

## ABSTRACT

Title of Dissertation: **QUANTUM ALGORITHMS FOR  
THERMAL STATE PREPARATION AND  
THE SUPPRESSION OF GAUGE DRIFT**

**Carter Ball**  
Doctor of Philosophy, 2025

Dissertation Directed by: **Professor Thomas Cohen**  
**Department of Physics**

The study of quantum computers and quantum algorithms has captured the attention of physics globally in recent decades. While some researchers are working towards constructing fault-tolerant large-scale quantum computers, others are theorizing about the capabilities of these new computers and developing quantum algorithms to leverage their increased abilities. This dissertation engages with the second area of research concerning quantum algorithms.

The first part of this dissertation discusses quantum algorithms for state preparation, with a focus on thermal state preparation. A selection of other state preparation methods is briefly summarized before a novel method of thermal state preparation is laid out wherein a so-called heat pump is employed in a technique of active cooling.

The second part of this dissertation discusses quantum algorithms that aim to address the problem of gauge violation during the simulation of lattice gauge theories. The problem of gauge violation in both quantum analog and digital simulations is presented before two related methods

are delineated. These methods leverage the quantum Zeno effect with frequent measurements on a system's physical subspace to keep the system's state physical throughout the course of a simulation. Furthermore, the properties of gauge transformations are utilized to help curtail the growth of unwanted unphysical amplitudes.

QUANTUM ALGORITHMS FOR THERMAL STATE PREPARATION  
AND THE SUPPRESSION OF GAUGE DRIFT

by

Carter Ball

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2025

Advisory Committee:

Professor Thomas D. Cohen, Chair/Advisor  
Professor Paulo Bedaque  
Professor Anson Hook  
Professor Carter Hall  
Professor Christopher Jarzynski

© Copyright by  
Carter Ball  
2025

## Dedication

I dedicate this dissertation to my Papa, who fostered a curiosity for the natural world within countless people throughout his life and served as a greatly influential role model for me. It is the honor of my life to call you my grandfather and I carry you with me every day.

## Acknowledgments

I am thankful to my Ph.D. advisor Tom Cohen for 4 years of collaboration and mentorship. I have enjoyed working on various projects together and appreciate the space I was afforded to work out my relationship to physics and academia. I am also thankful to my co-advisee Yukari Yamauchi for welcoming me to the University of Maryland and supporting me through the learning curve of graduate research.

I was fortunate to have many great faculty members around me, many of whom I had the pleasure of taking a course from. I greatly appreciate Kaustubh Agashe, Paulo Bedaque, Victor Yakovenko, James Drake, Zohreh Davoudi, Anson Hook, Tom Cohen, and Xiangdong Ji for their instruction and guidance.

I would also like to thank my friends, in particular Zoe Irving and Theo Palmisciano, for keeping me company throughout grad school. I very much cherish our friendships and look forward to a lifetime of fun together.

Finally, I would not be here without my family: Margo Ball, Damon Ball, Ethan Ball, Monique Desnoyers, Ruth Ann Buchanan, and James Buchanan. I thank each of you from the bottom of my heart for all the support you have given me throughout my life.

# Table of Contents

Dedication	ii
Acknowledgements	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
Chapter 1: Introduction	1
1.1 Overview	1
1.2 Quantum computation and simulation	3
1.3 Lattice gauge theory	6
Chapter 2: State preparation	9
2.1 Ground state preparation	9
2.2 Thermal state preparation	13
2.3 The active cooling method	16
Chapter 3: Suppression of gauge violations	25
3.1 Overview	25
3.2 Analog approaches	26
3.3 Digital approaches	27
3.4 The quantum Zeno effect	29
3.5 Zeno effect suppression of gauge drift	32
3.6 Suppression of gauge drift using gauge transformations	36
Chapter 4: Summary	41
Appendix A: Boltzmann distributions on a quantum computer via active cooling	44
A.1 Introduction	45
A.1.1 State preparation methods	49
A.2 The Active Cooling Scheme	53
A.2.1 Parameters	62
A.3 A Toy Model	64
A.3.1 The algorithm	66

A.3.2	Results	70
A.4	Generalization to larger systems	73
A.4.1	The refrigerator as a thermometer	73
A.4.2	Lattice models	74
A.4.3	Algorithm for large lattice systems	76
A.5	Conclusion	77
Appendix B: Zeno effect suppression of gauge drift in quantum simulations		78
B.1	Introduction	79
B.2	Method	82
B.2.1	Projection via the Rodeo Algorithm	82
B.2.2	Suppressing Coherent Gauge Drift	87
B.2.3	Calibration	89
B.3	Performance	91
B.4	Implementation	96
B.4.1	Implementing Rodeo Projections	97
B.4.2	Implementing Gauge Transformations	98
B.5	Conclusion	99
Appendix C: Suppressing gauge drift in quantum simulations with gauge transformations		101
C.1	Introduction	102
C.2	Method	105
C.3	Performance	113
C.4	Implementation	122
C.5	Conclusion	123
Bibliography		125

## List of Tables

C.1	The average $\langle G^2 \rangle$ value for four different methods of reusing gauge transformations throughout a simulation run . . . . .	121
-----	---	-----

## List of Figures

2.1	The circuit diagram for a quantum phase estimation algorithm [1] . . . . .	10
2.2	The circuit diagram for the rodeo algorithm [2] . . . . .	10
2.3	The logic diagram for the demon-like algorithmic cooling method [3]. The refer- ences to the variable $x$ are in analogy to a 1D random walker beginning at $x = 0$ , with a cooling result from the cooling module represented as a +1 step while a heating result is represented as a -1 step. . . . .	12
2.4	A diagram of the 4-step active cooling cycle . . . . .	18
2.5	Data (colored bars) vs Expected (black lines) . . . . .	23
3.1	Circuit diagram of the Zeno effect projection . . . . .	33
3.2	Comparing gauge drift suppression methods, including no suppression methods (dash-dot line), a random gauge transformation after each time step (dashed line), and a random gauge transformation followed by a projection after each time step (solid line). The right plot is the same as the left plot without the dash-dot plot so as to zoom in on the behavior of the other two. . . . .	35
3.3	Comparing performance of regimes with no gauge transformations and various frequencies of projections (circle markers) with regimes with gauge transforma- tions and various frequencies of projections (square markers) . . . . .	36
3.4	Circuit diagram of the projection used to suppress gauge drift . . . . .	37
3.5	Results from three simulations. The control simulation (green dashed-dot line) ran without any gauge drift mitigation techniques. The second (orange dashed line) simulated gauge drift with the method of gauge transformations after every time step (method 1). The third (blue solid line) simulated gauge drift with a gauge transformation (method 1) and a projection of the kind in Fig. 3.4 (method 2) after every time step. . . . .	39
A.1	The active cooling cycle . . . . .	57
A.2	Data (colored bars) vs Expected (black lines) . . . . .	72
B.1	The circuit diagram for the rodeo algorithm [2] . . . . .	83
B.2	The diagram of the toy model with 4 sites and 8 links . . . . .	91
B.3	Comparing gauge drift suppression methods, including no suppression methods (dash-dot line), a random gauge transformation after each time step (dashed line), and a random gauge transformation followed by a projection after each time step (solid line). The right plot is the same as the left plot without the dash-dot plot so as to zoom in on the behavior of the other two. . . . .	95

B.4	Comparing performance of regimes with no gauge transformations and various frequencies of projections (circle markers) with regimes with gauge transformations and various frequencies of projections (square markers) . . . . .	96
C.1	Circuit diagram of the projection used to suppress gauge drift . . . . .	107
C.2	The diagram of the 1D toy model with periodic boundary conditions, four sites (labelled 1-4), and four links (labelled A-D). . . . .	113
C.3	Results from three simulation runs: a control run, dashed-dot line, without any gauge drift suppression methods; a run, dashed line, using method 1 (conducting a gauge transformation after every time step); and a run, solid line, using both method 1 and method 2 (conducting a projection after every time step). . . . .	117
C.4	A closer look at the run from Fig. C.3 utilizing both methods of gauge drift suppression. . . . .	118

## Chapter 1: Introduction

### 1.1 Overview

The centuries old history of algorithms has cemented their relevance to the fields of mathematics, physics, chemistry, and computer science. They have proved vital to the study of computation. A massive movement has formed in recent decades around a new form of computation, called quantum computation. The study of quantum computation, from its basic building blocks to the crafting of functional quantum computers to its theorized far-term potential, has captured the attention of numerous scientists. Quantum computation is expected to far outperform classical computation methods for a select class of problems [4]. Importantly, there are many problems that classical computation has great difficulty with and many that classical computation simply cannot solve. A vital promise of quantum computation is to directly address these classically intractable problems.

One corner of these classically difficult problems is the simulation of lattice gauge theories from high energy physics. As lattice gauge theories are vital to the task of describing how the physical world works, the simulation of these fundamental theories promises to provide much insight into many open questions. Classical computers struggle to simulate certain aspects of lattice gauge theories, even with massive supercomputers dedicating their resources and time. Hence, there is much research into the construction of quantum computers. While small quantum

computers have been built, they are as yet nowhere near the size they need to be to get at the big questions that lattice gauge theories aim to address. In the meantime, much research has gone into the construction of quantum algorithms that probe various aspects of lattice gauge theories so that when quantum computers with the appropriate capabilities are built, the research community can hit the ground running.

This dissertation concerns these quantum algorithms that aid in the process of simulating quantum lattice gauge theories. In the next section, I review various aspects of quantum computation and simulation, including the components and the errors that can occur. I briefly review famous quantum algorithms as well as the important the Suzuki-Trotter formula. I conclude the section by defining the two main forms of quantum simulation: analog and digital. In Sec. 1.3, I discuss lattice gauge theory (LGT), starting with the various formulations describing these theories. Then I discuss the redundancy of degrees of freedom that characterize a gauge theory as well as the process of gauge fixing. Finally, I define Gauss laws and physical states.

Chapter 2 concerns what is considered to be one of the most arduous tasks of quantum simulation: state preparation. Sec. 2.1 begins the chapter with a discussion of ground state preparation, covering the need for ground states in quantum simulation as well as various methods. These methods include adiabatic state preparation, quantum phase estimation, projective cooling, the rodeo algorithm, and algorithmic cooling. Then Sec. 2.2 discusses thermal state preparation, noting various situations where thermal states are required. I define what it means for a quantum state to be thermal and discuss the process of thermalization as well as the eigenstate thermalization hypothesis. Finally, I cover thermal state preparation methods such as the heat bath method and the use of minimally entangled typical thermal states (METTS). I conclude the chapter in Sec. 2.3 by outlining the active cooling method, reviewing previous work [5] that can be found

in Appendix A.

In chapter 3, I consider a different kind of quantum algorithm that works to suppress gauge violations. I begin in Sec. 3.1 by briefly mentioning how gauge violations can crop up in quantum simulations. Then in Sec. 3.2 I cover analog approaches to gauge drift suppression, including tying gauge symmetry to an internal symmetry (eg angular momentum) and the energy penalty method. In Sec. 3.3 I cover digital approaches, including solving the Gauss laws, dynamical decoupling, using a Gauss law oracle, and using classical noise to leverage the classical Zeno effect. In Sec. 3.4 I review the quantum Zeno effect before moving into Sec. 3.5 where I review a method of gauge drift suppression that relies on the Zeno effect, referencing previous work [6] that can be found in Appendix B. I conclude the chapter in Sec. 3.6 by discussing a method of gauge drift suppression that relies on gauge transformations as well as the Zeno effect, reviewing previous work [7] that can be found in Appendix C. I conclude this dissertation in chapter 4 by summarizing the discussions in the previous three chapters with a particular focus on my work that forms the center of this dissertation.

## 1.2 Quantum computation and simulation

In 1981, Richard Feynman referenced a new kind of computer, i.e. a quantum computer, in a discussion of the simulation of quantum physics using quantum mechanical objects [8]. In the ensuing decades, an explosion of research has aimed to understand, build, and utilize quantum computers. The basic building block of a quantum computer, a qubit, is a two-level quantum system whose state can be written  $|q\rangle = c_1|0\rangle + c_2|1\rangle$  for complex amplitudes  $c_i$  and basis states  $|0\rangle$  and  $|1\rangle$ , often also called  $|\uparrow\rangle$  and  $|\downarrow\rangle$ , respectively, in analogy to a single spin system. A

common analogy to describe the increase of computational power when moving from classical computation to quantum computation describes the classical bit (with states 0 or 1) as a on/off light switch while a qubit is like a dimmer switch.

With the ability to store much more information than classical bits, qubits are a fundamental aspect of quantum computation; other important aspects include quantum gates and measurement. A quantum gate is a unitary operator acting on a set of qubits while quantum measurement is a non-unitary operation that collapses a qubit (or set of qubits) to a basis state. These parts of a quantum computer are vital for connecting the qubits together, performing computations, and returning results. Much work has gone into developing theory around these gates; multiple universal quantum gate sets have been delineated, where any unitary quantum gate can be reproduced as a finite sequence of gates from the universal set.

While quantum computation shows great promise at significantly increasing computational power, there are multiple sources of error that must be mitigated. This has spawned the field of quantum error correction [9] which has been an active area of research for the last few decades. Perhaps the biggest obstacle to efficient quantum computation is decoherence, where the quantum system loses information to the environment. Left unmitigated, a quantum system could fully decohere and thus be completely disconnected from any physical relevance. Other errors can occur throughout a simulation, including misapplied gates, qubits wrongly flipped, and measurements gone awry.

As quantum systems are fairly delicate, there is also the consideration of quantum noise, which describes the effects of quantum level disturbances on the system. These disturbances can include unwanted photons interacting with the system, physical vibrations, and thermal fluctuations. Due to all these sources of error, the current era of quantum computer is called the noisy

intermediate-scale quantum (NISQ) era. This describes intermediate sized quantum computers that are noisy due to their lack of ability to engage in continuous quantum error correction. These NISQ computers are useful in some contexts and have been used to obtain good results for certain problems, but they are quite limited in scope. Much research is working to build fault-tolerant quantum computers and move the field into a less noisy future.

With the current-day NISQ computers and a hopeful future of large-scale fault-tolerant quantum computers on the horizon, a lot of research has additionally gone into the utilization of quantum computers. Many quantum algorithms have been developed in recent decades, serving various purposes from theoretically demonstrating the potential of quantum computers to preparing for when more sophisticated quantum computers are available. The most well-known quantum algorithms are Shor's algorithm [10], which factors a number into its prime factorization, and Grover's algorithm [11], a sorting algorithm. These algorithms sparked much excitement over the much greater potential of quantum computing power in the 90s. Since then, many more quantum algorithms have been developed for a wide variety of purposes.

While these two famed examples highlight the competition between classical and quantum computation, there has also been much work to build quantum algorithms for the purposes of quantum simulation. The conditions for efficient quantum simulation on a quantum computer were developed by Lloyd [12] and subsequent work has discussed the quantum simulation of specifically field theories on quantum computers [13, 14]. With this large body of work, it is important to highlight a common method at the heart of quantum simulation. Any simulation studying a physical system will involve some sort of time evolution. This is canonically represented in quantum mechanics as the time evolution unitary  $e^{-iHt}$ . A problem arises here: this unitary is difficult to construct efficiently on a quantum computer for large times. Thus, the

Suzuki-Trotter formula is utilized:

$$e^{-iHt} \simeq \left( e^{-iH\frac{t}{N}} \right)^N \quad (1.1)$$

This formula breaks the full time unitary down into many smaller time unitaries, which makes time evolution during quantum simulation efficient.

To conclude this section, it is important to clarify that the above discussions have focused primarily on digital quantum simulation. This is the main form of interest to this dissertation; it is the use of qubits, quantum gates, and quantum circuits to simulate quantum systems. In the interest of completeness, it is sensible to briefly define the other popular form of quantum simulation: analog quantum simulation. This method of simulation takes the approach of finding a relation between a system of interest and (some part of) another system. The key here is that the secondary system can be built and controlled in a lab far easier than the system of interest (e.g. a spin system). Then, the system of interest can be simulated by simulating the secondary system in a lab and then translating the lab results to the system of interest using the relation mapping between the two systems.

### 1.3 Lattice gauge theory

As a major part of this dissertation concerns the simulation of specifically lattice gauge theories on a quantum computer, it is prudent to review the basic underpinnings of these theories. A common approach to studying LGTs on classical computers is to employ Markov chain Monte Carlo (MCMC) methods. These methods sample configurations of a physical system on a lattice using the action, ie the integral of the system's Lagrangian. Thus, a Lagrangian formulation of

the LGT is most useful in these cases.

While the MCMC methods for LGTs have had great success, they are not as effective when it comes to quantum simulation. For these considerations, a Hamiltonian formulation has proven to be much more effective. There is a subtlety here, though. For a gauge theory, there is a redundancy in the degrees of freedom of the Lagrangian; this redundancy arises from the Lagrangian being invariant under certain local transformations. The Lie group describing these local transformations define the gauge symmetry of the theory.

When it comes to defining a Hamiltonian for a gauge theory, the gauge field has a time component with no time derivatives and thus no associated canonical momentum. Thus, it is necessary to fix the gauge of the theory to write down a Hamiltonian. This process of fixing a gauge involves selecting a condition that the theory must satisfy, with this condition reducing (not necessarily fully eliminating) the redundancy in the degrees of freedom. A typical choice for the quantum simulation of LGTs is the temporal gauge, where the time component of the gauge field is set to zero.

As the temporal gauge is not a complete gauge (ie a residual redundancy in degrees of freedom remains), there are still gauge considerations to be made. The Euler-Lagrange equation describes this residual gauge freedom within the temporal gauge by providing an equation of constraint. This constraint is in fact the quantum field theory version of Gauss's law from classical electrodynamics when the LGT in question is quantum electrodynamics (QED); for quantum chromodynamics (QCD), this is called the color Gauss law. Moving forward, this equation of constraint will simply be referred to as a Gauss law. This Gauss law is quite important in that it divides the Hilbert space into two sectors: physical states that satisfy the Gauss law and unphysical states that don't satisfy the Gauss law. As will be discussed in detail in chapter 3, it is

often a major hurdle of the quantum simulation of LGTs to simulate only the physically relevant physical states of a system and avoid any contamination from the system's unphysical states.

## Chapter 2: State preparation

### 2.1 Ground state preparation

The first step for any quantum simulation is state preparation. This step, however, has often proved to be one of the hardest in constructing a viable simulation [15–22]. As this is the case, much research has been done to develop various methods of initial state preparation. This section outlines a selection of initial state preparation methods, with a focus on ground state preparation.

The first initial state preparation method of this section is adiabatic state preparation [23–25]. This method is based directly off of the adiabatic theorem [26]: begin with a system in an easy-to-construct ground state of a well-known Hamiltonian and then transform the system’s Hamiltonian slowly into the Hamiltonian of interest. If the Hamiltonian is changed slowly enough, the system will remain in its ground state, and thus end in the desired ground state of the Hamiltonian of interest. For the current NISQ era of quantum computing, this method has a major problem: given the long timescale required to implement this approach, there is a significant risk of decoherence [27].

Another popular method is quantum phase estimation (QPE) [1, 28–33]. This method takes a state that has significant overlap with a desired eigenstate and employs ancillary qubits along with a unitary operator, typically the time evolution operator, to construct the target eigenstate; see Figure 2.1 for an example circuit diagram of this method. This method’s main bottleneck is

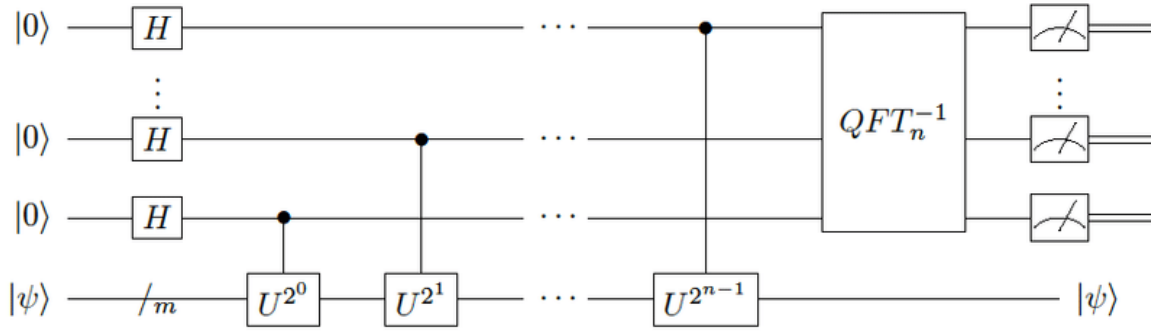


Figure 2.1: The circuit diagram for a quantum phase estimation algorithm [1]

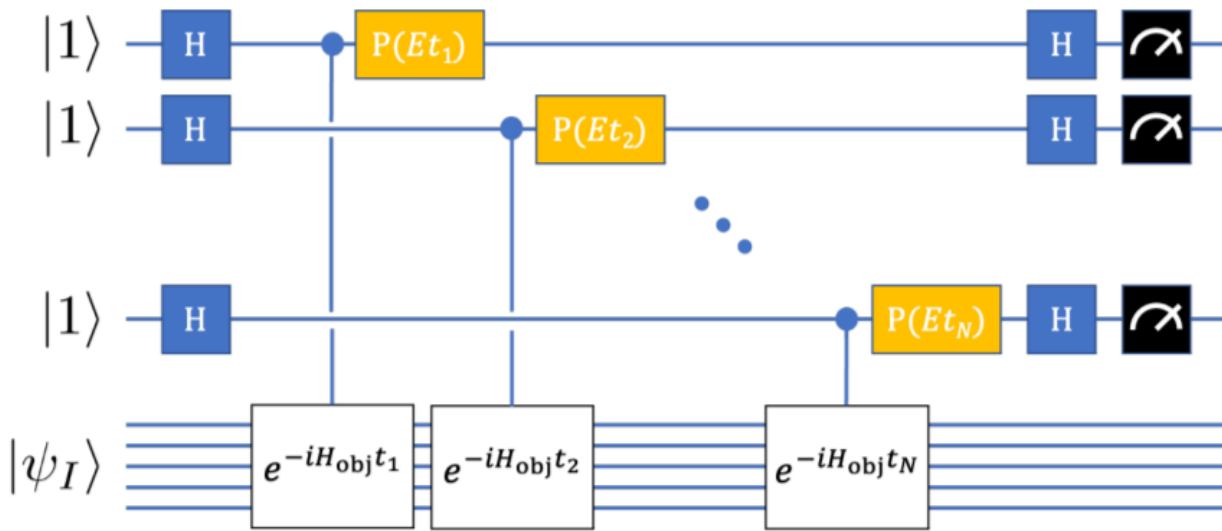


Figure 2.2: The circuit diagram for the rodeo algorithm [2]

the initial input state’s requirement to have significant overlap with the target eigenstate; this is particularly hard when the desired eigenstate is the ground state.

Yet another state preparation method is projective cooling [34, 35]. This method has the system of interest as a subspace of a much larger total system and drives energy excitations out of this region. To finish, the system of interest is measured, ideally into the ground state as its amplitude has been greatly increased by removing excitations to other regions of the total system. As the requirement of a much larger complementary system represents a large cost, a variation on this method was developed called the rodeo algorithm [2, 36–38]. The name of this method

comes from analogy to a rodeo where a bull shakes off its rider, where the rodeo algorithm aims to shake off amplitudes of unwanted eigenstates. This “shaking off” of undesirable amplitudes is done via destructive interference caused by considered use of ancilla qubits; see Figure 2.2 for the rodeo algorithm diagram. Here, an ancilla qubit starts in the  $|1\rangle$  state and is measured at the end; measuring the qubit back into the  $|1\rangle$  state is considered a success. If an ancilla qubit is successfully measured back in its initial state, the system will have undergone the (unnormalized) transformation

$$|\psi\rangle \rightarrow \frac{1}{2}(I + e^{-it(H_{obj}-E)})|\psi\rangle \quad (2.1)$$

where  $H_{obj}$  is the Hamiltonian of interest,  $t$  is a chosen time parameter, and  $E$  is the energy of the eigenstate being constructed. Note here that the rodeo algorithm does not need to be used for ground state preparation; it can target any energy eigenstate of known energy  $E$ . As equation (2.1) shows, any energy eigenstate with energy  $E$  is unaffected by the rodeo algorithm while all others experience destructive interference.

Note that the rodeo algorithm circuit diagram looks rather like the circuit diagram for QPE. It is instructive to note however that each instance of the unitary operator  $U$  in Fig. 2.1, typically the time evolution operator, the operator is squared, relating to a doubling of the length of time for which the system is evolved. For the rodeo algorithm, however, this is not necessarily the case. Each instance of time evolution in the rodeo algorithm uses a different time  $t_i$ ; the efficiency of this algorithm hinges heavily on the choice of these times as research has shown [38].

Finally, there is a class of state preparation methods called algorithmic cooling methods. Developed for NMR quantum computing, the algorithmic heat bath approach [39] attaches ancil-

lary qubits to the system, drives entropy onto them, and then decouples them from the system and lets them thermalize with the environment as way of releasing entropy. The name of the algorithmic heat bath method comes from the environment serving as a heat bath; the temperature of the environment serves as a limitation on this method. Similar in nature is the demon-like algorithmic cooling approach [3]; see Figure 2.3 for a logic diagram. This method couples ancilla qubits to the system in a way that when the qubit is measured in the  $|0\rangle$  ( $|1\rangle$ ) state, the system of interest will have its energy reduced (increased). As Fig. 2.3 shows, if a state is heated by the cooling module, it is recycled if possible (this depends on the particulars of the quantum simulator). If not, the process starts from scratch. If the cooling module successfully cools the system, then another cooling module is performed to further cool the system.

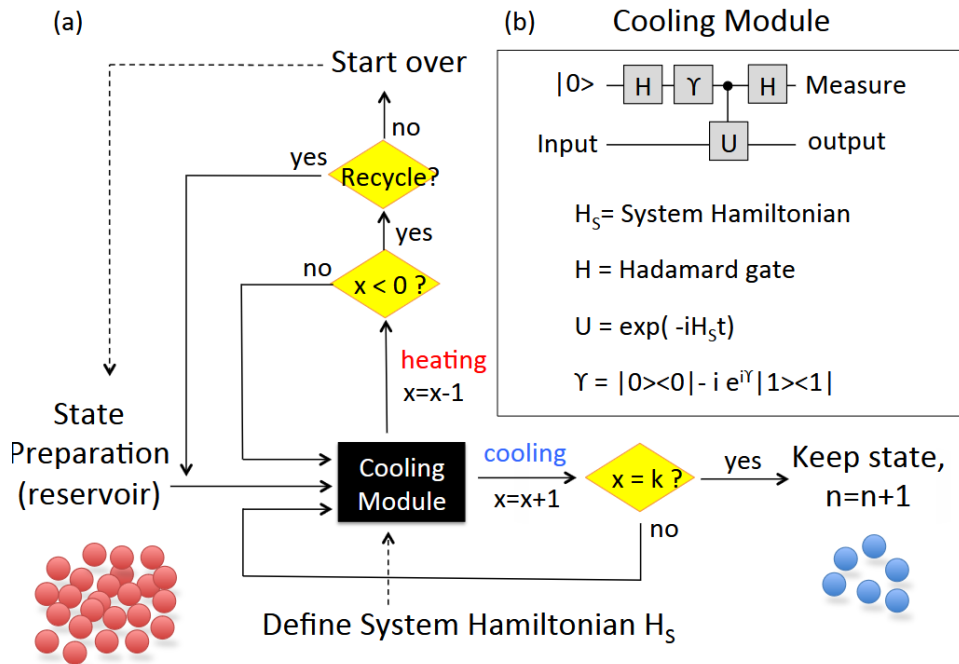


Figure 2.3: The logic diagram for the demon-like algorithmic cooling method [3]. The references to the variable  $x$  are in analogy to a 1D random walker beginning at  $x = 0$ , with a cooling result from the cooling module represented as a +1 step while a heating result is represented as a -1 step.

## 2.2 Thermal state preparation

While many topics of research require the use of ground states, others require states that can be used to study systems in thermal equilibrium. These areas of research include quark-gluon plasma (including the QGP phase of heavy-ion collisions), phase transitions in low temperature regions such as in the Ising model, and many systems of relevance to condensed matter physics, where the physics at, near, and out of thermal equilibrium are all active topics of inquiry [40].

The basic description of a thermal state comes from statistical mechanics, which defines a thermal density matrix as [40]

$$\rho_\beta = \frac{1}{\mathcal{Z}} e^{-\beta H} \quad (2.2)$$

for partition function  $\mathcal{Z} = \text{Tr}[e^{-\beta H}]$ , inverse temperature  $\beta = \frac{1}{T}$  (in a convention where  $k_B = 1$ ), and Hamiltonian  $H$ . To elucidate the connection of this quantum definition of a thermal state to classical thermodynamics, note that this thermal density matrix can be seen as one describing a state where every energy eigenstate  $|E_i\rangle$  has a probability density proportional to its Boltzmann weight  $e^{-E_i/T}$ .

Decades of research has shown that some quantum systems can thermalize, but also that others resist it (such as frustrated spin systems [41]). Furthermore, some quantum systems thermalize faster than others. It is an open question as to which quantum systems thermalize, which don't at all, and which thermalize over very long timescales, as well as why this is. The study of quantum thermalization has proposed a collection of ideas called the Eigenstate Thermalization Hypothesis (ETH) [42–44], which states that for large enough systems, a thermal expectation

value of a desired quantity can be calculated with an energy eigenstate defined by an energy that represents the average energy of the temperature associated with said thermal expectation value. While it is unclear how large a system needs to be for the ETH to hold, and for which quantities it holds, this denotes a possible avenue for calculating thermal expectation values with a relatively small ensemble of states.

Turning to the practical preparation of thermal states, it is intuitively clear that the preparation of a thermal state is more complicated than the preparation of a ground state, given the former's relative complexity. While some research requires systems to be in thermal equilibrium for much of an experiment's run, others can get around the task of producing a fully thermal state. This can be done for example if the desired results are thermal expectation values of some quantities of interest. In this case a thermal ensemble of states who are not necessarily thermal themselves can be constructed such that the resulting expectation values calculated with these states are thermal.

When it comes to calculating thermal expectation values, it should be first noted that it is possible to utilize a ground state preparation method to construct a close approximation of the system's ground state, inject some energy into the system, and finally time evolve until the system thermalizes. That being said, other methods aimed directly at thermal state preparation have been developed.

