UNIVERSITY OF CALIFORNIA,
IRVINE

Computing in GRW Quantum Mechanics & Lessons for Physical Computation

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Philosophy

by

Timothy Schmitz

Dissertation Committee:
Jeffrey A. Barrett, Chair
J.B. Manchak
James Owen Weatherall

2020

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to thank Jeff Barrett, Jim Weatherall, JB Manchak, Simon Huttegger, Pen Maddy, Chip Sebens, my fellow UCI PhD students, and the various reviewers and conference attendees who have gone out of their way to provide detailed and thorough feedback. The conversations we have had have and the responses you have given to my work have been crucial to the completion of this dissertation.

CURRICULUM VITAE

Timothy Schmitz

EDUCATION

Doctor of Philosophy in Philosophy **2020**
University of California, Irvine *Irvine, CA*

Masters of Arts in Philosophy **2017**
University of California, Irvine *Irvine, CA*

Bachelor of Science in Physics, Applied Mathematics, and Philosophy **2014**
Saint Louis University *Saint Louis, MO*

RESEARCH EXPERIENCE

Graduate Research Assistant **2007–2012**
University of California, Irvine *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant/Associate **20014–2020**
University of California, Irvine *Irvine, California*

ABSTRACT OF THE DISSERTATION

Computing in GRW Quantum Mechanics & Lessons for Physical Computation

By

Timothy Schmitz

Doctor of Philosophy in Philosophy

University of California, Irvine, 2020

Jeffrey A. Barrett, Chair

Our epistemic limits depend on the nature of the physical world we inhabit and the ways in which we can observe and manipulate it. This dissertation examines these limits in the realm of physical computation. It focuses on quantum computing and particularly computing in GRW quantum mechanics, although it has implications for both quantum computing in particular and physical computation more broadly. By rigorously investigating the conceptual issues that arise when looking at computation in GRW, we see how the conceptual issues in the foundations of quantum mechanics and our solutions to these issues determine what models of quantum computation can be realized and what quantum computing can achieve. More broadly, we see how grounding our discussions of physical computation in precise physical theories informs our understanding of computational and epistemic limits and illustrates the challenges and nuances involved in the work of developing a rigorous understanding of what is physically computable.

Chapter 1

Introduction

Our epistemic limits depend at least in part on the nature of the physical world we inhabit and the ways in which we can observe and manipulate physical objects. This dissertation examines these limits in the realm of physical computation by investigating what may be be physically computable. Though this dissertation focuses on quantum computing and particularly computing in GRW quantum mechanics, an alternative formulation of quantum mechanics, it has implications for both quantum computing in particular and physical computation more broadly. By rigorously investigating the conceptual issues that arise when looking at computation in GRW, we will see how grounding our discussions of physical computation in precise physical theories informs our understanding of computational and epistemic limits. We will also see how conceptual issues in the foundations of quantum mechanics have implications for what models of quantum computation can be realized.

Chapter 2 argues that the measurement problem of the von Neumann-Dirac ("Standard Collapse") formulation of quantum mechanics results in conceptual problems for quantum computational models that rely on it. The von Neumann-Dirac formulation's measurement problem arises from it having two contradictory dynamics that are supposed to govern

quantum systems, and it not rigorously specifying when each dynamics applies. Thus, this formulation is either incomplete or inconsistent. Without a clear delineation of when each of the dynamics applies, how are we to determine what operations computers in this theory will implement? However, formulations of quantum mechanics which do not suffer from this measurement problem exist. I briefly discuss how computation would differ in two such examples, Bohmian Mechanics and GRW quantum mechanics. The remainder of my dissertation primarily focuses on the case of GRW quantum mechanics, which avoids the measurement problem by incorporating spontaneous wavefunction collapses in position as part of its dynamics.

In the third chapter, I ground a discussion of computation in GRW in the Aitken/Barrett thesis, a prescription that questions regarding physical computation should be addressed with respect to a model of computation defined relative to a precise physical theory. Following this prescription, my initial examination of computation in GRW centers on Turing machines built in the theory. Turing Machines illustrate both the idiosyncrasies of GRW regarding physical computation and how one ought to cater models of physical computation to particular physical theories when following the Aitken/Barrett thesis, as a Turing machine model and other models of computation where positions of components play critical roles may be more susceptible to noise generated by GRW's stochastic dynamics than those which minimally rely on component positions. This shows the role a physical theory's unique properties can play in determining both possible physical computational strength and what models of computation are most capable of maximizing this computational strength. It also demonstrates how our canonical models of computation may be ill-suited for understanding the computational power possible in some physical theories.

In the fourth chapter, I extend the work of the previous chapter by presenting a model of error for a simple register machine in GRW, in which the only source of error is the stochastic nature of GRW's dynamics. This more vividly depicts the innate connection between the

physical details of a model of computation in GRW and the model's computational strength. I find mathematical results for the error rates attributable solely to the noise generated by GRW's stochastic dynamics for an idealized register machine running classical computations. My treatment also shows that a similar analysis of a quantum register machine in GRW is much more complicated, as determining error rates in this case requires precise specification of the shape of the wavefunction underlying all computational states, which is not the case in most error models. Thus, we cannot describe the noise GRW's dynamics would cause for a quantum computer without a highly precise specification of the physical details of a model of quantum computing in GRW. Additionally, this treatment highlights the uniquely unstable features of GRW systems which result in quantum error rates being a possible avenue for empirically testing GRW as a physical theory.

Chapter 5 takes our examination of GRW computing to the limit and looks at supertasks in the context of GRW. Because collapses in GRW occur based on the passage of time and the number of particles entangled in position, GRW distinguishes between methods of executing supertasks based on a variety of physical properties – for instance, GRW's dynamics would not allow supertasks in which the computing system experiences an infinite amount of time but would allow some supertasks in which the computing system experiences a finite amount of time, and whether a finite time supertask are allowed by GRW's dynamics can even depend on the nature of the correlations among the components of the computing system. This both gives us a more robust understanding of how GRW's particular features affect computational models in it and how complicated the effects of dynamical rules on the potential of supertasks can be.

The sixth chapter looks at physical computation more generally, focusing on how we determine what models of physical computation are epistemically useful for finite observers – meaning that finite observers are able to use them to obtain desired values of precisely defined functions. I examine conditions for when a model of physical computation is usable

for finite observers, given by Piccinini (2017), and argue that understanding what models of physical computation are usable is more complicated than Piccinini has acknowledged. One of his criteria for usability implicitly relies on a classical computational understanding of what operations can be performed, rather than an understanding that emerges from a particular physical theory. But the set of operations that are usable is innately tied to what physical theory holds. However, attempts to derive the set of usable operations directly from a physical theory fail due to a bootstrapping problem. Thus, we are left with a choice of how to select what physically possible states and operations to include in models of physical computation. I argue that in this situation, the classical notion used by Piccinini and others is a good provisional starting point, though we should be aware that it is an assumption we are making when we use it to discuss physical computation.

My dissertation rigorously investigates the process of fulfilling the Aitken/Barrett thesis by defining computational models relative to a particular physical theory. This investigation reveals conceptual issues in standard treatments of quantum computing that must be taken seriously if we are to fully understand how quantum computing works and what it can do. But this investigation also reveals broader insights into the process of investigating the empirical limits imposed by physical theories. I show how the precise characteristics of a physical theory determine what particular models of computing in the theory can do and which computational models might be epistemically useful for finite observers in the theory. Additionally, I show that physical computational limits may manifest in ways that are not generalizable to other physical theories. This dissertation demonstrates the complexity of considerations we must take into account when determining physics' constraints on the problems we can solve and the information we can acquire. It also serves both as a warning against ambitious claims about quantum computing or physical computation in general, whose plausibility may rely on uncertain assumptions about how the physical world works, and as an example of the positive process of rigorously investigating the limits of computation in a particular physical theory.

Chapter 2

Concerns with Standard Quantum Computing

2.1 Introduction

Quantum computing has typically been described in terms of the von Neumann-Dirac Formulation of quantum mechanics,[1] according to which quantum states evolve linearly, except in the case of measurement, where they instantly collapse to a eigenstate of the measurement basis. Treatments of quantum algorithms typically explicitly or implicitly assume this view. We will look particularly at the example of Shor's factoring algorithm in Section 2.2.4.

The flaws of the von Neumann-Dirac Formulation are well-documented (e.g. (11),(15)) As a physical theory it is either incomplete or inconsistent, raising concerns about any model of computation that is based on it. The conceptual issues of the von Neumann-Dirac Formulation also apply to any model of computation which relies on this theory and that utilizes the problematic aspects of the theory. Any model of quantum computation based on

[1] For some examples, see (50)(51)(53)(70).

this theory is thus not a complete and consistent description of the kinds of tasks performed by physical quantum computers. Though the standard accounts are useful for many of the practical tasks we currently wish to perform, one might hope for alternatives which are free of these conceptual issues, especially as we advance in our ability to implement quantum computational processes.

Two consistent alternatives to the von Neumann-Dirac Formulation which make explicit dynamical changes are Ghirardi-Rimini Weber (GRW) theory and Bohmian mechanics. Each can give rise to distinct models of quantum computation. GRW is a collapse theory which precisely defines how and when collapses occur. Bohmian mechanics is a nonlocal hidden variable approach in which collapses are unnecessary because particles always have definite trajectories. According to Bohmian Mechanics, particles are always governed by the linear dynamics — unlike in the von Neumann-Dirac Formulation and GRW, in Bohmian Mechanics there are no collapses.

We will look at how quantum computing arises out of these alternatives to the von Neumann-Dirac formulation and how it differs among the three approaches. We will find that qubits and the way computational operations act differ among the three formulations. Also, the GRW model introduces a source of computational noise for any given quantum computer that is not found in the standard or Bohmian models. Because of this noise, GRW incentivizes building certain kinds of physical quantum computers in a way that the von Neumann-Dirac Formulation and Bohmian Mechanics do not.

Section 2.2 examines at the von Neumann-Dirac Formulation and a model of quantum computation that arises out of it. Section 2.3 discusses the problems of the von Neumann-Dirac Formulation and how these problems undermine standard models of quantum computing. Section 2.4 describes GRW and examine how GRW models of quantum computation would differ from the von Neumann-Dirac model. Section 2.5 looks at Bohmian Mechanics and quantum computation in it. Section 2.6 compares quantum computation in these three

formulations of quantum mechanics. Section 2.7 concludes.

2.2 The von Neumann-Dirac Formulation

The von Neumann-Dirac Formulation,[2] of quantum mechanics dominates the literature on quantum mechanics and is almost certainly what the reader would have learned when first encountering quantum mechanics. Unsurprisingly, it is also the conceptual understanding of quantum physics on which almost all treatments of quantum computing are built. This formulation of quantum physics has two radically different dynamics, the linear dynamics and the collapse dynamics, and it relies on the use of both of these dynamics as well as our ability to cleanly separate the occasions in which each applies. Likewise, models of quantum computation that utilize this formulation will rely on this clear distinction.

Section 2.2.1 lays out the von Neumann-Dirac Formulation. Section 2.2.2 describes qubits in the von Neumann-Dirac Formulation and how they can behave, and section 2.2.3 describes a model of quantum computation which is based on the von Neumann-Dirac Formulation. Subection 2.2.4 looks at Shor's factoring algorithm as an example of how computational models based on this formulation are used.

2.2.1 The von Neumann-Dirac Formulation

The standard von-Neumann Dirac formulation of quantum mechanics, as presented by Barrett (11) is as follows:

1. States of physical systems are represented by unit length vectors $|\psi\rangle$ in a Hilbert space

[2]This view is often called the "Copenhagen Interpretation", though this is not without a tinge of controversy. Many different positions have been called the "Copenhagen Interpretation". I will not delve into the history of the moniker or the von Neumann-Dirac Formulation; I reference the Copenhagen name only in hopes that some readers will now recognize the view.

$\mathcal{H}$.

2. Every physical observable O is represented by Hermitian operator $\hat{O}$, and likewise every Hermitian operator represents a physical observable.

3. A system has a definite value for an observable only if it is in an eigenstate of that observable, in which case the eigenvalue is the result one would find if they were to measure that observable of the system. This is called the "eigenstate-eigenvalue link".

4. Systems are governed by two different dynamics.

   A. When no measurement is made of a physical system, the system evolves unitarily – that is, in a linear and deterministic way. Where $\hat{U}$ is some unitary transformation, from time t_0 to time t_1, the state will go from $|\psi(t_0)\rangle$ to $\hat{U}(t_0, t_1)|\psi(t_0)\rangle$.

   B. If a system is measured, it will instantly "collapse" into an eigenstate of the measurement basis. If the initial state is $|\psi\rangle$ and $|\chi\rangle$ is an eigenstate of O, the probability of the system collapsing to $|\chi\rangle$ is given by $|\langle\psi|\chi\rangle|^2$.

5. Compositions of systems are represented by tensor products. So the system composed of S_1 with state $|\psi\rangle \in \mathcal{H}_1$ and S_2 with state $|\phi\rangle \in \mathcal{H}_2$ is $|\psi\rangle \otimes |\phi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$. Likewise for properties of systems.

Two particularly salient aspects of the von Neumann-Dirac Formulation in comparison to other physical theories are the eigenstate-eigenvalue link espoused in 3 and the two radically different and mutually exclusive dynamics of 4. The eigenstate-eigenvalue link is a radical departure from classical mechanics because it implies that observables including position and momentum do not always have definite values. Because systems cannot possibly be in eigenstates of all observables at all times, not all observables will have definite values at any given time, though the collapse dynamics ensures that measurements will always result in definite values for whatever observables are measured.

Not only is the quantum state necessarily never in an eigenstate of all observables, the linear dynamics alone will often not put the system into an eigenstate of an observable in cases where we empirically find that that observable does have a definite value. Given the eigenstate-eigenvalue link, we need another dynamics to ensure that the system is in an eigenstate of an observable when our measurements indicate that it is. Fitting these empirical results is the role of the collapse dynamics. The collapse dynamics, rather than the linear dynamics, governs systems during measurement events. In a measurement, the collapse dynamics ensures that the system is in an eigenstate of an observable, so our measurements find a single definite value rather than a superposition of values. So the von Neumann-Dirac Formulation has two different mutually exclusive dynamics which govern all physical systems. The collapse dynamics governs systems in the special case of measurement, and the linear dynamics governs systems in all other cases.

In contrast, GRW makes all collapses spontaneous and in position at the cost of loosening the eigenstate-eigenvalue link. We will see that the way GRW treats collapses leads to differences between the GRW and standard computing models regarding the possible computational power of quantum computers and whether certain ways of constructing quantum computers are preferable to others.

We will also see that Bohmian Mechanics and the von Neumann-Dirac Formulation differ both in their understanding of quantum mechanical states and the dynamics governing these states. In lieu of the eigenstate-eigenvalue link, Bohmian Mechanics selects a preferred observable, position[3] which always has a definite value. And with positions of particles given determinate values at all times, Bohmian Mechanics can do away with the collapse dynamics entirely. In its place are additions to the linear dynamics that define real trajectories for particles. This leads to a different set of physical operations that could be used for computation in Bohmian Mechanics than in the von Neumann-Dirac Model.

[3]In some formulations of Bohmian Mechanics, momentum is the preferred observable. For our purposes, we will treat position as the preferred observable.

2.2.2 Qubits

Describing quantum bits, or "qubits", can help illustrate what quantum mechanics may be able to add to classical computation. Classical computing uses binary bits which can occupy one of two states and can be switched between the states. Bits always occupy one of these states, usually denoted by 0 and 1 (though the actual numbers used are insignificant; we could just as meaningfully call the two states "Joseph" and "Donald"). Computing operations on these bits can only either leave them in a state or flip their state to the opposite state (i.e. change a 0 to 1 or 1 to 0). For instance, the NOT operation sends an input of 0 to an output of 1 and vice versa. The classical computing operations are thus limited to the Boolean operators AND and NOT along with all combinations of these operators.

Classical gates take a sets of input bits and give a sets of output bits. Typically, these gates output a single bit; for instance, an AND gate will have two input bits and only one output bit. Because of this reduction in the number of bits from input to output, classical computing is not necessarily reversible. For instance, if I send two bits through an AND gate and receive a 0 (false), all I know is that at least one of my two original bits was 0; I do not know which one was 0 or whether both were 0, and thus I cannot undo this operation to get back my original pair of bits. However, all classical gates can be made reversible with little additional computing cost by storing bits which can be used to recover the bit of information that would typically be lost by gates with fewer output bits than input bits. This can be done through the use of controlled gates, which take some set of n bits as input, and, based on the states of the first $n-1$ bits, either flip the last bit or leave it unchanged. The $n-1$ "control" bits are left unchanged, so that if the computation were reversed and the set of bits were sent back through the gate, the n-th bit would be flipped if and only if it had been flipped in the initial computation, ensuring that the end product of the reversed computation is the initial input into the computer.

Instead of bits, quantum computing uses qubits. Let us describe qubits in the von Neumann-Dirac Formulation.[4] When measured in the "standard" basis, the collapse dynamics put qubits in one of two states, $|0\rangle$ and $|1\rangle$, akin to the 0 and 1 states of classical bits. However, unlike classical bits, they are not limited to these two states throughout the computing process. Instead, a qubit state can be any unit length vector in Hilbert space, which includes superposition of the two states. So while a classical bit must always be either 0 or 1, a qubit can be in $|0\rangle$, $|1\rangle$, or any of the superposition states $a|0\rangle + b|1\rangle$, where a and b are complex scalars such that $|a|^2 + |b|^2 = 1$. The states $|0\rangle$ and $|1\rangle$ are orthogonal, so they span the two dimensional space. Thus, the qubit given by $a|0\rangle + b|1\rangle$ can also be expressed as $\begin{pmatrix} a \\ b \end{pmatrix}$ where the basis $\{|0\rangle, |1\rangle\}$ is used. These superposition states do not have a classical analog.

Whereas a multi-bit system will consist of a string of 0s and 1s, a multi-qubit system will consist of a vector in a tensor product of Hilbert spaces. So a multi-qubit state consisting of the single qubits $|\phi\rangle \in \mathcal{H}_1$ and $|\psi\rangle \in \mathcal{H}_2$ written in some specified basis will be $|\phi\rangle \otimes |\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ in that basis, and likewise for multi-qubit states with more members. So an n-qubit state could be

$$\bigotimes_{i=1}^{n} |\phi\rangle_i \in \bigotimes_{i=1}^{n} \mathcal{H}_i. \tag{2.1}$$

All separable states can be written in this way, and one will often see the shorthands $|\phi\rangle_1|\phi\rangle_2...|\phi\rangle_n$ or even $|\phi_1\phi_2...\phi_n\rangle$ in the place of this explicit tensor product notation. Strings of classical bits are like separable qubit states in that they can always be written as strings of independent single bits. However, not all multi-qubit states can be written as tensor products of single qubit states; these states may still inhabit the tensor product of n Hilbert spaces, but one may not be able to factor apart their individual single qubit states. These are entangled states and are a crucial part of quantum computing. Nothing like them is

[4]Qubits in GRW and Bohmian Mechanics will differ slightly from those described in this section.

found in classical computing. Note that a multi-qubit state may be entangled according to one decomposition but not another. For instance, the four-qubit state

$$\begin{aligned}|\psi\rangle &= \frac{1}{2}(|0\rangle_1|0\rangle_2|0\rangle_3|0\rangle_4 + |0\rangle_1|1\rangle_2|0\rangle_3|1\rangle_4 + |1\rangle_1|0\rangle_2|1\rangle_3|0\rangle_4 + |1\rangle_1|1\rangle_2|1\rangle_3|1\rangle_4) \\ &= \frac{1}{\sqrt{2}}(|0\rangle_1|0\rangle_3 + |0\rangle_1|0\rangle_3) \otimes \frac{1}{\sqrt{2}}(|0\rangle_2|0\rangle_4 + |1\rangle_2|1\rangle_4)\end{aligned} \tag{2.2}$$

is unentangled when we decompose it into two systems, one composed of the odd and the other of the even qubits. However, when we decompose it into systems consisting of the first two and the last two qubits, it is entangled (61):

$$\begin{aligned}|\psi\rangle =& \frac{1}{2}(|0\rangle_1|0\rangle_2 \otimes |0\rangle_3|0\rangle_4 + |0\rangle_1|1\rangle_2 \otimes |0\rangle_3|1\rangle_4 \\ &+ |1\rangle_1|0\rangle_2 \otimes |1\rangle_3|0\rangle_4 + |1\rangle_1|1\rangle_2 \otimes |1\rangle_3|1\rangle_4).\end{aligned} \tag{2.3}$$

Thus, how we partition a system into subsystems can determine whether we have entangled or separable systems.

Because of their representation as vectors in a complex space, qubits can be transformed by the linear dynamics of the von Neumann-Dirac Formulation in many more ways than classical bits can be transformed. Qubits can undergo any unitary transformation, that is, any linear transformation that can be represented by a matrix that satisfies $U^\dagger = U^{-1}$. Unitary transformations can be classified into phase shifts, rotations, phase rotations, and combinations of these.

Phase shifts are just multiplications of the state with a magnitude 1 complex number, and are represented by

$$K(\delta) = e^{i\delta} I, \tag{2.4}$$

so $K(\delta)v = e^{i\delta}v$. As stated above, such a transformation alone will not change the proba-

bilities associated with measurement results.

Rotations rotate the state through the two-dimensional space and are represented by

$$R(\beta) = \begin{pmatrix} cos\beta & sin\beta \\ -sin\beta & cos\beta \end{pmatrix}. \tag{2.5}$$

Phase rotations are of the form

$$T(\alpha) = \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix}. \tag{2.6}$$

All quantum transformations can be expressed as combinations of these three kinds of unitary transformations (61).

As an example of a transformation which is expressible as a combination of these three types of unitary transformations, consider the negation ("NOT") transform

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{2.7}$$

which we can see behaves as expected, e.g.

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.8}$$

X can be written as $K(-\pi/2)R(\pi/2)T(\pi/2)$ since

$$K(-\pi/2)R(\pi/2)T(\pi/2) = -i \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{2.9}$$

Because any unitary transformation can be done on a qubit, quantum computing has access to not only those that correspond to classical operations like NOT and AND, but also things like Hadamard Gates

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{2.10}$$

which can toggle whether a qubit is in an eigenstate of the standard basis or a superposition in the standard basis, and the square root of NOT

$$\sqrt{(X)} = \frac{1}{2} \begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}, \tag{2.11}$$

which, when applied twice, has the same effect as the negation transformation X. The number of unitary operations that can be done on qubits is uncountable, since the number of phase shifts, rotations, and phase rotations is uncountable. This does not, however, imply that physical implementations of all the transformations are easily created. Instead, physical quantum computers will restrict themselves to a finite set of unitary transformations that are easily constructed, and a task for those working on quantum algorithms is finding sets of transformations that will be conducive to a wide variety of computational tasks. However, since we are concerned with the model of computation rather than any particular physical computer, we can ignore these concerns.

