e^{-r(A+B)}\right) dr\right|\right| \\
&\leq \int_0^{t/M} \underbrace{\left|\left|e^{rA}\right|\right|}_{=1} \underbrace{\left|\left|e^{rB}\right|\right|}_{=1} \left|\left|\int_0^r e^{-sB}[A, B] e^{sB} ds\right|\right| \underbrace{\left|\left|e^{-r(A+B)}\right|\right|}_{=1} dr \\
&\leq \int_0^{t/M} \int_0^r \underbrace{\left|\left|e^{-sB}\right|\right|}_{=1} ||\, [A, B]\, || \underbrace{\left|\left|e^{sB}\right|\right|}_{=1} ds\, dr \\
&= \int_0^{t/M} r\, ||\, [A, B]\, ||\, dr = ||\, [A, B]\, ||\, \frac{t^2}{2M^2},
\end{aligned} \tag{52}$$

where in the last line we have (explicitly) assumed operators A and B to be time-independent. As in the previous section, we bound $\left|\left|\left(S_M\right)^M - \left(T_M\right)^M\right|\right|$ using the bound on $||S_M - T_M||$:

$$\boxed{\left|\left|\left(S_M\right)^M - \left(T_M\right)^M\right|\right| \leq M\, ||S_M - T_M|| \leq ||\, [A, B]\, ||\, \frac{t^2}{2M}.} \tag{53}$$

We reiterate that this result holds provided that $||S_M|| \leq 1$ and $||T_M|| \leq 1$. Since S_M and T_M are unitary operators in the current discussion, these conditions are always obeyed. We conclude that eq. 53 holds

in any parameter regime if the time-independent operators A and B are anti-Hermitian (corresponding to real-time propagation). In contrast, the error bound that was derived in the previous section for arbitrary-time propagation (including imaginary-time propagation) scaling as t^2/M holds only in the parameter regime for which $||S_M|| \leq 1$, $||T_M|| \leq 1$ and $\frac{t\left(||A||+||B||\right)}{M} \leq 1$.

I.4.3 Trotterizing Hamiltonian Dynamics (in Real- and Imaginary-Time)

We are mostly interested in exploiting the concept of Trotterization for efficiently simulating the dynamics of a quantum system in either imaginary or real time (and thereby obtaining some of the eigenvalues of its Hamiltonian). Suppose $H = \sum_{i=1}^{N} H_i$ represents a k-local Hamiltonian of a quantum system. $\{H_i\}_{i=1}^{N}$ is generally a set of non-commuting terms but can be divided into groups, such that within each group all terms commute. For a given set $\{H_i\}_{i=1}^{N}$, we denote the minimum possible number of these groups by Γ. This number of groups is at most N (in the minimally separable case) and equals 1 in the trivial case where all H_i's commute with each other. The Hamiltonian H can thus be decomposed as $H = \sum_{\gamma=1}^{\Gamma} \hat{H}_\gamma$, where all $\hat{H}_\gamma$ do not commute with each other, but the terms of which each individual $\hat{H}_\gamma$ is composed do commute.

Using the results obtain thus far in this section, we evaluate the error associated with the following (first-order) Trotterizations for Hermitian matrices H and finite M:

$$e^{-itH} \approx \left(\prod_{\gamma=1}^{\Gamma} e^{-it\hat{H}_\gamma/M} \right)^M, \quad \text{for real-time evolution,} \tag{54a}$$

$$e^{-\tau H} \approx \left(\prod_{\gamma=1}^{\Gamma} e^{-\tau\hat{H}_\gamma/M} \right)^M, \quad \text{for imaginary-time evolution.} \tag{54b}$$

By successively applying eq. 53 to the operator $e^{-it\sum_\gamma \hat{H}_\gamma}$, we find the following upper bound on the absolute Trotter error for real-time Hamiltonian evolution:

$$\left|\left| e^{-itH} - \left(\prod_{\gamma=1}^{\Gamma} e^{-it\hat{H}_\gamma/M} \right)^M \right|\right| \leq \sum_{\gamma'=1}^{\Gamma-1} \sum_{\gamma>\gamma'} || \, [\hat{H}_{\gamma'}, \hat{H}_\gamma] \, || \, \frac{t^2}{2M}. \tag{55}$$

Similarly, we find the following upper bound on the absolute Trotter error for imaginary-time Hamiltonian evolution for the operator $e^{-\tau\sum_\gamma \hat{H}_\gamma}$ (by successively applying eq. 44):

$$\left|\left| e^{-\tau H} - \left(\prod_{\gamma=1}^{\Gamma} e^{-\tau\hat{H}_\gamma/M} \right)^M \right|\right| \leq 3e^2 \sum_{\gamma'=1}^{\Gamma-1} \sum_{\gamma>\gamma'} || \, [\hat{H}_{\gamma'}, \hat{H}_\gamma] \, || \, \frac{\tau^2}{2M}, \tag{56}$$

which holds provided that $\left|\left|e^{-\tau H/M}\right|\right| \leq 1$, $\left|\left|e^{-\tau\hat{H}_\gamma/M}\right|\right| \leq 1$ $(\forall\gamma)$ and $\frac{\tau\left(\sum_\gamma ||\hat{H}_\gamma||\right)}{M} \leq 1$. Γ has been defined as being the smallest number of subsets into which the set $\{H_i\}_{i=1}^{N}$ can be divided, such that within each subset all H_i's commute. We note that this number of subsets can, of course, always be taken bigger than Γ (N in the most trivial case). This leads to an upper bound that is generally less tight than in the Γ-subsets case, but which scales equally well in t and M.

Making sure that the number of subsets into which $\{H_i\}_{i=1}^{N}$ is divided is minimal also brings about another advantage: Suppose one wants to calculate the result of letting a Trotterized version of the operators e^{-itH} or $e^{-\tau H}$ act on a state $|\Phi\rangle$. If one takes the number of subsets equal to N (the trivial setting), then the number of subsequent operations required for the simulation will scale as MN. In contrast, if one is able to identify Γ, then one can let the commuting propagation operators within each subset act on the state in parallel. By doing this, the number of subsequent operations required for the simulation will scale as $M\Gamma$.

A specific example of this grouping of the set $\{H_i\}_{i=1}^{N}$ into subsets such that within each subset the H_i's commute is the *checkerboard decomposition*. Suppose we are simulating the dynamics of a spin system, whose lattice is bipartite and whose interactions are of locality $k = 2$. In that case, $\Gamma = 2$. The simulation in that case comprises of letting the two subsets of $\{H_i\}_{i=1}^{N}$ act on the state of the system in an alternating

fashion. For one-dimensional systems, this creates a checkerboard pattern as a function of time and the single spatial component.

In the context of quantum simulation (by means of quantum phase estimation) or quantum Monte Carlo methods, one is specifically interested in finding an upper bound on the absolute Trotter error on the signals $\langle\Phi| e^{-itH} |\Phi\rangle$ and $\langle\Phi| e^{-\tau H} |\Phi\rangle$, respectively (where $|\Phi\rangle$ is some normalized state of the system that is being considered). Suppose operator Δ_M denotes the difference between e^{-itH} (in the real-time setting) and $e^{-\tau H}$ (in the imaginary-time setting) and their Trotterized versions. Then we would like to find an upper bound on the quantity $\big|\langle\Phi| \Delta_M |\Phi\rangle\big|$. The Cauchy-Schwarz inequality for inner products and property 4 of the *Induced Matrix Norm* (introduced in the Preliminaries section) yield:

$$\begin{aligned}\big|\langle\Phi| \Delta_M |\Phi\rangle\big| &\leq \underbrace{\big|\big|\langle\Phi|\big|\big|}_{=1} \big|\big|\Delta_M |\Phi\rangle\big|\big| \\ &\leq \big|\big|\Delta_M\big|\big| \underbrace{\big|\big||\Phi\rangle\big|\big|}_{=1}.\end{aligned} \tag{57}$$

The norm of the absolute error between the actual signal of interest and its Trotterized version is thus upper bounded by the norm of the operator difference, for whom we have found upper bounds in this section. We summarize the results of this section in Lemma I.3 below.

Lemma I.3. ***First-Order Trotter Decomposition.*** *Given a k-local Hamiltonian $H = \sum_i^N H_i$. Furthermore, suppose the set $\{H_i\}_{i=1}^N$ can be divided into a minimum of Γ subsets $\{\hat{H}_\gamma\}_{\gamma=1}^\Gamma$, such that within each individual subset all H_i's commute. Then the quantities $\big|\langle\Phi| e^{-itH} |\Phi\rangle - \langle\Phi| \big(\prod_\gamma e^{-it\hat{H}_\gamma/M}\big)^M |\Phi\rangle\big|$ and $\big|\langle\Phi| e^{-\tau H} |\Phi\rangle - \langle\Phi| \big(\prod_\gamma e^{-\tau\hat{H}_\gamma/M}\big)^M |\Phi\rangle\big|$ (where $|\Phi\rangle$ is a normalized state and $t, \tau \in \mathbb{R}_+$) are bounded as follows:*

$$\Big|\langle\Phi| e^{-itH} |\Phi\rangle - \langle\Phi| \big(\prod_\gamma e^{-it\hat{H}_\gamma/M}\big)^M |\Phi\rangle\Big| \leq \sum_{\gamma'=1}^{\Gamma-1} \sum_{\gamma>\gamma'} || \, [\hat{H}_{\gamma'}, \hat{H}_\gamma] \, || \, \frac{t^2}{2M}, \tag{58a}$$

$$\Big|\langle\Phi| e^{-\tau H} |\Phi\rangle - \langle\Phi| \big(\prod_\gamma e^{-\tau\hat{H}_\gamma/M}\big)^M |\Phi\rangle\Big| \leq 3e^2 \sum_{\gamma'=1}^{\Gamma-1} \sum_{\gamma>\gamma'} || \, [\hat{H}_{\gamma'}, \hat{H}_\gamma] \, || \, \frac{\tau^2}{2M}, \tag{58b}$$

where the second inequality holds provided that $\big|\big|e^{-\tau H/M}\big|\big| \leq 1$, $\big|\big|e^{-\tau\hat{H}_\gamma/M}\big|\big| \leq 1$ $(\forall\gamma)$ and $\frac{\tau\left(\sum_\gamma ||\hat{H}_\gamma||\right)}{M} \leq 1$.

I.4.4 Higher-Order Trotter Decompositions

The upper bounds on the absolute Trotter error obtained thus far hold for first-order approximations – $T_M = e^{-tA/M} e^{-tB/M}$ is the first-order approximant of operator $S_M = e^{-t(A+B)/M}$. This is manifested in the fact that, as was shown above, the Taylor expansion around $\frac{t}{M} = 0$ of the function $S_M - T_M$ does not contain any terms that are constant or linear in $\frac{t}{M}$. Taking $T_M = e^{-tA/2M} e^{-tB/M} e^{-tA/2M}$ results in the Taylor expansion of $S_M - T_M$ around $\frac{t}{M} = 0$ not containing terms that are constant, linear or quadratic in $\frac{t}{M}$. This T_M is therefore a second-order approximant of S_M. In general, if T_M is a pth-order approximant of operator S_M, then the nth derivative of the Taylor expansion of $S_M - T_M$ around $\frac{t}{M} = 0$ equals zero, for $n \in \{0, 1, ..., p\}$.

The reason for considering higher-order approximants is that the scaling of the associated error bounds is better as a function of M. Suppose $e^{-itH/M}$ and $e^{-\tau H}$ are approximated by the pth-order approximants $\mathcal{T}_M(p,t)$ and $\mathcal{T}_M(p,\tau)$, respectively. Then the upper bounds on $\big|\big|e^{-itH} - \big(\mathcal{T}_M(p,t)\big)^M\big|\big|$ and $\big|\big|e^{-\tau H} - \big(\mathcal{T}_M(p,\tau)\big)^M\big|\big|$ scale as $t^{\,p+1}/M^p$ and $\tau^{\,p+1}/M^p$, respectively (provided that in the case of imaginary-time propagation, conditions similar to those mentioned in Lemma I.3 are obeyed) ([15],[16]).

The question is now how one constructs pth-order approximants of the operators $e^{-itH/M}$ and $e^{-\tau H/M}$. In [17], a widely used scheme is proposed for constructing pth order approximants: The first-order approximants of $e^{-itH/M}$ and $e^{-\tau H/M}$ are $\mathcal{T}_M(p=1,t) = \prod_{\gamma=1}^\Gamma e^{-it\hat{H}_\gamma/M}$ and $\mathcal{T}_M(p=1,\tau) = \prod_{\gamma=1}^\Gamma e^{-\tau\hat{H}_\gamma/M}$, respectively. Their second-order approximants are $\mathcal{T}_M(p=2,t) = \prod_{\gamma=\Gamma}^1 e^{-it\hat{H}_\gamma/(2M)} \prod_{\gamma=1}^\Gamma e^{-it\hat{H}_\gamma/(2M)}$

and $\mathcal{T}_M(p=2,\tau) = \prod_{\gamma=\Gamma}^{1} e^{-\tau \hat{H}_\gamma/(2M)} \prod_{\gamma=1}^{\Gamma} e^{-\tau \hat{H}_\gamma/(2M)}$. Higher-order approximants are recursively defined (starting from these second-order approximants) as follows:

$$\mathcal{T}_M\big(p+2,t\big) = \Big(\mathcal{T}_M\big(p,v_p t\big)\Big)^2 \times \mathcal{T}_M\big(p,(1-4v_p)t\big) \times \Big(\mathcal{T}_M\big(p,v_p t\big)\Big)^2, \tag{59a}$$

$$\mathcal{T}_M\big(p+2,\tau\big) = \Big(\mathcal{T}_M\big(p,v_p \tau\big)\Big)^2 \times \mathcal{T}_M\big(p,(1-4v_p)\tau\big) \times \Big(\mathcal{T}_M\big(p,v_p \tau\big)\Big)^2, \tag{59b}$$

where $v_p \equiv 1/\big(4-4^{1/(p+1)}\big)$.

For the practical implementation of the Trotterized versions of both the real-time and imaginary-time evolution, it is important to consider the total number of k-local propagation operators required to simulate e^{-itH} and $e^{-\tau H}$ (for a given order p and Trotter variable M). For the scheme in eq. 59, the number of these k-local propagation operators required to be implemented for the simulation of e^{-itH} and $e^{-\tau H}$ for $p>1$ is $2MN\,5^{\frac{p}{2}-1}$ (and for $p=1$ is MN). Using the intrinsic parallelizability due to the grouping of the k-local bond Hamiltonians into Γ groups, the number of required time slices needed to implement the Hamiltonian evolution reduces to $2M\Gamma\,5^{\frac{p}{2}-1}$ (where within each time slice, k-local propagation operators working on separate subsystems can be simultaneously implemented).

Now suppose the Trotter error of implementing e^{-itH} and $e^{-\tau H}$ by means of a pth-order approximant and Trotter variable M is given by ϵ. As mentioned above, $\epsilon \propto \frac{t^{p+1}}{M^p}$ and $\epsilon \propto \frac{\tau^{p+1}}{M^p}$ in that case. For a given ϵ, the number of k-local propagation operators into which the evolutions are decomposed then scales as $N\,5^{p/2}\frac{t^{1+1/p}}{\epsilon^{1/p}}$ and $N\,5^{p/2}\frac{\tau^{1+1/p}}{\epsilon^{1/p}}$. Similarly, the number of time slices into which the simulation can be decomposed (by making use of parallelization) scales as $\Gamma\,5^{p/2}\frac{t^{1+1/p}}{\epsilon^{1/p}}$ and $\Gamma\,5^{p/2}\frac{\tau^{1+1/p}}{\epsilon^{1/p}}$. We thus conclude that for large p (i.e. high-order decompositions) and a given ϵ, the number of k-local propagation operators in the Trotter decomposition scales approximately linearly in the evolution time of the system under consideration (for real-time and imaginary-time evolution).

Figure 2 depicts the scaling of the number of local propagation operators as a function of the order of the Trotterization scheme (according to the scheme from eq. 59). Clearly, for large p this number of operators increases exponentially with p. Interestingly, in parameter regimes where $\epsilon \ll \tau, t$, the number of operators first decreases as a function of p, before increasing exponentially for large p. We also indeed see that for large p and a given ϵ, the number of local operators scales approximately linearly in $\tau(=t)$.

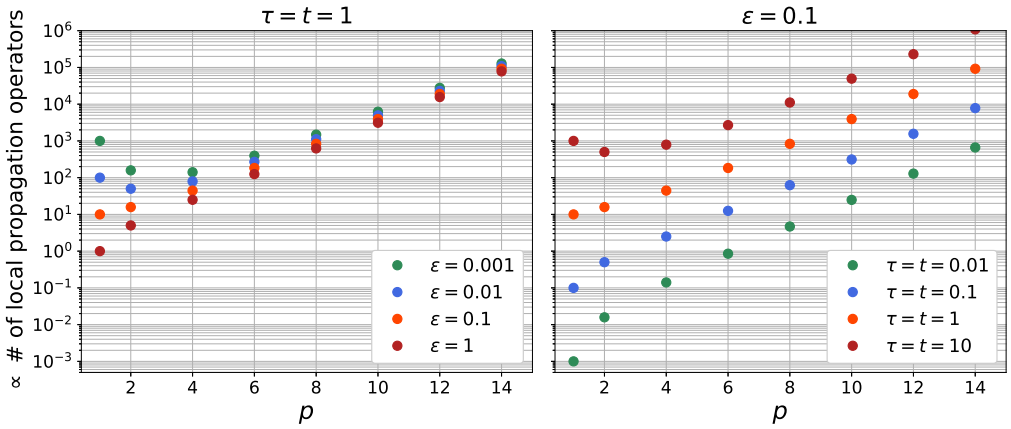


Figure 2: Scaling of the number of local propagation operators (propagating in real time or imaginary time) as a function of the order of the Trotterization scheme p (for $p=1$, and $p>1$ and even according to eq. 59). The left figure depicts the scaling as a function of p for a fixed $\tau = t = 1$ and several values of ϵ. The right figure depicts the scaling as a function of p for a fixed $\epsilon = 0.1$ and several values of $\tau = t$.

As will be discussed in the coming chapters of this report, imaginary-time k-local propagation operators can be implemented in a computationally tractable manner by classical means if the Hamiltonian H is piece-wise stoquastic and if the interaction locality k is at most $log(poly(n))$. In contrast, the efficient

implementation of real-time k-local propagation operators generally requires quantum machinery. We note that in the latter case, a decomposition needs to be found of the k-local gates in terms of an elementary set of 1-qubit and 2-qubit gates (generally to be implemented on a universal quantum computer).

II Quantum Simulation

As discussed in the introductory chapter of this report, simulating a quantum system by classical means is *in general* a computationally intractable task. Suppose one is given the task of obtaining the eigenvalues of a (time-independent) Hamiltonian H associated with a quantum system consisting of n degrees of freedom, where the 'state' of each degree of freedom lies in a σ-dimensional vector space. The Hamiltonian itself is then a $\sigma^n \times \sigma^n$ matrix, so direct diagonalization is generally inefficient. Given that the state of the system evolves in time according to the Schrödinger propagation unitary e^{-itH}, an alternative approach might be to track the system state in time t and determine (some of) the frequencies with which the state elements oscillate. These frequencies then correspond to the eigenvalues of the Hamiltonian.

The dimensionality of the Hilbert space of the quantum system is σ^n – i.e. exponentially large in the system size. To classically store the state of the system therefore requires an amount of memory that grows exponentially in the system size as well. In addition, exponentiating the $\sigma^n \times \sigma^n$ Hamiltonian to obtain the propagation unitary (which is required to implement the time evolution) is a computationally intractable task by classical means. If the Hamiltonian is sufficiently local, however, the latter task of efficiently obtaining the propagation unitary can be solved by Trotterization of the unitary. The actual implementation of the Trotterized propagation unitary, and the associated storing of the system state at several instances in evolution time, *can* be done efficiently using a qubit register and a digital quantum computer. Given for instance an n-qubit register, then the dimensionality of the Hilbert space of the qubit register equals that of the system (for $\sigma = 2$). Storing the state of the system on the qubit register therefore only requires a number of qubits that scales polynomially in the system size. The actual implementation of the time evolution by means of the Trotterized propagation unitary (working on the qubit register) is then implemented by means of a digital quantum computer. By now efficiently implementing the time evolution for many different time intervals, the state of the system can be efficiently tracked in time t by carefully measuring the state of the system for each of the different implemented time intervals.

When implementing a quantum simulation algorithm by means of e.g. quantum phase estimation, only those eigenvalues that have associated eigenstates with sufficiently large overlap with the initial state can be obtained through tracking the system state. It is therefore important to be able to systematically prepare the quantum system in a preferred initial state. In the remainder of this section, we will discuss digital quantum simulation in more detail (and the associated topics of *state preparation*, *system evolution* and *measurement*). In addition, we will briefly discuss the concept of analog quantum simulation. Finally, we will elaborately discuss a digital quantum simulation scheme based on quantum phase estimation.

II.1 Digital Quantum Simulation

Digital quantum simulation is a method that exploits a universal quantum computer to efficiently implement the time evolution (associated with some Hamiltonian H) of a quantum system that is initialized in some state $|\Phi(0)\rangle$. After the evolution has taken place, measurements are performed on the qubit registers to extract useful information. For a digital quantum simulation algorithm to run efficiently, all three stages of *initial-state preparation*, *Hamiltonian evolution* and *measurement* have to run using resources that scale polynomially in the system size.

Since the Schrödinger propagation operator e^{-itH} is a unitary operation, it can be expressed in terms of the elements of some universal set of quantum gates. However, not any operator e^{-itH} can be necessarily decomposed into a set quantum gates that is of $poly(n)$ size – i.e. not any mathematically allowed Hamiltonian gives rise to a propagation operator that can be efficiently implemented on a digital quantum computer.

A class of Hamiltonians that *can* be efficiently simulated is that of the finite-dimensional (i.e. finite σ) and sufficiently local Hamiltonians ([1],[18]). This class of Hamiltonians includes most Hamiltonians that appear in physics. The previously discussed concept of Trotterization is essential for the efficient implementation of the time dynamics associated with the Hamiltonians in this class: The Schrödinger evolution unitary is decomposed into an ordered time series of local propagation operators by means of Trotterization. Each of these local propagation operators is then decomposed into a some universal set of quantum gates – itself consisting in general of 1-qubit and 2-qubit gates.

II.1.1 Initial-State Preparation & Hamiltonian Evolution

In order to demonstrate the fact that not any mathematically allowed Hamiltonian gives rise to a propagation operator that can be efficiently implemented on a universal quantum computer, we consider a counting problem ([2]): Given a Hamiltonian H and its associated propagation unitary e^{-iHt} (which works on an n-qubit register), one can question whether it can be implemented by a circuit that is of size $poly(n)$. This question is closely related to the question of whether an arbitrary initial state $|\Phi(0)\rangle$ can be prepared from some n-qubit starting state using a circuit of $poly(n)$ size.

Suppose we have g gates available to mimic the unitary evolution e^{-iHt}, where each of the g gates acts on at most k qubits. For any particular gate, we have $\binom{n}{k}^g$ available options (taking into account both the g different gates, and the fact that each gate can work on $\binom{n}{k}$ different sets of qubits). This number of available gate choices can be upper bounded as follows:

$$\left.\begin{array}{l}\binom{n}{k} \leq \frac{n^k}{k!} \\ e^k = \sum_{j=0}^{\infty} \frac{k^j}{k!} \geq \frac{k^k}{k!} \implies k! \geq \big(k/e\big)^k\end{array}\right\} \binom{n}{k} \leq \Big(\frac{ne}{k}\Big)^k \implies \binom{n}{k}^g \leq \Big(\frac{ne}{k}\Big)^{kg} = \mathcal{O}(n^{kg}). \tag{60}$$

Suppose now that one has access to quantum circuits consisting of d gates, then the total number of distinct choices for this quantum circuit is bounded as follows:

$$\binom{n}{k}^{gd} \leq \Big(\frac{ne}{k}\Big)^{kgd} = \mathcal{O}(n^{kgd}) \tag{61}$$

This set of distinct quantum circuits can produce a set of states (starting from some initial state) of size that is at most that of the set of distinct quantum circuits. The size of this set of producable states is thus also upper bounded by $\big(\frac{ne}{k}\big)^{kgd} = \mathcal{O}(n^{kgd})$.

Suppose that one wishes to obtain an ϵ-estimate of a state $|\Phi\rangle$ by application of the quantum circuit (starting from initial state $|\Phi(0)\rangle$). The n-qubit state $|\Phi\rangle$ can be expressed in a basis $\{|x\rangle\}$ (of size 2^n) as $|\Phi\rangle = \sum_x \Phi(x)\,|x\rangle$. Due to normalization, $|\Phi\rangle$ lies on the unit sphere (in 2^{n+1} real dimensions) defined by $\sum_x |\Phi(x)|^2 = \sum_x \Big(Re\big(\Phi(x)\big)^2 + Im\big(\Phi(x)\big)^2\Big) = 1$. The surface area of this unit $(2^{n+1}-1)$-sphere is given by:

$$S_{2^{n+1}-1}(r=1) = \frac{2\pi^{2^n}}{\Gamma(2^n)}, \tag{62}$$

where $\Gamma(n) = (n-1)!$ (for $n > 0$). The surface area spanned by the points that are ϵ-close to $|\Phi\rangle$ is approximately equal to the volume of a sphere in $2^{n+1}-1$ real dimensions of radius ϵ. The volume of this $(2^{n+1}-2)$-sphere is given by:

$$V_{2^{n+1}-2}(r=\epsilon) = \frac{2\pi^{2^n-1/2}\epsilon^{2^{n+1}-1}}{(2^{n+1}-1)\Gamma(2^n-1/2)}. \tag{63}$$

The number of distinguishable ϵ-estimates of n-qubit states $|\Phi\rangle$ is thus given by:

$$\mathcal{N}(n,\epsilon) = \frac{S_{2^{n+1}-1}(r=1)}{V_{2^{n+1}-2}(r=\epsilon)} = \sqrt{\pi}\big(2^{n+1}-1\big)\frac{\Gamma(2^n-1/2)}{\Gamma(2^n)}\frac{1}{\epsilon^{2^{n+1}-1}}. \tag{64}$$

If the quantum circuit (consisting of d unitary operations) were to be used to obtain ϵ-estimates of n-qubit states, then the number of producable states (upper bounded by $(\frac{ne}{k})^{kgd}$) has to be at least equal to $\mathcal{N}(n,\epsilon)$:

$$\mathcal{N}(n,\epsilon) = \Big(\frac{ne}{k}\Big)^{kgd}. \tag{65}$$

The number of required unitaries per quantum circuit (d) that satisfies this condition is:

$$d = \frac{1}{gk\,\ln\big(\frac{ne}{k}\big)}\left(\ln\Big(\sqrt{\pi}\big(2^{n+1}-1\big)\frac{\Gamma(2^n-1/2)}{\Gamma(2^n)}\Big) + \big(2^{n+1}-1\big)\ln(1/\epsilon)\right) = \mathcal{O}\Big(\frac{2^n\ln(1/\epsilon)}{\ln(n)}\Big). \tag{66}$$

We thus conclude that – for a given accuracy ϵ – there exist n-qubit states of which an ϵ-estimate cannot

be efficiently obtained by means of implementation of a set of local unitary operations. This means, for instance, that not any desired initial state can be prepared on a qubit register with polynomial resources. In [19], it is argued in a more general sense that although one describes the state of a many-body system (of size n) as a vector in a Hilbert space, the vast majority of states in the Hilbert space is not reached through Hamiltonian dynamics in $poly(n)$ time. Therefore, one concludes that the vast majority of states in the Hilbert space is not physical as they are only produced after exponentially long times.

As mentioned above, the time dynamics according to most Hamiltonians encountered in physical systems *can* be efficiently implemented using a universal quantum computer. Also, numerous useful initial states can be efficiently prepared on a universal quantum computer. In [20], for example, the authors efficiently prepare a Hartree-Fock state on a superconducting-qubit register. The efficient preparation of Hartree-Fock states, due to their generally large overlap with low-lying eigenstates, is an important algorithmic primitive for quantum simulation of quantum chemistry and condensed matter systems.

II.1.2 Measurement

After the propagation unitary e^{-itH} has been (approximately) applied to the initial state $|\Phi(0)\rangle$ to obtain the final state $|\Phi(t)\rangle = e^{-itH}|\Phi(0)\rangle$, one has to perform measurements on the qubit register to extract useful information. One can try to characterize the final system state $|\Phi(t)\rangle$ through quantum state tomography ([21]). However, quantum state tomography in general requires resources that scale exponentially in the system size. Instead of using quantum state tomography to explicitly obtain the final system state, one can try to efficiently extract quantities of interest such as correlation functions and operator spectra directly.

II.2 Analog Quantum Simulation

Another way in which a quantum system can be simulated by quantum means is through analog quantum simulation ([18]). Analog quantum simulation differs from digital quantum simulation in that an analog quantum simulation platform cannot simulate any local quantum system, but rather emulates a particular other quantum system. The Hamiltonian of the system to be simulated H_{system} is directly mapped onto the Hamiltonian of the simulator $H_{\text{simulator}}$: $H_{\text{system}} \longleftrightarrow H_{\text{simulator}}$. The simulator is a system that is experimentally well-controllable, while the system to be simulated is generally not experimentally well-controllable. The simulation of the quantum system of interest by means of the simulator system can be realized if an invertible mapping exists that determines a correspondence between all operators and states of the simulated system and the simulator ([22]).

An advantage of analog quantum simulation is that if the simulator naturally evolves in the same way as the system to be simulated (i.e. their Schrödinger propagators are equal), then there is in principle no need for Trotterization of the Schrödinger propagator. Other advantages of analog quantum simulation relate to state preparation and measurement: Because the system and simulator are so similar, the preparation of desired initial states can occur naturally in processes that mimic the natural relaxation of system to be simulated to an equilibrium state. Direct measurement of some physical quantities in the simulator system yield information on their analog counterparts in the system to be simulated.

Prominent physical realizations of analog quantum simulators include: Cold atomic gases in optical lattices, which can be used for simulating Bose-Hubbard models and, in particular, the associated superfluid to Mott insulator phase transition ([23]). Superconducting circuits (and the related concepts of circuit-QED) which can be used to study (and control) strong light-matter interactions ([24]).

II.3 Quantum Phase Estimation

Quantum phase estimation is used to efficiently determine the phases of eigenvalues of a unitary U. When choosing the unitary $U = e^{-iH\Delta t}$ (the propagation operator of the Schrödinger equation), the eigenvalues of a Hamiltonian H can be approximately obtained. Hence, quantum phase estimation is useful as the basis for a digital quantum simulation scheme. We do note that only those eigenvalues that have associated eigenstates that overlap significantly with the initial state $|\Phi\rangle$ can be obtained. When H is non-stoquastic (and can thus suffer from a sign problem), quantum phase estimation is particularly useful. In later stages of this report, we compare a quantum-phase-estimation-based algorithm to a Monte Carlo method (closely resembling quantum phase estimation) for stoquastic Hamiltonians that will be extensively discussed in

the next chapters. We specifically consider here a form of quantum phase estimation where the unitary U is efficiently implemented to act on a qubit register by means of a digital quantum simulator, but the post-processing procedure to actually obtain the phases of the eigenvalues of U is performed classically. This differs from the approach that is described in e.g. [25], where the post-processing to obtain the eigenvalues requires a machine that can efficiently implement the *Quantum Fourier Transform* (such as a digital quantum simulator). In the current approach of classical post-processing, the use of quantum phase estimation to determine *multiple* eigenvalues arises more naturally.

Figure 3 schematically depicts a basic quantum circuit which is used for quantum phase estimation. It involves an n-qubit register and a single ancillary qubit. The state of the composite system can be tracked through the circuit as follows (where $R(\theta) \equiv e^{-i\theta\sigma_z/2}$):

$$\begin{aligned}
&\text{I}: \quad |+\rangle_a \otimes |\Phi\rangle\,, \\
&\text{II}: \quad \frac{1}{\sqrt{2}}\Big(|0\rangle_a \otimes |\Phi\rangle + U^k\,|1\rangle_a \otimes |\Phi\rangle \Big), \\
&\text{III}: \quad \frac{1}{\sqrt{2}}\Big(\big(e^{-i\theta/2}\,|0\rangle_a\big) \otimes |\Phi\rangle + U^k \big(e^{i\theta/2}\,|1\rangle_a\big) \otimes |\Phi\rangle \Big), \\
&\text{IV}: \quad \frac{1}{2}\Big(\big(e^{-i\theta/2} I + e^{i\theta/2} U^k\big)\,|0\rangle_a \otimes |\Phi\rangle + \big(e^{-i\theta/2} I - e^{i\theta/2} U^k\big)\,|1\rangle_a \otimes |\Phi\rangle \Big).
\end{aligned} \tag{67}$$

The probability to measure state $|0\rangle$ ($m = 0$) on the ancillary qubit after application of the depicted gates is then given by:

$$Pr\big(m = 0 \,|\, k, \theta\big) = \frac{1}{2} + \frac{1}{4}\Big(e^{i\theta}\,\langle\Phi|\,U^k\,|\Phi\rangle + e^{-i\theta}\big(\,\langle\Phi|\,U^k\,|\Phi\rangle\,\big)^* \Big) = \begin{cases} \frac{1}{2} + \frac{1}{2} Re\Big(\,\langle\Phi|\,U^k\,|\Phi\rangle\,\Big), & \text{for } \theta = 0, \\ \frac{1}{2} - \frac{1}{2} Im\Big(\,\langle\Phi|\,U^k\,|\Phi\rangle\,\Big), & \text{for } \theta = \frac{\pi}{2}. \end{cases} \tag{68}$$

In the last expression, we have restricted ourselves to $\theta = 0$ and $\theta = \frac{\pi}{2}$. As will become apparent shortly hereafter, these are the parameter values of interest.

We choose $U = e^{-iH\Delta t}$ (where $\Delta t \in \mathbb{R}_+$ is fixed, H is Hermitian and has 2^n possibly degenerate eigenvalues). Using the spectral decomposition of H (in terms of its eigenstates $\{|\psi_j\rangle\}$ and eigenvalues $\{E_j\}$), the 'signal' $\langle\Phi|\,U^k\,|\Phi\rangle$ can be rewritten as $\sum_j |\langle\psi_j|\Phi\rangle|^2 e^{-iE_j\Delta t k}$. Using the expression in eq. 68, one can determine (a subset of) the eigenvalues of H, provided that $Pr(m = 0 \,|\, k, \theta = \{0, \frac{\pi}{2}\})$ can be efficiently evaluated for a range of k values.

Since the direct evaluation, and therefore the implementation, of $U = e^{-iH\Delta t}$ is inefficient, U should be Trotterized to give $\tilde{U} \equiv \big(\mathcal{T}_M(p, \Delta t)\big)^M$ (where the ordered real-time series $\mathcal{T}_M(p, \Delta t)$ has been introduced in the previous chapter), where the Trotterization is of pth order. Since the Trotterized version of U corresponds to a product of local (unitary) propagation operators, $\tilde{U}$ is itself a unitary operator as well. Therefore, quantum phase estimation is in practice used to obtain the phases of the eigenvalues of unitary $\tilde{U}$, which converge to the eigenvalues of U in the limit of large Trotter variable M.

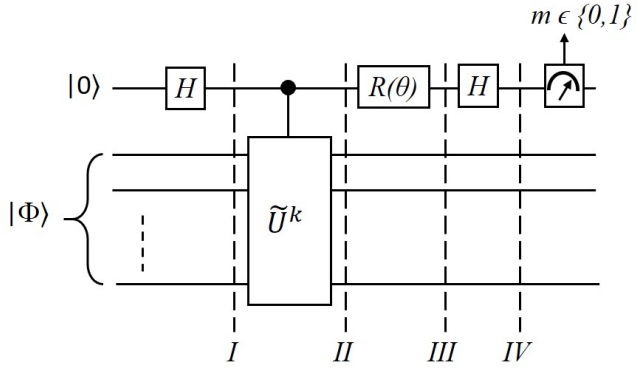


Figure 3: Basic quantum phase estimation circuit with a single ancillary qubit and an n-qubit register (initialized in state $|\Phi\rangle$). Quantum algorithms based on this basic circuit can be used to determine the phases of the eigenvalues of the unitary U efficiently. This is done by applying the unitary U on the n-qubit register a varying number of k times (conditioned on the ancillary qubit) and measuring the resulting state of the ancillary qubit (in the standard basis) sufficiently many times for each k. The single-qubit H gates denote Hadamard gates.