One such method employs minimally entangled typical thermal states, or METTS [45–47]. This method is a fairly straightforward way of sampling basis states such that a thermal expectation value can be calculated on the ensuing ensemble. It is a simple cyclic algorithm: begin with a basis state, produce a METTS from this basis state (this process will be detailed below), calculate the observable of a desired quantity using this METTS, then collapse the system

onto another basis state; this cycle repeats until an appropriate amount of data has been gathered.

As this is a simple algorithm, the key to its effectiveness lies in the production and properties of these METTS. These were developed in Refs. [45, 46] by considering the definition from statistical mechanics of a thermal density matrix  $\rho = \frac{1}{\mathcal{Z}} e^{-\beta H}$  and expanding the definition of an expectation value:

$$\langle A \rangle = \text{Tr}[\rho A] = \frac{1}{\mathcal{Z}} \text{Tr}[e^{-\beta H} A] \quad (2.3)$$

$$= \frac{1}{\mathcal{Z}} \text{Tr}[e^{-\beta H/2} A e^{-\beta H/2}] \quad (2.4)$$

$$= \frac{1}{\mathcal{Z}} \sum_i \langle i | e^{-\beta H/2} A e^{-\beta H/2} | i \rangle \quad (2.5)$$

$$= \frac{1}{\mathcal{Z}} \sum_i P(i) \langle \phi(i) | A | \phi(i) \rangle \quad (2.6)$$

for

$$P(i) = \langle i | e^{-\beta H} | i \rangle \quad (2.7)$$

$$|\phi(i)\rangle = P(i)^{-1/2} e^{-\beta H/2} | i \rangle \quad (2.8)$$

where  $\{|i\rangle\}$  is an orthonormal basis and the  $|\phi(i)\rangle$  states are METTS. The basis  $\{|i\rangle\}$  can in theory be any orthonormal basis but Ref. [46] argues that the best choice is the set of classical product states, or CPS, so as to minimize entanglements.

This is the utility of using METTS: one can calculate a thermal expectation value  $\langle A \rangle$  by calculating  $\langle \phi(i) | A | \phi(i) \rangle$  repeatedly, sampling these states  $|\phi(i)\rangle$  with probability distribution  $P(i)/\mathcal{Z}$ . These states get their name from their properties: they are referred to as “minimally entangled” due to the selection of the CPS basis in constructing the METTS and referred to as

“thermal” due to their definition outlined above. They are referred to as “typical” due to their ability to be sampled from the probabilities  $P(i)$  [46].

Another, more conventional method for calculating thermal states is the use of a heat bath [48]. As the next section discusses the active cooling method, which is based directly off of this heat bath method, the outline for the heat bath approach is saved for the following section.

### 2.3 The active cooling method

This section summarizes the 2023 article “Boltzmann distributions on a quantum computer via active cooling” by Carter Ball and Thomas D. Cohen, originally published in Nuclear Physics A [5]. See Appendix A for a full reprint of this article. This paper aims to expand upon the active cooling method outlined in Ref. [49], a paper concerning the calculation of thermodynamic transport coefficients for lattice gauge theories. The active cooling method leverages the concept of a heat bath while grappling with its large size requirement. Furthermore, as it was first proposed within the context of non-Abelian gauge theories, it was developed with an eye for the maintenance of gauge invariance. The method outlined in this paper uses an active cooling method to calculate thermal expectation values through the construction of a Boltzmann distribution on the ensemble level. Additionally, this method has a build-in ability to target a narrow temperature range, significantly reducing the number of states required to calculate reasonable thermal expectation values.

To begin, consider the conventional heat bath approach [48], which is typically employed for closed systems. With this approach, the system of interest represents a small subspace of the total system being used moving forward. The complementary subspace is referred to as the

heat bath or refrigerator; as the aim of this research primarily concerns cooling systems down, the term refrigerator will be used. The total heat bath system is governed by three Hamiltonians: a Hamiltonian for the system of interest,  $H^{sys}$ , one for the refrigerator,  $H^{Refrig}$ , and one that couples the two subspaces together,  $H^{coup}$ . It is important that the refrigerator Hamiltonian is chosen such that the ground state is known and easy to construct on the simulator being used. Also, the coupling Hamiltonian should be defined with the same gauge-fixing condition as  $H^{sys}$  so that coupling the refrigerator to the system of interest does not damage gauge invariance. Finally, it is required that

$$[H^{coup}, H^{sys}] \neq 0 \neq [H^{coup}, H^{Refrig}] \quad (2.9)$$

so that energy transfer between the system and the refrigerator is allowed.

Before a simulation begins, the coupling Hamiltonian is switched off while the refrigerator is initialized to its ground state and the system is initialized to any physical state that is easy to produce, which is typically one of very high energy density for non-Abelian lattice gauge theories. A key point here is that as long as the refrigerator is large enough, the details of the system's initial state do not matter. Once the simulation begins, the coupling Hamiltonian is switched on and the total system (system of interest + refrigerator) is allowed to time evolve and, to good approximation, equilibrate to a temperature close to the one associated with the refrigerator's initial energy density.

While this heat bath approach is effective, it is expensive due to the requirement of a very large refrigerator. Thus, the active cooling method of this paper proposes to use a smaller refrigerator repeatedly. This involves coupling the system to a smaller refrigerator, letting the total

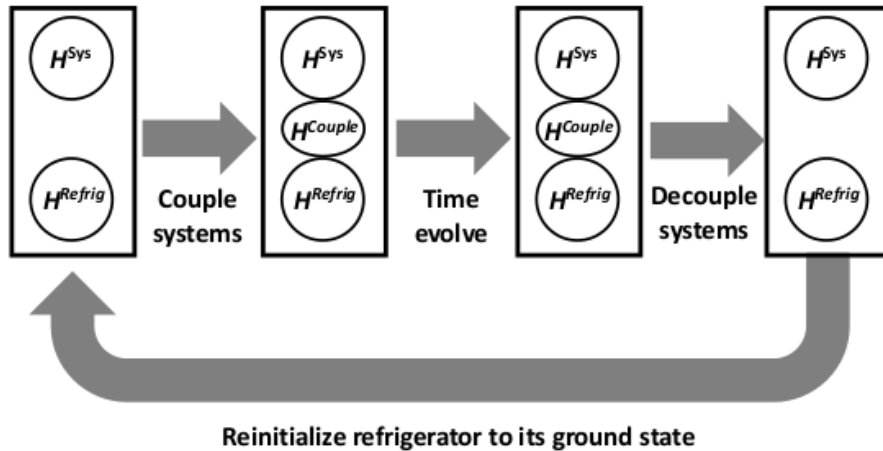


Figure 2.4: A diagram of the 4-step active cooling cycle

system time evolve for some time, and then stopping evolution and decoupling the refrigerator from the system. Finally, the refrigerator is reinitialized to its ground state. This cycle, diagrammed in Figure 2.4, is then repeated throughout the length of the simulation.

The reinitialization step is a key difference between the active cooling method and the heat bath approach. It should be noted that the ability to reinitialize the refrigerator to its ground state is assured. This is because of the DiVincenzo criteria [50] for the viability of a quantum computer, which dictate the existence of a fiducial state, or a state to which the system can be reinitialized consistently and reliably. Thus, the refrigerator is constructed such that its ground state is a fiducial state, allowing for the reinitialization step to occur reliably.

The process of the reinitialization step can consist of measuring the energy of the refrigerator and then conducting unitary operations to return the refrigerator to its ground state. This can be seen as a sort of quantum demon refrigerator, in analogy to Maxwell’s demon [51,52], as measurement requires coupling to the environment. It should be noted here that during time evolution where the system of interest is coupled to the refrigerator, entanglements develop. Thus, measuring the refrigerator will affect the system of interest as well as the refrigerator. This is why this

approach builds a thermal ensemble not a thermal state: even if the refrigerator and the system of interest fully thermalized each time they were coupled, the non-unitary measurement of the refrigerator would throw the system of interest out of thermalization. However, this measurement has no average affect on the system of interest on the ensemble level, allowing the ensemble to continue thermalizing.

Due to the cyclic nature of this algorithm, it is worth considering what one cycle should be expected to accomplish. Recall that a main reason that the refrigerator in the heat bath approach must be so large is because the system of interest and the refrigerator are only coupled once, so any desired cooling must occur in one go. This is not the case for the active cooling method; thus, one instance of coupling is required to remove only an appreciable amount of energy from the system of interest for the algorithm to be effective. This allows for the refrigerator to be smaller, perhaps on the same scale as the system of interest or even smaller.

Furthermore, there is more freedom to the active cooling method when it comes to controlling the parameters, as there is nothing requiring them to be fixed throughout the simulation. This is to say that the parameters of this method, including the coupling strength, the time of coupling, and even the structure of the refrigerator, can be manipulated throughout the course of multiple cycles. This freedom can be leveraged to optimize the method; see section [A.2.1](#) for a more detailed discussion of parameter control.

A key feature of this active cooling method is the ability to track the energy of the system, which comes from this strong control of the parameters. During the reinitialization step, the refrigerator's energy is measured before it is returned to its well-known ground state. This means that the energy removed from the total system is easily calculated as the measured energy of the refrigerator minus the refrigerator's ground state energy. This ability is crucial to this algorithm,

as it allows for the system of interest's energy to be tracked.

While this is very useful, it does not provide the full flexibility this algorithm promises. This comes from a mapping between energy and temperature. This can be achieved on classical computers by fixing the temperature of the system and calculating the average energy. This is theoretically supported due to the existence of the thermodynamic limit [53]: for large enough systems, there is an approximate equivalence between the value of a quantity at a fixed energy from a microcanonical perspective and the value of a quantity at a fixed temperature from a canonical perspective, where the average energy of the system at this temperature is the microcanonical energy. Thus, via the tracking of the system of interest's energy, we can calculate the expectation value of some desired quantity within a narrow energy range and then use the map discussed above to determine what temperature describes this expectation value in the canonical picture. This allows for fairly precise targeting of temperature ranges.

Before discussing the tests of this algorithm on a toy model, it should be noted that this algorithm would look a bit different for large lattice systems for which this algorithm is ultimately being developed than is initially discussed in the paper being summarized in this section; see section A.4 for a more in-depth discussion. For larger systems, where the refrigerator is assumed to be a spin or spin-like system with a ground state of all spins down, the probability of measuring a spin up can be used to treat the refrigerator as a thermometer, constituting an alternative method for targeting desired temperatures. Furthermore, it would be beneficial to decouple and reinitialize one spin at a time instead of doing the full refrigerator so that the cooling process is not periodically completely paused. Finally, since lattice systems tend to thermalize locally, it could be beneficial to couple spins to specific spatial regions of the lattice system and then decouple one region at a time, making sure that the pattern of decoupling regions never decouples

two adjacent regions back to back. This is so that each region is given the chance to re-thermalize locally.

With that said, it is instructive to discuss the toy model used in the paper being summarized in this section before discussing the results; see section [A.3](#). A random  $57 \times 57$  square matrix was used as the system's Hamiltonian,  $H^{sys}$ , so as to not prejudice the results to any particular structure of system. For every run of the algorithm, the Hamiltonian was generated as a real symmetric matrix where all entries were sampled from a Gaussian distribution of zero mean and a standard deviation that would lead to an energy spectrum that spans roughly 1000 energy units. For the refrigerator, we used an uncoupled spin model consisting of  $s = 4$  spins:

$$H^{Refrig} = \sum_{i=1}^s h_i \sigma_{z,i} \quad (2.10)$$

for field strengths  $h_i$  and Pauli spin matrices  $\sigma_{z,i}$ . Importantly, each  $h_i$  is made positive so that the ground state of the refrigerator is when all spins are down. Also, note that this refrigerator is smaller than the system of interest to provide evidence that the refrigerator used in an active cooling algorithm need not be much larger than the system of interest, or indeed even larger at all. Note that for this toy model, during the reinitialization step only one spin at a time is reinitialized, i.e. measured either up or down and if measured up, flipped down via  $\sigma_x$ . This is done to mimic how the algorithm is envisioned to work for larger lattice systems as discussed above.

Finally, the coupling Hamiltonian is defined as

$$H^{coup} = \sum_{i=1}^s \sigma_{x,i} \otimes K_i \quad (2.11)$$

where each spin is connected via the  $x$  Pauli matrix to a matrix  $K_i$  in the system space.  $K_i$  is a matrix of mostly zeroes except for a randomly chosen  $3 \times 3$  block which has entries chosen from a normal distribution of mean 0 and standard deviation  $\sigma$ , which defines the coupling strength. This coupling scheme was chosen to mimic the spatial region coupling scheme discussed above for large lattice systems.

When testing this algorithm with the toy model outlined above, two things were studied: the algorithm's ability to track the system's energy, and the algorithm's ability to construct a Boltzmann distribution on the ensemble level.

Concerning the energy tracking, a singular run of the algorithm would result in three energy values. One would be the energy of the system tracked via the methods outlined in the paper, the second is an energy expectation value, and the third is a conventional energy measurement. Note that the expectation value was calculated using information that a typical quantum simulation would not have access to and is being used only as a check on performance.

After running 1250 simulations, the average difference between the tracked energy and the measured energy was  $-60 \pm 269$  energy units; the average difference between the energy expectation value and the measured energy was  $11 \pm 203$  energy units. For reference, the energy span of the system's being simulated is roughly 1000 energy units. Both of these numbers are stated because smaller systems like this toy model have other sources of energy differences that are expected to be negligible for larger systems; see Appendix A for details. To put these values in context, a temperature of 218 energy units was extracted from the data (the claim that the ensemble did indeed thermalize will be proven below). Then multiple energy eigenstates were sampled from a Boltzmann distribution of the system of interest at this temperature, resulting in a standard deviation of 232 energy units. This shows that the tracked energies are in line with

expectations given the temperature of the ensemble and the probabilistic nature of measuring energy eigenstates.

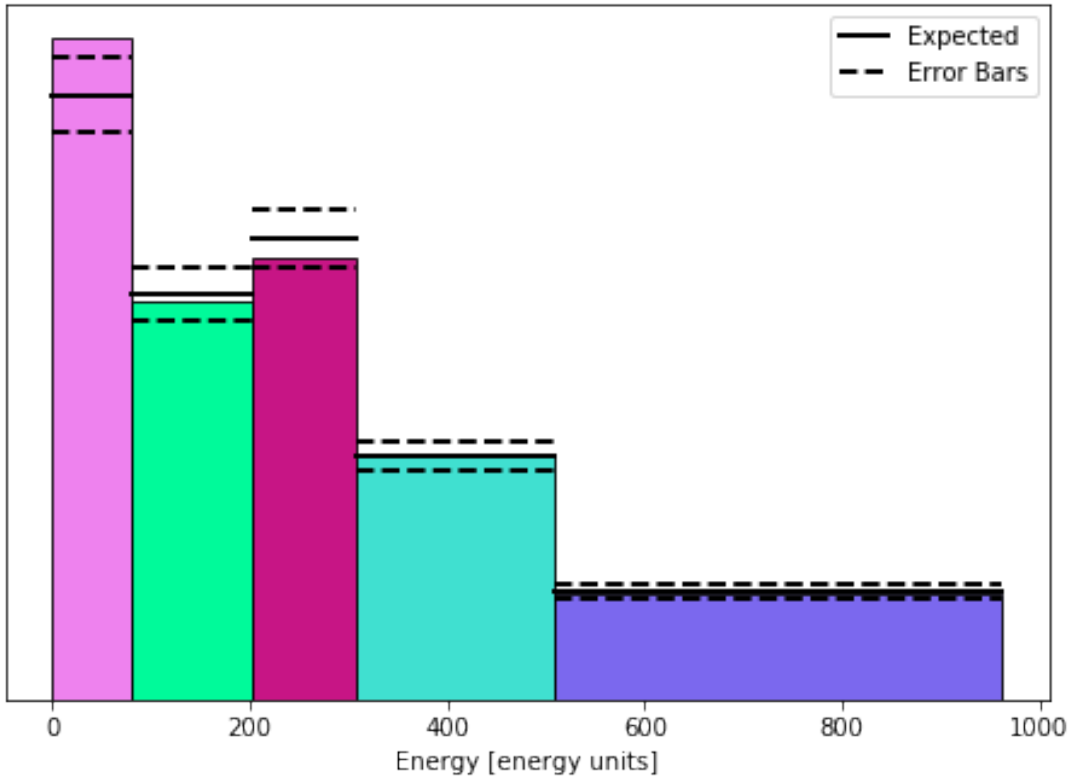


Figure 2.5: Data (colored bars) vs Expected (black lines)

Now it must be demonstrated that the ensemble did in fact thermalize; see Figure 2.5. The bins in Fig. 2.5 were designed so that a perfect Boltzmann distribution of 218 energy units would have exactly 20% in each bin; furthermore, each bin's height was chosen so they all have the same area. As this means the heights are meaningless, they are left off the y-axis.

The black lines along with their dashed error bars in Fig. 2.5 represent the ideal case of a Boltzmann distribution at a temperature of 218 energy units. The colored bars represent the results of the data. Note that four out of five colored bars land within one standard deviation of the expected values and the fifth lands within two standard deviations. This shows that the data is in fact consistent with a Boltzmann distribution, meaning that on the ensemble level the system

did in fact thermalize.

Overall, Ref. [5] outlines an active cooling method for calculating thermal expectation values via constructing a Boltzmann distribution on the ensemble level using a small quantum demon-like refrigerator in a cyclic nature to efficiently cool a system to desired energy ranges. This method could be quite useful in studying various thermal systems, such as the quark-gluon plasma stage of heavy-ion collisions, by leveraging its ability to target specific temperature ranges, allowing for far fewer states to be used in the calculation of thermal expectation values.

## Chapter 3: Suppression of gauge violations

### 3.1 Overview

The standard model of particle physics comprises of theories that are invariant under local gauge symmetries; for these theories, gauge invariance is fundamental. When it comes to the simulation of standard model theories, such as QED or QCD, care needs to be taken to assure gauge invariance. For analog simulations, the task of maintaining gauge invariance while simulating dynamical matter and gauge fields requires an untenable amount of fine-tuning. Thus, gauge-breaking processes inevitably exist within these simulations [54]. For digital simulations, in theory if the initial state is gauge-invariant and every subsequent operator is gauge-invariant, then the system remains in a gauge-invariant state throughout the course of the simulation. This however requires an error free simulation, which is not a reasonable expectation.

Thus, these gauge violations must be addressed. Left alone, the system will drift outside of the physical subspace of its Hilbert space, rendering any results garnered from the simulation unphysical and potentially virtually meaningless. Thus, a significant amount of work has gone into how to deal with gauge violations. In the next sections, I briefly review a selection of methods of suppressing gauge violations for quantum analog (Section 3.2) and digital (Section 3.3) simulation. In Section 3.4, I discuss the quantum Zeno effect, which is key to the methods of the remaining sections. Section 3.5 summarizes my work on one method that leverages the Zeno

effect to suppress gauge drift while section 3.6 summarizes my work on a similar method that leverages the properties of gauge transformations.

## 3.2 Analog approaches

There are various methods to constrain gauge violations when running quantum analog simulations. One such method ties gauge symmetry to an internal symmetry [55, 56] so that the naturally respected symmetry protects gauge symmetry as well. As an example, Ref. [55] realizes gauge fields through the use of bosonic fields that are tied to angular momentum through the Jordan-Schwinger map [57, 58]. As the symmetry of conservation of angular momentum is manifestly respected, so is the gauge symmetry tied to it.

One popular analog method, called the energy penalty method [54, 59–67], aims to separate the physical and unphysical subspaces of the system of interest in the analog system’s spectrum. Traditionally, this is done by adding a term to the analog system’s Hamiltonian that is proportional to the sum of the squared Gauss law operators of the system. Crucially, the constant of proportionality of this energy penalty term is made to be large compared to the norm of the other part of the Hamiltonian.

First note that this additional term is non-negative definite, meaning that the lowest eigenvalue is 0. Notably, due to the definition of physical states, this eigenvalue of 0 is in fact the eigenvalue of all states within the physical subspace of the system; all unphysical states would then have an eigenvalue of this additional term greater than 0. This reveals the utility of this additional term: due to the larger proportionality constant, unphysical states get a large energy penalty while the physical states don’t get any energy penalty. For large enough proportionality

constant, all unphysical states will have an energy penalty that separates them in the system's spectrum from the physical states with no energy penalty. This constructs a ground state sector of the penalty operator consisting entirely of the physical states of the system. Then, as long as the energy of the system is kept within this ground state sector, the system will remain in its physical subspace and thus be gauge-invariant.

There are multiple variations on this method, including one that uses not the square of the Gauss law operators but an energy penalty term linear in these operators [68]. This means physical states can either get a large positive or large negative energy contribution from this penalty term. This will still leave a sector the the system's energy spectrum that represents the physical subspace, with the unphysical states living above and below this sector in the spectrum.

Another variation on this method aims to get the same effect from so-called local pseudogenerators [69–71] instead of the Gauss operators, as those can be difficult to implement on the lattice. These local pseudogenerators are identical to the Gauss operators within the physical subspace of the system, but are not required to match the action of the Gauss operators for unphysical states. This allows for the same energy penalty effect with less complicated operators.

### 3.3 Digital approaches

Similar to analog simulation, much work has gone into addressing the question of maintaining gauge invariance during quantum digital simulation, including the work discussed in sections 3.5 and 3.6.

One method is to use the Gauss law constraints to eliminate redundant degrees of freedom. This is appealing for the noisy intermediate-scale quantum (NISQ) era of present as reducing de-

degrees of freedom allows for the use of fewer qubits. This method can be implemented in multiple ways; one such way eliminates gauge degrees of freedom [72–74] leaving only dynamical matter fields. The effect of the eliminated gauge fields then exists in long-range interactions comparable to spin models. Another way is to exchange fermionic degrees of freedom [75–79] for hard-core bosonic degrees of freedom.

Another method used to address gauge violation is called dynamical decoupling [80]. This method comes from quantum control theory and uses periodic driving to maintain the local gauge symmetry. Further methods use oracles to check for gauge violations [81] and filter out unphysical states.

One method of note uses classical noise to constrain gauge violation [82]. This method simulates a system of interest with the microscopic Hamiltonian

$$H_{micro}(t) = H_0 + H_1 + \sqrt{2\kappa} \sum_{x,a} \xi_x^a(t) G_x^a \quad (3.1)$$

where  $H_0$  is the gauge-invariant Hamiltonian of the system,  $H_1$  is a perturbation that drives the system out of the physical subspace of its Hilbert space,  $\kappa$  is a constant,  $\xi_x^a(t)$  is a white noise process, and  $G_x^a$  is a Gauss law operator associated with the  $a$ th generator of the gauge group at lattice site  $x$ . Note that the white noise processes are independent and satisfy  $\langle\langle \xi_x^a(t) \xi_y^b(t') \rangle\rangle = \delta_{xy} \delta_{ab} \delta(t - t')$  where the double arrow brackets denote averaging over the noise fluctuations. Defining the density operator  $\rho = \langle\langle |\psi(t)\rangle\langle\psi(t)| \rangle\rangle$  gives the following master equation that de-

scribes its evolution as well as the effective Hamiltonian:

$$\dot{\rho} = -iH_{eff}\rho + i\rho H_{eff}^\dagger + 2\kappa \sum_{x,a} G_x^a \rho G_x^a \quad (3.2)$$

$$H_{eff} = H_0 + H_1 - i\kappa \sum_{x,a} (G_x^a)^2 \quad (3.3)$$

Note that the final term in the effective Hamiltonian is proportional to the square of the Gauss law operators. This is directly related to the energy penalty method discussed in the previous section. For large enough  $\kappa$ , this penalty term in the effective Hamiltonian constrains the dynamics of the system to its physical subspace.

It should be noted that this method of classical noise can be seen as implementing a classical analog of the quantum Zeno effect, as it uses classical fluctuations, i.e. the white noise processes, to dissipate unwanted unphysical contributions while the quantum Zeno effect utilizes quantum fluctuations.

### 3.4 The quantum Zeno effect

The quantum Zeno effect has shown to be useful in the problem of maintaining gauge invariance throughout the simulation of a quantum system. Importantly, it serves as the underpinning for the rest of this chapter. Thus, it is worthwhile to discuss it before moving on.

The quantum Zeno effect was named after ancient Greek philosopher Zeno of Elea who was known for a series of paradoxes [83]. One of these is the arrow paradox, where he considers an arrow in flight. He argues that at every instant of time the arrow is motionless, so if time is comprised entirely of instants where the arrow is motionless, it cannot be in motion.

In analogy to this paradox, The quantum Zeno effect states that a quantum system's time evolution can be slowed down by frequent measurement of the system's state. To see this, consider a quantum system in an initial pure state  $|\psi_0\rangle$  at time  $t = 0$  governed by Hamiltonian  $H$ . The amplitude  $A(t)$  and probability  $p(t)$  of the system being in this initial state after a time  $t > 0$  can be defined as

$$A(t) = \langle \psi_0 | \psi_t \rangle = \langle \psi_0 | e^{-iHt} | \psi_0 \rangle \quad (3.4)$$

$$p(t) = |A(t)|^2 \quad (3.5)$$

Now, for short times, the expansion  $e^{-iHt} = 1 - itH - \frac{1}{2}t^2H^2$  is a good approximation. Then,

$$A(t) = 1 - it \langle \psi_0 | H | \psi_0 \rangle - \frac{1}{2}t^2 \langle \psi_0 | H^2 | \psi_0 \rangle \quad (3.6)$$

$$p(t) = (1 - it \langle \psi_0 | H | \psi_0 \rangle - \frac{1}{2}t^2 \langle \psi_0 | H^2 | \psi_0 \rangle)(1 + it \langle \psi_0 | H | \psi_0 \rangle - \frac{1}{2}t^2 \langle \psi_0 | H^2 | \psi_0 \rangle) \quad (3.7)$$

$$= 1 - t^2 (\langle \psi_0 | H^2 | \psi_0 \rangle - \langle \psi_0 | H | \psi_0 \rangle^2) \quad (3.8)$$

As shown above, the terms linear in  $t$  cancel out in  $p(t)$ . This is the crux of the quantum Zeno effect: for short times, the system leaves the initial state linearly in the amplitude but quadratically in the probability. Thus, measuring the system after a short time has a high probability of finding it in its initial state.

Now consider taking many measurements, each after a short time. Consider a total time evolution of  $t$  split up into  $N$  segments each of length  $t/N$ . Then using the approximation de-

rived above, the probability of measuring the system in its initial state  $N$  times, with  $\tau_Z^{-2} = \langle \psi_0 | H^2 | \psi_0 \rangle - \langle \psi_0 | H | \psi_0 \rangle^2$ , is

$$p\left(\frac{t}{N}\right)^N \approx \left(1 - \frac{t^2}{\tau_Z^2 N^2}\right)^N \approx 1 - \frac{t^2}{\tau_Z^2 N} \approx e^{-t^2/N\tau_Z^2} \quad (3.9)$$

where above along with the Taylor expansion of  $e^x$ , the Taylor expansion  $(1 - x)^a \approx 1 - ax$  was used. Taking  $N \rightarrow \infty$ , coined the limit of 'continuous observation' by Misra and Sudarshan [84], shows that the system freezes in its initial system:

$$p\left(\frac{t}{N}\right)^N \xrightarrow{N \rightarrow \infty} 1 \quad (3.10)$$

A crucial point concerning the Zeno effect is that the above demonstration focused on a measurement of a single state. This is for convenience; measurement of a subspace of the system's Hilbert space also respects the Zeno effect. This is particularly important to the discussions of the following sections.

To conclude this overview of the Zeno effect, it should be noted that the Zeno effect does not necessarily have to include the use of frequent measurements. Continuous versions of the Zeno effect have been studied [85] that couple the system to quantum fluctuations of a bath to get the desired constraint.

### 3.5 Zeno effect suppression of gauge drift

This section summarizes the 2024 article “Zeno effect suppression of gauge drift in quantum simulations” by Carter Ball and Thomas D. Cohen, originally published in Physical Review A [6]. See Appendix B for a full reprint of this article. The method of gauge drift suppression outlined in this paper is based off of the Zeno effect, proposing the use of frequent projections to keep the system within a good approximation of a physical state throughout a time evolution period.

This projection is outlined in the circuit diagram shown in Figure 3.1. It is based off of the projection used in the rodeo algorithm [2]. While the rodeo algorithm aims to build energy eigenstates, the projection of Fig. 3.1 projects onto the physical subspace. This is because of the properties of the operator  $G^2$ , which is defined as

$$G^2 = \sum_{a,x} (G_x^a)^2 \quad (3.11)$$

where  $G_x^a$  is the  $a$ th Gauss law operator at the  $x$ th lattice site. Recall that by definition, a physical state  $|\psi\rangle$  has eigenvalue 0 for all  $G_x^a$  operators; thus, every physical state is an eigenstate of  $G^2$  with eigenvalue 0. Furthermore, due to the non-negative definite definition of  $G^2$ , the only way for a state to have an eigenvalue of 0 for  $G^2$  is to have an eigenvalue of 0 for every  $G_x^a$ ; thus, the eigenspace of  $G^2$  characterized by eigenvalue 0 is every physical state and no unphysical states, i.e. the physical subspace of the system.

The projection in Fig. 3.1 couples the system with an ancillary qubit, initialized in the  $|1\rangle$

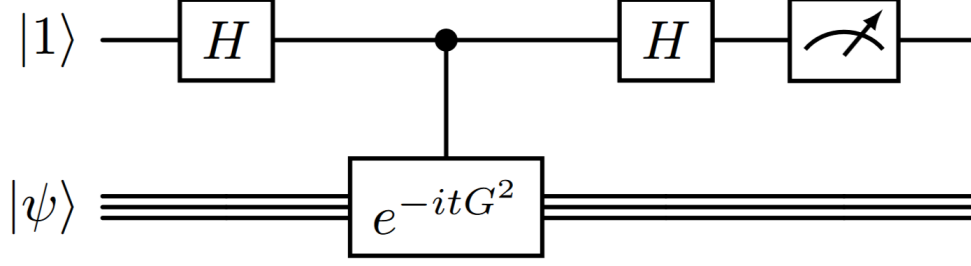


Figure 3.1: Circuit diagram of the Zeno effect projection

state. Then the three gates perform the following transformation:

$$\begin{bmatrix} 0 \\ |\psi\rangle \end{bmatrix} \rightarrow \begin{bmatrix} \frac{1}{2}(I - e^{-itG^2})|\psi\rangle \\ \frac{1}{2}(I + e^{-itG^2})|\psi\rangle \end{bmatrix} \quad (3.12)$$

For physical states,  $e^{-itG^2}$  is a trivial phase; thus, equation 3.12 shows that all physical states are contained in the  $\frac{1}{2}(I + e^{-itG^2})|\psi\rangle$  component, as  $\frac{1}{2}(I - e^{-itG^2})|\psi_{phys}\rangle = \frac{1}{2}(|\psi_{phys}\rangle - |\psi_{phys}\rangle) = 0$ .