Unitary operators are invertible, and the inverse of a unitary operator is also a unitary operator. This means that quantum computing processes that can be modeled by unitary transformations are reversible, as any computing process done with the application of unitary transformations can be undone with the application of the inverses of those unitary transformations.[5] A unitary transformation on a qubit can be undone by acting on the

[5]We will see that the extent to which unitary transformations apply is an interpretational matter that depends on the dynamics of the quantum theory from which a model of quantum computing is built. GRW

qubit with the inverse of that transformation, and likewise, if I send a qubit through a set of unitary transformations, there is another set of unitary transformations I could then send it through to return the qubit to its original state. Recall that this is not the case for all the operations one can perform on classical bits, but that reversible classical gates can be used without significant computing cost. This allows us to use analogs to these reversible classical gates in quantum computing, and the quantum versions of nonreversible classical gates such as AND or OR behave like the reversible versions of these classical gates. So reversible classical gates form a proper subset of the gates available in quantum computing.

But despite the additional freedom of qubits compared to bits, only one bit of information can be extracted from each of them — we are not able to take advantage of the additional complexity of the superposition to store multiple classical bits of information inside a single qubit, even though superposition states involve scalars which can be infinite-bit numbers.[6] This can be explained as the result of measurement in the standard basis $\{|0\rangle, |1\rangle\}$ of the qubit's state always finding the qubit to be in one of two states: either $|0\rangle$ or $|1\rangle$. When in a superposition state $a|0\rangle + b|1\rangle$, a measurement in the standard basis will find the qubit to be in state $|0\rangle$ with probability $|a|^2$ and in state $|1\rangle$ with probability $|b|^2$.[7] We can see then, that multiplying the entire state by a complex number c, where $|c|^2 = 1$, will not change these probabilities.

2.2.3 von Neumann-Dirac Quantum Computing

This section will briefly present a model of computation in the von Neumann-Dirac Formulation. The computational operations in this model will be performed on qubits, as described

limits the role of the unitary dynamics and thus the situations in which only unitary transformations are used in quantum computing. This is not the case for theories without collapse, like Bohmian Mechanics.

[6]The case of dense coding may look like an exception to this. However, the dense coding procedure requires the receiver of the the two bits of information, "Bob", to measure both entangled qubits to extract two bits of information.

[7]This is found to be true empirically, and is not an assumption of a particular interpretation of quantum mechanics.

by the respective formulations of quantum mechanics. Though states of qubits can be expressed in some standard basis $\{|0\rangle, |1\rangle\}$, the physical observable we select to correspond to eigenstates of this basis will depend on the physical system used to implement the qubits. Different ways of building a quantum computer will result in different measurement bases being used as the standard basis for qubits. But, in this formulation of quantum mechanics, which basis is used does not change the dynamical rules that apply, so we may treat qubits generally in our von Neumann-Dirac computing model and not assume that the standard basis is any particular observable, as the standard basis observable we select will not affect our model of computation.

In this section, I will use the description of a mathematical model of quantum computation given by Bernhard Ömer (1998). This model assumes the von Neumann-Dirac Formulation and utilizes the operations described in the previous section.

The majority of operations in most quantum computations will be unitary operations[8]. These will consist of the application of some unitary operator to a qubit state, as we have seen above.

To begin a computation, we will often need to set the qubits into a state of our choosing. Accomplishing this requires what Ömer calls a "reset operator" R that sets any state $|\psi\rangle$ to our specified $|\psi_0\rangle$ (51). This operator R allows us to begin a computation with our computer in any state we choose by putting the qubits in a particular state to which we can apply any unitary transformation we please. Physically implementing R in the von Neumann-Dirac Formulation simply requires measuring a system (via the collapse dynamics) with state $|\psi\rangle$ in the basis for which $|\psi_0\rangle$ is an eigenstate, then, depending on the result of the measurement, either leaving the system in $|\psi_0\rangle$ or applying an X (eq. 2.7) transformation (utilizing the linear dynamics) to the state. The collapse dynamics puts the qubit into an eigenstate of the

[8]Though Ömer states that these operators "fully describe the quantum computation itself", that will not always be the case.

standard basis after measurement, no matter what observable corresponds to the standard basis. This computational procedure demands that we are able to determine precisely when the collapse dynamics and not the linear dynamics will govern the evolution of our qubits. We will see in Section 2.3 that the von Neumann-Dirac Formulation does not give us a way to make this determination.

Ömer states that reading the results of a computation requires a "*measurement operator* $M(\mathcal{O})$ of the observable $\mathcal{O} = \{\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_k\}$, where $\mathcal{E}_i$ is a mutually orthogonal decomposition of $\mathcal{H}$" that will apply a random projection operator to the measured qubit system to project it into an eigenstate of the measurement basis (51). Projectors are selected according to the probabilities prescribed by the collapse dynamics of 4B. Ömer calls the "classical outcome" of the measurement $\mu(m)$, where $\mu : \{0, 1, ..., k\} \to \mathbb{R} \times \{\text{physical unit of } \mathcal{O}\}$, and $\mathbb{R} \times \{\text{physical unit of } \mathcal{O}\}$ is just the quantitative result of a measurement. This classical outcome gives the eigenvalue (with units) corresponding to the eigenstate of the measurement observable. The collapse dynamics puts the system in an eigenstate of the measurement observable. In both the measurement and reset operators, we stipulate that the collapse dynamics governs the qubits when and only when the measurement and reset operators demand it. To be adequate for this purpose, the von Neumann-Dirac Formulation ought to distinguish the cases where each dynamics applies, but it does not. However, GRW does, and no-collapse theories such as Bohmian mechanics (Section 2.5) will not even have a distinction to be made.

The measurement operator differs from the reset operator in that its final outcome is not guaranteed to be a particular state and its basis is not necessarily the standard basis selected for the quantum computer. It corresponds to any type of measurement, including those in the basis used in the reset operator's measurement. So a reset operator is just constructed out of a particular measurement operator followed by a conditional negation operator.

Quantum computations in the von Neumann-Dirac Formulation will thus be constructed using these three kinds of procedures — unitary transformations which only use the linear

dynamics, and reset and measurement operators which utilize the collapse dynamics. The tidy split between the two dynamics of the physical theory allows the tidy split between the type of computational actions in the model of computation which is based on the physical theory.

2.2.4 The Example of Shor's Algorithm

The von Neumann-Dirac Formulation is either explicitly or implicitly used in most work on quantum computing. Perhaps the most famous example to illustrate its use is Shor's factoring algorithm. Shor's algorithm is likely the most well-known quantum algorithm and was an early example of a quantum algorithm that could perform a task more quickly than any known classical algorithm. It uses a quantum Fourier transform to factor an integer in polynomial time instead of the exponential time of the fastest known classical factoring algorithm, the number field sieve (70). This algorithm has major consequences for cryptography, which often relies on the difficulty of factoring large numbers. In its original presentation, it assumes the von Neumann-Dirac Formulation and uses the operations allowed by this formulation.

Shor's algorithm explicitly uses unitary transformations, measurement operators, and the reset operator (in addition to some classical computational processes that we will not discuss here) to factor an integer. It also illustrates that a model of quantum computation based on the von Neumann-Dirac Formulation is good enough for many practical purposes, as the physics literature is full of successful physical implementations of Shor's algorithm.[9]

The algorithm requires the reset and measurement operators, which use the collapse dynamics. It uses a process called quantum parallelism to set the state of the qubit system. This process utilizes the reset operator to set the qubits in an initial state of $|00...0\rangle$ on which a

[9]The first of these was (76) in 2001. To my knowledge, the largest number successfully factored using Shor's algorithm is 56,153 by (21).

Walsh transformation $W = H \otimes H \otimes ... \otimes H$ is applied (61). Measuring the qubit system (with the collapse dynamics) then applying the appropriate unitary transformations (e.g. possibly by an operator of the form $X \otimes ... \otimes X$ followed by W) with the linear dynamics allows us to prepare the qubit system.

During the steps that follow in the algorithm, Shor advocates using measurement to ensure a bit is in the $|0\rangle$ state:

> "If it is not, we know that there has been an error somewhere in the quantum computation.... if we do find that b [the qubit] is 0 [that is, $|0\rangle$], then we know... that it is now exactly 0. This measurement thus may bring the quantum computation back on track in that any amplitude the b had for being non-zero has been eliminated" (70).

This is as much a check to make sure that the qubit is in the correct state as it is an attempt to put the qubit back into the appropriate state. In fact, Shor claims that measuring the qubit b every time it should be in the $|0\rangle$ state may increase the chance that the algorithm will succeed and states, "I believe that partial measurements such as this one are a promising way of trying to stabilize quantum computations" (p. 12 of (70)).[10] The collapse dynamics are not just used to set states and read the results of computations, but also to increase the reliability of computations.

The algorithm uses a unitary transformation A_q that acts like a discrete Fourier transform. It is composed of Hadamard transformations which act on the j-th qubit (which Shor calls "R_j") and another class of two-qubit unitary transformations (which Shor calls "$S_{j,k}$") which, if applied in the correct sequence, perform a quantum Fourier transform (70).[11] After this

[10] This is due to the quantum Zeno effect, which utilizes frequent measurements (and thus collapses) to keep a quantum system close to being in an eigenstate of the measurement observable.

[11] There are a couple more minor steps involved, but these are not important to our main point and would require a much more detailed account of Shor's algorithm to explain.

step, a measurement is made, and then classical steps will obtain the prime factorization from the measurement result.

In his discussion of quantum computing, Shor describes the collapse dynamics and relies on our ability to determine whether the collapse or the linear dynamics governs any particular step of the algorithm. An intuitive but imprecise notion of what counts as a measurement and thus when the collapse dynamics govern the qubits lurks in the background throughout his presentation. For instance he claims that "looking at the machine during the computation will invalidate the rest of the computation" because doing so "projects" the state into the "observed basis" (70). But what counts as the kind of "looking" that would result in a measurement operation? This lack of precision does not undermine Shor's result insofar as the algorithm has advanced our understanding of quantum computing and has promise for practical application, but presages that the computing model he uses relies on a flawed physical theory, even though versions of this algorithm will be executable in other formulations of quantum computing.

2.3 Problems with the von Neumann-Dirac Formulation

The problems with the von Neumann-Dirac formulation of quantum mechanics are well-documented, and the view has received criticism ever since its development (see (8) (11) (15) (67) (68)). Its troubles arise due to the "measurement problem", in which the the collapse dynamics and the linear dynamics, both which are supposed to govern the same quantum systems in at least some cases, contradict each other, and the theory does not provide a robust way to sort out in which cases each of the two dynamics apply. Worse, as we will see, any partitioning of physical systems into those which cause collapses and those

which do not will be arbitrary and ad hoc.

A simple thought experiment shows that the linear and collapse dynamics contradict each other. Consider two physical systems A and B, where B is larger than A. System A begins in the state

$$|\psi\rangle_A = \frac{1}{\sqrt{2}}(|0\rangle_A + |1\rangle_A), \tag{2.12}$$

and B begins in the state

$$|Ready\rangle_B = \frac{1}{\sqrt{2}}(|Zero\rangle_B + |One\rangle_B). \tag{2.13}$$

The experiment is set up so that, according to the linear dynamics the particles entangle and give the state

$$|\psi\rangle_{AB} = \frac{1}{\sqrt{2}}(|0, Zero\rangle_{AB} + |1, One\rangle_{AB}). \tag{2.14}$$

But we could just as well say that the collapse dynamics will govern the interaction, giving us either the state $|0, Zero\rangle_{AB}$ or the state $|1, One\rangle_{AB}$ but not the superposition state. So the collapse dynamics will put the system in a different state than the linear dynamics will. The eigenstate-eigenvalue link implies that different observables will have determinate values based on which dynamics governs the process. The different dynamics will lead to different states, and the different states are empririically distinct. Thus, the two different dynamics contradict each other.[12]

Thus, we must partition any experiment into quantum systems which follow the linear dynamics and observer systems which measure the quantum systems and cause them to collapse.

[12]There are cases when the two dynamics will give the same result. For instance, if A began in the state $|0\rangle_A$, then both the linear dynamics and the collapse dynamics for a measurement in the standard basis would keep the state the same.

A scientist may obviously be an observer system, and a single electron may obviously be a quantum system, but where between them lies the cutoff? If a particle that interacts with an N-particle system collapses, what about when it interacts with an $N-1$ particle system? An $N-2$ particle system? For the von Neumann-Dirac Formulation, the Sorites paradox is not merely semantic; there are dynamical consequences of how we categorize physical systems.

Unfortunately, the von Neumann-Dirac Formulation does not prescribe any way of categorizing systems into those whose interaction with another system will cause a collapse and those whose whose interaction will not. It makes no stipulation against you and I disagreeing about when a collapse occurs because it does not tell us when collapses will occur. It gives two dynamics which contradict each other and does not prescribe a way to separate the cases where each applies, without which it is inconsistent. If we assume there is a correct partitioning of cases where only one of the dynamics applies, then the physical theory is incomplete.

But any explicit cutoff to the view would be glaringly arbitrary. Consider a set of physical systems $\{S_1, S_2, ..., S_n\}$ where S_i is a system with i particles, and S_n is a system which, when it interacts with another system, causes that system to collapse. An obvious way of deciding which dynamics applies to each of these systems would be to select some j such that a particle interacting with S_j continues to follow the linear dynamics and a particle interacting with S_{j+1} collapses. This leads to unseemly scenarios, such as when the quantum system S_1 entangles with S_j to suddenly produce a system which is governed by a completely different dynamics than S_j itself. In such a case, S_1 measures S_j as much as the converse. The cutoff point has no principled physical justification and suggests a discrete change in the governing dynamics unlike what we typically see in nature. More, since particles can be entangled in one observable but not entangled in another, a definite cutoff point would require us to stipulate a preferred observable in terms of which the sizes of systems, and thus the dynamics governing them, is determined.

The view proposes two different dynamics that give contradictory outcomes. If both of these dynamics hold for all cases, the view is inconsistent. So the view needs to partition the universe into cases where each dynamics applies. But it does not tell us how to categorize systems into quantum systems that follow the linear dynamics and observer systems which trigger the collapse dynamics, so it is incomplete. Thus, in its current state, it is either inconsistent (if we assume that both dynamics apply in all cases) or incomplete (if we assume the two dynamics apply to mutually exclusive sets of cases). Partitioning all physical systems to avoid this conundrum requires that we arbitrarily draw a hard line distinguishing the kinds of systems governed by each of the two dynamics. This is not only blatantly ad hoc and physically unprincipled but also unnecessary given that we have a collapse view of quantum mechanics, GRW, in which there is a gradual transition between systems that(typically) follow the linear dynamics and those that (typically) follow the collapse dynamics.

Thus standard models of quantum computation arise from a physical theory which does not makes sense and uses the aspects of the theory (its two dynamics) which are responsible for its woes. This alone is reason for concern, but the existence of consistent alternatives to the von Neumann-Dirac Formulation means that we are using a model of computation based on a poor physical theory even when we have superior alternatives. The von Neumann-Dirac Formulation is not forced upon us because we lack working alternative accounts of quantum mechanics. If this were the case, its ubiquity in quantum computing would be more understandable, as it would represent our best working approximation of how a quantum computer could operate. But we have other choices which do not suffer from the damning problems of the von Neumann-Dirac Formulation, so the use of the von Neumann-Dirac Formulation cannot be justified in this way.

More, these alternatives differ from the von Neumann-Dirac Formulation in ways that imply differences in the corresponding model of computation. For instance, not all alternatives even have a collapse dynamics (e.g. Bohmian Mechanics), and those that do may treat it

significantly differently than the von Neumann-Dirac formulation does (e.g. GRW). The GRW collapse theory weakens the eigenstate-eigenvalue link, opts for spontaneous collisions in position, and makes precise when and how collapses occur in a way that may affect what quantum computers will be able to do. Bohmian Mechanics differs from the von Neumann-Dirac formulation by completely forsaking any collapse dynamics and instead giving particles definite positions at all times. Neither alternative collapse theories nor theories without collapse would allow the model of quantum computation we saw in 2.2.3.

2.4 GRW Quantum Mechanics

GRW quantum mechanics was first proposed in 1985 (33). It makes explicit when and how wavefunction collapses occur. GRW lacks the arbitrariness of the von Neumann-Dirac Formulation and consequently introduces sources for computational noise that do not exist in other formulations of quantum mechanics.

Section 2.4.1 describes the GRW picture in the format of 2.2.1. Unlike in the von Neumann-Dirac Formulation, in GRW a hard division between quantum systems that obey the linear dynamics and measuring systems that trigger the collapse dynamics is unnecessary. To achieve this, GRW adds spontaneous collapses, and, to accommodate the spontaneity of collapses, it makes all collapses collapses in position. Section 2.4.2 describes quantum computing in GRW and how it differs from quantum computing in the von Neumann-Dirac Formulation. Because of the way GRW bridges the divide between when systems evolve linearly or collapse, it can put theoretical limits on the reliability of a quantum computer, depending on its physical parameters.

2.4.1 The GRW Formulations of Quantum Mechanics

The GRW formulation of quantum mechanics is a collapse view which avoids the ambiguity of the von Neumann-Dirac picture by explicitly adding collapses into the dynamics. In it collapses occur spontaneously, rather than at particular types of events (such as a given type of inter-particle interaction). In my presentation, I will assume the original presentation of GRW that does not specify a primitive ontology, but I will state in footnotes how the two popular primitive ontology views of GRW, the GRWm "mass density" view and the GRWf "flash" view, differ from the theory I present. These differences will not be significant for the GRW model of quantum computing.

For a given particle in GRW, there is an mean time period τ between collapses, where τ is a new physical constant (on the order of magnitude of 10^{15}s). In these collapses, the wavefunction becomes a narrow Gaussian with a width determined by another new physical constant, α (on the order of magnitude of 10^{-7}m). The values of these constants are set so that GRW is in agreement with all known experimental results.[13]

In this picture, the wavefunction is real and particles consist of their wavefunctions. Both 1 and 2 of the von Neumann-Dirac Formulation are maintained in GRW.

G1. States of physical systems are represented by unit length vectors $|\psi\rangle$ in a Hilbert space $\mathcal{H}$.

G2. Physical observables O are represented by Hermitian operator $\hat{O}$, and likewise Hermitian operators represents physical observables.

Differences begin to arise in the interpretation of the wavefunction. In GRW, having a

[13]GRW always differs empirically from the von Neumann-Dirac Formulation (insofar as it is used to make particular empirical predictions) and Bohmian Mechanics as its collapses will slightly increase the energy of systems. As experiments become more precise, these constants will have to be fine-tuned to avoid conflict with experimental results.

wavefunction that is almost an eigenstate of an observable is sufficient to have a value for that observable. This ensures that a particle has a position when its wavefunction is narrowly concentrated in a small area, so the wavefunction does not have to be a delta function for a particle to have a definite position. If the wavefunction were a delta function the energy of the particle would blow up. The value of α, which determines the width of the collapsed wavefunctions, is chosen to be large enough that the energy of systems does not noticeably increase when they collapse but small enough that the narrow Gaussians still look like particles positioned at the Gaussian's peak.

G3. Being close to an eigenstate of an observable is sufficient for a system have a definite value for that observable.

For instance, we can consider a system that is in a state that is a narrow Gaussian centered around point x_0 to have position $x = x_0$.[14] Like in the von Neumann-Dirac formulation, wavefunctions in GRW will typically follow the linear dynamics and will occasionally undergo collapses. But the nature of and triggers for collapses in GRW are quite different from those in the von Neumann-Dirac formulation. Collapses in GRW are not collapses to an eigenstate of any measurement observable; they are always in position and whatever observables are entangled with position. GRW collapses occur spontaneously with a mean period of $\tau \sim 10^{15}$s for an isolated particle. This makes the applications of the linear and the collapse dynamics unambiguous. The linear dynamics typically holds, but the collapse dynamics will occasionally manifest to turn the wavefunction into a narrow Gaussian. The collapse dynamics is not triggered by a particular (but undefined) kind of "measurement" event, though we will see that physical circumstances of a measurement will lead to frequent

[14]GRWm differs slightly on this point, as it distributes matter according to the wavefunction. So when the wavefunction is "close" to an eigenstate of position, matter is densely concentrated near the center of the collapsed Gaussian and thus appears like a particle with a defined position. GRWf differs more significantly, as it places "flashes", which are points in spacetime, at the center of the collapsed wavefunction; matter in the universe consists entirely of these flashes. The ontology of the theory is the flashes, so there is no need for an eigenstate-eigenvalue link.

collapses; rather, collapses occur periodically.

G4. Systems are governed by a stochastic dynamics with two parts.

A. Physical systems typically evolve unitarily. Where $\hat{U}$ is some unitary transformation, from time t_0 to time t_1, the state will go from $|\psi(t_0)\rangle$ to $\hat{U}(t_0, t_1)|\psi(t_0)\rangle$.

B. Systems will spontaneously collapse and their states will be multiplied by narrow Gaussians of width determined by $\alpha \sim 10^{-7}$m, with a frequency of $\frac{N}{\tau}$, where N is the number of entangled particles in the systems and $\tau \sim 10^{15}$s is a constant. This amounts to a collapse in position by G3.[15] Entangled particles will collapse together, so the collapse of one of the particles in an entangled state implies the collapse of it partners.

Compositions of systems are treated the same way as in the von Neumann-Dirac formulation.

G5. Compositions of systems are represented by tensor products. So the system composed of S_1 with state $|\psi\rangle \in \mathcal{H}_1$ and S_2 with state $|\phi\rangle \in \mathcal{H}_2$ is $|\psi\rangle \otimes |\phi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$. Likewise for properties of systems.

So the collapsed Gaussian wavefunction must be narrow enough to give the appearance of a particle but wide enough that the energy of the system is not noticeably increased. The location of the collapse in spacetime is where the "particle" is found; particles are just their wavefunctions and so do not ever occupy definite positions. Position thus plays a privileged role among the observables in this view, as collapses are inherently in position. Thus, any measurement in GRW must ultimately be a position measurement, and a measurement device works by entangling its position with the observable of the system it supposed to detect.

[15] In GRWf, flashes are places at the centers of these collapses. Collapses take place in the wavefunction that is generating the flashes, but the wavefunction itself does not have ontological significance.

The behavioral differences between microscopic quantum bodies and macroscopic "classical" bodies emerge because all the particles in a many-particle entangled system will collapse when one of the particles collapses. As the frequency of collapse of the system is $\frac{N}{\tau}$, macroscopic bodies that consist of many particles entangled together will collapse very frequently. These frequent collapses keep the particles of a macroscopic body fairly constrained in position so that the collection of particles remains stable.