If one would be able to perform the experiment corresponding to the circuit depicted in Figure 3 many times for a particular $k \in \mathbb{N}$ (half of the times for $\theta = 0$ and half of the times for $\theta = \frac{\pi}{2}$), the true values of $Pr\big(m = 0 \mid k, \theta = \{0, \frac{\pi}{2}\}\big)$ can be approximately obtained. Thereby, the signal $\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle$ can be approximately obtained as well. Suppose that for $\theta = 0$ and a given k, the experiment is repeated $|\Sigma|$ times to obtain a set Σ of independent realizations of the ancillary-qubit state to be measured. The expectation value of the measurement outcome and its squared value are (using the expressions from eq. 68):

$$\langle m\rangle = \sum_m Pr(m \mid k, \theta = 0)\, m = Pr(m = 1 \mid k, \theta = 0) = \frac{1}{2} - \frac{1}{2} Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big), \tag{69a}$$

$$\langle m^2\rangle = \sum_m Pr(m \mid k, \theta = 0)\, m^2 = Pr(m = 1 \mid k, \theta = 0) = \frac{1}{2} - \frac{1}{2} Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big). \tag{69b}$$

$Pr(m = 0|k, \theta = 0)$ thus equals $1 - \langle m\rangle$, and its approximate value is obtained by sampling the expectation value $\langle m\rangle$. The variance of the measurement outcome m is:

$$Var(m) = \langle m^2\rangle - \langle m\rangle^2 = \frac{1}{4} - \frac{1}{4}\Big(Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)\Big)^2. \tag{70}$$

The sampled expectation value of m, denoted by $\tilde{m}$, is determined by averaging the measurement outcome m over $|\Sigma|$ independent realizations and therefore its variance is given by:

$$Var(\tilde{m}) = \frac{Var(m)}{|\Sigma|} = \frac{1}{4|\Sigma|}\left(1 - \Big(Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)\Big)^2\right). \tag{71}$$

From eq. 68, $Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)$ equals $2Pr(m = 0 \mid k, \theta = 0) - 1$, and thus $Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}$ equals $1 - 2\tilde{m}$. The variance of the sampled quantity $Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}$ is thus:

$$Var\Big(Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}\Big) = Var(1 - 2\tilde{m}) = 4Var(\tilde{m}) = \frac{1}{|\Sigma|}\left(1 - \Big(Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)\Big)^2\right). \tag{72}$$

Using the fact that $Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big) \in \mathbb{R}$, we infer that $Var\Big(Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}\Big) \leq \frac{1}{|\Sigma|}$. An equivalent analysis for the $(\theta = \frac{\pi}{2})$-case yields $Var\Big(Im\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}\Big) = \frac{1}{|\Sigma|}\left(1 - \Big(Im\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)\Big)^2\right) \leq \frac{1}{|\Sigma|}$. Suppose that the number of samples for both the $(\theta = 0)$-case and the $(\theta = \frac{\pi}{2})$-case is taken to be $|\Sigma| = \mathcal{N}\epsilon^{-2}$, then by means of Chebyshev's inequality (see Lemma I.2):[2]

$$Pr\left(\Big|Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled} - Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)\Big| \leq \epsilon\right) \geq 1 - \frac{1}{\mathcal{N}}, \tag{73a}$$

$$Pr\left(\Big|Im\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled} - Im\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)\Big| \leq \epsilon\right) \geq 1 - \frac{1}{\mathcal{N}}. \tag{73b}$$

We note that these last upper bounds for the variance of the real- and imaginary parts of the signal at a given k disregard the time dependence of the variance (i.e. the dependence of the variance on variable k): Although the variance of $Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}$ and $Im\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}$ is indeed upper bounded by $1/|\Sigma|$ (which is a time-independent quantity), it oscillates within this time-independent bound between 0 and $1/|\Sigma|$. In particular, the variance of $Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}$ and that of $Im\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}$ attain their maximum value when $Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)$ and $Im\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)$ are respectively equal to zero, and their minimum value (zero variance) when their norm equals unity.[3]

We now consider the variance of the estimates of the (complex-valued) signal of interest:

$$\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle_{sampled} = Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled} + i\, Im\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)_{sampled}. \tag{74}$$

[2] We note that since $m \in \{0, 1\}$ here, one could use the Chernoff-Hoeffding inequality as well (leading to a better confidence of $1 - \exp(-\mathcal{N})$ instead of $1 - 1/\mathcal{N}$). However, Chebyshev's inequality does suffice for the current discussion.

[3] Using eq. 68, one can intuitively infer that when the norm of $Re\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)$ or $Im\big(\,\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle\,\big)$ equals unity the variance vanishes: At those points, $Pr(m = 0 \mid k, \theta = \{0, \frac{\pi}{2}\})$ equals 0 or 1, which obviously implies that the sampled probability equals the exact probability, hence the absence of sampling noise.

Since $Re\big(\langle\Phi|\tilde{U}^k|\Phi\rangle\big)_{sampled}$, $Im\big(\langle\Phi|\tilde{U}^k|\Phi\rangle\big)_{sampled} \in \mathbb{R}$, the variance of $\langle\Phi|\tilde{U}^k|\Phi\rangle_{sampled}$ is given by:

$$\begin{aligned} Var\Big(\langle\Phi|\tilde{U}^k|\Phi\rangle_{sampled}\Big) &= Var\Big(Re\big(\langle\Phi|\tilde{U}^k|\Phi\rangle\big)_{sampled}\Big) + Var\Big(Im\big(\langle\Phi|\tilde{U}^k|\Phi\rangle\big)_{sampled}\Big) \\ &= \frac{1}{|\Sigma|}\Big(1 - \Big(Re\big(\langle\Phi|\tilde{U}^k|\Phi\rangle\big)\Big)^2\Big) + \frac{1}{|\Sigma|}\Big(1 - \Big(Im\big(\langle\Phi|\tilde{U}^k|\Phi\rangle\big)\Big)^2\Big) \\ &= \frac{1}{|\Sigma|}\Big(2 - \Big(Re\big(\langle\Phi|\tilde{U}^k|\Phi\rangle\big)\Big)^2 - \Big(Im\big(\langle\Phi|\tilde{U}^k|\Phi\rangle\big)\Big)^2\Big) \\ &= \frac{1}{|\Sigma|}\Big(2 - \big|\langle\Phi|\tilde{U}^k|\Phi\rangle\big|^2\Big) \end{aligned} \tag{75}$$

To bound the variance of $\langle\Phi|\tilde{U}^k|\Phi\rangle_{sampled}$, we first bound $\big|\langle\Phi|\tilde{U}^k|\Phi\rangle\big|^2$: The Cauchy-Schwarz inequality for inner products and property 4 of the *Induced Matrix Norm* (introduced in Section I.3 Preliminaries) yield:

$$\big|\langle\Phi|\tilde{U}^k|\Phi\rangle\big| \leq \underbrace{\big|\big|\langle\Phi|\big|\big|}_{=1}\,\big|\big|\tilde{U}^k|\Phi\rangle\big|\big| \leq \big|\big|\tilde{U}^k\big|\big|\,\underbrace{\big|\big||\Phi\rangle\big|\big|}_{=1} \quad\longrightarrow\quad \big|\langle\Phi|\tilde{U}^k|\Phi\rangle\big|^2 \leq \big|\big|\tilde{U}^k\big|\big|^2. \tag{76}$$

The quantity $\tilde{U}^k$ corresponds to a product of unitary matrices ($\tilde{U}$ is a unitary matrix as well). A product of unitary matrices is itself a unitary. Since the matrix norm of a unitary matrix equals unity, we find the following bounds on $\big|\langle\Phi|\tilde{U}^k|\Phi\rangle\big|^2$:

$$0 \leq \big|\langle\Phi|\tilde{U}^k|\Phi\rangle\big|^2 \leq 1. \tag{77}$$

This yields the following bounds for the variance of $\langle\Phi|\tilde{U}^k|\Phi\rangle_{sampled}$:

$$\boxed{\frac{1}{|\Sigma|} \leq Var\Big(\langle\Phi|\tilde{U}^k|\Phi\rangle_{sampled}\Big) \leq \frac{2}{|\Sigma|}.} \tag{78}$$

Assuming that the Trotter error is efficiently suppressed, one can also determine the behaviour of the variance of $\langle\Phi|\tilde{U}^k|\Phi\rangle_{sampled} = \langle\Phi|U^k|\Phi\rangle_{sampled}$ (within the aforementioned bounds) as a function of $t = \Delta t\, k$ using the spectral decomposition of H:

$$\begin{aligned} Var\Big(\langle\Phi|\tilde{U}^k|\Phi\rangle_{sampled}\Big) &= \frac{1}{|\Sigma|}\Big(2 - \Big|\sum_j e^{-iE_j t}\big|\langle\psi_j|\Phi\rangle\big|^2\Big|^2\Big) \\ &= \frac{1}{|\Sigma|}\Big(2 - \sum_{j,k} e^{-i(E_j-E_k)t}\big|\langle\psi_j|\Phi\rangle\big|^2\,\big|\langle\psi_k|\Phi\rangle\big|^2\Big) \\ &= \frac{1}{|\Sigma|}\Big(2 - \sum_j \big|\langle\psi_j|\Phi\rangle\big|^4 - 2\sum_{j<k}\cos\big((E_j-E_k)t\big)\big|\langle\psi_j|\Phi\rangle\big|^2\,\big|\langle\psi_k|\Phi\rangle\big|^2\Big). \end{aligned} \tag{79}$$

The variance of $\langle\Phi|\tilde{U}^k|\Phi\rangle_{sampled}$ is thus a function that is lower bounded by $\frac{1}{|\Sigma|}$ and upper bounded by $\frac{2}{|\Sigma|}$ and oscillates in between these bounds with frequencies that depend on the differences between eigenvalues of H. At $t = 0$, the minimum at $\frac{1}{|\Sigma|}$ is attained and the other values at which the lower and upper bounds are reached depend on the spectrum of H (which is the quantity to be determined). We note that if (for some $a \in \{0, 1, ..., 2^n - 1\}$);

$$\big|\langle\psi_j|\Phi\rangle\big|^2 \approx \begin{cases} 1, & \text{for } j = a, \\ 0, & \text{for } j \neq a, \end{cases} \tag{80}$$

i.e. $|\Phi\rangle$ is chosen such that the signal is dominated by a single eigenvalue of H, then the variance of $\langle\Phi|\tilde{U}^k|\Phi\rangle_{sampled}$ approximately equals $\frac{1}{|\Sigma|}$ for all t. If $|\Phi\rangle$ is chosen such that two eigenvalues of H dominate the signal to equal extents (for some $a, b \in \{0, 1, ..., 2^n - 1\}$);

$$\big|\langle\psi_j|\Phi\rangle\big|^2 \approx \begin{cases} \frac{1}{2}, & \text{for } j = a, \\ \frac{1}{2}, & \text{for } j = b, \\ 0, & \text{for } j \neq a, b, \end{cases} \tag{81}$$

then the variance of $\langle\Phi|\,\tilde{U}^k\,|\Phi\rangle_{sampled}$ approximately oscillates between $\frac{1}{|\Sigma|}$ and $\frac{2}{|\Sigma|}$ with a single frequency equal to $(E_b - E_a)$. We shall demonstrate this numerically as well in later stages of this report.

For the Monte Carlo scheme presented later in this report, a similar region of low variance is located at $t = 0$. However, in that case the variance of the signal of interest goes to zero at $t = 0$ (instead of going to $\frac{1}{|\Sigma|}$ such as the QPE signal). These regions of relatively low variance might be exploited to obtain higher-quality estimates of the eigenvalues of H to be determined.

We repeat that the process of measuring the ancillary-qubit a total of $2|\Sigma|$ times (for a given k) for many values of $k \in \{0, 1, ..., K\}$ produces a noisy and Trotterized version of the signal $g(k) = \langle\Phi|\,e^{-iH\Delta tk}\,|\Phi\rangle = \langle\Phi|\,U^k\,|\Phi\rangle$ (where the noise originates from the finite sampling of the ancillary-qubit state). When the quantum-phase-estimation scheme is run on e.g. a non-error-corrected NISQ device, the variable K is limited by the coherence times of the qubits involved and the fact that the implemented gates are noisy. The spectral decomposition of Hamiltonian H yields the following expression for $g(k)$:

$$g(k) = \sum_j e^{-iE_j\Delta tk}|\langle\psi_j|\Phi\rangle|^2. \tag{82}$$

By analyzing the signal $g(k)$, and in particular its oscillation frequencies as a function of $t = \Delta t\, k$, (a subset of) the eigenvalues of Hamiltonian H can be obtained. There are several methods for solving this problem, such as *the Discrete Fourier Transform (DFT)*, *Prony's method* ([26]) and the *Matrix Pencil Method* ([27],[28],[29]). The Matrix Pencil Method can, in principle, be regarded as a generalization of (amongst others) Prony's method. In this report, we consider the Matrix Pencil Method because of its efficiency and noise-tolerance ([27]).

Post-processing methods such as the Matrix Pencil Method in particular determine the quantities $e^{-iE_j\Delta t}$ instead of the eigenvalues E_j directly. Therefore, one is only able to potentially determine a subset $S = \{E_j \mid E_j\Delta t \in [0, 2\pi) \wedge |\langle\psi_j|\Phi\rangle|^2 \geq 1/poly(n)\}$ of the eigenvalues $\{E_j\}_j$. By carefully tweaking Δt and $|\Phi\rangle$, one can thus obtain the desired subset of eigenvalues. In the next chapter, it will be shown that subset S is at most of size $poly(n)$, and for two E_j's to be distinguishable, the gap between their associated $E_j\Delta t$'s cannot be too small in the presence of noise.

We note that the eigenvalues $e^{-iE_j\Delta t}$ of unitary U, as well as the eigenvalues of Trotterized unitary $\tilde{U}$, have unit norm. However, due to finite sampling, one determines a noisy version of the signal $g(k)$. This noise results in the fact that the apparent eigenvalues of $\tilde{U}$ might have norms that slightly deviate from unity (and therefore $\tilde{U}$ might seem to be slightly non-unitary). To ensure that the output estimates for the eigenvalues E_j are real (as they should be due to the Hermiticity of H), we take them to be the real parts of the expression $i\frac{log(z_j)}{\Delta t}$ (where z_j denotes the apparent eigenvalue of $\tilde{U}$).

We conclude this section by noting that the picture of quantum phase estimation that has been sketched here is a simplified one for the following reasons:

1. The string of gates produced by the Trotter decomposition of unitary U ($\tilde{U} \equiv \left(\mathcal{T}_M(p, \Delta t)\right)^M$) is assumed to act on the n-qubit register in a noiseless and error-free manner.

2. All qubits involved are assumed to remain coherent throughout single experiment rounds. As was mentioned before, in reality the limited coherence time would severely limit the value for K, and thereby the time interval over which the state dynamics can be tracked.

3. The outcome of the measurements of the ancillary-qubit state are assumed to be completely trustworthy.

4. The state $|\Phi\rangle$ is assumed to be prepared on an n-qubit register using a circuit of $poly(n)$ size. As was shown in the previous section, this efficient preparation is not always feasible for any n-qubit state.

We stress that the discussion of quantum-phase-estimation-based quantum simulation in this section is *not* limited to stoquastic Hamiltonians, and generally applies to any sufficiently local Hamiltonian.

III Signal Analysis

Quantum phase estimation can thus be used to approximately obtain a signal of the form $\langle\Phi| e^{-iHt} |\Phi\rangle$ – i.e. the overlap between state $|\Phi\rangle$ and the state that has evolved over time t (starting from $|\Phi\rangle$). Snapshots of this signal are obtained at times $k\Delta t$. The spectral decomposition of Hamiltonian H yields the following expression for this signal:

$$g(k) = \sum_{j=1}^{2^n} \left|\langle\psi_j|\Phi\rangle\right|^2 \left(e^{-iE_j\Delta t}\right)^k. \tag{83}$$

Given a noisy version of the signal $g(k)$ at $K \leq poly(n)$ points ($k \in \{0, 1, ..., K-1\}$), the task is to obtain a subset of the (possibly degenerate) eigenvalues of H, which correspond to the oscillation frequencies of $g(k)$ in time.

In Chapter IV of this report, we introduce a Monte Carlo scheme that can be used to efficiently obtain, up to an ϵ-additive error, a noisy version of the signal $\langle\Phi| e^{-\tau H} |\Phi\rangle$ ($\tau \in \mathbb{R}_+$), provided that H is piecewise stoquastic and consists of sufficiently local interactions. The signal $g(k)$ in this case is given by:

$$g(k) = \sum_{j=1}^{2^n} \left|\langle\psi_j|\Phi\rangle\right|^2 \left(e^{-E_j\Delta\tau}\right)^k. \tag{84}$$

Given a noisy version of $g(k)$ at $K \leq poly(n)$ data points ($k \in \{0, 1, ..., K-1\}$), the task is again to obtain the (possibly degenerate) eigenvalues of H, which now correspond to the decay rates of $g(k)$ in time.

There are several methods for obtaining a subset of the E_j's from the signal $g(k)$, such as *the Z-Transform (Discrete-Time Laplace Transform)*, *Prony's method* ([26]) and *the Matrix Pencil Method* ([27],[28]). In this report, we consider the Matrix Pencil Method because of its noise-tolerance ([27]).

We define $z_j \equiv e^{-iE_j\Delta t}$ in the setting of QPE, and $z_j \equiv e^{-\Delta\tau E_j}$ for the Monte Carlo algorithm. The Matrix Pencil Method determines a subset of these z_j's. For QPE, the z_j's lies along the unit circle. As mentioned in the previous chapter, for the sake of distinguishability of the eigenvalues E_j in a particular subset the associated $E_j\Delta t$'s must lie in the interval $[0, 2\pi)$. For the Monte Carlo algorithm, all z_j's lie in the interval $(0, 1]$ (provided that the lowest eigenvalue of H is non-negative). This is illustrated in Figure 4.

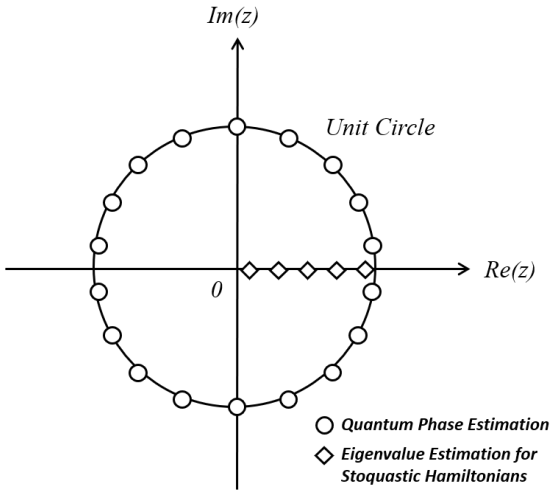


Figure 4: Illustration of the position of the z_j's in the complex plane in the case of real-time Hamiltonian propagation and in the case of imaginary-time Hamiltonian propagation. The eigenvalues of the (unitary) propagation operator of the Schrödinger equation e^{-iHt} lie on the complex unit circle. The eigenvalues of the imaginary-time propagation operator $e^{-\tau H}$ ($\tau \in \mathbb{R}_+$) lie on the positive real axis on the interval $(0, 1]$ (when the lowest eigenvalue of H is set to zero, or at least set to be positive).

III.1 Matrix-Pencil Method

We reiterate that the task for which we make use of the Matrix Pencil Method ([27],[28],[29]) can be formulated as follows: Suppose we are given a set of noisy values of a signal $g(k)$,

$$g(k) = \sum_{j=1}^{R} c_j e^{\lambda_j k} = \sum_{j=1}^{R} c_j z_j^k, \tag{85}$$

where $z_j \equiv e^{\lambda_j}$, then determine (a subset of) the parameters $\{\lambda_j\}$ (and possible of $\{c_j\}$). The number of parameters λ_j to be determined is R. In the case of the Monte Carlo scheme and the QPE scheme we have $\lambda_j = -E_j \Delta\tau$ ($\in \mathbb{R}$) and $\lambda_j = -iE_j \Delta t$ ($\in \mathbb{C}$), respectively.

The basic idea comprises of setting up a generalized eigenvalue problem, the solution of which is given by the set $\{z_j\}$. One is effectively given a noisy signal $y(k)$ (for $k \in \{0, 1, ..., K-1\}$), which is related to the original signal $g(k)$ by:

$$y(k) = g(k) + \eta(k) = \sum_{j=1}^{R} c_j z_j^k + \eta(k), \tag{86}$$

where $\eta(k)$ denotes e.g. the sampling noise. We introduce the following matrix Y, containing all K data points of the noisy signal $y(k)$ (for $k \in \{0, 1, ..., K-1\}$) and the *pencil parameter* L (whose function will become apparent shortly):

$$\begin{aligned} Y &= \begin{pmatrix} y(0) & y(1) & \dots & y(L) \\ y(1) & y(2) & \dots & y(L+1) \\ \vdots & \vdots & & \vdots \\ y(K-L-1) & y(K-L) & \dots & y(K-1) \end{pmatrix}_{(K-L)\times(L+1)} \\ &= \sum_{j=1}^{R} c_j \begin{pmatrix} 1 & z_j & \dots & z_j^L \\ z_j & z_j^2 & \dots & z_j^{L+1} \\ \vdots & \vdots & & \vdots \\ z_j^{K-L-1} & z_j^{K-L} & \dots & z_j^{K-1} \end{pmatrix}. \end{aligned} \tag{87}$$

Matrix Y thus equals the sum of R rank-1 matrices. The rank of a matrix $\mathcal{A} = \sum_j A_j$ (where each A_j has the same dimensions) is at most the sum of the ranks of the A_j matrices – i.e. $rank(\mathcal{A}) \leq \sum_j rank(A_j)$. Therefore, $rank(Y) \leq R$.

In principle, taking $K \geq 2R$ will suffice to extract the R values of interest $\{z_j\}$ from signal $g(k)$ (i.e. the noiseless signal). However, since the signal $y(k)$ is noise-contaminated, one must generally take $K \gg R$ to ensure a high quality of the estimates of $\{z_j\}$.

Noiseless Signal: We first consider the noiseless case: $\eta(k) = 0$ ($\forall k$). The set $\{z_j\}$ now corresponds to the set of generalized eigenvalues of the matrix pair $\{Y_1, Y_2\}$, where Y_1 and Y_2 are the matrices that are obtained by respectively deleting the last column and the first column of matrix Y (we note that Y_1 and Y_2 are square matrices for $L = K/2$). To illustrate this, we consider the representation of the matrices Y_1 and Y_2 in terms of the Vandermonde Matrices (Z_1, Z_2), which contain the *unique* z_j's to be determined:

$$Y_1 = Z_1 C Z_2 \; ; \; Y_2 = Z_1 C Z_0 Z_2, \tag{88}$$

where the matrices Z_0, Z_1, Z_2 and C are defined as follows:

$$Z_1 \equiv \begin{pmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_R \\ \vdots & \vdots & & \vdots \\ z_1^{K-L-1} & z_2^{K-L-1} & \dots & z_R^{K-L-1} \end{pmatrix}_{(K-L)\times R} \; ; \; Z_2 \equiv \begin{pmatrix} 1 & z_1 & \dots & z_1^{L-1} \\ 1 & z_2 & \dots & z_2^{L-1} \\ \vdots & \vdots & & \vdots \\ 1 & z_R & \dots & z_R^{L-1} \end{pmatrix}_{R\times L} \tag{89a}$$

$$Z_0 \equiv diag(z_1, z_2, ..., z_R) \; ; \quad C \equiv diag(c_1, c_2, ..., c_R). \tag{89b}$$

We consider the linear matrix pencil $Y_2 - zY_1 = Z_1 C\big(Z_0 - zI\big) Z_2$. If z equals z_j (for $j \in \{1, 2, ..., R\}$), then the jth row (and column) of $Z_0 - zI$ equals zero. In that case, the jth column of Z_1 and the jth row of Z_2 are annihilated in the expression above for the matrix pencil.

We now differentiate between three parameter regimes: $L < R$, $L > K - R$ and $R \leq L \leq K - R$. This last parameter regime is non-empty only for $K \geq 2R$.

- ***L < R:*** In this parameter regime, the rank of matrix Z_2 generically equals L. Furthermore, we have (for any L):

$$C\big(Z_0 - zI\big) Z_2 = \begin{pmatrix} c_1(z_1 - z) & c_1(z_1 - z)z_1 & \dots & c_1(z_1 - z)z_1^{L-1} \\ c_2(z_2 - z) & c_2(z_2 - z)z_2 & \dots & c_2(z_2 - z)z_2^{L-1} \\ \vdots & \vdots & & \vdots \\ c_R(z_R - z) & c_R(z_R - z)z_R & \dots & c_R(z_R - z)z_R^{L-1} \end{pmatrix}_{R \times L} . \tag{90}$$

  If z equals z_j, then the jth row of $C\big(Z_0 - zI\big) Z_2$ equals zero. However, since $L < R$, $C\big(Z_0 - zI\big) Z_2$ has full column rank irrespective of whether or not z equals any of the z_j's. The rank of $C\big(Z_0 - zI\big) Z_2$ thus equals L for any z. This implies that the rank of the matrix pencil $Y_2 - zY_1 = Z_1 C\big(Z_0 - zI\big) Z_2$ does not depend on whether or not parameter z equals any of the z_j's.

- ***L > K − R:*** In this parameter regime, the rank of matrix Z_1 generically equals $K - L$. Furthermore, we have (for any L):

$$Z_1 C\big(Z_0 - zI\big) = \begin{pmatrix} c_1(z_1 - z) & c_2(z_2 - z) & \dots & c_R(z_R - z) \\ c_1(z_1 - z)z_1 & c_2(z_2 - z)z_2 & \dots & c_R(z_R - z)z_R \\ \vdots & \vdots & & \vdots \\ c_1(z_1 - z)z_1^{K-L-1} & c_2(z_2 - z)z_2^{K-L-1} & \dots & c_R(z_R - z)z_R^{K-L-1} \end{pmatrix}_{(K-L) \times R} . \tag{91}$$

  If z equals z_j, then the jth column of $Z_1 C\big(Z_0 - zI\big)$ equals zero. However, since $L > K - R$, $Z_1 C\big(Z_0 - zI\big)$ has full row rank irrespective of whether or not z equals any of the z_j's. The rank of $Z_1 C\big(Z_0 - zI\big)$ thus equals $K - L$ for any z. This implies that the rank of the matrix pencil $Y_2 - zY_1 = Z_1 C\big(Z_0 - zI\big) Z_2$ does not depend on whether or not parameter z equals any of the z_j's.

- ***R ≤ L ≤ K − R:*** In this parameter regime, matrices Z_1 and Z_2 are both rank-R matrices. In particular, matrix Z_1 has full column rank and matrix Z_2 has full row rank. The $R \times R$ matrix $C\big(Z_0 - zI\big)$ has rank R if $z \neq z_j$ (for $j \in \{1, 2, ..., R\}$), and rank $R - 1$ if $z = z_j$ (for $j \in \{1, 2, ..., R\}$). For matrices A and B, $\text{Rank}(AB) = \text{Rank}(B)$ if A has full column rank and $\text{Rank}(AB) = \text{Rank}(A)$ if B has full row rank. Since Z_1 has full column rank and Z_2 has full row rank in this parameter regime, the rank of the matrix pencil equals:

$$\begin{aligned} \text{Rank}\big(Y_2 - zY_1\big) &= \text{Rank}\big(Z_1 C\big(Z_0 - zI\big) Z_2\big) \\ &= \text{Rank}\big(C\big(Z_0 - zI\big) Z_2\big) \\ &= \text{Rank}\big(C\big(Z_0 - zI\big)\big) \\ &= \begin{cases} R, & \text{for } z \neq z_j \ \ (\forall j \in \{1, 2, ..., R\}), \\ R - 1, & \text{for } z = z_j \ \ (\text{for any } j \in \{1, 2, ..., R\}). \end{cases} \end{aligned} \tag{92}$$

  In the current parameter regime, the rank of the matrix pencil $Y_2 - zY_1$ thus equals R if parameter z does not equal any of the z_j's and its rank is reduced to $R - 1$ if z does equal any of the z_j's. In the other parameter regimes mentioned above, the rank of the matrix pencil does not depend on whether or not z equals any of the z_j's.

Provided that $R \leq L \leq K - R$ (and thus $K \geq 2R$), the unique generalized eigenvalues of the matrix pair $\{Y_1, Y_2\}$ are given by z_j (for $j \in \{1, 2, ..., R\}$) because the z_j's reduce the rank of the matrix pencil $Y_2 - zY_1$ by one. In the remainder of this discussion, we assume the pencil parameter L to be in the regime $R \leq L \leq K - R$ unless stated otherwise.

The matrices Y_1 and Y_2 are generally not square matrices (they are only square if $L = K/2$) and not full-rank. Since a matrix is invertible if and only if it is full-rank, Y_1 and Y_2 are generally not invertible.

Generalized eigenvalue problems (*GEP*) such as $Y_2\boldsymbol{x}_j = z_j Y_1 \boldsymbol{x}_j$ encountered here are generally solved by solving the associated eigenvalue problem (*EP*) $Y_1^{-1}Y_2\boldsymbol{x}_j = z_j\boldsymbol{x}_j$ (if Y_1 were full-rank). If Y_1 (or Y_2) is indeed not full-rank, a Moore-Penrose pseudo-inverse (denoted by '$^+$') is used in the translation from *GEP* to *EP*. The Moore-Penrose pseudo-inverse of a matrix A is defined as $A^+ \equiv \left(A^\dagger A\right)^{-1} A^\dagger$. The eigenvalue problem to be solved to find the parameters z_j (for $j \in \{1, 2, ..., R\}$) in that case is thus $Y_1^+ Y_2 \boldsymbol{x}_j = z_j \boldsymbol{x}_j$. The matrix $Y_1^+ Y_2$ is an $L \times L$ matrix that can be written in terms of Z_0, Z_1, Z_2 and C as follows:

$$Y_1^+ Y_2 = \left(Z_1 C Z_2\right)^+ Z_1 C Z_0 Z_2 = Z_2^+ C^+ \underbrace{Z_1^+ Z_1}_{=I} C Z_0 Z_2 = Z_2^+ \underbrace{C^+ C}_{=I} Z_0 Z_2 = Z_2^+ Z_0 Z_2. \tag{93}$$

Since the Moore-Penrose pseudo-inverse of a matrix A has the same rank as A, and Z_2 has rank R (given that $R \leq L \leq K - R$), Z_2^+ has rank R as well. Since the $R \times L$ matrix Z_2 has full row rank, the $L \times R$ matrix Z_2^+ has full column rank. Using eq. 93, one can thus infer the rank of matrix $Y_1^+ Y_2$ to be equal to the rank of Z_0, which is equal to R ($\leq L$). Since the rank of a matrix equals its number of non-zero eigenvalues, $Y_1^+ Y_2$ has R eigenvalues that are non-zero (and are equal to the z_j's) and $L - M$ zero eigenvalues. The R *non-zero* eigenvalues that one finds when solving the eigenvalue problem associated with $Y_1^+ Y_2$ are thus equal to the z_j's (from which the λ_j's can be determined).

Figure 5 depicts the results of application of the Matrix Pencil Method to a noiseless signal. We consider separately a decaying signal and an oscillating signal, and for both cases we depict respectively the estimates of the decay rates and oscillation frequencies as a function of the number of measurement points of the signal (K). The true decay rates/oscillation frequencies (the eigenvalues E_j) have been generated at random and all the c_j's are set equal to $1/R$. The pencil parameter is set to be $L = K/2$. In the regime $K \geq 2R$, the eigenvalues are resolved both for the decaying and the oscillating signal, as expected based on the discussion in this section. In the regime $K < 2R$, the eigenvalues are not resolved and the estimates seem to be some sort of averaged value of the true eigenvalues.

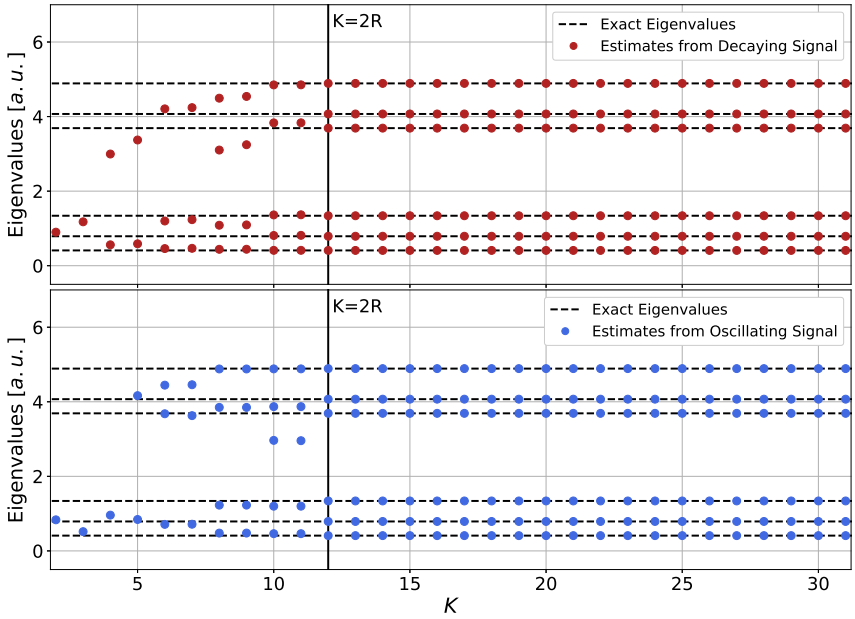


Figure 5: Estimates of the decay rates (of a decaying signal) and oscillation frequencies (of an oscillating signal) as a function of K (the number of equidistant points at which the signals are measured). The estimates are obtained from applying the Matrix Pencil Method to the noiseless signals $g(k) = \sum_{j=1}^{R} c_j z_j^k$, where $z_j = -E_j \Delta\tau$ (in the case of the decaying signal) and $z_j = -iE_j\Delta t$ (in the case of the oscillating signal). All c_j's are set equal to $1/R$ and the E_j's have been produced at random. The time interval over which the signals are analyzed is 3 (the values of $\Delta\tau$ and Δt are both equal to $3/K$). The pencil parameter is set to be $L = K/2$.

Noisy Signal: In the presence of (sampling) noise, an additional pre-filtering step is required to obtain the estimates of the parameters z_j. This includes carrying out a singular-value decomposition (SVD) of the data matrix Y:

$$Y = U\,\Sigma\,V^{\dagger}, \tag{94}$$

where Σ is a diagonal matrix containing all singular values ($\{\sigma\}$) of Y, and U and V are unitary matrices that are composed of the eigenvectors of $Y\,Y^{\dagger}$ and $Y^{\dagger}\,Y$, respectively.

The data matrix Y is filtered by truncating all singular values for which $\frac{\sigma}{\sigma_{max}} < TF$, where TF is a variable threshold (the truncation factor) and σ_{max} denotes the largest singular value. The R non-truncated (dominant) singular values are used to construct the diagonal matrix Σ'. The left-singular and right-singular eigenvectors associated with these truncated singular values are respectively deleted from matrices U and V to obtain matrices U' and V'. The new data matrix $Y' = U'\,\Sigma'\,\left(V'\right)^{\dagger}$ has now been filtered to reduce the effects of noise. As in the noiseless signal case, the sub-matrices Y_1' and Y_2' are obtained from Y' by deleting the last column and the first column, respectively. To obtain estimates of the parameters z_j, the eigenvalue problem $\left(Y_1'\right)^{+} Y_2' \boldsymbol{x}_j = z_j \boldsymbol{x}_j$ should now be solved.

The pencil parameter L is one of the variables governing the noise-tolerance of the method and should ideally be taken to be $K/3 < L < K/2$ ([27],[30]). For these values of L, the variance in the estimates of the parameters z_j has been found to be minimum in [27] and [30]. Unless indicated otherwise, we shall take $L = K/2$ in the remainder of this report.