It is because of this analysis that a successful projection is one where the ancillary qubit is measured in the  $|1\rangle$  state at the end of the projection, and when it is measured in the  $|0\rangle$  state it is considered a failure. While a successful projection does not affect the physical amplitudes of the system's state, it does suppress the unphysical amplitudes. This can be easily seen since unphysical states will have a non-trivial phase  $e^{-itG^2}$ , so the amplitude is split between the  $|0\rangle$  and  $|1\rangle$  states. Specifically, an unphysical eigenstate of  $G^2$  with eigenvalue  $\lambda$  is suppressed by a factor  $\cos^2(\frac{1}{2}t\lambda)$  for a successful projection (and  $\sin^2(\frac{1}{2}t\lambda)$  for a failed projection).

Because of the sinusoidal nature of this suppression factor, the method of Ref. [6] uses multiple qubits per projection with a variety of different times to get good suppression across the board. Note, though, that this could instead be one qubit reused repeatedly provided the simulation at hand allows for mid-circuit measurements. Regardless, with good suppression, the Zeno

effect comes into play: making frequent projections will not allow the system to evolve away from its physical subspace, therefore remaining (to a good approximation) physical throughout its simulation.

The paper in Appendix B discusses the costs of frequent projections and the risk of a failed projection. Due to these considerations, another method is discussed in this paper that is folded into the Zeno effect projection method outlined above. This technique was proposed by Lamm, Lawrence, and Yamauchi in 2020 [86] to suppress gauge drift by conducting a random gauge transformation on the system after every Trotterized time step. While these gauge transformations leave physical amplitudes of the state's system untouched, it scrambles unphysical amplitudes, leading to an overall suppression. While this technique does not fully stop gauge drift, it does slow it down. This is helpful to add to the Zeno effect projection method because slower gauge drift means less frequent projections are required to maintain the same level of effectiveness.

With the method outlined, the paper that this section is summarizing demonstrates this method with a toy model, namely a pure  $\mathbb{Z}_2$  gauge theory on a  $2 \times 2$  lattice with periodic boundary conditions. See Appendix B.3 for an in-depth outline of the toy model. Note that due to the discrete nature of  $\mathbb{Z}_2$ , for the toy model simulation,  $G^2$  is replaced with  $G_{tot}$  (defined in equation B.20) and a physical state has an eigenvalue of 4, i.e.  $G_{tot} |\psi_{phys}\rangle = 4 |\psi_{phys}\rangle$ .

As a first test of this method, three different simulations were compared: a control simulation without any gauge drift suppression methods, a simulation with only the technique of performing a gauge transformation after every time step, and a simulation with a gauge transformation and a projection after every time step. The results of this test can be seen in Figure 3.2, which shows that the gauge transformation technique does a good job at slowing down the gauge drift but the projections keep the system roughly within the physical system (denoted in

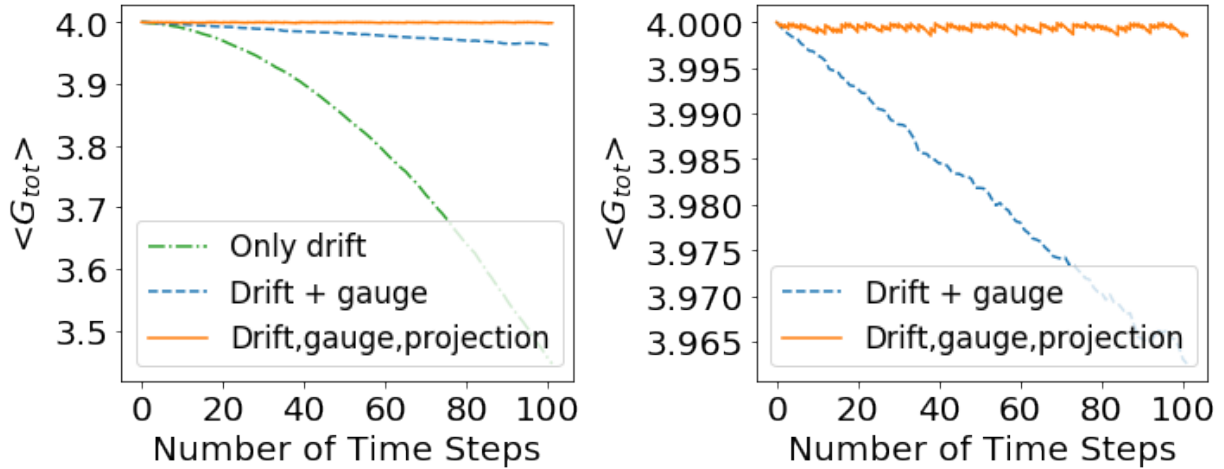


Figure 3.2: Comparing gauge drift suppression methods, including no suppression methods (dash-dot line), a random gauge transformation after each time step (dashed line), and a random gauge transformation followed by a projection after each time step (solid line). The right plot is the same as the left plot without the dash-dot plot so as to zoom in on the behavior of the other two.

this figure by the expectation value of 4).

It is useful to investigate the effect of including the gauge transformation technique on the required frequency of projections. Thus, multiple simulations were run, some with and some without gauge transformations performed after every time step, with a varying number of time steps between projections. Multiple simulations of each suppression method were run and then averaged together. These results are shown in Figure 3.3. Clearly, the inclusion of the gauge transformation technique makes a significant difference. Notably, performing a projection after every time step without gauge transformations shows to be comparable to performing a projection after every other time step with gauge transformations, and performing a projection after every other time step without gauge transformations shows to be comparable to performing a projection after every 5 time steps with gauge transformations.

See the reprint of this paper in Appendix B for a detailed discussion of implementation, focusing firstly on the problem of time selection when it comes to constructing effective projections

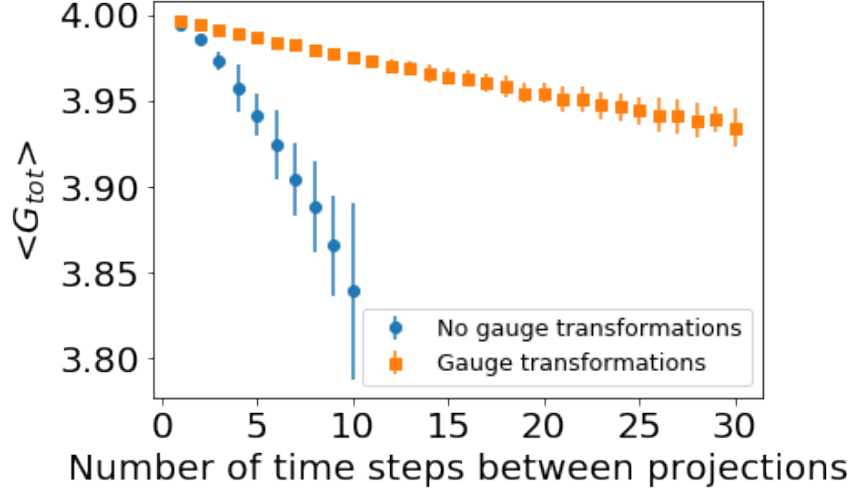


Figure 3.3: Comparing performance of regimes with no gauge transformations and various frequencies of projections (circle markers) with regimes with gauge transformations and various frequencies of projections (square markers)

and secondly on the actual implementation of the gauge transformations discussed above.

Overall, Ref. [6] proposes a novel method for gauge drift suppression that leverages the Zeno effect and a previously proposed gauge transformation technique to allow for a simulation of a gauge theory to time evolve while remaining, to a good approximation, within the system’s physical subspace.

### 3.6 Suppression of gauge drift using gauge transformations

This section summarizes the 2024 article “Suppressing gauge drift in quantum simulations with gauge transformations” by Carter Ball [7]. See Appendix C for a full reprint of this article. The research of this paper is build on the back of the research demonstrated in the paper summarized in the previous section.

This paper aims to use gauge transformations to suppress gauge drift in various ways by acting on unphysical amplitudes while leaving physical amplitudes untouched, per the defini-

tion of physical states. One way to do this was outlined in the section above, i.e. the technique conducting a random gauge transformation after every Trotterized time step proposed by Lamm, Lawrence, and Yamauchi [86]. Another way to use gauge transformations for gauge drift suppression detailed in the paper is an alternate version of the Zeno effect projection method outlined in the previous section. The difference between the version proposed in the paper of Appendix B and the version proposed in the paper of Appendix C can be straightforwardly seen by comparing the circuit diagram of the previous method, Fig. 3.1, and the circuit diagram of this new method, shown in Figure 3.4. Notably, Fig. 3.1 has the operator  $e^{-itG^2}$  in the controlled gate while Fig. 3.4 has the operator  $g$ , representing a random gauge transformation on the system.

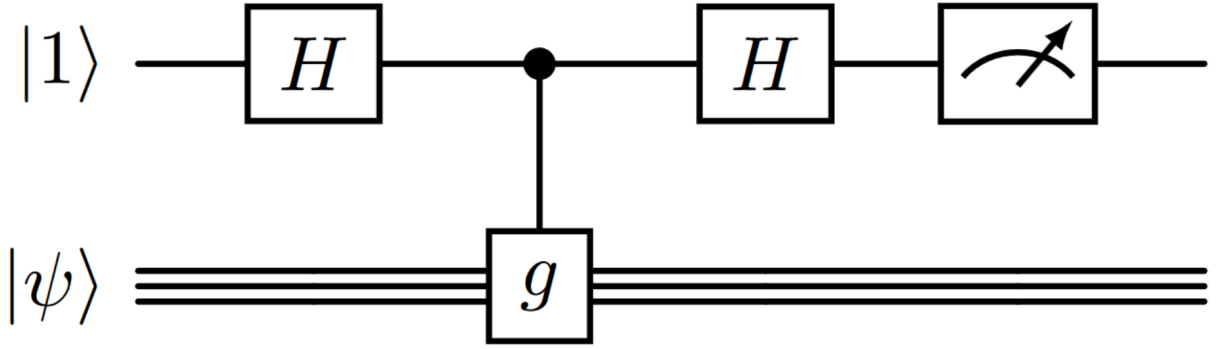


Figure 3.4: Circuit diagram of the projection used to suppress gauge drift

Thus, the circuit diagram of Fig. 3.4 performs the following transformation before the final measurement step:

$$\begin{bmatrix} 0 \\ |\psi\rangle \end{bmatrix} \rightarrow \begin{bmatrix} \frac{1}{2}(I - g) |\psi\rangle \\ \frac{1}{2}(I + g) |\psi\rangle \end{bmatrix} \quad (3.13)$$

At this point it is instructive to decompose the state  $|\psi\rangle$  into its physical,  $|\psi_p\rangle$ , and unphys-

ical,  $|\omega\rangle$ , components to see the utility of this transformation. Writing  $|\psi\rangle = |\psi_p\rangle + \epsilon|\omega\rangle$ , we see that, because  $g|\psi_p\rangle = |\psi_p\rangle$ ,

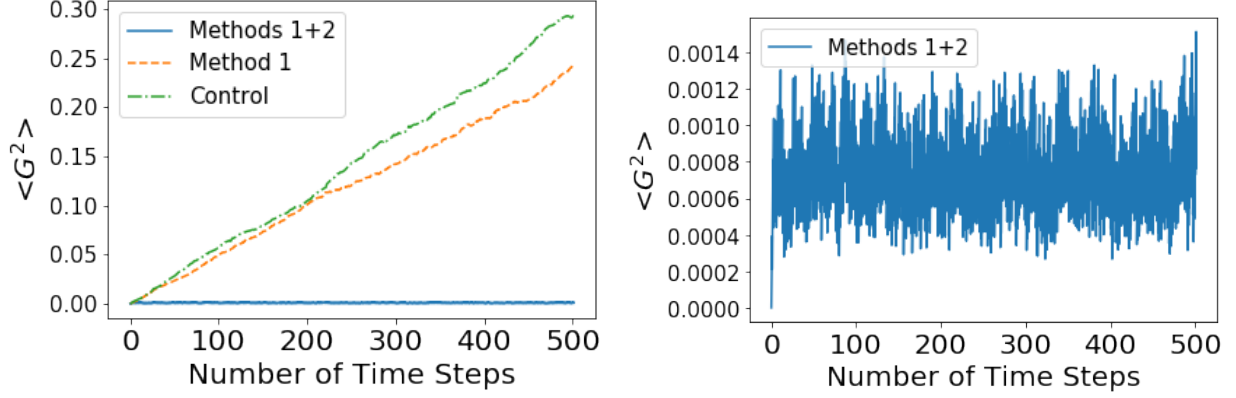
$$\begin{bmatrix} \frac{1}{2}(I - g)|\psi\rangle \\ \frac{1}{2}(I + g)|\psi\rangle \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\epsilon(I - g)|\omega\rangle \\ |\psi_p\rangle + \frac{1}{2}\epsilon(I + g)|\omega\rangle \end{bmatrix} \quad (3.14)$$

Note here that all the physical amplitude of  $|\psi\rangle$  is in the  $|1\rangle$  part of the state, while the  $|0\rangle$  part is fully comprised of unphysical amplitudes. Thus, like in the previous section, measuring the ancillary qubit in the  $|1\rangle$  state is considered a successful projection while measuring it in the  $|0\rangle$  state is considered a failed projection.

With the main method of Ref. [6] outlined above, this summary will now focus on the results. The paper tested its method on a toy model: a pure 1D SU(2) system cutoff at  $j_{max} = 1/2$ . The toy model comprised of 4 sites and was simulated with periodic boundary conditions. See Appendix C for more detail on this toy model.

The first test run on this toy model was simply to run simulations with the method compared to a control simulation without any gauge drift suppression techniques. The results of this test can be seen in Figure 3.5. This figure shows that the method of this paper does in fact successfully keep the system within a close approximation of a physical state for the duration of simulation.

The second test on this method aimed to determine the average suppression rate for each projection. This was done by initializing the system in a randomly selected physical state 50 times. With each run, the system was simulated for 101 time steps, each followed by a projection. The expectation value  $\langle G^2 \rangle$  was calculated before and after each of these projections, after which



(a) All three simulations

(b) Just the simulation with both methods used

Figure 3.5: Results from three simulations. The control simulation (green dashed-dot line) ran without any gauge drift mitigation techniques. The second (orange dashed line) simulated gauge drift with the method of gauge transformations after every time step (method 1). The third (blue solid line) simulated gauge drift with a gauge transformation (method 1) and a projection of the kind in Fig. 3.4 (method 2) after every time step.

the ratio  $\langle G^2 \rangle_{after} / \langle G^2 \rangle_{before}$  was calculated. The average of these ratios was determined to be  $0.512 \pm 0.063$ . Thus, the average projection reduces the unphysical amplitudes of the system by about half. As the paper mentions, this can be helpful in the calibration of this method; see Appendix C for details.

The final test of this method looked into a question of implementation. See Appendix C for the details of this test as well as a discussion on the implementation of this method. This test specifically studied the possibility of reusing gauge transformations throughout a simulation to cut down on the costs of generating random gauge transformations. This test showed that for this small toy model, using only two gauge transformations throughout the full simulation is quite viable, with little to no trade-off with effectiveness. While intuitively it is likely that for larger more complicated systems it will require more than 2 gauge transformations to roughly match the performance of a random gauge transformation every time one is called for, this result could mean that the requisite number of gauge transformations for a larger system is not unfeasibly

large; certainly, a study of the scaling of this value must be conducted to make any concrete claims on this point.

Overall, Ref. [6] proposes a new method of gauge drift suppression, built upon the method of the paper summarized in the previous section, based on the idea of utilizing gauge transformations and their properties vis-a-vis physical and unphysical states to reduce unphysical amplitudes in the system's state.

## Chapter 4: Summary

This dissertation concerns quantum algorithms for quantum computers, specifically those with an eye towards problems in quantum chromodynamics. Two kinds of quantum algorithms are featured: state preparation algorithms in chapter 2 and algorithms for the suppression of gauge violation in chapter 3.

Chapter 2 introduces the problem of initial state preparation for quantum simulations. Ground state preparation and thermal state preparation were both discussed. A review of existing methods was provided, including discussions of adiabatic state preparation, quantum phase estimation, projective cooling, the rodeo algorithm, algorithmic cooling methods, minimally entangled typical thermal states (METTS), and the heat bath approach.

In section 2.3, a paper detailing the active cooling method for the calculation of thermal expectation values was summarized, as this paper serves as a major part of the novel research culminating in this dissertation. This active cooling method was presented as being developed from the concept of the more conventional heat bath approach, with key improvements including a much smaller heat bath, or refrigerator, and good control over the resulting temperature. The theory supporting this method was discussed along with the practical implementation of the algorithm. A toy model was used to test the algorithm and demonstrate its effectiveness as well as the key properties that make this algorithm a useful addition to the literature of thermal state

preparation.

Following this, chapter 3 turns to the question of the maintenance of gauge invariance during the quantum simulation of lattice gauge theories. It begins with an overview of quantum simulation, gauge invariance, and gauge drift. This was followed by a review of existing gauge invariance maintenance schema from both analog (sec. 3.2) and digital (sec. 3.3) approaches. These include tying gauge symmetry to an internal symmetry, the energy penalty method, eliminating redundant degrees of freedom by leveraging the Gauss laws, dynamical decoupling, and the use of classical noise.

Section 3.4 then reviews the quantum Zeno effect, as it serves as a theoretical foundation of the research in the ensuing sections. Specifically, sections 3.5 and 3.6 each summarize a paper that lays out a method of suppressing gauge drift; these papers also serve as a major portion of the novel research culminating in this dissertation. Both of the methods laid out in these papers take inspiration from the rodeo algorithm, coupling an altered quantum gate circuit with the quantum Zeno effect as well as a method of decelerating coherent gauge drift developed by Lamm, Lawrence, and Yamauchi in Ref. [86] to construct an algorithm to suppress gauge drift.

The work summarized in section 3.5 aims to take the general idea of the rodeo algorithm, which uses an ancilla qubit to cause destructive interference for unwanted amplitudes, and apply it to the concept of gauge violation. This involves using a new operator  $G^2$  (defined in equation (3.11)) whose ground subspace consists of every physical state of the system of interest while all unphysical states are excited states. Utilizing the rodeo mechanism with this new operator  $G^2$  then allows for the ground subspace to be targeted, effectively allowing for the unphysical amplitudes to be suppressed via destructive interference all while the physical amplitudes are left untouched. This section then concludes by demonstrating the effectiveness of this method via a

toy model: a pure 2D  $\mathbb{Z}_2$  gauge theory.

The work summarized in section 3.6 iterates on this idea further, by replacing the exponential operators  $e^{-itH}$  (in the case of the rodeo algorithm) and  $e^{-itG^2}$  (in the case of sec. 3.5) with a gauge transformation on the system. This method provides the same destructive interference for unwanted unphysical amplitudes but with lower costs, such as avoiding the complex construction of the operator  $G^2$  and its exponential. This section concludes by demonstrating the method on a toy model: a pure 1D SU(2) system with a  $j_{max} = 1/2$  cutoff.

Appendix A: Boltzmann distributions on a quantum computer  
via active cooling

*Carter Ball and Thomas D. Cohen*

**NOTA BENE: What follows is an exact reprint of the 2023 article “Boltzmann distributions on a quantum computer via active cooling” by Carter Ball and Thomas D. Cohen, originally published in Nuclear Physics A [5].**

Abstract

Quantum computing raises the possibility of solving a variety of problems in physics that are presently intractable. A number of such problems involves the physics of systems in or near thermal equilibrium. There are two main ways to compute thermal expectation values on a quantum computer: construct a thermal state that reproduces thermal expectation values, or sample various energy eigenstates from a Boltzmann distribution of a given temperature. In this paper we address the second approach and propose an algorithm that uses active cooling to produce the distribution. While this algorithm is quite general and applicable to a wide variety of systems, it was developed with the specific intention of simulating thermal configurations of non-Abelian gauge theories such as QCD, which would allow the study of quark-gluon plasma created in heavy-ion collisions.

## A.1 Introduction

Quantum computers promise to open the door to many calculations that in practice can be performed using computers that behave classically [8]. Over the years, significant research has gone into the development of physical quantum computers [72, 87], the theory behind them [88, 89], and the design of quantum algorithms [90]. There are many open questions in physics for which it is hoped that once sufficiently powerful quantum computers are developed they may provide important insights. One significant class of such problems involve non-perturbative non-equilibrium dynamics [91].

This paper describes an algorithm that uses the method of active cooling to create a Boltzmann distribution. Furthermore, we demonstrate that this algorithm can predict the energy of the system of interest and so, given a mapping between energy and temperature, the algorithm can be used to target specifically chosen temperature ranges. The proposed algorithm can thus be used to sample energy eigenstates from a Boltzmann distribution of a chosen temperature range.

While the algorithm we develop in this paper is rather general, our motivation is at least in part to develop an approach suitable for applications to thermal properties of non-Abelian gauge theories, such as quantum chromodynamics (QCD), as would be needed for studies of heavy-ion reactions. As with most quantum computing approaches to quantum field theories, the formalism considered here is in the context of Hamiltonian dynamics, rather than a Lagrangian/action based formalism commonly used in Euclidean space-time. For the purposes of the discussion here, it does not matter whether one wishes to think of the dynamics as being realized on a quantum simulator where interactions are constructed to mimic the dynamics of interest or via a gate-based general quantum computer. In the latter case the Hamiltonian dynamics are realized via

Trotterization.

The basic approach discussed in this paper should be useful in situations in which it is comparatively easy to explicitly construct an initial state of a system that has a much higher energy density than the thermal system one wishes to study. In the context of the thermal properties of non-Abelian gauge theories, which serve as the key motivation of this paper, gauge invariance plays a central role. Thus, in describing the approach, we will emphasize places in which the preservation of gauge invariance is central and explain how gauge invariance is maintained. For applications to theories other than gauge theories this can be ignored; the rest of the algorithm goes through unchanged.

The implementation of a gauge theory in a Hamiltonian formalism is somewhat subtle compared to non-gauge theories. As one goes from the Lagrangian to Hamiltonian formulations, one faces the problem that in the Lagrangian formulation, the temporal component of the gauge field has no time-derivatives, meaning that it has no associated canonical momentum. Thus, one needs to choose a gauge-fixing condition to write a Hamiltonian. Perhaps the most natural of these for the purpose of quantum computing is the temporal gauge of  $A_0 = 0$ . Moreover, even after the gauge fixing is fixed, there is a residual gauge freedom that respects the gauge fixing condition.

Note that the Euler-Lagrange equation obtained from the Lagrangian in this temporal gauge is not strictly an equation of motion as it does not involve time; rather, it yields the color version of the Gauss law and serves as an equation of constraint. States that are physical must satisfy the Gauss-law constraint. Thus the full Hilbert space of the theory is larger than the physical space. The Gauss-law constraint operator acts as a generator for residual gauge transformations which implies that all physical states of the theory are invariant under residual gauge transformations

[92].

When it comes to the quantum simulation of gauge theories, it is important that any algorithm used has as an input that is a physical state and in its time evolution keeps the system in the physical space. This is nontrivial, as common digitization schemes [93, 94] can create non-physical states. One way to assure physical states is to turn to gauge invariance, as physical states are exactly those states that are invariant under residual gauge transformations. Fortunately, time-evolution commutes with gauge transformations, even under a Suzuki-Trotter approximation [14, 68, 95]. Thus, any state preparation scheme that begins with a gauge-invariant state and proceeds through time evolution will maintain a physical state, and thus respect gauge symmetry (up to noise, whose effects in pushing the system out of the physical subspace can be mitigated [86]).

As mentioned above, the motivation for this paper is specifically the study of various aspects of heavy-ion collisions, which face severe limitations in the world of classical computational methods. Thus, it is useful to ask what would be needed to obtain information crucial to the study of heavy-ion collisions if a sufficiently powerful quantum computer were available. The standard picture for heavy-ion collisions [96, 97] comes in three stages, each with their own theoretical description [98]. The first stage is the initial scattering of two high-energy nuclei, with collisions taking place at the partonic level. After this, a quark-gluon plasma (QGP) is formed, expands and cools, constituting the second stage. In the final stage, as the QGP cools sufficiently, the quarks and gluons hadronize. Notably, hydrodynamics has shown to be quite effective at modeling the QGP stage of the collisions [99, 100]. Data from LHC and RHIC [101] can help model hydrodynamics for QGP, but there are many free parameters and uncontrolled systematics in the phenomenological models used to estimate them from the data; this adds uncertainty to

analyses [102, 103].

Of particular interest to the hydrodynamic study of heavy-ion collisions are the transport coefficients [104–109]: diffusivity, conductivity, and viscosity. Previous research [49] has proposed quantum algorithms to determine these transport coefficients, including working out the requisite Hamiltonian lattice operators. To evaluate the expectation values of these operators, two things are needed: a way to time-evolve these operators under the physical Hamiltonian, and states for the operators to act on. A Trotterized time-evolution operator that respects gauge symmetries has been constructed in [14].

For transport coefficients, we want states to be chosen from a distribution corresponding to a system in thermal equilibrium. The aim of this paper is to propose an algorithm that can take a state that is straightforward to write on a quantum computer and show how to perform operations on it that allow for the sampling of energy eigenstates with approximately Boltzmann weights. In this way, thermal expectation values can then be estimated.

One concern about thermal state preparation is the time-scale. There exist certain systems for which one expects thermalization to take a long time. An example: certain frustrated spin systems [41] at low temperatures are known to require state preparation times to be exponential in the volume. Fortunately, in the context of the principal concern here—heavy ion reactions in their QGP phase—phenomenology strongly suggests that rapid equilibration occurs after the initial hard interactions [110, 111]. This equilibration is believed to occur on a time scale comparable to the QCD characteristic scale [112], and it is quite plausible that this quick equilibration is a general feature of strongly-coupled gauge theories. With this in mind, we expect developing a non-thermal state and letting it time-evolve will achieve approximate thermalization on approximately the natural time-scale of the theory. While this is far from a formal proof, this

phenomenological argument makes it reasonable to assume that polynomial-time thermalization schemes are possible. Thus, we can proceed to the practical task of achieving such a scheme.

### A.1.1 State preparation methods

There are many state preparation methods. It is believed that often the most arduous task in quantum simulation is this exact problem of preparing an initial state [15–22]; thus, no “one size fits all” scheme currently exists. Since quantum computing is naturally implemented via Hamiltonians, the goal of many state preparation methods is to prepare the system to be in the ground state of a chosen Hamiltonian.

Perhaps the most conceptually straightforward approach is a heat bath algorithm [48]. It aims to construct a large heat bath in thermal equilibrium (at a finite temperature) and then couple the system to this heat bath weakly and allow energy to transfer until the system itself has equilibrated. The goal of this method is to create a state whose expectation value of a given observable reproduces the thermal expectation value of the observable at a certain temperature. This approach suffers from a large auxiliary system and long interaction times. The algorithm proposed in this paper aims to improve upon this basic approach.

A classic scheme of ground state preparation is adiabatic state preparation [23–25], which aims to prepare the ground state of a desired Hamiltonian. This method begins with a Hamiltonian whose ground state is known and adiabatically time evolves the initial Hamiltonian into the Hamiltonian of interest. The adiabatic theorem [26] tells us that this would obtain the desired ground state. It should be noted that when working with a gauge theory, care must be taken that gauge invariance is maintained in any approach. For field theories, it is natural to use lattice ver-

sions of the theory. One challenge faced by lattice gauge theories in a Hamiltonian formulation is the one alluded to earlier: that the Hilbert space of the theory is much larger than the physical Hilbert space [92]. States in the physical space are constrained to satisfy the color Gauss law, which in a lattice formulation means that the net color electric flux flowing out of a lattice site must be equal to any color electric charge on the site. Equivalently, the state must be invariant under the residual gauge transformations permitted once a gauge-fixing condition that relates the temporal component of the gluon field (which has no conjugate momenta) to the spatial components is specified. In the adiabatic approach, if one can prepare a state that is the ground state of the physical Hilbert space of some limit of the theory and then adiabatically alter the theory while preserving gauge invariance, one is guaranteed that the system will remain in the physical Hilbert space of the system. For such gauge theories, an obvious starting state is the strong-coupling limit of the lattice gauge theory. The virtue of this limit is that an explicit prescription exists for the ground state in this limit in the physical space [113]. A concern is that noise or other sources of error may push the system out of the physical space; fortunately, as mentioned above, there is a technique that helps ameliorate this problem [86]. However, despite this technique, the long time evolution required for adiabatic state preparation is a major drawback: quantum computers must constantly fight decoherence [27], and long times imply large opportunities for decoherence.

Another popular method of state preparation is via quantum phase estimation [28–30]. This method starts with an approximate eigenstate and a unitary operator (often the time evolution operator). Then, through use of additional qubits (i.e. qubits not encoding the initial eigenstate) and operations of the unitary operator, the eigenvalue and true eigenvector are found to polynomial accuracy. This algorithm, however, requires that the time evolution be fully coherent. Variational methods [31–33] of phase estimation have been developed to replace long time evolutions with a

number of smaller time evolutions. However, these methods require an initial state with significant overlap to reconstruct the desired eigenstate, which means additional steps must be taken to get a good initial state which in turn can be used to construct the ground state. Moreover, finding an initial state with substantial overlap with the true ground state may well be exponentially hard as the size of the system increases.

A recently developed approach to state preparation is projective cooling [34, 35]. This method is similar in spirit to evaporative cooling. It acts to drive excitations out of the region of interest and into a larger complementary system. Then the wavefunction remaining in the region of interest is measured. A main drawback here is that the required complementary system must be significantly larger than the system of interest. A variation on this method is the Rodeo algorithm [2], which uses ancilla qubit measurements to shed excitations, thus driving the initial state to a desired eigenstate. These algorithms depend on the overlap of the initial state and the desired eigenstate, making initial state choice quite important for efficiency. Again, a major drawback is that it becomes exponentially hard to find an initial state with substantial overlap with the true ground state as the size of the system increases.