Measurements can take place when a small quantum system entangles with a large detector system. The joint system will have a collapse rate determined by its number of particles, and the positions of its particles when it collapses will indicate information about the small system. The joint system's collapse rate is determined by its number of particles, as the flashes for a physical system are described by a Poisson distribution:

$$p(m) = \frac{e^{-\frac{N}{\tau}t}(-\frac{N}{\tau}t)^m}{m!} \tag{2.15}$$

where t is a time interval and m is the number of collapses that occur in interval t (30). Thus, we have a precise characterization of how macroscopic measurement devices differ from quantum systems and when we can expect measurement or quantum behavior to occur. Unlike in the von Neumann-Dirac formulation, we do not have to treat measuring devices as wholly different kinds of entities from quantum systems, and whether the linear or collapse dynamics applies to any given situation is not an arbitrary or ambiguous matter. Instead, the same rules apply to all systems, and the behavior of large systems such as measurement devices naturally arises out of their size in terms of their number of particles. The von Neumann-Dirac Formulation's measurement problem regarding when the linear or collapse dynamics apply is solved.[16]

[16]One may think of GRW as making precise our working intuitions about how von Neumann-Dirac quantum mechanics should work. When using the von Neumann-Dirac formulation, we still expect systems of a certain size to act "classically" and be able to cause quantum systems to collapse. A von Neumann-Dirac fan will expect collapses to occur in mostly the same situations as a GRW enthusiast. Yet the intuitions of scientists about when collapses occur are not part of the theory, and only GRW gives an explicit way for the quantum mechanics to match these expectations.

2.4.2 GRW Computing

In many ways, qubits in a model of quantum computing in GRW will be the same as those in standard models of quantum computing. Qubits in GRW computing will have state spaces that are Hilbert spaces and states that can be expressed in some standard basis, just like the qubits in standard quantum computing. As the linear dynamics does not change between the von Neumann-Dirac formulation and GRW, each corresponding model of quantum computing will treat unitary transformations the same, with the caveat that qubits in GRW may spontaneously collapse during a unitary transformation via the collapse dynamics. Likewise, there is a small chance that qubits will not collapse during a measurement, but this probability can be made negligible by using a large enough measuring device. The specifications for a physical implementation of measurement and reset operators differ slightly between the two models, as measurements in GRW require entanglements in position. However, all this means is that whatever observable of a qubit one wants to measure ought to be entangled with the position of some large object. So a GRW model of quantum computing will have similar operations as a standard model, with slight differences due to the way GRW triggers collapses. Let us look at how these operations will differ.

Because collapses occur with a frequency based on the number of particles entangled together, to minimize the risk that the qubits spontaneously collapse when an algorithm requires them to evolve unitarily one would be incentivized to choose small physical systems to act as qubits in a physical quantum computer. GRW qubits will not exclusively follow the linear dynamics, though if they are small they will collapse extremely rarely unless many of them are entangled together. The risk of qubits collapsing during unitary transformations can not be wholly eliminated, though. Computational steps involving only unitary transformations cannot be guaranteed to be unitary in GRW computing, which leads to notable differences between the unitary transformations in standard quantum computing and those in GRW computing.

First, unitary transformations cannot be guaranteed to be reversible, as the qubit systems could spontaneously collapse to an eigenstate of position partway through the computational steps or their reverse.[17] Again, since GRW quantifies how frequently a qubit or a qubit system will collapse based on the number of particles in it, if the qubit system used in the computational steps consists of few particles the risk of losing reversibility is small to negligible.

Second, GRW gives theoretical limits on the reliability of quantum computers which depend on the way in which the computers are constructed. Computational processes that require large numbers of qubits to be entangled together risk spontaneous collapses during unitary transformations. Because these collapses occur at a frequency determined by the number of entangled particles in a system, huge qubit systems may be likely to prematurely collapse in position. These collapses could ruin the result of the computation, depending on the basis used as the standard basis of the qubits. If, for instance, the standard basis is spin, then a collapse in position will likely not affect the computation unless a qubit's spin is entangled with the position of some particle, but if the standard basis is position, then a collapse may ruin the computation if the collapsing qubit had been in a superposition of position.[18]

For instance, imagine a spin-$\frac{1}{2}$ particle is being used as a qubit such that $|\uparrow\rangle = |0\rangle$ and $|\downarrow\rangle = |1\rangle$. Imagine also that it is being acted on by some unitary transformation that is physically implemented by a uniform magnetic field; the particle will have been sent into this field regardless of its spin value. Should the particle collapse in position yet remain within the uniform magnetic field, the computation will not be disturbed, as the magnetic field will continue to affect the particle's spin as though the particle had not collapsed. However, if the particle collapses outside of the magnetic field, the unitary transformation

[17]This implies, for instance. that quantum key distribution cannot guarantee (with probability one) secure communication between two parties because the transmitted key may collapse before the receiving party can read it.

[18]Similarly, a unitary transformation may remain reversible despite a spontaneous collapse if the standard basis of the qubit on which it acts is neither position nor is entangled with position.

will not be performed on the qubit, and the computation will be ruined. No matter how large the uniform magnetic field is, if its spatial dimensions are finite, there is a nonzero chance that the spin-$\frac{1}{2}$ particle will collapse to a position outside of it. However, the extremely small probability that the particle will collapse outside of a certain region implies that the effect of unwanted spontaneous collapses on the reliability of this unitary transformation is negligible for this setup. But if the particle had been sent through a Stern-Gerlach apparatus that would send the particle through the magnetic field only if its spin were $|0\rangle$, and the particle had been in some superposition of $|0\rangle$ and $|1\rangle$, then a spontaneous collapse in position would likely ruin the computation, as the particle's spin is entangled with its own position. The theoretical limits GRW's dynamics put on the reliability of computers can thus be at least partially avoided by constructing a quantum computer whose qubits' standard basis is not position and which do not have their standard basis entangled with the position of any system (including themselves). The noise arising directly from GRW's dynamics would be difficult if not impossible to avoid, even if computers could be designed specifically to minimize it.[19]

Let us illustrate the noise produced by GRW's dynamics. For simplicity, assume that either the standard basis for qubits is position or the standard basis of the qubits is always entangled with their positions in such a way that any spontaneous collapse of any individual qubit causes an error. We can quantify how likely a collapse is to occur in a physical implementation of a set of unitary transformations if we know the processing speed of the quantum computer (the number of unitary operations it can perform on its qubits per second), the number of particles in each qubit, the number of qubits that will be involved in the computation, and the number of unitary transformations that will be performed on the qubits before the computation is over and its result is read via a some measurement operator. Increasing

[19]Consider one way one could try to build a model of GRW computing that avoids the noise arising from GRW's dynamics. One could, for instance, put our spin-$\frac{1}{2}$ particle in a universe which is completely filled with a uniform magnetic field to ensure that the unitary transformation performed by that field will succeed, but such a setup is both unphysical and only lets us perform trivial computations such as computing NOT by having the field apply an X transformation.

processor speed and decreasing the size of qubits and the number of unitary operations done in an algorithm can increase the reliability of a computation. But the noise arising from GRW's dynamics cannot be completely avoided in this way, as processing speed cannot be infinite, qubits must have at least one particle, and any string of unitary transformations can obviously be reduced to one transformation at minimum.

Let us get a rough idea of how this error might play out. Recall that $\tau \sim 10^{15}$s is the average time between collapses for a single particle system. Let n_q be the number of qubits in a qubit system, n_p be the number of particles per qubit (which are all entangled in position), S be the processing speed of our computer (the number of unitary transformations the computer can apply to qubits per second), and n_U be the the number of unitary transformations performed in a quantum computation. Assume that after the application of each unitary transformation, the system implementing the transformation is immediately disentangled from the qubit system on which it acts. So the total number of particles in the qubit system is $n_q n_p$ and (from G4B) the frequency of collapses of some part of the qubit system is

$$\frac{n_q n_p}{\tau}, \tag{2.16}$$

and the time the computation would take is

$$\frac{n_U}{S}. \tag{2.17}$$

Assuming the collapses for any given physical system are distributed in time according to a Poisson distribution in GRW gives

$$p(E = m) = \frac{e^{-\lambda t}(-\lambda t)^m}{m!} \tag{2.18}$$

where t is the duration of time in which events E can occur, m is some natural number, and

λ is "the parameter of the distribution" (30). So the time of computation is

$$t = \frac{n_U}{S}, \tag{2.19}$$

and the parameter of the distribution, the number of collapses per unit time, is

$$\lambda = \frac{n_q n_p}{\tau}. \tag{2.20}$$

Without any fault tolerance or quantum erroc correction procedures, this give us the probability that a set of n_U unitary transformations will succeed ($m = 0$) as

$$p(E = 0) = e^{-n_q n_p n_U / \tau S}. \tag{2.21}$$

Because we have assumed the unintended collapse of just one of the qubits can lead to an error, (2.21) holds for qubit systems whose standard basis is or is entangled with position, regardless of how or even if the qubit systems are entangled with each other. For those GRW computers whose qubit standard basis is not position nor is always entangled with position, (2.21) would represent the probability of success for any set of n_U unitary transformations on n_q qubits while their standard basis *is* entangled with position. If p is the desired probability that our computation will succeed, the maximum number of qubits for our idealized GRW computer is

$$n_q = \frac{\tau S}{n_p n_U} log(p). \tag{2.22}$$

Again, for quantum computers whose standard basis is neither position nor always entangled with position, n_q is the number of qubits which are entangled in position (or entangled with their own positions) during n_U unitary transformations. Note that even in this simple setup this limit is inseparable from the physical parameters of the quantum computer. For instance,

for physical qubit systems like photons, which are single-particle qubits that move quickly — and hence would allow many unitary transformations to be performed on them in a set amount of time — n_q can be very high. But for much larger physical implementations of qubits like Josephson junctions, this limit would be considerably smaller.

We can imagine how the noise arising from GRW's dynamics might affect potential uses of Shor's algorithm. For any given GRW computer that can run Shor's algorithm, it may only be able to reliably factor numbers up to a maximum size before GRW noise begins to frequently interfere with its computations, as larger numbers of qubit systems are necessary to factor larger numbers. Additionally, a GRW computer which uses position as the standard basis for qubits will have a lower limit on the computations it can run than a GRW computer which uses spin as the standard basis for qubits and which does not always have the spins of these qubits entangled with their positions. This implies that two GRW computers that can implement Shor's algorithm and which are equal in terms of processing speed and the number of particles in their qubits may differ in how reliably they can factor large numbers depending on how susceptible to collapses in position they are. Likewise, any attempt to use quantum computing to test the value of τ (by setting the physical parameters of a quantum computer and testing how often a set of unitary transformations succeeds according to (2.21)) would need to consider what the standard basis of the qubits is and during how many unitary transformations it is entangled with position.

Aaronson (1) has noted GRW likely allows "fairly large quantum computers" (p. 121). I will further investigate the limits on computation that arise from GRW's dynamics and what they tell us about quantum computing, physical computation, and GRW itself throughout much of the rest of this thesis. The von Neumann-Dirac Formulation gives no such limits and thus allows qubit systems of arbitrarily large size to be used in quantum computing no matter how the quantum computer is constructed. Therefore, the size of tasks which can be done by standard quantum computers is theoretically unlimited (even if in practice it may

be quite limited). We could use qubit states of arbitrarily large size because nothing in the theory tells us that the collapse dynamics will not allow systems of arbitrarily large size to follow the linear dynamics; a unitarily evolving qubit system the size of a galaxy would be no less principled than one consisting of a dozen particles. We will see that the same is true of Bohmian mechanics.

2.5 Bohmian Mechanics

In contrast to the two mutually exclusive dynamics of the von Neumann-Dirac Formulation, Bohmian Mechanics has one dynamics consisting of two parts — one which governs the evolution of the wavefunction and one which governs the movement of particles. Rather than having the state collapse in measurements, particles in Bohmian Mechanics are given definite positions at all times, so that they have real trajectories through spacetime. Thus, unlike particles in the von Neumann-Dirac Formulation, particles in Bohmian Mechanics can be described as curves through spacetime. All measurements in Bohmian Mechanics are ultimately position measurements, as position will be the only observable that takes definite values.

Section 6.1 discusses Bohmian Mechanics and how it differs from the von Neumann-Dirac Formulation and GRW. Section 6.2 looks at what a model of computation in Bohmian Mechanics might look like.

2.5.1 The Theory of Bohmian Mechanics

Bohmian Mechanics adds a guiding equation and a set of positions for all particles to the linear dynamics of the von Neumann-Dirac Formulation. Position is made the preferred observable, meaning that particles have definite positions at all times but do not have definite

values for the other observables. Bohmian Mechanics is as follows.

The physical state of a system S in Bohmian Mechanics is not just the state vector $|\psi\rangle_S$ (as seen in the von Neumann-Dirac formulation), but rather the state vector along with the configuration Q_S. The configuration is the position of every particle, placed in configuration space $\mathbb{R}^{3n}$, where n is the total number of particles. So we have

B1. States of physical systems are represented by unit length vectors $|\psi\rangle$ along with a configuration Q_S in $\mathbb{R}^{3n}$, where n is the number of particles.

As positions are always determinate, observables other than position are defined in terms of position, and all measurements are measurements in position. For instance, particles do not have a spin value that one can directly measure. Instead, one must measure the position of a particle after it goes through a suitable magnetic field, e.g. a Stern-Gerlach apparatus.

B2. All measurements are measurements of position.

B3. Positions of all particles are always determinate. (The eigenstate-eigenvalue link is dropped.)

B4A. The linear dynamics of 4A govern a system's state vector $|\psi\rangle$ at all times.

Bohmian Mechanics uses the same linear dynamics as the von Neumann-Dirac Formulation, except with no qualification for measurement. But to the linear dynamics it adds a guiding equation. The role of the guiding equation is to give trajectories for all particles so that if their distribution agrees with the empirical quantum statistics at one time, it will agree with the statistics at all other times. Bohmian mechanics models the trajectories of particles as though they were massless particles being pushed around by the evolution of $|\psi|^2$. So we have:

B4B. The configuration evolves according to

$$\frac{dQ_k}{dt} = \frac{Im(\psi^* \nabla_k \psi)}{m_k \psi^* \psi} \tag{2.23}$$

which shows the flow of particle density out of a unit of volume (10) (see also (17)). So like the von Neumann-Dirac Formulation, Bohmian Mechanics has two main parts to its dynamics. However, unlike the von Neumann-Dirac formulation, both of these parts of the dynamics govern all physical systems at all times and are not mutually exclusive.

We still need to ensure that Bohmian Mechanics fits the empirical results we see in quantum mechanics, so we need to make sure that particles are distributed accordingly. If the particles are initially distributed so that the particle density $\rho(x, t_i)$ satisfies

$$\rho(x, t_i) = |\psi(x, t_i)|^2, \tag{2.24}$$

then the particles will continue to be distributed according to the statistics given by the wavefunction, so $\rho(x, t) = |\psi(x, t)|^2$ for all t (52). So we stipulate

B5. At some time t_i, particles are distributed according to (2.24).

For any wavefunction, there are a host of configurations of particles that would satisfy 2.24. Though the Bohmian dynamics is wholly deterministic, the apparently probabilistic results we see in quantum mechanics are a result of our ignorance about the actual configuration of particles in the universe.

The state of a physical system in Bohmian Mecahnics includes both the wavefunction $|\psi\rangle$ and positions for particles, rather than just the wavefunction. Positions for particles are thus always determinate, meaning that the eigenstate-eigenvalue link does not hold and all measurements are of position. This means that we cannot obtain values for properties other

than position without correlating them with position. So, for instance, obtaining a spin value for a particle requires entangling spin with position (e.g. with the use of a Stern-Gerlach apparatus). These position measurements, however, do not involve a dynamics different from the linear dynamics. Particles always evolve according to the wavefunction; they act as massless particles pushed about by the magnitude squared of the wavefunction and are distributed so that they agree with the empirical quantum statistics.

So executing a measurement in Bohmian Mechanics requires one to entangle the position of some pointer system (which acts as a measuring apparatus) with whatever property of whatever system one wants to measure.[20] This must be done so that the spatial regions occupied by the pointer's wave packet corresponding to different values of the measured system are completely disjoint. Doing this ensures that the pointer will reliably indicate the value of the measured system.

For instance, measuring a qubit requires one to entangle the position of a pointer with the standard basis of the qubit; the positions of the particles in the pointer are then entangled with the value of the qubit in the measurement basis. If the qubit is $|0\rangle$, the pointer's particles will be in some region S_0, and if the qubit is $|1\rangle$, the qubit will be in some different region S_1, such that $S_0 \cap S_1 = \emptyset$. (See (52), 1.4.2.) Though this process does not involve a change in which dynamics governs a process, the Hamiltonian for the qubit and pointer system will lead to a different linear evolution than the Hamiltonian for just the qubit system. Measuring a system will still change how it evolves but will not change the dynamical rules governing it.

[20]Note that, as far as we know, Bohmian Mechanics is empirically equivalent to the von Neumann-Dirac Formulation, and position entanglements will arise naturally in realistic measurement scenarios.

2.5.2 Bohmian Computing

Qubits in Bohmian mechanics will differ from those in the collapse theories in that they will always have definite position, so no matter what basis is chosen as the standard basis for the qubits, measuring them requires entangling their state in the standard basis with the position of some measuring apparatus. Also, Bohmian mechanics lacks the separate collapse dynamics of the measurement and reset operators, so those processes must be performed in a dynamically different way than in the von Neumann-Dirac formulation or GRW. These differences between the physical theories imply theoretical differences between their respective models of quantum computation.

Qubits in the von Neumann-Dirac formulation consist of physical systems with states that are completely represented by unit length vectors $|\psi\rangle$. These have definite values for observables only when they are in eigenstates of those observables, and collapses put them into eigenstates of an observable when they are measured in that observable. None of these things are true about Bohmian qubits. The physical state of a Bohmian qubit q is represented by both the wavefunction $|\psi\rangle$ and the configuration Q_q. Position is the only observable that ever has a definite value for these qubits — it has a definite value at all times —, and these qubits never undergo a collapse dynamics. Rather, their state vectors always evolve according to the linear dynamics and their configuration always evolves according to B4B. This evolution of the configuration means that unitary transformations will not only change the wavefunction, they will also change the configuration and hence the positions of qubits.

For any given wavefunction, there are many possible configurations that would satisfy equation 2.23 and thus many possible states for the qubit. However, not all of these states will be importantly unique for the sake of quantum computing. For instance, various configurations correspond to the single qubit state $|0\rangle$, but slightly altering the positions of the particles in the qubit (but still satisfying eq. 2.23) may not change the qubit in a way that is rele-

vant for quantum computing — for this variety of states, the qubit will still have the state analogous to the classical bit state 0. Likewise, whereas in the von Neumann-Dirac formulation our inability to extract more than one bit from a single qubit is due to the collapse dynamics putting the qubit in one of only two states when measured in the standard basis, in Bohmian computation we do not get this simple explanation for the limit on information carried by qubits. Whereas the single bit extracted from a qubit in the von Neumann-Dirac formulation represents all the information about the qubits state post-collapse, in Bohmian mechanics qubits always contain much more physical information than we are able to extract. Instead, the limits of qubits are a result of Bohmian Mechanics' measurement by entanglement; because a pointer's state can only entangle with the wavefunction of a qubit and not its configuration, the pointer's position cannot indicate information about the qubit's configuration that would allow one to extract more than one bit of information from a single qubit.

The lack of a collapse dynamics implies that a Bohmian model of computation cannot have the reset and measurement operators allowed by the von Neumann-Dirac model. Alternatives to these operators that rely only on the unitary transformations are needed for a Bohmian model of computation. We cannot simply use a measurement to collapse a qubit's state into any basis of our choosing, leaving the qubit in a new state and disentangled with the measuring device. The interaction of a measuring device with a qubit is a more complicated process in Bohmian computing than in standard quantum computing, as we must keep track of the Hamiltonian for the system composed of both the measuring apparatus and the qubit. Likewise, the work of the measurement and reset operators requires multiple procedural steps for properly entangling the qubit and pointer systems, possibly reading the results (which will require the entanglement of the joint pointer-qubit system with classical computer or even a scientist), and disentangling the qubit system from the other systems. These steps are all done using only the linear dynamics.

Consider a pointer system P which is in the state $|\psi_0\rangle_P = |S_0\rangle_P$. We will use this system to measure a qubit system q which is in some normalized state

$$|\phi\rangle_q = a|0\rangle_q + b|1\rangle_q. \tag{2.25}$$

To do this, we apply a unitary transformation U_e which acts according to

$$U_e(a|0\rangle_q + b|1\rangle_q)|\phi_0\rangle_P = a|0\rangle_q|S_0\rangle_P + b|1\rangle_q|S_1\rangle_P \tag{2.26}$$

to give an entangled state. The configuration lies within one of the two wave packets; S_0 and S_1 are again disjoint regions in spacetime. The outcome of the measurement is determined by which of these regions of spacetimes the configuration occupies. This unitary transformation entangles the qubit system in the standard basis with the pointer system in position so that the position of the pointer indicates the value of the qubit. The Bohmian measurement operator consists of this unitary transformation; an additional unitary transformation can be used to entangle the pointer with some system in the environment which can read the result.

Future unitary transformations in the computational process will be of the form

$$U = U_q \otimes I \tag{2.27}$$

so that the pointer is left in a state which indicates the result of the measurement (because the configuration will remain in one of the two wave packets $|S_0\rangle_P$ or $|S_1\rangle_P$) while the qubit can continue to be used in the computation. These unitary transformations can disentangle the qubit state from the pointer state.

Note that, though a standard model's measurement operator would put a qubit in an eigenstate of the measurement basis, the Bohmian measurement operator will almost never do so.

Instead, a measurement in Bohmian mechanics entangles the position of a pointer system to an observable of the measured system. The Bohmian measurement thus selects an "effective" wavefunction – the wavepacket in which the system's configuration is located. So whereas a collapse in the von Neumann-Dirac Formulation might leave the joint system given by (2.26) in the eigenstate $|0\rangle_1|S_0\rangle_P$, the Bohmian measurement process will not (except if $a = 1$ for the original qubit state). Hence, the qubit and pointer systems remain entangled after the application of the measurement or reset operators and can be disentangled via future unitary transformations of the form of (2.27).

Because the Bohmian measurement operator typically does not put qubits into eigenstates of the measurement basis, the Bohmian reset operator will typically not do so, either. So a Bohmian reset operator that sets qubits in some initial state requires a set of unitary transformations which act on the wavefunction to shift the *configuration* into some wave packet $|\psi_0\rangle_q|S_0\rangle_P$ or $|\psi_1\rangle_q|S_1\rangle_P$, where $|\psi_0\rangle_q$ and $|\psi_1\rangle_q$ form the basis of some new observable. Based on the location of the configuration of the pointer system, a transformation $X \otimes I$ may be applied to the system to put the qubit's configuration into the $|\psi_0\rangle$ wave packet. The wavefunction itself, however, will not necessarily be $|\psi_0\rangle$. The Bohmian reset operator thus uses unitary transformations to move the configuration into some specified wave packet rather than move the wavefunction into some specified state, and it consists of U_e and an $X \otimes I$ transformation that will only be performed if the pointer's configuration indicates that the qubit is not in the specified initial state.

Bohmian Mechanics has additional epistemic concerns. Because it cannot use wavefunction collapses to disentangle system states, it must do so using unitary operations. This means that creating a state *exactly* as in (2.25) would require one to know all the entanglements the qubit system has with all other systems in the universe and then apply unitary operations to disentangle it from all the entangled systems.[21]

[21] Though for practical purposes, most of these entanglements will likely not cause notable problems for a real computer.