After having obtained the estimates of the parameters $\{z_j\}$, one can estimate the parameters $\{c_j\}$ as well: Using the noise-contaminated signal $y(k)$ and the estimates of $\{z_j\}$, one finds the approximate values of $\{c_j\}$ by solving the least-squares minimization problem $\min_{c_j} \sum_k \left|y(k) - \sum_j c_j \left(z_j\right)^k\right|^2$. The estimates of $\{z_j\}$ and $\{c_j\}$ together form the reconstructed signal which is less noise-contaminated than the original signal $y(k)$.

We furthermore note that in the Matrix Pencil Method described here, R is essentially an emergent quantity depending on how many singular values are larger than $TF * \sigma_{max}$ in the SVD of the data matrix Y. This is done since in principle one does not know a priori with how many eigenstates of the Hamiltonian the state $|\Phi\rangle$ has a significant overlap. In contrast, in [30] the value of R is fixed due to a change in the SVD truncation protocol: There, in any instance, only the largest R singular values (where R is a fixed number) of the SVD of the data matrix Y are kept, while all the other singular values are discarded.

We have discussed above that in a noiseless-signal setting, all R eigenvalues can be exactly obtained (for a proper choice of the pencil parameter) provided that $K \geq 2R$, irrespective of whether the signal is oscillating or decaying. We have presented a well-known pre-filtering method to reduce the effects of noise in the case of a noisy signal, and have mentioned that in that case one should take $K \gg 2R$. One might wonder, however, how large the magnitude of the noise can be for the eigenvalues to be accurately extracted from the noisy signal. In [31], it has been shown that the ability to extract all R eigenvalues from a signal that is purely oscillatory by means of the Matrix Pencil Method shows a sharp transition as a function of K. Specifically, the relation between K and the minimum gap between the eigenvalues Δ is relevant. That is, the following is proved in [31]: *Given that $\lambda_j \in [0, 2\pi)$, $\forall j$. If $K > 1/\Delta + 1$, then there exists a polynomial time algorithm that provides estimates of the c_j's and λ_j's, where these estimates converge to the exact values at an inverse polynomial rate in terms of the magnitude of the noise $||\eta|| = \sum_k |\eta(k)|^2$.* This statement is tight in the following sense: *If $K < (1-\epsilon)/\Delta$, then there exists two sets $\{c_j\}$,$\{\lambda_j\}$ and $\{c_j\}'$,$\{\lambda_j\}'$ (each with minimum gap Δ) where one would need $||\eta|| \leq exp(-\epsilon R)$ (i.e. noise that is exponentially small in magnitude) in order to tell these two sets apart.* Both of these statements are based on an analysis of the condition number of the Vandermonde matrices Z_1 and Z_2. We reiterate that it was shown in [31] that these claims hold for a purely oscillating signal. They are therefore applicable to the quantum-phase-estimation setting. It is an open question whether similar statements hold for purely decaying – i.e. Monte Carlo – signals. This will thus be an interesting and relevant point of future research.

In the settings considered in this report, the signal $g(k)$ is a sum of generally 2^n components, where each (possible degenerate) component is associated with an eigenvalue of the Hamiltonian. Therefore one would naively take R to be approximately 2^n. Since one generally requires $K \geq 2R$ – and even $K \gg 2R$ in the presence of noise – and $K \leq poly(n)$ must hold for the algorithm to run efficiently, the effective number of components that can be extracted from the signal will be no more than $poly(n)$ (and can therefore not be

2^n). There are two ways in which the number of components in the signal $g(k)$ can indeed be reduced to be $poly(n)$ instead of $exp(n)$: One can reside in a parameter regime in which the degeneracies of Hamiltonian H are such that the number of distinct eigenvalues of H are $poly(n)$ (if this parameter regime exists). This is e.g. the case when the eigenvalues of a Hamiltonian form bands which are separated by gaps, and one wishes to find an estimate of each band. In addition, one has the freedom to choose the state $|\Phi\rangle$: The coefficient c_j of component j equals $|\langle\psi_j|\Phi\rangle|^2$. Since one takes $|\Phi\rangle$ to be a state that has relatively large overlap with some set of eigenstates of interest and normalization requires $\sum_j |\langle\psi_j|\Phi\rangle|^2 = 1$, the coefficients c_j for the vast majority of components in the signal will be very small. One can thus attempt to choose $|\Phi\rangle$ such that the effective number of components in the signal will be $poly(n)$.

IV Monte Carlo Scheme for Local and Stoquastic Hamiltonians

In this chapter, we examine whether the decaying signal (i.e. the signal $g(k)$ that was introduced in Chapter III for $\Delta t = 0$ and $\Delta\tau$ non-zero), from which the eigenenergies of H are to be estimated by means of the Matrix Pencil Method, can be obtained efficiently by classical means for Hamiltonians that are sufficiently local and piece-wise stoquastic. This is done by first examining whether a single matrix element of the Gibbs density matrix associated with these Hamiltonians can be efficiently obtained by classical means. The discussion of the estimation of the single Gibbs density matrix element is then generalized to the estimation of the aforementioned signal of interest. We discuss the errors that arise when estimating this signal and the efforts one can undertake to suppress them. We additionally examine whether the partition function associated with these Hamiltonians can be classically determined in a computationally tractable manner. We provide an overview of the chapter near its end and in addition discuss how the algorithms presented in this chapter might be useful in a quantum computing setting as well.

IV.1 Efficiently determining single matrix elements of the Gibbs density matrix

We first analyze whether the quantity $\langle x| e^{-\tau H} |x'\rangle$ (where $\tau \in \mathbb{R}_+$, and $|x\rangle$, $|x'\rangle$ are (bit-string) basis states of a basis in which H is piece-wise stoquastic) can be obtained classically in an efficient way. We note that this quantity equals a single element of the Gibbs density matrix associated with Hamiltonian H (which is the Hamiltonian of which one wishes to determine the eigenvalues) at inverse temperature $\beta = \tau$. The system of which H is the Hamiltonian consists of n degrees of freedom (that are here taken to be qubits). We suppose that H is local (see Definition I.1). The *locality* of the H_i's is denoted by k. We stress that these H_i's, and their localities (which are all $log(poly(n))$ at most), generally do not have to be equivalent for all i: The interactions between different degrees of freedom need not be translationally invariant for example.

We require the decomposition of $e^{-\tau H}$ into a string of local propagation operators. Since generally $[H_i, H_j] \neq 0$ (for $i \neq j$), we will need to employ a Trotterization scheme. The decomposition, by means of Trotterization, of $e^{-\tau H}$ into an ordered string of local propagation operators $e^{-a_l \tau/M\, H_i}$ (where $a_l \in (0, 1]$ depends on the Trotterization scheme, and $a_l = 1\ \forall l$ if $p = 1$) has been discussed extensively in Section I.4. In the current discussion, we take for granted a non-zero Trotter error due to a finite Trotter variable M and note that this Trotter error is generally suppressible with polynomial resources (see Section I.4 of this report). The length of the ordered string of local propagation operators is denoted by L and equals MN for $p = 1$, and $2MN\, 5^{\frac{p}{2}-1}$ for $p > 1$ (when using the scheme introduced in [17]).

We shall denote each of the local propagation operators $e^{-a_l \tau/M\, H_i}$ by G_l (where $l \in \{1, ..., L\}$). $e^{-\tau H}$ is thus represented by $\prod_{l=1}^{L} G_l$, which is obtained through a Trotterization scheme of a given order p and for a given Trotter variable M. If H is piece-wise stoquastic, then each of the propagation operators G_l will be element-wise non-negative (since $\tau \in \mathbb{R}_+$).

In general, each G_l is reducible (see Definition I.5). However, it consists of B_l (for $1 \leq B_l \leq poly(n)$) element-wise non-negative sub-matrices – which each act on distinct sets of basis states – that are themselves *irreducible*. We denote these non-negative and irreducible sub-matrices by G_l^b, where $b \in \{1, ..., B_l\}$ and $G_l = \oplus_{b=1}^{B_l} G_l^b$. The set of bit string basis states on which the irreducible sub-matrix G_l^b acts is denoted by S_l^b, where $\cup_b S_l^b \subseteq \{0, 1\}^n$ (note that G_l can also have a non-zero null-space). For these non-negative and irreducible matrices G_l^b, the following Perron-Frobenius Theorem holds (see Theorem 8.4.4 in [32]):

Theorem IV.1. Perron-Frobenius Theorem. *Suppose $G_l^b \in \mathbb{R}^{h \times h}$ and all elements $\left(G_l^b\right)_{i,j} \geq 0$ (i.e. G_l^b is an element-wise non-negative matrix). Additionally, G_l^b is irreducible, then:*

1. *There is at least one eigenvalue λ_l^b of G_l^b that is real and non-negative. All other eigenvalues of G_l^b are smaller than or equal to λ_l^b in magnitude.*
2. *There exists an associated unique and strictly positive eigenstate $|\phi\rangle_l^b$ ($= \sum_{x \in S_l^b} \phi_l^b(x) |x\rangle$, where $\phi_l^b(x) > 0$, $\forall x$) of G_l^b, for which $G_l^b |\phi\rangle_l^b = \lambda_l^b |\phi\rangle_l^b$.*

Note that since G_l acts non-trivially on at most $log(poly(n))$ qubits, the maximum eigenvalue λ_l^b and the corresponding eigenstate $|\phi\rangle_l^b = \sum_{x \in S_l^b} \phi_l^b(x) |x\rangle$ of the square (sub-)matrix G_l^b are efficiently computable by classical means.

We can set the lowest eigenvalue of all H_i's to be zero ($\lambda_{min}(H_i) = 0$), and thus the corresponding largest eigenvalue of the non-negative G_l's to 1. We therefore conclude that the eigenvalues of G_l must all be either smaller than or equal to unity (according to the Perron-Frobenius Theorem). G_l is Hermitian (since H_i is Hermitian $\forall i$ and $\tau \in \mathbb{R}$) and therefore its eigenvalues are additionally real (in the current setting, G_l is even symmetric since its entries are real-valued). Furthermore, since $G_l = e^{-a_l \tau H_i}$ (and $\tau \in \mathbb{R}_+$, $a_l \in (0, 1]$), all eigenvalues of G_l are larger than 0. G_l is thus a positive definite matrix and in particular its eigenvalues lie in the interval $(0, 1]$, $\forall l$. We additionally note that since G_l^b denote the aforementioned sub-matrices of G_l and G_l itself is Hermitian, G_l^b is Hermitian as well ($\forall b$).

Using the fact that G_l can be decomposed into sub-matrices G_l^b which each act on distinct sets of basis sets, we state the following property of the eigenvalues of the G_l^b's in terms of those of G_l: *The set of eigenvalues of G_l^b is a subset of the set of eigenvalues of G_l: $\{\omega_l^b\} \subseteq \{\omega_l\}$, $\forall b$. Therefore, $\omega_{min}(G_l) \leq \omega_{max}(G_l^b) \leq \omega_{max}(G_l)$.* For the specific choice of G_l here, we thus conclude that $0 < \lambda_l^b \leq 1$ (since $\omega_{max}(G_l^b) = \lambda_l^b$), $\forall b$. This turns out to be a useful property later on.
We now present the following theorem (inspired by unpublished work by S. Bravyi), and provide its proof afterwards:

Theorem IV.2. (S. Bravyi, M.E. Stroeks, B.M. Terhal). *Suppose $|x\rangle$ (where $x \in \{0,1\}^n$) denotes the state of an n-qubit register ($\{|x\rangle\}$ is an orthonormal set). Furthermore, suppose $f(\tau) = \langle x| e^{-\tau H} |x'\rangle = \langle x| G_1 G_2 ... G_L |x'\rangle$, where:*

1. *$\tau \in \mathbb{R}_+$,*
2. *$G_l = \left(e^{-a_l \tau / M H_i}\right)_j$ is a Hermitian matrix, where $i \in \{1, ..., N\}$, $j \in \{1, ..., M\}$, $a_l \in (0, 1]$ ($\{a_l\}_{l=0}^L$ depends on the Trotterization scheme) and $l \in \{1, ..., L\}$ labels the time slice,*
3. *$H = \sum_{i=1}^N H_i$, where each H_i acts non-trivially on at most $log(poly(n))$, or typically $O(1)$, qubits. H_i is Hermitian and stoquastic in the 'x'-basis, $\forall i$ (hence G_l is element-wise non-negative and Hermitian, $\forall l$),*
4. *$\lambda_{min}(H_i) = 0$ ($\forall i$) so that $\omega_{max}(G_l) = 1$ ($\forall l$), where 'ω' denotes an eigenvalue of G_l.*

Then $f(\tau)$ can be estimated with absolute error ϵ in time $poly(\epsilon^{-1}, n, L)$. The algorithm runs efficiently for $\epsilon \geq \frac{1}{poly(n)}$ and $L \leq poly(n)$.

Proof. The proof of Theorem IV.2 consists of two steps: To construct an estimator for $f(\tau)$ and to show that the absolute error of this estimator can be bounded according to the theorem.

By inserting $L - 1$ complete sets of basis states in between the G_l operators, we can express $f(\tau)$ as follows:

$$f(\tau) = \sum_{x_1, x_2, ..., x_{L-1}} \langle x_0| G_1 |x_1\rangle \langle x_1| G_2 |x_2\rangle \ ... \ \langle x_{L-1}| G_L |x_L\rangle , \tag{95}$$

where $|x_0\rangle = |x\rangle$ and $|x_L\rangle = |x'\rangle$ are fixed basis states. $f(\tau)$ thus corresponds to the sum of an exponential number of products of matrix elements of $G_1, ..., G_L$. Evidently, only terms for which all the matrix elements in the product are non-zero contribute to the sum. Potentially non-zero terms in the sum can be efficiently constructed using Algorithm 1 below.

Algorithm 1: Efficiently constructing a (potentially) non-zero term from the sum in eq. 95.

Start off with state $|x_0\rangle$, which is in some set $S_1^{b(1)}$ of states that are connected through non-zero matrix elements of G_1. If $|x_0\rangle$ is not in any of the sets S_1^b, then output 0.
for $l \in \{1, ..., L-1\}$ **do**
  Pick a state $|x_l\rangle \in S_l^{b(l)}$ with probability $P_l(x_{l-1} \to x_l) = \frac{1}{\lambda_l^{b(l)}} \langle x_{l-1}| G_l |x_l\rangle \frac{\phi_l^{b(l)}(x_l)}{\phi_l^{b(l)}(x_{l-1})}$ (where $\phi_l^{b(l)}(x_{l-1}) \equiv \langle x_{l-1}|\phi\rangle_l^{b(l)}$ and $\phi_l^b(x_l) \equiv \langle x_l|\phi\rangle_l^{b(l)}$ are both positive since $|\phi\rangle_l^{b(l)}$ can be chosen to be the strictly positive eigenstate of $G_l^{b(l)}$, see Theorem IV.1). The state $|x_l\rangle$ can additionally be part of some set $S_{l+1}^{b(l+1)}$ of states that are connected through non-zero matrix elements of G_{l+1} – and if it is not in any of these sets, then output 0.
end
The last factor, $\langle x_{L-1}| G_L |x_L\rangle$, simply equals the matrix element of G_L between the state $|x_{L-1}\rangle$ and the fixed state $|x_L\rangle$. If this matrix element equals zero, then output 0.

For each of the steps $x_{l-1} \to x_l$ (where $l \in \{1, ..., L-1\}$), the associated probability distribution P_l can be shown to be normalized:

$$\sum_{x_l} P_l(x_{l-1} \to x_l) = \sum_{x_l} \frac{1}{\lambda_l^b} \langle x_{l-1}| G_l |x_l\rangle \frac{\phi_l^b(x_l)}{\phi_l^b(x_{l-1})} = \sum_{x_l \in S_l^b} \frac{1}{\lambda_l^b} \langle x_{l-1}| G_l^b |x_l\rangle \frac{\phi_l^b(x_l)}{\phi_l^b(x_{l-1})}$$

$$= \frac{1}{\phi_l^b(x_{l-1})} \frac{1}{\lambda_l^b} \langle x_{l-1}| G_l^b \sum_{x_l \in S_l^b} |x_l\rangle \phi_l^b(x_l) = \frac{1}{\phi_l^b(x_{l-1})} \frac{1}{\lambda_l^b} \langle x_{l-1}| \lambda_l^b |\phi\rangle_l^b = \frac{\langle x_{l-1}|\phi\rangle_l^b}{\phi_l^b(x_{l-1})} = 1. \quad (96)$$

We can now express $f(\tau)$ in terms of these P_l's as follows:

$$f(\tau) = \sum_{x_1, x_2, ..., x_{L-1}} \langle x_0| G_1 |x_1\rangle \langle x_1| G_2 |x_2\rangle \ ... \ \langle x_{L-1}| G_L |x_L\rangle =$$

$$\sum_{x_1, x_2, ..., x_{L-1}} P_1(x_0 \to x_1) P_2(x_1 \to x_2) \ ... \ P_L(x_{L-1} \to x_L) R(x_0, x_1, ..., x_L), \quad (97)$$

where $|x_0\rangle$ and $|x_L\rangle$ are fixed basis states and $R(x_0, x_1, ..., x_L) \equiv \prod_{l=1}^L \lambda_l^{b(l)} \frac{\phi_l^{b(l)}(x_{l-1})}{\phi_l^{b(l)}(x_l)}$.

We define $\boldsymbol{x} \equiv (x_0, x_1, ..., x_L)$. Then $P_L(x_{L-1} \to x_L) R(\boldsymbol{x})$ is an unbiased estimator of $f(\tau)$ – that is, $f(\tau) = \mathbb{E}\big(P_L(x_{L-1} \to x_L) R(\boldsymbol{x})\big)$ – if $\boldsymbol{x}$ is drawn from the following high-dimensional probability distribution:

$$\Pi(\boldsymbol{x}) \equiv P_1(x_0 \to x_1) P_2(x_1 \to x_2) ... P_{L-1}(x_{L-2} \to x_{L-1}). \quad (98)$$

Drawing a set Σ of independent $\boldsymbol{x}$'s from $\Pi(\boldsymbol{x})$ can efficiently be done by following the steps (except for the last one) in Algorithm 1 above. To show that $\Pi(\boldsymbol{x})$ itself is a normalized distribution, we successively exploit the normalization property of each individual P_l:

$$\sum_{x_1, x_2, ..., x_{L-1}} \Pi(\boldsymbol{x}) = \sum_{x_1} \Big(P_1(x_0 \to x_1) \sum_{x_2} \Big(P_2(x_1 \to x_2) ... \sum_{x_{L-1}} \Big(P_{L-1}(x_{L-2} \to x_{L-1}) \Big) ... \Big) \Big) = 1. \quad (99)$$

We have thus obtained an unbiased estimator of $f(\tau)$ and a way of calculating sample averages of this estimator. In order to complete the proof, we consider the variance of the value of the unbiased estimator: $\sigma^2 = \mathbb{E}\big(P_L^2(x_{L-1} \to x_L) R^2(\boldsymbol{x})\big) - \mathbb{E}\big(P_L(x_{L-1} \to x_L) R(\boldsymbol{x})\big)^2 = \mathbb{E}\big(P_L^2(x_{L-1} \to x_L) R^2(\boldsymbol{x})\big) - \big(f(\tau)\big)^2$. The quantity that is left to determine (or at least bound) is thus $\mathbb{E}\big(P_L^2(x_{L-1} \to x_L) R^2(\boldsymbol{x})\big)$, which is done in a similar way as has been done for $\mathbb{E}\big(P_L(x_{L-1} \to x_L) R(\boldsymbol{x})\big)$ above:

$$\mathbb{E}\Big(\big(P_L(x_{L-1} \to x_L) R(\boldsymbol{x})\big)^2 \Big) = \sum_{x_1, x_2, ..., x_{L-1}} \Pi(\boldsymbol{x}) \big(P_L(x_{L-1} \to x_L) R(\boldsymbol{x})\big)^2, \quad (100)$$

where again $|x_0\rangle = |x\rangle$ and $|x_L\rangle = |x'\rangle$ are fixed. Inserting the expressions for $\Pi(\boldsymbol{x})$ and $P_L(x_{L-1} \to x_L) R(\boldsymbol{x})$ yields:

$$\mathbb{E}\Big(\big(P_L(x_{L-1} \to x_L) R(\boldsymbol{x})\big)^2 \Big) =$$

$$\sum_{x_1, x_2, ..., x_{L-1}} P_1(x_0 \to x_1) P_2(x_1 \to x_2) ... P_{L-1}(x_{L-2} \to x_{L-1}) \big(P_L(x_{L-1} \to x_L)\big)^2 \left(\prod_{l=1}^L \lambda_l^{b(l)} \frac{\phi_l^{b(l)}(x_{l-1})}{\phi_l^{b(l)}(x_l)} \right)^2, \quad (101)$$

which, using $P_l(x_{l-1}, x_l) = \frac{1}{\lambda_l^b} \langle x_{l-1}| G_t |x_l\rangle \frac{\phi_l^b(x_l)}{\phi_l^b(x_{l-1})}$, reduces to:

$$\sum_{x_1, x_2, ..., x_{L-1}} \lambda_1^{b(1)} \langle x_0| G_1 |x_1\rangle \frac{\phi_1^{b(1)}(x_0)}{\phi_1^{b(1)}(x_1)} \lambda_2^{b(2)} \langle x_1| G_2 |x_2\rangle \frac{\phi_2^{b(2)}(x_1)}{\phi_2^{b(2)}(x_2)} ... \lambda_L^{b(L)} \langle x_{L-1}| G_L |x_L\rangle \frac{\phi_L^{b(L)}(x_{L-1})}{\phi_L^{b(L)}(x_L)}$$

$$\times P_L(x_{L-1} \to x_L). \quad (102)$$

We now define the non-negative quantity $Q_l(x,y) \equiv \lambda_l^b \langle x| G_l |y\rangle \frac{\phi_l^b(x)}{\phi_l^b(y)}$. By exploiting the Hermiticity of G_l^b, $Q_l(x,y)$ can be shown to have the following property:

$$\sum_x Q_l(x,y) = \sum_x \langle x| G_l |y\rangle \lambda_l^b \frac{\phi_l^b(x)}{\phi_l^b(y)} = \sum_{x \in S_l^b} \langle x| G_l^b |y\rangle \lambda_l^b \frac{\phi_l^b(x)}{\phi_l^b(y)} = \left(\lambda_l^b\right)^2 \frac{\langle \phi|_l^b |y\rangle}{\phi_l^b(y)} = \left(\lambda_l^b\right)^2 \leq 1. \tag{103}$$

$Q_l(x,y)$ thus satisfies $0 \leq Q_l(x,y) \leq 1$, $\forall x,y$ and $\forall l \in \{1,2,...,L\}$. Using this property of $Q_l(x,y)$, we infer:

$$\begin{aligned}
\mathbb{E}\Big(\big(P_L(x_{L-1} \rightarrow x_L) R(\boldsymbol{x})\big)^2\Big) &= \sum_{x_1,x_2,...,x_{L-1}} \underbrace{Q_1(x_0,x_1)}_{\leq 1} Q_2(x_1,x_2) \, ... \, Q_L(x_{L-1},x_L) \underbrace{P_L}_{\leq 1} \\
&\leq \sum_{x_1,x_2,...,x_{L-1}} Q_2(x_1,x_2) \, Q_3(x_2,x_3) \, ... \, Q_L(x_{L-1},x_L) \\
&\leq \sum_{x_2,x_3,...,x_{L-1}} Q_3(x_2,x_3) \, Q_4(x_3,x_4) \, ... \, Q_L(x_{L-1},x_L) \\
&... \\
&\leq 1,
\end{aligned} \tag{104}$$

where in the last few steps in eq. 104 we have successively used the property of $Q_l(x,y)$ in eq. 103. We thus conclude that the variance of the unbiased estimator $P_L(x_{L-1} \rightarrow x_L) R(\boldsymbol{x})$ is bounded as follows:

$$\sigma^2 \leq 1 - \big(f(\tau)\big)^2 \leq 1. \tag{105}$$

If we now draw a set Σ of (independent) $\boldsymbol{x}$'s from $\Pi(\boldsymbol{x})$ using Algorithm 1 above, then $f(\tau)$ can be estimated as follows:

$$\tilde{f}(\tau) = \frac{1}{|\Sigma|} \sum_{\boldsymbol{x} \in \Sigma} P_L(x_{L-1} \rightarrow x_L) R(\boldsymbol{x}), \tag{106}$$

where $|\Sigma|$ denotes the size of the set Σ. The standard deviation of the quantity $\tilde{f}(\tau)$ is now $\frac{\sigma}{\sqrt{|\Sigma|}}$. Hence if we take $|\Sigma| = \mathcal{N}\epsilon^{-2}$, then by means of Chebyshev's inequality:

$$Pr\Big(\big|\tilde{f}(\tau) - f(\tau)\big| \geq \epsilon\Big) \leq \frac{\sigma^2}{\mathcal{N}} \leq \frac{1}{\mathcal{N}}. \tag{107}$$

$$\boxed{Pr\Big(\big|\tilde{f}(\tau) - f(\tau)\big| \leq \epsilon\Big) \geq 1 - \frac{1}{\mathcal{N}}.} \tag{108}$$

This completes the proof of Theorem IV.2. ■

Since H is Hermitian, $f(\tau)$ can be decomposed as follows:

$$f(\tau) = \langle x| e^{-\tau H} \big|x'\big\rangle = \sum_{j=1}^{2^n} \langle x|\psi_j\rangle \big\langle \psi_j \big| x'\big\rangle e^{-\tau E_j}, \tag{109}$$

where $\{|\psi_j\rangle\}$ and $\{E_j\}$ respectively denote the sets of eigenstates and eigenvalues of H. Using a Monte Carlo algorithm such as the one described above, we can efficiently obtain a set $\{\tilde{f}(\Delta\tau\, k)\}_{k=1}^K$ (for $K \leq poly(n)$) of noisy measurements of $f(\tau)$ at the points $\{\Delta\tau\, k\}_{k=1}^K$ (which all lie on the positive real axis) in polynomial time. Using this set of estimates of $f(\tau)$, we would now like to estimate (a subset of) the eigenvalues $\{E_j\}$ of Hamiltonian H. The Matrix Pencil Method as described in the previous chapter will again be the tool for solving this problem.

The coefficient of the signal in eq. 109 associated with eigenvalue E_j equals the product of the overlaps between the eigenstate $|\psi_j\rangle$ and the basis states $|x\rangle$ and $|x'\rangle$. Since we are mostly interested in a particular set of eigenvalues (such as the lowest lying set) we must choose a pair of basis states ($|x\rangle$ and $|x'\rangle$) that has considerable overlap with the eigenstates that are part of this set. To ensure a larger success rate at completing this task, we must systematically select a set of pairs of basis states ($\{|x\rangle, |x'\rangle\}$) that

collectively has sufficiently large overlap with the particular set of eigenstates. For this reason, we consider superpositions of basis states in the following section.

IV.2 Considering superpositions of basis states

Ideally, we would be able to exploit superpositions of basis states to ensure sufficient overlap with a selected set of eigenstates of H. We would then choose a state which is known to have sufficient overlap with e.g. the ground state (and the first few excited states), such as a Hartree-Fock state.

We consider a particular class of superpositions of generally an exponential number of basis states:

$$|\Phi\rangle = \sum_{x=1}^{2^n} \Phi(x) |x\rangle , \tag{110}$$

where $\Phi(x) \in \mathbb{C}$ $(\forall x)$ and normalization ensures that $\sum_x \left|\Phi(x)\right|^2 = 1$. We now consider the task of estimating the following quantity: $\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle = \sum_{j=1}^{2^n} \left(\sum_{x,y} \Phi(x)\Phi(y) \langle x|\psi_j\rangle \langle\psi_j|y\rangle \right) e^{-\tau E_j}$. We note that $\mathcal{F}(\tau)$ is a real-valued quantity.

We present the following extension of Theorem IV.2 and provide its proof afterwards:

Theorem IV.3. (M.E. Stroeks, B.M. Terhal). *Suppose $|x\rangle$ (where $x \in \{0,1\}^n$) denotes the state of an n-qubit register ($\{|x\rangle\}$ is an orthonormal set). Furthermore, suppose $\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle = \langle\Phi| G_1 G_2 \ldots G_L |\Phi\rangle$, where:*

1. *$|\Phi\rangle = \sum_{x=1}^{2^n} \Phi(x) |x\rangle$, where $\Phi(x) \in \mathbb{C}$ $(\forall x)$ and $\sum_x \left|\Phi(x)\right|^2 = 1$,*
2. *$\tau \in \mathbb{R}_+$,*
3. *$G_l = \left(e^{-a_l \tau / M H_i}\right)_j$ is a Hermitian matrix, where $i \in \{1, ..., N\}$, $j \in \{1, ..., M\}$, $a_l \in (0, 1]$ ($\{a_l\}_{l=0}^L$ depends on the Trotterization scheme) and $l \in \{1, ..., L\}$ labels the time slice,*
4. *$H = \sum_{i=1}^N H_i$, where each H_i acts non-trivially on at most $log(poly(n))$, or typically $O(1)$, qubits. H_i is Hermitian and stoquastic in the 'x'-basis, $\forall i$ (hence G_l is element-wise non-negative and Hermitian, $\forall l$),*
5. *$\lambda_{min}(H_i) = 0$ $(\forall i)$ so that $\omega_{max}(G_t) = 1$ $(\forall t)$, where 'ω' denotes an eigenvalue of G_l.*

Assume additionally that $\frac{\Phi(y)}{\Phi(x)^*}$ (where z^* denotes the complex conjugate of z) can be efficiently calculated for given x and y and we can efficiently draw samples from $P(x) = \left|\Phi(x)\right|^2$. Then $\mathcal{F}(\tau)$ can be estimated with absolute error ϵ in time $poly(\epsilon^{-1}, n, L)$. The algorithm runs efficiently for $\epsilon \geq \frac{1}{poly(n)}$ and $L \leq poly(n)$.

Proof. The proof of this theorem strongly resembles the proof of Theorem IV.2. We rewrite the object of interest $\mathcal{F}(\tau)$ as follows:

$$\begin{aligned}\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle &= \sum_{x,y} \Phi(x)\Phi(y) \underbrace{\langle x| e^{-\tau H} |y\rangle}_{f(\tau;x,y)} = \sum_{x,y} |\Phi(x)|^2 \frac{\Phi(y)}{\Phi(x)^*} f(\tau; x, y) \\ &= \sum_{x_0, x_1, \ldots, x_L} |\Phi(x_0)|^2 \frac{\Phi(x_L)}{\Phi(x_0)^*} \langle x_0| G_1 |x_1\rangle \langle x_1| G_2 |x_2\rangle \ \ldots \ \langle x_{L-1}| G_L |x_L\rangle , \end{aligned} \tag{111}$$

where we have set $|x\rangle = |x_0\rangle$ and $|y\rangle = |x_L\rangle$ (we note that these basis states are no longer fixed in the current discussion). We now use $P_l(x_{l-1} \to x_l) = \frac{1}{\lambda_l^b} \langle x_{l-1}| G_l |x_l\rangle \frac{\phi_l^b(x_l)}{\phi_l^b(x_{l-1})}$ to obtain the following expression for $\mathcal{F}(\tau)$ (just like was done in the previous section for $f(\tau)$):

$$\mathcal{F}(\tau) = \sum_{x_0, x_1, \ldots, x_L} \underbrace{|\Phi(x_0)|^2 P_1(x_0 \to x_1) P_2(x_1 \to x_2) \ldots P_L(x_{L-1} \to x_L)}_{\tilde{\Pi}(\boldsymbol{x})} \times \underbrace{\frac{\Phi(x_L)}{\Phi(x_0)^*} \prod_{l=1}^{L} \lambda_l^{b(l)} \frac{\phi_l^{b(l)}(x_{l-1})}{\phi_l^{b(l)}(x_l)}}_{\mathcal{R}(\boldsymbol{x})}, \tag{112}$$

where $\boldsymbol{x} \equiv (x_0, x_1, ..., x_L)$. By thus sampling from the distribution $\tilde{\Pi}(\boldsymbol{x})$ (which is normalized due to the normalization property of state $|\Phi\rangle$ and the constituent P_l's) and calculating a sample average of $\mathcal{R}(\boldsymbol{x}) \in \mathbb{C}$, we can estimate $\mathcal{F}(\tau)$ without bias (except for the bias introduced by the Trotter error): $\mathcal{F}(\tau) = \mathbb{E}\big(\mathcal{R}(\boldsymbol{x})\big)$. A factor that governs the efficiency of this estimation is the ability to efficiently calculate $\frac{\Phi(y)}{\Phi(x)^*}$ for a given x and y and to efficiently draw samples from $P(x) = \big|\Phi(x)\big|^2$. We shall come back to this point at the end of this section. We note furthermore that since $\mathcal{F}(\tau) \in \mathbb{R}$, it holds that $Re\big(\mathcal{F}(\tau)\big) = \mathcal{F}(\tau)$. Therefore, $\mathcal{F}(\tau) = Re\big(\mathcal{F}(\tau)\big) = Re\big(\mathbb{E}\big(\mathcal{R}(\boldsymbol{x})\big)\big)$. Since $\mathbb{E}(Z) \equiv \mathbb{E}\big(Re(Z)\big) + i\mathbb{E}\big(Im(Z)\big)$ for a complex random variable Z, it holds that $Re\big(\mathbb{E}(Z)\big) = \mathbb{E}\big(Re(Z)\big)$. We thus conclude that $\mathcal{F}(\tau) = \mathbb{E}\big(Re\big(\mathcal{R}(\boldsymbol{x})\big)\big)$. We shall use this property near the end of this section.