Resonance transitions [114], provide an alternative approach to state preparation. In this approach an ancilla qubit is coupled to the physical system. Then a resonance transition between the ancilla qubit and the physical system's ground state is driven and measurements on the ancilla qubit are performed to drive the physical system towards its ground state. This algorithm requires knowledge of the physical system's ground state energy, which poses a significant problem for lattice gauge theories.

Finally, there are algorithmic cooling approaches. Heat bath algorithmic cooling [39] pushes entropy onto a number of qubits that are then replaced by qubits that thermalize quickly

with the environment; here, the environment is acting as a heat bath. This scheme was developed in the world of NMR quantum computing and has a fundamental limit of the temperature of the environment. Furthermore, it is not a universal method when it comes to cooling quantum many-body systems. Demon-like algorithmic cooling [3] builds on this approach by using ancilla qubits to entangle with the system in such a way that measuring the qubit down (up) results in a system state of lower (higher) energy. While this is very similar to our approach in that energy is removed from the system via measurement, our algorithm has an important feature wherein the energy/temperature can be controlled.

The proceeding approaches are aimed at finding the ground state. For thermal expectation values other state preparation methods are required.

One approach is to use minimally entangled typical thermal states, or METTS [45–47]. This approach calculates thermal expectation values by starting with a basis state, evolving it in imaginary time, and then calculating the desired observables. Then this state is collapsed via measurement to a new basis state and the process is repeated. For reference, the state created after the imaginary time evolution is called a METTS. This approach must contend with correlations between samples; it also requires a large number of measurements to get a reasonable thermal average.

In this paper we develop another algorithm for the calculation of thermal expectation values. It is most closely related to ref. ([3]) in that it uses a “Maxwell demon” type approach. Our approach has a number of virtues: It is designed to produce a thermal ensemble, and it can be implemented in a way that allows one to control the desired temperature. It is also designed to be implementable on gauge theories without damaging gauge invariance. Moreover, this scheme is designed to have favorable scaling behavior when implemented on lattice theories with local

interactions; the time taken to reach equilibrium at some temperature should be independent of the volume, while the total number of ancillary qubits needed grows linearly with the size of the system. Finally, this scheme might turn out to be useful in finding ground states, either via directly driving the temperature to zero or as a means to obtain sufficiently low temperatures so that states from the ensemble have a sufficiently large overlap with the ground state so that methods such as Rodeo algorithm [2] can be used efficiently to obtain the ground state.

This paper is organized as follows: Section II details our algorithm to construct Boltzmann distributions based on a method of active cooling. Furthermore, we argue that the energy of the system of interest can be predicted so that, given a mapping between energy and temperature, the Boltzmann distribution that is constructed will come from a chosen temperature range. This algorithm has been developed via the initial inspiration of ref. [49]. Section III describes a “toy” model that we used to illustrate this algorithm. Section IV discusses how to generalize our algorithm developed for the toy model to larger systems with a particular emphasis on large lattice models. We conclude in Section V with a discussion of the required future work.

## A.2 The Active Cooling Scheme

The approach in this paper is an extension of the method outlined in ref. [49]. That scheme was designed to exploit the basic thermodynamic notion of a heat bath [53], but to do so in a manner that does not require an overly large number of qubits. In a closed system with a heat bath approach, one uses a total system that is much larger than the subspace representing the system of interest. One would denote the system complementary to the system of interest as a heat bath. We assume that the subspace representing the physical system of interest is governed

by Hamiltonian  $H^{sys}$ , and the heat bath is governed by a Hamiltonian  $H^{HB}$  (which for gauge theories is defined according to a gauge-fixed scheme), which are initially decoupled. A key feature of  $H^{HB}$  is that it is simple enough that an explicit form of its ground state is known and can be implemented easily on a quantum computer. The heat bath is initialized to its ground state and the system subspace is initialized in a physical state of higher energy than of interest—as it would be if it were in an easily produced physical state for a non-Abelian gauge theory.

The key point is that the specifics of the initial physical state of the system of interest should not matter provided that the heat bath is sufficiently large. Accordingly, we will simply use a state that is relatively simple to write on a quantum computer. For physical states in the system of interest, this will require a state with very high energy density if one is studying a non-Abelian gauge theory.

The subsystem for the system of interest and the heat bath would then be coupled via a Hamiltonian  $H^{coup}$ .  $H^{coup}$  is derived from a Lagrangian that is gauge-invariant in the system of interest and is defined with the same gauge-fixing condition as  $H^{sys}$ , and thus it is invariant with respect to the same residual gauge transformations in the system subspace. Initially this coupling Hamiltonian is switched off and the system of interest is set in a known physical state, specifically one that satisfies the Gauss law. Since  $H^{coup}$  is gauge invariant, Hamiltonian dynamics will ensure that the system will continue to obey the Gauss law and hence remain physical.

Importantly, we require

$$[H^{coup}, H^{sys}] \neq 0 \neq [H^{coup}, H^{HB}] \quad (\text{A.1})$$

This condition assures that energy will be allowed to transfer from the system subspace to the

heat bath when the coupling is turned on. The next step is to switch on the coupling and allow the dynamics to run. To good approximation, the full system will eventually equilibrate with a temperature corresponding very closely to the one associated with the initial energy density of the heat bath.

The major defect of this heat bath approach is that the heat bath must be quite large for it to be effective. The logic of the heat bath is that it can thermalize with the system of interest without noticeably changing its own temperature.

Before describing alternatives to the large heat bath, it is useful to note that there are many challenges facing practical quantum computing for gauge theories. The optimal method of calculating thermal expectation values may well depend on what is the most significant source of limitation for the problem under study; different approaches face different limitations. If the principle source of limitation is the number of well-connected qubits in the machine, then heat bath methods become problematic. The active cooling approach described below is a sensible way of dealing with this problem, but this may involve a trade off: it is possible that it will take longer to run than a single heat bath and that could prove problematic if the principal limitation is coherence time.

With this consideration in mind, we continue examining our principal motivation in ref. [49]. To understand the motivation for the scheme, one might first choose to imagine replacing a single large heat bath with many smaller disconnected ones, each initialized to their ground state. Acting together and left for a sufficiently long time, they would still act to thermalize the system subspace. There is no real gain in doing this as opposed to using a single large heat bath; however, suppose that instead of connecting all of these heat baths simultaneously, one connected them sequentially and allowed each to act long enough to thermalize approximately

before disconnecting it. Again one would expect that if one had a sufficient number of these small heat baths, the system would again thermalize to the desired temperature.

At this point one can ask whether instead of using many different heat baths, one can simply use the same heat bath over and over again. To do so, after disconnecting the small heat bath from the system of interest one would have to reinitialize it to its ground state after each use. The act of reinitializing the heat bath is non-unitary, since all states of the heat bath get mapped into its ground state. However, provided the ground state of the heat bath system is sufficiently simple that it can be explicitly specified, it is possible to do such a reinitialization; in fact, the DiVincenzo criteria [50] for the viability of a quantum computer requires this ability to reinitialize qubits to a specific state consistently. This state is denoted a fiducial state and here we will take the fiducial state of the heat bath to be its ground state.

It is important to note that due to the cyclic nature of this scheme, we do not require that all the energy that must be removed from the system of interest be removed in one go; we only need that each cycle removes an appreciable amount of energy. Thus, we can use a heat bath that is much smaller than the large heat bath discussed above.

This process of repeatedly connecting and disconnecting the system of interest to a small heat bath, and reinitializing the heat bath to its ground state in between, is a form of active cooling. Rather than having energy simply passively flowing to a large heat bath, the step of reinitializing in effect transfers the extracted energy to the environment. Thus, the small heat bath is functioning as a refrigerator, and from here on it will be denoted as such and its Hamiltonian will be denoted  $H^{Refrig}$ .

The active cooling proposed in ref. [49] is a cycle that contains four steps; the cycle is illustrated in Fig. A.1. In the first step, the refrigerator is initialized to its ground state and is not

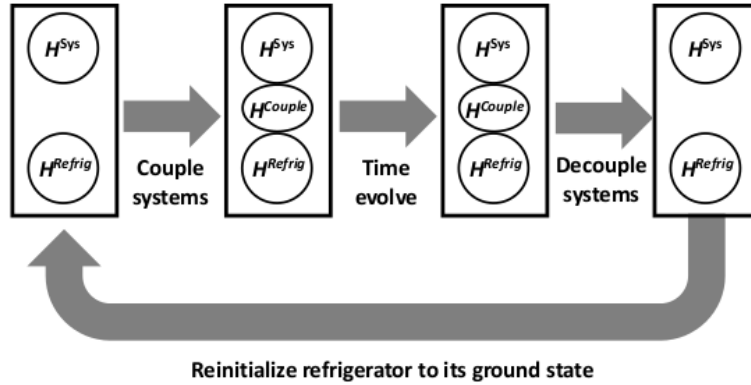


Figure A.1: The active cooling cycle

coupled to the system of interest. In the next step, the two systems are coupled via  $H^{coup}$ . In the the third step, the coupled system is allowed to time evolve under the dynamics of  $H^{sys} + H^{Refrig} + H^{coup}$  for some time. In a quantum simulator this would be done via direct Hamiltonian time evolution and in a gated quantum computer approximated via Trotterization. Next the coupling Hamiltonian is switched off. Finally, the refrigerator system is reinitialized to its ground state and the cycle begins again.

If the coupling Hamiltonian is sufficiently weak that the energy contained in it is always negligible, then the cycle will always act to remove energy from the system of interest. This is because the refrigerator system starts in its ground state so its energy can only increase during the time evolution; on the other hand, energy is conserved in Hamiltonian time evolution so energy must flow from the system of interest to the refrigerator. The active step in the cycle is the act of reinitializing the refrigerator; it is in this step that the energy extracted from the system of interest is removed from the combined system. Note that if the goal is to remove energy from the system, the precise length of time that the system is coupled to the refrigerator is unimportant, as energy will always flow out of the system during a cycle.

Note that even if one were to start the first cycle with both the system of interest and the

refrigerator in pure states, after a single cycle the combined system will be in a mixed state. The time evolution of the coupled system will induce entanglement between the system of interest and the refrigerator. It is instructive to understand how the entanglement can affect the system of interest, especially during the reinitialization step. The reinitialization can be done by measuring the refrigerator's energy and then conducting unitary transformations that return the refrigerator to its ground state. This measurement will collapse all the entanglements, which affects the reduced density matrix of the system of interest. The unitary transformations that follow, however, do not affect the physical system.

If one's goal is to push the system of interest into a reasonable approximation of the ground state, this scheme should be a viable alternative to other state preparation methods and, in the future, could be investigated as such. In such an application, one would need to run a sufficient number of cycles to ensure the system was close enough to the ground state for the application under consideration. The motivation for proposing this scheme, however, was not to prepare the system in the ground state but rather for the eventual study of gauge theories at nonzero temperatures. In ref. [49], it was suggested that this active cooling scheme could be used for such studies provided that they were done in the context of the microcanonical distribution.

The basic idea is that one could run through enough cycles to get the characteristic energy of the system in approximately the correct energy range. The system would be in a mixed state, but its density matrix would not be an approximate thermal distribution. We see this since, even if the system and refrigerator were coupled together long enough to thermalize with each other, the reinitialization step would throw the system of interest out of thermalization due to entanglements. Furthermore, as mentioned above, one cycle is only tasked with removing an appreciable amount of energy. It seems intuitively clear that the time needed to remove a reasonable amount

of energy is significantly smaller than the time needed to allow for thermalization.

Thus, this algorithm is not aimed at constructing an approximately thermal state. It is, however, designed to construct a Boltzmann distribution on the ensemble level. That is to say that the state produced by the algorithm is such that measuring the energy will choose an energy eigenstate with a probability given by a Boltzmann weight. This comes about because the measurement inherent to the reinitialization step affects the system, as mentioned above; however, on average this change to the system is null. Thus, on the ensemble level the system is not pushed out of thermalization. In this way the algorithm constructs a Boltzmann distribution.

This still leaves the matter of how to use this algorithm's thermal construction to, say, calculate thermal expectation values. This is straightforward, as one could directly measure the energy, casting the system into an energy eigenstate at known energy, and then measure the quantity of interest on that particular energy eigenstate. Such a procedure could be repeated and the results binned by energy with each bin covering a narrow range of energies. The quantity of interest could then be averaged over each bin. Since each bin corresponds to a narrow band of energies, this procedure would yield an approximate extraction of the microcanonical value for the quantity of interest. If the system is sufficiently large to well approximate the thermodynamic limit [53], the value of the quantity at fixed energy in a microcanonical description will closely approximate its value in a canonical description at the temperature whose mean energy equals the microcanonical energy.

Since we wish to compute observables as a function of temperature, a key step would be to relate average energies to temperature. In principle, this can be done on a classical computer using standard Monte Carlo methods. One would fix  $T$  and calculate the average  $E$ , repeating this process for a large number of temperatures. One could then invert this relationship to obtain

a mapping from  $E$  to  $T$ .

Such a scheme may be rather efficient at removing energy. It was argued in ref. [49] that if  $H^{Refrig}$  and  $H^{coup}$  were allowed to vary with the cycle and are optimally chosen, then, up to fluctuations, one ought to be able to achieve a fixed fractional energy loss per cycle. Thus the number of cycles would be

$$N^{cycles} = A \log\left(\frac{E_i}{E_f}\right) \quad (\text{A.2})$$

where  $A$  is a constant and  $E_i$  and  $E_f$  are the initial and final energies. Thus, one could achieve very large fractional cooling in comparatively few cycles.

As noted above, this approach is based on a kind of quantum refrigerator [115]. For a review of quantum refrigerators, see [116]. Note also that we can call this a sort of quantum demon refrigerator, as it acts akin to Maxwell's demon [51, 52]. This is because of the reinitialization step of the refrigerator: this is obviously not a unitary operation and thus requires coupling to the environment.

Such an approach should work; however, it has a few drawbacks. One of these is that it is not straightforward to have the process yield states with an energy corresponding to a temperature in the range that one wishes to study. In practice, this may mean that in order to study anything, one must study a broad range of energies/temperatures rather than focus on getting good statistics on a handful of temperatures of interest. This could be problematic if the quantum computing resources are limited.

If the size of the system being studied is sufficiently large this may not matter, assuming that the eigenstate thermalization hypothesis (ETH) [42–44] is valid for the quantity of interest. the ETH implies that a single energy eigenstate for a large system should typically reproduce

the thermal expectation value for a system whose temperature yields an average energy corresponding to the energy of the state. If the system were large enough for the ETH to hold, one would not need to acquire many states to do a statistical average, greatly alleviating the problem. However, it is unclear whether the ETH will hold for quantities such as transport coefficients of field theories. Even if the ETH holds in principle, it is also unclear how large the system will need to be for it to be valid in practice. This latter issue is significant since the earliest practical calculations on quantum computers are likely to be on relatively small systems.

With this in mind, it would be greatly beneficial to have the ability to predict and control the characteristic energy of the system so that one could know when the system is in the energy range that is desired. This ability would allow for this active cooling scheme to be used efficiently to produce data in the desired range without many data points falling outside of said range.

This is where a key benefit of the present scheme lies. Active cooling mainly consists of two steps: time evolution while the refrigerator and system of interest are coupled (which notably conserves energy) and the reinitialization step (which notably does not conserve the energy of the total system). Thus, if one keeps track of the energy that is removed during the reinitialization step, then the characteristic energy of the system of interest can be approximately known. This ability to track energy, plus the mapping between energy and temperature mentioned above, allows the construction of a Boltzmann distribution of a temperature lying in a chosen range.

Keeping track of the energy removed is fairly straightforward: we assume, *a priori*, that the ground state energy of the refrigerator is known. During the reinitialization step, one can measure the energy of the refrigerator and then transform it to its ground state. Given that the energy of the refrigerator before and after is known, the energy that was removed is also known. Note that the ability to do these procedures define the kinds of refrigerators that must be used for

this active cooling scheme. A simple example of this is a refrigerator consisting of spins.

### A.2.1 Parameters

The main parameters of this scheme are the size of the refrigerator, the parameters controlling the Hamiltonian of the refrigerator, the overall coupling strength (which we will denote  $\sigma$ ), the number of cycles  $N$ , and the time that the refrigerator and the system of interest are coupled during each cycle, denoted  $\tau$ . It is straightforward to see that the number of cycles  $N$  will be determined by the amount of energy that has been removed (which is monitored) and the target energy.

Here, we restate that due to the cyclic nature of this scheme, we do not require that the all the energy that must be removed from the system of interest be removed in one go; we only need that each cycle removes an appreciable amount of energy. Thus, we can use a refrigerator that is much smaller than the large heat bath discussed above. In fact, we claim that the refrigerator's size can be on the same scale as the system of interest, or even smaller, and still work efficiently. Again we will mention that this is subject to the trade off of qubit resources and coherence times.

The studies with a “toy model” in this work demonstrate that in fact a smaller refrigerator works well. Future work aims to show this for larger and more physically relevant systems. Future work also aims to study the optimal value of  $\tau$ , since the small systems studied in this paper have energy transfer occurring almost instantaneously.

When it comes to the strength of the coupling, ideally we would want to impose a condition that  $H^{coup}$  is small; specifically, we want a coupling strength  $\sigma$  that is small enough so the coupling cannot appreciably give energy to the system subspace, as discussed above. This condition

can be made explicit: we say  $H^{coup}$  is sufficiently small when  $\langle H^{coup} \rangle \ll \langle H^{sys} \rangle - \langle H^{sys} \rangle_{vac}$  holds at all times. Thus, an ideal choice of coupling strength would be the largest value of  $\sigma$  that satisfies the above constraint. We want  $\sigma$  to be as large as possible because stronger coupling allows for faster energy transfer.

It is important to note, at this point, that our ultimate goal is to develop a scheme that is blind to the system's energy spectrum since *a priori* this is unknown. This means that we want to assume no knowledge of or ability to determine  $\langle H^{sys} \rangle_{vac}$ . Thus, we must dispense with this condition, at least *a priori*, and deal with a coupling that has the ability to heat up the physical subsystem. In practice,  $\sigma$  is chosen based off of a best guess of this ideal value.

A similar best guess approach is taken for the parameters that describe the refrigerator's Hamiltonian. Ideally, we want the average energy density of the levels of the refrigerator to match closely the average energy level density of the system of interest to promote efficient energy transfer. Again, we do not assume knowledge of this level density at all.

At this point the selection of the refrigerator parameters as well as  $\sigma$  seems to be almost blind, which is concerning as they are important parameters to the efficiency of this scheme. However, it is important to recall that this scheme starts with an initial state that has an energy much higher than the target energy since those states are the only ones we know how to straightforwardly write on quantum computers. Thus, we suggest that choosing a  $\sigma$  that is not too big is not a difficult task. We also believe that picking refrigerator parameters that are serviceable is a similarly manageable task. Again, we point out that we do not require huge amounts of energy removed every cycle; we simply want appreciable amounts of energy removed. Said another way, we claim that this algorithm is relatively robust in that it is not overly sensitive to the initial parameter inputs.

In this paper we develop a detailed algorithm in the context of a simple “toy” model to illustrate how such a scheme can work. Before describing this algorithm, we want to highlight the beneficial aspects of this algorithm. First and foremost, it is capable of tracking the energy of the system to reasonably accurately aim at the desired characteristic energy and thus, with the  $E$  to  $T$  map mentioned above, the desired temperature. Also, unlike the projective cooling algorithms, its complementary system can be of a size on the same order of, instead of significantly larger than, the size of the system subspace. We also utilize the complementary system iteratively to significantly reduce the amount of time required. Moreover, the algorithm was designed to work for any initial state and without any knowledge of the energy spectrum. The algorithm has two other useful features that apply to its use on larger systems. One is that the refrigerator can be used a thermometer to both test whether the system has achieved thermal equilibrium and if so at what temperature. The second is that the system can thermalize (to good approximation) at an early stage and should remain close to equilibrated throughout. These aspects are discussed in more detail in Section IV.

While this algorithm was developed and illustrated in the context of a particular toy model, it is highly plausible that it should work quite generally. Moreover, as will be discussed later, there are good reasons to believe that the model should have good scaling properties when applied to lattice models with local interactions.

### A.3 A Toy Model

In testing our algorithm, we want to demonstrate its applicability to different Hamiltonians, so we want to test it on a class of Hamiltonians instead just one. Thus, we will not present a

singular Hamiltonian for our “toy” model but rather the method we used to generate it. We claim that our algorithm works for every Hamiltonian this method generates. The results portrayed in subsection B pertain to only one of these Hamiltonians, but similar results can be obtained from other Hamiltonians generated from the method described below.

In constructing the Hamiltonian for our “toy” model, we wanted to pick a rather generic Hamiltonian but we also wanted to mimic the property of physical systems wherein the level density increases exponentially with energy. Thus, we handcrafted an energy spectrum that has the latter property by picking an energy span (0 to 1000 energy units), identifying 10 regions of 100 energy units each, and using the function  $round(e^{0.8\sqrt{i}})$  (where  $i$  ranged from 0 to 9) to determine the number of energy levels in each region; these levels were evenly spaced in their respective energy regions. With this energy spectrum in hand, we moved to a randomly chosen eigenbasis for coupling reasons that will become clear below. Finally, to introduce some randomness to the system’s Hamiltonian, we added to our existing Hamiltonian a randomly generated matrix where each entry has chosen from a normal distribution with mean 0 and a standard deviation that is small enough to not destroy the exponentially increasing level density.

We note here that the system was chosen to have 57 levels, as that is enough levels to be able to appreciably see the desired exponential increase in level density while also not overtaxing the classical computers on which simulations of this system were conducted. We also note that since this algorithm was developed for QCD problems that obey time-reversal symmetry, we made sure that the system Hamiltonian was symmetric.

### A.3.1 The algorithm

For the refrigerator, we used a simple spin model with 4 spins uncoupled to each other:

$$H^{Refrig} = \sum_{i=1}^s h_i \sigma_{z,i} \quad (\text{A.3})$$

where  $s = 4$  is the number of spins in the refrigerator,  $\sigma_{z,i}$  is the z Pauli matrix for the  $i$ th spin, and  $h_i$  is the field strength for the  $i$ th spin (*i.e.* it fixes the energy difference between up and down). Note that the  $h_i$ 's are chosen to be positive, so that the ground state is when all the spins are down. Note also that since we are using 4 spins,  $H^{Refrig}$  is an 16x16 matrix as opposed to  $H^{sys}$  which is a 57x57 matrix. This represents a refrigerator that is significantly smaller than the system of interest. Clearly for larger systems it would make sense to use more than four spins.

We used a different field strength for each spin to ensure that correlations between the energies of the spin do not distort the results. The phenomenon of concern is where if there are two levels in the combined system with very close energies (where one level arises from a higher energy level in the system and a lower energy level in the refrigerator while the other level arises from a lower energy level in the system and a higher energy level in the refrigerator), energy transfer from the system to the refrigerator will proceed preferentially by the combined system moving from one of these levels to the other. This effect favors certain energy levels in the system of interest, which we want to avoid to promote equilibration. Thus, we used a different field strength for each spin to suppress these effects. We suspect that, for larger systems, this effect will become negligible and thus a single field strength for all the spins might well be used. For similar reasons, and also to encourage more efficient energy transfer, we change these field

strength values periodically throughout the algorithm's run. We will be more explicit about this below.

The reinitialization step of a cycle is conducted by picking a singular spin and measuring it, which forces it into either  $|\uparrow\rangle$  or  $|\downarrow\rangle$ . If it was measured up, it is then flipped down via  $\sigma_{x,i}$ ; if it was measured down, the reinitialization step is complete. Note that we are only measuring one spin per reinitialization step instead of reinitializing all spins. This is because flipping one spin is much less of a disturbance to the total system than flipping all of the spins. The idea is to flip one spin at a time and while flipping one spin, the physical subsystem can remain coupled to the remaining spins. If the system were to achieve thermal equilibrium, flipping a single spin would only minimally disturb it. Thus, if thermal equilibrium is reached, the system can subsequently be kept in quasi-thermal equilibrium as the algorithm proceeds. Our toy model is too small to demonstrate this benefit, but it is straight forward to see that a single spin flip will be a small perturbation for larger systems.

At this point we will make it clear that there are three main sources of energy change for this algorithm: 1) the energy lost from flipping a spin down; 2) the energy change due to measuring a spin (either up or down); and 3) the energy change due to changing the  $h_i$ 's. The first source of energy change is the most controlled: we set the parameters of the refrigerator and thus we know exactly how much energy is lost when flipping a spin from up to down (namely  $2h_i$ ). We keep track of this energy loss so that the energy of the system can be approximately predicted.

The second source of energy change, arising from spin measurement, is the main source of uncertainty when it comes to predicting the energy of the system of interest. Note, however, that the energy changes arising from many spin measurements should average to 0 (holding the field strengths constant). Thus, this source of error in the physical subsystem's energy prediction

should be suppressed for larger systems, as they naturally require more spin measurements.

The third source of energy change is due to manipulation of the refrigerator parameters. This energy change comes because when the field strength values  $h_i$  are changed, the probability of the refrigerator being in any of its energy levels is not changed but of course the energy of the levels is changed. For this toy model, each time we chose the  $h_i$ 's we chose them from a uniform distribution with fixed endpoints. It makes intuitive sense that with more spins in the refrigerator, the changes in the energy expectation value of the refrigerator will be less severe. This source of energy change is another source of error in the prediction of the physical subsystem's energy, but for the reasons just stated we believe this error is also suppressed for larger systems, which come with larger refrigerators.

Finally, we now define the coupling scheme for our toy model. Unlike the choice of Hamiltonian for the refrigerator, the coupling Hamiltonian described here is specific to the toy model. As will be discussed below, the appropriate coupling Hamiltonian for lattice models is straightforward to construct: each spin in the refrigerator is coupled to a small region of space of the physical subsystem.

$$H^{coup} = \sum_{i=1}^s \sigma_{x,i} \otimes K_i \quad (\text{A.4})$$

For the toy model, the coupling Hamiltonian  $H^{coup}$  is defined as in (4), where the  $i$ th spin in the refrigerator is coupled to the physical system via a Kronecker product of  $\sigma_{x,i}$  in the refrigerator subspace and a matrix  $K_i$  in the system subspace. Each matrix  $K_i$  is defined as follows: a 57x57 matrix (same size as the system Hamiltonian) whose entries are all 0 except for a randomly chosen 3x3 block. This 3x3 block has entries chosen from a normal distribution with mean 0 and standard deviation  $\sigma$  which defines the coupling strength. Also, to ensure time-reversal symmetry

is again respected while keeping the Hamiltonian hermitian, the transpose of this matrix is added to itself before the Kronecker product in (4) is taken. Note that the coupling is via the  $x$  Pauli matrix; this is done so that, when the refrigerator is in its ground state (i.e. all spins down), the expectation value of the coupling Hamiltonian is zero.

This 3x3 block coupling scheme was chosen to be as close to analogous as possible to the case of lattice models wherein each spin in the refrigerator is coupled to a small region of space. We use the random eigenbasis for the system Hamiltonian to make this 3x3 block coupling effective.

Now with all the parts of our toy model defined, we can delineate a run of the algorithm. The system is initialized to its highest energy state while the refrigerator is initialized to its ground state. Parameters  $\sigma$  and  $\tau$  are chosen. When choosing  $\sigma$ , we ran multiple values from a wide range to assure that the algorithm is not too sensitive to this initial input. As discussed above, we reset the field strengths  $h_i$  multiple times. Each time, they are randomly generated from a uniform distribution whose endpoints are chosen at the outset. These endpoints are chosen so that there is a large enough gap to give a good range of  $h_i$ 's so that even if some are too small/large, some will be much more reasonable. On the flip side, a larger range of possible  $h_i$ 's will increase the error due to this change, so it should not be made too wide a range. These considerations determine the choices for these endpoints.

Now, between reinitializations, we reset  $h_i$  as well as the coupling (meaning the positions of the 3x3 blocks and the entries of these blocks are regenerated randomly),  $p$  times. For this toy model, we used  $p = 4$ . This process of resetting the refrigerator parameters and the coupling is done to promote efficient energy transfer. Thus, the system is coupled to the refrigerator for a time  $\tau/p$ , then they are decoupled and the refrigerator parameters and coupling are reset. This is

repeated until a full time  $\tau$  has passed with the system and the refrigerator coupled to each other. At this point, the reinitialization step occurs: a single spin is chosen, measured, and flipped down if it was measured up. The spins are chosen in cyclical manner: 1,2,3,4,1, etc. (where the spins are indexed between 1 and  $s$ ).

This process is repeated  $N$  times. The coupling strength  $\sigma$  is set for each cycle as  $\sigma(t) = \frac{A}{1+Bt^\alpha}$  where  $A$  and  $B$  are constants that determine the starting and ending  $\sigma$ s,  $\alpha$  determines the speed of decrease, and  $t$  ranges from 1 to  $N$ . For the last iteration, the reinitialization step is slightly different: since we want to compare the initial energy to the current energy of the system, we want the refrigerator to be in its ground state both times. Thus, we reset all  $s$  spins instead of just one. Finally, the energy of the system is measured, forcing the system into a singular energy eigenstate.

Before presenting results from numerical experiments, we note that the choices made in developing the algorithm for this toy model was aimed at illustrating the concept with small systems. As mentioned, future work aims to show that the concept of this algorithm can work efficiently for larger systems.

### A.3.2 Results

We ran our algorithm on an ensemble of 1250 copies of a system chosen in the way described above. Throughout each run, we kept track of the energy that was removed due to spin flips. At the end of each run, the energy of the system of interest was measured. Then we compared our energy estimate (initial system energy minus the energy removed via spin flips) with the energy measurements. As discussed above, there are other sources of energy change in the

system besides the spin flips, but for larger systems these sources will be suppressed; thus, we will compare both our energy estimates defined above and the expectation value of the system with the energy measurements to show the algorithm's ability to control the energy. We stress here that in reality, taking the expectation value of the system is quite costly, and we are in no way including it as part of our algorithm; we are simply using the expectation value of the system energy as a proxy for the energy estimate (initial energy minus spin flip energy) for a larger system which will have other energy changes suppressed.

For reference, the system used has an energy span of about 1000 energy units. The average difference between the energy estimate and the energy measurements was  $-60 \pm 269$  energy units; The average difference between the expectation value of the system and the energy measurements was  $11 \pm 203$  energy units.

To determine if these standard deviations are consistent with what we expect, we must first turn to our claim of thermalization. Assuming the ensemble is thermalizing (an assumption that will be justified below), we can estimate the temperature by calculating an average energy from the energy expectation values. From this average energy and a knowledge of the energy levels, we can extract a temperature. We note here that this algorithm does not require knowledge of all energy levels, it only requires a mapping from energy to temperature as discussed above. This procedure provides a temperature of 218 energy units.