So the prescription in Shor's algorithm to measure a qubit in the standard basis would involve entangling the qubit's state in the standard basis with the position of another system, (possibly) reading the result by applying a unitary transformation to the qubit-pointer-environment state to entangle the qubit-pointer system with some part of the environment, then (possibly) disentangling the qubit's state in the standard basis from the positions of all other systems. All these processes use only the linear dynamics. Likewise, a Bohmian reset operator would use the same process except that the result of the "measurement" is read (possibly by some classical computer), and, based on the result, an $X \otimes I$ transformation may be applied to the qubit-pointer state. Both our measurement and reset operators would be special cases of unitary transformations instead of a distinct class of computing tasks.

Thus, a Bohmian model of quantum computation must differ from a von Neumann-Dirac model or a GRW model. Even its qubits, the building blocks of quantum computation, consist of different mathematical objects and behave in different ways than standard model qubits. It will lack the collapsed-based measurement and reset operators of standard models and will instead use only unitary transformations. Getting the effects of measurement requires bringing measurement apparatuses into the model,[22] as measurements can only be achieved by modeling the entire entangled qubit-pointer system in the dynamics. For a more thorough discussion of how quantum computational processes would work in Bohmian Mechanics, see (62)(63).

[22]As we have seen, the GRW computing model also includes references to measuring devices.

2.6 Comparing Quantum Computing in the Different Formulations

These three quantum theories give different accounts of the physical world and hence allow different models of quantum computing with different understandings of operations and qubits. Let us look at some of the ways in which these models compare.

Imagine we are given the power to make a quantum computer in absolute isolation; this idealized computer inhabits its own universe. The practical problem of preventing accidental entanglements between qubits and the environment is completely mitigated. The computer is free to perform unitary operations, and we are free to poke into the universe at any time to perform a measurement. All operations of this physical computer are subject only to the limitations of the model of computation which it implements. How large can the computer we make be without it losing its ability to do quantum computations? How many qubits could we put into a qubit system on which this computer performs unitary transformations? Our presented models of quantum computing give different answers.

The von Neumann-Dirac Formulation does not specify whether any system will follow the collapse or linear dynamics at any given time, so in its corresponding model of quantum computing we are free to stipulate that any given process will be a unitary transformations or a measurement operation. Thus, we are free to make our quantum computer arbitrarily large and use arbitrarily large entangled qubit systems in our computer. This may contradict the intuitions that guide the use of the von Neumann-Dirac Formulation, but the formulation itself does not contain these intuitions.

Similarly, Bohmian mechanics, with its constant use of the linear dynamics, will allow us to make our quantum computer as large as we like. Measurements occur by entanglement with outside systems; if we completely isolate our computer, we eliminate the risk that outside

systems will accidentally entangle with our qubits and ruin our computation, and thus we can make the computer as large as we like. So neither the standard nor the Bohmian model of quantum computation places any theoretical limits on the power of quantum computers. The only limit is a practical one that follows from the linear dynamics — how well can we isolate our quantum computer from the outside world to ensure that our qubits don't accidentally become entangled with the environment?

The case of GRW is more complicated. Because GRW specifies the circumstances in which collapses occur — periodically, but with a period that decreases based on the number of entangled particles in a system — a GRW model of quantum computing lacks the freedom of a standard model to choose to let arbitrarily large qubit systems follow the linear dynamics. The larger we make our qubit systems, the more likely they are to spontaneously collapse before our computation has completed all its unitary transformations, even in this idealized setup. Noise is an inherent part of what it is to be a model of computation in GRW. This is not the case in the von Neumann-Dirac formulation or Bohmian Mechanics, and thus GRW produces a different *kind* of noise for models of computation than either of those formulations of quantum mechanics. All three have practical limits based on how well we can isolate our quantum computer, but GRW's dynamices produces noise for quantum computing that still applies even if practical limits are completely avoided. GRW's dynamics incentivize certain ways of constructing quantum computers that the other two models do not. For instance, a GRW computer which uses qubits whose standard basis is not position nor is always entangled with position can be more powerful than an otherwise similar GRW computer which uses qubits whose standard basis is position or is always entangled with position, for this reason alone. And error correction and fault tolerance have a role in even idealized models of computation in GRW. This is not true in the von Neumann-Dirac or or Bohmian formulations.

Relatedly, unitary transformations are not guaranteed to be reversible in GRW. A GRW

qubit that has undergone a unitary transformation U may spontaneously collapse before the inverse U^{-1} of that transformation can be performed on it. GRW computing is not reversible full-stop; it is probabilistically reversible with a probability that depends on the size of a qubit system, the number of unitary transformations performed on the system, and the physical implementation of the qubit system.

Now consider the measurement and reset operators. In both the standard and GRW models, these use the collapse dynamics. A standard model simply stipulates that the collapse dynamics will occur in these operators; implementing them physically relies on a designer to judge that the collapse dynamics will govern at a particular step. The measurement operator will put the qubit system in an eigenstate of the measured observable. The GRW model, with its explicit dependence on the physical parameters of a quantum computer, gives an explicit account of the degree to which we should expect the collapse dynamics to arise at any particular stage in a physical quantum computer. The GRW measurement operator and reset operators put the qubit in a near eigenstate of the measurement observable. In the Bohmian model, measurement is done unitarily by entangling the position of a pointer system with the qubit property to be measured. The Bohmian measurement operator does not put the qubit in an eigenstate, but the wave packet in which the configuration is found indicates the value of the measured property.

Among the three approaches, whether a measurement occurs within a given time is a different kind of issue. In GRW, it is inherently probabilistic, with lower probabilities for small pointer systems and higher probabilities for large pointer systems. In principle we can never be certain that our qubits are following the linear or measurement dynamics in unitary or measurement operators, though we can be confident that they will behave appropriately as long as our systems are appropriately scaled. The size of an entangled system determines how likely we are to see a collapse, so we can stipulate that the system that measures the qubits has enough particles that the probability of collapse is, for all practical purposes, one, and

thus we can assume that a measurement operator has been applied. In Bohmian Mechanics, whether a measurement occurs is wholly deterministic and based on whether particular kinds of entanglement have occured. As we have noted, however, the von Neumann-Dirac formulation is deficient in that it does not specify when measurements will occur. It leaves open to the trials of experimentalists which dynamics will govern any particular interaction and is at best an incomplete – though perhaps pragmatically useful – description of any actual quantum computer.

As one should expect, the fundamental features of quantum computing are nonetheless shared among these three formulations of quantum mechanics. All these models have unitary transformations and treat them essentially the same. Any processes done with unitary transformations will look similar in the three models, the set of possible unitary transformations will be the same in all three models, and unitary transformations will be (typically, in GRW) reversible for qubit systems. The information capacity of qubits will be the same; only one qubit can be extracted from a classical bit even in Bohmian Mechanics where the configuration provides degrees of freedom that can physically distinguish qubits that are in the same state. And all these models will be able to give some kind of measurement and reset operators which allow other physical systems to obtain information about qubit systems, though we have seen that the procedure that accomplishes this will look different in each model

2.7 Remarks

As we will see more thoroughly throughout the rest of this project, the models of quantum computation we can construct differ depending on the theory of quantum mechanics we are using; models of quantum computation in Bohmian Mechanics, in GRW, and in the von Neumann-Dirac Formulation will differ in notable ways. For instance, presentations of

quantum algorithms such as Shor's factoring algorithm often explicitly rely on operations which will not be treated the same way and may not exist in the same form in computing models within different quantum theories. GRW in particular will produce noise in a way that would not be seen in other formulations of quantum mechanics. The dependence of models of quantum computation on the selection of a physical theory does not leave us with a diverse set of equally deserving models of computation. Rather, a model of quantum computation based on the von Neumann-Dirac Formulation relies on a physical theory that is either inconsistent or incomplete when there are complete and consistent alternatives, and these problems echo in the corresponding model of computation. We will look in detail at the GRW formulation of quantum mechanics.

One may contend that the problems of the von Neumann-Dirac formulation are not relevant to computing or represent impractical concerns that can be safely ignored by those working in quantum computing. Given our early successes in implementing quantum algorithms, these claims are not without merit; standard models of quantum computation have proven to be more than adequate given our limited ability to construct physical quantum computers. However, as the field of quantum computing gains robustness and precision, we will have to face questions about what operators we can implement and what the theoretical capabilities of quantum computers are. We will also need increasingly robust understanding of the sources and nature of quantum computational noise. These considerations will require us to tackle interpretational questions, and the quantum theory we choose will determine what our models of quantum computation can be.

Chapter 3

Computation in GRW & Lessons for the Aitken/Barrett Thesis

3.1 Introduction

The possible power of physical computers ought to depend on the nature of physical reality and, given the nature of physical reality, the way in which computers are constructed. In (4), Aitken and Barrett codify these intuitions. They use a Turing machine in a hyperreal spacetime to show that questions of physical computability must be answered with respect to a selection of a computational model relative to a particular physical theory (4). They then observe that questions of physical computation must be answered relative to a model of computation selected relative to a precise physical theory. I call this prescription the "Aitken/Barrett thesis". I will apply this prescription to the case of quantum mechanics and quantum computing and discuss the considerations involved in and lessons learned from applying it to GRW quantum mechanics, a collapse formulation of quantum mechanics in which wavefunction collapses occur spontaneously.

Accounts of quantum computing typically rely on some version of the von Neumann-Dirac collapse formulation of quantum mechanics.[1] Due to the measurement problem, this formulation results in a physical theory that is either inconsistent or incomplete, as it proposes two incompatible dynamical laws for systems but does not specify the cases where each law does or does not apply (5)(11)(15)(18)(19)(28). However, formulations of quantum mechanics that avoid this problem exist (11)(41). One such formulation that is particularly interesting for computation is GRW quantum mechanics. GRW is a collapse view with a stochastic dynamics (33). Its dynamics act as a source of noise that is inherent to physical objects. Though this source of noise is likely insignificant enough to allow large quantum computers (1), we will see that an investigation of computation in GRW still reveals lessons for the Aitken/Barrett thesis, quantum computing, and physical computation more broadly.

Geroch's conceptually rigorous treatment of quantum computing in *Perspectives in Computation* will act as a foil for the treatment of computation in GRW. Here Geroch states standard assumptions in treatments of quantum computing clearly and explicitly. But we will see that the careful treatments both of Geroch and of Aitken and Barrett could use additional clarification. Applying the Aitken/Barrett thesis to GRW helps illustrate where Geroch has made important insights, such as in his discussion of the plausibility constraints necessary for the problems computable by quantum computers to be the same as the problems computable by classical computers, and where his treatment needs additional conceptual grounding in light of the Aitken/Barrett thesis. Specifically, a fully conceptually rigorous treatment of quantum computing should be done relative to a specific formulation of quantum mechanics. Additionally, we will see some considerations that arise when applying the Aitken/Barrett thesis to plausible physical theories. One of these is that the particular properties of a physical theory should be taken into account in order to build a model of computation that takes advantage of them. In the context of GRW, we will see that a Turing machine model – though a

[1] See (77) for the original presentation of the von Neumann-Dirac formulation and (50)(51)(53)(70) for examples of treatments of quantum computing that rely on this view.

perfectly adequate computational model in many physical theories – is a particularly poorly suited model of computation for GRW.

This will lead to an extension of the Aitken/Barrett thesis: questions of physical computability must be answered relative to a specific physical theory and a computational model described relative to this theory, and this computational model should leverage the properties of the theory while being limited to what could potentially be epistemically useful for finite observers such as ourselves — we will discuss epistemically useful physical computation later, in the aptly named "On Epistemically Useful Physical Computation" chapter. While Aitken and Barrett do not discuss how to select computational models when applying their thesis to particular physical theories, the criteria used to select computational models are crucial for answering questions about computational power, such as whether a particular version of the physical Church-Turing thesis holds. Thus, as we will briefly discuss in Section 3.5, the extended Aitken/Barrett thesis has consequences for our understanding of physical Church-Turing theses and how notions of physical computational power can be evaluated (e.g. the discussions in (55)(56)(66)).

3.2 The Aitken/Barrett Thesis

In their "On the Physical Possibility of Ordinal Computation", Aitken and Barrett argue that giving a notion of physical computability requires specifying a physical theory and describing a computational model relative to this theory. For instance, they show that what is physically computable depends on the structure of spacetime by considering what Turing machine models constructed in various spacetimes can compute. Where traditional Turing machines "provid[e] natural computational models for ω-recursive functions," an α-Turing machine provides a computational model for α-recursive functions, where α is an arbitrary ordinal (p. 1 of (4), see also (3)). They show that a spacetime constructed from $\mathbb{R}$ only

allows α-Turing machines for countable α, but a spacetime constructed from the hyperreal numbers $*\mathbb{R}$ allows ω_1-Turing machines, where ω_1 is the first uncountable ordinal. Thus, more is physically computable by a Turing machine constructed in a hyperreal spacetime than in a real spacetime, and they conclude that more than what the Church-Turing thesis considers computable may be physically computable, depending on what the correct physical theory is. Consequently, the topological properties of a spacetime can put upper limits on what can be computed by a Turing machine in that spacetime.

This methodological approach is broader than just considerations of the spacetime structure in which a computer is built. Our notion of physical computing power must be defined relative to the physical theory we select and the computational model we construct in this theory. As our computational models must be selected relative to the physical theory, the various properties of the theory limit what could be made computable according to a model of computation in the theory.

This does not tell us, however, what considerations should be taken into account when selecting a model of computation relative to a physical theory. One approach would be constructing canonical computational models, like Turing machines, relative to a physical theory. But these may not necessarily be the best candidates for models of computation in that theory and may unnecessarily restrict our understanding of the capabilities of physical computation – had we remained blindly devoted to classical computational operations and bits, we would have been unable to realize the ways that the particular properties of quantum mechanical states and dynamics could be utilized to create algorithms (like Shor's algorithm) which are faster than any known classical algorithm. GRW vividly illustrates the importance of attending to particular properties of a physical theory when constructing models of physical computation. In this paper, we will look at a canonical quantum Turing machine model of computation and investigate what one finds when one tries to construct it according to the dynamical laws of GRW. This investigation will demonstrate the role

of selecting a precise quantum theory to underpin models of quantum computing and what considerations arise when applying the Aitken/Barrett thesis to plausible physical theories.

3.3 Geroch's Quantum-Assisted Computing Model

Before we turn to GRW quantum mechanics, let us first look at Geroch's register machine model of a quantum computer. This "quantum-assisted computer" provides an example of a standard model of quantum computing for Geroch to ground his discussion some conceptual issues in the foundations of quantum computing. It extends a model of a classical computer with two classes of quantum operations. It consists of five classes of classical commands along with the pair of quantum commands. Where S is a string over some set character set $\mathcal{C}$, $L(S)$, the length of the string, is one plus the number of characters in S, and $C(S)$ is the string in the location labelled by S, the classes of classical commands are:

1 *INPUT TO* $C(S)$*: allows the user to enter any string, which is then placed in location* S $\{L(\textit{whatever string is entered})\}$

2 *OUTPUT FROM* $C(S)$*: allows the user to retrieve the string stored in location* S $\{L(C(S))\}$

3 *APPEND* x *TO* $C(C(S))$*: replace whatever string is stored in location* $C(S)$ *with that same string, but with character* x *appended to the right.* $\{L(C(S))\}$

4 *DELETE LAST OF* $C(C(S))$*: replace whatever string is stored in location* $C(S)$ *with the string that results from deleting its rightmost character (if any) if* $C(C(S)) = \emptyset$*, do nothing.* $\{L(C(S)\}$

5 *IF (LAST* $C(C(S)) == x$*) SKIP* n *LINES: if the last character (if any) of the string in location* $C(S)$ *is "x", then skip forward* n *program lines (if* n *is positive), backward*

$|n|$ lines (if negative). If $C(S) = \emptyset$, or if the last character of $C(S)$ is other than x, or if the line to be skipped to is an INPUT command, or if there are insufficient lines in the program to carry out the indicated skip, do nothing. $\{L(C(S))\}$ (p. 50 of (32))

These describe a classical computer. The classes of quantum commands are:

6 APPLY $C(S)$

which does nothing if $C(S)$ is not a unitary operator and applies the unitary operator to the state $|\Psi\rangle \in \mathcal{H}$ of a separate quantum system that exists in addition to our classical computer if $C(S)$ is a unitary operator[2] (p. 92 of (32)).

And finally, we have:

7 OBSERVE $C(S)$, APPEND result to $C(S')$.

Similarly to the APPLY command, if $C(S)$ is not a projection operator, nothing is done. But if $C(S)$ is a projection operator, then $\mathcal{H}$ is extended as necessary (as in an APPLY command) after which an observation is made using a self-adjoint projection operator. The result of the observation (0 or 1) is appended to the string at $C(S')$ (p. 93 of (32)). Thus, this model appears to assume some sort of collapse view.

Geroch notes some reasons to be cautious about claims that quantum-assisted computers can compute some problems more efficiently than classical computers.[3] But to make the claim

[2]The operator will apply to $H_{S_1} \otimes ... \otimes H_{S_k}$ for some k, but $|\Psi\rangle$ exists in $H_{S_1} \otimes ... \otimes H_{S_n}$ for some n that may not be equal to k. If $n > k$, then I is applied to the remaining factors. If $n < k$, then $\mathcal{H}$ is extended by taking its tensor product with copies of $|\psi_0\rangle$, the initial state of the quantum system, until $k = n$ and then applying the unitary operator to this state (32).

[3]One is that given any classical algorithm that computes a problem, we cannot eliminate the possibility that a more efficient classical algorithm for computing that problem exists. We have not found lower limits for the classical difficulty of computing problems (e.g. factoring numbers) for which we have quantum-assisted algorithms (algorithms such as Shor's algorithm which utilize quantum processes in addition to classical processes) that are more efficient than any known classical algorithm. Thus, we do not yet know with certainty that a quantum-assisted computer can be used to compute some problems more efficiently than any classical computer.

that the set of problems that quantum computers can solve is the same as the set of problems that classical computers can solve, we must specify how we are going to restrict the components of our computer so that our model of quantum computation represents computers that we could plausibly build; these restrictions do not arise from quantum mechanics itself but instead arise from extra-theoretical considerations on what kind of physical systems we could plausibly construct for use in a computer. First we must note that "we cannot look at the state $|\psi_0\rangle$ [the initial state], the unitary operators that appear on our list, and the projection operators that appear on our list in isolation"; the entire set must be considered, for reasons we will soon see (p. 98 of (32)). So we must put restrictions that apply to all of these components. Let us look at these restrictions and then consider what could happen without them and why they are realistic constraints.

For the selection of the quantum components (the initial state, set of unitary operators, and set of projection operators) Geroch proposes the following restriction. Given a Hilbert space H and a standard basis for this Hilbert space, we construct the standard basis for $H \otimes ... \otimes H$. For any unitary operator U in our set, we demand that there exists a classical program which computes a rational number within $\frac{1}{n}$ of a given matrix element of U, for any positive integer n, and we demand the existence of similar programs for the initial state and all the projection operators. The initial state and unitary and projection operators are computable in this sense. We also demand that the sets of unitary and projection operators be finite. These restrictions will ensure that what is quantum-assist computable is classically computable (32).

If we drop this restriction and allow any unitary operator to be used in a quantum-assisted computer, then we can select a single unitary operator that takes the input state to the correct output state in one step. We would even be able to compute classically noncomputable numbers, making our quantum-assisted computer able to compute more problems than a classical computer. One way to do this would be to construct a physical system to act as

a unitary operator that performs rotation through a noncomputable number (32). But this would require that we know the value of that number and that we could devise a method of reliably building a device that could act as such an operator.[4] What procedure could we use to complete this task? We could not use some classical algorithm to construct the system, as a classical computer cannot compute the operator — this is precisely the restriction we dropped —, and using a quantum procedure to construct the operator would require that we have already solved this problem (32).

An unrestricted quantum computer could also compute a (classically) noncomputable number by using more than one operator. For instance, we could use two unitary operators which perform 90° rotation but in planes which form an angle with is a (classically) noncomputable number c, as these operators together would allow one to use the quantum-assisted computer to compute c (p. 98 of (32)). We could perform a Monte-Carlo estimate of c by repeatedly applying these operators in succession then taking a measurement. But building multiple systems to act as these unitary operators to accomplish this is implausible. Again, we would have to know the noncomputable number and be able to reliably construct physical systems which could act as these operators. While we can construct individual operators which perform 90° rotations, we could not give a classical process which, given one of these operators, reliably places the second operator at a (classically) noncomputable angle relative to the first in the Hilbert space. Similarly, the initial state and projection operators must also be selected to avoid these issues.

Geroch's restrictions thus attempt to limit our discussion of quantum computers to those we could plausibly build. Assuming these restrictions, we find that the maximum speedup of a quantum computer relative to a classical computer is exponential; without these restric-

[4]The requirement that we are able to "reliably" construct these operators is crucial. Because classically noncomputable numbers are measure one among the complex numbers, any physical system that we construct according to the above process to act as a unitary operator is going to act on qubits as some noncomputable unitary operator U', though one whose matrix elements are within $\frac{1}{n}$ (for any n) of some (classically) computable U. We could not, however, reliably construct physical systems which act as unitary operators whose elements are within $\frac{1}{n}$ (for any n) of the elements of U'.

tions, a quantum-assist computer could compute any classical computation in a single step, simply by applying whatever unitary operator takes the initial state to the final state of the computation (32). But this ability would require us to build a particular unitary operator to transform a specific initial state to a specific final state, which we must know in order to design the operator, and to do this, we would need to have access to a unitary operator for any initial input, whereas Geroch restricts us to finite sets of operators.

Geroch has clarified extra-theoretical constraints on the components allowed in a reasonable model of quantum computation that are implicit in most discussions of quantum computing. But to which precise physical theory do they apply? His account of measurement suggests a collapse view: When an observation of A is made on a system in state $|\psi\rangle$, "*the 'result' of the observation is the projection of* $|\psi\rangle$ *into the corresponding eigenspace and the probability of that result is the squared-norm of that projection* [emphasis in original]" (p. 72 of (32)). A straightforward reading of the OBSERVE class of commands suggests some sort of collapse theory. But different quantum theories account for measurement in different ways, and some quantum theories do not allow the class of OBSERVE commands as described in his quantum-assisted computing model. For instance, where OBSERVE operations (rather than APPLY operations) would act is unclear in Everettian and Bohmian theories, which are governed entirely by pure wave mechanics and hence do not include wavefunction collapses.

Thus, to fulfill the Aitken/Barrett thesis, Geroch would need to specify the quantum theory (or set of theories) to which these restrictions and hence the quantum-assisted model apply and how the quantum-assisted model arises from this theory (or set of theories). Instead, he gives a model of physical computation and leaves open how it arises from a specific quantum theory. But different theories of quantum mechanics may lead to different models of computation and even different notions of computational power. GRW is a particularly instructive example in that, among the theories that support OBSERVE operations, it arguably does so in the clearest way. However, we will see that constructing a model of computation in

GRW introduces unique concerns arising from GRW's idiosyncratic properties.