To determine whether $\mathcal{F}(\tau)$ can be estimated efficiently, we must determine the variance of the unbiased estimator $\mathcal{R}(\boldsymbol{x}) \in \mathbb{C}$. This variance is given by:

$$\sigma_{\mathcal{F}}^2 = \mathbb{E}\Big(\big|\mathcal{R}(\boldsymbol{x})\big|^2\Big) - \underbrace{\Big|\mathbb{E}\Big(\mathcal{R}(\boldsymbol{x})\Big)\Big|^2}_{= |\mathcal{F}(\tau)|^2 = \mathcal{F}(\tau)^2} \leq \mathbb{E}\Big(\big|\mathcal{R}(\boldsymbol{x})\big|^2\Big), \tag{113}$$

where the inequality holds because $|\mathcal{F}(\tau)|^2 \geq 0$ (since $|\mathcal{F}(\tau)| \in \mathbb{R}$). To obtain an upper bound on the variance, we shall investigate this expression in more detail (in a way closely resembling the analysis of the variance of $f(\tau)$ in the previous section):

$$\begin{aligned}
\mathbb{E}\Big(\big|\mathcal{R}(\boldsymbol{x})\big|^2\Big) &= \sum_{\boldsymbol{x}} \tilde{\Pi}(\boldsymbol{x})\big|\mathcal{R}(\boldsymbol{x})\big|^2 \\
&= \sum_{\boldsymbol{x}} |\Phi(x_L)|^2 \ \langle x_0|\, G_1\, |x_1\rangle \, \langle x_1|\, G_2\, |x_2\rangle \ ... \ \langle x_{L-1}|\, G_L\, |x_L\rangle \prod_{l=1}^{L} \lambda_l^{b(l)} \frac{\phi_l^{b(l)}(x_{l-1})}{\phi_l^{b(l)}(x_l)} \\
&= \sum_{\boldsymbol{x}} |\Phi(x_L)|^2 \, Q_1(x_0, x_1) Q_2(x_1, x_2) \, ... \, Q_L(x_{L-1}, x_L),
\end{aligned} \tag{114}$$

where in the last step we identify $Q_l(x, y) = \langle x|\, G_l\, |y\rangle \, \lambda_l^{b(l)} \frac{\phi_l^{b(l)}(x)}{\phi_l^{b(l)}(y)}$. This quantity has been shown in Section IV.1 to have the properties: $\sum_x Q_l(x, y) \leq 1$ and $0 \leq Q_l(x, y) \leq 1$ $(\forall l)$. By consecutively exploiting the former property for all Q_l's and the normalization property of state $|\Phi\rangle$, the variance can be shown to be upper bounded by unity:

$$\sigma_{\mathcal{F}}^2 \leq 1. \tag{115}$$

When a set Σ of independent samples $\{\boldsymbol{x}\}$ distributed according to $\tilde{\Pi}(\boldsymbol{x})$ is obtained, an unbiased estimate of $\mathcal{F}(\tau)$ can be computed by calculating the sample average of $\mathcal{R}(\boldsymbol{x})$ over the sample set Σ:

$$\tilde{\mathcal{F}}(\tau) = \frac{1}{|\Sigma|} \sum_{\boldsymbol{x} \in \Sigma} \mathcal{R}(\boldsymbol{x}), \tag{116}$$

where $|\Sigma|$ denotes the size of set Σ. We note that since $\mathcal{R}(\boldsymbol{x}) \in \mathbb{C}$, the quantity $\tilde{\mathcal{F}}(\tau)$ will generally be complex-valued as well. The variance of $\tilde{\mathcal{F}}(\tau)$ is now given by $\frac{\sigma_{\mathcal{F}}^2}{|\Sigma|} \leq \frac{1}{|\Sigma|}$. Since $\mathrm{Var}\big(\tilde{\mathcal{F}}\big) = \mathrm{Var}\big(Re(\tilde{\mathcal{F}})\big) + \mathrm{Var}\big(Im(\tilde{\mathcal{F}})\big) \geq \mathrm{Var}\big(Re(\tilde{\mathcal{F}})\big)$, we note that the variance of the real part of observable $\tilde{\mathcal{F}}$ (which itself is a real-valued random variable) is also upper bounded by $\frac{1}{|\Sigma|}$. Furthermore, we have previously shown that $\mathbb{E}\big(Re\big(\mathcal{R}(\boldsymbol{x})\big)\big) = \mathcal{F}(\tau)$, so that the mean value of $Re(\tilde{\mathcal{F}})$ equals $\mathcal{F}(\tau)$. Hence if we take $|\Sigma| = \mathcal{N}\epsilon^{-2}$, then by means of Chebyshev's inequality:

$$Pr\Big(\big|Re\big(\tilde{\mathcal{F}}(\tau)\big) - \mathcal{F}(\tau)\big| \geq \epsilon\Big) \leq \frac{1}{\mathcal{N}}. \tag{117}$$

$$\boxed{Pr\Big(\big|Re\big(\tilde{\mathcal{F}}(\tau)\big) - \mathcal{F}(\tau)\big| \leq \epsilon\Big) \geq 1 - \frac{1}{\mathcal{N}}.} \tag{118}$$

$\mathcal{F}(\tau)$ can thus be estimated with absolute error ϵ in time $poly(\epsilon^{-1}, n, L)$. This completes the proof of Theorem IV.3. ■

As is stated in Theorem IV.3, the efficiency of the evaluation of $\mathcal{F}(\tau)$ depends on the ability to, for each Monte Carlo sample, efficiently obtain the quantity $\frac{\Phi(y)}{\Phi(x)^*}$ and a sample from the distribution $\left|\Phi(x)\right|^2$. Since $|\Phi\rangle$ is a superposition of generally exponentially many terms, the trivial strategy of storing the complete state $|\Phi\rangle$ to obtain the quantities of interest will not be computationally tractable. The extent to which the aforementioned quantities of interest can be *efficiently* obtained depends on the exact form of the state $|\Phi\rangle = \sum_x \Phi(x)\,|x\rangle$. In many practical settings, $|\Phi\rangle$ is such that one can define a function $f : \{0,1\}^n \to \mathbb{C}$ which takes as input the n-bit string x, and efficiently outputs the corresponding coefficient $\Phi(x)$ (this is e.g. the case for (matrix) product states or states produced by very low-depth quantum circuits). Thereby, given x and y, the fraction $\frac{\Phi(y)}{\Phi(x)^*}$ can be efficiently obtained. A potential strategy for the efficient sampling from distribution $\left|\Phi(x)\right|^2$ is the Metropolis algorithm (see Definition I.15): By setting up a Markov Chain Monte Carlo scheme, one can (attempt to) sample from the distribution $\left|\Phi(x)\right|^2$ in $poly(n)$ time. Whether or not the sampling can indeed be conducted in $poly(n)$ time depends on with what pace the 'walkers' equilibrate to a distribution that closely resembles $\left|\Phi(x)\right|^2$ (the stationary distribution). We note that in many interesting cases one can also obtain strategies to sample from $\left|\Phi(x)\right|^2$ that do not depend on the Metropolis algorithm, and do not require the introduction of walker dynamics. An example of such a strategy is presented in the next chapter of this report.

IV.3 Extension to Non-Hermitian Propagation Operators

Since G_l equals $e^{-a_l \tau/M\, H_i}$ and H_i is Hermitian (it corresponds to a physical observable), G_l is Hermitian as well. However, to make use of the Theorems from the previous sections in a setting that is more general than the estimation the eigenvalues of stoquastic Hamiltonians, we consider an extension of Theorems IV.2 and IV.3 to the case for which the local propagation operators G_l are not Hermitian (which is the case when e.g. H_i itself is not Hermitian). Since Theorem IV.2 is equivalent to a particular case of Theorem IV.3, we consider the estimation of quantity $\mathcal{F}(\tau) = \langle\Phi|\, G_1 G_2 ... G_L\, |\Phi\rangle$ from the previous section, where we now lift the condition of G_l being Hermitian.

In addition to the n-qubit register, we exploit a single ancillary qubit. The matrices G_l are still element-wise non-negative. The state $|a\rangle$ denotes the state of the single ancillary qubit. By making use of the single ancillary qubit, the propagation operators can be symmetrized as follows:

$$F_l \equiv \begin{cases} G_l \otimes |0\rangle\langle 1| + G_l^\dagger \otimes |1\rangle\langle 0| \;, & \text{if } l \text{ is odd} \\ G_l \otimes |1\rangle\langle 0| + G_l^\dagger \otimes |0\rangle\langle 1| \;, & \text{if } l \text{ is even.} \end{cases} \tag{119}$$

In this form, F_l (the 'new' propagation operator) is element-wise non-negative and Hermitian. It is important to note that each G_l acts on a subset of the n-qubit register and that F_l acts on the same subset of the n-qubit register, as well as on the ancillary qubit.

What is left to prove is that estimating the signal for the string of F_l's is equivalent to estimating the signal for the string of G_l's. Specifically, we want to prove the following identity: $\langle\Phi|\, G_1 G_2 ... G_L\, |\Phi\rangle = \langle\Phi| \otimes \langle 0|\, F_1 F_2 ... F_L\, |\Phi\rangle \otimes |L \bmod 2\rangle$, for $L \in \mathbb{Z}_+$. This is done below by means of induction.

- For $L = 1$:

$$\begin{aligned} \langle\Phi| \otimes \langle 0|\, F_1 \left|\Phi'\right\rangle \otimes |1\rangle &= \langle\Phi| \otimes \langle 0| \left(G_1 \otimes |0\rangle\langle 1| + G_1^\dagger \otimes |1\rangle\langle 0|\right) \left|\Phi'\right\rangle \otimes |1\rangle \\ &= \langle\Phi|\, G_1 \left|\Phi'\right\rangle \otimes \langle 0|0\rangle\, \langle 1|1\rangle + \langle\Phi|\, G_1^\dagger \left|\Phi'\right\rangle \otimes \langle 0|1\rangle\, \langle 0|1\rangle \\ &= \langle\Phi|\, G_1 \left|\Phi'\right\rangle, \end{aligned} \tag{120}$$

  where $|\Phi\rangle$ and $|\Phi'\rangle$ can be all possible (potentially unnormalized) n-qubit states.

- Assuming the identity $\langle\Phi|\, G_1 G_2 ... G_L\, |\Phi'\rangle = \langle\Phi| \otimes \langle 0|\, F_1 F_2 ... F_L\, |\Phi'\rangle \otimes |L \bmod 2\rangle$ (for all sets of two n-qubit states $\{|\Phi\rangle, |\Phi'\rangle\}$) holds for L, it holds for $L+1$ as well:

  Making use of the definition in eq. 119, we write F_{L+1} as follows:

$$F_{L+1} = G_{L+1} \otimes |L \bmod 2\rangle\langle L+1 \bmod 2| + G_{L+1}^\dagger \otimes |L+1 \bmod 2\rangle\langle L \bmod 2| \,. \tag{121}$$

  The quantity of interest – in the case of the length of the operator string being $L+1$ – can now be

rewritten as follows:

$$\begin{aligned}\langle\Phi|\otimes\langle 0|F_1F_2...F_LF_{L+1}\left|\Phi'\right\rangle\otimes|L+1\,mod\,2\rangle &= \langle\Phi|\otimes\langle 0|F_1F_2...F_L\underbrace{G_{L+1}\left|\Phi'\right\rangle}_{\left|\tilde{\Phi}\right\rangle}\otimes|L\,mod\,2\rangle \\ &= \langle\Phi|G_1G_2...G_L\left|\tilde{\Phi}\right\rangle \\ &= \langle\Phi|G_1G_2...G_LG_{L+1}\left|\Phi'\right\rangle,\end{aligned} \tag{122}$$

where in the second equality we have used the aforementioned presumption (note that $\left|\tilde{\Phi}\right\rangle \equiv G_{L+1}\left|\Phi'\right\rangle$ is an n-qubit state). This equation holds for all sets of two n-qubit states $\{|\Phi\rangle, |\Phi'\rangle\}$.

We conclude, by induction, that the identity $\langle\Phi|G_1G_2...G_L|\Phi'\rangle = \langle\Phi|\otimes\langle 0|F_1F_2...F_L|\Phi'\rangle\otimes|L\,mod\,2\rangle$ holds for all $L\in\mathbb{Z}_+$. Therefore $\langle\Phi|G_1G_2...G_L|\Phi\rangle = \langle\Phi|\otimes\langle 0|F_1F_2...F_L|\Phi\rangle\otimes|L\,mod\,2\rangle$ (for $L\in\mathbb{Z}_+$) holds as well, since it is the particular case of the previous identity corresponding to $|\Phi\rangle = |\Phi'\rangle$. Hence determining the signal $\langle\Phi|G_1G_2...G_L|\Phi\rangle$ (up to an absolute ϵ error) can be efficiently done in the case of non-Hermitian G_l's by introducing a single ancillary qubit and determining the signal $\langle\Phi|\otimes\langle 0|F_1F_2...F_L|\Phi\rangle\otimes|L\,mod\,2\rangle$ (where F_l is defined in eq. 119) using the methods developed in the previous sections.

Remark. ***Extension of Theorem IV.3:*** *Assuming access to a single ancillary qubit (in addition to the n-qubits) and fulfillment of the conditions set in Theorem IV.3 with the exception that G_l is allowed to be non-Hermitian ($\forall l$), the quantity $\langle\Phi|G_1G_2...G_L|\Phi\rangle$ can be obtained up to an ϵ-additive error in time $poly(\epsilon^{-1}, n, L)$.*

IV.4 Short-Time Simulations

Thus far, we have not restricted the allowed values that τ can attain other than them being real and non-negative. In this section, we argue why taking τ to be a *small* (real-valued and non-negative) quantity is beneficial for the quality of the simulations considered here. There are four reasons to believe that taking τ small is indeed advantageous:[4]

1. For a given number of Monte Carlo samples ($|\Sigma|$), the (sampling) noise contaminates the signals to be determined to a given extent (the variance is upper bounded by $1/|\Sigma|$). Since the signals decay in imaginary time τ, there will inevitably be a point in imaginary time *after* which the noise dominates the signal to be determined. To avoid this effect, one must choose a measurement τ-interval at sufficiently small τ such that, for a given number of Monte Carlo samples, the signal dominates the noise within that interval.

2. In the previous sections, the variance of a single realization (for a single Monte Carlo sample) of the observables of interest ($P_L(x_{L-1}\rightarrow x_L)R(\boldsymbol{x})$ and $\mathcal{R}(\boldsymbol{x})$) has been upper bounded by unity (which, clearly, is a time-independent quantity). To arrive at this upper bound on the variance, we used that both $f(\tau)$ and $\mathcal{F}(\tau)$ are real-valued and therefore their square is non-negative: $\sigma_f^2 \leq 1-\big(f(\tau)\big)^2 \leq 1$ and $\sigma_{\mathcal{F}}^2 \leq 1-\big(\mathcal{F}(\tau)\big)^2 \leq 1$. Since the signals $f(\tau)$ and $\mathcal{F}(\tau)$ are not known before running the simulation, upper bounding these variances by this τ-independent unit bound is reasonable, and useful in proving Theorems IV.2 and IV.3. However, since $f(\tau) = \langle x|e^{-\tau H}|x'\rangle$ and $\mathcal{F}(\tau) = \langle\Phi|e^{-\tau H}|\Phi\rangle$ both equal unity at $\tau = 0$, the aforementioned variances must equal zero there. In fact, the unity variance bounds are only tight when $\lim_{\tau\rightarrow\infty}$. For small τ, the variances are smaller than the time-independent bounds due to the fact that the signals have non-zero magnitude. One can therefore try to exploit this fact to reduce the noise on the approximated signals.

3. As was extensively discussed in the introductory chapter of this report, the systematic Trotter error imposed on the signals to be determined grows with τ. Therefore, choosing the measurement τ-interval to be at relatively small τ (for a given Trotter variable M) suppresses the systematic error originating from Trotterization.

[4] These four reasons are not necessarily mutually independent.

4. The algorithms presented in the previous sections determine the signals of interest up to an additive constant ϵ, where for the algorithm to run efficiently ϵ must be at least $1/poly(n)$. Therefore, the signals are only estimated well if they themselves are also of size at least $1/poly(n)$ (and preferably larger). As was discussed in the introductory chapter, frustration of the Hamiltonian H is one of the effects that reduces the magnitudes of the signals. In particular, we have shown in Section I.2 that the signals are of size at least $1/poly(n)$ if $\tau\Lambda \leq log(poly(n)) \ \wedge \ |\langle\Phi|\psi_0\rangle|^2 \geq 1/poly(n)$ (where Λ denotes the degree to which the Hamiltonian H is frustrated). For a given Λ, therefore, τ must be chosen relatively small.

In the next chapter, the claim of improved quality of the simulations when choosing small τ values will also be supported by means of numerical results. In fact, one wants to choose $\Delta\tau$ so small that all $k\,\Delta\tau$ for $k \in \{0, 1, ..., K\}$ (and a given K) lie at sufficiently small τ. We do note that $\Delta\tau$ must not be chosen so small that discretization errors start to play a role when numerically obtaining the signal.

IV.5 Determining the Partition Function

Thus far we have focused on extracting eigenvalues of a Hamiltonian H from the real-time or imaginary-time evolution of the state of a system with Hamiltonian H. To further study a system and its physical properties, one might also consider determining its canonical partition function $Z(\beta)$ (at inverse temperature $\beta = \frac{1}{k_B T} \in \mathbb{R}_+$). This partition function is directly related to quantities such as the Helmholtz free energy $F = -k_B T\, ln\big(Z(\beta)\big)$ and the specific heat $C_V = \frac{\partial\langle E\rangle}{\partial T}$ (where $\langle E\rangle = -\frac{\partial ln(Z)}{\partial\beta}$). By studying e.g. the derivatives of the Helmholtz free energy – and in particular discontinuities of these derivatives – one can investigate the occurrence of phase transitions of the system.

In this section we argue that, using similar reasoning as for the quantities $f(\tau)$ and $\mathcal{F}(\tau)$ (introduced in the previous sections), the partition function $Z(\beta)$ can be efficiently obtained by classical means (up to an ϵ-additive error and a Trotter error) for sufficiently local and piece-wise stoquastic Hamiltonians. The partition function is given by the following expression (where $\{|x\rangle\}$ constitutes the basis in which $e^{-\beta H}$ is expressed):

$$Z(\beta) = Tr\big(e^{-\beta H}\big) = \sum_x \langle x|\, e^{-\beta H}\, |x\rangle = \sum_j \sum_x e^{-\beta E_j} |\langle\psi_j|x\rangle|^2 = \sum_j e^{-\beta E_j}, \tag{123}$$

where $\{\psi_j\}$ and $\{E_j\}$ respectively denote the eigenstates and eigenvalues of H. The partition function thus equals the sum over all eigenvalues of $e^{-\beta H}$. Since $\beta \in \mathbb{R}_+$ and $E_j \in \mathbb{R}$, $\forall j$ (since H is Hermitian), the partition function is a real-valued quantity. For the estimation algorithm discussed here, we shall use the expression $\sum_x \langle x|\, e^{-\beta H}\, |x\rangle$. We now present the following theorem, and provide its proof afterwards.

Theorem IV.4. (S. Bravyi, M.E. Stroeks, B.M. Terhal). *Suppose $|x\rangle$ (where $x \in \{0,1\}^n$) denotes the state of an n-qubit register ($\{|x\rangle\}$ is an orthonormal set). Furthermore, suppose $Z(\beta) = Tr\big(e^{-\beta H}\big) = \sum_x \langle x|\, e^{-\beta H}\, |x\rangle = \sum_x \langle x|\, G_1 G_2 \, ... \, G_L\, |x\rangle$, where:*

1. *$\beta \in \mathbb{R}_+$,*
2. *$G_l = \big(e^{-a_l \beta / M H_i}\big)_j$ is a Hermitian matrix, where $i \in \{1, ..., N\}$, $j \in \{1, ..., M\}$, $a_l \in (0, 1]$ ($\{a_l\}_{l=0}^L$ depends on the Trotterization scheme) and $l \in \{1, ..., L\}$ labels the time slice,*
3. *$H = \sum_{i=1}^N H_i$, where each H_i acts non-trivially on at most $log(poly(n))$, or typically $O(1)$, qubits. H_i is Hermitian and stoquastic in the 'x'-basis, $\forall i$ (hence G_l is element-wise non-negative and Hermitian, $\forall l$),*
4. *$\lambda_{min}(H_i) = 0$ ($\forall i$) so that $\omega_{max}(G_l) = 1$ ($\forall l$), where 'ω' denotes an eigenvalue of G_l.*

Then $Z(\beta)$ can be estimated with absolute error ϵ in time $poly(\epsilon^{-1}, n, L)$. The algorithm runs efficiently for $\epsilon \geq \frac{1}{poly(n)}$ and $L \leq poly(n)$.

Proof. We consider the Trotterized expression for the partition function $Z(\beta)$ and insert $L-1$ complete sets of basis states to obtain the following expression:

$$Z(\beta) = \sum_{x} \langle x| G_1 G_2 ... G_L |x\rangle = \sum_{x_0, x_1, ..., x_{L-1}} \langle x_0| G_1 |x_1\rangle\langle x_1| G_2 |x_2\rangle ... \langle x_{L-1}| G_L |x_0\rangle , \tag{124}$$

where $|x\rangle = |x_0\rangle$ is a variable basis state. We note that each term in the sum in eq. 124 corresponds to a string of basis states of length L, where the starting basis state always coincides with the final basis state in the string (here denoted by $|x_0\rangle = |x_L\rangle$). We now use $P_l(x_{l-1} \to x_l) = \frac{1}{\lambda_l^b} \langle x_{l-1}| G_l |x_l\rangle \frac{\phi_l^b(x_l)}{\phi_l^b(x_{l-1})}$ to obtain the following expression for $Z(\beta)$:

$$Z(\beta) = \sum_{x_0, x_1, ..., x_{L-1}} \underbrace{P_1(x_0 \to x_1) P_2(x_1 \to x_2) ... P_{L-1}(x_{L-2} \to x_{L-1})}_{\Pi(\boldsymbol{x})} P_L(x_{L-1} \to x_0) \times \underbrace{\prod_{l=1}^{L} \lambda_l^{b(l)} \frac{\phi_l^{b(l)}(x_{l-1})}{\phi_l^{b(l)}(x_l)}}_{R(\boldsymbol{x})}, \tag{125}$$

where $\boldsymbol{x} \equiv (x_0, x_1, ..., x_{L-1})$. The distribution $\Pi(\boldsymbol{x})$ is normalized due to the normalization property of each constituent P_l. By sampling from $\Pi(\boldsymbol{x})$ (in the same manner as described in the previous sections of this chapter) and calculating the sample average of $P_L(x_{L-1} \to x_0) R(\boldsymbol{x})$, one can estimate (a Trotterized version of) $Z(\beta)$ in an unbiased manner. Note that $P_L(x_{L-1} \to x_0)$ is part of the estimator since for a given $\boldsymbol{x} = (x_0, x_1, ..., x_{L-1})$ it is a fixed quantity. This is again due to the fact, as mentioned above, for each string of basis states the starting basis state must coincide with the final basis state.

To determine whether $Z(\beta)$ can be determined efficiently, we now consider the variance of the unbiased estimator $P_L(x_{L-1} \to x_0) R(\boldsymbol{x})$. This variance is given by:

$$\sigma_Z^2 = \mathbb{E}\Big(\Big(P_L(x_{L-1} \to x_0) R(\boldsymbol{x}) \Big)^2 \Big) - \underbrace{\mathbb{E}\Big(P_L(x_{L-1} \to x_0) R(\boldsymbol{x}) \Big)^2}_{Z(\beta)^2} \leq \mathbb{E}\Big(\Big(P_L(x_{L-1} \to x_0) R(\boldsymbol{x}) \Big)^2 \Big), \tag{126}$$

where the inequality holds because $Z(\beta)^2 \geq 0$ (since $Z(\beta) \in \mathbb{R}$). We now examine this expression for the variance in more detail to obtain an upper bound for it:

$$\begin{aligned} \mathbb{E}\Big(\Big(P_L(x_{L-1} \to x_0) R(\boldsymbol{x}) \Big)^2 \Big) &= \sum_{\boldsymbol{x}} \Pi(\boldsymbol{x}) \Big(P_L(x_{L-1} \to x_0) R(\boldsymbol{x}) \Big)^2 \\ &= \sum_{\boldsymbol{x}} \langle x_0| G_1 |x_1\rangle\langle x_1| G_2 |x_2\rangle ... \langle x_{L-1}| G_L |x_0\rangle P_L(x_{L-1} \to x_0) \\ &\qquad \times \prod_{l=1}^{L} \lambda_l^{b(l)} \frac{\phi_l^{b(l)}(x_{l-1})}{\phi_l^{b(l)}(x_l)} \\ &= \sum_{\boldsymbol{x}} Q_1(x_0, x_1) Q_2(x_1, x_2) ... Q_L(x_{L-1}, x_0) P_L(x_{L-1} \to x_0), \end{aligned} \tag{127}$$

where we identify $Q_l(x, y) = \langle x| G_l |y\rangle \lambda_l^b \frac{\phi_l^b(x)}{\phi_l^b(y)}$, which has been shown to have the properties: $\sum_x Q_l(x, y) \leq 1$ and $0 \leq Q_l(x, y) \leq 1$ $(\forall l)$. By now using these properties of $Q_l(x, y)$ and the fact that $0 \leq P_L \leq 1$, we show that the variance σ_Z^2 is upper bounded by unity:

$$\sigma_Z^2 \leq 1. \tag{128}$$

When a set Σ of independent samples $\{\boldsymbol{x}\}$ distributed according to $\Pi(\boldsymbol{x})$ is obtained, an unbiased estimate of $Z(\beta)$ can be computed by calculating the sample average of $P_L(x_{L-1} \to x_0) R(\boldsymbol{x})$ over the sample set Σ:

$$\tilde{Z}(\beta) = \frac{1}{|\Sigma|} \sum_{\boldsymbol{x} \in \Sigma} P_L(x_{L-1} \to x_0) R(\boldsymbol{x}), \tag{129}$$

where $|\Sigma|$ denotes the size of set Σ. The standard deviation of $\tilde{Z}(\beta)$ now equals $\frac{\sigma_Z}{\sqrt{|\Sigma|}}$. Hence if we take $|\Sigma| = \mathcal{N}\epsilon^{-2}$, then by means of Chebyshev's inequality:

$$Pr\Big(|\tilde{Z}(\beta) - Z(\beta)| \geq \epsilon\Big) \leq \frac{\sigma_Z^2}{\mathcal{N}} \leq \frac{1}{\mathcal{N}}. \tag{130}$$

$$\boxed{Pr\Big(|\tilde{Z}(\beta) - Z(\beta)| \leq \epsilon\Big) \geq 1 - \frac{1}{\mathcal{N}}.} \tag{131}$$

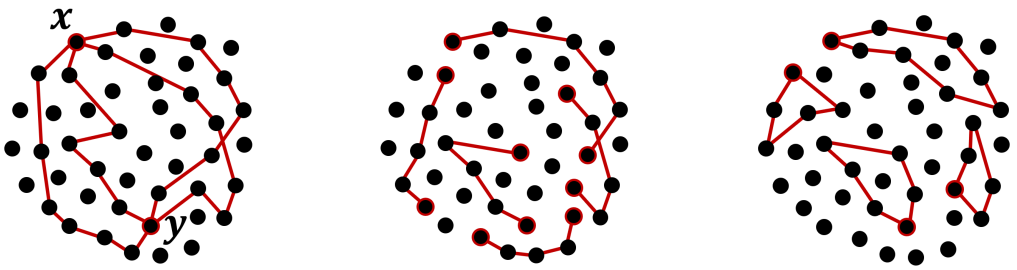
$Z(\beta)$ can thus be estimated with absolute error ϵ in time $poly(\epsilon^{-1}, n, L)$. This completes the proof of Theorem IV.4. ■

Ideally, one would obtain an estimate of the partition function (denoted by $\tilde{Z}(\beta)$) up to a relative error ϵ: $1 - \epsilon \leq \tilde{Z}/Z \leq 1 + \epsilon$. Such an estimate will be of high quality irrespective of the magnitude of Z, while an estimate of $Z(\beta)$ up to an ϵ-additive error will be of high quality only if Z is a relatively large quantity. An algorithm that provides an estimate of $Z(\beta)$ up to a relative error ϵ is called a fully polynomial-runtime approximation scheme (FPRAS). The range of systems for which such an FPRAS has been obtained, however, is limited. In [33], an FPRAS is obtained for the partition function of a family of stoquastic Hamiltonians associated with spin systems with ferromagnetic spin-spin interactions (including the ferromagnetic XY model and the ferromagnetic Ising model on any graph). More recently ([34]), an FPRAS was obtained for the partition function of the transverse-field Ising model at temperatures above a threshold that (amongst other variables) depends on the maximum interaction strength between the spins.

Since our algorithm obtains an ϵ-additive estimate of $Z(\beta)$ in an efficient manner for $\epsilon \geq 1/poly(n)$, one requires $Z(\beta)$ to be at least $1/poly(n)$ for the estimate to be accurate. As discussed in Section I.2 of this report, for our scheme the size of $Z(\beta)$ is directly related to the extent to which Hamiltonian H is frustrated: If Hamiltonian H is frustrated by an amount Λ, then $Z(\beta)$ is of size at least $1/poly(n)$ if $\beta\Lambda \leq log(poly(n))$. For a given scaling of the frustration with the system size $\Lambda(n)$, the Monte Carlo scheme described in this section will only provide accurate estimates of $Z(\beta)$ for $\beta \leq \frac{log(poly(n))}{\Lambda(n)}$. $Z(\beta)$ can thus be accurately obtained using our scheme in a high-temperature regime, where the *minimum* temperature in this regime increases progressively with the (scaling) of the degree of frustration. We thus conclude that our scheme provides an accurate estimate of $Z(\beta) = Tr\big(e^{-\beta H}\big)$ if H is sufficiently local and piece-wise stoquastic, and in addition H is mildly frustrated and the system temperature is not too low.

IV.6 Overview

In this section, we provide a short overview of the approximation schemes discussed in this chapter for the quantities $f(\tau)$ (a single matrix element of the Gibbs density matrix), $\mathcal{F}(\tau)$ (the overlap between an initial system state and its imaginary-time-evolved counterpart) and the $Z(\beta)$ (the partition function). We have argued in this chapter that these quantities, after applying a Trotterization scheme and inserting complete sets of basis states, correspond to a sum of exponentially many terms (the direct evaluation of which is, obviously, not computationally tractable). Using a classical stochastic scheme, one can estimate the quantities of interest up to an ϵ-additive error in a computationally tractable manner – given that the Hamiltonian is sufficiently local and piece-wise stoquastic. We have schematically depicted (an impression of) the classical stochastic schemes in Figure 6. In the representation used in Figure 6, single terms from the sums that make up the quantities $f(\tau)$, $\mathcal{F}(\tau)$ and $Z(\beta)$ correspond respectively to paths from a fixed basis state to another fixed basis state, paths from a variable basis state to another variable basis state (where the choice of these variable basis states depends on $|\Phi\rangle$) and closed loops starting from variable basis states. If one wishes to estimate $f(\tau)$, $\mathcal{F}(\tau)$ or $Z(\beta)$, one must sample $|\Sigma|$ of these associated paths using the algorithm described in the previous sections and calculate the sample average of an observable over the sample set Σ. The form of these observables for the quantities $f(\tau)$, $\mathcal{F}(\tau)$ and $Z(\beta)$ have also been discussed in the previous sections. By obtaining bounds on the variance of the sample average of these observables, we have shown that the quantities of interest can indeed be obtained up to an ϵ-additive error.



(a) $f(\tau) = \langle x| e^{-\tau H} |y\rangle$. (b) $\mathcal{F}(\tau) = \langle \Phi| e^{-\tau H} |\Phi\rangle$. (c) $Z(\beta) = \sum_x \langle x| e^{-\beta H} |x\rangle$.

Figure 6: Schematic representation of the algorithms used to determine $f(\tau) = \langle x| e^{-\tau H} |y\rangle$, $\mathcal{F}(\tau) = \langle \Phi| e^{-\tau H} |\Phi\rangle$ and $Z(\beta) = \sum_x \langle x| e^{-\beta H} |x\rangle$ as described in this chapter. The black dots represent basis states (from a basis in which Hamiltonian H is piece-wise stoquastic) and each of the red paths connecting several basis states corresponds to a single term in the sum that makes up the quantities $f(\tau)$, $\mathcal{F}(\tau)$ and $Z(\beta)$. For the case of estimating a single matrix element of $e^{-\tau H}$ (depicted in **(a)**), all red paths begin at the same basis state ($|x\rangle$) and end at the same basis state ($|y\rangle$). For the case of estimating the signal $\langle \Phi| e^{-\tau H} |\Phi\rangle$ (depicted in **(b)**), the red paths can begin at different basis states and end at different basis states. When estimating the partition function $Tr\left(e^{-\beta H}\right)$ (depicted in **(c)**), the red paths begin at any basis state ($|x\rangle$) and end at that same basis state – i.e. the red paths form closed loops. We note that all paths should be of length L (which depends on the Trotterization scheme).

IV.7 Use of the Monte Carlo Scheme in a Quantum Computing Setting

We have discussed the use of Theorem IV.3 for efficiently tracking the imaginary-time state evolution of a system with a sufficiently local and piece-wise stoquastic Hamiltonian (by classical means). In addition, we have discussed how one can obtain eigenvalues of the Hamiltonian from this imaginary-time state evolution using the Matrix Pencil Method. We note, however, that the set of operators satisfying the conditions stated for the G_l's in Theorem IV.3 does not solely include local imaginary-time propagators for stoquastic Hamiltonians. The use of Theorem IV.3 can be extended to e.g. determining the overlap between an initial state $|\Phi\rangle$ and a state $G_1 G_2 ... G_L |\Phi\rangle$ to which a set of L (classical) quantum gates has been applied.[5] By determining this overlap, one can investigate the action of particular quantum circuits on a state $|\Phi\rangle$. The set of (well-known) quantum gates that fit in the framework of Theorem IV.3 are the Pauli-X, CNOT, SWAP and Toffoli gates:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad ; \quad \text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad ;$$

$$\text{Pauli-X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad ; \quad \text{TOFF} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{132}$$

In fact, these operators are all symmetric basis-state permutation matrices. Therefore, we note that they form a small (and relatively plain) subset of all operators that in principle fit in the framework of Theorem IV.3. We do note that permutation matrices that are *not* symmetric fit in the framework of Theorem IV.3 as well.

[5] Provided that one can efficiently determine $\frac{\Phi(y)}{\Phi(x)^*}$ for a given x and y, and one can efficiently sample from the distribution $|\Phi(x)|^2$.

The signal $\mathcal{F}(\tau)$, that was discussed in the context of imaginary-time state propagation, lies on the interval $(0, 1]$ for all $\tau \in \mathbb{R}_+$. In contrast, the quantity $\langle\Phi| G_1 G_2 ... G_L |\Phi\rangle$ (where the G_l's are now any of the above gates) lies on the interval $[-1, 1]$: This quantity is the inner product between $|\Phi\rangle$ and another norm-1 state $G_1 G_2 ... G_L |\Phi\rangle$ and can therefore lie anywhere in between -1 and $+1$. Since the classical stochastic scheme from Theorem IV.3 estimates $\langle\Phi| G_1 G_2 ... G_L |\Phi\rangle$ up to an ϵ-additive error, we note that here as well the estimates will only be of high quality if $\langle\Phi| G_1 G_2 ... G_L |\Phi\rangle$ is at least $1/poly(n)$ in norm.

Although one is able to efficiently estimate $\langle\Phi| G_1 G_2 ... G_L |\Phi\rangle$ (where the G_l's are now quantum gates that fit in the framework of Theorem IV.3) by classical means, it is worth discussing the usefulness of this estimate: Suppose one wishes to determine the overlap between an input state and an output state of some quantum circuit (containing arbitrary quantum gates). Some of the gates in this circuit might fit in the framework of Theorem IV.3. In a general setting, one would not only like to determine the action of a set of gates on an initial state, but also have the resulting state physically available so that other gates (which might not fit in the framework of Theorem IV.3) can be applied to it. In that case, one is required to *actually* apply all gates to a qubit register. Only when one arrives at a point *after* which solely quantum gates that fit in the framework of Theorem IV.3 are going to be applied, it is in principle useful to classically estimate the quantity discussed here.

A more elaborate and general discussion of the ability to simulate quantum circuits by means of classical stochastic schemes is given in [35]. In addition, the author identifies the crucial steps in Simon's and Shor's algorithm that ensure the exponential speed-ups and discusses instances in which these algorithms would be efficiently simulatable by classical means.

IV.7.1 (Proposing) A Hybrid Algorithm for Estimating Eigenvalues of Stoquastic Hamiltonians

We have noted in Chapter II that the quantum-phase-estimation-based digital quantum simulation scheme discussed there is idealized for a number of reasons. The implementation of the quantum simulation scheme on actual current *noisy intermediate-scale quantum* (NISQ) devices would reduce its performance compared to an ideal implementation. Due to the resemblance between the quantum-phase-estimation-based digital quantum simulation scheme and the classical Monte Carlo scheme discussed in the current chapter, we propose the exploration of hybrid algorithms to alleviate the computational stress on current NISQ devices. One could try to design algorithms that make use of both the classical scheme and the quantum scheme to e.g. attempt reducing the number of physical qubits (and their coherence times) required for running a particular simulation algorithm.

V Ising Chain in a Transverse Field

To apply the methods developed thus far to a physical system, we consider the Ising chain in a transverse field in this section. This system has been extensively studied ([8]) and will serve as a proof-of-principle framework for the Monte Carlo algorithm developed here. The system consists of spin-$\frac{1}{2}$ particles at fixed positions (on a lattice) interacting via an Ising interaction. They are exposed to an external magnetic field that is applied in the transverse direction. The system exhibits a quantum phase transition (at $T = 0$) as a function of a dimensionless transverse field variable g at $g = g_c = 1$. Its Hamiltonian is given by:

$$H = -J\Big(\sum_i \sigma_i^z \sigma_{i+1}^z + g\sum_i \sigma_i^x\Big), \tag{133}$$

where $J > 0$ (for a ferromagnetic interaction) and $g \geq 0$, and H is thus stoquastic in the standard basis. Note that the field can be chosen to point in the x-direction without loss of generality: The Hamiltonian can be transformed to $\tilde{H} = UHU^\dagger$ by the unitary transformation $U = \bigotimes_i exp\big(\frac{i\theta\sigma_i^z}{2}\big)$, which alters the direction of the field in the transverse plane while preserving the spectrum.