From this temperature, we then sample multiple energy eigenstates from a Boltzmann distribution of this temperature. The standard deviation of these sampled energies was 232 energy units. This puts our standard deviations above in perspective: our standard deviations are consistent with what one would expect when sampling from a Boltzmann distribution at a temperature of 218 energy units.

Now that we have shown the algorithm's ability to monitor the energy, we turn to proving the assumption that we do in fact have a Boltzmann distribution. These results can be seen in Figure A.2. From our 1250 runs, we bin the measured energies into 5 bins. These bins were constructed so that a perfect Boltzmann distribution of a temperature 218 energy units would have 20% probability for each bin. The bins are designed to each have equal area; as such the heights are irrelevant and so the y-axis labels have been left off Figure A.2.

As Figure A.2 shows, the data (represented by the colored bars) lie within 1 standard deviation in four out of the five bins; the fifth bin lies within 2 standard deviations. Thus, we conclude that we have in fact constructed a Boltzmann distribution with our algorithm.

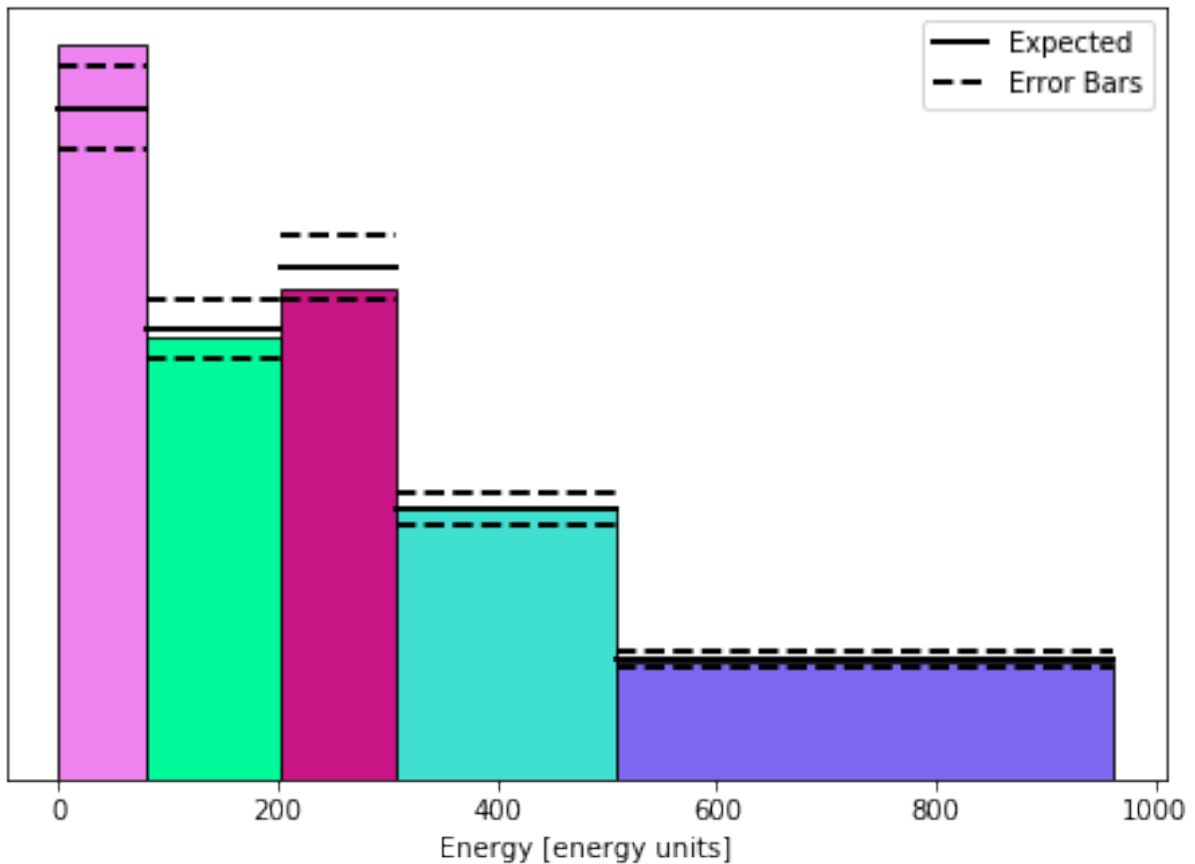


Figure A.2: Data (colored bars) vs Expected (black lines)

## A.4 Generalization to larger systems

We have developed this algorithm in the context of a small simple toy model. It is hoped that quantum computers will continue to develop to the point that they can be used for much larger systems of physical interest. The algorithm developed here should work for such systems. Additionally, in larger systems some refinements are possible that should act to make it both more accurate and more controlled. Clearly one would want a refrigerator with more than four spins, but it would still be useful to have them uncoupled and measured and flipped as needed to lower the energy.

As noted above, it is plausible that one does not need to randomize the field strengths of the refrigerator for large systems as the correlations leading to spurious results become increasingly less likely. However, there is very little cost to randomizing them.

### A.4.1 The refrigerator as a thermometer

For large systems, one expects the system to thermalize at least approximately relatively early in the process. Indeed, if the initial state is very high energy and refrigerator is relatively small it is likely to thermalize at a negative temperature. Thus, there is a potentially useful alternate version of the proposed algorithm wherein only one spin at a time is reinitialized instead of the entire refrigerator. Since each spin flip only removes a very small fraction of the total energy, one expects that shortly after a spin flip, the system remains in approximate equilibrium with nearly the same temperature. Thus, by looking at the probability that a spin is measured up over a number of successive spin measurements, one should be able to determine an approximate temperature.

The obvious advantage to this is that in thermal studies of large systems we typically wish to study a system at a known fixed temperature as opposed to a fixed energy density. Continuously monitoring the temperature is a very simple way to do this—one simply runs the algorithm until the system reaches the temperature of interest and then stops.

It is noteworthy that one needs to monitor the probability that spins are up or down in any case if one wants to run the algorithm efficiently. As discussed above, the field strength can neither be too big or too small for the efficient running of the algorithm. If it is too large, the probability of a spin flip is very low and rapid energy transfer from the system to the refrigerator is inhibited. Conversely, if it is too small, then spin flips can occur easily but each one deposits very little energy and again the energy transfer is inhibited. Since whether it is too big or too small (in this sense) varies with time as the algorithm runs, one needs to change the field strength as the algorithm evolves. A useful way to tell whether the field strength is in the “Goldilocks” region where it is neither too large nor too small is to focus on the temperature or equivalently the average probability of spin up. The Goldilocks region is typically where the field strength is similar to  $|T|$ , where the absolute value sign accounts for negative temperatures. This criterion is useful except in the immediate vicinity of the region where  $T$  diverges. As the system moves out of the Goldilocks region one needs to adjust the field strength.

#### A.4.2 Lattice models

Lattice models of formally infinite systems play an important role in both condensed matter physics and in nuclear and particle physics where they are used as proxies for continuous field theories by choosing the lattice spacing to be much smaller than the relevant physical scales in

the problem. Such models are studied via numerical simulation and they are studied on lattices whose physical extent is large compared to the relevant physical scales. Note that the present algorithm was principally developed to describe thermal properties of QCD and related gauge theories which are typically treated numerically as lattice models.

In many ways, the algorithm is better suited to lattice models than the toy model considered here since lattice models tend to thermalize locally. Suppose that one couples spins in the refrigerator to parts of the system in fixed spatial regions of the lattice model. If the model was in thermal equilibrium prior to a measurement of the refrigerator, the effect of a measurement of a single spin can at worst effect the equilibrium of the system in the region coupled to the spin and regions close to it. Thus, if one does not make another measurement of a spin that is coupled to a region nearby within a time characteristic of thermalization, then after the system equilibrates once, all future measurements will be of an equilibrated system.

Accordingly, when implementing the algorithm on a lattice models one should couple the spins in the refrigerator to particular spatial regions of the model. Moreover, unlike in the toy model, there is no reason to randomly change the states or regions to which a given spin is assigned. Rather one should sample the the various regions in a manner that ensures that one does not sample a region near to one that has previously been sampled unless a sufficient time has elapsed to allow the system to equilibrate locally. One can do this for example by randomly sampling spins coupled to a region subject to a constraint that no spin coupled to a nearby region has been sampled within a prescribed time.

### A.4.3 Algorithm for large lattice systems

While it is too expensive to simulate large lattice systems on classical computers, it is instructive to outline how the algorithm would work for large lattice systems. The refrigerator will still consist of uncoupled spins; these spins will be coupled to localized regions of the system of interest. These localized regions are to be spaced out, ideally to "cover" the entirety of the physical system. There are many ways to define the coupling of the spins to the lattice system; of course, this must be done in a gauge invariant way. A simple example would be to couple spins in the refrigerator via  $\sigma_x$  to a single spatial plaquette in the lattice system. In theory, any gauge invariant scheme in the lattice system could work, including any size of spatial Wilson loop, and empirical studies would need to be conducted to illuminate the most efficient form of coupling.

With a refrigerator and coupling scheme in hand, the algorithm proceeds with a number of cycles. These cycles are to be slightly different from the ones diagrammed in Fig. A.1: while the physical system and the refrigerator are still to be coupled and then the combined system time evolved, we do not require the full decoupling of the refrigerator with the physical system. Instead, we propose that one spin at a time is decoupled from the physical system and reinitialized (i.e. measured and if it was measured up, flipped down). We take this one-spin-at-a-time approach so that the physical system can, in the meantime, continue to evolve with all the other spins. The selection of the spin to be reinitialized should be done in the way discussed above: if a spin from a certain neighborhood is reinitialized, another spin from this neighborhood will not be reinitialized for an amount of time that should allow for the system to locally equilibrate again.

Furthermore, the results from the reinitialization steps should be recorded. The energy removed from the total combined system can be monitored, as is done with our small toy model,

but also the results of spin up measurements should be recorded. As discussed above, this allows for the refrigerator to be used as a thermometer: the average probability of measuring a spin up relates to a temperature. With this tracking ability, one can stop the time evolution (by decoupling all the spins) once the target temperature is reached.

## A.5 Conclusion

We have outlined an algorithm developed to work for a generic initial state and system of interest that produces a good way to sample energy eigenstates from a thermal distribution so that thermal expectation values can be calculated, and we have shown that is effective for small systems. We have outlined how the approach should work for larger systems described by lattice field theories. Future work aims primarily to test this algorithm for larger systems. With this, other variations of the algorithm can be tested out, including different refrigerator Hamiltonians. Future work also aims to test this algorithm on simple gauge theory models, to show that this algorithm may be effective for the specific problems it was motivated to address.

## Acknowledgments

This work was supported in part by the U.S. Department of Energy, Office of Nuclear Physics under Award Number(s) DE-SC0021143 and DE-FG02-93ER40762.

## Appendix B: Zeno effect suppression of gauge drift in quantum simulations

*Carter Ball and Thomas D. Cohen*

**NOTA BENE: What follows is an exact reprint of the 2024 article “Zeno effect suppression of gauge drift in quantum simulations” by Carter Ball and Thomas D. Cohen, originally published in Physical Review A [6].**

### Abstract

Quantum simulation of lattice gauge theories is a promising tool for the study of many complicated problems including ones with real-time dynamics. For gauge theories, however, there is a major challenge in maintaining gauge invariance during time evolution. Such theories have a full Hilbert space that is larger than the physical space—the set of states which are gauge invariant or equivalently respect the Gauss law. While an exact implementation of Hamiltonian dynamics starting in the physical Hilbert space will keep the system in the physical space, various types of errors will inevitably produce components outside of it. This work proposes a method of suppressing this gauge drift via the Zeno effect. As in the standard picture of the Zeno effect, our method relies on frequent projection onto the physical subspace. Additionally, a technique is discussed to reduce the speed of the gauge drift, which helps to reduce the required frequency of projections. We demonstrate our method on a  $\mathbb{Z}_2$  gauge theory toy model.

## B.1 Introduction

Quantum computers promise to be vital tools for the simulation of many fundamental yet complicated problems [8, 12, 13, 117, 118] from a range of subfields, including particle physics, nuclear physics, and condensed matter. The simulation and study of complex systems often requires much more computing power than classical computers can provide [119, 120]; thus, there is a present need to map these systems onto quantum devices. Analog quantum devices manufacture a system that can imitate, at least in some regime, the desired theory while digital quantum devices map a theory onto an array of qubits and control it via quantum gates [121]. Much research has gone into studying these quantum devices and how they can be used for quantum simulation [122].

Like many complicated theories, lattice gauge theories can, in principle, benefit enormously from efficient and reliable quantum simulation [61, 73, 123–136]. These theories typically place matter degrees of freedom on the sites of the lattice while gauge field operators are placed on the links; it is typical to work in a Hamiltonian formulation with discretized space but continuous time [92, 137, 138]. This is usually done in the temporal gauge [139] (which sets the time component of gauge fields to zero) and by taking the continuous time limit. Note that the temporal gauge does not fully exhaust the gauge freedom, leaving a residual gauge freedom for the spatial components. Transformations of this residual gauge are generated by Gauss law operators (in analogy to classical electrodynamics). Physical states are then defined as those that are invariant under all residual gauge transformations.

Much work has gone into the study of both Abelian and non-Abelian lattice gauge theories. Notably, maintaining non-Abelian gauge invariance is particularly difficult. When it comes to

analog quantum simulation, many methods have been developed to address this gauge invariance problem, including constructing a system that makes gauge invariance a consequence of internal symmetry such as angular momentum [55, 56]. Another analog simulation method is to impose an energy penalty on the unphysical states [54, 59–67], often by adding a term to the Hamiltonian proportional to the square of the Gauss’ law operators. Then, the lowest energy levels of the system are the physical ones, so keeping the system in the low energy regime results, to good approximation, in keeping the system within the physical subspace of the much larger Hilbert space. Variations on this method have been developed, including one where the addition to the Hamiltonian is directly proportional to the Gauss’ law operators [68]. Another such variation on the energy penalty method uses local pseudogenerators, which exactly match the Gauss’ law operators in the relevant physical subspace but not in unphysical sectors, allowing these operators to be much less complex and thus cheaper to engineer in practice [69–71]. Interestingly, for sufficiently large energy penalties, the protection term added to the Hamiltonian, constructed with local pseudogenerators, can be seen as a strong projector which constrain the dynamics of the system to quantum Zeno subspaces within a given timescale [71, 140].

The method presented in this paper concerns digital quantum simulation. Ideally, any method to simulate a non-Abelian gauge theory of interest, such as quantum chromodynamics (QCD), would only simulate the gauge invariant (i.e. physical) states. Methods of this kind have been proposed which solve the Gauss law, either eliminating the gauge fields [72–74] or the matter fields [75–79]. Another method uses dual formulations of the gauge theories to enforce the Gauss law [141–143]. While the methods discussed above don’t have to deal with unphysical states, their physical implementation may be more challenging.

Thus, many methods simulate all states in the Hilbert space of the system, and must contend

with keeping the system within the exponentially smaller physical subspace. Crucially, work has been done to develop methods of time evolution that respect gauge invariance [14, 126]. Thus, ideally time evolving an initial state that is constructed to be physical would keep the system within the physical subspace. However, quantum noise, gate errors, and various errors from approximations can lead to the system picking up contributions from unphysical states. Previous work has addressed this problem in a variety of ways. Perhaps the most straightforward is to simply accept the unphysical contributions on the premise that these contributions are reduced as the time step is decreased [128]. Another method is dynamical decoupling [80], a concept from quantum control theory where local symmetries are maintained via periodic driving. Also, oracles have been developed [81] to check for violations of Gauss' law for Abelian lattice gauge theories, with similar oracles for non-Abelian theories under development.

Furthermore, a method of engineering classical noise to constrain the system to the physical subspace [82] has been proposed. This method serves as a classical analog to the quantum Zeno effect by using classical fluctuations of a perturbing field to dissipate unwanted evolution into unphysical regions of the Hilbert space. The method of this paper, in contrast, rests on the standard quantum Zeno effect [83], which uses frequent projections to keep the system within a desired space; for our purposes the desired space is of course the physical subspace of the system's Hilbert space.

This paper is organized as follows: section II outlines our method, which takes inspiration from the rodeo algorithm [2]—a method of projection—and a technique of unitary gauge drift suppression. Section III demonstrates our method on a toy model of simulating pure  $\mathbb{Z}_2$  gauge theory on a  $2 \times 2$  lattice with periodic boundary conditions. Section IV goes into more detail about the implementation of our method, and Section V provides concluding thoughts and discusses

the required future work.

## B.2 Method

The method exploits the quantum Zeno effect to maintain a close approximation of gauge invariance. For a non-Abelian theory, the gauge invariant, or physical, states are given by those states  $|\psi\rangle$  that satisfy

$$G_x^a |\psi\rangle = 0 \tag{B.1}$$

for all  $a$  and  $x$ , where  $G_x^a$  is the  $a$ th Gauss' law operator at site  $x$ . Note that for compact Lie gauge groups (e.g.  $SU(N)$ ), these  $G_x^a$  operators generate local gauge transformations.

To leverage the Zeno effect, we must invoke a method of projection onto the physical subspace. Our method adapts previous work on the rodeo algorithm [2, 36–38]. We also incorporate a method of suppressing coherent gauge drift, discussed below.

### B.2.1 Projection via the Rodeo Algorithm

The rodeo algorithm was developed to prepare eigenvectors of a given Hamiltonian. See figure B.1 for the circuit diagram of the algorithm [2].

The rodeo algorithm aims to construct, to good approximation, an eigenstate of a system with a chosen energy by "shaking off" amplitudes of other, unwanted eigenstates, like a bull shaking off its rider at a rodeo. The algorithm accomplishes this task with ancillary qubits as shown in the circuit diagram of Fig. B.1. To see how the rodeo algorithm constructs the desired eigenstate, we will write out how it works for one ancillary qubit. Starting with the system in the

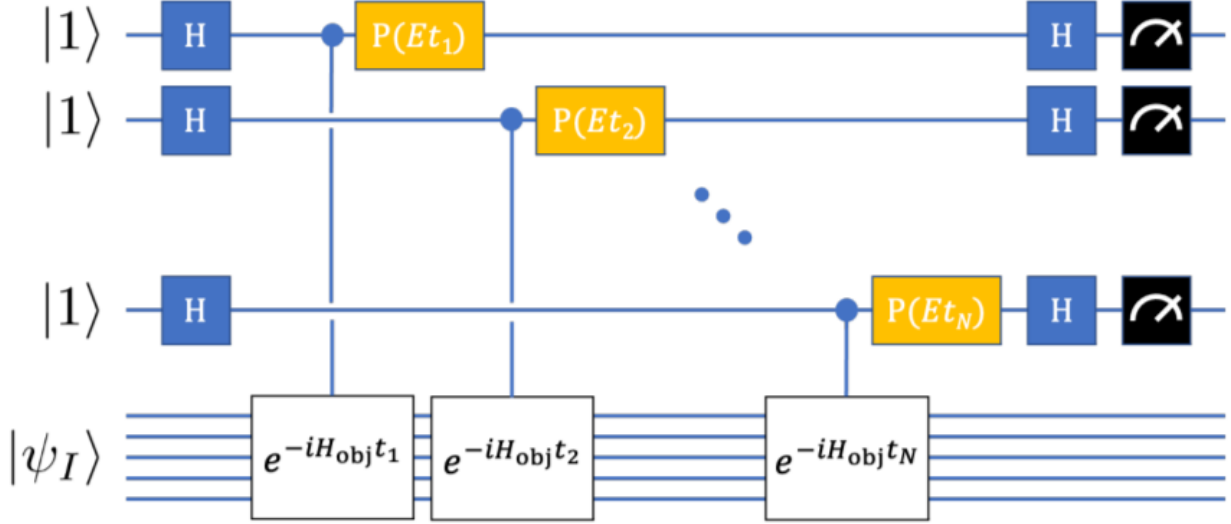


Figure B.1: The circuit diagram for the rodeo algorithm [2]

initial state  $|\psi_I\rangle$  and the ancillary qubit initialized to  $|1\rangle$ , we have

$$\begin{bmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & -\frac{I}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & Ie^{iEt} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & e^{-iH_{obj}t} \end{bmatrix} \begin{bmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & -\frac{I}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 0 \\ |\psi_I\rangle \end{bmatrix} = \quad (\text{B.2})$$

$$\begin{bmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & -\frac{I}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & Ie^{iEt} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & e^{-iH_{obj}t} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}|\psi_I\rangle \\ -\frac{1}{\sqrt{2}}|\psi_I\rangle \end{bmatrix} = \quad (\text{B.3})$$

$$\begin{bmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & -\frac{I}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}|\psi_I\rangle \\ -\frac{1}{\sqrt{2}}e^{-i(H_{obj}-E)t}|\psi_I\rangle \end{bmatrix} = \quad (\text{B.4})$$

$$\begin{bmatrix} \frac{1}{2}(I - e^{-i(H_{obj}-E)t})|\psi_I\rangle \\ \frac{1}{2}(I + e^{-i(H_{obj}-E)t})|\psi_I\rangle \end{bmatrix} \quad (\text{B.5})$$

where above,  $H_{obj}$  is the Hamiltonian of the system of interest (ie the object system) and  $E$  is the desired energy of the constructed state. Equation (2) writes out the actions of (reading from right to left) the Hadamard, controlled time evolution, phase shift, and Hadamard gates, respectively,

on the initial state. Note that the first Hadamard gate gives equal amplitude, up to a sign, to the up (ie  $|0\rangle$ ) and down (ie  $|1\rangle$ ) components of the ancillary qubit (equation (3)), then the controlled time evolution and phase shift gates give a phase only to the down component (equation (4)), and finally the last Hadamard gate causes interference between the up and down components (equation (5)).

The final step in the rodeo algorithm is to measure the ancillary qubit(s). A successful implementation of the rodeo algorithm measures every ancillary qubit in the state  $|1\rangle$  at the end. In the single ancillary qubit example above, this results in the transformation

$$|\psi_I\rangle \rightarrow \frac{1}{2}(I + e^{-i(H_{obj}-E)t}) |\psi_I\rangle \quad (\text{B.6})$$

To see the benefits of this transformation, we decompose our initial state into energy eigenstates  $|\psi_I\rangle = \sum_n c_n |n\rangle$  where  $H_{obj} |n\rangle = E_n |n\rangle$ . The amplitude of each energy eigenstate then transforms as

$$c_n \rightarrow \frac{1}{2}c_n(1 + e^{-i(E_n-E)t}) \quad (\text{B.7})$$

which means the probability of each energy eigenstate transforms as

$$|c_n|^2 \rightarrow |c_n|^2 \left| \frac{1}{2}(1 + e^{-i(E_n-E)t}) \right|^2 \quad (\text{B.8})$$

$$\rightarrow |c_n|^2 \cos^2\left((E_n - E)\frac{t}{2}\right) \quad (\text{B.9})$$

Thus, the probability for the system to be in a particular energy eigenstate  $E_n$  is multiplied by

a factor  $\cos^2((E_n - E)\frac{t}{2})$ , which is notably 1 if  $E = E_n$  and less than 1 otherwise. This is how the rodeo algorithm works: it constructs an approximate version of the state of energy  $E$  by suppressing all other states with a  $\cos^2(\frac{1}{2}t\Delta E)$  factor. Stronger suppression of contributions from unwanted energy eigenstates can be accomplished with multiple rounds of this algorithm, as discussed in [2, 36, 37]. Note that in order to implement a round of the rodeo algorithm, a time  $t$  must be chosen. If the times are chosen strategically for the various rounds, remarkably strong suppression can be achieved with very modest resources [38].

We note here that the rodeo algorithm will fail to project onto the state of interest with finite probability (i.e. if any ancillary qubit is not measured in  $|1\rangle$ ), where failure means that the resulting state is orthogonal to the target state. Crucially, we note that the probability of a successful projection is proportional to the overlap of the initial state of the system with the target state of the projection. As we plan to use this projection to leverage the Zeno effect, which means making sufficiently many measurements so that the overlap with the target subspace is close to unity, the rate of failure can be made negligibly small. The chance of failure serves as the probabilistically defined time span for which our method works; for example, if the probability for a failed projection is one in a million, we would expect one failure within one million successive projections.

So far, we have focused on the rodeo algorithm due to its key feature of leaving desired amplitudes (ie the amplitude of the desired energy eigenstate) untouched while suppressing all other amplitudes. Here, we note that there is no reason this algorithm works only for the construction of the eigenstates of the Hamiltonian. Thus, going forward, we will refer to rodeo projections as any projection of the form expressed in Fig. B.1 with some hermitian operator in place of the Hamiltonian.

Recall that our goal is to use frequent projections onto the physical subspace to maintain gauge invariance of the system via the Zeno effect. With this in mind, we now discuss the form of a rodeo projection that we will use to project onto the physical subspace. We replace the Hamiltonian with a new hermitian operator that we will call  $G^2$ :

$$G^2 = \sum_{a,x} (G_x^a)^2 \quad (\text{B.10})$$

From the above definition of the Gauss' law operators, we see that all physical states are eigenstates of  $G^2$  with eigenvalue 0. Furthermore, all the other eigenstates will be unphysical and with eigenvalues greater than 0, since  $G^2$  is positive definite by construction. It is this spectrum of  $G^2$  that we will exploit when using rodeo projections: instead of constructing some eigenstate of a given Hamiltonian, we will construct a physical state by constructing a ground state of  $G^2$  via rodeo projections. This means replacing  $H_{obj}$  in Fig. B.1 with  $G^2$ ; furthermore, as the eigenvalue of the desired eigenstate (ie a physical state) is 0, we will remove the phase shift gates, since the required phase shift is  $0 * t = 0$ .

This is the core of the method outlined in this paper: a rodeo projection with this  $G^2$  operator will leave all physical amplitudes untouched while suppressing all unphysical amplitudes. Doing this projection frequently enough will keep the system in a close approximation of a gauge invariant state via the Zeno effect.

There is one minor detail that we must clarify before moving on. The effectiveness of this projection is helped if the spectrum of the operator, in our case  $G^2$ , is discrete. That way there will be less contamination from the eigenstates nearest to the target eigenstate. Luckily, the spectrum of  $G^2$  for our principal case of interest  $SU(3)$  gauge theories, is in fact discrete. This can be

seen very straightforwardly using basic quantum mechanics, though We will point out one small detail: the basic argument implicitly relies on the fact that all states have non-negative norms. While this is obviously true for physical states it is not necessarily true for the full Hilbert space of QCD. However, we are working within a Hamiltonian formulation and using the temporal gauge, which is a ghostless gauge [144], meaning that we do in fact have non-negative norms for all states. Thus, all components  $G^a$  have discrete spectra, so  $G^2$  has a discrete spectrum as well. This means the ground state of  $G^2$  is gapped, which is helpful when it comes to projection.

## B.2.2 Suppressing Coherent Gauge Drift

The basic idea of the Zeno effect is that if you project back to the desired space before straying too far out of it (i.e. frequently enough), then the system stays in the desired space with high probability. This is for a simple reason: projections depend on the probability that the system is in the desired state, which corresponds to the square of the magnitude of the amplitude. Thus, if one projects when the amplitude for the system drifting into an unwanted state is some small value  $\epsilon$ , the probability of failure is  $\epsilon^2$ —a much smaller value.

Consider what this means in practice if one wishes to study a system’s time evolution over a time,  $T$ , and gauge-invariance-violating errors in the amplitude accrue at a rate  $r$ . Suppose one makes  $N$  evenly spaced projections (which for the sake of this analysis are assumed to be perfect) during the time evolution. The probability of a failure—obtaining a state outside the physical space—at each measurement is  $(rT/N)^2$ , and the probability of a successful run for the full time evolution (i.e. the probability of  $N$  successes and 0 failures) is given by

$$P_s = \left(1 - \frac{r^2 T^2}{N^2}\right)^N \approx 1 - \frac{r^2 T^2}{N} \quad (\text{B.11})$$

where the approximate equality holds in the limit of large  $N$  and  $1 - P_s \ll 1$ . The key point is that for  $N \gtrsim r^2 T^2$  there is a substantial probability that the full time evolution can occur with the system restricted to the physical space. Note that this means that by making sufficient measurements, the evolution can stay in the physical space even if the evolution time is much larger than  $1/r$ . This a major improvement.

Of course, if one makes a finite number of projections there is a finite probability of failure—obtaining a state outside the physical space. If that happens, the only recourse is to stop the calculation and begin again—an expensive prospect. Thus, there is an optimization problem in choosing  $N$ : if  $N$  is too large, there is the computational cost of making more projections than are necessary, whereas if  $N$  is too small than the failure rate is too high requiring the additional cost of starting over. The optimal value depends on the computation costs of the projection as well as the costs of running the simulation.

With this in mind, the method has been enlarged to include an additional way to suppress gauge drift, an approach developed in ref. [86]. we show that a mixed approach using both quantum Zeno and the the approach of ref. [86] is likely to be more efficient than either one separately.

The method of ref. [86] is essentially an alternative version of the energy penalty method for suppressing coherent gauge drift. Here, we define coherent gauge drift as a unitary process  $U_D$  that transforms a physical state  $|\psi\rangle$  to

$$U_D |\psi\rangle = \sqrt{1 - \epsilon^2} |\psi\rangle + \epsilon |\omega\rangle \tag{B.12}$$

where  $|\omega\rangle$  is a nonphysical state and  $\epsilon$  characterizes the strength of the drift. Recall that an energy

penalty method constructs a Hamiltonian  $H_G$  with the behavior that  $H_G |\psi\rangle = 0$  for physical states  $|\psi\rangle$  and  $H_G |\omega\rangle > 0$  for unphysical states  $|\omega\rangle$ . Adding  $\lambda_G H_G$  for  $\lambda_G \gg 0$  to the Hamiltonian of the system suppresses transitions to unphysical states via an energy penalty. The authors of [86] note that in principle, building  $H_G$  out of the fundamental fields of the lattice gauge theory would be quite difficult for a non-Abelian theory; however, quantum simulation via Trotterized time steps depends upon  $e^{-iH_G t}$  not solely  $H_G$ . Furthermore, they point out that this operator basically performs a random gauge transformation on the state of the system, which leaves physical states unchanged while unphysical states take on a phase. Thus, this approach dispenses with the difficult task of building  $H_G$  and instead simply implements the principal effect of including  $H_G$  via a random gauge transformation. In summary, the method of [86] is to implement, after every Trotterization time step, a random gauge transformation.