3.4 GRW Computing

3.4.1 Machine Models & GRW Idiosyncrasies

Applying the Aitken/Barrett thesis to GRW illustrates the importance of model-selection relative to a particular physical theory for both understanding how one should select computational models relative to a theory and the dependence of models of quantum computing on particular formulations of quantum mechanics.. In particular, we will compare traditional quantum and GRW Turing machines to see how they differ and why specific theories – and in particular, quantum theories – like GRW may have properties which uniquely influence what models of computation are well-suited for them and which are not.

For our purposes, a traditional quantum Turing machine (QTM) will never collapse, and instead operate solely on the unitary dynamics. A QTM M_Q over alphabet A consists of a processor, a bilateral infinite tape, and a head that reads and writes symbols on this tape (53). It is defined as a sixtuple $(Q, A, \delta, q_0, q_a, q_r)$, where $q_0, q_a, q_r \in Q$ are the initial, accepting, and rejecting states, Q is the set of internal states, and δ is the transition function. Additionally, T is the tape configuration, ξ is the cell at which the head is placed, and $T(\xi)$ is the symbol at cell ξ (53). Where $\tilde{\mathbb{C}}$ is the set of computable complex numbers such that for all $c \in \tilde{\mathbb{C}}$, $|c|^2 \leq 1$, the transition function takes the form

$$\delta : Q \times A \times Q \times A \times \{-1, 1\} \to \tilde{\mathbb{C}}. \tag{3.1}$$

The set $\{-1, 1\}$ denotes whether the machine head moves one cell to the left (for -1) or one

cell to the right (for 1). Where $q \in Q$ and $a \in A$,

$$\sum_{(q_i,a_j,d)\in(Q,A,\{-1,1\})} |\delta(q,a,q_i,a_j,d)|^2 = 1. \tag{3.2}$$

This transition function carries out the work of unitary operators on the state. The state of the machine is given by the vector

$$|\psi(t)\rangle = |x; n_0, n_1, ...; m\rangle, \tag{3.3}$$

where the processor is composed of the n_i two-state observables, the memory is composed of an infinite sequence of m_i and the head position is $x \in \mathbb{Z}$ (47). A universal QTM can compute the same problems that a (reversible) universal classical TM can compute and can also execute algorithms than no classical TM can (22).

Similarly, a GRW Turing machine (GRWTM) M_{GRW} over alphabet A again consists of a processor, a bilateral infinite tape, and a head that reads and writes symbols on this tape. However, it can be defined as a octuple $(Q, A, \delta, q_0, q_a, q_r, N, t)$, where Q, A, δ, and the three q_x are defined as before, and N is a constant defined as the number of particles in the machine's head and t is a constant defined as the amount of time taken by each step of the Turing machine. From the Poisson distribution (**??**), we know that with probability $e^{-\frac{Nt}{\tau}}$ there is no collapse during a given step, and transition function $\delta : Q \times A \times Q \times A \times \{-\infty, \infty\} \to \mathbb{C}$ is

$$\delta(q, a, q', a', d \in \{-1, 1\}) \approx c \in \tilde{\mathbb{C}} \tag{3.4}$$

where similarly to the QTM case

$$\sum_{(q_i,a_j,d)\in(Q,A,\{-1,1\})} |\delta(q,a,q_i,a_j,d)|^2 \approx 1. \tag{3.5}$$

The transition function is only approximately c and the sum is slightly less than one due to the "tails" of the GRW wavefunction, which are the states where $d \notin \{-1, 1\}$. But with probability

$$1 - e^{-\frac{Nt}{\tau}}, \tag{3.6}$$

the machine head collapses in position. The position is selected according to the Born rule, though the wavefunction collapses into a narrow Gaussian and not an eigenstate. So if the machine head is at a definite position at step i, but

$$|\delta(q_i, a_i, q_{i+1}, a_{i+1}, -1)|^2, |\delta(q_i, a_i, q'_{i+1}, a'_{i+1}, 1)|^2 > 0, \tag{3.7}$$

at step $i + 1$ with probability $1 - e^{-\frac{Nt}{\tau}}$ the machine behaves *as though*

$$|\delta(q_i, a_i, q_{i+1}, a_{i+1}, -1)|^2 \approx 1 \text{ with probability } |\delta(q_i, a_i, q_{i+1}, a_{i+1}, -1)|^2 \tag{3.8}$$

or

$$|\delta(q_i, a_i, q_{i+1}, a_{i+1}, 1)|^2 \approx 1 \text{ with probability } |\delta(q_i, a_i, q_{i+1}, a_{i+1}, 1)|^2, \tag{3.9}$$

again with the caveat that the collapse is to a narrow Gaussian and thus these values will be slightly less than one. Similarly, if the machine head is not at a (approximately) definite cell position at step i, a collapse may give it a definite position before step $i + 1$ even if without collapse the transition function would not give it a definite position by step $i + 1$.

For the rest of the paper, I will assume that the standard basis observable (the basis for the values of symbols $T(\xi_i)$) for the GRWTM is neither position nor is ever entangled with particle positions besides those of the machine head. This ensures that spontaneous position collapses only act on cell symbols $T(\xi_i)$ when the symbols are entangled with the machine

head's position and thus reduces the occasions where spontaneous collapses would be problematic for the computation, as collapses which occur when a cell value is in a superposition of standard basis values but the machine head is not in a superposition of position are extremely unlikely to ruin the computation.[5]

Algorithms which never rely on superpositions of head position maximize the reliability of a GRWTM. However, a narrow Gaussian wavefunction always has nonzero "tails" which are outside the width of a given tape cell, so there is a nonzero probability that the machine head collapses into the wrong tape position. So, for instance, whereas a QTM may be in the state

$$\sum_{i=1}^{n} \alpha_i |q_i, T_i, \xi_i\rangle \tag{3.10}$$

at some point while executing an algorithm, then a GRWTM will be in a state

$$\sum_{i=1}^{n} (\alpha_i - \epsilon)|q_i, T_i, \xi_i\rangle + n\epsilon |q_i, T_i, x\rangle \tag{3.11}$$

at the same point of its execution of that algorithm, where $|q_i, T_i, x\rangle$ is the machine's state if the head collapses into a Gaussian tail, so x is a position outside of all cells ξ_i. Note that $\|\epsilon\|^2$ is extremely small, but if $\xi_i \neq \xi_j$ for any $i, j \in \{1, ..., n\}$, *any* collapse to a (near) eigenstate of position will put the GRWTM in some state other than (3.11) at that point of the algorithm and will likely ruin the computation. Since the probability of collapse at any step, (3.6), is nonzero for any $t, N > 0$, we know that the GRWTM cannot compute any non-identity problem with probability one.

To see this, consider any problem for which the input differs from the output. So there is some (finite) non-empty set of cells $\{\xi_1, \xi_2, ..., \xi_n\}$ such that, where $T_0(\xi_i)$ is the initial symbol at ξ_i

[5]Collapses still be able to ruin a computation, as they could still occur into the wavefunction tails. Tails collapses may result in a loss of a component from the computer, potentially ruining a computation.

and $T_f(\xi_i)$ is the final symbol at ξ_i, $T_0(\xi_i) \neq T_f(\xi_i)$. Thus, at some point in the computation the machine head must act on a particular cell ξ_j and thus must be located at it at some point in the computation. But the machine head cannot be guaranteed to ever occupy a particular position, as the head's wavefunction could collapse into its nonzero tails or a component of any of the intended superpositions of position that arise during the execution of the algorithm.[6] Thus, the GRWTM head may not reach a cell whose symbol must be changed in a computation and thus cannot be guaranteed to complete any computation. Even if the GRWTM runs a classical algorithm which does not put the machine head in superpositions of position, the nature of GRW tails prevents the machine head from occupying any position with probability one.[7]

Thus, the GRWTM model differs from the QTM model in that it is not quite Turing equivalent, in the sense of being able to compute with probability one a problem a classical TM can compute with probability one, but rather may only be Turing equivalent in the sense that, with probability $1 - \epsilon$, for some $\epsilon > 0$, it can compute a problem that a classical TM can compute. Even for an implementation of a GRWTM without any error in its construction and without any environmental decoherence, the reliability of a GRWTM decreases as more steps are required for a computation, as more steps involve superpositions of position, and as the physical parameters of the machine change (i.e. as N and t grow and the cell width shrinks). However, whether GRW's dynamics would allow one to reduce the error in the GRWTM such that its probability of success could be within *any* ϵ of one remains uncertain.

[6]To illustrate this for a quantum algorithm that utilizes superpositions, have a program which puts the head in a superposition of visiting ξ_i and changing its symbol before visiting ξ_k then going to ξ_j and of visiting ξ_k before going to ξ_i and changing its symbol before going to ξ_j, where $i \neq j \neq k$. If the machine head collapses while the machine is in a superposition of the two paths, the symbol at ξ_i may not be changed (depending on the path into which the machine collapses and when it collapses) and the machine may not reach ξ_j.

[7]The situation is worse than even this description suggests, as the other components of the machine are subject to the same dynamics and can collapse into Gaussian tails. Hence, the GRWTM cannot even be guaranteed to compute identity.

But even if a GRW's dynamics do not allow a Turing machine to be brought within some ϵ of being Turing-equivalent, we could not at this point conclude from the GRWTM example that GRW does not permit computers that are as powerful as a QTM. In fact, we would be able to construct a GRW computer arbitrarily close in power to a QTM if we could completely avoid position superpositions and use physical systems as unitary operators that minimize the risk of spontaneous collapses putting the qubits outside the region where our unitary operator systems would act on them.

The first step to improving on our GRWTM would be eliminating superpositions of position as much as possible when using quantum algorithms. Suppose we want to build a GRW computer that avoids the GRWTM's problems with spontaneous position collapses. Ideally, we would want any superposition states to be unentangled with position, so that with probability $1 - \|\epsilon\|^2$ (where ϵ is the weight of the tails portion of the state, as in (3.11)) a spontaneous collapse in our computer does not change its state in a way that would affect the computation. That is, only collapses into Gaussian tails would ruin the computation because (unlike the GRWTM) the computer does not rely on superpositions of position in its computation.

We could get such a computer by adapting Geroch's quantum-assisted computing model to GRW. The classical operations would remain mostly unchanged, with the caveat that there is a very small probability that the components would collapse to the wrong position (akin to the what we saw in the GRWTM). We select a standard basis that is neither position nor is (typically) entangled with position, so the APPLY command behaves mostly the same as in the quantum-assisted model, again with the possibility of collapses into tails. The OBSERVE command acts by entangling the measurement observable with the position of some auxiliary system. The larger this auxiliary system is (in terms of number of particles), the more likely the joint system is to collapse in a given period of time, though this probability is never one for a finite time period.

Both a GRWTM and a GRW-assisted computer experience noise that is intrinsic to GRW's dynamics and does not arise from any imperfection in the construction of a particular computer, but a GRW-assisted computer can fare better than a GRWTM given similar values for N and t, if it is constructed to be more resilient to spontaneous collapses in position. This difference is due to the unique way that GRW treats position, and would not appear in many other theories.[8] Consider when the quantum system in the GRW-assisted computer is put in a superposition state of an observable that is not entangled with position. So the system's state is

$$|\Psi\rangle = \sum_{i=1}^{n}(\alpha_i - \epsilon)|\psi_i\rangle|X\rangle + n\epsilon|\psi_{tails}\rangle|\overline{X}\rangle, \tag{3.12}$$

where X is the region the quantum system is supposed to inhabit, and $|\psi_{tails}\rangle|\overline{X}\rangle$ represents the wavefunction tails. Compare a spontaneous collapse in position of this state with the superposition state of the GRWTM given by (3.11). Whereas any collapse in position could ruin the computation of a GRWTM if $\xi_i \neq \xi_j$ for any $i, j \in \{1, ..., n\}$, only a collapse into the tails would ruin the computation of the GRW-assisted computer. So a GRW-assisted computer could be an improvement on the GRWTM.

A GRW-assisted computer can be made significantly safer from problems associated with spontaneous position collapses than the GRWTM by constructing it so that the vast majority of spontaneous position collapses it undergoes do not affect its computation. Such a distinction would not arise in Everettian theories, which do not feature spontaneous collapses into a preferred observable. So GRW promotes additional prescriptions on how to build more powerful physical computers that other quantum theories may not (and which

[8]Error correction would be able to help in either case, though a GRW-assisted computer could be constructed to have a smaller underlying error rate than a GRWTM, given the same number of bits, particles per bit system, and time per computational step. Thus, GRW's dynamics alone can make an register machine model, like the GRW-assisted computer, preferable to a Turing machine model, because such a model could be made to be less reliant on position. The possible difference between Turing and register machines would not necessarily be the case in other (quantum) theories.

are certainly not seen in the QTM model), and we can take these into account to build a GRW computer that is more reliable than a GRWTM by limiting its reliance on position.[9] For instance, a GRW-assisted computer with a standard basis that is not position will have its computations ruined by a smaller portion of its spontaneous collapses than a GRWTM. In fact, a Turing Machine model is a particularly bad model of computation in GRW, as it relies on precisely the feature of GRW which could be the most damaging to a model of computation in the theory, the position observable in which spontaneous collapses occur.

The nature of GRW's collapses and wavefunction tails implies that, in a nonzero time interval t and given a finite region R of configuration space, there is always a nonzero probability that a system will collapse outside of R. But a finite-energy system that acts as a unitary operator can only apply its intended operation to bit systems within some finite region of configuration space; for a system to act uniformly on systems across all of configuration space, it would need to have infinite energy. Restricted to finite energy systems, GRW's collapse dynamics does not in principle allow computers that can apply a unitary operator with probability one.[10] That is, even if the components of a computer were constructed perfectly, GRW's dynamics themselves produce noise in a way that the dynamics of other theories – even other quantum theories – do not. Even error correction and fault tolerance procedures have an inherent and in principle unavoidable nonzero chance of failure arising from GRW's dynamics, as the probability of tails collapses for error correcting systems must also be nonzero.

A good model of computation in GRW minimizes these effects of spontaneous collapses, whereas a good model of computation in another quantum theory – or any other physical theory – takes into account the particular features of that theory. Thus we see the value of

[9] Similarly, the reliability of GRWTMs depends on machine head size, processing speed, and cell size.

[10] Additionally, one can easily see that GRW allows the construction of computers that execute tasks that Turing equivalent computers cannot, for instance, by allowing computers that can, with positive probability, generate random (not merely pseudorandom) strings by executing repeated measurements of superposition states.

Aitken and Barrett's prescription to select a precise physical theory from which to construct a model of physical computing, as the particular properties of a theory can have interesting effects on the computational power of models in the theory. But we also must carefully choose which model of computation in a physical theory we use in order to take advantage of the properties of that theory; we cannot assume that even a canonical model of computation (e.g. a Turing machine model) in one physical theory can be transferred into some other theory to provide a good (or even plausible) model of computation in that theory.

3.5 Remarks

One cannot say what is physically computable without stating how a computer is to be constructed within a precise physical theory. Geroch rightly notes that considerations about what quantum computers could be constructed determine what notion of quantum computing power we get. An application of the Aitken/Barrett thesis to a plausible physical theory should take similar considerations into account. But as Aitken and Barrett argue and GRW illustrates, our physical theory determines what models of computation we can construct, and Geroch has not yet specified what formulation of quantum mechanics underlies his model of quantum computation. Additionally, to get a *good* model of computation in a theory, we must leverage the particular properties of that theory. This may mean that canonical models of computation that may be perfectly adequate in some physical theories may be ill-suited for others. Leveraging GRW's properties means minimizing the role of position and especially superpositions of positions in the computational model, meaning Turing machines (both classical and quantum) are particularly ill-suited for GRW, though that may not be the case in other formulations of quantum mechanics.

In general, when fulfilling the demands of the Aitken/Barrett thesis, we must attend to the properties of a physical theory and take advantage of them to construct a good model

of computation in that theory, while also attending to plausibility concerns, such as those Geroch explicitly notes. This leads to an extension of the Aitken/Barrett thesis: questions of physical computability must be answered relative to a specific physical theory and a computational model described relative to this theory, and this computational model should leverage the particular properties of the theory while following plausibility constraints on what models of physical computation to consider. The latter component of this thesis alludes to the freedom one has when constructing a computational model relative to a physical theory. Without plausibility constraints, one can run into issues such as unlimited pancomputationalism, in which any sufficiently complex physical system is considered to compute many different problems (57)(60)(69). But in cases where plausibility constraints have been given, similarly to what we saw in the case of Geroch, the selection of a precise physical theory to ground the discussion is bypassed. Without such a selection, a treatment of physical computation amounts to speculation on what will likely be possible in our physical theories, given some generalized plausibility constraints.[11] But if we wish to have a more concretely grounded conversation, we ought to follow the Aitken/Barrett thesis, and if we follow the Aitken/Barrett thesis, we should be mindful both of the useful tools that our physical theories can give us and the plausible limits on the ways in which we can utilize these tools.

[11] Piccinini does this explicitly in (55).

Chapter 4

A GRW Register Machine Error Model

4.1 Introduction

Presentations of quantum computing typically appeal – often implicitly – to some version of the von Neumann-Dirac collapse formulation of quantum mechanics, in which the deterministic unitary dynamics and the probabilistic collapse dynamics can be cleanly parsed into different domains of governance (e.g. (50)(51)(70)). Such a quantum theory is incomplete at best. It succumbs to the measurement problem, in which each of the theory's two different dynamics describe behavior that contradicts the behavior described by the other dynamics, and the theory does not provide a way to determine precisely when each dynamics will apply. (5)(11)(15)(18)(19)(28) However, there are other formulations of quantum mechanics that do not encounter the measurement problem. (41) Thus, since the von Neumann-Dirac formulation is at best incomplete, examining models of quantum computing in more robust formulations of quantum mechanics is valuable. GRW is one such theory, with the feature

that its dynamical rules alone act as a source of computational error, which is not the case in other formulations of quantum mechanics.

Looking at computing in GRW can give us insights into quantum error – both its possible sources and new ways that it could be modeled, since computers in GRW experience error that arises from the GRW's dynamics alone, rather than from environmental decoherence or noisy gates. GRW's dynamics include spontaneous collapses to approximate eigenstates of position. These act as a source of noise that is unavoidable in principle for computers constructed in a world governed by GRW. Thus, any model of computation in GRW must intrinsically include error, and one can empirically test GRW by attempting to make quantum computers which are less error-prone than GRW would suggest.

To begin the project of developing a concrete understanding of this particular source of error, we will look at a simple register machine model in GRW and model the error it would experience due exclusively to GRW's stochastic dynamics. Because the frequency of spontaneous collapses in GRW increases with the number of particles entangled in position, increasing the size or number of bit systems can increase the risk of error for a computer governed by GRW's dynamics. Nonetheless, GRW likely allows large quantum computers (1). But how large and how reliable computers in GRW can be is determined by the error inherently caused by GRW's stochastic dynamics. To investigate this error, this paper presents a simple idealized register machine model in GRW and examines the computational error that arises from GRW's dynamics alone. This model illustrates the unique noise a quantum computer in a GRW-governed universe would experience and how we ought to think about this noise. Section 4.5 concludes with what this error model means for our understandings of GRW and quantum computing more generally.

4.2 The Register Machine

The error experienced by a computer governed by GRW is inherently tied to the precise details of its physical characteristics. Thus, to get a handle on error in GRW, we will need to analyze a concrete model of computation in GRW. Consider a register machine consisting of a finite number n of one particle bit systems. This will have two computationally relevant bit states, $|0\rangle$ and $|1\rangle$; suppose the basis for these computational states is position. The computationally relevant states correspond to finite spatial regions, which, for simplicity, will be two dimensional squares of side length $2r$. These squares lie in a grid, so that there is no spatial gap between adjacent squares. This ensures that computational operations do not act as an additional source of error to the error which arises from GRW's stochastic dynamics. A bit b_i is in computational state $|x\rangle$ insofar as it is in state $|R_{xi}\rangle$, where $|R_{xi}\rangle$ corresponds to the physical system representing b_i being in the square region R_{xi} (see Figure 4.1).

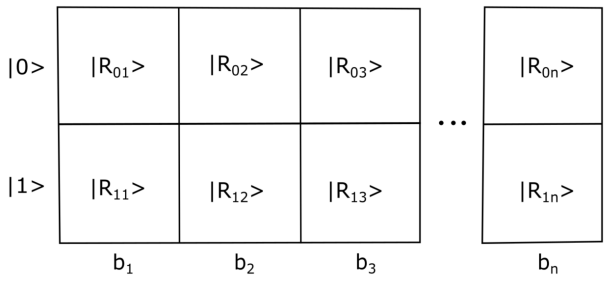

Figure 4.1

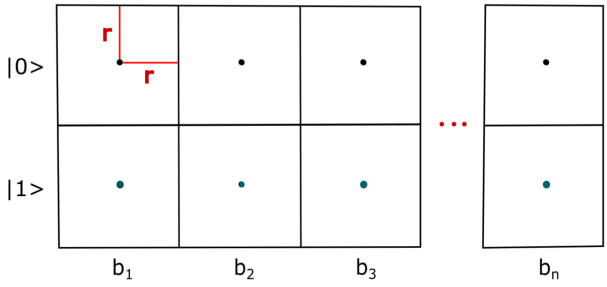

Figure 4.2

On the GRW dynamics, if a system experiences a collapse, it will be multiplied by a Gaussian wavefunction with nonzero "tails" such that, no matter how far x_0 is from the peak of a collapsed wavefunction, $|\psi(x_0, t)|^2 > 0$ for all t. Under plausible physical conditions, the wavefunction will not be fully contained in one of the boxes.[1] So GRW results in the wavefunction corresponding to a par-

[1] An infinite potential well could be used to compress the wavefunction so that it it fully contained in a finite spatial region. But without such a potential well, there will always be wavefunction tails outside any finite spatial region and thus a positive probability for a collapse outside that finite spatial region.

ticle not being wholly contained to a finite spatial region. Thus a bit system in our model will never be in an eigenstate of a computational state. Rather, the bit system will always be in some state of the form $\alpha|R_{0i}\rangle + \beta|R_{1i}\rangle + \gamma|tails_i\rangle$. A bit will be considered to be in a computational state insofar as the center of collapse is in the corresponding spatial region.[2]

As an idealizing assumption, suppose that the bits are initialized so that they are in the precise centers of the R_{0i} regions (See Figure 4.2), and if we take the center of R_{0i} to be the origin, the bit system b_i begins in the state

$$|\psi(x,y)|^2 = \frac{1}{\pi\alpha^2} e^{-\frac{x^2+y^2}{\alpha^2}}. \tag{4.1}$$

This minimizes the amount of wavefunction that lies outside of the intended state and hence the probability that a spontaneous collapse will change the computational state. Similarly, in the initial presentation of this model, wavefunction dispersion will be ignored, again minimizing the risk of error posed by collapses.

