

University of Massachusetts Boston

ScholarWorks at UMass Boston

Applied Physics Masters Theses Collection

Physics

6-2011

Approach to Typicality in Quantum Systems

Shawn Dubey

University of Massachusetts Boston, shawn.dubey001@umb.edu

Follow this and additional works at: https://scholarworks.umb.edu/physics_theses



Part of the [Atomic, Molecular and Optical Physics Commons](#), and the [Quantum Physics Commons](#)

Recommended Citation

Dubey, Shawn, "Approach to Typicality in Quantum Systems" (2011). *Applied Physics Masters Theses Collection*. 1.

https://scholarworks.umb.edu/physics_theses/1

This Open Access Thesis is brought to you for free and open access by the Physics at ScholarWorks at UMass Boston. It has been accepted for inclusion in Applied Physics Masters Theses Collection by an authorized administrator of ScholarWorks at UMass Boston. For more information, please contact scholarworks@umb.edu.

APPROACH TO TYPICALITY IN QUANTUM SYSTEMS

A Thesis Presented

by

SHAWN DUBEY

Submitted to the Office of Graduate Studies,
University of Massachusetts Boston,
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

June 2011

Applied Physics Program

©2011 by Shawn Dubey
All rights reserved

APPROACH TO TYPICALITY IN QUANTUM SYSTEMS

A Thesis Presented

by

SHAWN DUBEY

Approved as to style and content by:

Stephen Arnason, Associate Professor
Chairperson of Committee

Kurt Jacobs, Assistant Professor
Member

Bala Sundaram, Professor
Member

Stephen Arnason, Program Director
Applied Physics Program

Bala Sundaram, Chairperson
Physics Department

ABSTRACT

APPROACH TO TYPICALITY IN QUANTUM SYSTEMS

August 2011

Shawn Dubey, B.S., Northeastern University
M.S., University of Massachusetts Boston

Directed by Assistant Professor Kurt Jacobs

The study of quantum mechanics has greatly broadened since its inception in the early twentieth century. Recent research has focused on the emergence of thermalization in quantum many-body systems. In this thesis I will demonstrate the approach to typicality—the notion that for specific sets of objects, most of the objects share a common property—in a single, many-body spins chain of spin half particles. This notion of typicality is new. But it serves as a good explanation for the emergence of thermalization.

ACKNOWLEDGMENTS

I must give a great thanks to Kurt Jacobs for allowing and helping me to delve into the world of quantum mechanics. Special thanks to Sai Vinjanampathy for his many extremely helpful talks and advice on this project. Thanks to Vanja Dunjko for his helpful talks. Thanks to Julie Morris for advice on memory debugging in C++, without which, this would have been much harder. And thanks to Melody Wan for drawing figures 2.1 and 3.1.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	viii
CHAPTER	Page
1 INTRODUCTION	1
1.1 Direction	1
1.2 Motivations from Statistical Mechanics	2
1.3 Connecting to Quantum Mechanics	5
2 EIGENSTATE THERMALIZATION HYPOTHESIS AND TYPICALITY .	11
2.1 Typicality	11
2.2 Eigenstate Thermalization Hypothesis	15
3 NUMERICAL RESULTS FOR TYPICALITY	21
3.1 Classifying Systems and Results	21
3.2 Hamiltonian Construction	25
3.3 Results and Conclusion	28
3.4 Technical Information	31
3.5 Suggestions for Further Research	31
APPENDIX	
A C++ CODE	32
CITATIONS	58
LIST OF REFERENCES	59

LIST OF FIGURES

Figure	Page
2.1 Chain of qubits with nearest-neighbor interaction.	18
3.1 Example of Numerical Method	24
3.2 RMS Deviation-odd qubit chain	29
3.3 log-log RMS Deviation-odd qubit chain	30
3.4 log-log RMS Deviation-reduced odd qubit chain	30

LIST OF TABLES

Table	Page
3.1 Even Qubit Spin Chains	28
3.2 Odd Qubit Spin Chains	28

CHAPTER 1

INTRODUCTION

1.1 Direction

This presentation will be focused primarily on matters of statistical mechanics. We hope to understand the mechanisms and conditions for thermalization in quantum systems. We examine non-integrable Hamiltonians. Our discussion focuses on a concept known as typicality: the idea that almost all elements in a set have a common property. The concept of eigenvector typicality is explored and is used to help understand the idea of Eigenstate Thermalization Hypothesis (ETH)

In the first section of the first chapter, classical statistical mechanics is discussed. Several important topics in statistical mechanics are touched upon: phase space, Newton's Equations, Liouville's Theorem, temperature, ergodicity and other concepts. They are used to help reach an understanding of an ensemble and distribution which describe systems in thermal equilibrium. They are the Boltzmann Distribution and microcanonical ensemble.

The second part of the first chapter is dedicated to recasting the classical statistical mechanics into the language and paradigm of quantum mechanics and realizing the form of the Boltzmann Distribution. These are what we want to recover after we postulate ETH.

The second chapter elaborates on the idea and background of Eigenstate Thermalization Hypothesis and how it can be understood through the lens of typicality. We build up from the work of previous researchers. We show we can approximate pure states of the

universe as the appropriate thermal state. The concepts of the previous chapter are invoked to understand typicality and ETH.

The third chapter focuses on the numerical results and how they support Eigenstate Thermalization Hypothesis and the approach to typicality in quantum systems.

1.2 Motivations from Statistical Mechanics

It is our intention here to first motivate the results in chapters two and three with concepts from classical statistical mechanics. We begin this here by first examining a simple example and continue with this in section 1.2. Then we connect to quantum mechanics in section 1.3.

Consider a system, a box for example, which is isolated from the larger universe. It is a closed system. Also consider an ensemble of particles contained within this box (the box has rigid walls, and completely elastic collisions). The box is partitioned and the particles are contained in one half of the box initially (with a fixed volume).

Given some time after the partition is removed, the system will go to an equilibrium (macro)state. It is unlikely that one would be able to measure the trajectories and properties of each particle and gain information from them. It is favorable to employ statistical mechanics to solve this problem.

The particles will fill the box, with equal density everywhere. This equilibrium state cannot be reversed, in other words the approach to equilibrium is an irreversible process. This is a process which happens spontaneously and by definition is not allowed to go in reverse [1]. For specific processes, such as the approach to equilibrium, the entropy maximum is seen at equilibrium. This is an example of thermodynamic irreversibility [1].

The standard procedure in studying dynamical systems is to determine the positions and momenta of the particle (i.e. phase space) or rather the trajectories in phase space of the

particles. Both position (q , a generalized position) and momenta (p) are $3N$ dimensional, where N is the number of degrees of freedom.

Solving Newton's Equation's

$$\dot{p} = F(q) \quad (1.1)$$

$$\dot{q} = \frac{p}{m} \quad (1.2)$$

would also be too complex. Given the number of particles, it would simply be too difficult. As such we would like to be able to get information from the particles en masse.

In our classical systems, chaotic behavior will remove any knowledge we may have had about the initial conditions in the first place. The only thing we then will know about the system are the constants of motion, in this case the energy. Let us assume that energy is the only quantity required to describe the system at equilibrium. If we follow this assumption, we arrive at the familiar microcanonical ensemble in statistical mechanics [2].

We want to use this knowledge to obtain the microcanonical average.