In a (second-order) phase transition such as the paramagnetic to ferromagnetic transition of a two-dimensional Ising model, the phase transition occurs as a function of *temperature*: The system is in the paramagnetic phase (with a net zero magnetization) above the Curie temperature, and in the ferromagnetic phase (with a net non-zero magnetization) below the Curie temperature. The equilibrium configuration of a system is given (at any temperature) by the minimum of the Helmholtz free energy. Above the Curie temperature, the Helmholtz free energy exhibits a single minimum as a function of magnetization (at zero magnetization). Below the Curie temperature, the Helmholtz free energy exhibits two minima as a function of magnetization (both at non-zero magnetization). We note that throughout a process of going from $T > T_C$ to $T < T_C$, the equilibrium configuration clearly changes, while the ground state of the system remains the same. This is due to the fact that the ground state of the system is a property of its (temperature independent) Hamiltonian H. This behaviour of the two-dimensional Ising lattice differs from that of a transverse-field Ising chain system in the following sense: For the transverse-field Ising chain, a phase transition occurs at $T = 0$ as a function of the transverse field strength g (instead of as a function of temperature). Since the Helmholtz free energy is defined to be $F \equiv E - TS$ (where E is the internal energy and S the entropy of the system), at $T = 0$ the minimum of the free energy coincides with the ground state (for any g). If the equilibrium configuration changes (due to a phase transition), we thus conclude that this should be directly related to a change in the ground state of the system. This is indeed the case for the transverse-field Ising chain.

The phase transition at $T = 0$ as a function of g can be characterized by the ground states in two parameter limits:

- **Strong-coupling limit** ($g \gg g_c$): In this limit, the Hamiltonian is dominated by the field terms and the ground state is given by $|GS\rangle_{g>>g_c} \approx |+\rangle^{\otimes n}$. This corresponds to a paramagnetic (disordered) phase. The p-particle excitations correspond to states $|-\rangle_{q_1}|-\rangle_{q_2}...|-\rangle_{q_p}\prod_{i\neq q_1,q_2,...,q_p}|+\rangle_i$, which equals the ground state with spin flips at p (generally non-adjacent) positions $q_1, ..., q_p$ along the chain. These p-particle excited states are $\binom{n}{p}$-fold degenerate.

- **Weak-coupling limit** ($g \ll g_c$): In this limit, the Hamiltonian is dominated by the Ising interaction terms and the ground state is given by either $|GS\rangle_{g<<g_c} \approx |0\rangle^{\otimes n}$ or $|GS\rangle_{g<<g_c} \approx |1\rangle^{\otimes n}$ – i.e. the spins will be either all pointing up or all pointing down. The system now resides in a ferromagnetic (ordered) phase. The excitations w.r.t. the ground state correspond to domain walls separating ferromagnetic regions of opposite spin: The lowest energy excitation corresponds to the state $|\ldots 00000111\ldots\rangle$ (a single domain wall), the second-to-lowest energy excitation corresponds to the state $|\ldots 11100011\ldots\rangle$ (two domain walls) etc. These p domain wall excited states are approximately $2\binom{n}{p}$-fold degenerate. If one imposes periodic boundary conditions, only even p values correspond to valid excitations.

As the system moves from being in the strong coupling limit to being in the weak coupling limit, it crosses the $g = g_c$ critical point and the Z_2-symmetry is spontaneously broken (the system ends up in either $|0\rangle^{\otimes n}$ or $|1\rangle^{\otimes n}$). This second order phase transition between the two coupling regimes is accompanied by an

avoided level-crossing at $g = g_c$ between the ground state and an excited state, the sharpness of which progressively increases with system size and leads to a point of non-analyticity (in the ground state energy as a function of g) at g_c in the infinite size limit.

Since eq. 133 can be written as $H = \sum_i^N H_i$, where $N = n\big(= poly(n)\big)$ and $H_i = -J\big(\sigma_i^z \sigma_{i+1}^z + g\,\sigma_i^x\big)$ is 2-local and stoquastic in the standard basis for $g > 0$, Theorem IV.3 ensures that a Trotterized version of $\langle\Phi|\, e^{-\tau H}\, |\Phi\rangle$ (where $|\Phi\rangle = \sum_x \Phi(x)\,|x\rangle$, $\Phi(x) \in \mathbb{R}$ and $\tau \geq 0$) can be efficiently estimated with absolute error $\epsilon = \frac{1}{poly(n)}$, if the number of local propagation operators in the Trotter decomposition (L) is at most $poly(n)$. To run the Monte Carlo algorithm that is used to obtain this estimate, the imaginary-time propagation operator $e^{-\tau H}$ must be decomposed (by means of Trotterization) in terms of the local propagation operators $e^{-a_l \tau/M\, H_i}$. These local propagation operators corresponding to eq. 133 are given by:

$$e^{-\tilde{\tau} H_i} = \begin{pmatrix} \frac{\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} + \cosh(\lambda\tilde{\tau}) & 0 & \frac{g\,\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} & 0 \\ 0 & \frac{-\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} + \cosh(\lambda\tilde{\tau}) & 0 & \frac{g\,\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} \\ \frac{g\,\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} & 0 & \frac{-\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} + \cosh(\lambda\tilde{\tau}) & 0 \\ 0 & \frac{g\,\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} & 0 & \frac{\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} + \cosh(\lambda\tilde{\tau}) \end{pmatrix}, \tag{134}$$

where $\lambda = J\sqrt{1+g^2}$ and $\tilde{\tau} = a_l \tau/M$. For $g \geq 0$, this operator is element-wise non-negative but reducible. By performing a basis transformation corresponding to a permutation of the 2-site standard basis states $|01\rangle$ and $|10\rangle$, we obtain a block-diagonal representation of $e^{-\tau H_i}$, where the blocks along the diagonal are irreducible (and non-negative) themselves:

$$e^{-\tilde{\tau} H_i} = \begin{pmatrix} \frac{\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} + \cosh(\lambda\tilde{\tau}) & \frac{g\,\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} & 0 & 0 \\ \frac{g\,\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} & \frac{-\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} + \cosh(\lambda\tilde{\tau}) & 0 & 0 \\ 0 & 0 & \frac{-\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} + \cosh(\lambda\tilde{\tau}) & \frac{g\,\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} \\ 0 & 0 & \frac{g\,\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} & \frac{\sinh(\lambda\tilde{\tau})}{\sqrt{1+g^2}} + \cosh(\lambda\tilde{\tau}) \end{pmatrix}. \tag{135}$$

We identify the sets of states (in the 2-site standard basis) connected through non-zero elements of $e^{-\tau H_i}$ as: $S^{(1)} = \{|00\rangle, |10\rangle\}$ and $S^{(2)} = \{|01\rangle, |11\rangle\}$. After setting the lowest eigenvalue of each H_i to zero, the eigenvalues of the propagation operator $e^{-\tau H_i}$ lie in the interval $(0, 1]$ on the real axis.

The signal to be analyzed as a function of imaginary time τ to obtain the eigenvalues of H is:

$$\mathcal{F}(\tau) = \langle\Phi|\, e^{-\tau H}\, |\Phi\rangle \approx \langle\Phi|\, G_1 G_2 ... G_L\, |\Phi\rangle\,, \tag{136}$$

where $G_t = e^{-a_l \tau/M\, H_i}$, and a_l and $L \leq poly(n)$ depend on the Trotterization scheme. For a first-order Trotterization scheme, $L = MN$ and $a_l = 1$, $\forall l$. Due to the fact that the set $\{H_i\}$ can be divided into two subsets such that within each subset all terms commute, the application of the operators can be parallelized such that only $2M$ time slices are required (i.e. a checkerboard decomposition). The setting described here is a particular case of the setting described in Theorem IV.3 since the form of G_l is l-independent – i.e. the bond Hamiltonian H_i is the same for all i.[6] By identifying the block matrices along the diagonal of G_l ($G_l^{(1)}$ and $G_l^{(2)}$) and finding their (largest) positive eigenvalue $\lambda_l^{(1)}$ and $\lambda_l^{(2)}$, and their associated positive eigenstates $|\phi\rangle_l^{(1)}$ and $|\phi\rangle_l^{(2)}$ (which are guaranteed to exists due to Theorem IV.1), the Monte Carlo scheme described in the previous chapter can be constructed. In the remainder of this section we investigate several aspects of the Monte Carlo scheme by applying it to the transverse field Ising model, and in particular to reconstructing parts of its spectrum.

The spectrum of the transverse field Ising model is depicted in Figure 7 (when setting the smallest eigenvalues of all H_i's to zero). It is worth considering in some more detail what happens when one sets the smallest eigenvalues of all H_i's to zero. The Hamiltonian $H = \sum_i H_i$ has eigenvalues $\{E_j\}$ and eigenstates $\{\psi_j\}$:

$$H\,|\psi_j\rangle = E_j\,|\psi_j\rangle\,,\ \ \forall j. \tag{137}$$

[6] Efficiently obtaining an absolute ϵ-estimate $\mathcal{F}(\tau)$ is equally possible for a site-dependent Ising interaction strength J_i and transverse field strength g_i (as long as $g_i \geq 0\ \forall i$ in the chosen basis).

When setting the smallest eigenvalue of each H_i to zero, the 'new' Hamiltonian becomes $\sum_i \big(H_i - \lambda_{min}(H_i)I\big)$:

$$\sum_i \big(H_i - \lambda_{min}(H_i)I\big)\,|\psi_j\rangle = H\,|\psi_j\rangle - \big(\sum_i \lambda_{min}(H_i)\big)I\,|\psi_j\rangle = \big(E_j - \sum_i \lambda_{min}(H_i)\big)\,|\psi_j\rangle\,,\ \forall j. \tag{138}$$

We thus conclude that the 'new' Hamiltonian $\sum_i \big(H_i - \lambda_{min}(H_i)I\big)$ has the same set of eigenstates $\{|\psi_j\rangle\}$ as H, but its spectrum is shifted by an amount $\sum_i \lambda_{min}(H_i)$ to become $\{E_j - \sum_i \lambda_{min}(H_i)\}$. We note that in the case of the transverse-field Ising chain, the quantity $\sum_i \lambda_{min}(H_i)$ is g-dependent. All eigenvalues at a fixed g are thus shifted by the same amount when setting the smallest eigenvalue of each H_i to zero, but this amount does vary with the field strength g.

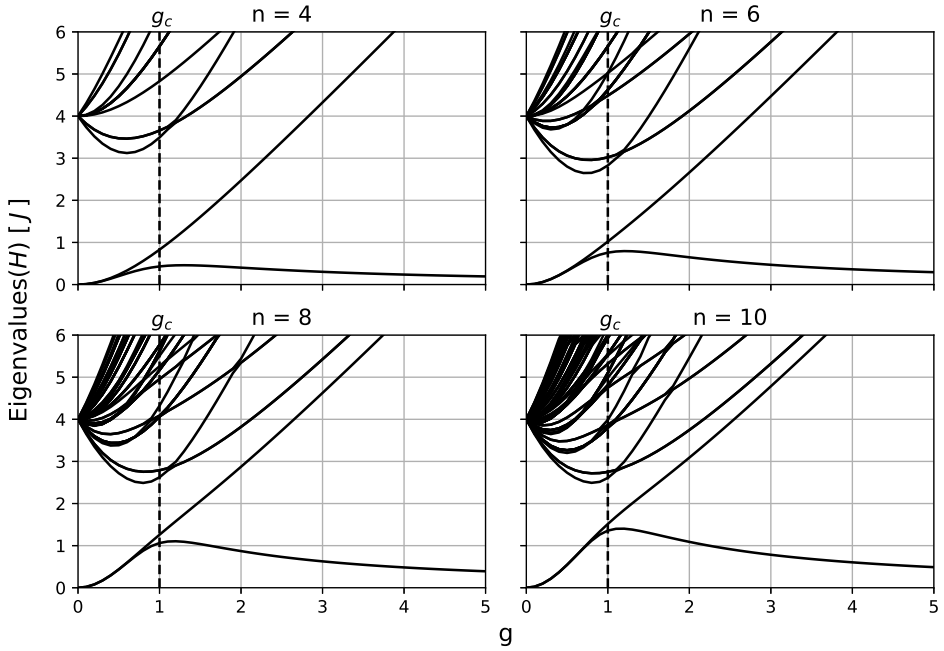


Figure 7: Eigenspectrum of the Ising chain in a transverse field (obtained by direct diagonalization of H) as a function of the dimensionless field strength g for several values of system size. The lowest eigenvalue of all bond Hamiltonians H_i is set to zero. This allows for direct observation of the extent to which Hamiltonian H is frustrated: The system is maximally frustrated at $g = g_c = 1$ and frustration free at $g = 0$ and $g \gg g_c$. The ground state degeneracy that is present for $g < g_c$ is seen to be lifted at the critical point $g = g_c$. Additionally, the sharpness of the ground state energy curve as a function of g around $g = g_c$ is seen to increase progressively with system size (the curve is known to display a point of non-analyticity at $g = g_c$ in the infinite size limit [8]).

V.1 Choice of the State $|\Phi\rangle$

A point of particular importance is the choice of state $|\Phi\rangle$: The signal $\mathcal{F}(\tau) = \sum_j e^{-\tau E_j}\big|\langle\Phi|\psi_j\rangle\big|^2$ can be used to obtain a subset of the E_j's for which the overlap of $|\Phi\rangle$ with the associated eigenstates $|\psi_j\rangle$ is sufficiently large. Choices for $|\Phi\rangle$ for which the overlap with a particular eigenstate $|\psi_j\rangle$ will be (nearly) zero, will lead to not being able to determine the associated eigenvalue E_j: Suppose the overlap scales as $\frac{1}{exp(n)}$, then the associated component of the signal to be analyzed is exponentially small in the system size and hence the algorithm is inefficient for obtaining this eigenvalue. Even if the overlap is of size $\frac{1}{poly(n)}$, the eigenvalue E_j might still not be efficiently attainable: This is due to the fact that we obtain a noisy version of the signal $\mathcal{F}(\tau)$ and the magnitude of the decaying signal corresponding to eigenvalue E_j might

become of the same order as the magnitude of the noise (which itself depends on the number of Monte Carlo samples used to obtain $\mathcal{F}$ at point τ). When using the Matrix Pencil Method to determine the decay rates of $\mathcal{F}(\tau)$ as a function of τ, the noise is (mostly) filtered out in the SVD truncation step. It is thus of importance to have prior knowledge about the form of the eigenstates of which one wants to obtain the associated eigenvalues (as it is in the case of quantum-phase-estimation-based quantum simulation). To obtain e.g. the ground state eigenvalues of the transverse field Ising chain Hamiltonian H, one must choose $|\Phi\rangle$ to have considerable overlap with $|+\rangle^{\otimes n}$ for $g \gg g_c$ and with $|0\rangle^{\otimes n}$ or $|1\rangle^{\otimes n}$ for $g \ll g_c$.

In Figure 8, we have depicted the spectrum for the transverse field Ising chain (for $n = 7$) that has been obtained through direct diagonalization. In addition, the quantities $\langle 0|^{\otimes n} H |0\rangle^{\otimes n}$ and $\langle +|^{\otimes n} H |+\rangle^{\otimes n}$ have been plotted as a function of the dimensionless field variable g. Clearly, in the regime $g \ll g_c$, the state $|0\rangle^{\otimes n}$ almost coincides with the ground state. Similarly, in the regime $g \gg g_c$, the state $|+\rangle^{\otimes n}$ almost coincides with the ground state. For ground state calculations, therefore, $|\Phi\rangle$ can be chosen $|0\rangle^{\otimes n}$ and $|+\rangle^{\otimes n}$ in the weak coupling and strong coupling regimes, respectively. It is apparent from Figure 8 that the states $|0\rangle^{\otimes n}$ and $|+\rangle^{\otimes n}$ have significant overlap with states other than the ground state near the point g_c. Therefore, one might in principle use these states as $|\Phi\rangle$ to extract excited state eigenvalues through the aforementioned Monte Carlo procedure. However, since the Hamiltonian has 2^n (possibly degenerate) eigenstates, the states $|0\rangle^{\otimes n}$ and $|+\rangle^{\otimes n}$ might not have sufficiently large overlap with *any* of the excited states for the Monte Carlo scheme to properly extract the associated eigenvalues. Instead, one constructs a state $|\Phi\rangle$ in a systematic manner, such that it is highly likely that it has sufficiently large (at least $1/poly(n)$) overlap with some eigenstates of interest. Below, we construct such a state that has large overlap with the ground state and first excited state of the transverse field Ising model in the strong coupling regime.

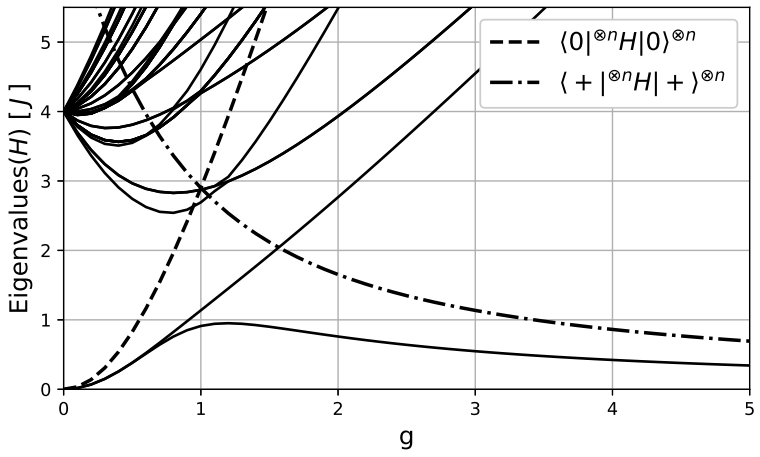


Figure 8: Eigenspectrum of the Ising chain in a transverse field obtained by direct diagonalization for $n = 7$. The expectation values $\langle 0|^{\otimes n} H |0\rangle^{\otimes n}$ and $\langle +|^{\otimes n} H |+\rangle^{\otimes n}$ are plotted as a function of g as well. In the parameter regimes $g \ll g_c$ and $g \gg g_c$, the ground states closely resemble the states $|0\rangle^{\otimes n}$ and $|+\rangle^{\otimes n}$, respectively.

By using the available information about the eigenstates of the transverse field Ising chain, we now construct a state $|\Phi\rangle$ that has at least $\frac{1}{poly(n)}$ overlap between the ground state and the (n-fold degenerate) first excited state in the $g \gg g_c$ regime. We consider the following state:

$$|\Phi\rangle = \frac{1}{\sqrt{2}}\Big(\underbrace{\prod_{i=1}^{n}|+\rangle_i}_{|\psi_{p=0}\rangle} + \sum_{q=1}^{n}\frac{1}{\sqrt{n}}\underbrace{|-\rangle_q \prod_{i\neq q}|+\rangle_i}_{|\psi_{p=1,q}\rangle}\Big). \tag{139}$$

This state is chosen such that $|\langle\Phi|\psi_{p=0}\rangle|^2 = \sum_{q=1}^{n}|\langle\Phi|\psi_{p=1,q}\rangle|^2 = \frac{1}{2}$ – i.e. the $E_{p=0}$ (ground state) and $E_{p=1}$ (excited state) components of the $\mathcal{F}(\tau)$ signal thus have equal magnitudes in the regime $g \gg g_c$. Hereafter we shall call this state $|\Phi_{optimal}(p=0,1)\rangle$. Since the Monte Carlo procedure is going to be ran in the standard basis, we now consider the decomposition of $|\Phi_{optimal}(p=0,1)\rangle$ in terms of standard-basis

states:

$$
\begin{aligned}
|\Phi_{optimal}(p=0,1)\rangle &= \frac{1}{\sqrt{2}}\sum_{q=1}^{n}\Big(\frac{1}{n}\prod_{i=1}^{n}|+\rangle_i + \frac{1}{\sqrt{n}}\,|-\rangle_q \prod_{i\neq q}|+\rangle_i\Big) \\
&= \frac{1}{2}\sum_{q=1}^{n}\Big(\Big(\frac{1}{n}+\frac{1}{\sqrt{n}}\Big)|0\rangle_q + \Big(\frac{1}{n}-\frac{1}{\sqrt{n}}\Big)|1\rangle_q\Big)\prod_{i\neq q}|+\rangle_i \qquad (140)\\
&= \frac{1}{2^{(n+1)/2}}\sum_{q=1}^{n}\Bigg(\Big(\Big(\frac{1}{n}+\frac{1}{\sqrt{n}}\Big)|0\rangle_q + \Big(\frac{1}{n}-\frac{1}{\sqrt{n}}\Big)|1\rangle_q\Big)\sum_{x\in\{0,1\}^n\atop\setminus\{0,1\}_q}|x\rangle\Bigg),
\end{aligned}
$$

where $\sum_{x\in\{0,1\}^n\atop\setminus\{0,1\}_q}|x\rangle$ denotes an equal superposition of $(n-1)$-bit strings that exclude the bit in register q.

As extensively discussed in the previous chapter, an essential feature of the state $|\Phi\rangle=\sum_{x\in\{0,1\}^n}\Phi(x)\,|x\rangle$ (where $\Phi(x)\in\mathbb{R}\ \forall x$ in the current discussion) in our Monte Carlo scheme is that one can efficiently obtain the quantity $\frac{\Phi(y)}{\Phi(x)}$ (for a given $x,y\in\{0,1\}^n$) and one can efficiently sample from the distribution $\big(\Phi(x)\big)^2$. For ground state calculations in the strong-coupling regime, we shall take:

$$
|\Phi\rangle = |+\rangle^{\otimes n} = \frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}|x\rangle\,. \qquad (141)
$$

In that case, $\Phi(x)=\frac{1}{2^{n/2}},\ \forall x$. Therefore, the quantity $\frac{\Phi(y)}{\Phi(x)}$ is efficiently obtainable since it equals unity for any x,y. Furthermore, the distribution $(\Phi(x))^2$ equals $1/2^n$ for any x. One can thus efficiently sample from this distribution since sampling from it corresponds to selecting, uniformly at random, a bit string from $\{0,1\}^n$ (which is a computationally tractable task).

For ground state and first excited state calculations in the strong-coupling regime, we take $|\Phi\rangle = |\Phi_{optimal}(p=0,1)\rangle$. From eq. 140, one can infer the function $\Phi(x)$ ($\{0,1\}^n\rightarrow\mathbb{R}$) that gives the coefficient of the state $|\Phi\rangle$ associated with an n-bit string x:

$$
\Phi(x) = \frac{1}{2^{(n+1)/2}}\Big(\Big(\frac{1}{n}+\frac{1}{\sqrt{n}}\Big)\big(n-|x|\big) + \Big(\frac{1}{n}-\frac{1}{\sqrt{n}}\Big)|x|\Big), \qquad (142)
$$

where $\Phi(x)$ in this case depends only on the Hamming weight $|x|$ of bit string x. We thus conclude that given bit strings x and y, the quantity $\frac{\Phi(y)}{\Phi(x)}$ can be efficiently determined. Furthermore, since $\Phi(x)$ only depends on n and $|x|$, the distribution $(\Phi(x))^2$ also only depends on these quantities. This implies that one can sample from this distribution with resources that scale polynomially in n: First, one draws a Hamming weight $|x|$ from the distribution $(\Phi(x))^2=(\Phi(|x|))^2$. Then, given $|x|$, one constructs at random an n-bit string with this Hamming weight $|x|$. This latter step can be efficiently implemented by starting from some n-bit string with Hamming weight $|x|$ (such as $\{1\}^{|x|}\{0\}^{n-|x|}$) and then randomly and one-by-one assigning the n bits a new position along the string. Obviously, once a bit has been placed at a particular position along the string, this position is no longer available for other bits to be placed at. In that way, an n-bit string with Hamming weight $|x|$ (where $|x|$ has been drawn efficiently from $(\Phi(x))^2$) can be constructed in an efficient way. An n-bit string can thus be sampled from $(\Phi(x))^2$ efficiently for $|\Phi_{optimal}(p=0,1)\rangle$. We thus conclude that for the states $|+\rangle^{\otimes n}$ and $|\Phi_{optimal}(p=0,1)\rangle$, one does not have to reside to methods such as the Metropolis algorithm for sampling from $(\Phi(x))^2$.

In the remainder of this chapter, we investigate several features of our Monte Carlo scheme and, in particular, discuss its performance compared to quantum-phase-estimation-based quantum simulation.

V.2 Monte Carlo and Quantum Phase Estimation Signals

In Figure 9, we have depicted the Monte Carlo signals $\mathcal{F}(\tau)$ for a transverse-field Ising chain of length $n=7$ at $g=4*g_c$ for $|\Phi\rangle=|+\rangle^{\otimes n}$ and $|\Phi\rangle=|\Phi_{optimal}(p=0,1)\rangle$. In addition, the real and imaginary parts of the QPE signal for $|\Phi\rangle=|+\rangle^{\otimes n}$ and $|\Phi\rangle=|\Phi_{optimal}(p=0,1)\rangle$ are displayed. $|\Sigma|$ is set to be 4000. The Trotter variable M is taken to be 60 and the implemented Trotterization scheme is of order 1. Apart from the region near $t=5$ and $\tau=5$, no significant systematic deviation of the signals w.r.t. the exact signals due to Trotterization can be observed. We note that, as expected, the MC signal values lie in the interval $(0,1]$, while the QPE signal values lie in the interval $[-1,1]$.

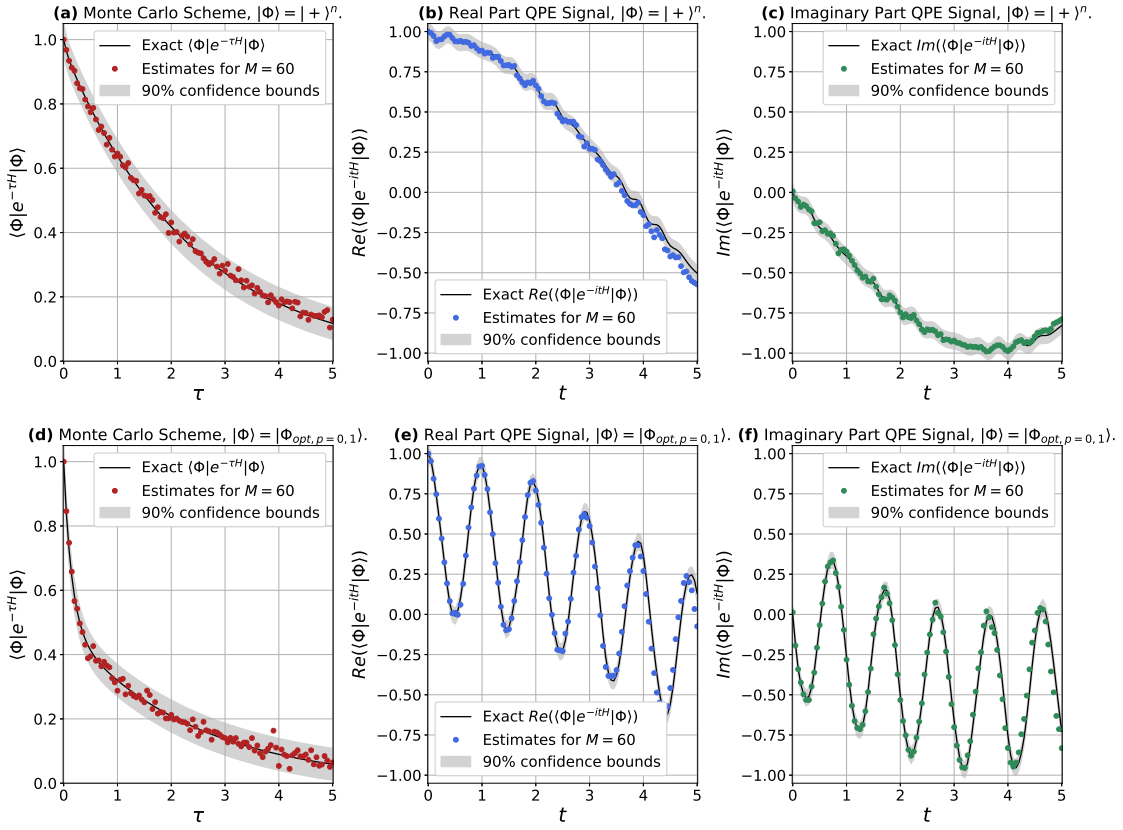


Figure 9: The evolution of the states $|+\rangle^{\otimes n}$ (in **(a)**, **(b)** and **(c)**) and $|\Phi_{optimal,p=0,1}\rangle$ (in **(d)**, **(e)** and **(f)**) of a ferromagnetic Ising chain in a transverse field (for $n = 7$ and $g = 4 * g_c$) is tracked in imaginary time τ (in **(a)** and **(d)**) and in real time t (in **(b)**, **(c)**, **(e)** and **(f)**). The signals shown in **(a)** and **(d)** are obtained by the previously discussed Monte Carlo scheme. The signals shown in **(b)**, **(c)**, **(e)** and **(f)** are obtained by quantum phase estimation (that is inefficiently implemented on a classical computer). In both cases, the time interval is taken to be 5 and time increment has been taken to be 0.05 (i.e. $K = 100$). The Trotter variable is taken to be $M = 60$. $|\Sigma|$ is set to be 4000.

The upper three figures correspond to $|\Phi\rangle = |+\rangle^{\otimes n}$. For this choice of $|\Phi\rangle$, the signals are clearly dominated by a single eigenvalue – i.e. the MC signal decays (approximately) with a single decay rate, and the QPE signals oscillate (approximately) with a single oscillation frequency. The lower three figures correspond to $|\Phi\rangle = |\Phi_{optimal}(p = 0, 1)\rangle$. For this choice of $|\Phi\rangle$, there are two eigenvalues present in the signals. The MC signal is a superposition of a rapidly decaying signal and a slowly decaying signal (both with approximately equal amplitudes). Similarly, the QPE signals correspond to a slowly oscillating signal with a superposed rapidly oscillating signal (both with approximately equal amplitudes). In Table 1, we give the overlaps between the two choices of $|\Phi\rangle$ considered here and their overlaps with the five lowest-lying eigenstates of the transverse-field Ising chain at $g = 4 * g_c$.

	$\lvert\langle\Phi\lvert\psi_0\rangle\rvert^2$	$\lvert\langle\Phi\lvert\psi_1\rangle\rvert^2$	$\lvert\langle\Phi\lvert\psi_2\rangle\rvert^2$	$\lvert\langle\Phi\lvert\psi_3\rangle\rvert^2$	$\lvert\langle\Phi\lvert\psi_4\rangle\rvert^2$
$\lvert\Phi\rangle = \lvert+\rangle^{\otimes n}$	0.97252	$2.9613 \cdot 10^{-30}$	$7.7037 \cdot 10^{-32}$	$6.2400 \cdot 10^{-32}$	$4.9303 \cdot 10^{-32}$
$\lvert\Phi\rangle = \lvert\Phi_{optimal}(p = 0, 1)\rangle$	0.48626	0.48628	$5.2077 \cdot 10^{-31}$	$6.7041 \cdot 10^{-31}$	$3.0814 \cdot 10^{-33}$

Table 1: Overlap between the states $\{|\Phi\rangle = |+\rangle^{\otimes n}, |\Phi\rangle = |\Phi_{optimal}(p = 0, 1)\rangle\}$ and the five lowest-lying eigenstates of the transverse-field Ising chain at $g = 4 * g_c$ (and $n = 7$).

As has been stressed in previous stages of this report, the MC signals have a finite lifetime: After some point in τ, their magnitude will become smaller than the noise amplitude. In addition, it is clear from Figure 9 that the high-energy part of the Monte Carlo signals dies out rather quickly (the speed with which it dies out obviously depends on the excited-state energy). If one thus wants to efficiently capture high-energy eigenvalues, one should choose the measurement interval to lie at sufficiently small τ values.

By analyzing the signals depicted in Figure 9 by means of the Matrix Pencil Method, one can estimate the lowest-lying eigenvalue (upper three figures) and the two lowest-lying eigenvalues (lower three figures) of the transverse-field Ising chain at $g = 4 * g_c$. In Section V.4, these eigenvalues are estimated for several values of g to (partially) reconstruct the spectrum of the transverse-field Ising chain.

In order to explore aspects of the Trotter error that is imposed on the Monte Carlo and QPE signals, we have determined the absolute error of these (noisy) signals w.r.t. their exact counterparts for the transverse-field Ising chain of length $n = 6$ at $g = 4 * g_c$. In particular, we have implemented the N-term first order and the $\Gamma = 2$-term (checkerboard) first order decomposition. We have depicted the absolute errors of the signal estimates for $|\Phi\rangle = |+\rangle^{\otimes n}$ as a function of the Trotter variable M in Figure 10, together with the upper bounds derived in the introductory chapter of this report. The variables τ and t are fixed and both equal 4. The absolute errors can be seen to be well below their upper bounds. Despite the fact that a $\Gamma = 2$-term (checkerboard) decomposition provides an advantage in terms of simulation speed (due to the associated parallelization opportunities), it does not provide a significant advantage in terms of the absolute errors in this case. Furthermore, the obtained absolute errors do not notable differ between the imaginary-time (MC) and real-time (QPE) propagation settings. We note that the noise magnitude is (approximately) constant as a function of M; the apparent increase of the noise magnitude (the error 'spread') at large M is due to the log-scale on the y-axis.

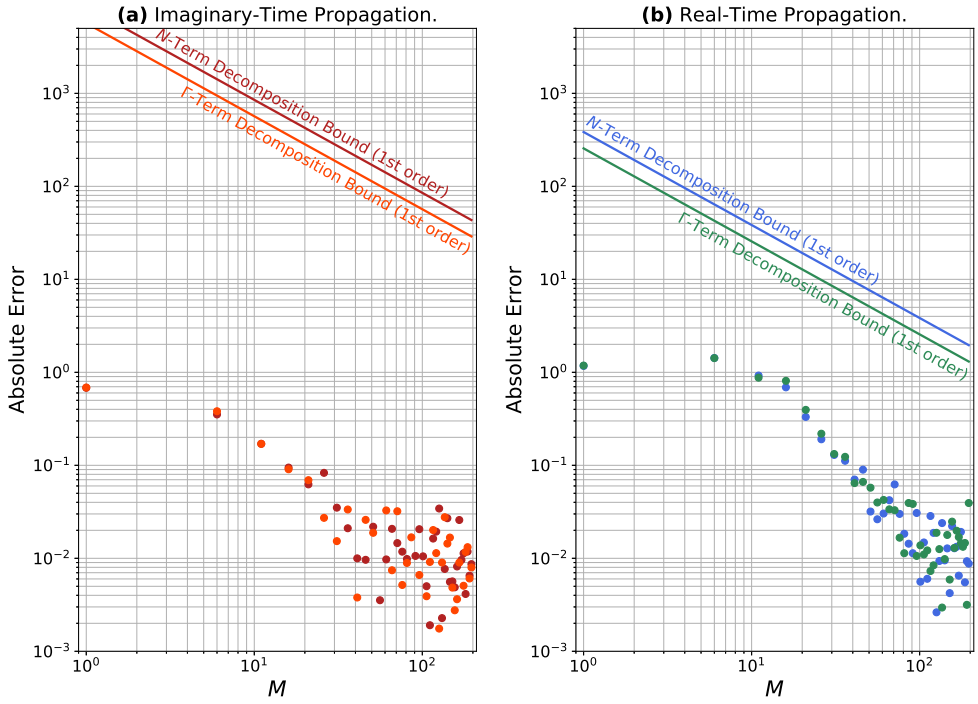


Figure 10: Comparison of the absolute Trotter error as a function of the Trotter variable M between the imaginary-time case and the real-time case. The noisy and Trotterized versions of the signals $\langle+|^{\otimes n} e^{-\tau H} |+\rangle^{\otimes n}$ (in **(a)**) and $\langle+|^{\otimes n} e^{-itH} |+\rangle^{\otimes n}$ (in **(b)**) for a ferromagnetic Ising chain in a transverse field (for $g = 4 * g_c$ and $n = 6$) are evaluated for a fixed value of the variables τ and t (both equal 4) and several values of M. The Trotterization schemes that have been employed to determine the signal errors are the first-order N-term and the first-order $\Gamma(= 2)$-term schemes. The associated error bounds that were derived in Section I.4 of this report are included in matching colors.