Suppose each computational step takes time interval t_s. A computational operation either leaves a bit b_i in $|R_{xi}\rangle$ or changes its state to $|R_{(1-x)i}\rangle$ by displacing it vertically by a distance of $2r$. We will assume that these operators are implemented perfectly and thus do not contribute to the risk of error. Immediately after an operation is performed, the operator and bit systems are unentangled.

4.3 Classical Computing Error Rate

The model described in Section 4.2 for describing bits is explicitly quantum mechanical, but let us first look at how it performs given only classical computational steps – that is, we

[2]What this means will vary among different versions ("primitive ontologies") of GRW, though these differences will not be relevant to this treatment of GRW Register Machine models.

will restrict ourselves to operations corresponding to classical computational operations. As will become apparent in the following section, a classical computational model gives a much simpler starting point than a quantum computational model for describing the bit error that a GRW Register Machine experiences from spontaneous collapses. In this simple model, the only source of error will be tails collapses. Thus, the probability of error will be given by

$$P(error) = P(> 0\, tails\, collapses) = 1 - P(0\, tails\, collapses) \tag{4.2}$$

The probability distribution of collapses in time are given by (**??**), so the probability of at least one collapse is given by

$$P(m > 0\, collapses) = 1 - P(0\, collapses) = 1 - e^{\frac{Nt}{\tau}}. \tag{4.3}$$

On this model, errors are caused by collapses that move a bit system outside its intended R_{xi} region. So (4.3) is not the collapse rate for the tails collapses outside the region R_{xi} that would cause error. To find the tails collapse rate, we should integrate the wavefunction magnitude over the region outside of the intended $2r \times 2r$ region in which a bit system is supposed to reside. This gives us the probability that a given collapse puts the bit system outside of its intended region, either by placing it in the region corresponding to the same bit being in the other computational state (bit flip), by placing it in a region reserved for some other bit system, or by placing it outside of the computer altogether (bit loss). Any of these tails collapses will be considered to result in an error, so we will not distinguish among them when calculating the error rate.[3] Thus, if we set the origin to the center of the intended $2r \times 2r$ square, the probability that any given collapse will be into a

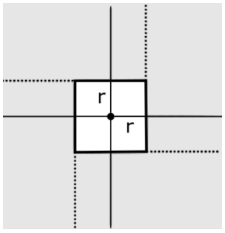

Figure 4.3

[3]However, for the purposes of error correction, which kind of error occurs would be important for determining what process should be used to correct for the error.

wavefunction tail and thus cause an error is

$$\begin{aligned} P(tails\,collapse|collapse) &= \int_r^\infty dx \int_{-r}^\infty dy|\psi(x,y)|^2 \\ &+ \int_{-r}^\infty dx \int_{-\infty}^{-r} dy|\psi(x,y)|^2 \\ &+ \int_{-\infty}^{-r} dx \int_{-\infty}^{r} dy|\psi(x,y)|^2 \\ &+ \int_{-\infty}^{r} dx \int_{r}^{\infty} dy|\psi(x,y)|^2 \\ &= 4\int_r^\infty dx \int_{-r}^\infty dy|\psi(x,y)|^2 \end{aligned} \tag{4.4}$$

by the symmetry of the Gaussian wavefunction (see figure 4.3). Any non-tails collapse will still move the center of the Gaussian away from the center of its intended region and thus increase the chance that a subsequent collapse will be a tails collapse out of its intended region. So

$$\begin{aligned} P(tails\,collapse|collapse) &\geq 4\int_r^\infty dx \int_{-r}^\infty dy \tfrac{1}{\pi\alpha^2} e^{-\frac{x^2+y^2}{\alpha^2}} \\ &= \tfrac{2}{\alpha\sqrt{\pi}}(1-erf(\tfrac{r}{\alpha}))\int_{-r}^\infty e^{-(\frac{y}{\alpha})^2}dy \\ &= 1-[erf(\tfrac{r}{\alpha})]^2, \end{aligned} \tag{4.5}$$

where erf is the error function. The rate of tails collapses is $f_{col}P(tails\,collapse|collapse)$, where $f_{col} = \frac{Nt}{\tau}$ is the collapse rate. This gives an error rate of $\frac{N}{\tau}(1-[erf(\frac{r}{\alpha})]^2)$. Thus, the Poisson distribution governing errors caused by tails collapses is given by

$$P(m\,tails\,collapses) \geq e^{-\frac{Nt}{\tau}(1-[erf(\frac{r}{\alpha})]^2)}\frac{(-\frac{Nt}{\tau}(1-[erf(\frac{r}{\alpha})]^2))^m}{m!} \tag{4.6}$$

and thus

$$P(Error) = P(m>0\,tails\,collapses) \geq 1-e^{\frac{Nt}{\tau}([erf(\frac{r}{\alpha})]^2-1)}. \tag{4.7}$$

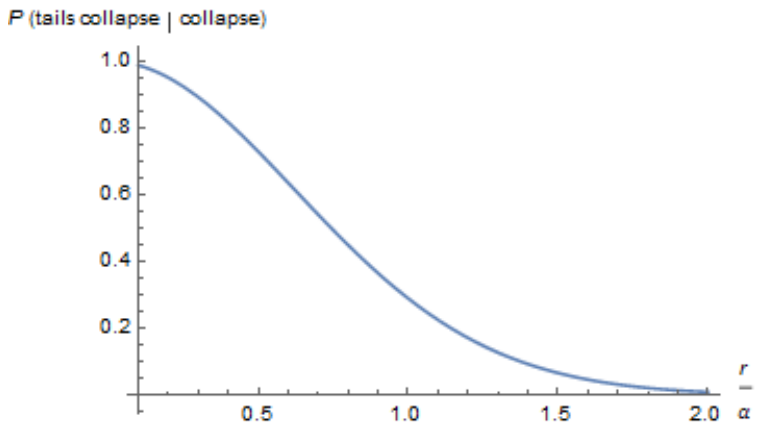

Figure 4.4

This is the probability of at least one bit error from a tails collapse over the course of a computation that takes place over time interval t. The behaviors of the functions for the error rate and the probability of error are intuitive. When the sole source of error comes from spontaneous tails collapses, increasing the number of particles (each which represents a qubit) or the time taken by the computation increases the risk of error whereas increasing the size of the boxes reduces the risk of error. The error rate cannot exceed the collapse rate $\frac{N}{\tau}$ and, since $erf(\frac{r}{\alpha})$ goes to one as r goes to infinity, the error rate and error probabilities go to zero as r goes to infinity.

Our model has made several major idealizations which should reduce the risk of error: bits are represented by single particle systems, these systems experience no dispersion, we have an idealized initial state in which bit systems are perfectly centered in their corresponding regions, and the entire register machine is perfectly shielded from environmental decoherence. But even with these significant idealizations, there is an inherent risk of error that results from the GRW dynamics. This noise depends on the formulation of quantum mechanics governing the model of computation as well as the details of the model of computation itself.

GRW's dynamics themselves inherently generate error for our register machine, as they are the only source of error in the model we have given. Note that our analysis of error depends on the minute details of GRW's dynamics and the computational model. The nature of GRW means that the noise experienced by a computer is highly contingent on the GRW dynamics, the values of the GRW constants, and the physical make-up of the computer.

4.4 Quantum Computing – A Dilemma

So far we have only discussed classical computation in the context of our GRW Register Machine. Quantum computing introduces two aspects not seen in our classical computing model — entanglement and operators that are meant to put bit systems in superpositions of register states. In our classical computational model, each bit b_i is always *supposed* to be in either state $|R_{0i}\rangle$ or state $|R_{1i}\rangle$, but in a quantum computation, the intended state for b_i could also be a superposition state $\alpha_s|R_{0i}\rangle + \beta_s|R_{1i}\rangle$. Additionally, bit systems could be intended to be in an entangled state such as $\alpha|R_{0i}\rangle|R_{0j}\rangle + \beta|R_{1i}\rangle|R_{1j}\rangle$. This adds potential for a new type of error due to spontaneous collapse that was not present in the purely classical model. Any collapse – not just a tails collapse – that occurs when a bit system is supposed to be in a superposition can lead to a computational error, as any collapse will change a bit's state as

$$\alpha_s|R_{0i}\rangle + \beta_s|R_{1i}\rangle + \gamma_s|tails_i\rangle \rightarrow \alpha_c|R_{0i}\rangle + \beta_c|R_{1i}\rangle + \gamma_c|tails_i\rangle. \tag{4.8}$$

A collapse may result in a state that is very close to a state described by $\alpha_s|R_{0i}\rangle + \beta_s|R_{1i}\rangle + \gamma_s|tails_i\rangle$, and thus may not cause an error. But if the collapsed state $\alpha_c|R_{0i}\rangle + \beta_c|R_{1i}\rangle + \gamma_c|tails_i\rangle$ differs too much from the superposition state, then we can say an error occurs. Thus, we must set some threshold past which we consider the computation to have erred, but there are different ways in which we can define this error threshold.

One approach would be to say that an error occurs if $|\alpha_s|^2$ and $|\alpha_c|^2$ or $|\beta_s|^2$ and $|\beta_c|^2$ differ by too much. We can measure the "distance" between the intended superposition state and a post-collapse state by

$$D = \sqrt{(|\alpha_s|^2 - |\alpha_c|^2)^2 + (|\beta_s|^2 - |\beta_c|^2)^2} \tag{4.9}$$

and define a threshold ϵ such that if $D > \epsilon$, an error is said to have occurred.

In our model as we have described it, regions R_{0i} and R_{1i} are adjacent. This allows for spontaneous collapses that are near the boundary of the two regions to produce $|\alpha_c|^2$ and $|\beta_c|^2$ that are similar to $|\alpha_s|^2$ and $|\beta_s|^2$, respectively. For instance, if the superposition state is supposed to be $\frac{1}{\sqrt{2}}(|R_{0i}\rangle + |R_{1i}\rangle)$, a collapse very close to the boundary between R_{0i} and R_{1i} will produce a state that is very similar to this superposition state and is unlikely to result in an error. This leads to a host of complexities for our model – we would need to find a probability distribution for D based on the initial superposition state's wavefunction, but there are many such states that could describe the superposition state, and their different properties can lead to radically different probability distributions for D; a superposition wavefunction that is more heavily concentrated at the border between R_{0i} and R_{1i} will have a probability distribution that is more favorable to small values of D than a superposition wavefunction that is more heavily concentrated at the outer edges of $R_{0i} \cup R_{1i}$.

Another approach for defining error would be relative to the distance from the collapsed Gaussians centered somewhere in the region $R_{0i} \cup R_{1i}$. Instead of defining D as in (4.9), we would define it as the distance from the specific collapsed Gaussian that most closely approximates the intended state in terms of $|R_{0i}\rangle$ and $|R_{1i}\rangle$. For instance, if the superposition state is supposed to be $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, D would be measured relative to the collapsed Gaussian on the center of the boundary between R_{0i} and R_{1i}.

We could avoid these complexities altogether by putting a spatial gap between R_{0i} and R_{1i} regions, so that instead of these being adjacent, they are separated by a distance d. The value of d can be selected so that for our selection of error threshold ϵ, for either way of calculating D, any collapsed wavefunction will leave us with $D > \epsilon$ for all superposition states the bit systems are supposed to be in during the computation. Thus, the value of d is a function both of ϵ and the minimum intended values for $|\alpha_s|^2$ and $|\beta_s|^2$. Larger values of ϵ and smaller minima for $|\alpha_s|^2$ and $|\beta_s|^2$ would require d to be larger in order to ensure that

any spontaneous collapse will be counted as an error.

Both of these attempts to determine whether a collapse of an intended superposition state causes an error have implicitly assumed that the collapse happens at the end of the computation, after all computational operations (besides measurement) have been performed. While a collapse may not result in a state that differs largely from the intended state in terms of $|R_{0i}\rangle$, $|R_{1i}\rangle$, and $|tails_i\rangle$, the wavefunctions underlying the intended and collapsed state may differ significantly, meaning that subsequent computational operations can act on them in significantly different ways, resulting in error later in the computation. For instance, in Figure 4.5, both the intended superposition state in B and the state in A that collapsed near the boundary between the R_{0i} and R_{1i} regions can be described by

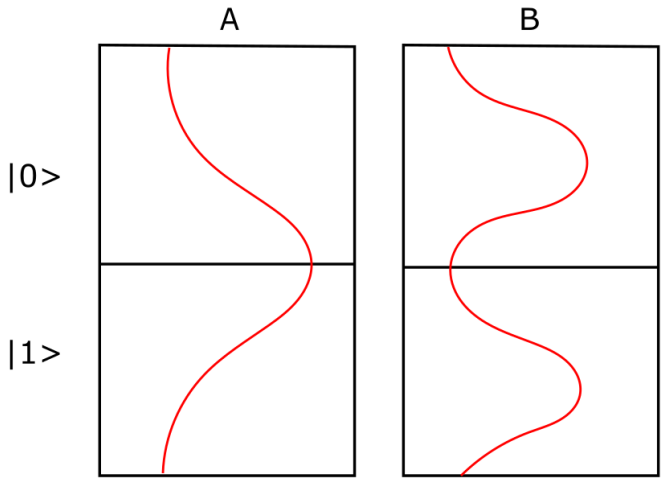

Figure 4.5: Both the collapsed state in A and the intended superposition state in B will be closely approximated by $\frac{1}{\sqrt{2}}(|R_{0i}\rangle + |R_{1i}\rangle)$, though a unitary operation will act differently on each.

$$(\tfrac{1}{\sqrt{2}} - \gamma)(|R_{0i}\rangle + |R_{1i}\rangle) + \gamma|tails_i\rangle \tag{4.10}$$

but a unitary operation can act on them in significantly different ways, resulting in states that cannot be described as the same computational state. Thus, the precise shape of the wavefunction underlying the computational state can make our GRW Register Machine more error-prone than a naive application of these error thresholds would suggest. A collapse that does not on its own cause an error could result in uncertainty about how future operations are acting on the state, and hence what computational operations are actually being performed.

Thus, a complete treatment of error in this register machine model would require precise specifications of the shape of the wavefunction underlying our computational states and the ways that unitary operators change the wavefunction. The properties of GRW imply that the minute physical details of a computational model are tied to the noise that it must experience from GRW's collapse dynamics.

An analysis of error for our GRW Register Machine relies on the particular physical properties of the machine, and similarly an analysis of the error caused by GRW's dynamics for some other model of computation would rely on the particular physical properties of that model. The error experienced by our model of computation is highly contingent on the minute physical details of the model; changes of the size or placement of the R_{xi} regions or size of the bit systems would change our assessment of the noise the model experiences, to say nothing of how more significant changes to the computational basis or the nature of the computer would affect our analysis of error.

Additionally, stochastic collapses would not be a source of error in other formulations of quantum mechanics, such as Bohmian mechanics or Many Worlds.[4] For instance, the bit errors described in this paper occur without any flaws in the construction of the register machine and without any environmental decoherence, because GRW's dynamics themselves produce noise. Generally, the unique features of different formulations of quantum mechanics should be taken into account when constructing models of computation in those theories and may have implications for the design and construction of the real quantum computers that we build.

[4]See (40) for an overview on different formulations of quantum mechanics, (16)(34) for discussions of Bohmian Mechanics in particular, and (64)(75) for discussions of Many Worlds

4.5 Lessons

The investigation of this GRW Register Machine model leaves us with three major lessons: The first is about what features are particular to computing in GRW quantum mechanics – the idiosyncratic properties of GRW require considerations of how a computer is constructed that do not exist in other formulations of quantum mechanics. Relatedly, the second lesson is that the computational noise produced by GRW's dynamics is an inherent and unavoidable part of any model of computation in GRW that must be taken into account. The third is that, even though GRW's stochastic dynamics acts as source of noise that is unique to GRW, it still generally leads to error that fits into known categories of error in the fault tolerance and error correction literature.

First, this model demonstrates how the unique features of GRW's dynamics act as a new source of error that is unique to GRW and is innately tied to the physical parameters of GRW computers. This model only describes bit error due to spontaneous collapses, and such a GRW register machine would also be susceptible to operator error. Even so, our model presents challenges that would not be faced by similar register machine models in some other formulations of quantum mechanics; the GRW Register Machine suffers from bit error even when it is in perfect isolation, and the precise underlying wavefunction must always be characterized to calculate the risk of error when quantum operations are allowed. These features are not universal to formulations of quantum mechanics; Bohmian Mechanics, for instance, would have neither of them.[5] Thus, we see the necessity of selecting a precise formulation of quantum mechanics if we are to have a robust understanding of quantum computing.

Because the error produced by GRW's spontaneous collapses is inherently tied to precise details of how a computer is built, the physical construction of a computer must be included

[5]See footnote 4.4

in the development of a GRW error model that describes it. For instance, minimizing a computer's reliance on precise positions and particularly superpositions of position could reduce the computational noise caused by GRW's dynamics – our GRW Register Machine is a particularly poor GRW computer in this sense, as its computational states are directly tied to particle positions. But the quantum computers that we actually build will typically look quite different from the register model presented in this paper. There are a host of ways that quantum computers may be built (23)(39)(71). The noise caused by GRW's dynamics will not affect these different kinds of computers equally; computers that are less susceptible to noise from spontaneous collapses in position will be able to perform better than computers that are more reliant on their component's positions. These considerations arise from the uniquely noisy nature of the position observable in GRW that is not present in other quantum theories.

The second lesson is that the either the noise predicted by GRW acts as a lower bound for the noise experienced by a quantum computer, or the reliability of a quantum computer serves as empirical evidence against GRW with particular parameter values. Given some values for the parameters α and τ, GRW will predict some noise for a quantum computer, either through bit error as seen in the error model presented about or through operator error, as the operator systems are susceptible to the same tails collapses. The noise from GRW's dynamics acts as a lower bound for the amount of noise that this computer will experience, should it be built in a universe governed by GRW with the selected values of α and τ.[6] If the actual error experienced by a quantum computer is less than GRW would predict, given selections of α and τ, it constitutes evidence against those values of α and τ being included in the empirically allowed window of GRW parameters. If, however, if the error experienced by a quantum computer cannot be reduced below what GRW predicts and also matches the types of error GRW predicts, then we may have evidence for GRW.

[6]Building a computer such that the base rate of noise given by GRW's dynamics is extremely low is possible if one can minimize the computer's reliance on positions and entanglements with positions and thus the effect that spontaneous position collapses will have.

The third lesson is that, although the error caused by GRW's dynamics is governed by features which are unique to GRW, the kinds of error that GRW's stochastic dynamics produce fit in to known types of quantum computational error. The bit errors described in our register machine model fall into the categories of qubit loss (when a particle collapses to a position outside the computer), qubit flips (when a particle collapses from $|R_{0i}\rangle$ to $|R_{1i}\rangle$ or vice versa), and incorrectly applied gates (as when an intended superposition collapses during a quantum computation, causing the wrong unitary operation to be applied at a later step, or even when an operator system itself collapses into wavefunction tails and fails to act as intended). Thus, we should expect that known error correction and fault tolerance methods to be at least somewhat effective at mitigating this error. However, GRW-specific computational error is tied to particular features of GRW's dynamics, so the effects of this noise can be minimized (or exacerbated) due to choices in a computer's design that would not affect non-GRW error in the same way. For instance, we can minimize GRW-specific error in our register machine by increasing the size of the bit state regions and thus reducing the likelihood that a tails collapse putting the bits outside of their intended region occurs.

There is much potential future work to be done in the realm of GRW computational error, both in making more robust and realistic models of GRW error for increasingly realistic quantum computers and looking at the effectiveness of various fault tolerance and error correction procedures in mitigating the effects of GRW error.

Chapter 5

GRW Dynamics and Supertasks

5.1 Introduction

GRW quantum mechanics imposes constraints and considerations for models of physical computation within it that differ from those seen in classical mechanics and other quantum theories, as a result of the inherent noisiness of its dynamics. This also applies to the case of supertasks, computational tasks which involve an infinite number of computational steps. Much of the discussion of physical supertasks focuses on the relationship between spacetime considerations and supertasks (e.g. (27)(37)(43)(44)). But in the case of GRW, GRW's stochastic dynamics limits the ways in which supertasks could be executed; because the number of GRW collapses a system undergoes increases with its number of particles entangled in position and the passage of time, the ability of computing systems in GRW to execute supertasks will depend greatly on the manner in which they are supposed to execute an infinite number of steps. For instance, GRW computers would be inherently incapable of executing supertasks that take an infinite amount of time or many supertasks that require an infinite number of wavefunction collapses. GRW's dynamics discriminates between infinite

and finite time supertasks and even between supertasks based on the nature of the correlations between the components of computing systems. This provides an interesting example of how dynamics can affect the physical possibility of supertasks and how the capabilities of physical supertasks can depend on the precise details of how they are supposed to be implemented and the precise details of the dynamics which governs the systems executing them.

5.2 Supertasks in GRW

A supertask is a computational task that consists of at least a countably infinite number of tasks. Supertasks can either take place over an infinite time or over a finite time (from the perspective of the system executing the supertask). In an infinite-time supertask, the system is given infinite time to operate at a finite processing speed; in a finite-time supertask it is given a finite amount of time in which an infinite number of steps are executed. Completing an infinite sequence of steps in a finite amount of time can be accomplished with an exponential speedup in the rate at which steps are completed or an infinite number of steps being executed in parallel. We will look at how GRW's stochastic dynamics square with these different approaches of implementing supertasks.

Because GRW collapses occur based on the passage of time, whether a supertask is meant to take place over an infinite or finite time interval might determine whether it is in principle impossible according to GRW's dynamics. The collapse rate in GRW also depends on the number of particles entangled in position, so the degree of correlation among the positions of particles in a computing can also play an important role. We will look at how all of these factors affect the possibility of supertasks in a world governed by GRW's dynamics. GRW's dynamics affect different cases in different ways and can rule out supertasks in principle in some cases but not others, even disregarding the general infeasibility of building systems

that could execute an infinite number of tasks.

5.2.1 Finite Time Supertasks

A finite time supertask is an infinite sequence of steps which are completed in a finite temporal distance along the worldlines corresponding to the components of the machine executing the task. We will consider two ways in which such a task could be executed. First, we are given an infinite number of particles with the goal of executing an infinite number of tasks in a finite amount of time in parallel. Second, we are given a finite number of particles, but are granted the ability to infinitely speed up the rate at which tasks are executed, so that an infinite number of tasks can be executed in a finite amount of time. Because the GRW collapse rate depends on the number of particles entangled in position, these different approaches result in different behaviors.

Infinite Particles

Because GRW's dynamics ties the collapse rate of a system to the number of particles which are entangled in position, we will have to consider the cases of an infinite number of entangled and unentangled particles separately. Let us first consider how our supertasking computer behaves if it is composed of particles that are perfectly correlated in position then consider its behavior if it is composed of particles that are uncorrelated in position.