First, we define the function $\Omega(E)$ as the phase space volume of an energy "shell" [$E, E+\delta E$], divided by δE :

$$\Omega(E)\delta E = \int_{E < H(p,q) < E+\delta E} dpdq \quad (1.3)$$

We average over states in the shell and take the limit as δE goes to zero:

$$\langle A \rangle_E = \frac{1}{\Omega(E)\delta E} \int_{E < H(p,q) < E+\delta E} A dpdq \quad (1.4)$$

What has been done was to average over states in phase space for a restricted energy shell on an energy surface. In doing this we have implicitly assumed that all states are equally likely [2].

Next we note systems of point particles obeying the Newton's Laws (excluding dissi-

pation) are to be considered examples of Hamiltonian dynamical systems (these systems are energy conserving) [2]. We can then recover Newton's Laws of Motion providing we invoke Hamilton's equations:

$$\dot{p} = -\frac{\partial H}{\partial q} \quad (1.5)$$

$$\dot{q} = \frac{\partial H}{\partial p} \quad (1.6)$$

In our quest to understand equilibrium statistical mechanics it is instructive to introduce Liouville's Theorem. This states that a probability density, P , in phase space is incompressible, i.e. the probability density is conserved [2]. This probability density may be deformed in phase space but it always retains its original volume. Mathematically this is (from the continuity equation for a probability current in phase-space)

$$\frac{dP}{dt} = \frac{\partial P}{\partial t} + \sum_{i=1}^{3N} \frac{\partial P}{\partial q^i} \dot{q}^i + \frac{\partial P}{\partial p^i} \dot{p}^i + P \frac{\dot{q}^i}{q^i} + P \frac{\dot{p}^i}{p^i} = 0 \quad (1.7)$$

the total time derivative carries the phase space volume to itself at all times (phase space volume is a constant). Inserting Hamilton's equations into this continuity equation we get the final form of Liouville's Theorem [2]:

$$\frac{dP}{dt} = \frac{\partial P}{\partial t} + \sum_{i=1}^{3N} \frac{\partial P}{\partial q^i} \dot{q}^i + \frac{\partial P}{\partial p^i} \dot{p}^i = 0 \quad (1.8)$$

As a consequence of this, (for Hamiltonian systems) there will be no preferred states to which any system will settle. If a system is momentarily in one state it will quickly leave [2]. This is equivalent to equilibrium in statistical mechanics. That is, in the assumption of all states of the system being equal, it is implied that the system spends no more time in one state than any other, and if it does, its time there is fleeting as time evolution

will carry it from that temporarily preferred state.

To proceed further we define a new term, ergodic, whose definition we adapt from Boltzmann's original: a system in which the path of every point in phase space passes arbitrarily close to every other point on a constant energy surface [2].

Another consequence of Liouville's Theorem in Hamiltonian systems is the time-independence of the microcanonical ensemble. Since energy is conserved, a uniform density in phase space is bound to stay uniform throughout time, on a path, for a small range of energies [2]. Invoking ergodicity, we can then say that the density will be uniform throughout the energy surface.

Following from these is the idea that time averages are equal to microcanonical averages. Since the microcanonical ensemble is time independent the time average is equal to the ensemble average. This implies-since the time average is constant-for ergodic systems, the ensemble average is the time average at all places [2].

Now that we have a background in classical statistical mechanics, it is now possible to move on and understand the quantum mechanics needed to analyze our systems. These concepts will be important in interpreting our results in the coming chapters.

For a more in depth discussion of the preceding section, consult Sethna [2].

1.3 Connecting to Quantum Mechanics

Let us now try to understand a significant part of the previous section in the language of quantum mechanics and quantum states, as we will be using it to describe our quantum systems later in this exposition.

Suppose one had a system of particles and the only information available to any experimenter are properties of this system, e.g. total energy, number of particles, volume, pressure, et cetera. States of the system described by these properties are known as macrostates.

As an simpler, more concrete example, suppose one had a basket of apples; some are peeled some are not. An example of the macrostate of this apple-system would be knowing that there are five peeled and five unpeeled apples.

Each macrostate has associated with it a number of microstates. From the apple-system above, knowing the microstates would mean knowing the state of each apple. If we knew specifically which apples were peeled and which were not, we would know the microstate of each apple. In quantum mechanics microstates are described by mutually orthogonal quantum states [1]. These are the states in which a system may be in, given a particular macrostate. As is always the case in quantum mechanics, one works in Hilbert Space. The dimensionality of Hilbert Space for the systems considered here is equal to the number of microstates, given a particular macrostate and having all of the properties fixed (for a many body system this Hilbert Space is large and consists of many microstates). This can be an extremely large number and as such any person doing measurements on the system does not have access to the individual microstates, only the macrostates.

We now make a large and important leap in our understanding of statistical mechanics. Using the definition of microstates we come to the fundamental postulate of statistical mechanics, that of assigned equal a priori probability [3]. From Jacobs, this states: "all the accessible microstates of a closed system are equally likely." [1]. This can be interpreted in two ways: the first being the system will spend as much time in one microstate as any other, regardless of which microstate in which the system was initially. This means (similar to the previous discussion) that the system is usually in one of the "typical" states [1] i.e. subsets of states which contain most of the microstates. This can also be interpreted as the assumption of an equal mixture of microstates which will give rise to specific properties of the system [1].

The question quickly arises: how does this help one understand the approach to equilibrium, or thermalization? To see, let us briefly consider the familiar concepts of entropy

and temperature.

The microcanonical ensemble dictates that given an energy E , the equilibrium behavior of a system is an equal weighting of all possible states of that system. Let us define the number of microstates of a system Ω , as being those states having equal probability (this is the fundamental postulate of statistical mechanics), of a system of fixed energy. It is more convenient to work with the logarithm of the number of microstates, Ω , though:

$$S_{th} = k_B \ln(\Omega) \quad (1.9)$$

This is the familiar Boltzmann entropy of a system, with k_B being the Boltzmann constant. It is the entropy of a system in equilibrium. We will refer to it as the equilibrium entropy, it is an extensive quantity, i.e. it scales with the size of the system. In thermodynamics this can be thought of a measure of "disorder" or in a more information-theoretic sense, the uncertainty in the knowledge that an experimenter has about the system [1]. This entropy can be thought of as the number of possible microstates in which a given macrostate can be, given the state of knowledge of the system. In these senses we can identify the thermodynamic entropy with the von Neumann entropy (characterizing our state of knowledge of the system) [1]

$$S_{th} = S_v \quad (1.10)$$

We can extend the definition of thermodynamic entropy to microstates which are not equally likely. This does not contradict the fundamental postulate of thermodynamics as that applies to closed, isolated systems. If there are N systems interacting but they are isolated from the rest of the universe, the microstates of the total, combined system are still equally likely; those of the subsystems are not [1]. Then we want a thermodynamic entropy for a system that has different probabilities for being in different states. Suppose

we have N systems which are in an arbitrary mixture of several orthogonal states. We can still find out the total number of microstates of the large combined system, consisting of the N subsystems (this large system's state is thus a mixture of all of the orthogonal states of the subsystems and each state has its own probability of occurring) [1]. As the number of subsystems increases, the probability of the large system to be in anyone one of the number of typical states (states within the set that contains most of the microstates) increases while the probability of being in any atypical state goes to zero [1].

As the number of systems grow, these typical states become more likely and as the number of particles N goes to infinity all become equally likely.

The entropy of this large, combined system is the sum of the entropies of the smaller subsystems. These entropies are all weighted by the number of typical states in our subsystems:

$$S = \sum_{n=1} p_n \ln(p_n) \quad (1.11)$$

where p_n is the probability of being in a particular state.

With this, we define the temperature, at constant volume and number, via the following relation:

$$\frac{1}{T} = \frac{\partial S}{\partial E} \quad (1.12)$$

This inverse temperature describes the rate and direction of heat flow between systems [2]. When systems reach thermal equilibrium there is virtually no heat flow between them and their entropy must by definition, be at a maximum.