V.3 Temporal Dependence of the Noise on the MC and QPE Signals

We have argued in Chapter IV and Section of this report that the Monte Carlo and QPE signals ($\langle\Phi| e^{-\tau H} |\Phi\rangle$ and $\langle\Phi| e^{-itH} |\Phi\rangle$, respectively) are obtained up to an ϵ-additive error and an efficiently suppressible Trotter error. The former effectively corresponds to noise. We have derived that the variance of a Monte Carlo estimate of $\mathcal{F}(\tau)$ (at some τ) is upper bounded by $1/|\Sigma|$, where $|\Sigma|$ is the number of samples taken to estimate the signal at each instance in time. In addition, we have shown that the variance of the QPE estimate is lower bounded by $1/|\Sigma|$ and upper bounded by $2/|\Sigma|$. In this section, we explore these features numerically in the setting of the transverse-field Ising chain. In particular, we shall numerically study how the variance behaves *within* the aforementioned bounds for both the Monte Carlo and the QPE signal.

In Chapter II, we have shown that the variance of the estimate of the real part of the QPE signal, and that of the imaginary part of the QPE signal respectively have the following time-dependence:

$$\frac{1}{|\Sigma|}\Big(1 - \big(Re(\langle\Phi| e^{-itH} |\Phi\rangle)\big)^2\Big)\,; \quad \frac{1}{|\Sigma|}\Big(1 - \big(Im(\langle\Phi| e^{-itH} |\Phi\rangle)\big)^2\Big). \tag{143}$$

The variance of the estimates of the complete QPE signal were found to have the following time-dependence: $\frac{1}{|\Sigma|}\Big(2 - |\langle\Phi| e^{-itH} |\Phi\rangle|^2\Big)$. In particular, we have shown that if the QPE signal essentially contains a single component (i.e. a single eigenvalue), then the variance of the complete signal is constant in time and given by $1/|\Sigma|$. If the QPE signal essentially contains two components, then the variance of the complete signal oscillates in time between $1/|\Sigma|$ and $2/|\Sigma|$ with a frequency that equals the difference between the two eigenvalues associated with these components.

In Chapter IV, we have shown that the variance of the MC signal is upper bounded by the following τ-dependent quantity: $\big(1 - \mathcal{F}(\tau)^2\big)/|\Sigma|$, which indeed leads to the τ-independent upper bound of $1/|\Sigma|$.

First, we set $|\Phi\rangle$ equal to $|+\rangle^{\otimes n}$. For this choice of $|\Phi\rangle$ and $g = 4 * g_c$, the Monte Carlo and QPE signals essentially contain only a single component: Namely, that of the ground state. We examine the variance in the MC and QPE signals as follows: We estimate the MC and QPE signals at some τ and t using $|\Sigma| = 200$. We run this simulation for 500 times for the same τ and t and determine the variance of the estimates over this set of size 500. We then repeat this procedure for many different values τ and t, to examine how the variance changes with τ and t. The results for $|\Phi\rangle = |+\rangle^{\otimes n}$ are depicted in Figure 11.

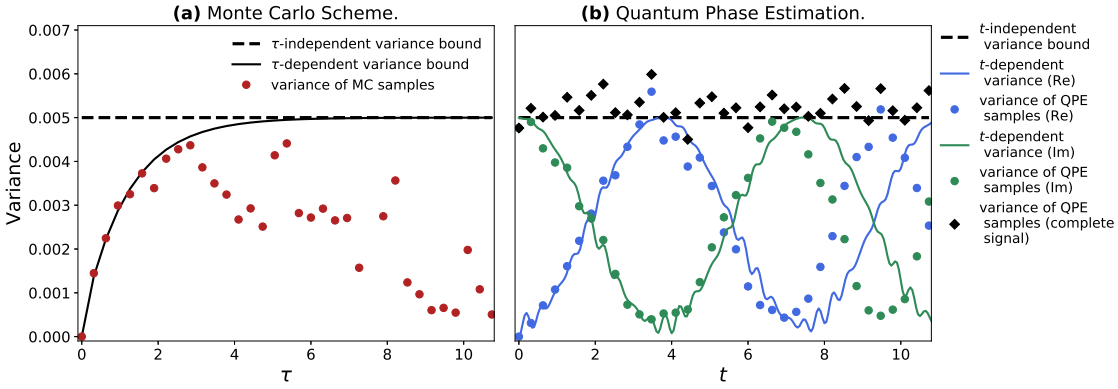


Figure 11: The variance in the Monte Carlo samples and quantum phase estimation samples are plotted as a function of the (imaginary) time over which the system has evolved. Each sample is obtained by means of a $|\Sigma| = 200$ simulation, and the variance is determined of 500 of such samples at each instance in time. The data obtained is from a simulation of a ferromagnetic Ising chain in a transverse field (for $g = 4 * g_c$, $n = 7$, $M = 60$ and $|\Phi\rangle = |+\rangle$). In **(a)** (MC scheme), the time-independent variance bound $(1/|\Sigma|)$ is plotted along with the time-dependent bound $(\big(1 - \mathcal{F}(\tau)^2\big)/|\Sigma|)$ and the sample variances obtained by running the MC scheme. In **(b)** (QPE), the time-independent variance bound $(1/|\Sigma|)$ is plotted along with the time-dependent variances of the real- and imaginary part of the QPE signal. The sample variances (of the real- and imaginary part of the signal, as well as those of the complete signal) that are obtained by running the QPE algorithm are plotted as well.

Figure 11 numerically confirms several aspects: The variances of the MC signal, of the real part of the QPE signal and of the imaginary part of the QPE signal are upper bounded by $1/|\Sigma|$. Furthermore, the variance of the MC signal is upper bounded by a τ-dependent quantity that goes to zero as $\tau \to 0$ (since $\mathcal{F}(\tau = 0)$ equals unity). Furthermore, since the QPE signal essentially only consists of a single (ground state) component, the variance of the complete QPE signal remains approximately constant (at $1/|\Sigma|$) as a function of time. We note clear deviations of the variances of the real and imaginary parts of the QPE signal from their exact counterparts at large times. This is due to the Trotter error, which becomes more pronounced at large times. We note that this deviation is not present for the variance of the complete signal: The deviation that is imposed on the variance of the real and imagnary parts of the QPE signal due to the Trotter error in this case is such that the variance of the complete signal is unaffected by the Trotter error.

We now take $|\Phi\rangle = |\Phi_{optimal}(p = 0, 1)\rangle$ to numerically study the behaviour of the variance of the MC and QPE estimates when the signals contain two components: Namely, the ground state and first-excited state components. The variance of the MC and QPE signal estimates – including their bounds – are depicted in Figure 12. We note that the scales of the y-axis and x-axis in Figure 12 are different from those in Figure 11. For the time interval considered in Figure 12, the Trotter error is not as pronounced as it is in the large time regime of Figure 11.

The τ-dependent upper bound $((1 - \mathcal{F}(\tau)^2)/|\Sigma|)$ on the variance of the MC signal estimates converges to the τ-independent bound $(1/|\Sigma|)$ faster than in Figure 11. This is due to the fact that the signal $\mathcal{F}(\tau)$ decays more rapidly since it now contains a high-energy component as well. Since the signals now essentially contain two components, the variance of the complete QPE signal estimates will no longer be constant as a function of time. In fact, it oscillates between its lower bound $(1/|\Sigma|)$ and upper bound $(2/|\Sigma|)$. The frequency with which it oscillates equals the difference between the two eigenvalues present in the signal (the solid black curve in Figure 12b is $\frac{1}{|\Sigma|}(3/2 - 1/2\cos((E_1 - E_0)t))$.

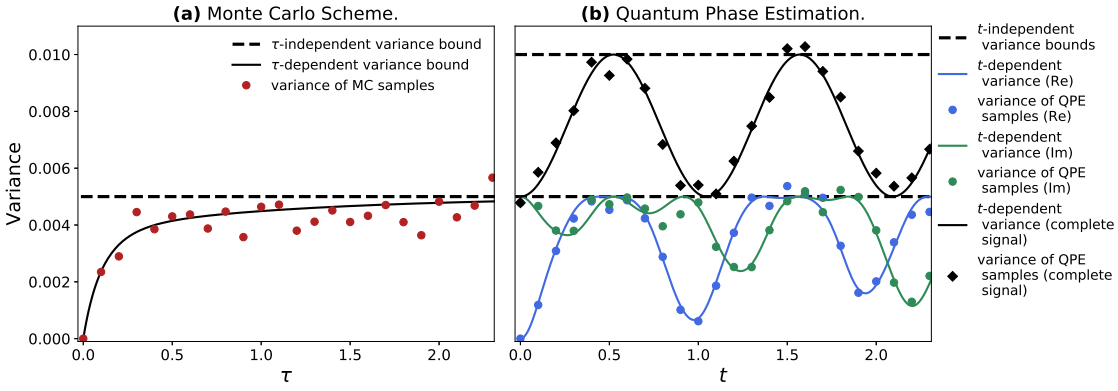


Figure 12: The variance in the Monte Carlo samples and quantum phase estimation samples are plotted as a function of the (imaginary) time over which the system has evolved. Each sample is obtained by means of a $|\Sigma| = 200$ simulation, and the variance is determined of 500 of such samples at each instance in time. The data obtained is from a simulation of a ferromagnetic Ising chain in a transverse field (for $g = 4 * g_c$, $n = 7$, $M = 60$ and $|\Phi\rangle = |\Phi_{opt,p=0,1}\rangle$). In **(a)** (MC scheme), the time-independent variance bound $(1/|\Sigma|)$ is plotted along with the time-dependent bound $\left((1 - \mathcal{F}(\tau)^2\right)/|\Sigma|)$ and the sample variances obtained by running the MC scheme. In **(b)** (QPE), the time-independent variance lower bound $(1/|\Sigma|)$ and upper bound $(2/|\Sigma|)$ are plotted along with the time-dependent variances of the real- and imaginary part of the QPE signal. The sample variances (of the real- and imaginary part of the signal, as well as those of the complete signal) that are obtained by running the QPE algorithm are plotted as well. Note that the (imaginary-) time interval over which the variance is tracked is smaller than in Figure 11.

V.4 Estimates of the Spectrum of the Transverse-Field Ising Chain

Now that several features of the Monte Carlo (i.e. *imaginary-time*) and QPE (i.e. *real-time*) signals have

been numerically established, we turn to extracting eigenvalues from these signals by means of the Matrix Pencil Method.

First, we consider ground state estimates from the Monte Carlo signal at $g = 3 * g_c$. We set $|\Phi\rangle = |+\rangle^{\otimes n}$ and determine $\langle\Phi| e^{-\tau H} |\Phi\rangle$ for $\tau \in [0, 5]$ and several values of the Trotter variable M. This is done for both $|\Sigma| = 200$ and $|\Sigma| = 4000$. Figure 13 depicts the relative error in the MC signal $\langle\Phi| e^{-\tau H} |\Phi\rangle$ compared to its noiseless and Trotter-error-free counterparts, as well as the relative error in the associated estimates of the ground state energy. For both the $|\Sigma| = 200$ and $|\Sigma| = 4000$ cases, the systematic deviation of the MC signal w.r.t. its exact counterpart can be seen to be the largest in the small M and large τ limit. As described in the introductory chapter of this report, this can be attributed to the Trotter error. The τ interval for which the Trotter error is negligible becomes progressively larger as the Trotter variable M is increased. Furthermore, it is clear that the signal for which $|\Sigma| = 200$ is noisier than the signal for which $|\Sigma| = 4000$.

The bottom part of Figure 13 displays the relative error in the ground state energy estimates obtained from the $|\Sigma| = 200$ and $|\Sigma| = 4000$ signals for several values of M. At $M = 1$, the relative error is about 100%. At $M = 25$, the relative error has reduced to a few percent. We note that the noise magnitude in the case of the $|\Sigma| = 4000$ signal is about $\sqrt{\frac{4000}{200}} \approx 4.47$ times smaller than in the case of the $|\Sigma| = 200$ signal. Despite this difference in noise magnitude, it is apparent from Figure 13 that the associated ground state energy estimates follow approximately the same trend as a function of M. The Matrix Pencil Method thus suppresses the noise relatively efficiently when determining this ground state energy.

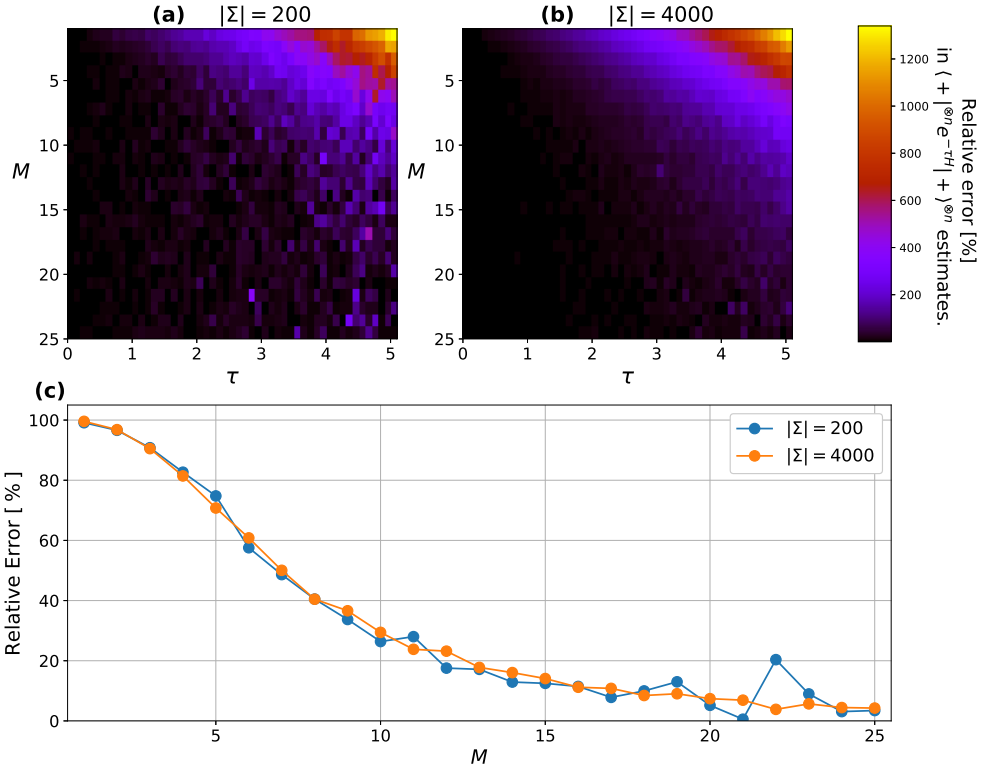


Figure 13: Simulation of the ferromagnetic Ising chain in a transverse field (for $g = 3 * g_c$ and $n = 7$). In **(a)** and **(b)**, the relative error [%] of the MC signal $\mathcal{F}(\tau)$ (w.r.t. its noiseless and Trotter-error-free counterpart) is plotted for $|\Phi\rangle = |+\rangle^{\otimes n}$ as a function of τ and the Trotter variable M (for $K = 50$). The number of samples at each instance in τ (i.e. $|\Sigma|$) is taken to be 200 in **(a)** and 4000 in in **(b)**. In **(c)**, the associated estimates of the ground state energy from the signals $\mathcal{F}(\tau)$ are plotted as a function of the Trotter variable M, for both the $|\Sigma| = 200$ and the $|\Sigma| = 4000$ case. The truncation factor (defined in Chapter III) is set to be 10%.

We now set $|\Phi\rangle = |\Phi_{optimal}(p = 0, 1)\rangle$ and attempt to estimate simultaneously the ground-state energy and the first-excited-state energy in the region $g > g_c$, thereby reconstructing part of the spectrum of the transverse-field Ising chain. We estimate the eigenvalues using both the Monte Carlo signals and the QPE signals, as well as their exact – i.e. Trotter-error-free and noiseless – counterparts. The Trotter variable is set to be $M = 60$ and the number of measurement points in real or imaginary time equals $K = 100$. We estimate the eigenvalues for several values of $g(> 1)$ using the Matrix Pencil Method and set the truncation factor to 2%. The results are depicted in Figure 14.

At the start of this chapter, the degeneracies of the eigenvalues of the transverse-field Ising chain in the two coupling regimes were discussed. These degeneracies are such that in the strong-coupling regime, there are approximately n distinct eigenvalues, while in the weak-coupling regime, there are approximately $n/2$ distinct eigenvalues. In Chapter III, it was discussed that if the MC or QPE signals were *not* noisy, then the quantities $e^{-\Delta\tau E_j}$ and $e^{-i\Delta t E_j}$ (respectively) can be obtained exactly if $K \geq 2R$ (where R is the number of distinct $e^{-\Delta\tau E_j}$'s and $e^{-i\Delta t E_j}$'s). R is thus $poly(n)$ in the weak-coupling and strong-coupling regimes. Therefore, in these regimes and in the noiseless setting, all distinct eigenvalues can be obtained exactly since K can efficiently be chosen to be $poly(n)$ as well. In a more general setting, however, the number of distinct eigenvalues will *not* be $poly(n)$ and instead will be $exp(n)$. Therefore, even in the noiseless setting, the $exp(n)$ distinct eigenvalues cannot be obtained exactly in an efficient manner (since K can be at most $poly(n)$). To make for a fair discussion, we shall therefore consider a scenario for which $K < 2R$: In the intermediate regime, which includes the regime that is considered in Figure 14, the number of distinct eigenvalues for a chain of length $n = 7$ will be 53 just below g_c and 54 just above g_c.[7] If we thus take $K = 100$, then $K < 2R$ will hold. However, as discussed in earlier stages of this report, one can make sure that the *effective* number of distinct eigenvalues in the signal will be rather small by carefully choosing their associated coefficients in the signal (i.e. by carefully choosing the state $|\Phi\rangle$).

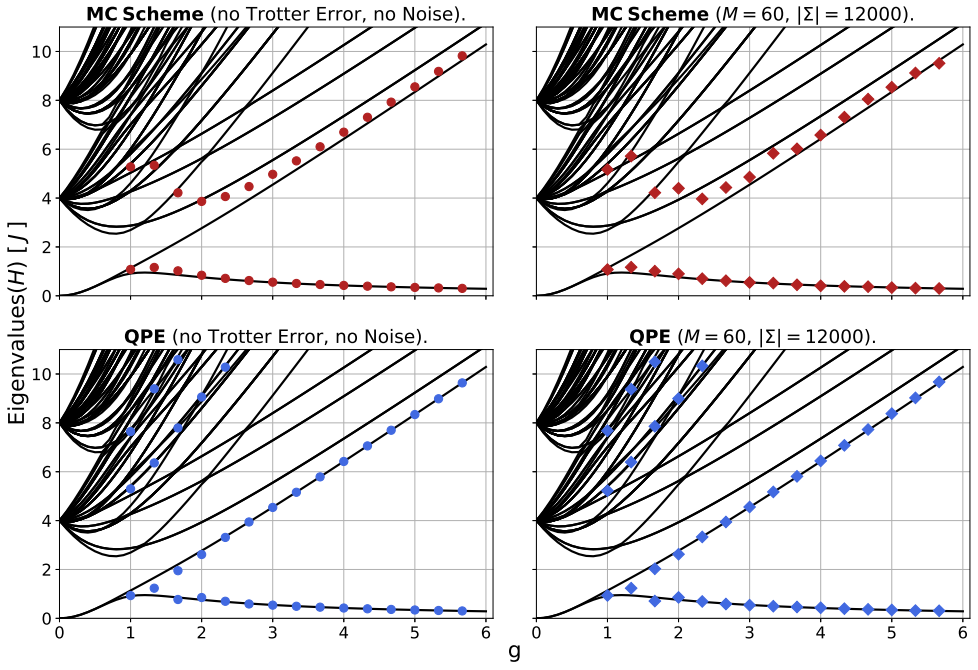


Figure 14: Spectral estimates of the ferromagnetic Ising chain in a transverse field (in the $g \gtrsim 1$ regime for $n = 7$), which are obtained by analyzing the evolution of $|\Phi_{optimal,p=0,1}\rangle$ in real time (QPE) and imaginary time (MC scheme) using the Matrix Pencil Method. In the MC setting and QPE setting we consider the Trotterized and noise-infected signals, as well as their noiseless and Trotter-error-free counterparts. In all instances, the time interval is taken to be 2.5 and time increment has been taken to be 0.025 (i.e. $K = 100$). The truncation factor is chosen to be 2%.

[7] Note that the ground state degeneracy that is present at $g < g_c$ is lifted at $g > g_c$.

We infer from Figure 14 that the excited-state eigenvalue is extracted from the QPE signal with a higher accuracy than from the MC signal. That is, the noise is suppressed more effectively for the QPE signal than it is for the MC signal when determining the excited-state eigenvalue. Furthermore, it is clear that the eigenvalues are extracted from the MC signal quite well in the large-g regime, while in the $g \gtrsim 1$ regime the excited-state eigenvalue is not extracted well – even for the noiseless MC signal. The reason for this is the following: The state $|\Phi_{optimal}(p=0,1)\rangle$ is known to have a large overlap with the ground state $|\psi_0\rangle$ and the first-excited state $|\psi_1\rangle$ in the large-g regime (see Table 1 for $g = 4 * g_c$). For smaller values of g (while $g \gtrsim 1$), the overlap of the state $|\Phi_{optimal}(p=0,1)\rangle$ with other higher-lying eigenstates of H becomes progressively larger. These higher-lying eigenstates correspond to components in the MC signal that decay rapidly and therefore will quickly have a magnitude that is smaller than that of the noise. This leads to the fact that in the SVD truncation step of the Matrix Pencil Method, effectively only two components are extracted: One ground-state component, and one mixed excited-state component. The eigenvalue that is then associated with the mixed excited-state component is some type of average of the actual eigenvalues of the excited-state components. One can resolve this by truncating the SVD less severely – i.e. choosing a smaller truncation factor. This will be demonstrated numerically shortly. This does, however, lead to a worse suppression of the noise and the quality of the eigenvalues estimates (in the large-g regime) will therefore be lower. This, in turn, can be resolved by taking the sample size $|\Sigma|$ larger, thereby reducing the noise magnitude in the first place.

We note that the effect of averaging of the excited state eigenvalues is less dominantly present in the case of the QPE signal: Although the state $|\Phi_{optimal}(p=0,1)\rangle$ will similarly have significant overlap with other higher-lying eigenstates of H, their associated components in the QPE signal will *not* die out. Therefore, the number of singular values in the SVD above the truncation factor threshold will be larger. As a result, the excited state eigenvalues are resolved more accurately.

To gain more insight into the averaging process as a result of a (too) large truncation factor, we have estimated the eigenvalues from the QPE signal for a transverse-field Ising chain of length $n = 7$ at $g = 4 * g_c$ for several values of the truncation factor. Figure 15 depicts the results of these spectral estimates as a function of the sample size $|\Sigma|$. Since for $g = 4 * g_c$ the overlap of $|\Phi_{optimal}(p=0,1)\rangle$ with the ground state and first excited state is approximately equal (see Table 1) and the associated components in the QPE signal remain of equal magnitude in time, we expect the estimate of the eigenvalues for (too) large truncation factors to be roughly midway between the actual eigenvalues of the ground state and first excited state. From Figure 15, we conclude that this is indeed the case. We note that the deviation from this midway point for small sample sizes $|\Sigma|$ is a consequence of more pronounced noise on the signal to be analyzed. We furthermore note that the transition from being able to distinguish the two eigenvalues (at relatively small truncation factors) to *not* being able to do so (at larger truncation factors) is abrupt as a function of the truncation factor.

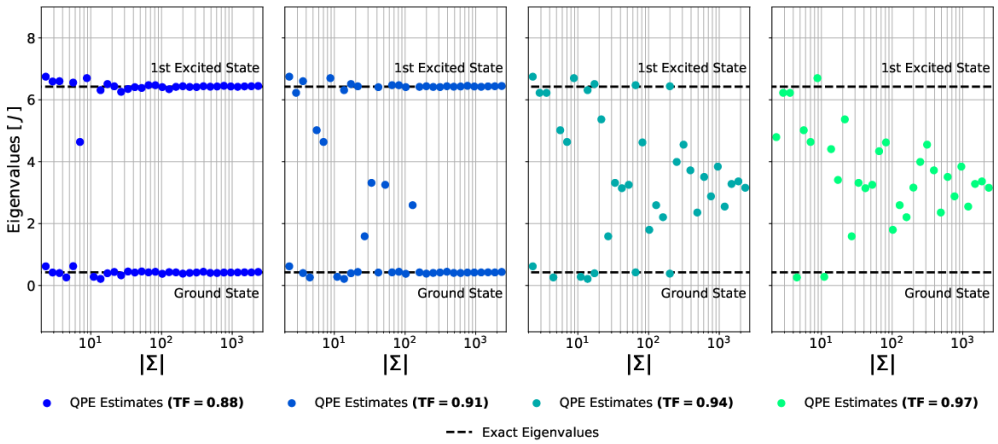


Figure 15: Spectral estimates of the transverse-field ferromagnetic Ising chain (for $n = 7$ and at $g = 4 * g_c$) from the real-time (QPE) evolution of the state $|\Phi_{optimal}(p=0,1)\rangle$ for several values of $|\Sigma|$ and TF. The exact eigenvalues are plotted as well.

We now demonstrate that the low quality of the eigenvalue estimates from the MC signal in the $g \gtrsim 1$ regime is indeed due to the choice of the truncation factor. In Figure 16, we have estimated the eigenvalues from the MC signal for three different values of the truncation factor: 2%, 0.2% and 0.02%. The estimates from the Trotter-error-free and noiseless signal, as well as from the (efficiently obtained) noisy and Trotterized signal are depicted. The estimates from the Trotter-error-free and noiseless MC signals become of higher quality in the $g \gtrsim 1$ regime as the truncation factor is chosen to be smaller. However, the associated estimates from the Trotterized and noisy MC signals become progressively more noise-infected for smaller values of the truncation factor: At $TF = 0.2\%$, the estimates of the ground-state energy are still accurate, while the excited-state eigenvalue cannot be resolved. At $TF = 0.02\%$, the estimates are dominated by noise.

The estimates of the eigenvalues at $TF = 0.02\%$ from the Trotter-error-free and noiseless MC signals demonstrate that, in principle, high-quality eigenvalue estimates in the $g \gtrsim 1$ are possible. That is, by choosing $|\Sigma|$ arbitrarily large – and thereby making the noise arbitrarily small – the eigenvalue estimates in the $g \gtrsim 1$ regime are accurate. However, since $|\Sigma|$ can be at most $poly(n)$, one must explore whether a $|\Sigma|$ that is $poly(n)$ suffices to accurately estimate the eigenvalues of a system of general system size n. As will be mentioned in the Conclusions and Outlook chapter of this report, this will be a relevant point of further research.

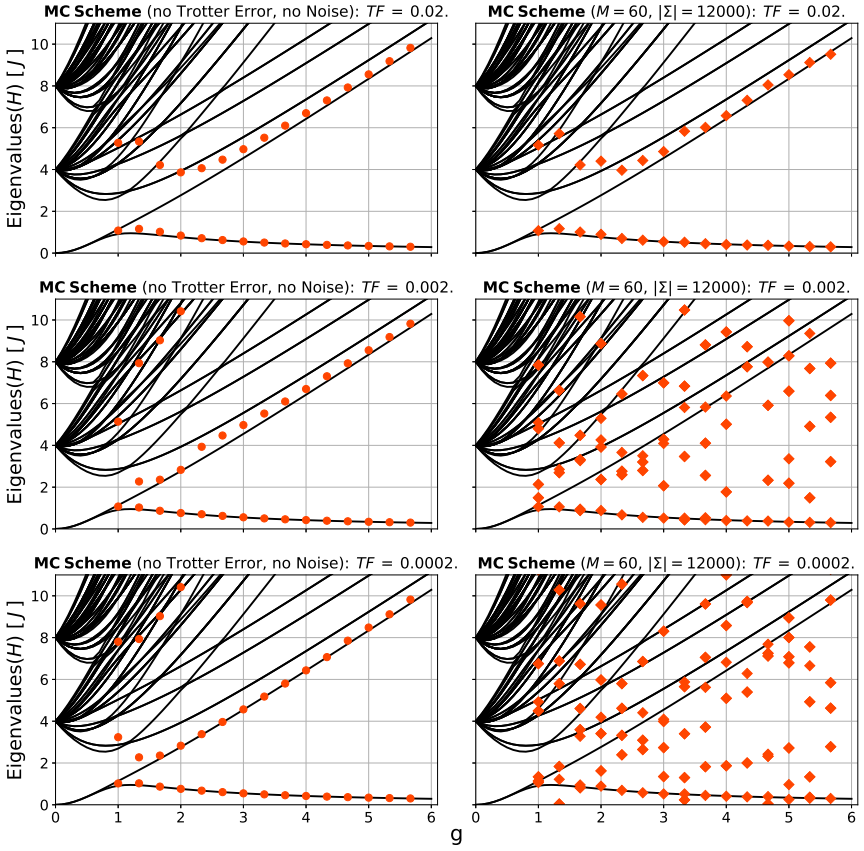


Figure 16: Spectral estimates of the ferromagnetic Ising chain in a transverse field (in the intermediately strong-coupling regime for $n = 7$), which are obtained by analyzing the evolution of $|\Phi_{optimal,p=0,1}\rangle$ in real time (QPE) and imaginary time (MC scheme) using the Matrix Pencil Method. We consider the Trotterized and noise-infected signals in the Monte Carlo setting, as well as their noiseless and Trotter-error-free counterparts. In all instances, the time interval is taken to be 2.5 and time increment has been taken to be 0.025 (i.e. $K = 100$). The truncation factor is taken to be 2% in the top two figures, 0.2% in the middle two figures and 0.02% in the bottom two figures.

V.5 Convergence of Eigenvalue Estimates as a Function of Sample Size $|\Sigma|$

In Chapter III, it was concluded that if the MC and QPE signals would be noiseless and would contain R eigenvalues, then these eigenvalues could be obtained exactly for $K \geq 2R$ – regardless of whether the signal is a MC or a QPE signal. In earlier stages of the current chapter, we have concluded that the magnitude of the noise that is imposed on the QPE signal is always at least that of the noise that is imposed on the MC signal. In addition, we have concluded that the Trotter errors that are imposed on the MC and QPE signals are comparable in magnitude. Given the noisy MC and QPE signals, we now examine numerically (and in the context of the transverse-field Ising chain) how large $|\Sigma|$ has to be in both settings in order for two eigenvalues to be distinguishable.

In Figure 17, we have depicted the spectral estimates that were obtained by analyzing the real-time and imaginary-time evolution of the state $|\Phi_{optimal,p=0,1}\rangle$ at $g = 4 * g_c$. This analysis is performed for many different values of $|\Sigma|$. The truncation factor is taken to be 3%. It is clear that the Matrix Pencil Method is very effective at extracting eigenvalues from the real-time (QPE) evolution signal: The two eigenvalues have already been distinguished when the sample set Σ contains only a few samples. In addition, the noise is almost completely filtered out at $|\Sigma| = O(10^2)$. In comparison, when analyzing the MC signal, the two eigenvalues are distinguished at around $O(10^2)$. Although the noise is filtered out for the ground-state energy estimate at around $O(10^2) - O(10^3)$, the excited-state energy estimate remains slightly noisy even up until the point where the sample set Σ contains a few thousand samples. We attribute this to the fact that the excited-state component of the MC signal rather rapidly becomes smaller in magnitude than the noise that is imposed on the signal, resulting in only a relatively small interval in τ over which this component of the signal can effectively be measured.

We conclude that although the magnitude of the noise that is imposed on the QPE signal is always at least that of the noise that is imposed on the MC signal, and in a noiseless setting the eigenvalues can in principle be extracted equally well, the Matrix Pencil Method accurately extracts the eigenvalues from the QPE signal at a $|\Sigma|$ that is smaller than that for the MC signal.

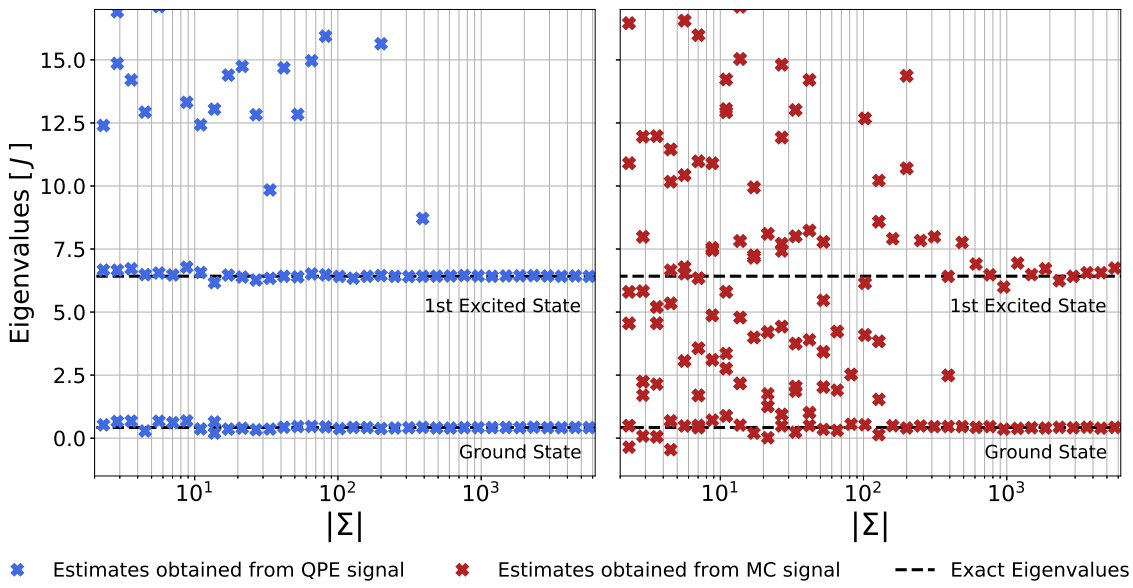


Figure 17: Spectral estimates of the ferromagnetic Ising chain in a transverse field (for $n = 7$ and at $g = 4 * g_c$) as a function of the sample size $|\Sigma|$. The spectral estimates are obtained by analyzing the evolution of $|\Phi_{optimal,p=0,1}\rangle$ in real time (QPE) and imaginary time (MC scheme) using the Matrix Pencil Method (both for a time interval of 2 and $K = 100$). The exact eigenvalues associated with the ground state and the first excited state are plotted as well. The truncation factor (TF) is taken to be 3%.

VI Conclusions and Outlook

The classical simulation of quantum systems is *in general* not a computationally tractable task: It requires memory and computational resources that scale exponentially in the system size. We consider obtaining (some) eigenvalues of a stoquastic – i.e. a sign-problem-free – Hamiltonian H by means of tracking the evolution of the system state in time. We differentiate between the evolution of the system state in real time and imaginary time. In both cases, we examine the use of the Matrix Pencil Method in extracting eigenvalues of H from the state evolution signal. These eigenvalues correspond to the oscillation frequencies of the real-time evolution signal, and to decay rates of the imaginary-time evolution signal. The real-time evolution signal is obtained through a quantum-phase-estimation (QPE) based quantum simulation scheme (which is *inefficiently* implemented on a classical computer). The imaginary-time signal is obtained through a Monte Carlo (MC) scheme that *is* implemented in a computationally tractable manner on a classical computer. Since stoquastic Hamiltonians are conjectured to be efficiently simulatable by classical means, it is interesting to examine whether QPE provides an advantage over the MC scheme for extracting eigenvalues of Hamiltonians in this class. We perform simulations of the ferromagnetic transverse-field Ising chain (which is piece-wise stoquastic in the standard basis) in order to explore some of the properties of the MC scheme and QPE scheme numerically.