An infinite number of particles in the perfectly correlated case would encounter a serious problem: an energy catastrophe in which the wavefunction is compressed to a delta function. The collapse rate of our system goes to infinity – meaning that the system's state is multiplied by narrow Gaussians an infinite number of times in any finite time interval.[1] So imagine our

[1]For more general discussions of collapses in GRW, see (12)(13)(14).

initial wavefunction is a Gaussian of width σ,

$$\psi_0 = \beta e^{-\frac{1}{2}(\frac{x}{\sigma})^2}. \tag{5.1}$$

Because the rate of collapse goes to infinity as $N \to \infty$, we can ignore wavefunction dispersion — the time between collapses in which dispersion could act goes to zero. Each collapse multiplies this wavefunction by a Gaussian of the form

$$\psi_{col} = \beta_{col} e^{-\frac{1}{2}(\frac{x-x_{col}}{\alpha})^2}, \tag{5.2}$$

where x_{col} is the location of the collapse. The expected location of the collapses is the origin; for simplicity, we will put our collapses at the origin, though even if the collapses occur elsewhere the same infinitely fast compression of the wavefunction will occur. So after one collapse we have

$$\psi_1 = \beta_1 e^{-\frac{1}{2}(\frac{x}{\sigma})^2} e^{-\frac{1}{2}(\frac{x}{\alpha})^2}. \tag{5.3}$$

The next collapse occurs immediately afterward, giving

$$\psi_2 = \beta_2 e^{-\frac{1}{2}(\frac{x}{\sigma})^2} e^{-(\frac{x}{\alpha})^2}, \tag{5.4}$$

and, after n collapses, we have

$$\psi_n = \beta_n e^{-\frac{1}{2}(\frac{x}{\sigma})^2} e^{-\frac{n}{2}(\frac{x}{\alpha})^2} \tag{5.5}$$

Because the collapse rate is infinite, $n \to \infty$ in any finite time interval, and our wavefunction converges to a delta function in position. This results in an energy catastrophe, in which an infinite energy wavefunction would be entirely located at a single point in configuration space.

Our computing system becomes an infinite energy singularity – or rather, an infinite number of these singularities, one for each particle in the system. The computational usefulness of a system that results in this outcome is difficult to defend.

The uncorrelated case is more nuanced. For simplicity, assume that our qubits are N_q-particle systems that are all perfectly uncorrelated in position with each other, but there is perfect correlation among their constituent particles.[2] We can treat an arbitrary qubit as being described by a Gaussian wavefunction centered at the origin, and consider the probability that it collapses outside of an arbitrary region described by $x \in [-r, r]$, where r is a finite number. For each qubit, there is a positive probability p_c that the qubit will collapse once in any finite time interval and a positive probability p_t that this collapse will be centered outside of $[-r, r]$. This gives a lower bound of $p_c p_t$ for the probability that a given qubit will undergo at least one tails collapse, since the qubit could collapse multiple times, and any of these collapses could center the qubit outside of this region. Thus, there will be a positive probability for each qubit to collapse outside of this arbitrary finite region in which it is supposed to inhabit. Given that there is an infinite number of particles, there will be an infinite subset of these particles that undergo these tails collapses, and the proportion of these that collapse into wavefunction tails will be equivalent to the probability that any particular qubit collapses into a wavefunction tail. So a portion of the qubits greater than $p_c p_t$ will collapse out of any arbitrary finite region. A supertask in these conditions must be robust to a random subset of its bits disappearing outside of any given finite region in which they are supposed to lie.

This problem similarly applies to the storage of an infinite amount of information using an infinite number of particles. Spontaneous collapses introduce a positive probability for a stored bit to collapse into a wavefunction tail over any nonzero time interval. If an infinite

[2] A similar analysis will hold if the particles are partially correlated in position, as there will be a nonzero probability that each collapse will put the partially correlated infinite particle system collapses into a state which breaks the entanglement.

subset of an infinite number of qubits undergoes tails collapses in any nonzero interval of time, the computing system cannot store any particular value with infinite precision, as it is continuously losing bits. Error correction would be of limited help as well, as the same risk of loss would also apply to any error correcting systems. This means, for instance, that any real number generated by a supertasking GRW computer would be maximally unreliable. Any passage of time would result in an infinite subset of qubits collapsing to tails. If, for instance, we had used our GRW supertasking computer to generate a random real number, this could still leave us with a random real number, though this random number would change from moment to moment. This instability prevents a GRW computer from storing a *particular* real number in computations.[3] Any supertask which requires an unbounded amount of storage for its execution – even one that could be executed by a classical supertasking computer – could fall prey to this unreliability. For instance, if an algorithm used to decide the Goldbach conjecture required unbounded storage via unbounded particles to check larger and larger numbers, a supertasking GRW computing would not be able to execute that algorithm.

Infinite Speedup

Now consider a supertask in which each successive step of the computation takes half as much as the step which preceded it. So if the supertask is completed in some finite time t_s, step n takes $\frac{t_s}{2^n}$. Let's consider two different categories of computations that our supertask could undertake: Those which require an infinite number of wavefunction collapses and those which require a finite number (including zero) of wavefunction collapses.

Executing an infinite number of collapses results in a similar problem with wavefunction collapses as we saw in the correlated infinite particles case. In order to execute an infinite number of collapses in a finite time interval, we must increase the collapse rate so that it

[3]This leads to questions about the extent to which an infinite particle supertask's ability to generate a random real number is meaningfully different from its ability to generate random bits.

goes to infinity in a finite time interval. But the collapse rate is $\frac{N}{\tau}$, and τ is a constant, so the number of entangled (in position) particles N of our computer must grow fast enough that for finite t_s there is a positive probability that the number of collapses goes to infinity,

$$\lim_{m\to\infty} e^{-\frac{Nt_s}{\tau}} \frac{(-\frac{Nt_s}{\tau})^m}{m!} > 0. \tag{5.6}$$

Given that t_s is finite, this requires that $N \to \infty$. So completing a supertask which uses an infinite number of collapses would require an infinitely large computer, and put us in the catastrophic situation we just discussed above in the infinite number of correlated particles case.

But not all supertasks will require an infinite number of collapses. In fact, classical computational supertasks can be executed by quantum computers entirely without collapses. There is a positive probability that a finite number of particles will remain in their intended region over a finite time interval, that is, they will not undergo a tails collapse. Thus, given an infinite speedup, supertasks requiring only a finite number of collapses could be allowed in principle by GRW's dynamics. Tails collapses mean that the probability that a finite-time GRW supertask will successfully complete will always be less than one, but this would be the case for ordinary GRW computations as well. Yet with some positive probability, a supertasking GRW computer could compute any problem that a supertasking classical computer could compute, with the caveat we discussed above – a GRW computer cannot use an infinite number of particles to reliably and precisely store a value.

5.2.2 Infinite Time Supertasks

Because the probability of a GRW system collapsing in a time interval is governed by a Poisson distribution, in an infinite time interval, the probability of having no collapses goes to zero. This immediately rules out any supertask that would fail given one collapse – one

that requires a superposition of position states to hold throughout its entire process. But as the expected number of collapses in an interval t_I is $\frac{Nt_I}{\tau}$, our computer will undergo an infinite number of collapses in an infinite time interval. We would need our computation to be robust despite undergoing an infinite number of collapses.

The central issue is then how an infinite number of collapses over an infinite amount of time will affect a finite particle system. This depends on whether the expected width of the wavefunction will be nonzero; that is, whether the wavefunction converges to a delta function, as we saw in the infinite correlated particle case, or if it maintains some nonzero width. If there is no lower bound for the expected width of the wavefunction, then it will converge to a delta function, as we saw in the infinite particle case, except after an infinite rather than a finite time interval. However, if the expected width of the wavefunction does not converge to 0 as $t_I \to \infty$, we can describe our system as undergoing a random walk in which each collapse counts as a step. The average step size is the expected width of the wavefunction. For a random walk with a Gaussian step distribution where b^2, is the mean-squared displacement of any given step, the mean-square displacement after n steps is (from (9))

$$\langle R_n^2 \rangle = nb^2. \tag{5.7}$$

Since b^2, the square of the width of the wavefunction, is nonzero, as the number of collapses n goes to infinity, $R^2 \to \infty$, meaning that our computing system will leave any finite region of the universe. We are left with a "reverse space invader" or "space evader", in the vein of (42) (see also (45)). To execute an infinite time supertask, the positions of every component of a GRW computer must be irrelevant in the strongest possible sense to whether the supertask is executed, as the parts of the computer would leave any finite region of the universe. The computation would need to be robust to the computer's exit from the universe – quite an absurd requirement! In this sense, GRW's dynamics rule out infinite time supertasks. Error

correcting techniques will also be of no help – any system used to correct errors would itself be a space envader. And attempting to compress the wavefunction so that the random walk travels a finite distance would result in an energy catastrophe as the wavefunction becomes a delta function. Either way, GRW's dynamics is incompatible with infinite time supertasks.

5.3 Remarks

Because the stochastic element of GRW's dynamics is inherently linked to time and the number of particles in a system, some ways that one might attempt to execute a supertask will be rendered impossible by GRW's dynamics or at least limited compared to other ways one could attempt to execute a supertask. We can summarize the situation for supertasks governed by GRW's stochastic dynamics as:

	Infinite Time	Infinite Particles	Infinite Speedup
Perfectly Correlated	Singularity or Space Evader	Singularity	Finite collapse STs possible
Uncorrelated	Space Evaders	Limited by Noise	Finite collapse STs possible

GRW's dynamics does not allow infinite time supertasks, but can allow finite time supertasks, with some caveats. Finite time supertasks that involve a finite number of wavefunction collapses and do not need unbounded storage (via an unbounded number of bits) would be allowed. GRW introduces a chance of failure for any computation, but it guarantees failure for certain kinds of supertasks – such as those which utilize an infinite timeline or which require an infinite number of wavefunction collapses. Even the nature of the correlations among particles can differentiate whether some supertasks are possible, as we saw in the infinite particles case.

Though this discussion has focused on the particulars of GRW quantum mechanics, we should also note a more general lesson. The dynamics of physical theories can limit the capabilities of supertasks and can lead to distinctions in what implementations of supertasks are possible in principle. Not only can dynamics separate supertasks that take place over an infinite amount of time versus a finite amount of time, but it can also distinguish whether supertasks are possible based on a wide variety of physical properties. The properties of physical theories can affect the possibility of supertasks in complicated and surprising ways.

Chapter 6

On Epistemically Useful Physical Computation

6.1 Introduction

This discussion emerges from two fundamental questions: What is physically computable? And what is the relationship between Turing computability and physical computability? As Turing computability is the central force of computability theory, the former question is often posed in terms of the latter (e.g. in (7)(20)(37)(65)(78) and countless others). Piccinini's discussion of the physical Church-Turing thesis in (59) and (55) follows this format. He argues that, if notions of computability are to be linked to what is epistemically useful to finite observers, a modest version of the physical Church-Turing thesis likely holds. This Modest Physical Church-Turing Thesis states that what is Turing computable acts as an upper limit for what is physically computable, given some constraints on what is considered a physical computation. These constraints are meant to restrict discussion to physical computations which could be epistemically useful to finite observers.

Though the Modest Physical Church-Turing thesis is plausible, we will see that the account of what counts as an epistemically useful physical computation that Piccinini uses to argue for this thesis needs more concrete conceptual grounding. In particular, I will argue that it begs the question regarding what physical processes one considers possible computational operations and implicitly fills in this gap with the assumption that physical computational operations correspond to processes that can be executed by physical systems that can be built using classical computational processes on physical systems. Determining which physical processes can correspond to epistemically useful computational operations leads to a bootstrapping problem, in which our criteria for determining which physical processes finite observers can implement as part of a computation require us to already know which physical processes finite observers can implement to build systems to implement that computation. Piccinini implicitly assumes a classical computation-based position. This position is not without merits – I will argue that it is a good default position in 6.4.2 – it is also not the only possible position, and it might even be ill-suited at describing what is possible for finite observers in some physical theories. Additionally, this position may lead us to overestimate the likelihood that the Modest Physical Church-Turing thesis holds.

Section 6.2 discusses what is meant by physical computability in the context of this paper. Section 6.3 discusses what a model of physical computation is, in light of Piccinini's conditions for ensuring physical computational models are epistemically useful to finite observers. Section 6.4 argues that, given a physical theory, there is an inherent freedom for selecting physical operations to ground what counts as a model of computation in that theory. This results in a bootstrapping problem unless we impose a selection of what physically possible processes to consider possible building blocks for models of computation in a theory. The selection, implicitly made by Piccinini, derived from a classical computational understanding of operations, though a particularly compelling response to this freedom, is not without its weaknesses. Section 6.5 concludes.

6.2 Physical Computability Questions

Recall that, using Turing machines in classical universes whose spatial dimensions are described by hyperreal – rather than real – lines, Aitken and Barrett show that what is computable using a Turing machine depends on the physical theory in which it resides (3)(4). They conclude that, generally, what is physically computable depends on the physical theory. Additionally, they propose that questions of physical computing power can only be answered relative to a physical theory and a computational model constructed relative to this theory. Thus, questions about physical computability should be answered relative to a particular computational model constructed relative to a particular physical theory.

On its own, this proposal tells us nothing about how we should select models of computation relative to physical theories. But determining the limits of physical computability and whether different versions of the physical Church-Turing thesis hold relies on some assessment of what counts as a model of physical computation in some physical theory. If we do not properly restrict how computational states and operations can be mapped to physical states and operations, then we can end up with some version of pancomputationalism where every physical system performs either some computation or every computation.[1]

One might reasonably hope that the selection of appropriate physical computational models can be done in a principled way, ideally by catering our selection of computational model to leverage a physical theories properties while reflecting what observers governed by this theory could possibly achieve. We will explore this approach in Section 6.4.1, and find that it succumbs to a bootstrapping problem without supplementary assumptions that do not arise from the physical theory itself. Thus, Piccinini's Usability Constraint must – and in fact does – require additional specification of what operations finite observers can utilize which do not arise from the physical theory or the Usability Constraint itself. The necessity of additional

[1] see (57) for an overview, or (49)(58)(54) for some examples of critical responses to pancomputationalism

constraints that do not arise from a particular physical theory on allowed computational operations and arguments for and against Piccinini's position will be discussed in Section 6.4.2.

6.3 Models of Physical Computation

For the purposes of this paper, a physical theory $\mathcal{T}$ will specify sets of objects J, possible physical states S_j for these objects, and dynamical rules $\mathcal{R}$ which describe how the states of these objects evolve and interact. For instance, the objects of J may be particles or perturbations of a field, and will constitute the physical systems used to construct a model of computation. The states S will describe the properties of these objects, such as location or energy level. And the dynamical rules $\mathcal{R}$ determine how these objects and states can evolve. This definition is meant to be broad enough to encompass a wide variety of physical theories.

Given an appropriate physical theory $\mathcal{T}$, there are an immense number of ways to construct a model of computing in that theory. Any process can be considered to be a computation of almost anything if one is liberal enough with what one considers a computation, as the pancomputationalism literature demonstrates.[2] We will want to consider models of physical computation that follow the Aitken and Barrett's prescription in light of Piccinini's epistemological concerns. To give us a tractable starting point, I will define a model of computation as follows.

A model of computation relative to $\mathcal{T}$, $\mathcal{M}_{\mathcal{T}}$, will consist of finite numbers of units of information ("bits" of some sort) $\mathscr{B}$ and operations $\mathscr{O}$ that can be performed on these units of information; these will arise from information-carrying objects $B \subseteq J$ and operator systems $O \subseteq J$ which act on the elements of B according to dynamical rules $\mathcal{R}$ to change their

[2]See footnote 1.

corresponding physical states. There may not be a one-one correlation between units of information $\mathscr{B}$ and elements or even subsets of B. For instance, a unit of information may be carried by one particle or multiple particles, perhaps even within a single computer. Importantly, the physical states of B serve to transmit information, and the evolution of these states according the dynamical rules $\mathcal{R}$ of the theory and through the influence of operator systems allows computation.

Piccinini proposes a usability constraint to restrict discussion of physical computation to those physical computational models that would be epistemically useful for finite observers:

Usability Constraint: *if a physical process is a computation, it can be used by a finite observer to obtain the desired values of a function*[3] (55).

This constraint is broken down into four sub-constraints on physical processes:

> "An *executable* physical process is one that a finite observer can set in motion to generate the values of a desired function until it generates a readable result."
>
> "An *automatic* physical process is one that runs without requiring intuitions, ingenuity, invention, or guesses."
>
> "A *uniform* physical process is one that doesn't need to be redesigned or modified for different inputs."
>
> "[A] *reliable* physical process is one that generates results at least some of the time, and, when it does so, its results are correct." (p. 741 of (55))

Together, these are meant to describe the conditions that a physical process must fulfill in order for it to possibly be an epistemically useful physical computation. A physical process

[3]Here Piccinini only considers processes that fit this constraint to be computations. I will instead distinguish between usable physical computations that fit this constraint and physical computations more broadly, which may or may not fit this constraint.

that fails any one of these sub-constraints would not be considered a physical computation that could be epistemically useful to finite observers.

Our concern will be a particular aspect of the executability criterion. To be *executable*, a physical process must have inputs and outputs that are readable for a finite observer, the ability to solve problems that can be "defined independently of the processes that compute them", the ability to be repeated, the ability to set the computing system into a particular initial state, and *a physically constructible manifestation* (55). This limits discussion of physical computation to things which can be executed and epistemically useful (in a particular sense) for finite observers.

The last component of the executability condition – that the physical process corresponds to some system that can be physically constructed – will be the main target of our concern, because an investigation of this criterion reveals an unrecognized assumption that guides the rest of Piccinini's work on the Modest Physical Church-Turing Thesis. What we now want to consider is how this restriction would play out when developing models of computation relative to a physical theory. For the sake of avoiding confusion with other concepts, we will use "build" where Piccinini uses "construct".

6.4 Determining What is Buildable

We will see that Piccinini's executability sub-constraint, due to ambiguities in how to determine what is buildable, can leave open a wider range of possibilities for computational operations than seems desirable and which seems to have been acknowledged in the literature. By investigating ways one can describe what counts as "physically constructible" (what we call "physically buildable" to avoid confusion with other ideas tied to the word "constructible"), we will see that Piccinini implicitly supplements this concept with a notion

of what operations finite observers can use which is based in classical computation, rather than on the physical theory itself, in contrast with the prescription described by Aitken and Barrett. Nonetheless, I will argue that adapting the implicit notion underpinning Piccinini's discussion of his usability constraint to particular physical theories is a strong starting point for investigating physical computation, though we should be explicitly aware of its nature as an assumption we are making.

6.4.1 Buildable Physical Systems & a Bootstrapping Problem

If our concern is what a physical theory allows finite observers to do, not all physically possible system states may be useful for computation. For a model of physical computation to pass Piccinini's executability condition, finite observers must be able to reliably build devices that manifest it. One then needs to understand what processes finite observers can reliably use to build physical systems to act as computers. If a finite observer cannot systematically put a system in (or extremely near to) a state, then that state cannot be part of a model of computation that is useful for finite observers, even if the state can be realized in a world governed by theory $\mathcal{T}$. This applies to both the information-carrying systems and the operator systems; we need systematic processes within the theory to built the elements of B and O and to set them up for a computation, otherwise these states cannot be epistemically usefully harnessed. These restrictions are motivated by epistemic considerations – they are meant to limit usable models of physical computation to those which are not just physically possible simpliciter but are physically possible ways for an observer within the physical reality to use other systems within the same physical reality to solve some problem (disregarding surmountable technological limitations). Ideally, we would be able to define a notion of "buildability" that emerges from an account of finite observers in a particular physical theory alone. However, we will see that the situation seems not to be so simple, and deciding what counts as buildable requires us to make stipulations that

do not arise from the physical theory itself.

What can be built within different physical theories obviously varies; for instance, quantum mechanics allows the construction of Hadamard gates, while classical mechanics may not. Also, what building procedures are available may vary from one theory to another. One theory may allow infinite-step processes to be completed while another does not, or, more normally, one theory may allow access to different physical systems and different ways physical systems can be manipulated than another. If theories allow different processes to manipulate physical systems, they may also allow different operator and bit systems to be built and perhaps different $\mathscr{B}$ and $\mathscr{O}$ which can be used in computational models relative to the theories. But finite physical systems can implement many processes that should not be included in an account of epistemically useful physical computation. The task for understanding useful physical computing is determining which of the physical processes that finite systems can implement can be used to build operator and bit systems for use in physical computers.

Suppose we want to determine what models of computation finite observers can build relative to some physical theory. To determine what can be built for use in a model of computation relative to some physical theory, we must first determine what physical processes in the theory can be used for building computer components. But here we run into a problem – building some component for a computer requires a set of operations we can perform on systems to change their states in order to build that component. But this set of operations we can perform on systems is precisely what we are attempting to discover, as these are the operations which can be used as part of a model of physical computation. So we have a bootstrapping problem – we want to know which physical processes a physical theory allows finite observers to reliably implement (that would be used in computers), but to discover this we must already know what physical processes the physical theory allows finite observers to reliably implement (that would be used in building computers).

Let us make this more concrete. Imagine we have some physical theory $\mathcal{T}$ with a set of objects J which have possible physical states S_j and dynamical rules $\mathcal{R}$ which describe how these states evolve. Now we want to know what bits $\mathscr{B}$ and operations $\mathscr{O}$ could be used in epistemically useful models of computation $\mathcal{M}_{\mathcal{T}}$ in this physical theory. To find this out, we must know what the sets of bit systems B and operator systems O can be; these are the physical systems finite observers can build to act as components in computing systems. To determine what B and O can be, we must know how the states S_j of physical systems J can be systematically manipulated to build B and O. To determine this, we would have to already know what operations we can perform on the states of physical systems, but *this is what we were trying to use buildability to figure out.* At this point, the requirement that there be a "physically constructible [buildable] manifestation" of a computation needs supplementation to be able to do its job as a constraint on what can be considered an epistemically useful physical computation.

Can the rest of Piccinini's Usability Constraint free us from this problem? First, look at the rest of the executability sub-constraint. Again, this requires that there be readable inputs and output, the ability to solve problems defined independently of the processes that compute them, repeatability, and the ability to set computing systems into particular initial states, in addition to having a physically buildable manifestation. Whether a process is repeatable and whether a computing system can be set to a particular initial state are consequences of what is physically buildable – if we can build components to act out some physical process, then we can do so repeatedly and we can do so in order to initialize the state of a computing system. The ability to solve problems defined independently of the processes that compute them is a matter of what computations can be said to compute but not of what basic operations can be implemented on physical systems. Requiring readable inputs and outputs seems to show the most promise, since presumably finite observers cannot access the complete dynamical consequences of all of their actions. For instance, I may be able to throw a ball some classically noncomputable distance (in meters), but I certainly

cannot read off the real number corresponding to this distance. But readability itself is at least partially dependent on buildability, in that finite observers may be able to use physical processes to change an output they cannot read into one they can.

The other sub-constraints will be of no help. Whether a purported computing system can be considered "Automatic" will depend on what operations are considered to be automatic processes, which in turn depends on what can be built to act as a computing component and thus cannot be considered to be acting with intuition or ingenuity. Whether a process is Uniform over different inputs and whether a process is capable of getting correct Results are properties of a purported computation as a whole, not its individual components.