Our method aims to leverage this method wholesale: frequent projections back to the physical subspace is expensive, so we include a random gauge transformation after each Trotterization time step. This reduces the gauge drift per time step, allowing us to do projections to the physical subspace less frequently while still staying close enough to the physical subspace to leverage the quantum Zeno effect, which allows the suppression of whatever gauge drift that survives the random gauge transformations.

### B.2.3 Calibration

Our method, as outlined above, is to conduct a random gauge transformation after every Trotterization time step as well as to conduct projections in the form discussed above to project back to the physical subspace. One important detail that needs to be specified is the frequency of

projections. To this end, we include a calibration phase in the method—a set of initial studies—to best estimate the frequency of projections that constitutes a useful compromise between the minimization of gauge drift and the minimization of the number of projections. We note up front that in principle doing a projection after every time step would be most effective in terms of reducing gauge drift as the probability of failure of the projection is very low. The problem with this is simply that it is extremely expensive computationally. In the next section we will outline results from a toy model that show that other, less expensive forms of the method work well. This suggests that realistic calculations can balance the resource costs with the requirements on effective gauge drift suppression.

We propose an approach that should be useful in situations when a simulation needs to be done repeatedly—in that case, a simple calibration phase to optimize the calculation is done at the outset: start with an initial trial frequency of projections based on whatever information and intuition one has about the system. Here, any previous knowledge about the system being simulated should be exploited, especially concerning the speed of gauge drift. With an initial frequency chosen, the time evolution of the system is run a small number of times with the chosen trial frequency of projections to see how long before the projection step fails. If it appears that failure will in general occur before the desired length of time of the simulation, the frequency of projections can be increased. This process can be repeated until an acceptable balance is found.

Behind this simple calibration phase is the fact that the average gauge drift is linearly proportional to the number of time steps between projections. Intuitively, this also means the probability of a failed projection is linearly proportional to the number of time steps between projections. With these relations in mind, it is simple to tune the frequency of projections given a relatively few trial runs.

### B.3 Performance

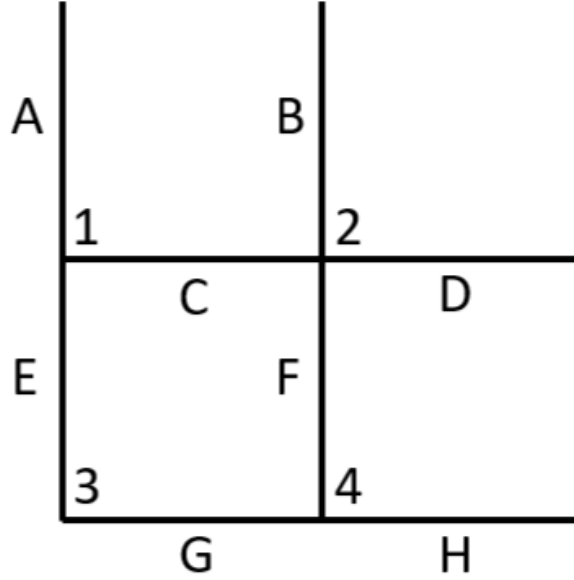


Figure B.2: The diagram of the toy model with 4 sites and 8 links

To study our algorithm, we consider a small “toy” model. It is a pure  $\mathbb{Z}_2$  gauge theory in 2 spatial dimensions [145]. We simulate this on a 2x2 lattice with periodic boundary conditions (see Figure B.2). The purpose of using such a small simulation is simply that it allows one to explore the method with very little cost. The Hamiltonian for the toy model is

$$H = -g \sum_{i=A}^H \sigma_i^x - k \sum_{j=i}^4 \square_j \quad (\text{B.13})$$

where  $g$  and  $k$  are parameters,  $\sigma_i^x$  is the Pauli x operator on the  $i$ th link, and  $\square_j$  is the  $j$ th plaquette operator (where the  $j$ th plaquette is defined as the plaquette with the  $i$ th site in the bottom left corner). A plaquette operator in this context is a product of the  $\sigma^z$  operators living on the four

sides of a plaquette in the lattice; for example, the first plaquette operator is

$$\square_1 = \sigma_A^z \sigma_B^z \sigma_C^z \sigma_G^z \quad (\text{B.14})$$

In this system, each site has an associated local gauge transformation  $g_i$ , constructed by taking the product of the  $\sigma^x$  operators living on the four links emanating from the  $i$ th site on the lattice:

$$g_1 = \sigma_A^x \sigma_C^x \sigma_D^x \sigma_E^x \quad (\text{B.15})$$

$$g_2 = \sigma_B^x \sigma_C^x \sigma_D^x \sigma_F^x \quad (\text{B.16})$$

$$g_3 = \sigma_A^x \sigma_E^x \sigma_G^x \sigma_H^x \quad (\text{B.17})$$

$$g_4 = \sigma_B^x \sigma_F^x \sigma_G^x \sigma_H^x \quad (\text{B.18})$$

We note here that our gauge group  $\mathbb{Z}_2$  is discrete, which means we cannot formulate Gauss' law via generators of gauge transformations. Instead, we note that the Hilbert space of the system is split into sectors defined by the eigenvalues of the gauge transformations in equations (15)-(18). Note that each  $g_i$  has eigenvalues  $\pm 1$ . With this, we define a physical state as a state  $|\psi\rangle$  that is unchanged by all local gauge transformations:

$$g_i |\psi\rangle = |\psi\rangle \quad \forall i \quad (\text{B.19})$$

At this point we will note that in previous sections we described our algorithm using a  $G^2$  operator (defined in equation (10)), but this was for systems governed by compact Lie gauge groups. Here,

we have the discrete group  $\mathbb{Z}_2$ , which requires two small alterations. Firstly, we replace the  $G^2$  operator with a new operator  $G_{tot}$ :

$$G_{tot} = g_1 + g_2 + g_3 + g_4 \quad (\text{B.20})$$

We note that the spectrum of  $G_{tot}$  is  $-4, 0$ , and  $4$ ; furthermore, using equations (19) and (20), we have that

$$G_{tot} |\psi\rangle = 4 |\psi\rangle \quad (\text{B.21})$$

for all physical states. This  $G_{tot}$  operator serves the same purpose as the  $G^2$  operator: every physical state is an eigenstate with the same eigenvalue ( $4$  for  $G_{tot}$  and  $0$  for  $G^2$ ) and all unphysical states have different eigenvalues. Thus,  $G_{tot}$  will function the same in our algorithm for our discrete group as  $G^2$  will for compact Lie gauge groups.

Here, we must highlight the second alteration needed for our discrete group. Figure [B.1](#) shows phase rotation gates as part of the circuit diagram, but we argued that they are not required in our algorithm in section IIA due to physical states having an eigenvalue of  $0$ . As just discussed, this is not true for our toy model, so we must include the phase rotation gates with  $E = 4$ , the eigenvalue of physical states for our system.

Now, our aim is to study gauge drift, so we include in our toy model an artificial gauge drift operator  $D$  with each time step. We base our constructed gauge drift operators off of eq. [\(B.12\)](#); specifically, we connect each of the, in our case  $32$ , physical states of the system to a randomly selected unphysical state in exactly the way laid out in eq. [\(B.12\)](#). All other states are unchanged.

This provides a way for the system to drift out of the physical subspace of its Hilbert space in a unitary way, and a parameter,  $\epsilon$ , to control the strength of this drift. For our simulations, we set this parameter to  $\epsilon = 0.01$ . Note that each time our method calls for time evolution, we select a different set of 32 unphysical states to appear in the gauge drift operator so as to not bias any unphysical states.

With this defined, the algorithm can be used for the toy model. Each time step takes a state  $|\psi\rangle$  to  $\tilde{g}_i D U_t |\psi\rangle$  where  $\tilde{g}_i$  is a randomly chosen gauge transformation,  $D$  is a gauge drift operator, and  $U_t = \exp\{-iHt\}$  is the standard time evolution operator. Note that  $\tilde{g}_i$  represents any gauge transformation on the system, including the  $g_i$ 's in equations (15)-(18) as well as any product of them. The remaining part of the algorithm is the rodeo projection, which occurs after a set number of time steps (determined in the calibration phase). As discussed above, a successful projection takes a state  $|\psi\rangle$  to  $\frac{1}{2}(1 + \exp(-i(G_{tot} - 4)t_r))|\psi\rangle$  where here we note that the time used within the projection  $t_r$  is in general different from the time step size  $t$ . Recall that all physical states are eigenstates of  $G_{tot}$  with an eigenvalue of 4, so they are unaffected while unphysical states will suffer destructive interference.

Figure B.3 demonstrates the basic performance of the method. It shows no suppression, a method of using a random gauge transformation after each time step, and our method of using a random gauge transformation after each time step plus projections, in this case after every time step. It can be seen both that the use of random gauge transformations after every time step greatly suppresses the gauge drift as well as that frequent rodeo projections does in fact keep the system very close to the physical subspace.

In the rest of this section we will consider the frequency of projections. To study this, we have run simulations to test many different regimes to compare their effectiveness. In particular,

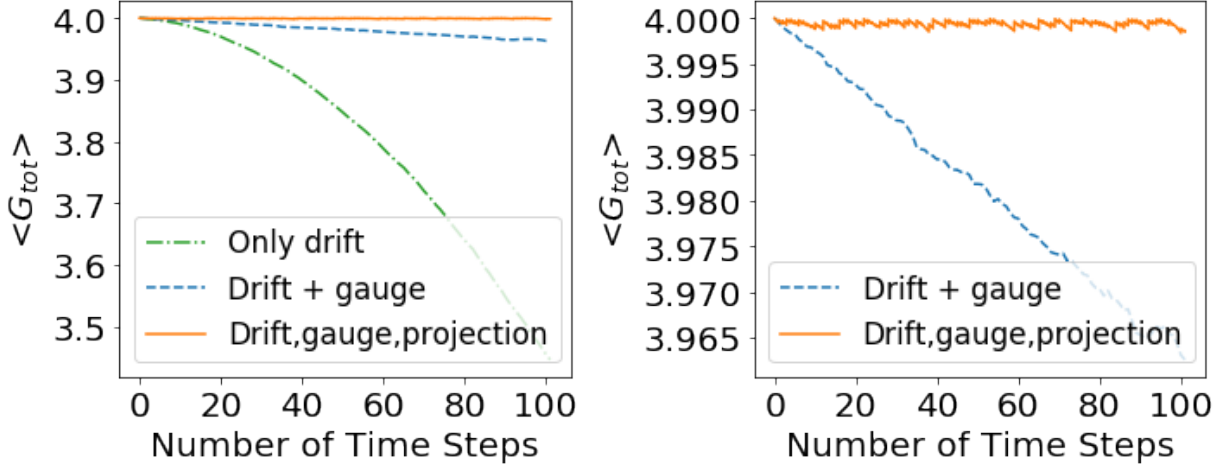


Figure B.3: Comparing gauge drift suppression methods, including no suppression methods (dash-dot line), a random gauge transformation after each time step (dashed line), and a random gauge transformation followed by a projection after each time step (solid line). The right plot is the same as the left plot without the dash-dot plot so as to zoom in on the behavior of the other two.

systems without any gauge transformations after time steps with a frequency of rodeo projections ranging from after one time step to after 10 time steps. We also considered systems with a random gauge transformation after each time step with a frequency of gauge projections ranging from after one time step to after 30 time steps. For each regime, we ran our algorithm on 200 systems and calculated the average of the farthest from physical each run reached (which is to say the lowest value of  $\langle G_{tot} \rangle$ ).

Figure B.4 shows these results. It is clear that including the random gauge transformations into the method significantly helps to reduce the required frequency of gauge projections. In particular, it shows the following approximate equivalences: a projection after every time step performs similarly to a projection every two time steps when gauge transformations are included; a rodeo projection after every two time steps performs similarly to a gauge projection every five time steps when gauge transformations are included; and a gauge projection after every three time steps performs similarly to a gauge projection every ten time steps when gauge transformations

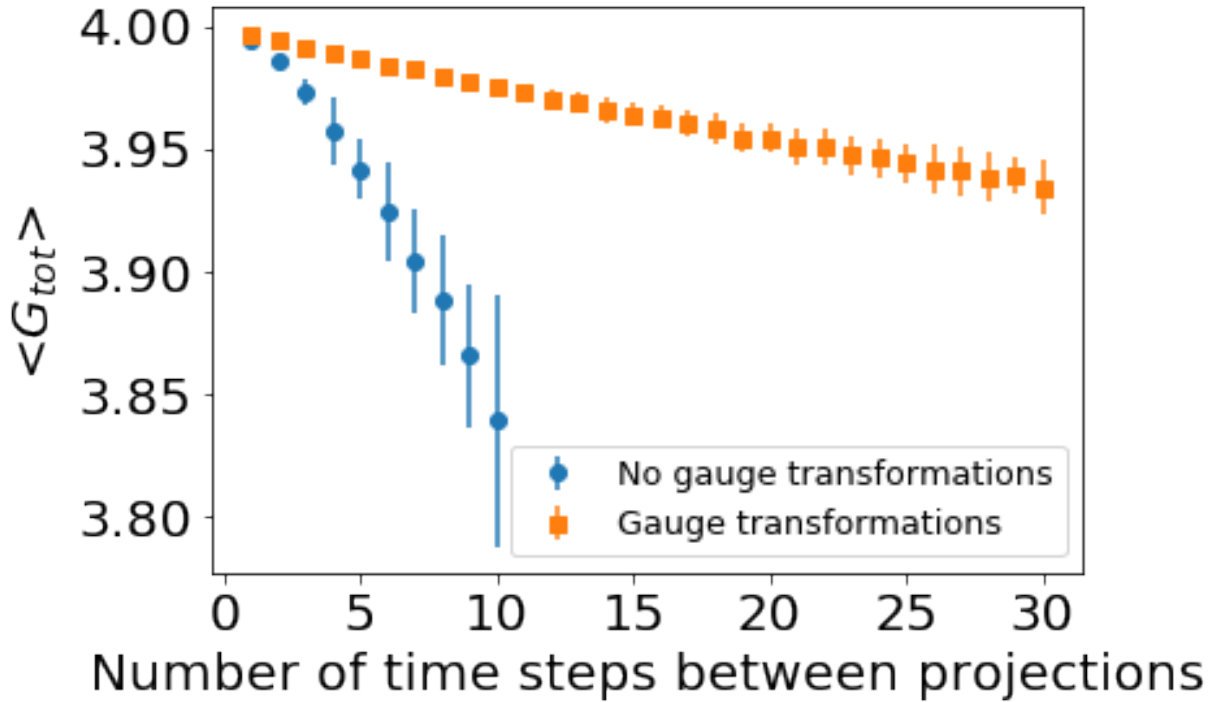


Figure B.4: Comparing performance of regimes with no gauge transformations and various frequencies of projections (circle markers) with regimes with gauge transformations and various frequencies of projections (square markers)

are included. This shows that including the gauge transformations reduces the required frequency of rodeo projections by more than half. In particular, we see that for a system that drifts out of the physical subspace slowly, one could perform projections infrequently and still be able to utilize the Zeno effect to great advantage.

## B.4 Implementation

It is important to consider the implementation of this method, as well as the associated costs. As our method involves two main operations (rodeo projections and gauge transformations), we will take them in turn.

### B.4.1 Implementing Rodeo Projections

With a circuit diagram of Fig. B.1 plus the small alterations discussed in section IIA in hand, there remain two main questions when it comes to the implementation of our version of rodeo projections: how should the time parameters, call them  $t_r$ , be chosen, and how many ancillary qubits per projection are required?

Firstly, we address the issue of the selection of the  $t_r$  parameters. Here, we note that previous work has shown that the rodeo algorithm can be optimized and made far more efficient [38] via considered choices for the parameter  $t_r$ . We suggest that this optimized approach makes sense for our purposes as well. This work notes that the dominant cost of the rodeo algorithm is the controlled time evolution gate. In our case it is then the controlled U gate with  $U = \exp(-iG^2t_r)$ . We will also note that constructing  $G^2$  will be much more difficult than  $H$  so our version of a rodeo projection suffers a much heftier cost in construction of the required gates, but so will alternative projection schemes. Due to the controlled U gate dominating costs, the authors of [38] argue that the total time to conduct a rodeo projection acts as a proxy for computational costs and then proceeds to lay out a method for selecting the  $t_r$ s that maximizes efficiency of the algorithm given fixed computational resources [38]. The authors note that choosing the  $t_r$ s randomly (eg via Gaussian distributions) can lead to exponentially large fluctuations in the suppression of undesired amplitudes; furthermore, they show that considered choices of the  $t_r$ s, specifically choosing  $t_r$ s that vary across exponentially many timescales, can eliminate the problem of these large fluctuations, thus leading to a more efficient implementation of the rodeo projection.

Additionally, we must address the question of qubit costs. Notably, if mid-circuit measurements are allowed, then only one ancillary qubit is required, as it can be reused again and again.

If this is not the case, then we estimate one projection would need roughly as many ancillary qubits as sites in the lattice. Future work aims to study how the qubit per projection requirement scales with system size.

## B.4.2 Implementing Gauge Transformations

It is important to discuss the practical implementation of random gauge transformations as [86] outlines in section III of their paper. First, we must sample from the Lie group  $G$  of gauge transformations. One way is to construct an ancillary 'clock' register that increases its value by 1 after each gauge transformation is conducted in the course of our algorithm. Then a random circuit (eg of the kind developed in [146]) will take the value of the clock and select a gauge transformation for each site and store it on an ancillary  $G$ -register. Once sampled, these gauge transformations can be performed on the system.

The first major cost consideration of this process is qubit costs. The sampling process requires two additional registers (the clock and the  $G$ -registers); notably, this qubit cost does not scale with the size of the system. Furthermore, the actual implementation of the gauge transformations on the system "requires about as many  $G$ -multiplication gates as the original quantum simulation did" [86].

A second cost consideration is the depth of the random circuit. Intuitively, shorter circuits will result in less fair sampling from the Haar measure of the Lie group than longer circuits. Luckily, we are helped significantly by the fact that we have no strict requirements on fair sampling from the Haar measure. Here, we point to a common method of sampling uniformly from the Haar measure of a group  $G$ , which is to select a small number (eg 2) of elements of  $G$ , call

them  $g_1$  and  $g_2$ , and construct strings of these elements of a chosen length (an example of a string of length 5 is  $g_1g_2g_2g_1g_2$ ). Importantly, this method of sampling elements of  $G$  converges to the Haar measure in the limit of long strings. This tells us that unfair sampling can be made more fair by conducting multiple unfairly sampled gauge transformations. Thus, we can use a short random circuit when sampling gauge transformations and then perform multiple gauge transformations in a row if need be to increase the fairness of the sampling. Future work aims to clarify the relationship between the fairness of sampling and the performance of the algorithm.

## B.5 Conclusion

We have outlined an algorithm developed to maintain gauge invariance (or more precisely suppress gauge drift) during the simulation of lattice gauge theories. Our algorithm was based off of the Zeno effect, wherein we use frequent projections (i.e. altered rodeo projections) to keep the system within the physical subspace of its exponentially larger Hilbert space. We also include a technique of performing gauge transformations after every time step to help reduce the system's gauge drift into the unphysical sector which in turn allows for less frequent projections while still taking full advantage of the Zeno effect. We have shown that our method is effective for a simple small toy model of pure  $\mathbb{Z}_2$  gauge theory. Future work aims to test our algorithm out on larger systems, and more complicated gauge theories such as  $SU(N)$  or gauge theories that include matter.

## Acknowledgments

This work was supported in part by the U.S. Department of Energy, Office of Nuclear Physics under Award Number(s) DE-SC0021143 and DE-FG02-93ER40762.

Appendix C: Suppressing gauge drift in quantum simulations  
with gauge transformations

*Carter Ball*

**NOTA BENE: What follows is an exact reprint of the 2024 article “Suppressing gauge drift in quantum simulations with gauge transformations” by Carter Ball, originally posted to ArXiv [7] and currently under review at Physical Review A .**

Abstract

The simulation of quantum lattice gauge theories faces the major challenge of maintaining gauge invariance, as various errors in the simulation push the state of the system out of the physical subspace of the system’s exponentially larger Hilbert space. This paper outlines a method, based off of previous work [6], that uses gauge transformations in two ways. Firstly, the method exploits the Zeno effect by conducting frequent projections to suppress gauge drift. These projections utilize local gauge transformations to destructively interfere unphysical amplitudes via coupling to an ancillary qubit while the physical amplitudes are left untouched, up to a less than unity normalization factor. Secondly, gauge transformations are conducted throughout the time evolution of the system to hamper the speed of gauge drift. This paper demonstrates this method on a pure 1D  $SU(2)$  toy model.

## C.1 Introduction

Lattice gauge theories [137, 147] are vital to the study of high energy physics, especially the important theories of quantum electrodynamics (QED), quantum chromodynamics (QCD), and the Standard Model. A common approach to lattice gauge theories is to use a Hamiltonian formulation, in which space is placed on a discrete lattice while time is left continuous [92, 137, 138]. On this spatial lattice, matter fields live on the sites while gauge fields live on the links between sites. Time is kept continuous by taking the continuous time limit and employing the temporal gauge [139], a partial gauge that requires the time component of all gauge fields to be zero. A gauge in this context is a redundancy of degrees of freedom in the Lagrangian of a system.

As the temporal gauge is a partial gauge, it does not fully control this redundancy; thus, there is still gauge freedom present in the spatial components of the fields. This gauge freedom is governed by a set of equations called Gauss laws, in analogy to the gauge freedoms of electromagnetic potentials in electrodynamics [148]. These Gauss laws define Gauss law operators that generate local gauge transformations at each site of the lattice. An important aspect of these lattice gauge theories is that the subspace of physically relevant states, those that are unchanged by all local gauge transformations [149], is exponentially smaller than the total system Hilbert space. Thus, when simulating these theories, one must contend with the maintenance of gauge invariance, i.e. active consideration must be paid to keeping the system within the exponentially small subspace of gauge invariant, or physical, states. This is particularly difficult for non-abelian lattice gauge theories, such as QCD which is governed by the non-abelian gauge group  $SU(3)$ .

The simulation of lattice gauge theories [73, 122–127, 129–136] is a major topic of re-

search, with great potential to understand the theories underlying the structure of the physical world. These simulations contend with multiple big hurdles, including the maintenance of gauge invariance as well as the limits of classical computation. In addressing the second hurdle, much work has gone into the development of quantum computers in the hopes of this new form of computation having the power to greatly increase the productivity of quantum simulations of lattice gauge theories [8, 12, 13, 117–120]. To that end, there are many quantum algorithms being developed to conduct these quantum simulations.

Particular to the topic of this paper, many methods have been researched to tackle the issue of maintaining gauge invariance during quantum simulations. While in theory, if a system is initiated in a gauge invariant state and then only undergoes gauge invariant operations, then it will remain gauge invariant for the duration of the simulation. Perhaps the most important operation for simulation is time evolution; crucially, methods for gauge invariant time evolution have been developed [14, 126]. Unfortunately, this is often an unattainable ideal, as errors in the gauge invariance of the system's state can develop in a variety of ways, including due to necessary approximations of operations as well as quantum noise and gate errors.

One form of quantum simulation is called analog simulation [121], which sidesteps the difficulties of building and manipulating the system of interest by instead building a much more workable physical system that behaves like the system of interest at least in some reachable regime. Methods for analog simulations to maintain gauge invariance include satisfying gauge invariance automatically by tying it to an internal symmetry such as angular momentum [55, 56] or adding an energy penalty, effecting all unphysical states, to the system's Hamiltonian. The idea behind the energy penalty method is to craft a range of energies of the system that contains all of the physical states and no unphysical ones. There are multiple variations to this method, including

adding to the Hamiltonian a term proportional to the square of the Gauss law operators [54, 59–67], adding to the Hamiltonian a term directly proportional to the Gauss law operators [68], and a technique similar to those above but replacing the Gauss law operators with much cheaper local pseudogenerators [69–71]. Relevant to this paper’s method, this local pseudogenerator method points out that term added to the Hamiltonian, when constructed of local pseudogenerators, can act as a strong projector leveraging the quantum Zeno effect, at least within some timescale, to limit the system to a quantum Zeno subspace, ie the physical subspace [71, 140].

The method laid out in this paper concerns a different form of quantum simulation, called digital simulation [121], where qubits and quantum gates are used to simulate states of the system as well as operations on these states. While some methods take care to only simulate physical states, employing techniques such as solving the Gauss laws [72–79, 150] or using a dual formulation [141–143], this paper focuses on methods that simulate the full Hilbert space and then propose an algorithm to run within the simulation that restrain the system’s state from drifting outside of the physical subspace. Such methods include living with the errors under the premise that they are reduced for smaller timesteps [128], a quantum control theory method of dynamical decoupling [80], using an oracle to check for violations of gauge invariance [81], implementing a dynamical post-selection protocol using ancilla qubits coupled to local gauge operators [151, 152], and using classical noise [82]. This last method of classical noise relies on the continuous Zeno effect to constrain the gauge drift. This is in contrast to the method to be outlined in this paper, that relies on the standard quantum Zeno effect [83] of frequent measurement as a method of curtailing drift.

This paper is organized as follows. Section II outlines the method of utilizing gauge transformations in two ways to suppress gauge drift. Section III exhibits the method on a pure four-site

1D SU(2) toy model with periodic boundary conditions. Section IV discusses the implementation of the method, specifically the sampling and implementation of the gauge transformations as well as any costs associated with the method. Section V concludes with a short summary and points to potential future work.

## C.2 Method

For non-abelian gauge theories governed by a compact Lie gauge group, a gauge invariant, or physical, state is defined by the following set of generator equations:

$$G_x^a |\psi\rangle = 0 \quad \forall a, x \quad (\text{C.1})$$

where  $|\psi\rangle$  is a physical state and  $G_x^a$  is the  $a$ -th generator of local gauge transformations at site  $x$ . For systems governed by unitary gauge groups, such as  $U(N)$  and  $SU(N)$ , a local gauge transformation at a site  $x$  can be written as

$$g_x(\alpha_x^a) = e^{i \sum_a \alpha_x^a G_x^a} \quad (\text{C.2})$$

for some set of constants  $\alpha_x^a$ . Combining equations (1) and (2) gives a useful second definition of a physical state as a state that is unchanged by all local gauge transformations:

$$g_x(\alpha_x^a) |\psi\rangle = |\psi\rangle \quad \forall x, \alpha_x^a \quad (\text{C.3})$$

The idea behind this paper is to leverage this relationship between physical states and gauge

transformations to suppress gauge drift during the course of quantum simulations. The key insight that motivates the method of this paper concerns the action of operators of the type  $e^{-it\tilde{G}}$  where  $\tilde{G}$  is some operator constructed out of the gauge generators  $G_x^a$ . While this kind of operator can be used in gauge drift suppression schemes, as discussed below, they are expensive to construct. The function of these operators is to leave physical states untouched, as per equation (1), while affecting unphysical states in a way that reduces their prevalence in the system's overall state. The key insight of this paper, as pointed out by Lamm, Lawrence, and Yamauchi in their paper Ref. [86], is that a gauge transformation does this exact thing: leaves physical states unchanged while acting on unphysical states in some way. Thus the method of this paper, designed for quantum digital simulations employing Trotterized time evolution, is to combine two methods that use  $e^{-it\tilde{G}}$  operators for gauge drift suppression but instead use gauge transformations, which are less expensive to construct on the lattice.

The first method this paper considers is the one laid out in Ref. [86], which builds their method as an alternative to the established energy penalty method of Refs. [54, 59–67]. This method adds an energy penalty term  $H_G$  to the system Hamiltonian  $H_0$  such that  $H_G|\psi\rangle = 0$  for every physical state and  $\|H_G|\omega\rangle\| \gg 0$  for every unphysical state. Then, the system time evolves under the unitary  $e^{-it(H_0+H_G)}$ . As the operator  $H_G$  is constructed out of gauge generators for it to act as an energy penalty, the additional term to the time evolution operator  $e^{-itH_G}$  is the kind of operator this paper is interested in.

The authors of Ref. [86] note that for non-abelian theories, the construction of  $H_G$  is difficult; thus, they propose a method to get around this by considering the effect of including the  $e^{-itH_G}$  term in the time evolution operator. They point out that this term does not affect physical states but adds a phase to unphysical states, and that this is analogous to performing a gauge

transformation on the system. Thus, their method dispenses with the additional energy penalty term  $H_G$  in the Hamiltonian and instead conducts a gauge transformation after every Trotterized time step. Their paper shows that this method is effective at slowing down the speed of gauge drift, but it does not fully stop it.

The method of this paper incorporates this first method into a second method to be outlined below, as they are nicely complementary. Specifically, the method below relies on the Zeno effect and thus frequent projections, so it is beneficial to slow down the speed of the gauge drift using this method of [86] to reduce the required frequency of projections.

The second method this paper considers was developed previously by the author and their collaborator [6], which uses the operator  $e^{-itG^2}$ , where  $G^2 = \sum_{a,x} (G_x^a)^2$ , within a projection for gauge drift suppression. Below, this paper outlines this method with the crucial replacement of the  $e^{-itG^2}$  operator with a random gauge transformation.

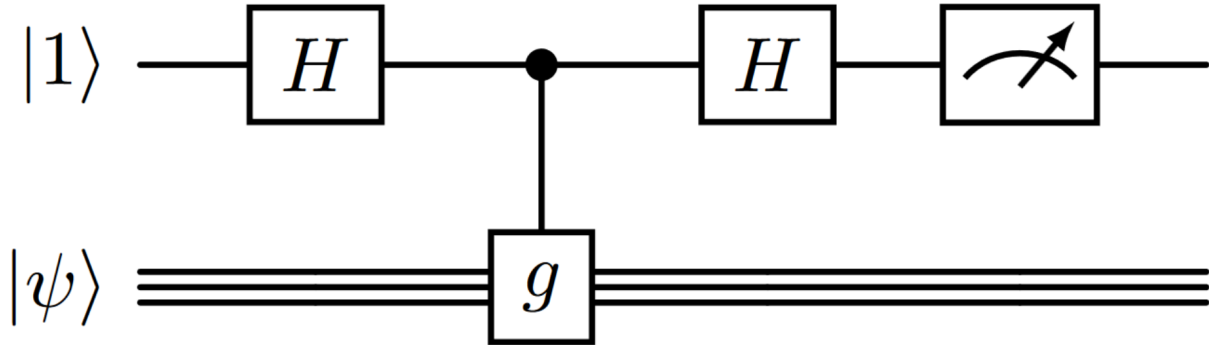


Figure C.1: Circuit diagram of the projection used to suppress gauge drift

The idea behind this method is to leverage the Zeno effect to suppress gauge drift by conducting frequent measurements which utilize gauge transformations to destructively interfere unphysical amplitudes, thus effectively projecting onto the physical subspace. The projection at the heart of this method is diagrammed in Fig. C.1. Note that the form of this projection is heavily

inspired by the projections described by the rodeo algorithm [2, 36, 37], an algorithm designed to construct energy eigenvectors of a given Hamiltonian.