We have developed in this report the aforementioned Monte Carlo scheme that can be used to obtain eigenvalues of piece-wise stoquastic Hamiltonians by tracking the system state in imaginary time. In other words, we track evolution of the system state according to the imaginary-time propagation operator $e^{-\tau H}$ (where $\tau \geq 0$). This propagation operator is element-wise non-negative for stoquastic Hamiltonians. In particular, we show that the quantity $\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle$ (where $|\Phi\rangle$ is a state of generally $exp(n)$ elements) can be obtained up to a Trotter error and an ϵ-additive error in time $poly(\epsilon^{-1}, n, L)$ when $H = \sum_{i=1}^{N} H_i$ (where each H_i is stoquastic and acts non-trivially on at most $log(poly(n))$, or typically $O(1)$, variables of the system). We additionally require that $|\Phi\rangle = \sum_x \Phi(x) |x\rangle$ (where $\{|x\rangle\}$ is the basis in which the MC scheme is ran) is such that the quantity $\frac{\Phi(y)}{\Phi(x)^*}$ can be efficiently obtained (for two given basis states $|x\rangle$ and $|y\rangle$) and a sample from $|\Phi(x)|^2$ can be efficiently drawn. L is a quantity that depends on the Trotterization scheme that is used to decompose the propagator $e^{-\tau H}$ into a product of local propagation operators. L is generally proportional to N. The algorithm for estimating $\mathcal{F}(\tau) = \langle\Phi| e^{-\tau H} |\Phi\rangle$ runs efficiently for $\epsilon \geq \frac{1}{poly(n)}$ and $L \leq poly(n)$. By estimating $\mathcal{F}(\tau = \Delta\tau k)$ at $k \in \{0, 1, ..., K \leq poly(n)\}$, one obtains a noisy and Trotterized version of the signal $g(k) = \sum_j e^{-\Delta\tau E_j k} |\langle\Phi|\psi_j\rangle|^2$. By analyzing the signal $g(k)$ by means of the Matrix Pencil Method, (a subset of) the decay rates of $g(k)$ can be obtained. These decay rates correspond to the eigenvalues of Hamiltonian H. We have established several extensions of this Monte Carlo scheme, one of which is using the scheme to obtain (the Trotterized version of) the partition function $Z(\beta)$ of a piece-wise stoquastic Hamiltonian up to an ϵ-additive error. We have discussed, in addition, how the use of scheme can be extended to determining the overlap between an initial and a final state of a quantum circuit consisting of a particular set of (classical) gates.

The quantum-phase-estimation based quantum simulation scheme is used to obtain the real-time evolution of the state $|\Phi\rangle$: $\langle\Phi| e^{-itH} |\Phi\rangle$. In particular, a Trotterized version of this signal is obtained up to an ϵ-additive error. By estimating the value of the signal at times $\Delta t\, k$ (for $k \in \{0, 1, ..., K \leq poly(n)\}$), one obtains a noisy and Trotterized version of the signal $g(k) = \sum_j e^{-i\Delta t E_j k} |\langle\Phi|\psi_j\rangle|^2$. By analyzing the signal $g(k)$ by means of the Matrix Pencil Method, (a subset of) its oscillation frequencies can be obtained. These oscillation frequencies correspond to the eigenvalues of Hamiltonian H.

The ϵ-additive errors that are imposed on the MC and QPE signals effectively correspond to sampling noise. The magnitude of this noise on the MC signal is directly related to the number of Monte Carlo samples (the MC sample size) that are used to estimate the quantity $\mathcal{F}(\tau)$ at each instance in τ. The magnitude of the noise on the QPE signal is determined by the number of measurements that are performed of the ancillary qubit state (the QPE sample size) for each value of t. We have shown – analytically as well as numerically – that, for a given MC/QPE sample size, the variance of the QPE signal estimate is always at least that of the MC estimate, but does never exceed twice the variance of the MC estimate. In addition, we have determined numerically that the Trotter error that is imposed on the MC and QPE signals is approximately equal in magnitude for both settings.

We have established that in the noiseless signal case, the eigenvalues can be extracted from the decaying (MC) signal and from the oscillating (QPE) signal equally well using the Matrix Pencil Method (provided

that K is sufficiently large). It was found numerically that in the noisy signal setting, in order to distinguish the ground-state and first-excited-state eigenvalues of the transverse-field Ising chain at intermediately strong coupling, the MC sample size needs to be larger (by about two orders of magnitude) than the QPE sample size. Since in the noiseless-signal setting the Matrix Pencil Method performs equally well for a MC as for a QPE signal, and the magnitude of the sampling noise on the QPE signal is always at least that on the MC signal, we conclude that the Matrix Pencil Method performs worse in extracting (multiple) eigenvalues from a MC signal than from a QPE signal. To prove more rigourously that the Matrix Pencil Method for extracting eigenvalues from a QPE signal indeed outperforms that for extracting eigenvalues from a MC signal in the presence of noise, we will have to perform an analysis for the MC signal that is similar as the one for QPE signals in [31] in the future.

Since the MC signal can be expressed as $g(k) = \sum_j e^{-\Delta\tau E_j k} |\langle\Phi|\psi_j\rangle|^2$, one of the aspects that governs its magnitude is the value of the eigenvalues $\{E_j\}$. If one would set the smallest eigenvalue of each H_i to zero (which is a feature of the MC scheme considered here), then the smallest eigenvalue of H can be non-zero due to frustration of Hamiltonian H. Therefore, the magnitude of the MC signal is directly related to the extent to which H is frustrated. The MC signal can be obtained efficiently up to an ϵ-additive error, where $\epsilon \geq 1/poly(n)$. If one thus wishes to estimate the MC signal within reasonable accuracy, the signal can *not* be exponentially small as a function of the system size. This applies to the estimation of the partition function $Z(\beta)$ equivalently. As a result, estimates of the MC signal (and the partition function) can only be obtained accurately and in an efficient manner if the Hamiltonian is *mildly* frustrated. We note that the τ and β interval in which the MC signal and partition function (respectively) can be accurately obtained relates directly to the degree of frustration: For a given scaling of the degree of frustration with the system size $\Lambda(n)$, the partition function can for instance be accurately estimated for $\beta \leq \frac{log(poly(n))}{\Lambda(n)}$ (i.e. in some high-temperature regime). The scaling of the degree of frustration of a Hamiltonian H with the system size ($\Lambda(n)$) is thus a prominent factor limiting the range of applicability of the MC scheme considered here. The frustration of (stoquastic) Hamiltonians will therefore be a relevant point of further study.

We have argued that there are several reasons to believe that estimating the signal $\mathcal{F}(\tau)$ on a τ interval that lies in the small-τ regime is beneficial. These (not necessarily mutually independent) reasons are: Since the signal $\mathcal{F}(\tau)$ decays as a function of τ, there will be a point in τ after which the noise magnitude exceeds that of the signal itself. The Trotter error that is imposed on the signal increases with τ, and in fact even vanishes at $\tau = 0$. The sampling noise on the signal increases with τ and saturates to a maximum value at large values of τ (the sampling noise vanishes at $\tau = 0$ as well). As discussed above, the diminishing effect of frustration on the magnitude of $\mathcal{F}(\tau)$ is less pronounced at small values of τ.

We intend to explore, in addition, how the MC scheme for estimating $\mathcal{F}(\tau)$ and $Z(\beta)$ for stoquastic Hamiltonians that is developed here might be extended to estimation of e.g. thermal averages of an observable $\mathcal{O}$:

$$\langle\mathcal{O}\rangle = \frac{Tr\big(\mathcal{O}\, e^{-\beta H}\big)}{Z(\beta)}. \tag{144}$$

We note that this observable $\mathcal{O}$ will need to be element-wise non-negative when expressed in the basis in which the MC scheme is ran. Extensions of the MC scheme to estimations of $\langle\mathcal{O}\rangle$ may allow for a more direct estimation of interesting physical quantities of stoquastic Hamiltonian systems. We do note, however, that due to the partition function $Z(\beta)$ in the denominator (which itself might not be known to high accuracy), being able to efficiently determine $\langle\mathcal{O}\rangle$ using our Monte Carlo scheme with sufficient accuracy seems implausible in general.

We believe that it will be interesting to investigate whether the equivalence between the quantum-phase-estimation based scheme and the Monte Carlo scheme that have been discussed in this report can be used to develop a hybrid algorithm. The current machinery (NISQ devices) that is required to run the quantum-phase-estimation based scheme efficiently is limiting in terms of the number of available qubits and their coherence times. Therefore, a compelling point of study might be how the MC scheme that is developed here might alleviate the computational stress on NISQ devices and specifically their use for efficiently running large-scale QPE schemes.

Another point of further study is the following: Suppose one is given a (simple) graph G of n vertices, then the $n \times n$ Laplacian matrix is given by:

$$L \equiv D - A, \tag{145}$$

where D is the degree matrix of graph G (a diagonal matrix whose entries equal the number of vertices to which a given vertex is connected) and A is the adjacency matrix of graph G (where $\langle x| A |y\rangle = 0$ if vertices x and y are not connected, $\langle x| A |y\rangle = 1$ if vertices x and y are connected and $\langle x| A |x\rangle = 0$, $\forall x$). The Laplacian matrix L is a stoquastic matrix. Since L is Hermitian, the signal $\langle \Phi| e^{-\tau L} |\Phi\rangle$ (where $\tau \geq 0$ and $|\Phi\rangle$ is some state) equals a superposition of decaying signals, the decay rates of which correspond to the eigenvalues of L. Although L is stoquastic, it is *not* necessarily local (in the sense of Definition I.1). Therefore, Trotterization schemes in which one exploits locality cannot be used directly and one has to exploit other properties of L to be able to fit it in the Monte Carlo framework of Theorem IV.3 (by means of altered Trotterization schemes). One way of potentially doing this is to assume sparsity of matrix L. By further developing this subject of further study, we can examine whether eigenvalues of sparse Laplacian matrices can be obtained efficiently by classical means.

References

[1] S. Lloyd. "Universal Quantum Simulators". In: *Science* 273 (23 August 1996).

[2] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary Edition)*. Cambridge University Press, 2010. Chap. 4, pp. 204–213.

[3] W.M.C. Foulkes et al. "Quantum Monte Carlo simulations of solids". In: *Review of Modern Physics* 73.1 (January 2001).

[4] M. Troyer and U. Wiese. "Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations". In: *Physical Review Letters* 94.17 (May 2005).

[5] S. Bravyi et al. "The Complexity of Stoquastic Local Hamiltonian Problems". In: arXiv:quant-ph/0606140 (1 February 2008).

[6] D. Hangleiter et al. "Easing the Monte Carlo sign problem". In: arXiv:1906.02309 (24 January 2020).

[7] S. Bravyi. "Monte Carlo simulation of stoquastic Hamiltonians". In: arXiv:1402.2295v2 (January 2015).

[8] S. Sachdev. *Quantum Phase Transition (Second Edition)*. Cambridge University Press, 2011. Chap. 5.

[9] M. Ioannou et al. "Sign-curing local Hamiltonians: termwise versus global stoquasticity and the use of Clifford transformations". In: arXiv:2007.11964 (July 2020).

[10] N. Metropolis et al. "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6 (1953).

[11] J.M. Thijssen. *Computational Physics (Second Edition)*. Cambridge University Press, 2007. Chap. 10,12.

[12] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 2005.

[13] H.F. Trotter. "On the Products of Semi-Groups of Operators". In: *Proceedings of the American Mathematical Society* 10.4 (Augustus 1959).

[14] J. Huyghebaert and H. De Raedt. "Product formula methods for time-dependent Schrödinger problems". In: *Journal of Physics A: Mathematical and General* 23 (1990).

[15] A.M. Childs et al. "A Theory of Trotter Error". In: arXiv:1912.08854v2 (8 January 2020).

[16] A.M. Childs and Y. Su. "Nearly optimal lattice simulation by product formulas". In: *Physical Review Letters* 123 (August 2019).

[17] M. Suzuki. "General theory of fractal path integrals with applications to many-body theories and statistical physics". In: *Journal of Mathematical Physics* 32.2 (February 1991).

[18] I.M. Georgescu, S. Ashhab, and F. Nori. "Quantum Simulation". In: *Reviews of Modern Physics* 86 (January 2014).

[19] D. Poulin et al. "Quantum Simulation of Time-Dependent Hamiltonians and the Convenient Illusion of Hilbert Space". In: *Physical Review Letters* 106 (April 2011).

[20] F. Arute et al. “Hartree-Fock on a superconducting qubit quantum computer”. In: arXiv:2004.04174v2 (April 2020).

[21] G.M. D’Ariano, M.G.A. Paris, and M.F. Sacchi. “Quantum Tomography”. In: arXiv:quant-ph/0302028v1 (February 2008).

[22] S. Somaroo et al. “Quantum Simulations on a Quantum Computer”. In: *Physical Review Letters* 82 (June 1999).

[23] M. Greiner et al. “Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms”. In: *Nature* 415 (January 2002).

[24] A. Blais, S.M. Girvin, and W.D. Oliver. “Quantum information processing and quantum optics with circuit quantum electrodynamics”. In: *Nature Physics* 16 (March 2020).

[25] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary Edition)*. Cambridge University Press, 2010.

[26] T.E. O’brien, B. Tarasinski, and B.M. Terhal. “Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments”. In: *New Journal of Physics* 21.(023022) (2019).

[27] T.K. Sarkar and O. Pereira. “Using the Matrix Pencil Method to Estimate the Parameters of a Sum of Complex Exponentials”. In: *IEEE Antennas and Propagation Magazine* 37.1 (February 1995).

[28] Y. Hua and T.K. Sarkar. “On SVD for Estimating Generalized Eigenvalues of Singular Matrix Pencil in Noise”. In: *IEEE Transactions on Signal Processing* 39.4 (April 1991).

[29] Y. Hua and T.K. Sarkar. “Matrix Pencil Method for Estimating Parameters of Exponentially Damped/Undamped Sinusoids in Noise”. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 38.5 (May 1990).

[30] J. Helsen, F Battistel, and B.M. Terhal. “Spectral Quantum Tomography”. In: *npj Quantum Information* 5.74 (2 September 2019).

[31] A. Moitra. “Super-resolution, Extremal Functions and the Condition Number of Vandermonde Matrices”. In: arXiv:1408.1681v4 (30 April 2015).

[32] R.A. Horn and C.R. Johnson. *Matrix Analysis (Second Edition)*. Cambridge University Press, 2013.

[33] S. Bravyi and D. Gosset. “Polynomial-Time Classical Simulation of Quantum Ferromagnets”. In: *Physical Review Letters* 119 (September 2017).

[34] E. Crosson and S. Slezak. “Classical Simulation of High Temperature Quantum Ising Models”. In: arXiv:2002.02232v1 (February 2020).

[35] M. Van den Nest. “Simulating quantum computers with probabilistic methods”. In: arXiv:0911.1624v3 (October 2018).

[36] F.F. Assaad and H.G. Evertz. *Computational Many-Particle Physics*. Springer, 2008. Chap. 10, pp. 277–356.

A Appendix: The Sign Problem

To gain insight into when the sign problem occurs in a Monte Carlo simulation, we consider a world-line Monte Carlo simulation of a one-dimensional Heisenberg chain (with periodic boundary conditions) ([36]). The Hamiltonian of such a system is the following:

$$H = \sum_{i=1}^{n} H_i = J \sum_{i=1}^{n} \Big(\sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z \Big), \tag{146}$$

where $\sigma_{n+1}^{\alpha} = \sigma_1^{\alpha}$ ($\alpha \in \{x, y, z\}$) and each of the bond Hamiltonians H_i acts non-trivially on two spins and is therefore of locality $k = 2$. The partition function is given by $Z(\beta) = Tr\big(e^{-\beta H}\big)$. The following first-order Trotter decomposition in terms of the individual bond Hamiltonians H_i can now be performed:

$$Z(\beta) \approx Tr\big((e^{-\Delta\tau H_1} e^{-\Delta\tau H_2} ... \, e^{-\Delta\tau H_n})^M\big), \tag{147}$$

where $\Delta\tau = \beta/M$.

We shall work in the standard basis (local-σ^z basis), with the associated basis states denoted by $\{|x\rangle\}$. The Hilbert space of the spin system is of dimensionality 2^n and the completeness relation is thus given by $\sum_{x=1}^{2^n} |x\rangle\langle x| = I$. By inserting complete sets of basis states in the expression from eq. 147 and by noting that $Tr(A) = \sum_{x=1}^{2^n} \langle x| A |x\rangle$, we obtain the following expression for $Z(\beta)$:

$$Z(\beta) = \sum_{x_0, x_1, ..., x_{Mn-1}} \langle x_0| e^{-\Delta\tau H_1} |x_1\rangle\langle x_1| e^{-\Delta\tau H_2} |x_2\rangle \ ... \ \langle x_n| e^{-\Delta\tau H_1} |x_{n+1}\rangle \ ... \ \langle x_{Mn-1}| e^{-\Delta\tau H_n} |x_0\rangle \, . \tag{148}$$

We can imagine each term in this sum as comprising a loop consisting of Mn nodes, where each node represents a state $|x_l\rangle$ (for $l \in \{0, 1, ..., Mn-1\}$). The nodes are connected by non-zero matrix elements of the $e^{-\Delta\tau H_i}$ operators, which propagate the state $|x_l\rangle$ to $|x_{l+1}\rangle$ in imaginary-time.

To proceed, we investigate further the form of the propagation operators $e^{-\Delta\tau H_i}$. To that extent, we express the bond Hamiltonians H_i explicitly in the standard basis:

$$H_i = J \begin{pmatrix} 1/4 & 0 & 0 & 0 \\ 0 & -1/4 & 1/2 & 0 \\ 0 & 1/2 & -1/4 & 0 \\ 0 & 0 & 0 & 1/4 \end{pmatrix}. \tag{149}$$

The propagation operator $e^{-\Delta\tau H_i}$ can be conveniently calculated by means of its Taylor expansion: $\sum_{k=0}^{\infty} \frac{(-\Delta\tau)^k}{k!} \big(H_i\big)^k$. $\big(H_i\big)^k$ can be determined explicitly by diagonalizing H_i in terms of its eigenstates $\{|\psi_j\rangle\}$ and eigenvalues $\{\lambda_j\}$:

$$\frac{H_i}{J} = \sum_{j=1}^{2^n} \lambda_j |\psi_j\rangle\langle\psi_j| \quad \longrightarrow \quad \underbrace{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}}_{\{|\psi_j\rangle\}} \quad ; \quad \underbrace{\frac{1}{4}, \frac{1}{4}, \frac{-3}{4}, \frac{1}{4}}_{\{\lambda_j\}}. \tag{150}$$

Since H_i is Hermitian, $\big(\frac{H_i}{J}\big)^k = \sum_{j=1}^{2^n} (\lambda_j)^k |\psi_j\rangle\langle\psi_j|$. Filling in the expression for $\big(H_i\big)^k$ (using the explicit expressions for $\{|\psi_j\rangle\}$ and $\{\lambda_j\}$) in the Taylor expansion representation of $e^{-\Delta\tau H_i}$, we represent $e^{-\Delta\tau H_i}$ in the standard basis as follows:

$$e^{-\Delta\tau H_i} = e^{\Delta\tau J/4} \begin{pmatrix} e^{-\Delta\tau J/2} & 0 & 0 & 0 \\ 0 & \cosh(\Delta\tau J/2) & -\sinh(\Delta\tau J/2) & 0 \\ 0 & -\sinh(\Delta\tau J/2) & \cosh(\Delta\tau J/2) & 0 \\ 0 & 0 & 0 & e^{-\Delta\tau J/2} \end{pmatrix}. \tag{151}$$

The propagation operator $e^{-\Delta\tau H_i}$ couples only the pairs of states $|x_l\rangle$ and $|x_{l+1}\rangle$ for which $\langle x_l| e^{-\Delta\tau H_i} |x_{l+1}\rangle$ is non-zero. Every loop (represented by a single term in the sum from eq. 148) thus consists of a set of

nodes $\{|x_0\rangle, |x_1\rangle, ..., |x_{Mn-1}\rangle\}$ (denoted by $\boldsymbol{x}$) where each pair of adjacent states $|x_l\rangle$ and $|x_{l+1}\rangle$ on the loop obeys $\langle x_l| e^{-\Delta\tau H_i} |x_{l+1}\rangle \neq 0$. Otherwise, the term in the sum from eq. 148 would equal zero and therefore not contribute to $Z(\beta)$. The allowed propagation plaquettes in imaginary-time between two neighbouring spins for the system considered here are visualized in Figure 18.

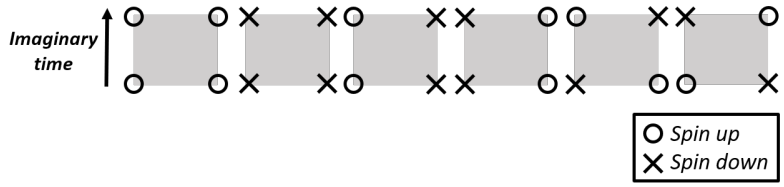


Figure 18: All allowed propagation plaquettes in the imaginary-time direction between two neighbouring spins on the chain, according to the Heisenberg Hamiltonian.

Each world line configuration is denoted by a set of states $\boldsymbol{x}$ and the corresponding term in the sum of eq. 148 is denoted by the weight $W(\boldsymbol{x})$. The partition function thus reduces to:

$$Z(\beta) = \sum_{\boldsymbol{x}} W(\boldsymbol{x}). \tag{152}$$

Suppose we wish to estimate the expectation value of an observable $\mathcal{O}$. This expectation value can be expressed in terms of the weights $W(\boldsymbol{x})$ as follows:

$$\langle \mathcal{O} \rangle = \frac{Tr\big(e^{-\beta H}\mathcal{O}\big)}{Tr\big(e^{-\beta H}\big)} = \frac{\sum_{\boldsymbol{x}} W(\boldsymbol{x})\mathcal{O}(\boldsymbol{x})}{\sum_{\boldsymbol{x}} W(\boldsymbol{x})}. \tag{153}$$

The sampling of the world line configurations can be efficiently done by means of local updates of the configurations and global updates (e.g. loop algorithm). These two types of updates collectively make for an ergodic updating scheme ([36]).

It is of importance to note that $e^{-\Delta\tau H_i}$ has off-diagonal elements in the standard basis which might become negatively valued. In particular, when $J > 0$ (and the coupling between neighbouring spins on the chain is thus anti-ferromagnetic) the non-zero off-diagonal elements of $e^{-\Delta\tau H_i}$ become negatively valued. However, this does not directly imply that the simulation suffers from a sign problem: A sign problem might occur when negatively valued terms appear in the sum of eq. 148. A factor $\langle x_l| e^{-\Delta\tau H_i} |x_{l+1}\rangle$ might be negative (if it coincides with the abovementioned off-diagonal terms), but a weight $W(\boldsymbol{x})$ in eq. 148 consists of a product of Mn of these factors. As long as an even number of these Mn factors is negatively valued, the overall weight will remain positive and a sign problem will be absent. For bipartite lattices (in our case for n even) this will always be the case. For non-bipartite lattices (n is odd and frustration can occur), negative weights in eq. 148 might appear and can cause a sign problem. Figure 19 displays one world line configuration corresponding to a positive weight and one associated to a negative weight for the case where $n = 3$ and $M = 2$.

The appearance of negative weights $W(\boldsymbol{x})$ yields direct sampling from the distribution $\frac{W(\boldsymbol{x})}{\sum_{\boldsymbol{x}} W(\boldsymbol{x})}$ intractable. Instead, we consider sampling from a (non-negative) distribution (of a related system) that is given by $\frac{|W(\boldsymbol{x})|}{\sum_{\boldsymbol{x}} |W(\boldsymbol{x})|}$ and which includes the signs of the weights ($sign(\boldsymbol{x})$) in the quantity being sampled:

$$\langle \mathcal{O} \rangle = \frac{\sum_{\boldsymbol{x}} |W(\boldsymbol{x})|\, sign(\boldsymbol{x})\mathcal{O}(\boldsymbol{x})}{\sum_{\boldsymbol{x}} |W(\boldsymbol{x})|\, sign(\boldsymbol{x})} = \frac{\sum_{\boldsymbol{x}} |W(\boldsymbol{x})|\, sign(\boldsymbol{x})\mathcal{O}(\boldsymbol{x}) / \sum_{\boldsymbol{x}} |W(\boldsymbol{x})|}{\sum_{\boldsymbol{x}} |W(\boldsymbol{x})|\, sign(\boldsymbol{x}) / \sum_{\boldsymbol{x}} |W(\boldsymbol{x})|} = \frac{\langle sign\, \mathcal{O} \rangle'}{\langle sign \rangle'}, \tag{154}$$

where the prime indicates that the expectation value is determined by sampling from the alternative distribution $\frac{|W(\boldsymbol{x})|}{\sum_{\boldsymbol{x}} |W(\boldsymbol{x})|}$.

The denominator in eq. 154 is of particular interest and we therefore study it in more detail:

$$\langle sign \rangle' = \frac{\sum_{\boldsymbol{x}} |W(\boldsymbol{x})|\, sign(\boldsymbol{x})}{\sum_{\boldsymbol{x}} |W(\boldsymbol{x})|}. \tag{155}$$

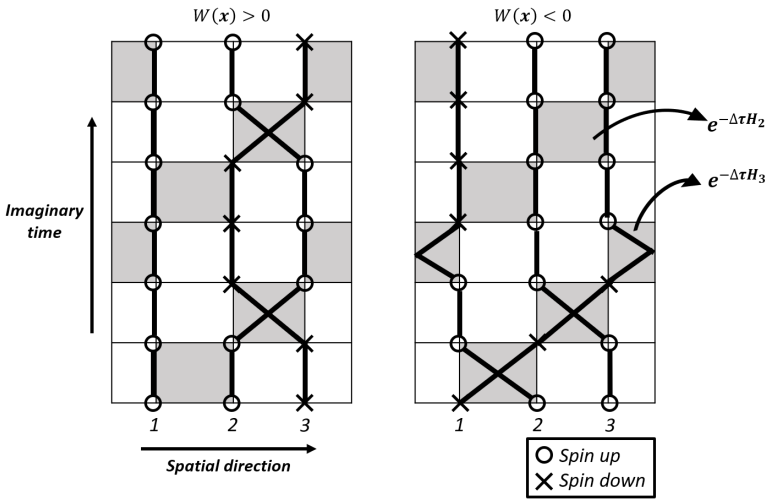


Figure 19: Two world line configurations for an anti-ferromagnetic Heisenberg chain (with periodic boundary conditions) consisting of $n = 3$ spins. The Trotterization parameter M is chosen to be 2. The left configuration corresponds to a positive weight $W(\boldsymbol{x})$, while the right configuration is associated with a negatively valued weight.

By noting that $Z = \sum_{\boldsymbol{x}} W(\boldsymbol{x}) = \sum_{\boldsymbol{x}} |W(\boldsymbol{x})|\ sign(\boldsymbol{x})$ (the partition function of the original system) $Z' = \sum_{\boldsymbol{x}} |W(\boldsymbol{x})|$ (the partition function of the related system with only non-negative weights), we relate $\langle sign \rangle'$ to these partition functions as follows:

$$\langle sign \rangle' = \frac{Z}{Z'} = \frac{e^{-\beta F}}{e^{-\beta F'}} = e^{-\beta \Delta F} = e^{-\beta n \Delta f}, \tag{156}$$

where $\Delta F = F - F'$ and Δf is the Helmholtz free energy density difference (which is an intensive quantity). We conclude that $\langle sign \rangle'$ decreases exponentially with the inverse temperature β and system size n. If one would draw $\mathcal{N}$ independent samples to determine $\langle sign \rangle'$, the relative error in the estimate of $\langle sign \rangle'$ would be:

$$\frac{\sigma_{sign}}{\langle sign \rangle'} = \frac{\sqrt{\left(\langle sign^2 \rangle' - (\langle sign \rangle')^2\right)/\mathcal{N}}}{\langle sign \rangle'} = \frac{\sqrt{1 - (\langle sign \rangle')^2}}{\sqrt{\mathcal{N}} \langle sign \rangle'} \approx \frac{e^{\beta n \Delta f}}{\sqrt{\mathcal{N}}}, \tag{157}$$

where the last equality holds for large n and β. Similarly, the relative error in the numerator of eq. 154 scales exponentially in n and β. The expectation value $\langle \mathcal{O} \rangle$ thus equals the quotient of two stochastically determined quantities, whose relative errors scale exponentially in n and β. The time needed to compute $\langle \mathcal{O} \rangle$ with a given accuracy is thus exponentially large as a function of the system size n and inverse temperatures β. We thus encounter a sign problem, yielding this Monte Carlo simulation of a large frustrated anti-ferromagnetic Heisenberg chain at low temperature intractable (in the chosen basis). We do note that the appearance of a sign problem is representation dependent. That is, if a Monte Carlo simulation would have been performed in another basis or representation, the sign problem might be absent. The current discussion merely serves as an illustration of how the sign problem can occur in a Monte Carlo simulation.

B Appendix: Code

In this appendix, we include our code that is used to track the real-time and imaginary-time of a one-dimensional[8] spin system with Hamiltonian (where k denotes the locality of the interactions):

$$H = \sum_i H_i = \sum_i \Big(J_x \, \sigma_i^x \sigma_{i+1}^x ... \, \sigma_{i+k}^x + J_y \, \sigma_i^y \sigma_{i+1}^y ... \, \sigma_{i+k}^y + J_z \, \sigma_i^z \sigma_{i+1}^z ... \, \sigma_{i+k}^z + g_x \, \sigma_i^x + g_y \, \sigma_i^y + g_z \, \sigma_i^z \Big), \quad (158)$$

where k is a variable to be chosen. We note that although our code does not include this feature, the MC scheme and QPE scheme do allow for the quantities k, $\{J_x, J_y, J_z\}$ and $\{g_x, g_y, g_z\}$ to be a function of i. We note that one has a choice of multiple Trotterization schemes that can be implemented for the real-time or imaginary-time tracking of the system state. The code employs the Matrix Pencil Method to extract estimates of eigenvalues from the MC and QPE signals. The user is responsible for ensuring that the Hamiltonian H is piece-wise stoquastic. A pseudocode is depicted in Figure 20.

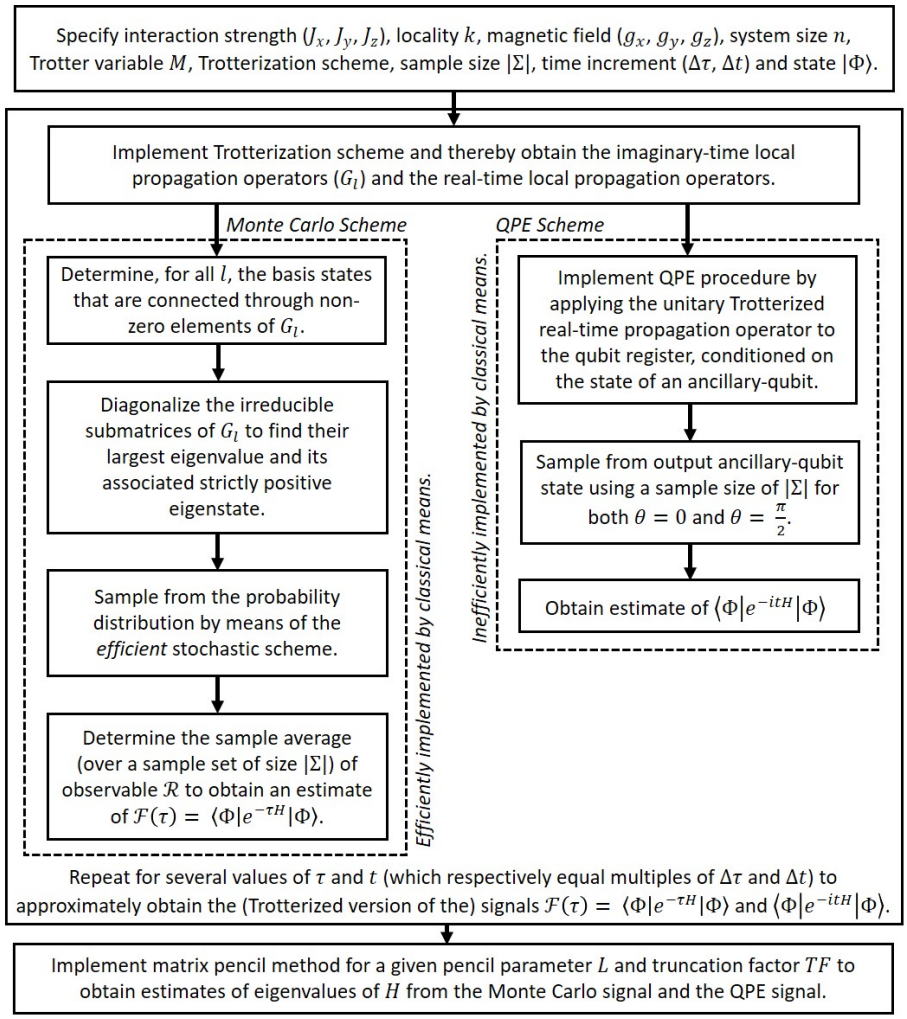


Figure 20: Pseudocode implementing the real-time (QPE) and imaginary-time (MC) tracking of the state of a one-dimensional spin chain with a piece-wise stoquastic Hamiltonian H. Using the evolution trajectories of this state, estimates of eigenvalues of H are obtained by means of the Matrix Pencil Method.

[8] Note that the code can in principle be extended to simulate systems of higher spatial dimensionality.