Thus, Piccinini's executability criterion leaves us with a bootstrapping problem: we want to figure out which physical processes we can implement on physical systems (as part of a model of computation), but to determine this we need to already know what these operations are (for use in building computer components to implement a model of computation). In the following section, we will discuss how to avoid this bootstrapping problem, and the solution described by Geroch – namely, to take operations corresponding to classical computational operations as a starting point – and why it is likely the best starting point for investigating physical computation, despite some weaknesses.

6.4.2 Avoiding the Bootstrapping Problem By Stipulating Constraints

Developing a notion of what is buildable in a physical theory requires us to already have a concept of what operations and systems are allowed in a notion of buildability. Because of the bootstrapping problem, we cannot simply read a notion of buildability off of a physical theory's treatment of finite systems. Instead, we must stipulate on the theory some external criteria to determine which physical processes finite systems can implement to consider

relevant to physical computation. These criteria do not arise directly from the physical theory or its properties but instead are external constraints on which physically processes are allowed in building procedures and hence which operations and states are considered buildable. This helps us determine what to count as an epistemically useful model of physical computation in a particular physical theory.

In order avoid the bootstrapping problem, we could appeal to classical computational processes as a starting point. That is, we would assume that analogs to classical computational operations can be executed in a physical theory; namely, *we can execute finite-step processes where steps involve discrete changes in a physical system's state, and these discrete changes are classically (Turing) computable.* For instance, a rotation of a state through an angle described by a classically computable number is allowed whereas a rotation of a state through a classically noncomputable angle is not – unless we are able to reliably build an operator system that rotates a state through a classically noncomputable angle by using a finite number of classically-derived operations. This solution allows us to bypass the bootstrapping problem by selecting the physical processes that allow us to build systems to act as computational states and operations.

Geroch describes a version of this strategy in his discussion of plausible limits to quantum computability that the quantum computational literature implicitly assumes. He argues that unitary operators that rotate a quantum state through a classically noncomputable angle, though they are allowed by the laws of quantum mechanics, should not be permitted in a model of quantum computation, because a classical, stepwise method of building systems that perform these operations would require hypercomputation. (See chapter 8 of (32).) Though finite quantum systems can execute classically noncomputable unitary operations, Geroch argues for their exclusion from quantum computation based on an appeal to classical computational principles. Similarly, Piccinini assumes this classical computational strategy in his use of physical constructibility, as seen in his discussion of why "unconstrained appeals

to real-valued quantities" should not be considered part of epistemically useful physical computation: "[T]here is no reason to believe that a finite observer can use the Turing-uncomputable operations... to compute in the epistemological sense that motivates CT [the Modest Physical Church-Turing thesis] in the first place..." (55)

One reason for adopting the classical starting point for buildability – at least regarding the exclusion of real-valued quantities – noted by Piccnini, is that "There is no reason to believe that... unbounded precision is available to a finite observer" (55). Relatedly, the original arguments for what Piccinini calls the "Mathematical Church-Turing thesis"[4] – that "any function that is computable by following an effective procedure is Turing computable" – may bolster this position. Particularly, Turing argues in favor of the notion of computability given by Turing machines on the grounds that they limit the set of computational states and operations to those corresponding to what a human (with pen and paper) would be able to use (73) (reprinted in (74)). This suggests that Turing machines and Turing (classical) computable processes probably capture the types of processes that (human) finite observers can reliably execute, and thus are likely the best starting point for developing physical computation in some physical theory, even though the classical computational processes were not developed to most optimally leverage the properties of that particular physical theory.

Also, appealing to classical computational operations is a particularly salient strategy when investigating physical theories like quantum mechanics or relativity that are supposed to subsume physical theories that give rise to classical computation, such as classical physics. We can often assume that a physical theory at least allows classical finite state automata, which then provide us with information on some, if not all, B and O systems we may be able to build. This strategy also allow us to utilize an enormous amount of information about the processes we can use to build B and O systems in a theory, as we have a robust

[4]Not to be confused with either the Modest or Bold Physical Church-Turing Thesis

understanding of what classical computational processes look like and what they can do.

Though the strategy of appealing to classical computational processes is clearly a strong one, objections to it should be noted. Most significantly, it is not a strategy that emerges from considerations of a particular physical theory. Restricting ourselves to computational systems that can be built using classical processes and not theory-specific processes may unnecessarily limit the models of computation that we can build relative to a theory. Operations which the theory allows but which cannot be implemented by systems built using classical procedures cannot be used, and there may exist some systematic theory-specific methods for finite observers to build systems which carry out these operations. So while the classical approach may be a powerful starting point in theories that are amenable to classical computational operations, it may also unduly constrain the models of computation we can build relative to a theory because it may not allow us to fully leverage non-classical properties of that theory. To better utilize the particular properties of a physical theory, we may want to select some plausible theory-specific operations as our starting point instead of, or perhaps in addition to, the classical operations. The bit systems B and operator systems O that we deem credible will thus be whatever can be built using some preselected set of operations allowed by the theory's dynamics, similar to the classical solution but not necessarily reliant on concepts or operations from classical computing models.

6.5 Remarks

Tying our accounts of physical computation to what would in fact be epistemically useful for finite observers requires restrictions on what models of physical computation are considered. Piccinini gives a Usability Constraint which is meant to achieve this goal. But part of this constraint lacks adequate conceptual grounding, in lieu of which Piccinini has implicitly appealed to classical computing notions to determine which physical processes can be included

in epistemically useful models of computation. As Aitken and Barrett note, what is physically computable depends on a model of computation relative to a particular physical theory, so we may hope that our understanding of what is buildable, and hence usable, in a physical theory will emerge from the physical theory and its descriptions of finite systems. However, attempting to derive a notion of buildability directly from an account of finite systems in a physical theory can succumb to a bootstrapping problem, in which our notion of what is "buildable" is meant to allow us to determine which physical processes finite observers can reliable implement (as part of a computation), but determining what counts as buildable requires that we already know which physical processes finite observers can reliably implement (as part of building the components of computing systems). We are then left with a choice of how to select what physically possible states and operations we will permit for use in computation. In this situation, the classical understanding used by Piccinini and explicitly stated by Geroch is an especially strong option, though we should be aware of its status as one of many possible selections we can make when discussion physical computation.

Chapter 7

Conclusion

We began our investigation by looking at conceptual problems in the von Neumann-Dirac formulation of quantum mechanics and arguing that we should be wary of founding our understanding of quantum computing on this either inconsistent or incomplete physical theory. But this investigation has broader lessons for physical computation generally. By rigorously applying the Aitken/Barrett thesis – that questions of physical computation can only be answered with respect to a particular computational model relative to a particular physical theory – to quantum mechanics, we learn lessons about quantum computing, physical computation more generally, and the Aitken/Barrett thesis itself.

For quantum computing, this investigation shows some of the consequences that conceptual issues in the foundations of quantum mechanics and our solutions to these issues have for quantum computing. For instance, GRW's stochastic dynamics creates a host of considerations that must be taken into account when constructing models of computation in GRW. GRW's dynamics provide a unique source of computational error that is inherently tied to the precise details of how a computer is constructed and which would not be seen in other formulations of quantum mechanics.

For physical computation, rigorously applying the Aitken/Barrett thesis to a plausible physical theory has resulted in several lessons. We should not necessarily expect that we can build a canonical model of computation in a physical theory and acquire a good model of computation in that theory – Turing machines were somewhat poorly suited for GRW due to their reliance on position, but even more extreme problems may arise in other physical theories. Generally, when applying the Aitken/Barrett thesis to a particular physical theory, we should attend to and attempt to leverage the unique properties of the theory. Even in the most idealized examination of supertasks, we saw that the precise physical properties of a physical theory can have dramatic impacts on what supertasks can accomplish and how they could be executed. Universal claims about the potential power of physical computers seem particularly difficult to justify given the complications we encountered when examining the potential power of physical computers in just one specific physical theory.

There is still much work to be done on these topics. Similar projects could be done looking at computation in other formulations of quantum mechanics, especially formulations such as Bohmian Mechanics which have explicit dynamical rules that differ from those of the von Neumann-Dirac formulation. There is still much work that could be done exploring the computational error that arises solely from GRW's dynamics in terms of modelling what this error would be for realistic quantum computers, comparing this to real error rates for real quantum computers, and rigorously investigating the effectiveness of fault tolerance and error correction techniques in a GRW-governed universe. And there are surely many more lessons for physical computation that can be learned from rigorously applying the prescription of the Aitken/Barrett Thesis to other physical theories. Hopefully this dissertation serves to help others researching the foundations of quantum computing and the limits of physical computation in general.

Bibliography

[1] Scott Aaronson, "Multilinear Formulas and Skepticism of Quantum Computing" *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing - STOC '04* (2004) doi10.1145/1007352.1007378

[2] D. Aharanov and M. Ben-Or. "Fault-Tolerant Quantum Computation with Constant Error." *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing – STOC '97*, 1997, doi:10.1145/258533.258579.

[3] W. Aitken & J.A. Barrett "A Note on the Physical Possibility of Transfinite Computation" *British Journal of Philosophy of Science* 61: 867-874 (2010)

[4] W. Aitken & J.A. Barrett. "On the Physical Possibility of Ordinal Computation". 21 Mar. 2009, www.socsci.uci.edu/ jabarret/bio/publications/ncompSubmit.pdf.

[5] David Z. Albert, *Quantum Mechanics and Experience*. Harvard University Press, 2000.

[6] V. Allori, S. Goldstein, R. Tumulka, N. Zhanghì. "On the Common Structure of Bohmian Mechanics and the Ghirardi–Rimini–Weber Theory". *British Journal for the Philosophy of Science* 59: 353–389 (2008). arXiv:quant-ph/0603027.

[7] Pablo Arrighi and Gilles Dowek. "The Physical Church-Turing Thesis And The Principles Of Quantum Theory." *International Journal of Foundations of Computer Science*, vol. 23, no. 05, 2012, pp. 1131–1145., doi:10.1142/s0129054112500153.

[8] Guido Bacciagaluppi and Antony Velentini, *Quantum Theory at the Crossroads: Reconsidering the 1927 Solvay Conference* Cambridge, UK: Cambridge UP. Print (2009)

[9] Barber, Michael N., and B. W. Ninham. *Random and Restricted Walks: Theory and Applications*. Gordon and Breach, 1970.

[10] J.A. Barrett, "The Preferred Basis Problem and the Quantum Mechanics of Everything," *British Journal for the Philosophy of Science*. 56(2): 199-220. (2005)

[11] J.A. Barrett, *The Quantum Mechanics of Minds and Worlds* Oxford University Press: Oxford (1999)

[12] A. Bassi, Angelo, et al. "The Hilbert-Space Operator Formalism within Dynamical Reduction Models." *Journal of Physics A: Mathematical and Theoretical*, vol. 40, no. 45, 2007, pp. 13755–13772., doi:10.1088/1751-8113/40/45/015.

[13] A. Bassi, and D. Dürr. "On the Long-Time Behavior of Hilbert Space Diffusion." *EPL (Europhysics Letters)*, vol. 84, no. 1, 2008, p. 10005., doi:10.1209/0295-5075/84/10005.

[14] A. Bassi, and Davide G M Salvetti. "The Quantum Theory of Measurement within Dynamical Reduction Models." *Journal of Physics A: Mathematical and Theoretical*, vol. 40, no. 32, 2007, pp. 9859–9876., doi:10.1088/1751-8113/40/32/011.

[15] John Bell, "Against 'Measurement'" *Physics World* (Aug., 1990) pp. 33-40

[16] K. Berndl et al. "A Survey of Bohmian Mechanics." *Il Nuovo Cimento B*, vol. 110, no. 5-6, 1995, pp. 737–750., doi:10.1007/bf02741477.

[17] David Bohm, "A Suggested Interpretation of the Quantum Theory in Terms of 'Hidden' Variables. I" *Physical Review*, Vol. 85, No. 2 (Jan. 15, 1952), pp. 166-179

[18] D. Bohm and J. Bub, "A Proposed Solution of the Measurement Problem in Quantum Mechanics by a Hidden Variable Theory." *Reviews of Modern Physics*, vol. 38, no. 3, 1966, pp. 453–469., doi:10.1103/revmodphys.38.453.

[19] J. Bub. "Quantum Measurement Problem." *Routledge Encyclopedia of Philosophy*, doi:10.4324/9780415249126-q086-1.

[20] Paolo Cotogno. "Hypercomputation and the Physical Church-Turing Thesis." *The British Journal for the Philosophy of Science*, vol. 54, no. 2, June 2003, pp. 181–223., doi:10.1093/bjps/54.2.181.

[21] Nakesh Dattani and Nathanial Bryans, "Quantum Factorization of 56153 with only 4 Qubits" preprint arXiv:1411.6758 [quant-ph] (2014)

[22] D. Deutsch "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer" *Proceedings of the Royal Society of London A* 400 97-117 (1985)

[23] David P. DiVincenzo, "The Physical Implementation of Quantum Computation." *Scalable Quantum Computers*, 2005, pp. 1–13., doi:10.1002/3527603182.ch1.

[24] Matthias Egg and Michael Esfeld. "Primitive Ontology and Quantum State in the GRW Matter Density Theory." *Synthese*, vol. 192, no. 10, 2014, pp. 3229–3245., doi:10.1007/s11229-014-0590-3.

[25] Michael Esfeld. "The Primitive Ontology of Quantum Physics: Guidelines for an Assessment of the Proposals." *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, vol. 47, 2014, pp. 99–106., doi:10.1016/j.shpsb.2014.06.003.

[26] Michael Esfeld, and Nicolas Gisin. "The GRW Flash Theory: A Relativistic Quantum Ontology of Matter in Space-Time?" *Philosophy of Science*, vol. 81, no. 2, 2014, pp. 248–264., doi:10.1086/675730.

[27] Gàbor Etesi, and István Németi. "Non-Turing Computations Via Malament-Hogarth Space-Times." *International Journal of Theoretical Physics*, vol. 41, no. 2, 2002, pp. 341–370., doi:10.1023/a:1014019225365.

[28] Hugh Everett. "'Relative State' Formulation of Quantum Mechanics." *Reviews of Modern Physics*, vol. 29, no. 3, 1957, pp. 454–462., doi:10.1103/revmodphys.29.454.

[29] W. Feldmann and R. Tumulka. "Parameter Diagrams of the GRW and CSL Theories of Wavefunction Collapse" *Journal of Physics A: Mathematical and Theoretical*, vol. 45, no. 6, 2012, p. 065304., doi:10.1088/1751-8113/45/6/065304.

[30] Roman Frigg and Carl Hoefer, "Probability in GRW Theory" *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics* 38.2: 371-89. (2007)

[31] R. Geroch "The Everett Interpretation" *Noûs* Vol 18, No. 4, *Special Issue on the Foundations of Quantum Mechanics* (Nov., 1984), pp. 617-633

[32] R. Geroch *Perspectives in Computation* University of Chicago Press. Chicago, IL (2009)

[33] G. C. Ghirardi, A. Rimini, and T. Weber. "A Model for a Unified Quantum Description of Macroscopic and Microscopic Systems." *Quantum Probability and Applications II Lecture Notes in Mathematics* (1985): 223-32.

[34] Sheldon Goldstein. "Bohmian Mechanics." Stanford Encyclopedia of Philosophy, Stanford University, 27 Mar. 2017, seop.illc.uva.nl/entries/qm-bohm/.

[35] S. Goldstien, R. Tumulka, and N. Zhanghì "The Quantum Formalism and the GRW Formalism" *Journal of Statistical Physics* vol. 149 (2012): 142-201

[36] H. Greaves "Probability in the Everett Interpretation" *Philosophy Compass* Vol 2, No. 1, 109-128. (2006)

[37] Mark Hogarth "Non-Turing Computers and Non-Turing Computability." *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, vol. 1994, no. 1, 1994, pp. 126–138., doi:10.1086/psaprocbienmeetp.1994.1.193018.

[38] E. Knill, "Quantum Computing with Realistically Noisy Devices." *Nature*, vol. 434, no. 7029, 2005, pp. 39–44., doi:10.1038/nature03350.

[39] T.D. Ladd et al. "Quantum Computers." Nature, vol. 464, no. 7285, 2010, pp. 45–53., doi:10.1038/nature08812.

[40] Peter J. Lewis. "Six Quantum Worlds." Quantum Ontology, 2016, pp. 179–182., doi:10.1093/acprof:oso/9780190469825.003.0009.

[41] Peter J. Lewis. *Quantum Ontology: a Guide to the Metaphysics of Quantum Mechanics*. Oxford University Press, 2016.

[42] D. Malament (2009). "Note on Carnap's 'On the Dependence of the Properties of Space Upon Those of Time'".

[43] J.B. Manchak. "On the Possibility of Supertasks in General Relativity." *Foundations of Physics*, vol. 40, no. 3, 2009, pp. 276–288., doi:10.1007/s10701-009-9390-x.

[44] J.B. Manchak. "Malament–Hogarth Machines." *The British Journal for the Philosophy of Science*, 2018, doi:10.1093/bjps/axy023.

[45] J. Manchak and Bryan W. Roberts. "Supertasks." *Stanford Encyclopedia of Philosophy*, Stanford University, 5 Apr. 2016, plato.stanford.edu/entries/spacetime-supertasks/.

[46] Tim Maudlin "Can the World Be Only Wavefunction?" *Many Worlds?: Everett, Quantum Theory & Reality*. Ed. Simon Saunders, Jonathan Barrett, Adrian Kent, and David Wallace. Oxford: Oxford UP (2012). Print.

[47] J.A. Miszczak "Models of Quantum Computation and Quantum Programming Languages" *Bull. Pol. Acad. Sci. Technol. Sci.* 59(3), 305–324 (2011)

[48] C. Monroe "What Quantum Computers May Tell Us about Quantum Mechanics" *Science and Ultimate Reality* (2010): 345-60. Web.

[49] Vincent C. Müller (2014). "Pancomputationalism: Theory or Metaphor?" In Ruth Hagengruber & Uwe Riss (eds.), *Philosophy, Computing and Information Science*. Pickering & Chattoo. pp. 213-221. Available on PhilArchive: https://philarchive.org/archive/MLLPTO-2

[50] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[51] Bernhard Ömer, "A Procedural Formalism for Quantum Computing", masters thesis in Technical Physics, TU Vienna (1998)

[52] Xavier Oriols and Jordi Mompart, *Applied Bohmian Mechanics* Pan Stanford:Singapore (2012)

[53] M. Ozawa, "Quantum Turing Machines: Local Transition, Preparation, Measurement, and Halting" *Quantum Communication, Computing, and Measurement 2* (1998): 241-48. Web

[54] Gualtiero Piccinini. "Computational Modelling vs. Computational Explanation: Is Everything a Turing Machine, and Does It Matter to the Philosophy of Mind?" *Australasian Journal of Philosophy*, vol. 85, no. 1, 2007, pp. 93–115., doi:10.1080/00048400601176494.

[55] Gualtiero Piccinini, "The Physical Church-Turing Thesis: Modest or Bold?" *The British Journal for the Philosophy of Science*, vol. 62, no. 4, 2011, pp. 733–769., doi:10.1093/bjps/axr016.

[56] Gualtiero Piccinini, "The Modest Physical Church-Turing Thesis." *Physical Computation*, 2015, pp. 263–273., doi:10.1093/acprof:oso/9780199658855.003.0017.

[57] Gualtiero Piccinini, "Computation in Physical Systems." Stanford Encyclopedia of Philosophy, Stanford University, 29 May 2017, plato.stanford.edu/entries/computation-physicalsystems/.

[58] Gualtiero Piccinini, and Neal G. Anderson. "Ontic Pancomputationalism." *Physical Perspectives on Computation, Computational Perspectives on Physics*, by Michael E. Cuffaro and Samuel C. Fletcher, Cambridge University Press, 2018, pp. 23–37.

[59] Gualtiero Piccinini *Physical Computation: a Mechanistic Account*, Oxford University Press, 2018.

[60] Hilary Putnam, *Representation and Reality*. MIT Press, 1988.

[61] Eleanor Rieffel and Wolfgang Polak, *Quantum Computing: A Gentle Introduction* Cambridge, MA: MIT (2011)

[62] Phillipp Roser. "Quantum computing in the de Broglie-Bohm pilot-wave picture." arXiv preprint arXiv:1012.4843 (2010).

[63] Phillipp Roser. "Quantum computation from a de broglie-bohm perspective." arXiv preprint arXiv:1205.2563 (2012).

[64] Saunders, Simon, et al., editors. *Many Worlds?: Everett, Quantum Theory, and Reality*. Oxford University Press, 2012.

[65] Oron Shagrir, and Itamar Pitowsky. "Physical Hypercomputation and the Church–Turing Thesis." *Minds and Machines*, vol. 13, no. 1, 2003, pp. 87–101., doi:10.1023/a:1021365222692.

[66] O. Shagrir and I. Pitowsky, 2003, "Physical Hypercomputation and the Church-Turing Thesis", *Minds and Machines*, 13: 87–10.

[67] Erwin Schrödinger, "Are There Quantum Jumps? Part I" *The British Journal for the Philosophy of Science*, Vol. 3, No. 10 (Aug., 1952), pp. 109-123

[68] Erwin Schrödinger, "Are There Quantum Jumps? Part II" *The British Journal for the Philosophy of Science*, Vol. 3, No. 11 (Nov., 1952), pp. 233-242

[69] John R. Searle, *The Rediscovery of the Mind*. MIT Press, 1992.

[70] Peter W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer" *Lecture Notes in Computer Science Algorithmic Number Theory* (1994)

[71] K.-A. Suominen. "Physical Implementation of Large-Scale Quantum Computation" in *Handbook of Natural Computing*, pp. 1493-1520, Springer Verlag, Heidelberg, Germany, 2012.

[72] R. Tumulka, "A Relativistic Version of the Ghirardi-Rimini-Weber Model" *Journal of Statistical Physics* 125.4 (2006): 821-40.

[73] A. M. Turing, 1936–7, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceeding of the London Mathematical Society*, 42(1): 230–265

[74] A. M. Turing "On Computable Numbers, with an Application to the Entscheidungs Problem". M. Davis (ed.) *The Undecidable* Ewlett, NY: Raven, 1965 pp. 116-154.

[75] Lev Vaidman. "Many-Worlds Interpretation of Quantum Mechanics." Stanford Encyclopedia of Philosophy, Stanford University, 17 Jan. 2014, seop.illc.uva.nl/entries/qm-manyworlds/.

[76] Lieven M. K. Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, and Isaac L. Chuang, "Experimental Realization of Shor's Quantum Factoring Algorithm Using Nuclear Magnetic Resonance." *Nature* 414.6866 (2001): 883-87.

[77] John Von Neumann, *Mathematical Foundations of Quantum Mechanics*. Princeton University Press, 1996.

[78] Martin Ziegler. "Physically-Relativized Church–Turing Hypotheses: Physical Foundations of Computing and Complexity Theory of Computational Physics." *Ap-*

plied Mathematics and Computation, vol. 215, no. 4, 2009, pp. 1431–1447., doi:10.1016/j.amc.2009.04.062.