This is easy to describe, as the system has evolved to a particular macrostate in which it will spend most of its time. This will be the macrostate containing most of the microstates (the typical states). As the entropy is the natural logarithm of the number of microstates, this is then the maximum entropy macrostate, i.e. the state in which the system is in thermal

equilibrium.

The temperature then, can be defined as:

$$T = \frac{\partial E}{\partial S} \quad (1.13)$$

at constant volume, because in changing volume, work is being done and energy is changing irrespective of the entropy [2]. This temperature is fixed by the external environment of the system, the heat bath. One can then show that if a system is coupled to a large bath, the equilibrium behavior of the bath is dependent only on the temperature of the bath [3].

If we now take a system and put it in contact with a large thermal bath (the total system-bath being known as the universe, with total energy E_t) at temperature T , we will obtain a distribution from which all properties of systems in equilibrium can be derived. This distribution is known as the Boltzmann Distribution.

Briefly sketching the derivation: we assume all quantum microstates of the universe (with associated energy E), are equally likely and then pose the question: what is the probability of finding a system microstate $|s\rangle$ (with associated energy E_s) [1]? This will be proportional to the amount of bath states that have energy $E_t - E_s$, which is the energy E of the bath states we wish to find [1]. This boils the quest down to figuring out the number of such states n_s . Given our assumption that the temperature T of the bath is fixed, we can also declare that $\frac{\partial S}{\partial E}$ is fixed [1]. The entropy will be linear in energy and thus the number of states will be exponential in E [1]:

$$S = S_o + \frac{\partial S}{\partial E} E \quad (1.14)$$

We also recall that the entropy is the natural logarithm of the number of microstates.

Equating the two (when the bath has energy E), we have:

$$-K_B \ln(n_s) = S_o + \frac{\partial S}{\partial E} E \quad (1.15)$$

$$-K_B \ln(n_s) = S_o + \frac{E}{T} \quad (1.16)$$

$$\ln(n_s) = -\frac{S_o}{K_B} - \frac{E}{K_B T} \quad (1.17)$$

$$n_s = e^{\frac{-S_o}{K_B} + \frac{-E}{K_B T}} \quad (1.18)$$

$$n_s = n_o e^{\frac{-E}{K_B T}} \quad (1.19)$$

We now realize the form of the Boltzmann Distribution. This Boltzmann distribution is known as the canonical ensemble, as we have assumed energy exchange between the system and bath [2]. One may still derive the Boltzmann Distribution from the microcanonical regime if one considers a large system comprised of numerous subsystems. The large system functions as a bath for the subsystems [1] and this large system was in the microcanonical ensemble i.e. the equal a priori probability scheme.

CHAPTER 2

EIGENSTATE THERMALIZATION HYPOTHESIS AND TYPICALITY

2.1 Typicality

With a thorough background in classical statistical mechanics and quantum mechanics, we are now in a position to rethink notions of equilibrium and study them in quantum systems. It is now possible to quickly discuss ergodicity and typicality and how they relate to thermalization.

Chronologically, von Neumann's ergodic theorem for macroscopic systems was a first step forward in statistical mechanics in quantum systems. But it does not hold for all observables in a closed system [4]. To acquire a more general ergodic theorem, it is suggested that random perturbations should be induced to accomplish this [4]. This is why the need to couple to a bath was introduced. This brings the system to equilibrium and brings the system to the bath temperature [4].

However, there is another approach [5] which brings us to the result we will call "Eigenstate Thermalization Hypothesis" (ETH) [6]. In this approach, every eigenstate of an appropriate system (usually non-integrable systems) gives a thermal state, meaning each eigenstate gives expectation values in agreement with microcanonical ensemble. Stated differently, each eigenstate of the Hamiltonian reproduces the microcanonical average of an observable [7]. This reproduces results from statistical mechanics in closed systems. It is now irrelevant whether or not we are decomposing into a system-bath scheme (but will be helpful in explaining typicality).

A way to understand ETH is through typicality. The notion of typicality in general simply means that nearly all members of a large set share a common property. We look at eigenvectors of a Hamiltonian which share a common property: as we increase the dimension of the examined sectors (particular subspaces), they put the system into a state with the correct population of up-spin states as predicted by statistical mechanics.

To clarify, a sector is just a particular subspace of a system. As an example, recall our apple-basket system. If we now have a basket which can hold ten pieces of fruit, we place in it five apples and five oranges. The number of ways we can arrange the apples and oranges in the basket constitutes one sector. We could also have seven apples and three oranges and the number of ways to arrange those in the basket is another sector. For the purposes of this paper, we deal with sectors of chains of qubits/spins, in which there are certain number of them in the up state.

With all of this knowledge in mind, we need to have a better understanding of foundations of statistical mechanics. Here we emphasize the work of Popescu et al [3]. This work shows we can replace the fundamental postulate of statistical mechanics: assume equal probabilities of the pure states of the universe a priori, by a principle that focuses on individual states [3]. This will be the first rigorous introduction to typicality.

2.1.1 A First Mathematical Justification for Typicality

Consider a system and a bath, where the dimension of the bath larger is than that of the system and with dimensionalities D_s and D_b respectively. The state of the system-bath universe is constrained by a global constraint C [3] (which would be the fixed total energy of the combined system-bath universe in statistical mechanics) which can be seen in quantum mechanics by:

$$H_C \subseteq H_s \otimes H_b \tag{2.1}$$

where H_C is a subspace of the full Hilbert space $H_s \otimes H_b$.

We define the universe when it is in a maximally mixed state as [3]:

$$\phi_C = \frac{1}{D_C} \mathbf{I}_C \quad (2.2)$$

where \mathbf{I}_C is the identity matrix and D_C is the dimensionality of H_C . The maximally mixed state is by definition proportional to the identity.

The quantum state of the system when the universe is in ϕ_C , is (termed a "canonical state")

$$\Delta_s = Tr_b \phi_C \quad (2.3)$$

Then assume the universe to be in a pure state $|\psi\rangle$ and then show that the reduced state of the system:

$$\Psi_s = Tr_b |\psi\rangle \langle \psi| \quad (2.4)$$

is extremely close to Δ_s for most of the Δ_s [3], i.e.:

$$\Psi_s \approx \Delta_s \quad (2.5)$$

Restated this means that for most $|\psi\rangle \in H_C$ the system in consideration acts as if it were in an equiprobable state, ϕ_C [3]. This statement is redefined in [3] "Given a sufficiently small subsystem of the universe, almost every pure state of the universe is such that the subsystem is approximately the canonical state Δ_s ". This can be extended to thermal systems [3]: "Given that the total energy of the universe is approximately E, interactions between the system and the rest of the universe are weak, and that the density of states of the environment increases approximately exponentially with energy, almost every pure state of the universe is such that the state of the system alone is approximately equal to the

thermal canonical state $e^{\frac{-H_s}{k_b T}}$, with temperature T (corresponding to energy E) [3]. This assumes that the Hamiltonian for the entire universe is:

$$H_u = H_s + H_b + H_{int} \quad (2.6)$$

where H_{int} is the Hamiltonian for the interaction between the system and bath.