StoqMC.py

```
'''
-------------------------------------------------------------------------------------------------------------
This code tracks the (Trotterized) evolution of the state of a (1D) spin system of size n in im-
aginary (Euclidian) and real time. The real time evolution is inefficiently tracked in general.
The imganary evolution is efficiently tracked given the Hamiltonian is piece-wise stoquastic wi-
th interactions of (general) locality k = log(poly(n)). The interactions need not be translatio-
nally invariant. Using the Matrix Pencil Method, the eigenvalues of the Hamiltonian are estimat-
ed from the state evolutions.

This main code file (StoqMC.py) calls functions from the separate code files (StoqMCfunctions.p-
y) and (StoqMCsimulation.py). The files StoqMCfunctions.py and StoqMCsimulation.py contain desc-
riptions of each function that is called in this main code file.

We note that the Hamiltonian of the spin system used here is of the form (where periodic bounda-
ry conditions are imposed):
    H = sum_i (J_x*X_i*...*X_i+k + J_y*Y_i*...*Y_i+k + J_z*Z_i*...*Z_i+k
                                                        + g*X_i + g_y*Y_i + g_z*Z_i).
We note it is the users own responsibility to make sure that the Hamiltonian is expressed in a
basis in which it is piece-wise stoquastic (if possible).
-------------------------------------------------------------------------------------------------------------
'''
import numpy as np
import matplotlib.pyplot as plt
import StoqMCfunctions as f
import StoqMCsimulations as s

# Initialization
N = 7                               # Length of Spin Chain
M = 60                              # Number of Trotter Steps
Jx = 0                              # X Spin-Spin Interaction Strength
Jy = 0                              # Y Spin-Spin Interaction Strength
Jz = -1                             # Z Spin-Spin Interaction Strength
g = -4                              # (External) Transverse Magnetic Field (in x direction)
gy = 0                              # (External) Magnetic Field (in y direction)
gz = 0                              # (External) Magnetic Field (in z direction)
k = 2                               # Locality of the Spin-Spin Interactions
numsamples = 4000                   # Sample Size at each instance in time (|Sigma|)
measpoints = 101                    # Number of Measurement Points in time (K)
tau_increment = 0.025               # Time Increment (Measured Time Interval = K*Time Increment)
checkerboard = 0                    # 0 (Trivial Trotter Decomposition), 1 (Checkerboard Decomposition: when
      in use, it is assumed that k = 2 and n is even)
L = np.int(measpoints/2)            # Pencil Parameter
truncation_factor = 1*10**(-4)  # Truncation Factor

gfullmeas = np.zeros(measpoints)                # Initialization of Noisless and Trotter-Errorless
     Decaying Signal
ftaumeas = np.zeros(measpoints)                 # Initialization of Noise- and Trotter-Error Infected
     Decaying Signal (MC Signal)
QPEnontrotsignalreal = np.zeros(measpoints)     # Initialization of Real Part of Noisless and Trotter-
     Errorless Oscillating Signal
QPEnontrotsignalimag = np.zeros(measpoints)     # Initialization of Imag Part of Noisless and Trotter-
     Errorless Oscillating Signal
QPEsampled_realsignal = np.zeros(measpoints)    # Initialization of Real Part of Noise- and Trotter-
     Error Infected Oscillating Signal (Real Part QPE Signal)
QPEsampled_imagsignal = np.zeros(measpoints)    # Initialization of Imag Part of Noise- and Trotter-
     Error Infected Oscillating Signal (Imag Part QPE Signal)

# Initialization of the state |Phi>
# |+>:
#phi_coefficients = np.ones(2**N)/(2**(N/2)) # |\Phi> = |+>
# |Phi_{opt,p=0,1}>:
alpha = 1/((N+1)**(1/2))
beta = (N/(N+1))**(1/2)
phi_coefficients = np.zeros(2**N)
phi_labels = np.arange(2**N)
phi_binary = np.flip(np.unpackbits(np.reshape(phi_labels, (1,2**N)).astype(np.uint8), axis = 0), axis =
     0)[0:N,:]
for i in range(N):
    phi_coefficients = phi_coefficients + (phi_binary[i,:] == 0).astype(int)*(alpha + beta)/((2)**(1/2))
         /((2**(N-1))**(1/2))
    phi_coefficients = phi_coefficients + (phi_binary[i,:] == 1).astype(int)*(alpha - beta)/((2)**(1/2))
         /((2**(N-1))**(1/2))
phi_coefficients = phi_coefficients/((N*(alpha**2 + beta**2) + N*(N-1)*alpha**2)**(1/2))

eigvaldata = np.zeros((4,100,1), dtype = np.complex)            # Initialization of Storage Matrix for
     Eigenvalue Estimates
gfulldata = np.zeros((1,measpoints))                            # Initialization of Storage Matrix for
     Idealized Decaying Signal
ftaudata = np.zeros((1,measpoints))                             # Initialization of Storage Matrix for
     MC Signal
qpedata = np.zeros((1,measpoints), dtype = np.complex)          # Initialization of Storage Matrix for
     Idealized Oscillating Signal
qpenoisydata = np.zeros((1,measpoints), dtype = np.complex)     # Initialization of Storage Matrix for
     QPE Signal

for m in range(1):
    #numsamples = np.int(4546*1.25*1.25**x[m])

    for Q in range(measpoints):
        tau = tau_increment*(Q) # 'Time (Imaginary and/or Real)'

        # Calculate the Matrix Element Directly and Inefficiently Implement the QPE Procedure (!!!USE
             ONLY FOR SMALL N!!!)
        Gfull, FullHeigvals, FullHeigvecs, QPEsampled_real, QPEsampled_imag, realt_nontrot_evolution,
             trivialtrotterbound, CBtrotterbound = s.fullHandQPE(k,N,M,Jx,Jy,Jz,g,tau,phi_coefficients,
             numsamples,checkerboard)
        gfullmeas[Q] = np.matmul(np.matmul(np.transpose(phi_coefficients),Gfull),phi_coefficients)
```

```
            QPEsampled_realsignal[Q] = QPEsampled_real
            QPEsampled_imagsignal[Q] = QPEsampled_imag
            QPEnontrotsignalreal[Q] = np.real(realt_nontrot_evolution)
            QPEnontrotsignalimag[Q] = np.imag(realt_nontrot_evolution)
            #Ntermtrotterbound[Q] = trivialtrotterbound
            #Twotermtrotterbound[Q] = CBtrotterbound

            # Simulate String of Local Imaginary Time Propagation Operators
            Gtrot = s.GtrotSim(k,N,M,Jx,Jy,Jz,g,tau)

            # Simulate Connected Sets of States, and Eigenstates & Eigenvalues of the Block Diagonal
                Matrices
            maxeigval, poseigvec, connectedstates, numsets = s.DiagBlocksEig(M,N,Gtrot)

            # Obtain |Sigma| Samples of High-Dimensional Probability Distribution (Pi) and Estimate the MC
                Signal at tau (F(tau))
            if checkerboard == 0:
                ftau = s.PiandFTau(maxeigval,poseigvec,connectedstates,numsets,M,N,k,Gtrot,numsamples,
                    phi_coefficients)
            if checkerboard == 1:
                ftau = s.PiandFTauCB(maxeigval,poseigvec,connectedstates,numsets,M,N,k,Gtrot,numsamples,
                    phi_coefficients)
            ftaumeas[Q] = ftau

            print(m,'step ',Q) # To indicate the progress of the simulation

        # Construct Complete QPE Signals (with and without Noise and Trotter Error) using their Real and
            Imaginary Parts
        QPEsampled_signal = QPEsampled_realsignal + 1j*QPEsampled_imagsignal
        QPEnontrotsignal = QPEnontrotsignalreal + 1j*QPEnontrotsignalimag

        # Store State Evolution Data in Storage Matrices
        ftaudata[m,:] = ftaumeas
        gfulldata[m,:] = gfullmeas
        qpenoisydata[m,:] = QPEsampled_signal
        qpedata[m,:] = QPEnontrotsignal

        # Implement Matrix Pencil Method to obtain Eigenvalue Estimates from State Evolution Data
        generalizedeigvals, I = s.MatrixPencilMethod(ftaudata[m,:],measpoints,L,truncation_factor)
        generalizedeigvals = generalizedeigvals[generalizedeigvals >= 10**(-10)]
        eigval_estimates = np.sort(-np.log(generalizedeigvals.real)/tau_increment)
        eigvaldata[0,0:np.size(eigval_estimates),m] = eigval_estimates

        generalizedeigvals, I = s.MatrixPencilMethod(gfulldata[m,:],measpoints,L,truncation_factor)
        generalizedeigvals = generalizedeigvals[generalizedeigvals >= 10**(-10)]
        eigval_estimates = np.sort(-np.log(generalizedeigvals.real)/tau_increment)
        eigvaldata[1,0:np.size(eigval_estimates),m] = eigval_estimates

        generalizedeigvals, I = s.MatrixPencilMethod(qpenoisydata[m,:],measpoints,L,truncation_factor)
        generalizedeigvals = generalizedeigvals[np.abs(generalizedeigvals) >= 10**(-10)]
        eigval_estimates = np.sort(1j*np.log(generalizedeigvals)/tau_increment)
        eigvaldata[2,0:np.size(eigval_estimates),m] = eigval_estimates

        generalizedeigvals, I = s.MatrixPencilMethod(qpedata[m,:],measpoints,L,truncation_factor)
        generalizedeigvals = generalizedeigvals[np.abs(generalizedeigvals) >= 10**(-10)]
        eigval_estimates = np.sort(1j*np.log(generalizedeigvals)/tau_increment)
        eigvaldata[3,0:np.size(eigval_estimates),m] = eigval_estimates

# Plot the Imaginary and Real Time State Evolution
f.propagation_plot(measpoints, tau_increment, numsamples, gfulldata, ftaudata, qpedata, qpenoisydata, M)

# Save State Evolution Data and Eigenvalue Estimate Data
'''
np.save('eigvaldata_g',eigvaldata)
np.save('ftaudata',ftaudata)
np.save('gfulldata',gfulldata)
np.save('qpenoisydata',qpenoisydata)
np.save('qpedata',qpedata)
'''
```

StoqMCfunctions.py

```
import numpy as np
import matplotlib.pyplot as plt

def propagation_plot(measpoints, tau_increment, numsamples, gfulldata, ftaudata, qpedata, qpenoisydata,
    M):
    epsilon = np.sqrt(10/numsamples)
    fig, axs = plt.subplots(1,3)
    plt.subplots_adjust(wspace = 0.35)
    fig.set_size_inches(21, 7.0)
    axs[0].plot(np.arange(measpoints)*tau_increment, gfulldata[0,:], 'k', label='Exact $\\langle \\Phi|
        e^{-\\tau H} |\\Phi \\rangle$')
    axs[0].set_xlabel('$\\tau$', fontsize = 23)
    axs[0].set_ylabel('$\\langle \\Phi| e^{-\\tau H} |\\Phi \\rangle$', fontsize = 20, labelpad = 7)
    axs[0].set_title('$\mathbf{(a)}$ Monte Carlo Scheme.', fontsize = 18)
    axs[0].fill_between(np.arange(measpoints)*tau_increment,gfulldata[0,:]-epsilon,gfulldata[0,:]+
        epsilon,color='lightgrey',label='90% confidence bounds') # Confidence bounds
    axs[0].set_ylim(0,1.05)
    axs[0].set_xlim(-0.01,(measpoints-1)*tau_increment+0.01)
    axs[0].plot(np.arange(measpoints)*tau_increment,ftaudata[0,:],'o',color='firebrick', markersize=5.5,
        label = 'Estimates for $M=$'+str(M))
    axs[0].legend(fontsize=10,prop={'size': 17})
    axs[0].grid(alpha = 0.4)
    axs[0].xaxis.set_tick_params(labelsize=17)
    axs[0].yaxis.set_tick_params(labelsize=17)
```

```
    axs[1].plot(np.arange(measpoints)*tau_increment, np.real(qpedata[0,:]), 'k', label='Exact $Re(\\
        langle \\Phi| e^{-it H} |\\Phi \\rangle)$')
    axs[1].set_xlabel('$t$', fontsize = 23)
    axs[1].set_ylabel('$Re(\\langle \\Phi| e^{-it H} |\\Phi \\rangle)$', fontsize = 20, labelpad = 7)
    axs[1].fill_between(np.arange(measpoints)*tau_increment,np.real(qpedata[0,:])-epsilon,np.real(
        qpedata[0,:])+epsilon,color='lightgrey',label='90% confidence bounds') # Confidence bounds
    axs[1].set_title('$\mathbf{(b)}$ Real Part QPE Signal.', fontsize = 18)
    axs[1].set_ylim(-1.05,1.05)
    axs[1].set_xlim(-0.01,(measpoints-1)*tau_increment+0.01)
    axs[1].plot(np.arange(measpoints)*tau_increment,np.real(qpenoisydata[0,:]),'o',color='royalblue',
        markersize=5.5,label = 'Estimates for $M=$'+str(M))
    axs[1].legend(fontsize=10,prop={'size': 17})
    axs[1].grid(alpha = 0.4)
    axs[1].xaxis.set_tick_params(labelsize=17)
    axs[1].yaxis.set_tick_params(labelsize=17)
    axs[2].plot(np.arange(measpoints)*tau_increment, np.imag(qpedata[0,:]), 'k', label='Exact $Im(\\
        langle \\Phi| e^{-it H} |\\Phi \\rangle)$')
    axs[2].set_xlabel('$t$', fontsize = 23)
    axs[2].set_ylabel('$Im(\\langle \\Phi| e^{-it H} |\\Phi \\rangle)$', fontsize = 20, labelpad = 7)
    axs[2].fill_between(np.arange(measpoints)*tau_increment,np.imag(qpedata[0,:])-epsilon,np.imag(
        qpedata[0,:])+epsilon,color='lightgrey',label='90% confidence bounds') # Confidence bounds
    axs[2].set_title('$\mathbf{(c)}$ Imaginary Part QPE Signal.', fontsize = 18)
    axs[2].set_ylim(-1.05,1.05)
    axs[2].set_xlim(-0.01,(measpoints-1)*tau_increment+0.01)
    axs[2].plot(np.arange(measpoints)*tau_increment,np.imag(qpenoisydata[0,:]),'o',color='seagreen',
        markersize=5.5,label = 'Estimates for $M=$'+str(M))
    axs[2].legend(fontsize=10,prop={'size': 17})
    axs[2].grid(alpha = 0.4)
    axs[2].xaxis.set_tick_params(labelsize=17)
    axs[2].yaxis.set_tick_params(labelsize=17)
```

StoqMCsimulations.py

```
import numpy as np
from scipy.linalg import expm

'''
-------------------------------------------------------------------------
'GtrotSim' takes as input the locality, system size, Trotter variable,
interaction strength, magnetic field, and tau. It outputs an array co-
ntaining the Trotterized imaginary-time propagation operator.
-------------------------------------------------------------------------
'''
def GtrotSim(k,N,M,Jx,Jy,Jz,g,tau):
    Paulis = np.zeros([2,2,4], dtype = np.complex128)
    Paulis[:,:,0] = np.array([[1,0],[0,1]])
    Paulis[:,:,1] = np.array([[0,1],[1,0]])
    Paulis[:,:,2] = np.array([[0,-1j],[1j,0]])
    Paulis[:,:,3] = np.array([[1,0],[0,-1]])
    seqint = np.zeros((k,3))
    Int = np.zeros([2**k,2**k,3])
    MagInt = np.zeros([2**k,2**k,3])
    seqmagx = np.zeros(1)
    Gsingle = np.zeros([2**k,2**k,1])
    Gtrot = np.zeros([2**k,2**k,M*N])

    for j in range(k):
        seqint[j,0] = 1
        seqint[j,1] = 3
        seqint[j,2] = 2
        if j == 0:
            seqmagx[j] = 1

    # Interactions with neighbouring sites
    for m in range(3):
        for i in range(k-1):
            if i == 0:
                Intper = np.kron(Paulis[:,:,int(seqint[i,m])],Paulis[:,:,int(seqint[i+1,m])])
            else:
                Intper = np.kron(Intper,Paulis[:,:,int(seqint[i+1,m])])
        Int[:,:,m] = Intper.real
    # Interactions with external (transverse) magnetic field
    for i in range(k-1):
        if i == 0:
            MagInt = np.kron(Paulis[:,:,int(seqmagx[i])],Paulis[:,:,0])
        else:
            MagInt = np.kron(MagInt,Paulis[:,:,0])

    subHi = Jx*Int[:,:,0] + Jz*Int[:,:,1] + Jy*Int[:,:,2] + g*MagInt
    # Rescaling subHi such that lambda_min = 0
    eigvaluesubHi = np.linalg.eig(subHi)[0]
    eigvectorsubHi = np.linalg.eig(subHi)[1]
    subHiRescaled = np.matmul(np.matmul(eigvectorsubHi,(np.diag(eigvaluesubHi) - np.amin(eigvaluesubHi)*
        np.eye(2**k))),np.linalg.inv(eigvectorsubHi))

    Gsingle = expm(-tau*subHiRescaled/M)
    Gtrot = np.repeat(Gsingle[:,:,np.newaxis], M*N, axis = 2)
    Gtrot[np.abs(Gtrot) < 10**(-10)] = 0

    return Gtrot

'''
-------------------------------------------------------------------------
'DiagBlocksEig' takes as input the system size, Trotter variable, and
Gtrot. It outputs the maximum eigenvalue and associated strictly posi-
tive eigenvector of all irreducible and element-wise non-negative blo-
```

```
cks along the diagonal of the local propagation operators.
-------------------------------------------------------------------------
'''
def DiagBlocksEig(M,N,Gtrot):
    maxeigval = np.zeros((np.shape(Gtrot[:,:,0])[0],M*N))
    poseigvec = np.zeros((np.shape(Gtrot[:,:,0])[0],np.shape(Gtrot[:,:,0])[0],M*N))
    connectedstates = np.ones((np.size(Gtrot[:,:,0])+1,np.size(Gtrot[:,:,0])+1,M*N))*(-1)
    numsets = np.zeros(M*N)

    for t in range(M*N):
        connectedindices = np.c_[np.nonzero(Gtrot[:,:,t])[0],np.nonzero(Gtrot[:,:,t])[1]]
        numsetssing = 0
        csetcheck = np.array([[-1,-1],[-1,-1]])

        for m in range(np.shape(connectedindices)[0]):
            cset = np.reshape(connectedindices[m,:], (1,2))
            if ((np.reshape(cset.astype(int), (2,))).tolist() in csetcheck.tolist()) == False:

                n = 0
                while n < np.shape(cset)[0]:
                    startnode = cset[n,1]
                    n += 1
                    newnodes = np.where(connectedindices[:,0] == startnode)
                    for i in range(np.size(newnodes)):
                        cset = np.unique(np.vstack((cset, connectedindices[np.asarray(newnodes)[0,i],:])
                            ), axis = 0)

                csetcheck = np.concatenate((csetcheck,cset), axis = 0)
                connectedset = cset

                Gb = np.zeros((int((np.shape(connectedset)[0])**(1/2)),int((np.shape(connectedset)[0])
                    **(1/2))))
                for i in range(int((np.shape(connectedset)[0])**(1/2))):
                    for j in range(int((np.shape(connectedset)[0])**(1/2))):
                        Gb[i,j] = ((Gtrot[:,:,t])[connectedset[:,0],connectedset[:,1]])[i*int((np.shape(
                            connectedset)[0])**(1/2))+j].real

                maxeigval[m,t] = np.max(np.linalg.eig(Gb)[0])
                poseigvec[0:int((np.shape(connectedset)[0])**(1/2)),numsetssing,t] = abs((np.linalg.eig(
                    Gb)[1])[:,np.argmax(np.linalg.eig(Gb)[0])])

                connectedstates[0:np.shape(connectedset)[0],(2*numsetssing):(2*numsetssing+2),t] =
                    connectedset
                numsetssing = numsetssing + 1
        numsets[t] = numsetssing

    return maxeigval, poseigvec, connectedstates, numsets

'''
-------------------------------------------------------------------------
'PiandFtau' takes as input the locality, system size, Trotter variable
, the set of connected states and the associated maximum eigenvalues
and strictly positive eigenvectors of the blocks along the diagonal of
the local propagation operators, Gtrot, numsamples and the elements of
the state Phi. It implements an N-term first-order Trotterization sch-
eme and sets up the stochastic process to estimate F(tau). It outputs
the estimate of F(tau).
-------------------------------------------------------------------------
'''
def PiandFTau(maxeigval,poseigvec,connectedstates,numsets,M,N,k,Gtrot,numsamples,phi_coefficients):
    ftau = 0
    for samples in range(numsamples):
        zeroth_state = (np.where(np.cumsum(phi_coefficients**2) > np.random.rand(1))[0])[0]
        x0 = np.reshape(np.flip(np.unpackbits(zeroth_state.astype(np.uint8),axis=0),axis=0)[0:N], (N,1))
        phi0 = phi_coefficients[zeroth_state]

        xMN = np.zeros((N,1))

        x = np.concatenate((np.concatenate((x0, np.zeros((N,M*N-1))), axis = 1), xMN), axis = 1)
        P = np.zeros(M*N)
        singleR = np.zeros(M*N)

        for t in range(M*N):
            if t == 0:
                xleft = np.roll(x[:,t], -t)[0:k]
            xleft_dec = int(np.array2string(xleft.astype(int)).replace('[','').replace(']','').replace('
                ',''), 2)

            for c in range(int(numsets[t])):
                if np.any(connectedstates[:,2*c,t] == xleft_dec) == True:
                    rightset = c
            lambdat = ((maxeigval[:,t])[maxeigval[:,t] != 0])[rightset]
            randnum = np.random.rand(1)

            if t < M*N:
                Pot_xright_dec = np.ones((2**k)**2)*(2**k + 1)
                Pot_xright_dec = connectedstates[np.argwhere(connectedstates[:,2*rightset,t] ==
                    xleft_dec),2*rightset+1,t]
                Pot_xright_dec = Pot_xright_dec[Pot_xright_dec != 2**k + 1]

                phix_t = np.zeros(np.size(Pot_xright_dec))
                phix_tmin = np.zeros(np.size(Pot_xright_dec))
                Prob = np.zeros(np.size(Pot_xright_dec))
                for i in range(np.size(Pot_xright_dec)):
                    eigindexright = np.argwhere(np.unique(connectedstates[:,2*rightset,t]) ==
                        Pot_xright_dec[i]) - 1
                    phix_t[i] = poseigvec[eigindexright,rightset,t]
                    eigindexleft = np.argwhere(np.unique(connectedstates[:,2*rightset,t]) == xleft_dec)
                        - 1
                    phix_tmin[i] = poseigvec[eigindexleft,rightset,t]
```

```
                        Prob[i] = (Gtrot[int(xleft_dec),int(Pot_xright_dec[i]),t]/lambdat*phix_t[i]/
                             phix_tmin[i]).real
                        if randnum < np.sum(Prob):
                            singleR[t] = lambdat*phix_tmin[i]/phix_t[i]
                            P[t] = Prob[i]
                            xright_dec = Pot_xright_dec[i]
                            xright = np.flip(np.flip(np.unpackbits(xright_dec.astype(np.uint8)),axis=0)[0:k
                                ],axis=0)
                            x[:,t + 1] = np.roll(np.concatenate((xright,np.roll(x[:,t], -(t))[k:N])), t)
                            xleft = (np.roll(x[:, t + 1], -(t + 1)))[0:k]
                            break

                if t == M*N-1:
                    Lth_state_bin = x[:,t+1]
                    Lth_state = int(np.array2string(Lth_state_bin.astype(int)).replace('[','').replace(']','
                        ').replace(' ',''), 2)
                    phiL = phi_coefficients[Lth_state]

            Pi = np.prod(P[0:M*N])
            R = np.prod(singleR)
            estimatorforF = phiL/phi0*R
            ftau = ftau + estimatorforF/numsamples

        return ftau

'''
-------------------------------------------------------------------------
'PiandFtauCB' takes as input the locality, system size, Trotter variable, the set of connected states and the associated maximum eigenvalues
and strictly positive eigenvectors of the blocks along the diagonal of
the local propagation operators, Gtrot, numsamples and the elements of
the state Phi. It implements an Gamma-term (checkerboard for k=2 and
even system size) first-order Trotterization scheme and sets up the stochastic process to estimate F(tau). It outputs the estimate of F(tau).
-------------------------------------------------------------------------
'''
def PiandFTauCB(maxeigval,poseigvec,connectedstates,numsets,M,N,k,Gtrot,numsamples,phi_coefficients):
    ftau = 0
    for samples in range(numsamples):
        zeroth_state = (np.where(np.cumsum(phi_coefficients**2) > np.random.rand(1))[0])[0]
        x0 = np.reshape(np.flip(np.unpackbits(zeroth_state.astype(np.uint8),axis=0),axis=0)[0:N], (N,1))
        phi0 = phi_coefficients[zeroth_state]

        xM2 = np.zeros((N,1))

        x = np.concatenate((np.concatenate((x0, np.zeros((N,M*2-1))), axis = 1), xM2), axis = 1)
        P = np.zeros(M*N)
        singleR = np.zeros(M*N)

        for t in range(M*2):
            xleft_dec = np.zeros(np.int(N/2))
            rightset = np.zeros(np.int(N/2))
            lambdat = np.zeros(np.int(N/2))
            randnum = np.random.rand(np.int(N/2))
            xright_saved = np.zeros(N)

            if t == 0:
                xleft = x[:,0]
            for i in range(np.int(N/2)):
                if np.remainder(t,2) == 0:
                    xleft_dec[i] = int(np.array2string(xleft[2*i:2*i+2].astype(int)).replace('[','').
                        replace(']','').replace(' ',''), 2)
                if np.remainder(t,2) == 1:
                    xleft_dec[i] = int(np.array2string((np.roll(xleft,-1)[2*i:2*i+2]).astype(int)).
                        replace('[','').replace(']','').replace(' ',''), 2)

                for c in range(int(numsets[np.int(np.round(t/2)*N/2 + np.remainder(t,2) + 2*i)])):
                    if np.any(connectedstates[:,2*c,np.int(np.round(t/2)*N/2 + np.remainder(t,2) + 2*i)]
                        == xleft_dec[i]) == True:
                        rightset[i] = c
                lambdat[i] = ((maxeigval[:,np.int(np.round(t/2)*N/2 + np.remainder(t,2) + 2*i)])[
                    maxeigval[:,np.int(np.round(t/2)*N/2 + np.remainder(t,2) + 2*i)] != 0])[np.int(
                    rightset[i])]

            if t < M*2:
                for i in range(np.int(N/2)):
                    Pot_xright_dec = np.ones((2**2)**2)*(2**2 + 1)
                    Pot_xright_dec = connectedstates[np.argwhere(connectedstates[:,2*np.int(rightset[i])
                        ,np.int(np.round(t/2)*N/2 + np.remainder(t,2) + 2*i)] == xleft_dec[i]),2*np.int
                        (rightset[i])+1,np.int(np.round(t/2)*N/2 + np.remainder(t,2) + 2*i)]
                    Pot_xright_dec = Pot_xright_dec[Pot_xright_dec != 2**2 + 1]

                    phix_t = np.zeros(np.size(Pot_xright_dec))
                    phix_tmin = np.zeros(np.size(Pot_xright_dec))
                    Prob = np.zeros(np.size(Pot_xright_dec))
                    for p in range(np.size(Pot_xright_dec)):
                        eigindexright = np.argwhere(np.unique(connectedstates[:,2*np.int(rightset[i]),np
                            .int(np.round(t/2)*N/2 + np.remainder(t,2) + 2*i)]) == Pot_xright_dec[p]) -
                            1
                        phix_t[p] = poseigvec[eigindexright,np.int(rightset[i]),np.int(np.round(t/2)*N/2
                            + np.remainder(t,2) + 2*i)]
                        eigindexleft = np.argwhere(np.unique(connectedstates[:,2*np.int(rightset[i]),np.
                            int(np.round(t/2)*N/2 + np.remainder(t,2) + 2*i)]) == xleft_dec[i]) - 1
                        phix_tmin[p] = poseigvec[eigindexleft,np.int(rightset[i]),np.int(np.round(t/2)*N
                            /2 + np.remainder(t,2) + 2*i)]
                        Prob[p] = (Gtrot[int(xleft_dec[i]),int(Pot_xright_dec[p]),np.int(np.round(t/2)*N
                            /2 + np.remainder(t,2) + 2*i)]/lambdat[i]*phix_t[p]/phix_tmin[p]).real
                        if randnum[i] < np.sum(Prob):
                            singleR[np.int(t*N/2 + i)] = lambdat[i]*phix_tmin[p]/phix_t[p]
                            P[np.int(t*N/2 + i)] = Prob[p]
```

```
                                xright_dec = Pot_xright_dec[p]
                                xright = np.flip(np.flip(np.unpackbits(xright_dec.astype(np.uint8)),axis=0)
                                    [0:2],axis=0)
                                xright_saved[2*i:2*i+2] = xright
                                break

                    if np.remainder(t,2) == 1:
                        xright_saved = np.roll(xright_saved, 1)

                    x[:,t + 1] = xright_saved
                    xleft = x[:,t + 1]

                if t == M*2-1:
                    Lth_state_bin = x[:,t + 1]
                    Lth_state = int(np.array2string(Lth_state_bin.astype(int)).replace('[','').replace(']','
                        ').replace(' ',''), 2)
                    phiL = phi_coefficients[Lth_state]

            Pi = np.prod(P[0:M*N])
            R = np.prod(singleR)
            estimatorforF = phiL/phi0*R
            ftau = ftau + estimatorforF/numsamples

    return ftau

'''
-------------------------------------------------------------------------
'FullHandQPE' takes as input the locality, system size, Trotter varia-
ble, interaction strength, magnetic field, tau, elements of the state
Phi, numsamples and the choice of Trotterization schemes. It (!ineffi-
ciently!) implements the exact diagonalization of H, the exact calcul-
ation of the MC signal, and the QPE scheme.
-------------------------------------------------------------------------
'''
def fullHandQPE(k,N,M,Jx,Jy,Jz,g,tau,phi_coefficients,numsamples,checkerboard):
    Paulis = np.zeros([2,2,4], dtype = np.complex128)
    Paulis[:,:,0] = np.array([[1,0],[0,1]])
    Paulis[:,:,1] = np.array([[0,1],[1,0]])
    Paulis[:,:,2] = np.array([[0,-1j],[1j,0]])
    Paulis[:,:,3] = np.array([[1,0],[0,-1]])
    Intcompl = np.zeros([2**N,2**N,3*N])
    MagIntcompl = np.zeros([2**N,2**N,N])
    seqmagx = np.zeros(N)
    seqint = np.zeros((N,3))
    H_i = np.zeros([2**N,2**N,N])
    HiRescaled = np.zeros([2**N,2**N,N])
    HRescaled = np.zeros([2**N,2**N])
    G = np.zeros([2**N,2**N])
    GtrotQPE = np.zeros((2**N,2**N,N),dtype=np.complex)
    GtrotQPEsingleoperator = np.zeros((2**N,2**N),dtype=np.complex)
    GnontrotQPE = np.zeros((2**N,2**N),dtype=np.complex)

    for j in range(k):
        seqint[j,0] = 1
        seqint[j,1] = 3
        seqint[j,2] = 2
        if j == 0:
            seqmagx[j] = 1

    for i in range(N):
        for m in range(3):
            l = 2
            for p in range(N-1):
                if p == 0:
                    Intpercompl = np.kron(Paulis[:,:,int(seqint[np.remainder(i,N),m])],Paulis[:,:,int(
                        seqint[np.remainder(i+1,N),m])])
                else:
                    Intpercompl = np.kron(Intpercompl,Paulis[:,:,int(seqint[np.remainder(i+l,N),m])])
                    l = l+1
            Intcompl[:,:,i + N*m] = Intpercompl.real

        l = 2
        for p in range(N-1):
            if p == 0:
                Intmagpercompl = np.kron(Paulis[:,:,int(seqmagx[np.remainder(i,N)])],Paulis[:,:,int(
                    seqmagx[np.remainder(i+1,N)])])
            else:
                Intmagpercompl = np.kron(Intmagpercompl,Paulis[:,:,int(seqmagx[np.remainder(i+l,N)])])
                l = l+1
        MagIntcompl[:,:,i] = Intmagpercompl.real

    for i in range(N):
        H_i[:,:,i] = Jx*Intcompl[:,:,i] + Jz*Intcompl[:,:,N+i] + Jy*Intcompl[:,:,2*N+i] + g*MagIntcompl
            [:,:,i]
        # Rescaling each H_i such that lambda_min(H_i) = 0
        eigvalueHi = np.linalg.eig(H_i[:,:,i])[0]
        eigvectorHi = np.linalg.eig(H_i[:,:,i])[1]
        HiRescaled[:,:,i] = np.matmul(np.matmul(eigvectorHi,(np.diag(eigvalueHi) - np.amin(eigvalueHi)*
            np.eye(2**N))),np.linalg.inv(eigvectorHi))
        #HiRescaled[:,:,i] = H_i[:,:,i]
        GtrotQPE[:,:,i] = expm(-(1j)*tau*HiRescaled[:,:,i]/M)
        HRescaled = HRescaled + HiRescaled[:,:,i]
    FullHeigvals = np.linalg.eig(HRescaled)[0]
    FullHeigvecs = np.linalg.eig(HRescaled)[1]
    G = (expm(-tau*HRescaled)).real
    GnontrotQPE = expm(-(1j)*tau*HRescaled)
    realt_evolution_overlap_nontrot = np.matmul(np.matmul(np.transpose(phi_coefficients),GnontrotQPE),
        phi_coefficients)

    # Calculating Trotter Bounds
```

```
    # First-Order N-Term Decomposition
    commutatornorm = 0
    for i in range(N-1):
        for n in range(N-1-i):
            j = i + 1 + n
            commutatornorm = commutatornorm + np.linalg.norm(np.matmul(HiRescaled[:,:,i],HiRescaled[:,:,
                j]) - np.matmul(HiRescaled[:,:,j],HiRescaled[:,:,i]), ord = 2)
    trivialtrotterbound = commutatornorm*tau**2/(2*M)
    # First-Order Checkerboard Decomposition
    H_even = np.zeros((2**N,2**N))
    H_odd = np.zeros((2**N,2**N))
    for i in range(N):
        if np.remainder(i,2) == 0:
            H_even = H_even + HiRescaled[:,:,i]
        if np.remainder(i,2) == 1:
            H_odd = H_odd + HiRescaled[:,:,i]
    CBtrotterbound = np.linalg.norm(np.matmul(H_even,H_odd) - np.matmul(H_odd,H_even), ord = 2)*tau
        **2/(2*M)
    '''
    normcheck = 0
    for i in range(N):
        normcheck = normcheck + np.linalg.norm(HiRescaled[:,:,i], ord = 2)
    print(normcheck*tau/M)
    '''

    GtrotQPEsingleoperator = GtrotQPE[:,:,0]
    for i in range(N-1):
        if checkerboard == 0:
            GtrotQPEsingleoperator = np.matmul(GtrotQPEsingleoperator,GtrotQPE[:,:,i+1])
        if checkerboard == 1:
            if i < np.int(N/2)-1:
                GtrotQPEsingleoperator = np.matmul(GtrotQPEsingleoperator,GtrotQPE[:,:,2*(i+1)])
            if i >= np.int(N/2)-1:
                GtrotQPEsingleoperator = np.matmul(GtrotQPEsingleoperator,GtrotQPE[:,:,2*(i-np.int(N/2)
                    +1)+1])
    GtrotQPEsingleoperator = np.linalg.matrix_power(GtrotQPEsingleoperator, M)
    realt_evolution_overlap = np.matmul(np.matmul(np.transpose(phi_coefficients),GtrotQPEsingleoperator)
        ,phi_coefficients)

    # sampling from real and imaginary parts of realt_evolution_overlap
    realpart_overlap = np.real(realt_evolution_overlap)
    imagpart_overlap = np.imag(realt_evolution_overlap)
    prob_ancilla_zero = 1/2 + 1/2*realpart_overlap
    prob_ancilla_pitwo = 1/2 - 1/2*imagpart_overlap

    rand_zero = np.random.rand(numsamples, 1)
    rand_pitwo = np.random.rand(numsamples, 1)

    prob_ancilla_zero_sampled = np.mean((rand_zero < prob_ancilla_zero).astype(int))
    prob_ancilla_pitwo_sampled = np.mean((rand_pitwo < prob_ancilla_pitwo).astype(int))

    sampled_realpart_overlap = 2*prob_ancilla_zero_sampled - 1
    sampled_imagpart_overlap = 1 - 2*prob_ancilla_pitwo_sampled

    return G, FullHeigvals, FullHeigvecs, sampled_realpart_overlap, sampled_imagpart_overlap,
        realt_evolution_overlap_nontrot, trivialtrotterbound, CBtrotterbound

'''
-------------------------------------------------------------------------
'MatrixPencilMethod' takes as input a (noisy) signal, measpoints, the
pencil parameter L and the truncation factor. It outputs the estimates
of the parameters z_{j}.
-------------------------------------------------------------------------
'''
def MatrixPencilMethod(signal,measpoints,L,truncation_factor):

    Y = np.zeros((measpoints-L,L + 1), dtype = np.complex)
    for i in range(measpoints-L):
        Y[i,:] = signal[i:L + i + 1]

    # Without truncating the Y1 and Y2 SVD's (unstable when signal is noisy)
    '''
    Y1 = Y[:,0:L]
    Y2 = Y[:,1:L + 1]
    generalizedeigvals = np.linalg.eig(np.matmul(np.linalg.pinv(Y2),Y1))[0]
    generalizedeigvals = generalizedeigvals[generalizedeigvals > 10**(-10)]
    I = np.matmul(np.linalg.pinv(Y2),Y2)
    '''

    # Truncating the Y1 and Y2 SVD's (allows for noisy signal)
    Y1 = Y[:,0:L]
    Y2 = Y[:,1:L + 1]
    u1, sigma1, vh1 = np.linalg.svd(Y1) # SVD Y1
    u2, sigma2, vh2 = np.linalg.svd(Y2) # SVD Y2
    # Locating the set of largest singular values
    relevantargs1 = np.argwhere(sigma1/np.max(sigma1) > truncation_factor)
    relevantargs2 = np.argwhere(sigma2/np.max(sigma2) > truncation_factor)

    sigma1 = sigma1[sigma1/np.max(sigma1) > truncation_factor]
    sigma_matrix1 = np.diag(sigma1)
    sigma2 = sigma2[sigma2/np.max(sigma2) > truncation_factor]
    sigma_matrix2 = np.diag(sigma2)

    vh1prime = np.transpose(np.take(np.transpose(vh1),relevantargs1,axis = 1))
    vh2prime = np.transpose(np.take(np.transpose(vh2),relevantargs2,axis = 1))
    u1prime = np.take(u1,relevantargs1,axis = 1)
    u2prime = np.take(u2,relevantargs2,axis = 1)

    vh1prime = np.reshape(vh1prime, (np.shape(vh1prime)[1],np.shape(vh1prime)[2]))
    vh2prime = np.reshape(vh2prime, (np.shape(vh2prime)[1],np.shape(vh2prime)[2]))
```

```
    u1prime = np.reshape(u1prime, (np.shape(u1prime)[0],np.shape(u1prime)[1]))
    u2prime = np.reshape(u2prime, (np.shape(u2prime)[0],np.shape(u2prime)[1]))

    Y1_truncatedSVD = np.matmul(np.matmul(u1prime, sigma_matrix1),vh1prime)
    Y2_truncatedSVD = np.matmul(np.matmul(u2prime, sigma_matrix2),vh2prime)
    generalizedeigvals = np.linalg.eig(np.matmul(np.linalg.pinv(Y1_truncatedSVD),Y2_truncatedSVD))[0]
    I = np.matmul(np.linalg.pinv(Y1_truncatedSVD),Y1_truncatedSVD) # Check

    return generalizedeigvals, I
```