In the projection of Fig. C.1,  $g$  is defined as

$$g = \prod_x g_x \quad (\text{C.4})$$

Thus,  $g$  is a product of a local gauge transformation on every site  $x$ , where each  $g_x$ , defined by constants  $\alpha_x^a$ , is chosen at random; the selection of these gauge transformations will be discussed in Sec. IV. Note that since quantum gates must be unitary operations [153], these gauge transformations must be unitary as well.

This projection (minus the final measurement) is mathematically written out below:

$$H(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes g)H|1\rangle \otimes |\psi\rangle = \quad (\text{C.5})$$

$$H(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes g) \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |\psi\rangle = \quad (\text{C.6})$$

$$H \frac{1}{\sqrt{2}} (|0\rangle \otimes |\psi\rangle - |1\rangle \otimes g|\psi\rangle) = \quad (\text{C.7})$$

$$\frac{|0\rangle + |1\rangle}{2} \otimes |\psi\rangle - \frac{|0\rangle - |1\rangle}{2} \otimes |\psi\rangle = \quad (\text{C.8})$$

$$|0\rangle \otimes \frac{I-g}{2} |\psi\rangle + |1\rangle \otimes \frac{I+g}{2} |\psi\rangle \quad (\text{C.9})$$

where the action of the Hadamard gate  $H$  is understood to act only on the ancillary qubit.

This projection involves an ancillary qubit initialized to the  $|1\rangle$  state that is first hit with a Hadamard gate to give equal amplitude, up to a sign, to its  $|1\rangle$  and  $|0\rangle$  components. Then a controlled-U gate entangles the ancillary qubit to the system, acting the  $g$  gauge transformation on the system only for the  $|1\rangle$  component of the ancillary qubit. Then, another Hadamard gate

interferes the two components of the ancillary qubit, resulting in equations (8-9). Finally, the ancillary qubit is measured. A successful projection measures the qubit in the  $|1\rangle$  state, otherwise the projection is said to have failed, as will become clear below.

To elucidate the effect of this projection, decompose a state into its physical,  $|\psi_p\rangle$ , and unphysical,  $\epsilon|\omega\rangle$ , parts:  $|\psi\rangle = \sqrt{1-\epsilon}|\psi_p\rangle + \sqrt{\epsilon}|\omega\rangle$ . Then the two components of Eq. (9) can be better analyzed using Eq. (3):

$$\frac{I-g}{2}|\psi\rangle = \frac{\sqrt{1-\epsilon}}{2}(I-g)|\psi_p\rangle + \frac{\sqrt{\epsilon}}{2}(I-g)|\omega\rangle \quad (\text{C.10})$$

$$= \frac{\sqrt{\epsilon}}{2}(I-g)|\omega\rangle \quad (\text{C.11})$$

$$\frac{I+g}{2}|\psi\rangle = \frac{\sqrt{1-\epsilon}}{2}(I+g)|\psi_p\rangle + \frac{\sqrt{\epsilon}}{2}(I+g)|\omega\rangle \quad (\text{C.12})$$

$$= \sqrt{1-\epsilon}|\psi_p\rangle + \frac{\sqrt{\epsilon}}{2}(I+g)|\omega\rangle \quad (\text{C.13})$$

Thus, the  $|0\rangle$  component of the ancillary qubit contains only unphysical parts of  $|\psi\rangle$  while the  $|1\rangle$  component contains all of the physical parts of  $|\psi\rangle$  and only a fraction of the unphysical parts. This is why measuring the ancillary qubit in the  $|1\rangle$  state is considered success and measuring in the  $|0\rangle$  state is considered failure.

To look deeper into the suppression of unphysical states in equation (13), write the unphysical component  $|\omega\rangle$  in the basis of the unitary operator  $g$ :

$$|\omega\rangle = \sum_n c_n |\omega_n\rangle \quad (\text{C.14})$$

$$g|\omega\rangle = \sum_n c_n e^{i\theta_n} |\omega_n\rangle \quad (\text{C.15})$$

where  $|\omega_n\rangle$  are eigenstates of  $g$  with eigenvalues  $e^{i\theta_n}$ . Thus, a successful projection conducts the

following (unnormalized) transformation:

$$|\psi\rangle = \sqrt{1-\epsilon}|\psi_p\rangle + \sqrt{\epsilon}\sum_n c_n |\omega_n\rangle \quad (\text{C.16})$$

$$\rightarrow \sqrt{1-\epsilon}|\psi_p\rangle + \sqrt{\epsilon}\sum_n \frac{1}{2}c_n(1+e^{i\theta_n})|\omega_n\rangle \quad (\text{C.17})$$

Note that while the magnitude of the system's unphysical amplitude, characterized by  $\epsilon$ , affects a projection's probability of success, it does not affect the suppression mechanism of each unphysical state  $|\omega_n\rangle$  picking up a multiplicative factor  $\frac{1}{2}(1+e^{i\theta_n})$  (which has a magnitude  $\leq 1$ ). The average magnitude squared of this suppression factor is

$$\left\langle \left| \frac{1}{2}(1+e^{i\theta}) \right|^2 \right\rangle = \frac{1}{2\pi} \int_0^{2\pi} d\theta \left| \frac{1}{2}(1+e^{i\theta}) \right|^2 \quad (\text{C.18})$$

$$= \frac{1}{2} \quad (\text{C.19})$$

While *a priori* there is nothing to say that the  $\theta_n$  will always be uniformly distributed, as is the implicit assumption of Eq. (18), it is a good rule of thumb that a successful projection reduces the unphysical probability density by roughly a factor of  $\frac{1}{2}$ , as is borne out by tests run on a SU(2) toy model (see Sec. C.3).

With the projection of this method outlined, it remains to be discussed the role of the Zeno effect. To that end, let a system with Hamiltonian  $H$  start in the initial state  $|\psi_0\rangle$  and consider the initial state amplitude  $A(t)$  as well as the probability of finding the system in its initial state after

a time  $t$  [83]:

$$p(t) = |A(t)|^2 = \left| \langle \psi_0 | e^{-iHt} | \psi_0 \rangle \right|^2 \quad (\text{C.20})$$

Now, for short times  $e^{-iHt} \simeq 1 - iHt - \frac{1}{2}H^2t^2$ . Applying this expansion to the amplitude and probability gives, for short times,

$$A(t) \simeq 1 - it \langle \psi_0 | H | \psi_0 \rangle - \frac{1}{2}t^2 \langle \psi_0 | H^2 | \psi_0 \rangle \quad (\text{C.21})$$

$$p(t) \simeq 1 - t^2 \left( \langle \psi_0 | H^2 | \psi_0 \rangle - \langle \psi_0 | H | \psi_0 \rangle^2 \right) \quad (\text{C.22})$$

This is the crux of the quantum Zeno effect: for short times the initial state amplitude decreases linearly, to leading order, while the probability decreases quadratically due to the cancellation of linear terms. Thus, if the system is time evolved for only a short time and then measured, there is a high probability of finding the system in its initial state. Note here that while the above case deals with an individual state, the quantum Zeno effect works with subspaces as well. Thus, if frequent enough measurements are taken throughout the course of a simulation, the system will to good approximation remain within the desired state or subspace for the duration of the simulation. For the purposes of this paper the desired subspace is of course the physical subspace of the system's Hilbert space.

Previous work by the author and their collaborator [6] explicitly sketches this basic scenario out: consider a system that will evolve for a total time  $T$  during which  $N$  projections take place, spaced evenly. While the projections outlined above are not perfect, for the sake of this exercise they will be treated as such. Furthermore, say that  $r$  is the rate at which the system's

state accumulates amplitude outside of the desired space, i.e. unphysical amplitudes. Then the probability of a failed projection is  $(\frac{rT}{N})^2$ . Then the probability of the full simulation completing with 0 failures is

$$P_s = \left(1 - \frac{r^2 T^2}{N^2}\right)^N \approx 1 - \frac{r^2 T^2}{N} \quad (\text{C.23})$$

in the limit of large  $N$  and  $1 - P_s \gg 1$ . Notably, this is close to one when  $N \gtrsim r^2 T^2$ . Thus, frequent enough projections can keep the system within the physical subspace for the entirety of the simulation.

At this point, failed projections must be addressed. If a projection does fail, then the whole simulation must be restarted. Obviously, this comes at great cost and so should be avoided if at all possible. A first solution would be to do a projection after every single Trotterized time step. As this gives the system very little time to drift out of the physical subspace, this greatly suppresses the chance of a failed projection; however, this is costly. Therefore, an optimization problem balancing the cost of projections and the risk of a failed projection must be addressed.

Recall that the method of this paper has two components: one being the frequent projections to leverage the Zeno effect and the other being the use of gauge transformations after every Trotterized time step. As discussed above, this second component slows down the rate of the gauge drift, which in turn helps address this optimization problem since slower gauge drift reduces the necessary frequency projections.

Furthermore, a simple calibration phase can be used to tune the frequency of projections to an optimal value. This calibration phase is best used for simulations that plan to be run many times. This phase begins by selecting an initial frequency of projections based off of knowledge

about the system and the cost of projections. With the initial frequency chosen, run the simulation a handful of times till failure. With multiple runs of this test, the simulators should get a good idea of if the initial frequency of projections will work for the parameters of their simulation; if not, adjust the frequency accordingly and go again. In this way, a workable frequency of projections can be found fairly simply.

In summary, the method of this paper is to leverage the Zeno effect to suppress gauge drift by performing frequent projections (diagrammed in Fig. C.1) while also performing a random gauge transformation after every Trotterized time step of the simulation.

### C.3 Performance

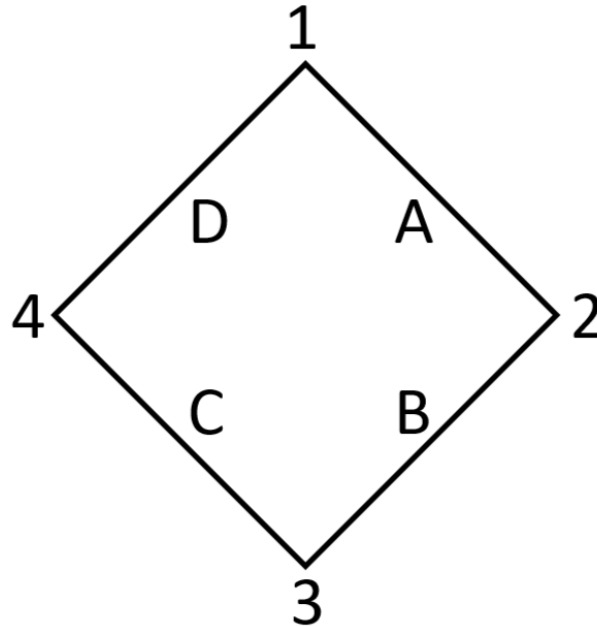


Figure C.2: The diagram of the 1D toy model with periodic boundary conditions, four sites (labelled 1-4), and four links (labelled A-D).

In order to see the method of this paper in action, consider a 1D pure  $SU(2)$  toy model with 4 sites, taking a Kogut-Susskind lattice approach. Periodic boundary conditions were used

to avoid large edge effects one might see for such a small system and instead focus on bulk behavior. Fig. C.2 diagrams this toy model, representing it as a diamond shape to clarify that this is a one-dimensional system with periodic boundary conditions instead of a singular plaquette.

The Hamiltonian for this toy model is

$$H = \lambda \sum_{i=A}^D E_i^2 \quad (\text{C.24})$$

where  $E_i^2$  is the quadratic Casimir operator of the  $i$ th link and  $\lambda$  is a constant, set to 1 for convenience. Note that the toy model is simulated in a representation basis with a cutoff  $j_{max} = 1/2$  imposed. This  $j$ -cutoff is an analogy to angular momentum and limits the allowed eigenvalues of the quadratic Casimir operator  $E_i^2$  which have the form  $j(j+1)$ . Thus, this cutoff limits the available energy levels of each link. This does not affect the work of this paper as limiting a system to its lower energy levels still allows for the study of gauge drift and its suppression.

Each site has 3 generators of local gauge transformations at that site; as an example

$$G_1^x = L_A^x + R_D^x \quad (\text{C.25})$$

$$G_1^y = L_A^y + R_D^y \quad (\text{C.26})$$

$$G_1^z = L_A^z + R_D^z \quad (\text{C.27})$$

where the superscripts  $x, y, z$  reference the three Pauli matrices and the  $L_i$  ( $R_i$ ) operator is the left (right) electric field operator on the  $i$ th link. This means a local gauge transformation at site

1 can be written

$$g_1 = e^{i(\alpha_1^x G_1^x + \alpha_1^y G_1^y + \alpha_1^z G_1^z)} \quad (\text{C.28})$$

$$= e^{i\vec{\alpha}_1 \cdot \vec{G}_1} \quad (\text{C.29})$$

where the  $\alpha_1^i$ 's are constants and we have defined  $\vec{G}_i = (G_i^x, G_i^y, G_i^z)$  for convenience. With this in hand, any gauge transformation on the whole system can then be written

$$g = g_1 g_2 g_3 g_4 = e^{i\vec{\alpha}_1 \cdot \vec{G}_1} e^{i\vec{\alpha}_2 \cdot \vec{G}_2} e^{i\vec{\alpha}_3 \cdot \vec{G}_3} e^{i\vec{\alpha}_4 \cdot \vec{G}_4} \quad (\text{C.30})$$

As the goal of this paper is to study the suppression of gauge drift, an artificial drift operator  $D$  is used to simulate gauge drift. For concreteness, it is helpful to label the toy model's two physical states as  $|0\rangle$  and  $|1\rangle$  and all other, unphysical states  $|\omega_i\rangle$ . The idea behind the drift operator is to unitarily connect each physical state to a randomly selected unphysical state; call them  $|\omega_a\rangle$  and  $|\omega_b\rangle$ . Thus, the drift operator was constructed as following:

$$\begin{aligned} D(\epsilon, \omega_a, \omega_b) &= \sum_{i \neq a, b} |\omega_i\rangle \langle \omega_i| + \\ &+ \sqrt{1 - \epsilon^2} \left( |0\rangle \langle 0| + |1\rangle \langle 1| + |\omega_a\rangle \langle \omega_a| + |\omega_b\rangle \langle \omega_b| \right) + \\ &+ \epsilon \left( |0\rangle \langle \omega_a| - |\omega_a\rangle \langle 0| + |1\rangle \langle \omega_b| - |\omega_b\rangle \langle 1| \right) \end{aligned} \quad (\text{C.31})$$

where  $\epsilon$  parametrizes the strength of the drift operator.

As can be seen, the drift operator exchanges amplitude between  $|0\rangle$  and  $|\omega_a\rangle$  as well as

between  $|1\rangle$  and  $|\omega_b\rangle$ ; for the mostly physical states that this drift operator will act on, this means the drift operator reduces the physical probability density and increases the unphysical probability density. When it comes to implementation, the drift operator is used on the toy model’s state right before every time evolution. Crucially, for every instance of the drift operator, new unphysical states are randomly selected so as to not preference any in particular.

The final aspect of studying the suppression of gauge drift is to quantify the “physicalness”, or equivalently “unphysicalness”, of a given state. This can be done with the following operator:

$$G^2 = \sum_{a,x} (G_x^a)^2 \quad (\text{C.32})$$

This operator, discussed above in Sec. II as well as in Ref. [6], is defined such that the ground state eigenspace, with eigenvalue 0, is exactly all physical states and no unphysical states. Thus a given state  $|\psi(t)\rangle$  can be evaluated on its “physicalness” by considering the expectation value  $\langle\psi(t)|G^2|\psi(t)\rangle$ . If this expectation value is 0, the state is fully physical; the further from 0 it gets the more unphysical the state is said to be. Note that by construction  $G^2$  is positive definite, so  $\langle G^2 \rangle \geq 0$  for any state.

The simulations run for this paper employed the Trotterization technique, wherein the system was time evolved by repeated use of the unitary  $e^{-iH\Delta t}D(\epsilon)$  for  $\Delta t = 0.01$  time units and  $\epsilon = 0.01$ . After each instance of this unitary there was an opportunity for a gauge drift mitigation technique to be used, such as a random gauge transformation for a projection; the choices concerning which method is used and how frequently they are used define a simulation’s overall mitigation scheme.

Every run of the algorithm laid out in this paper starts by initializing the system in a fully

physical state. As there are two physical states for this system (one where all links are  $j = 0$  and another where all links are  $j = 1/2$  with the appropriate  $m_j$  values at the ends), this initialization amounts to randomly generating an amplitude for one physical state, which then determines the amplitude (up to a phase, chosen to make the amplitude real and positive) of the other physical state, given the constraint that the initial state is normalized.

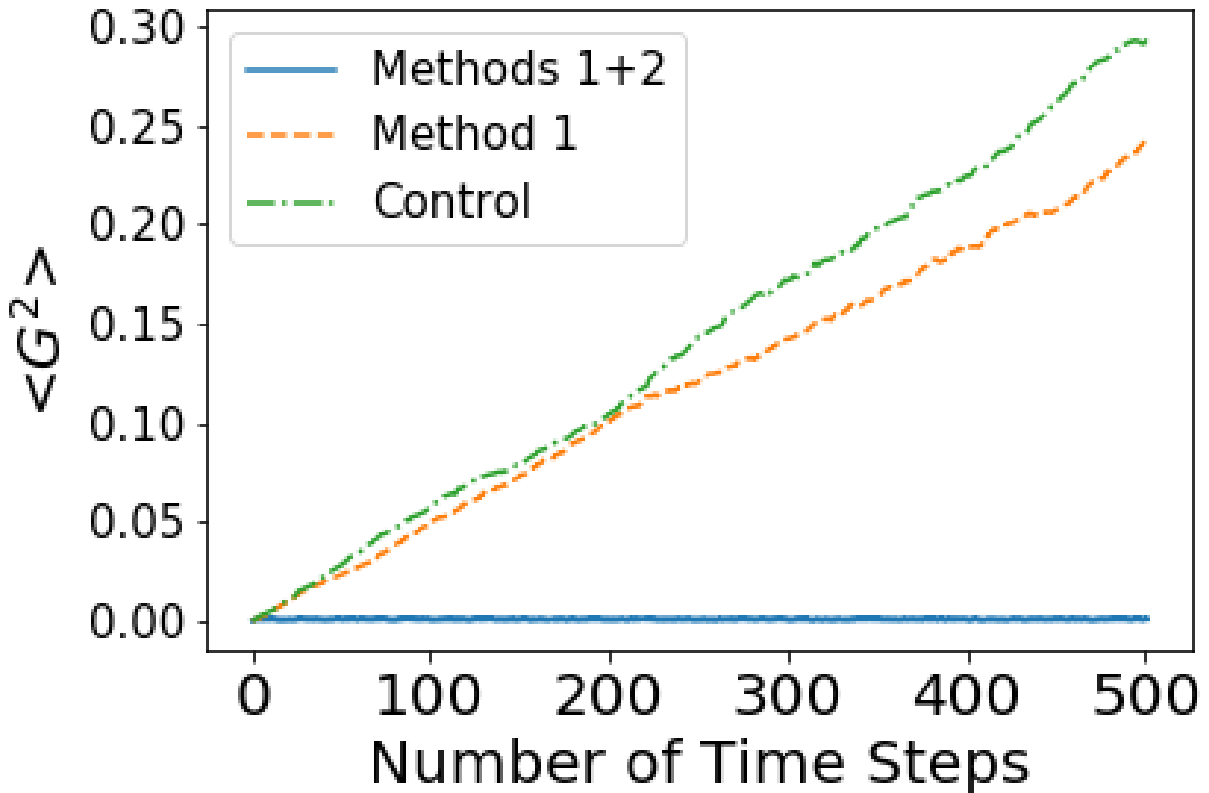


Figure C.3: Results from three simulation runs: a control run, dashed-dot line, without any gauge drift suppression methods; a run, dashed line, using method 1 (conducting a gauge transformation after every time step); and a run, solid line, using both method 1 and method 2 (conducting a projection after every time step).

As a first test, three simulations were run on a classical computer for 500 time steps each, calculating  $\langle G^2 \rangle$  after every time step. These three simulations were chosen to demonstrate the basic performances of the methods used; see Figure C.3. The first run, a control run, did not employ any suppression techniques after each Trotterized time step (dashed-dot line); the second

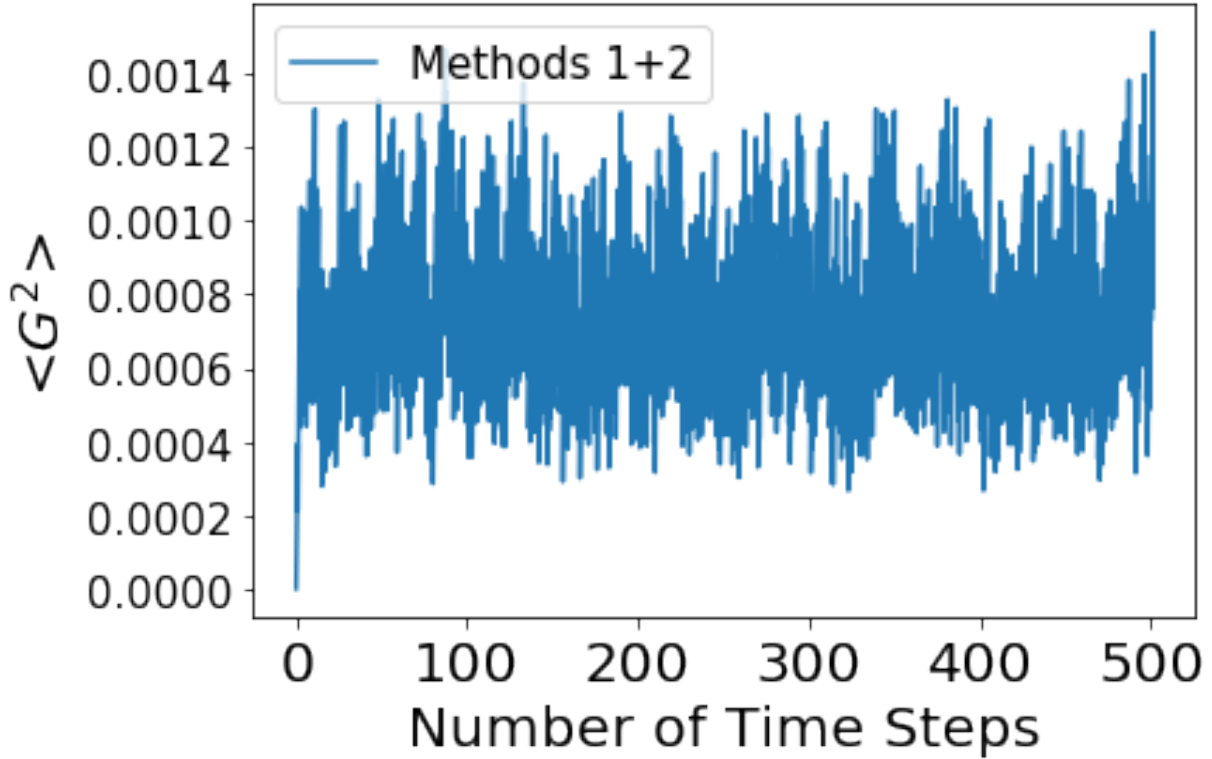


Figure C.4: A closer look at the run from Fig. C.3 utilizing both methods of gauge drift suppression.

run implemented the first method of gauge drift suppression discussed in this paper, i.e. the use of a random gauge transformation after each time step (dashed line); the third run used both this first method with gauge transformations as well as the second method of performing the projection diagrammed in Fig. C.1 after every time step (solid line). Figure C.4 plots only this third run to highlight its performance.

Figures C.3 and C.4 show that the method of this paper is quite effective at suppressing gauge drift for this toy model. As expected, leaving the system without any gauge drift suppression technique (control) allows for unphysical amplitudes to build up over time, represented by a growing  $\langle G^2 \rangle$  value. Furthermore, the use of method 1 slows down this growth but does not bound it. Finally, employing both methods bounds the growth of unphysical amplitudes to a

small region close to the physical subspace, represented by  $\langle G^2 \rangle = 0$ . Thus, Fig. C.4 shows that the system remains highly physical throughout its simulation when both methods of this paper are implemented.

Looking further into the method of this paper, it is useful, particularly during the calibration phase, to consider the average suppression factor of a projection. To that end, a test was run wherein the system was initialized to a random physical state 50 times and then allowed to run for 101 time steps. Every run was simulated with a projection conducted after each Trotterized time step; additionally, the expectation value  $\langle G^2 \rangle$  was calculated before and after each projection. A suppression factor for each projection was then calculated as  $\langle G^2 \rangle_{after} / \langle G^2 \rangle_{before}$ . The average of all 101 suppression factors from all 50 initializations (for a total of 5,050 suppression factors) was calculated to be  $0.512 \pm 0.063$ . Thus, the working estimate of a 50% suppression factor posited in Sec. II is borne out in the data.

Now, this method relies fundamentally on performing random gauge transformations frequently throughout the evolution of a system, which can be a very large number of gauge transformations that must be sampled and performed. Thus, it is worthwhile to explore the possibility of instead using a small set of gauge transformations repeatedly throughout a run of a simulation.

To that end, simulations were run with the reuse of gauge transformations to study how this might affect the effectiveness of the gauge suppression method. For the purposes of direct comparison, each of the subsequent simulations followed the same general simulation pattern, consisting of four steps:

1. Time step
2. Gauge transformation (method 1)

3. Time step

4. Projection (method 2)

Crucially, recall that a gauge transformation is used in step 4 as well as in step 2; see the projection's diagram in Fig. C.1. For each simulation, this pattern was repeated 250 times, for a total of 500 time steps. Furthermore,  $\langle G^2 \rangle$  was calculated after each of the four steps listed above.

With this basic pattern, four different methods of simulation were created. Each method of simulation was run 200 times, and then the  $\langle G^2 \rangle$  values from all 200 runs were averaged; these averages are shown in Table C.1. The first method, a control, randomly generated every gauge transformation used throughout the simulation, in both steps 2 and 4. For the remaining three methods of simulation, a simulation's initialization would include generating two gauge transformations, call them  $g_1$  and  $g_2$ , to be used repeatedly throughout said simulation. The second method of this test, called "Alternate time step  $g$ 's" in Table C.1, alternated between using  $g_1$  and using  $g_2$  for step 2, while the gauge transformations in step 4 were randomly generated. The third method, called "Alternate projection  $g$ 's" in Table C.1, did the opposite: the gauge transformations in step 2 were randomly generated while step 4 alternatively used  $g_1$  and  $g_2$ . The fourth method, called "Alternate all  $g$ 's" in Table C.1, used  $g_1$  and  $g_2$  as the only gauge transformations throughout the simulation; more specifically, the first step 2 used  $g_1$ , then the first step 4 used  $g_2$ , followed by the second step 2 using  $g_2$  and the second step 4 using  $g_1$ . This pattern repeated throughout each simulation.

Now, the method of this paper posits that the use of a singular gauge transformation is not going to provide good suppression for every unphysical state, but that one instance of a gauge transformation can provide good suppression cumulatively. Furthermore, this method poses that

Method	Average $\langle G^2 \rangle$ value
Control (fully random $g$ 's)	$(14.8 \pm 4.2) \times 10^{-4}$
Alternate time step $g$ 's	$(14.9 \pm 4.2) \times 10^{-4}$
Alternate projection $g$ 's	$(14.6 \pm 4.1) \times 10^{-4}$
Alternate all $g$ 's	$(17.2 \pm 4.3) \times 10^{-4}$

Table C.1: The average  $\langle G^2 \rangle$  value for four different methods of reusing gauge transformations throughout a simulation run

multiple different gauge transformations used throughout a simulation can provide strong suppression, the assumption being that if one gauge transformation fails to adequately suppress a certain unphysical state then the next gauge transformation(s) can pick up the slack. Thus, the method of randomly generating gauge transformations was used as the baseline method to avoid any suppression-affecting biases.

The results of Table C.1, however, show that this might not be too significant a concern. These results demonstrate, for this toy model, that reusing gauge transformations does not have an outsized effect on the performance of the method. Thus, while the toy model in this paper is quite small and trivial, this data provides evidence suggesting that it is unnecessary to fully randomly generate a new gauge transformation every time one is needed and that perhaps a less arduous implementation scheme, such as reusing a small number of gauge transformations, can be effective.

It should be noted that this concept of reusing gauge transformations does have a lower limit. Even for this toy model, using only one gauge transformation for the entirety of a simulation destroyed the efficacy of the method. At least for this toy model, this was because any given gauge transformation has at least some unphysical states with an eigenvalue close to 1, so they don't receive adequate gauge drift suppression to counter the drift coming from the drift

operator  $D$  used in conjunction with each time step. While this turned out to be the case for only using one gauge transformation, this did not damage the efficacy the case of using two gauge transformations throughout as one gauge transformation could cover for the weaknesses of the other.

## C.4 Implementation

The method of this paper relies heavily on sampling and performing gauge transformations on the system of interest. Thus, it is prudent to explicitly discuss the implementation of these actions. When it comes to sampling the gauge transformation, especially if gauge transformations are being reused as discussed above, it might be best to sample them on a classical computer and then hard-code them into the quantum circuit before beginning any simulation [86]. This runs into the risk of ‘weighing down’ the circuit with too much information, depending on the required number of gauge transformations to hard-code.

Another option, discussed in both [86] and [6], is to use an ancillary ‘clock’ register that simply counts the number of gauge transformations that have been performed during the simulation. This clock register feeds its value into a random circuit of the type in Ref. [146] that will output a gauge transformation, storing it in an ancillary G-register. This method clearly comes with qubit costs, namely the additional clock and G-registers. Note however that these costs do not scale with the size of the system.