Now the idea is to create some distance measure in which they show that the distance between Δ_s and Ψ_s , which is how easy it is to tell Δ_s and Ψ_s apart [3]. The main theorem is that the volume V, of states in the constrained subspace, which are far from the "canonical" state, decreases exponentially with the dimension of the constrained subspace. This is stated this mathematically as [3]:

$$\frac{V[(|\psi\rangle \in H_C | d(\Psi_s(\psi), \Delta_s) \geq \delta]}{V[(|\psi\rangle \in H_C)]} \leq \delta' \quad (2.7)$$

where

$$d(\Psi_s(\psi), \Delta_s) = \frac{1}{2} \sqrt{(\Psi_s - \Delta_s)^\dagger (\Psi_s - \Delta_s)} \quad (2.8)$$

$$\delta = \epsilon + \frac{1}{2} \sqrt{\frac{D_s}{D_b^{eff}}} \quad (2.9)$$

and

$$\delta' = 4e^{-K D_C \epsilon^2} \quad (2.10)$$

for $\epsilon > 0$, where K is a positive constant and

$$D_b^{eff} = \frac{1}{Tr \Delta_b^2} \geq \frac{D_C}{D_s} \quad (2.11)$$

To prove this Levy's Lemma is invoked [3]. Levy's Lemma is a lemma from high-dimensional geometry which states that for most points P, on a hypersphere of dimension

$D \gg 1$ and area $A[(P)]$ and all functions f which are not quickly changing, $|\nabla f| \leq 1$, $f(P)$ is approximately equal to the mean value $\langle f \rangle$. Specifically:

$$\frac{A[(Pf(P) - \langle f \rangle \geq \epsilon)]}{A[(P)]} \leq 4e^{-(\frac{1}{9\pi^3})(D+1)\epsilon^2} \quad (2.12)$$

This clarifies, mathematically, our notion of (canonoical) typicality, in which almost all states of the universe are approximately equal to the thermal state, by considering the universe in a system-bath setup. This allows one to recover the equilibrium results in statistical mechanics. We specifically look at Hamiltonian typicality, in which the eigenstates of the most Hamiltonian are states which put subsystems in thermal equilibrium.

2.2 Eigenstate Thermalization Hypothesis

We now seek to clarify the concept of eigenstate thermalization [5, 6, 7], which need not invoke the system-bath setup. Let us consider eigenstates of a Hamiltonian \hat{H} . The eigenstates are:

$$|\Psi(t)\rangle = e^{-i\hat{H}t} |\Psi(0)\rangle = \sum_i C_i e^{-iE_i t} |\phi_i\rangle \quad (2.13)$$

where E_i are the eigenstate energies and $C_i = \langle \Psi_i | \Psi(0) \rangle$.

For an observable, represented by an operator in quantum mechanics, \hat{O} , the mean is given by:

$$\langle \hat{O}(t) \rangle \equiv \langle \Psi(t) | \hat{O} | \Psi(t) \rangle = \sum_{ij} C_i^* C_j e^{i(E_i - E_j)t} O_{ij} \quad (2.14)$$

In the long-time limit, the mean is:

$$\overline{\langle \hat{O}(t) \rangle} = \lim_{t \rightarrow +\infty} \frac{1}{t} \int_0^t \langle \Psi(t) | \hat{O} | \Psi(t) \rangle dt = \sum_i |C_i|^2 O_{ii} \quad (2.15)$$

This has been described as the "diagonal ensemble" which is identified as exactly the gen-

eralized canonical ensemble, provided that the constants of motion are the projectors:

$$\hat{P}_i = |\Psi_i\rangle\langle\Psi_i| \quad [7].$$

If a system equilibrates, it will be to this average. When settling to a steady-state, the time-dependent off-diagonal terms in our average goes to zero in time because of the dephasing effects [7].

If a system equilibrates at all it should also equilibrate to the appropriate ensemble average. This statement simply means the previous equation should equal the microcanonical average [5]:

$$\langle O(t) \rangle_t = \sum_i |C_i|^2 \langle \hat{O} \rangle_{E_i} \quad (2.16)$$

i.e. a weighted sum of the averages, where the weights are the probabilities of being in any macroscopic state. We state this as:

$$\sum_i |C_i|^2 O_{ii} = \frac{1}{N_{E_o, \Delta E}} \sum_i O_{ii} \quad (2.17)$$

with the condition, $|E_o - E_i| < \Delta E$ [7]. N is a normalization factor; it is the number of states in the energy window $[E_o - \Delta E, E_o + \Delta E]$ where E_o is the average energy of an initial state and ΔE is the half width of the energy window [7]. This is analogous to the idea presented in chapter one's discussion of microcanonical averages. There we averaged over states in phase space given an energy shell. Here we choose an energy range such that the range contains a non-zero number of eigenstates [7].

There exists an interesting interpretation to this analysis, especially equation 2.17. It can be interpreted to mean that the O_{ii} ("eigenvalue expectation values") [7] do not change (or change only slightly) between eigenstates [7]. From the previous analysis then, we have shown that every state in which we begin the system, we will always recover the

microcanonical average.

Recall the earlier discussion on classical statistical mechanics. One of the goals was to use the idea of ergodicity to arrive at the microcanonical average. The same idea has been done in this section, though in a less obvious and not fully understood way. We have recovered the notion of ergodicity in quantum mechanics in that each eigenstate gives back the correct microcanonical averages.

It is conjectured that for initial states in a narrow energy window O_{ii} , do not fluctuate between eigenstates close in their energies [7]. In this case equation 2.5 holds.

But what of time dynamics? It is revealed that they play merely an ancillary role [7]. As stated before, there is initially coherence between eigenstates but time evolution destroys it through dephasing revealing thermalization [7]. We see that in quantum mechanical systems every eigenstate gives a thermal state for the system. This is in contrast to the classical idea of ergodic motion in phase space through time evolution, producing results given by microcanonical predictions (see discussion in chapter one). Rigol et al provide more information on this [7].

To elaborate, we start with a bare Hamiltonian, $H = H_o$, which describes our unperturbed quantum mechanical system. To this we add our random Hamiltonian, H_r , so we can recover statistical mechanics. H_r is, for the time being, a Gaussian random matrix, particularly one whose elements are drawn from a distribution with zero mean and variance one, the Gaussian Orthogonal Ensemble (GOE). This has been well motivated since the time when Wigner postulated that the energy spectrum of nuclei is well described by random matrices and reproduces the Wigner-Dyson distribution. These random matrices (the Hamiltonians) are real-symmetric matrices [8]. We will again use real-symmetric matrices in our analysis to provide results consistent with statistical mechanics as Wigner did.

The full Hamiltonian is then:

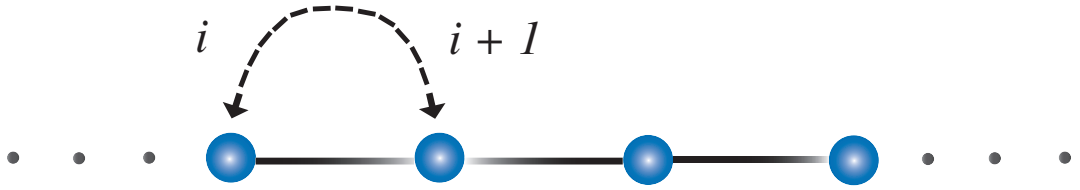


Figure 2.1: Chain of qubits with nearest-neighbor interaction.

$$H = H_o + H_r \quad (2.18)$$

We can also expect similar results when H_r is replaced by a two-body, nearest-neighbor interaction Hamiltonian [5], H_i . The interaction examined in this paper is between nearest-neighbor qubits (fig. 2.1). The Hamiltonian is then:

$$H = H_o + H_i \quad (2.19)$$

We shall see this demonstrated in the results section. The Hamiltonians discussed are real-symmetric matrices.

Ergodicity is given by the eigenvectors. We induce small but non-negligible interaction. In the limit of arbitrarily small interactions, it is as if there is no system-bath coupling and thus no thermal behavior. The eigenvectors mix with random phases and we note two things: the distribution of eigenvectors of the Hamiltonian in the basis where the non-interacting Hamiltonian is diagonal is exponential in the inverse of the interaction strength and that deviations from microcanonical predictions decrease exponentially with number of degrees of freedom due to the small, non-negligible interactions [5].

2.2.1 Example of ETH Applicability

As a brief aside, let us take note of other classes of systems that thermalize, separate from the ones we just described, so as to get an example of how ETH applies.

It has been shown that any quantum mechanical system whose classical counterpart is chaotic will exhibit eigenstate thermalization [7]. More interestingly, systems which follow from Berry's conjecture thermalize in the semiclassical limit [6].

Berry's conjecture states that locally, eigenfunctions of chaotic system behave like a random superposition of plane waves [6]. This condition is (in the example of a box of hard spheres confined to a particular region of the box, with $3N$ momentum and position: \mathbf{P}, \mathbf{X}):

$$\Psi_i(\mathbf{X}) = K_i \int d^{3N} P A_i(\mathbf{P}) \delta(\mathbf{P}^2 - 2mU_i) e^{i\frac{\mathbf{P}\cdot\mathbf{X}}{\hbar}} \quad (2.20)$$

where K_i is determined through normalization:

$$\int d^{3N} X \Psi_i(\mathbf{X}) = 1 \quad (2.21)$$

In the example, the amplitudes $A(\mathbf{P})$, with the property

$$A^*(\mathbf{P}) = A(-\mathbf{P}) \quad (2.22)$$

are Gaussian random and this condition equates to Berry's conjecture. For his system, it is possible to recover thermal predictions and eigenstate thermalization [6].

Eigenstate thermalization is not solely confined to our Hamiltonian system with random perturbations. Indeed, it can be shown to arise for many other classes of systems like the one presented.

2.2.2 Focus of Results

The class of Hamiltonians for which we want to demonstrate eigenstate thermalization are non-integrable hamiltonians. Here integrability means a system which is exactly solvable. It is typically thought that quantum integrable systems do not exhibit thermalization. This is essentially correct. We see though, that thermalization is induced in integrable systems which are weakly perturbed, which is the reason for the introduction of Hamiltonians with Gaussian random perturbations, or two-body, nearest-neighbor interactions acting as perturbations.

To study non-integrable Hamiltonians we shall invoke the use of chains of quantum bits (qubits) and their spin states. Qubits are highly relevant in quantum information theory and their properties allow one to understand some basic concepts as well as perform numerical calculations easier.

CHAPTER 3

NUMERICAL RESULTS FOR TYPICALITY

3.1 Classifying Systems and Results

Now we discuss the background and results of our numerical experiments, demonstrating the notion of eigenvector typicality. This is the property of typicality we wish our Hamiltonians to exhibit; that each of the eigenvectors of the Hamiltonian places a qubit in a particular spin state (up) with the correct frequency. For example, we want to realize the probability of 0.5 for even number qubit spin-chains with half of the qubits in the up state. And we check for a calculated probability (ratio) for odd number qubit spin chains (given below).

Put another way, our main objective is to check whether, if we have a spin chain of four qubits with two in the up state, does this occur with a probability of 0.5 or does the deviation from 0.5 decrease with increasing sector dimensionality? We gather this information from the eigenvectors of a sector Hamiltonian.

We show from numerical simulations that the root mean square (rms) deviation from predicted values decreases as a power-law for odd numbered spin chains (as the dimensionality of the sector increases). For even numbered spin chains there is no deviation. This means the eigenvectors are perfectly typical, due to some symmetry (though it is not clear exactly what symmetry). This implies eigenstate thermalization.

The analysis will show the deviation from the thermal state scales as an inverse power with the dimension of the sector. A sector is merely the subspace of the full system, whose

elements are all the permutations of the spins with a particular number of spins in the up state.

Again, for our purposes we chose to examine a chain of spin- $\frac{1}{2}$ (qubits) particles with open boundary conditions. Our model is an Ising model, where the usual form is:

$$H = \sum_i \sigma_z^i \sigma_z^{i+1} \quad (3.1)$$

where σ_z is the usual Pauli z-spin operator. This is the same as our earlier H_o , the unperturbed Hamiltonian. To model the weak perturbations we choose a nearest-neighbor interaction:

$$H = J_z \sum_i \sigma_z^i \sigma_z^{i+1} + J_x \sum_i \sigma_x^i \sigma_x^{i+1} \quad (3.2)$$

where σ_x is the usual Pauli x-spin operator and J_z and J_x are constants which we have defined as $J_z = \cos^2\theta$ and $J_x = \sin^2\theta$ with $\theta = 0.375\pi$

We want to analyze our model in some way. We examine eigenvector typicality, as stated previously. This examines the probability of our spin chain being in a particular configuration.

We characterize the deviation δ , as the rms deviation in probability from what one would expect for a given configuration of qubits in the spin-chain. Explicitly the expected probability, or ratio, is:

$$ratio = \frac{\binom{q-1}{j}}{\binom{q-1}{j} + \binom{q-1}{q-j}} \quad (3.3)$$

where q is the number of qubits in the spin-chain and j is the number of qubits in the up-state. Equation 3.1 takes into account that in the C++ code the qubits are being bit-flipped starting at the end of the spin-chain.

In short, we have our real-symmetric Hamiltonian which we diagonalize and obtain a unitary matrix U. U diagonalizes the Hamiltonian and is not an evolution operator. The

columns are the eigenvectors, the elements of which are the amplitudes of the states of the sector. We look at the probability that each qubit is in the "up" state by looking at the basis states of an eigenvector, which are the possible configurations of the system in a sector. Looking at the first qubit in each sector element-a configuration of the spin-chain-we look to see if the qubit is in the up state, if it is, we sum its probability (the columns in our unitary matrix are the coefficients/amplitudes of the basis states, therefore their squares are the probability). This gives us the total probability of the first qubit is in the up state. We compare it to the previously described ratio to make sure the qubit is in the correct population (e.g. for half of the qubit in the up state, so we see how far the sum deviates from the ratio of 0.5, the ideal case). Numerically, this can be described as:

$$\delta = \sqrt{\frac{\sum_i \sum_j [\langle T_i | S_j \rangle - ratio]^2}{q * \binom{q}{j}}} \quad (3.4)$$

or

$$\delta = \sqrt{\frac{\sum_i \sum_j [\langle T_i | S_j \rangle - \frac{j}{q}]^2}{q * \binom{q}{j}}} \quad (3.5)$$

where T_i are the vectors whose elements are the square moduli of the elements of the eigenvectors (columns of U), S_j is the vector whose i-th element is the state of the qubit (up-denoted numerically as 1, down-denoted numerically as 0) in all the i-th basis state, q is the number of qubits in the spin chain and $\binom{q}{j}$ is the number of eigenvectors. Numerically, the S_j pick out the up-state of the i-th qubit in all of the sector elements and T_i gives their probabilities:

$$\begin{array}{c}
\mathbf{S}_j \\
\left[\begin{array}{cccc}
0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1
\end{array} \right] \\
\textit{Elements of a Sector}
\end{array}
\qquad
\begin{array}{c}
\mathbf{T}_j \\
\left[\begin{array}{c}
\mathbf{U}_{11}^2 \\
\vdots \\
\mathbf{U}_{61}^2
\end{array} \right]
\end{array}
\qquad
\mathbf{U}_{66}^2$$

Figure 3.1: Example of Numerical Method

To put it more simply we want to see if a particular spin-chain configuration happens with a certain probability (the ratio, described above). We do this by seeing how the probability of each qubit in the spin-chain being in the up-state, deviates from the expected probability (ratio). For each eigenvector, we go through each qubit, we get a final rms deviation in probability. The rms deviations getting smaller implies that each sector happens with a probability closer that predicted by statistical mechanics.