Another cost of this method is the depth of the random circuit that generates gauge transformations. One thing to note here is that Sec. III shows the repeated use of only two gauge transformations was effective; this implies that the efficacy of this paper’s method is not highly

sensitive to the level of fairness of the gauge transformation sampling. Thus, a shorter random circuit, which intuitively samples less fairly than longer random circuits, can be used without really damaging the efficacy of the method. Furthermore, Ref. [86] argues that unfair sampling can be made more fair by conducting multiple unfairly sampled gauge transformations in a row. Thus, it is quite probable that a relatively short random circuit would suffice.

When it comes to the actual implementation of the gauge transformations on the system in the course of the simulation, this “requires about as many G-multiplication gates as the original quantum simulation did” [86].

One final point to discuss is the qubit cost of the projections discussed above. Crucially, note that each measurement only requires one ancillary qubit. Thus, when mid-circuit measurements are a feature of the simulation, only one ancillary qubit is required, to be used for each projection. If this is not the case, then the qubit cost for the full simulation is equal to the total number of measurements to be performed.

## C.5 Conclusion

This work outlines a method of suppressing gauge drift with gauge transformations during quantum simulation of unitary lattice gauge theories. This method uses frequent projections to leverage the Zeno effect to achieve this suppression; furthermore, it includes the use of gauge transformations in the course the time evolution of the system that hamper the gauge drift. This paper shows this method to be effective for a toy model of a four-site pure 1D  $SU(2)$  system with periodic boundary conditions. Future work aims to test the method out on larger systems as well as systems with matter fields. Furthermore, deeper research into the reuse of gauge

transformations and the required level of fairness of their sampling is warranted.

## Acknowledgments

Useful conversations with Thomas D. Cohen are gratefully acknowledged, as well as the work done with him previously that vitally informed this work. This work was supported in part by the U.S. Department of Energy, Office of Nuclear Physics under Award Number(s) DE-SC0021143 and DE-FG02-93ER40762.

## Bibliography

- [1] Stephen DiAdamo, Marco Ghibaudi, and James Cruise. Distributed quantum computing and network control for accelerated vqe. *IEEE Transactions on Quantum Engineering*, 2:1–21, 2021.
- [2] Kenneth Choi, Dean Lee, Joey Bonitati, Zhengrong Qian, and Jacob Watkins. Rodeo algorithm for quantum computing. *Physical Review Letters*, 127(4), jul 2021.
- [3] Jin-Shi Xu, Man-Hong Yung, Xiao-Ye Xu, Sergio Boixo, Zheng-Wei Zhou, Chuan-Feng Li, Alán Aspuru-Guzik, and Guang-Can Guo. Demon-like algorithmic quantum cooling and its realization with quantum optics. *Nature Photonics*, 8(2):113–118, Jan 2014.
- [4] *Classical and quantum computing*, pages 203–217. Springer New York, New York, NY, 2007.
- [5] Carter Ball and Thomas D. Cohen. Boltzmann distributions on a quantum computer via active cooling. *Nucl. Phys. A*, 1038:122708, 2023.
- [6] Carter Ball and Thomas D. Cohen. Zeno effect suppression of gauge drift in quantum simulations. *Phys. Rev. A*, 110(2):022417, 2024.
- [7] Carter Ball. Suppressing gauge drift in quantum simulations with gauge transformations, 2024.
- [8] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1982.
- [9] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, jun 2013.
- [10] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.
- [11] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325–328, July 1997.
- [12] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

- [13] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. Quantum algorithms for quantum field theories. *Science*, 336(6085):1130–1133, 2012.
- [14] Henry Lamm, Scott Lawrence, and Yukari Yamauchi. General methods for digital quantum simulation of gauge theories. *Physical Review D*, 100(3), August 2019.
- [15] Ersen Bilgin and Sergio Boixo. Preparing thermal states of quantum systems by dimension reduction. *Phys. Rev. Lett.*, 105:170405, Oct 2010.
- [16] Arnau Riera, Christian Gogolin, and Jens Eisert. Thermalization in nature and on a quantum computer. *Phys. Rev. Lett.*, 108:080402, Feb 2012.
- [17] Henry Lamm and Scott Lawrence. Simulation of nonequilibrium dynamics on a quantum computer. *Physical Review Letters*, 121(17), Oct 2018.
- [18] Fernando G. S. L. Brandao and Michael J. Kastoryano. Finite correlation length implies efficient preparation of quantum thermal states, 2019.
- [19] Giuseppe Clemente, Marco Cardinali, Claudio Bonati, Enrico Calore, Leonardo Cosmai, Massimo D’Elia, Alessandro Gabbana, Davide Rossini, Fabio Sebastiano Schifano, Raffaele Tripiccion, and et al. Quantum computation of thermal averages in the presence of a sign problem. *Physical Review D*, 101(7), Apr 2020.
- [20] Siddhartha Harmalkar, Henry Lamm, and Scott Lawrence. Quantum simulation of field theories without state preparation, 2020.
- [21] Erik J. Gustafson and Henry Lamm. Toward quantum simulations of  $z_2$  gauge theory without state preparation. *Physical Review D*, 103(5), Mar 2021.
- [22] Mario Motta, Chong Sun, Adrian T. K. Tan, Matthew J. O’Rourke, Erika Ye, Austin J. Minnich, Fernando G. S. L. Brandão, and Garnet Kin-Lic Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics*, 16(2):205–210, Nov 2019.
- [23] Kianna Wan and Isaac Kim. Fast digital methods for adiabatic state preparation, 2020.
- [24] Bipasha Chakraborty, Masazumi Honda, Taku Izubuchi, Yuta Kikuchi, and Akio Tomiya. Digital quantum simulation of the schwinger model with topological term via adiabatic state preparation, 2020.
- [25] Jiangfeng Du, Nanyang Xu, Xinhua Peng, Pengfei Wang, Sanfeng Wu, and Dawei Lu. Nmr implementation of a molecular hydrogen quantum simulation with adiabatic state preparation. *Phys. Rev. Lett.*, 104:030502, Jan 2010.
- [26] A. Messiah. *Quantum mechanics, volume II*. North-Holland Publishing Company, 1969.
- [27] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.
- [28] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995.

- [29] Daniel S. Abrams and Seth Lloyd. Simulation of many-body fermi systems on a universal quantum computer. *Physical Review Letters*, 79(13):2586–2589, Sep 1997.
- [30] Daniel S. Abrams and Seth Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.*, 83:5162–5165, Dec 1999.
- [31] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alan Aspuru-Guzik, and Jeremy L. O’Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5, Jul 2014.
- [32] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski. Cloud quantum computing of an atomic nucleus. *Phys. Rev. Lett.*, 120:210501, May 2018.
- [33] Alessandro Roggero, Andy C. Y. Li, Joseph Carlson, Rajan Gupta, and Gabriel N. Perdue. Quantum computing for neutrino-nucleus scattering. *Physical Review D*, 101(7), Apr 2020.
- [34] Dean Lee, Joey Bonitati, Gabriel Given, Caleb Hicks, Ning Li, Bing-Nan Lu, Abudit Rai, Avik Sarkar, and Jacob Watkins. Projected cooling algorithm for quantum computation. *Physics Letters B*, 807:135536, 2020.
- [35] Erik J. Gustafson. Projective cooling for the transverse ising model. *Physical Review D*, 101(7), Apr 2020.
- [36] Zhengrong Qian, Jacob Watkins, Gabriel Given, Joey Bonitati, Kenneth Choi, and Dean Lee. Demonstration of the rodeo algorithm on a quantum computer. *The European Physical Journal A*, 60(7), July 2024.
- [37] Max Bee-Lindgren, Zhengrong Qian, Matthew DeCross, Natalie C. Brown, Christopher N. Gilbreth, Jacob Watkins, Xilin Zhang, and Dean Lee. Rodeo algorithm with controlled reversal gates, 2022.
- [38] Thomas D. Cohen and Hyunwoo Oh. Optimizing the rodeo projection algorithm. *Phys. Rev. A*, 108:032422, Sep 2023.
- [39] P. Oscar Boykin, Tal Mor, Vwani Roychowdhury, Farrokh Vatan, and Rutger Vrijen. Algorithmic cooling and scalable nmr quantum computers. *Proceedings of the National Academy of Sciences*, 99(6):3388–3393, 2002.
- [40] Álvaro M. Alhambra. Lecture notes on quantum thermal states, August 2021.
- [41] Matthias Troyer and Uwe-Jens Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Phys. Rev. Lett.*, 94:170201, May 2005.
- [42] J. M. Deutsch. Quantum statistical mechanics in a closed system. *Phys. Rev. A*, 43:2046–2049, Feb 1991.

- [43] Mark Srednicki. Chaos and quantum thermalization. *Phys. Rev. E*, 50:888–901, Aug 1994.
- [44] Marcos Rigol, Vanja Dunjko, and Maxim Olshanii. Thermalization and its mechanism for generic isolated quantum systems. *Nature*, 452:854–858, Apr 2008.
- [45] Steven R. White. Minimally entangled typical quantum states at finite temperature. *Phys. Rev. Lett.*, 102:190601, May 2009.
- [46] E M Stoudenmire and Steven R White. Minimally entangled typical thermal state algorithms. *New Journal of Physics*, 12(5):055026, may 2010.
- [47] G. Alvarez. Production of minimally entangled typical thermal states with the krylov-space approach. *Phys. Rev. B*, 87:245130, Jun 2013.
- [48] Barbara M. Terhal and David P. DiVincenzo. Problem of equilibration and the computation of correlation functions on a quantum computer. *Phys. Rev. A*, 61:022301, Jan 2000.
- [49] Thomas D. Cohen, Henry Lamm, Scott Lawrence, and Yukari Yamauchi and. Quantum algorithms for transport coefficients in gauge theories. *Physical Review D*, 104(9), nov 2021.
- [50] David P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48:771–783, 2000.
- [51] Seth Lloyd. Quantum-mechanical maxwell’s demon. *Phys. Rev. A*, 56:3374–3382, Nov 1997.
- [52] Nathanaël Cottet, Sébastien Jezouin, Landry Bretheau, Philippe Campagne-Ibarcq, Quentin Ficheux, Janet Anders, Alexia Auffèves, Rémi Azouit, Pierre Rouchon, and Benjamin Huard. Observing a quantum maxwell demon at work. *Proceedings of the National Academy of Sciences*, 114(29):7561–7564, 2017.
- [53] R. Pathria and P. Beale. *Statistical Mechanics*. Academic Press, Boston, 2011.
- [54] Jad C. Halimeh and Philipp Hauke. Reliability of lattice gauge theories. *Physical Review Letters*, 125(3), July 2020.
- [55] Erez Zohar, J. Ignacio Cirac, and Benni Reznik. Cold-atom quantum simulator for su(2) yang-mills lattice gauge theory. *Physical Review Letters*, 110(12), March 2013.
- [56] L. Tagliacozzo, A. Celi, P. Orland, M. W. Mitchell, and M. Lewenstein. Simulation of non-abelian gauge theories with optical lattices. *Nature Communications*, 4(1), October 2013.
- [57] P. Jordan. Der Zusammenhang der symmetrischen und linearen Gruppen und das Mehrkörperproblem. *Zeitschrift für Physik*, 94:531–535, 1935.
- [58] J Schwinger. On angular momentum. Technical report, US Atomic Energy Commission Report NYO-3071, 1952.

- [59] Erez Zohar and Benni Reznik. Confinement and lattice quantum-electrodynamic electric flux tubes simulated with ultracold atoms. *Physical Review Letters*, 107(27), December 2011.
- [60] Erez Zohar, J. Ignacio Cirac, and Benni Reznik. Simulating compact quantum electrodynamics with ultracold atoms: Probing confinement and nonperturbative effects. *Physical Review Letters*, 109(12), September 2012.
- [61] D. Banerjee, M. Dalmonte, M. Müller, E. Rico, P. Stebler, U.-J. Wiese, and P. Zoller. Atomic quantum simulation of dynamical gauge fields coupled to fermionic matter: From string breaking to evolution after a quench. *Physical Review Letters*, 109(17), October 2012.
- [62] D. Banerjee, M. Bögli, M. Dalmonte, E. Rico, P. Stebler, U.-J. Wiese, and P. Zoller. Atomic quantum simulation of  $U(n)$  and  $SU(n)$  non-abelian lattice gauge theories. *Physical Review Letters*, 110(12), March 2013.
- [63] D. Marcos, P. Rabl, E. Rico, and P. Zoller. Superconducting circuits for quantum simulation of dynamical gauge fields. *Physical Review Letters*, 111(11), September 2013.
- [64] P. Hauke, D. Marcos, M. Dalmonte, and P. Zoller. Quantum simulation of a lattice schwinger model in a chain of trapped ions. *Physical Review X*, 3(4), November 2013.
- [65] Maarten Van Damme, Jad C. Halimeh, and Philipp Hauke. Gauge-symmetry violation quantum phase transition in lattice gauge theories, 2020.
- [66] Jad C Halimeh, Haifeng Lang, and Philipp Hauke. Gauge protection in non-abelian lattice gauge theories. *New Journal of Physics*, 24(3):033015, March 2022.
- [67] Jad C. Halimeh, Lukas Homeier, Annabelle Bohrdt, and Fabian Grusdt. Spin exchange-enabled quantum simulator for large-scale non-abelian gauge theories, 2023.
- [68] Jad C. Halimeh, Haifeng Lang, Julius Mildenerger, Zhang Jiang, and Philipp Hauke. Gauge-symmetry protection using single-body terms. *PRX Quantum*, 2(4), October 2021.
- [69] Jad C. Halimeh, Lukas Homeier, Christian Schweizer, Monika Aidelsburger, Philipp Hauke, and Fabian Grusdt. Stabilizing lattice gauge theories through simplified local pseudogenerators. *Physical Review Research*, 4(3), August 2022.
- [70] Maarten Van Damme, Julius Mildenerger, Fabian Grusdt, Philipp Hauke, and Jad C. Halimeh. Suppressing nonperturbative gauge errors in the thermodynamic limit using local pseudogenerators, 2021.
- [71] Lukas Homeier, Annabelle Bohrdt, Simon Linsel, Eugene Demler, Jad C. Halimeh, and Fabian Grusdt. Realistic scheme for quantum simulation of  $\mathbb{Z}_2$  lattice gauge theories with dynamical matter in (2+1)d. *Communications Physics*, 6(1), June 2023.

- [72] Esteban A. Martinez, Christine A. Muschik, Philipp Schindler, Daniel Nigg, Alexander Erhard, Markus Heyl, Philipp Hauke, Marcello Dalmonte, Thomas Monz, Peter Zoller, and Rainer Blatt. Real-time dynamics of lattice gauge theories with a few-qubit quantum computer. *Nature*, 534(7608):516–519, June 2016.
- [73] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage. Quantum-classical computation of schwinger model dynamics using quantum computers. *Phys. Rev. A*, 98:032331, Sep 2018.
- [74] Yasar Y. Atas, Jinglei Zhang, Randy Lewis, Amin Jahanpour, Jan F. Haase, and Christine A. Muschik.  $Su(2)$  hadrons on a quantum computer via a variational approach. *Nature Communications*, 12(1), November 2021.
- [75] Erez Zohar and J. Ignacio Cirac. Eliminating fermionic matter fields in lattice gauge theories. *Physical Review B*, 98(7), August 2018.
- [76] Erez Zohar and J. Ignacio Cirac. Removing staggered fermionic matter in  $U(N)$  and  $SU(N)$  lattice gauge theories. *Physical Review D*, 99(11), June 2019.
- [77] Guy Pardo, Tomer Greenberg, Aryeh Fortinsky, Nadav Katz, and Erez Zohar. Resource-efficient quantum simulation of lattice gauge theories in arbitrary dimensions: Solving for gauss’s law and fermion elimination. *Physical Review Research*, 5(2), May 2023.
- [78] Reinis Irmejs, Mari-Carmen Bañuls, and J. Ignacio Cirac. Quantum simulation of  $\mathbb{Z}_2$  lattice gauge theory with minimal resources. *Physical Review D*, 108(7), October 2023.
- [79] Pavel P. Popov, Michael Meth, Maciej Lewenstein, Philipp Hauke, Martin Ringbauer, Erez Zohar, and Valentin Kasper. Variational quantum simulation of  $u(1)$  lattice gauge theories with qudit systems. *Physical Review Research*, 6(1), February 2024.
- [80] Valentin Kasper, Torsten V. Zache, Fred Jendrzejewski, Maciej Lewenstein, and Erez Zohar. Non-abelian gauge invariance from dynamical decoupling. *Phys. Rev. D*, 107:014506, Jan 2023.
- [81] Jesse R. Stryker. Oracles for gauss’s law on digital quantum computers. *Physical Review A*, 99(4), April 2019.
- [82] K. Stannigel, P. Hauke, D. Marcos, M. Hafezi, S. Diehl, M. Dalmonte, and P. Zoller. Constrained dynamics via the zeno effect in quantum simulation: Implementing non-abelian lattice gauge theories with cold atoms. *Physical Review Letters*, 112(12), March 2014.
- [83] P Facchi and S Pascazio. Quantum zeno dynamics: mathematical and physical aspects. *Journal of Physics A: Mathematical and Theoretical*, 41(49):493001, oct 2008.
- [84] B. Misra and E. C. G. Sudarshan. The Zeno’s paradox in quantum theory. *Journal of Mathematical Physics*, 18(4):756–763, 04 1977.
- [85] A. G. Kofman and G. Kurizki. Universal dynamical control of quantum mechanical decay: Modulation of the coupling to the continuum. *Physical Review Letters*, 87(27), December 2001.

- [86] Henry Lamm, Scott Lawrence, and Yukari Yamauchi. Suppressing coherent gauge drift in quantum simulations, 2020.
- [87] I. Chuang, L. Vandersypen, X. Zhou, et al. Experimental realization of a quantum algorithm. *Nature*, 393:143–146, 1998.
- [88] Andrew Steane. Quantum computing. *Rep. Prog. Phys.*, 61:117, 1998.
- [89] Colin P. Williams. *Explorations in Quantum Computing*. Springer, 2011.
- [90] C. Shao, Y. Li, and H. Li. Quantum algorithm design: Techniques and applications. *J Syst Sci Complex*, 32:375–452, 2019.
- [91] M. Carena, H. Lamm, S. Lawrence, Y.-Y. Li, J. D. Lykken, L.-T. Wang, and Y. Yamauchi. *Snowmass 2021 LOI*, 2020.
- [92] John Kogut and Leonard Susskind. Hamiltonian formulation of wilson’s lattice gauge theories. *Phys. Rev. D*, 11:395–408, Jan 1975.
- [93] João Barata, Niklas Mueller, Andrey Tarasov, and Raju Venugopalan. Single-particle digitization strategy for quantum computation of a  $\phi^4$  scalar field theory. *Phys. Rev. A*, 103:042410, Apr 2021.
- [94] Andrei Alexandru, Paulo F. Bedaque, Siddhartha Harmalkar, Henry Lamm, Scott Lawrence, and Neill C. Warrington. Gluon field digitization for quantum computers. *Phys. Rev. D*, 100:114501, Dec 2019.
- [95] Minh C. Tran, Yuan Su, Daniel Carney, and Jacob M. Taylor. Faster digital quantum simulation by symmetry protection. *PRX Quantum*, 2:010323, Feb 2021.
- [96] Cheuk-Yin Wong. *Introduction to high-energy heavy-ion collisions*. World Scientific Publishing Co. Pte. Ltd., 1994.
- [97] François Gelis. Some aspects of the theory of heavy ion collisions. *Reports on Progress in Physics*, 84(5):056301, Apr 2021.
- [98] Yun Cheng, L. P. Csernai, V. K. Magas, B. R. Schlei, and D. Strottman. Matching stages of heavy-ion collision models. *Phys. Rev. C*, 81:064910, Jun 2010.
- [99] Paul Romatschke and Ulrike Romatschke. Relativistic fluid dynamics in and out of equilibrium – ten years of progress in theory and numerical simulations of nuclear collisions, 2019.
- [100] Chun Shen and Li Yan. Recent development of hydrodynamic modeling in heavy-ion collisions, 2020.
- [101] M. Stephanov, K. Rajagopal, and E. Shuryak. Event-by-event fluctuations in heavy ion collisions and the qcd critical point. *Phys. Rev. D*, 60:114028, Nov 1999.

- [102] Cheng Chiu and Chun Shen. Exploring theoretical uncertainties in the hydrodynamic description of relativistic heavy-ion collisions. *Physical Review C*, 103(6), Jun 2021.
- [103] Christopher Plumberg, Dekrayat Almaalol, Travis Dore, Jorge Noronha, and Jacquelyn Noronha-Hostler. Causality violations in realistic simulations of heavy-ion collisions, 2021.
- [104] Emma McLaughlin, Jacob Rose, Travis Dore, Paolo Parotto, Claudia Ratti, and Jacquelyn Noronha-Hostler. Building a testable shear viscosity across the qcd phase diagram, 2021.
- [105] Romulo Rougemont, Renato Critelli, Jacquelyn Noronha-Hostler, Jorge Noronha, and Claudia Ratti. Dynamical versus equilibrium properties of the qcd phase transition: A holographic perspective. *Phys. Rev. D*, 96:014032, Jul 2017.
- [106] W. Cassing, O. Linnyk, T. Steinert, and V. Ozvenchuk. Electrical conductivity of hot qcd matter. *Phys. Rev. Lett.*, 110:182301, May 2013.
- [107] Jacquelyn Noronha-Hostler, Jorge Noronha, and Carsten Greiner. Hadron mass spectrum and the shear viscosity to entropy density ratio of hot hadronic matter. *Phys. Rev. C*, 86:024913, Aug 2012.
- [108] Gabriel Denicol, Akihiko Monnai, and Björn Schenke. Moving forward to constrain the shear viscosity of qcd matter. *Phys. Rev. Lett.*, 116:212301, May 2016.
- [109] Sukanya Mitra and Vinod Chandra. Thermal relaxation, electrical conductivity, and charge diffusion in a hot qcd medium. *Phys. Rev. D*, 94:034025, Aug 2016.
- [110] Travis Dore, Jacquelyn Noronha-Hostler, and Emma McLaughlin. Far-from-equilibrium search for the qcd critical point. *Physical Review D*, 102(7), Oct 2020.
- [111] Alekski Kurkela, Wilke van der Schee, Urs Achim Wiedemann, and Bin Wu. Early- and late-time behavior of attractors in heavy-ion collisions. *Phys. Rev. Lett.*, 124:102301, Mar 2020.
- [112] E. Shuryak. Two-stage equilibration in high energy heavy ion collisions. *Phys. Rev. Lett.*, 68:3270–3272, Jun 1992.
- [113] Scott Lawrence. Sign problems in quantum field theory: Classical and quantum approaches, 2020.
- [114] Hefeng Wang. Quantum algorithm for preparing the ground state of a system via resonance transition. *Scientific Reports*, 7, Nov 2017.
- [115] Johannes Roßnagel, Samuel T. Dawkins, Karl N. Tolazzi, Obinna Abah, Eric Lutz, Ferdinand Schmidt-Kaler, and Kilian Singer. A single-atom heat engine. *Science*, 352(6283):325–329, Apr 2016.
- [116] Ronnie Kosloff and Amikam Levy. Quantum heat engines and refrigerators: Continuous devices. *Annual Review of Physical Chemistry*, 65:365–393, 2014.

- [117] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22:563–591, 1980.
- [118] Katherine L. Brown, William J. Munro, and Vivien M. Kendon. Using quantum computers for quantum simulation. *Entropy*, 12(11):2268–2307, November 2010.
- [119] Gregg Jaeger. *Quantum Information: An Overview*, chapter Classical and quantum computing, pages 203–217. Springer New York, New York, NY, 2007.
- [120] Yiqing Zhou, E. Miles Stoudenmire, and Xavier Waintal. What limits the simulation of quantum computers? *Phys. Rev. X*, 10:041038, Nov 2020.
- [121] Adrian Parra-Rodriguez, Pavel Lougovski, Lucas Lamata, Enrique Solano, and Mikel Sanz. Digital-analog quantum computation. *Phys. Rev. A*, 101:022305, Feb 2020.
- [122] Erez Zohar. Quantum simulation of lattice gauge theories in more than one space dimension—requirements, challenges and methods. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2216), December 2021.
- [123] Mari Carmen Bañuls, Rainer Blatt, Jacopo Catani, Alessio Celi, Juan Ignacio Cirac, Marcello Dalmonte, Leonardo Fallani, Karl Jansen, Maciej Lewenstein, Simone Montangero, Christine A. Muschik, Benni Reznik, Enrique Rico, Luca Tagliacozzo, Karel Van Acoleyen, Frank Verstraete, Uwe-Jens Wiese, Matthew Wingate, Jakub Zakrzewski, and Peter Zoller. Simulating lattice gauge theories within quantum technologies. *The European Physical Journal D*, 74(8):165, Aug 2020.
- [124] M. Dalmonte and S. Montangero. Lattice gauge theory simulations in the quantum information era. *Contemporary Physics*, 57(3):388–412, 2016.
- [125] Tim Byrnes and Yoshihisa Yamamoto. Simulating lattice gauge theories on a quantum computer. *Phys. Rev. A*, 73:022328, Feb 2006.
- [126] Erez Zohar, Alessandro Farace, Benni Reznik, and J. Ignacio Cirac. Digital lattice gauge theories. *Physical Review A*, 95(2), February 2017.
- [127] Erez Zohar, Alessandro Farace, Benni Reznik, and J. Ignacio Cirac. Digital quantum simulation of  $F_2$  lattice gauge theories with dynamical fermionic matter. *Phys. Rev. Lett.*, 118:070501, Feb 2017.
- [128] A. Mezzacapo, E. Rico, C. Sabín, I. L. Egusquiza, L. Lamata, and E. Solano. Non-abelian  $su(2)$  lattice gauge theories in superconducting circuits. *Physical Review Letters*, 115(24), December 2015.
- [129] Erez Zohar, J. Ignacio Cirac, and Benni Reznik. Quantum simulations of gauge theories with ultracold atoms: Local gauge invariance from angular-momentum conservation. *Phys. Rev. A*, 88:023617, Aug 2013.

- [130] A. Bazavov, Y. Meurice, S.-W. Tsai, J. Unmuth-Yockey, and Jin Zhang. Gauge-invariant implementation of the abelian-higgs model on optical lattices. *Phys. Rev. D*, 92:076003, Oct 2015.
- [131] Jin Zhang, J. Unmuth-Yockey, J. Zeiher, A. Bazavov, S.-W. Tsai, and Y. Meurice. Quantum simulation of the universal features of the polyakov loop. *Phys. Rev. Lett.*, 121:223201, Nov 2018.
- [132] Daniel C. Hackett, Kiel Howe, Ciaran Hughes, William Jay, Ethan T. Neil, and James N. Simone. Digitizing gauge fields: Lattice monte carlo results for future quantum computers. *Physical Review A*, 99(6), jun 2019.
- [133] Indrakshi Raychowdhury and Jesse R. Stryker. Solving gauss’s law on digital quantum computers with loop-string-hadron digitization. *Physical Review Research*, 2(3), jul 2020.
- [134] T V Zache, F Hebenstreit, F Jendrzejewski, M K Oberthaler, J Berges, and P Hauke. Quantum simulation of lattice gauge theories using wilson fermions. *Quantum Science and Technology*, 3(3):034010, jun 2018.
- [135] Julian Bender, Erez Zohar, Alessandro Farace, and J Ignacio Cirac. Digital quantum simulation of lattice gauge theories in three spatial dimensions. *New Journal of Physics*, 20(9):093001, sep 2018.
- [136] Federica M. Surace, Paolo P. Mazza, Giuliano Giudici, Alessio Lerose, Andrea Gambassi, and Marcello Dalmonte. Lattice gauge theories and string dynamics in rydberg atom quantum simulators. *Phys. Rev. X*, 10:021041, May 2020.
- [137] John B. Kogut. An introduction to lattice gauge theory and spin systems. *Rev. Mod. Phys.*, 51:659–713, Oct 1979.
- [138] Jan Smit. *Introduction to Quantum Fields on a Lattice*. Cambridge Lecture Notes in Physics. Cambridge University Press, 2002.
- [139] G.C. Rossi and M. Testa. The structure of yang-mills theories in the temporal gauge: (i). general formulation. *Nuclear Physics B*, 163:109–132, 1980.
- [140] P. Facchi and S. Pascazio. Quantum zeno subspaces. *Physical Review Letters*, 89(8), August 2002.
- [141] David B. Kaplan and Jesse R. Stryker. Gauss’s law, duality, and the hamiltonian formulation of  $u(1)$  lattice gauge theory. *Physical Review D*, 102(9), November 2020.
- [142] Judah F. Unmuth-Yockey. Gauge-invariant rotor hamiltonian from dual variables of 3d  $U(1)$  gauge theory. *Physical Review D*, 99(7), April 2019.
- [143] Christian W. Bauer and Dorota M. Grabowska. Efficient representation for simulating  $u(1)$  gauge theories on digital quantum computers at all values of the coupling, 2021.
- [144] Yu.L. Dokshitzer, D.I. Dyakonov, and S.I. Troyan. Hard processes in quantum chromodynamics. *Physics Reports*, 58(5):269–395, 1980.

- [145] Subir Sachdev.  $\mathbb{Z}_2$  gauge theory, February 2018.
- [146] Andrei Alexandru, Paulo F. Bedaque, and Scott Lawrence. Quantum algorithms for disordered physics. *Physical Review A*, 101(3), March 2020.
- [147] John B. Kogut. The lattice gauge theory approach to quantum chromodynamics. *Rev. Mod. Phys.*, 55:775–836, Jul 1983.
- [148] Kurt Haller. Quantum electrodynamics in the temporal gauge. *Phys. Rev. D*, 36:1830–1838, Sep 1987.
- [149] Carlo Rovelli. What is a gauge transformation in quantum mechanics? *Phys. Rev. Lett.*, 80:4613–4616, May 1998.
- [150] Roland C. Farrell, Ivan A. Chernyshev, Sarah J. M. Powell, Nikita A. Zemlevskiy, Marc Illa, and Martin J. Savage. Preparations for quantum simulations of quantum chromodynamics in 1 + 1 dimensions: (i) axial gauge. *Physical Review D*, 107(5), March 2023.
- [151] Matteo M. Wauters, Edoardo Ballini, Alberto Biella, and Philipp Hauke. Symmetry-protection zeno phase transition in monitored lattice gauge theories, 2024.
- [152] Tobias Schmale and Hendrik Weimer. Stabilizing quantum simulations of lattice gauge theories by dissipation. *Phys. Rev. Res.*, 6:033306, Sep 2024.
- [153] Colin P. Williams. *Quantum Gates*, pages 51–122. Springer London, London, 2011.