Since it is the eigenvectors of the sector Hamiltonian that give us the probability (through T_i , above), we expect virtually all eigenvectors of a given sector Hamiltonian to give a deviation consistent with statistical mechanics, i.e. goes to zero to be consistent with thermal predictions. This is the concept of eigenvector typicality, described through the rms deviation, δ and implies eigenstate thermalization.

This procedure is done for even qubit chains and odd qubit chains. It was of particular interest to monitor the rms deviations for when half of the qubits are in the up state; this

would be the maximally mixed state and maximum entropy state. The dimension of these "half-up" sectors are the largest dimensions of any sector because of the combinatorics of the system. Again, as the dimension of the sector gets larger, the deviations discussed earlier should approach zero, giving no deviation from the ideal case.

For odd-numbered chains, we fix as "half" $\text{floor}(\frac{n}{2})$, where n is the number of qubits in the spin chain and $\text{floor}()$ denotes the largest integer less than the argument. For example if there are 7 qubits in the spin chain, then the number of them in the up state is $\text{floor}(\frac{7}{2}) = \text{floor}(3.5) = 3$ qubits in the up state.

3.2 Hamiltonian Construction

We now explain how a Hamiltonian for a given sector is constructed numerically. Calculations using the full Hamiltonian are possible. But as the spin chain grows, the dimensionality of the full Hamiltonian grows as a power of two. For sufficiently large spin chains, the computing resources needed for diagonalizing the full Hamiltonian and computing become too large. This why it is important to construct the sector Hamiltonians in a way that is less computationally intensive.

As a specific example lets first define the up and down states in the spin- $\frac{1}{2}$ system:

$$up = |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (3.6)$$

$$down = |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (3.7)$$

For computational reasons, it is desirable to work with the Hamiltonians for each individual sector, where a sector is a particular configuration of the system. An example would be a four-qubit spin chain with two of the qubits in the up state. All possible permutations

of the spin chain constitutes a sector. When we diagonalize the Hamiltonians for the sectors, we recover our eigenvectors and calculate our rms deviation, described previously. We do this procedure each time a qubit is added, again noting the sector of interest is the half-spin-up s , maximally-mixed sector.

One could, in principle, find the full Hamiltonian and extract the Hamiltonians for the individual sectors, however that becomes impractical as the dimensionality of the full Hamiltonian goes as 2^n where n is the number of qubits in the spin chain.

It is possible to construct the Hamiltonians for the sectors based only on the knowledge of how the Pauli gates operate on our $|1\rangle$, $|0\rangle$ states. Explicitly they are:

$$\sigma_x |1\rangle = |0\rangle \tag{3.8}$$

$$\sigma_x |0\rangle = -|1\rangle \tag{3.9}$$

$$\sigma_y |1\rangle = i |0\rangle \tag{3.10}$$

$$\sigma_y |0\rangle = -i |1\rangle \tag{3.11}$$

$$\sigma_z |1\rangle = -|1\rangle \tag{3.12}$$

$$\sigma_z |0\rangle = |0\rangle \tag{3.13}$$

where i is the imaginary unit.

Take as an example the Hamiltonian: $H = J_z \sum_i Z_i Z_{i+1}$ and we examine a four-qubit spin chain with two qubits in the up state; the elements of the sector are 0011, 0101, 0110, 1010, 1100, 1001, where 1 represents the up state and 0 the down state.

To construct an element of the Hamiltonian for this sector we use the 0011 state (in

bra-ket notation):

$$\begin{aligned}
& \langle 0| \langle 0| \langle 1| \langle 1| Z \otimes Z \otimes I \otimes I |0\rangle |0\rangle |1\rangle |1\rangle + \\
& \langle 0| \langle 0| \langle 1| \langle 1| I \otimes Z \otimes Z \otimes I |0\rangle |0\rangle |1\rangle |1\rangle + \\
& \langle 0| \langle 0| \langle 1| \langle 1| I \otimes I \otimes Z \otimes Z |0\rangle |0\rangle |1\rangle |1\rangle \\
& = \\
& \langle 0|Z|0\rangle \langle 0|Z|0\rangle \langle 1|I|1\rangle \langle 1|I|1\rangle + \\
& \langle 0|I|0\rangle \langle 0|Z|0\rangle \langle 1|Z|1\rangle \langle 1|I|1\rangle + \\
& \langle 0|I|0\rangle \langle 0|I|0\rangle \langle 1|Z|1\rangle \langle 1|Z|1\rangle \tag{3.14} \\
& = \\
& \langle 0|0\rangle \langle 0|0\rangle \langle 1|1\rangle \langle 1|1\rangle + \\
& \langle 0|0\rangle \langle 0|0\rangle \langle 1| - 1\rangle \langle 1|1\rangle + \\
& \langle 0|0\rangle \langle 0|0\rangle \langle 1| - 1\rangle \langle 1| - 1\rangle \\
& = \\
& - 1
\end{aligned}$$

where I is the 2x2 identity matrix.

This is one of the (diagonal) matrix elements in the Hamiltonian for the particular sector being observed, without the scaling of the constant J_z . We can repeat this procedure to construct any Hamiltonian for any sector, rapidly.

For the integrable case, the Hamiltonian $H = J_z \sum_i Z_i Z_{i+1}$ is exactly solved by the Bethe ansatz and it is known that these classes of systems (integrable) do not thermalize but non-integrable Hamiltonians such as the one we have seen previously (2.19) do. Recall, the extra term is a small perturbation which is expected to bring our system to equilibrium.

3.3 Results and Conclusion

Our results can be summarized as follows:

Table 3.1: Even Qubit Spin Chains

qubits	up qubits	Sector Dimension	RMS Deviation
4	2	6	0
6	3	20	0
8	4	70	0
10	5	252	0
12	6	924	0
14	7	3432	0
16	8	12870	0

Table 3.2: Odd Qubit Spin Chains

qubits	up qubits	Sector Dimension	RMS Deviation
3	1	3	0.170255
5	2	10	0.112704
7	3	35	0.0812252
9	4	126	0.0616182
11	5	462	0.0488317
13	6	1716	0.0399212
15	7	6435	0.0334508
17	8	24310	0.0285675

The results displayed in Table 1 were unexpected. We thought the deviations would decrease exponentially in the dimensionality of the sector. As one can see the rms deviation is always zero for the maximally mixed state, meaning that the probability of finding the system in one of the completely mixed states is exactly what one would expect it to be from theoretical predictions, that is, all the states are typical states.

It is important to note that the deviation decreases as a power-law whose power is -0.13510 , in the sector dimension, as the number of spins increases to infinity (asymptotic power), giving increased typicality. This is easily seen graphically. Clearly we can see in

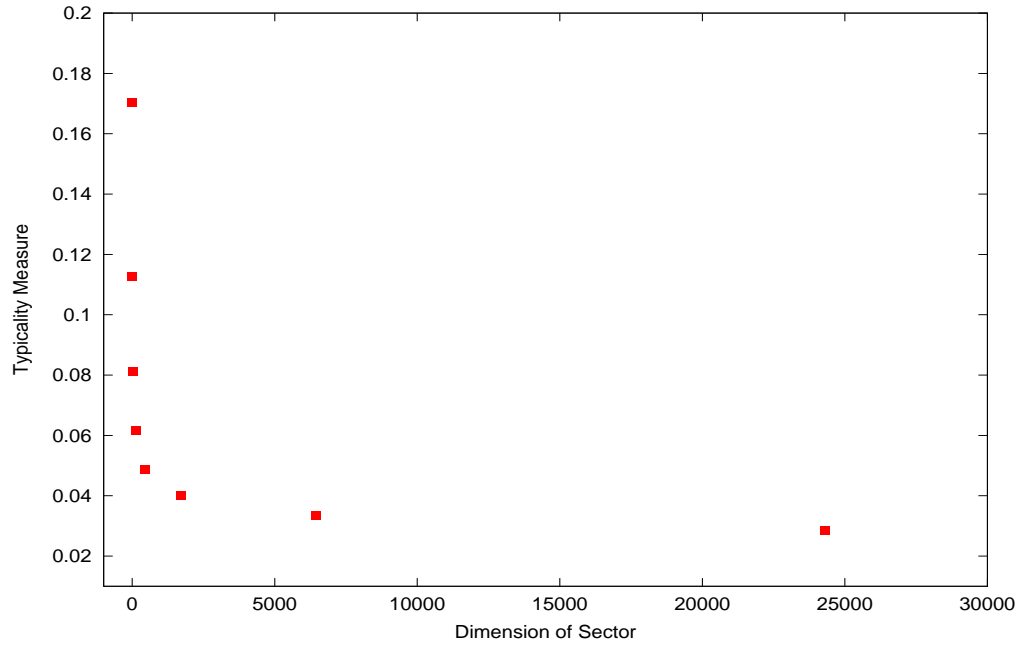


Figure 3.2: RMS Deviation-odd qubit chain

Fig. 3.4 (below) the power-law nature of δ . The deviations from the expected probability decrease with increasing spin-chain size.

The deviations δ , a measure of eigenvector typicality, indicate that the eigenvectors of a sector Hamiltonian are increasingly typical. That is, the rms deviation in probability (given by the eigenvectors), in finding the system in one of the configurations of the spin-chain in a sector, decreases. It decreases with increasing sector size. This, the typicality of the eigenvectors, our smoking gun, is our implication for Eigenstate Thermalization Hypothesis.

This idea of typicality goes a long way in remolding our understanding of statistical mechanics. We can show typicality numerically and it gives us a new platform from which to study the approach to equilibrium in quantum systems.

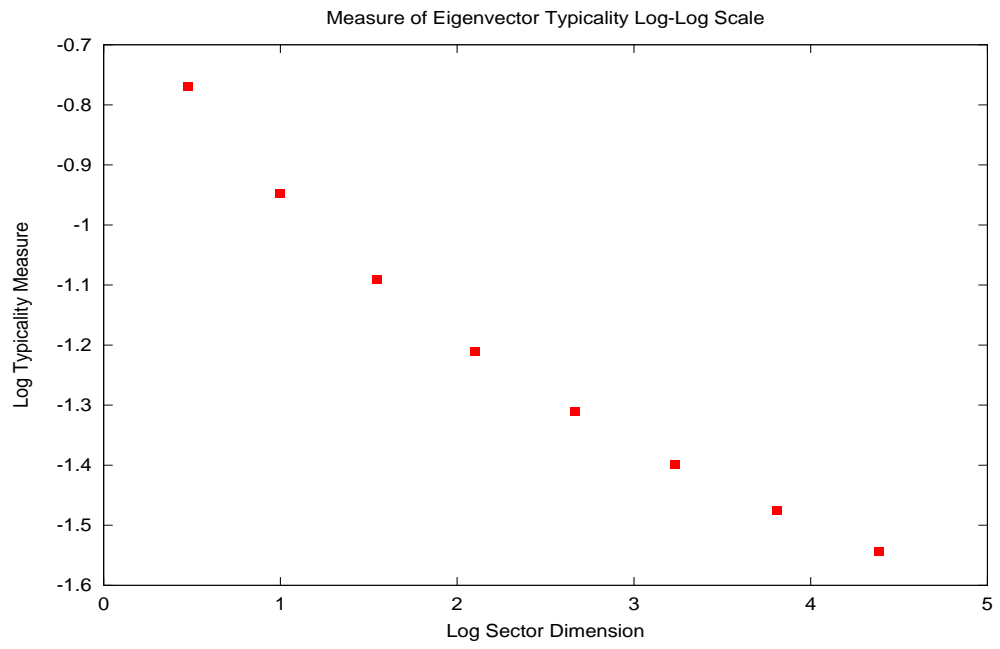


Figure 3.3: log-log RMS Deviation-odd qubit chain

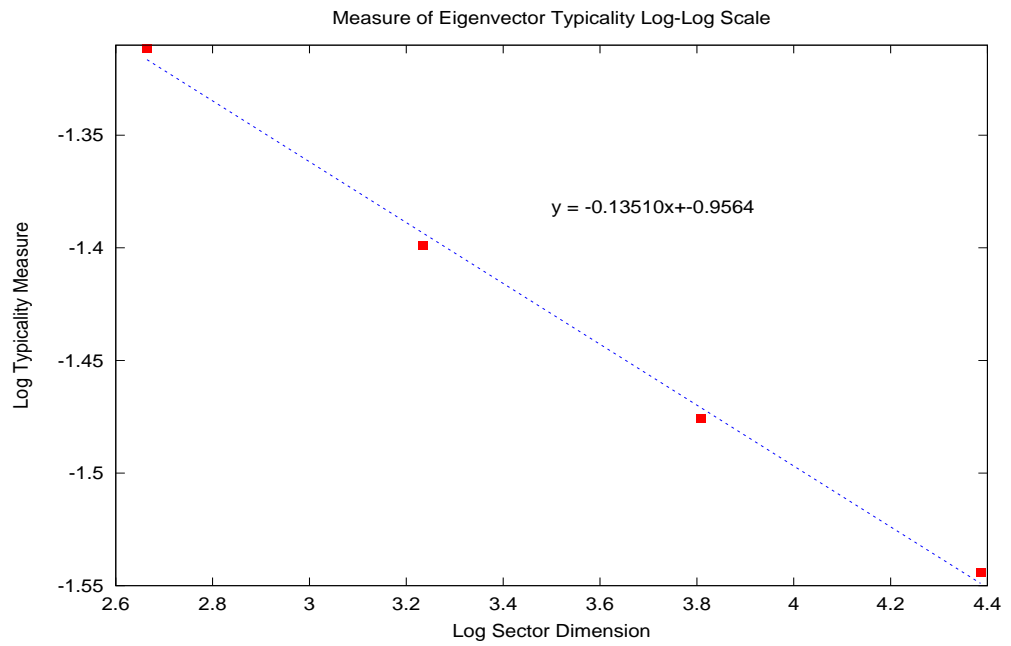


Figure 3.4: log-log RMS Deviation-reduced odd qubit chain

3.4 Technical Information

The simulations were done with code written in C++ on a Linux platform. Code execution took place on a Linux cluster at the University of Oregon, with 128 gigabytes of shared random access memory (RAM), provided by Daniel Steck. The cluster was necessary as there were large matrices to be diagonalized, the largest of which was 24310×24310 .

3.5 Suggestions for Further Research

This data gives numerical evidence for the eigenstate thermalization hypothesis through the measure of eigenvector typicality. This method can easily be extended to qutrits, hopefully reproducing the same results. Another avenue of investigation would be to look at eigenstate thermalization through the lens of many-body localization. It would also be of interest to be able to reproduce these results in a laboratory setting.

APPENDIX A

C++ CODE

```
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <cmath>
#include <iomanip>
#include <algorithm>
#include <cstring>
#include <complex>
#include <vector>
#include <ctime>
#include <functional>
#include <numeric>

using namespace std;

extern "C" {
    // ————— LAPACK routines
    // // diagonalize symmetric real
    void dsyev_(const char* choose, const char* uplow, long int* N, double*
        A, long int* rownum, double* Evals, double* WORK, long int* LWORK,
        long int* INFO );
}
```

```

long int bin(long int n, long int k); //Binomial coeffs, CF
vector<int> XXActOn(long int a, vector<int> &array, vector<int> &
    arrayTemp2, int length);
vector<int> ZZActOn(long int a, vector<int> &array, vector<int> &
    arrayTemp2, int length);
vector<int> XZActOn(long int a, vector<int> &array, vector<int> &
    arrayTemp2, int length);
vector<int> ZXActOn(long int a, vector<int> &array, vector<int> &
    arrayTemp2, int length);
double computeXX(long int col_num, vector<int> &arrayZZ, vector<int> &
    arrayTemp);
double computeZZ(long int col_num, vector<int> &arrayZZ, vector<int> &
    arrayTemp);
double computeXZ(long int col_num, vector<int> &arrayXZ, vector<int> &
    arrayTemp);
double computeZX(long int col_num, vector<int> &arrayZX, vector<int> &
    arrayTemp);
long int diagonalize_real(long int Dim, double* Mat, double* Evals); //
    KJ

int main(int argc, char* argv[])
{
    int maxQubit;

    const double A = -0.3827;//-0.3827;
    const double B = 0.9239;//0.9239;

    clock_t start = clock();

    maxQubit = 18;

```

```

vector<double>data; //Data saved here
vector<int>spinDat;
vector<double>Udat;
vector<int>num_spins;

for(int spins=13; spins <= maxQubit; spins = spins + 2)
{

num_spins.push_back(spins);

vector<int> flips;

//Initialize system to all "down" (0)
for(int k=0; k<spins; k++)
{
flips.push_back(0);
}

//Flip spins one-by-one, start at the end of the array
for(long int j=spins-1; j>-1; --j)
{
flips[j]=1;

//Dimensions of "matrix"
long int cols = spins;
long int rows = bin(spins, j);

//Other declarations
vector<int>store;

```

```

//double elemXZ;
//double elemZX;
double elemXX;//XX element of subspace
double elemZZ;//ZZ element of subspace

//double SelemXZ;
//double SelemZX;
double SelemXX;//scaled XX element of subspace
double SelemZZ;//scaled ZZ element of subspace

//vector<int>storeTempXZ;
//vector<int>storeTempZX;
vector<int>storeTempXX;
vector<int>storeTempZZ;

//vector<int>storeTemp2XZ;
//vector<int>storeTemp2ZX;
vector<int>storeTemp2XX;
vector<int>storeTemp2ZZ;

//vector<int>storeTemp3XZ;
//vector<int>storeTemp3ZX;
vector<int>storeTemp3XX;
vector<int>storeTemp3ZZ;

double* Evs = new double[rows];

vector<double>Hs;//Subspace Hamiltonian

//vector<double> Hxz;
//vector<double> HxzTemp;

```

```

//vector<double> Hzx;
//vector<double> HzxTemp;

vector<double> Hxx;
vector<double> HxxTemp;

vector<double>Hzz;
vector<double>HzzTemp;

//Keeping this as a matter of principle
sort (flips.begin(), flips.end());

if(j == ((spins/2)+1))
{

do
{
for(int x=0; x<cols; x++)
{
store.push_back(flips[x]);
}
}
while(next_permutation(flips.begin(), flips.end())); //Permute and
store all possible permutations row at a time

/* if(spins == 7)
{

```

```

for(int n=0; n<rows; n++)
{
for(int m=0; m<cols; m++)
{
cout<<store[n*cols+m]<<" ";
}
cout<<endl;
}
}*/

//Begin Hxx
for(int l=0; l<cols-1; l++)
{
for(int q=0; q<rows*cols; q = q+cols)
{
storeTemp2XX.resize(cols);
copy(store.begin() + q, store.begin() + q+cols , storeTemp2XX.begin
());
XXActOn(l, storeTemp2XX, storeTempXX, cols);

for(int incl=0; incl<rows*cols; incl = incl+cols)//After loop
completes the loop condition , gives l column
{
storeTemp3XX.resize(cols);
copy(store.begin() + incl, store.begin() + incl+cols ,
storeTemp3XX.begin());

elemXX = computeXX(cols , storeTempXX, storeTemp3XX);
SelemXX = B*B*elemXX;
HxxTemp.push_back(SelemXX);
}
}
}

```

```

        storeTemp3XX.clear();
    }

    storeTemp2XX.clear();
    storeTempXX.clear();

}

if(l == 0)
{
    for(int it2=0; it2<HxxTemp.size(); it2++)
    {
        Hxx.push_back(HxxTemp[it2]);
    }
}
else
{
    transform(HxxTemp.begin(), HxxTemp.end(), Hxx.begin(), Hxx.begin(),
        plus<double>());
}

HxxTemp.clear();
}//End Hxx

//Begin Hzz
for(int l=0; l<cols-1; l++)
{
    for(int q=0; q<rows*cols; q = q+cols)
    {

```

```

storeTemp2ZZ.resize(cols);
copy(store.begin() + q, store.begin() + q+cols , storeTemp2ZZ.begin
    ());
ZZActOn(1, storeTemp2ZZ, storeTempZZ, cols);

for(int incl=0; incl<rows*cols; incl = incl+cols)
{
    storeTemp3ZZ.resize(cols);
    copy(store.begin() + incl, store.begin() + incl+cols ,
        storeTemp3ZZ.begin());

    elemZZ = computeZZ(cols, storeTempZZ, storeTemp3ZZ);
    SelemZZ = A*A*elemZZ;
    HzzTemp.push_back(SelemZZ);

    storeTemp3ZZ.clear();
}

storeTemp2ZZ.clear();
storeTempZZ.clear();
}

if(1 == 0)
{
    for(int it2=0; it2<HzzTemp.size(); it2++)
    {
        Hzz.push_back(HzzTemp[it2]);
    }
}
else
{

```



```

    transform(HzzTemp.begin(), HzzTemp.end(), Hzz.begin(), Hzz.begin(),
              plus<double>());
}

HzzTemp.clear();
} // End Hzz

```

```

transform(Hzz.begin(), Hzz.end(), Hxx.begin(), Hxx.begin(), plus<
          double>());
// transform(Hxx.begin(), Hxx.end(), Hxz.begin(), Hxz.begin(), plus<
            double>());
// transform(Hxz.begin(), Hxz.end(), Hzx.begin(), Hzx.begin(), plus<
            double>());

```

```

for(int p=0; p<Hxx.size(); p++)
{
    Hs.push_back(Hxx[p]);
}

```

```

Hzz.clear();
Hxx.clear();

```

```

diagonalize_real(rows, &(Hs[0]), Evs); //Diagonalizes Hs, Hs comes
    back as unitary for the subspace, Evs stores evals

```

```

//Squares elements of the unitary

```

```

transform(Hs.begin(), Hs.end(), Hs.begin(), Hs.begin(), multiplies<
            double>());

```

```

double total2 = 0.0;
double delta = 0.0;
double t1 = bin(spins-1, j);
double t2 = bin(spins-1, j);
double t3 = bin(spins-1, spins-j);
double ratio = (t1/(t2+t3));

// Calculations
for(long int b=0; b<rows; b++)//Indexes e-vec
{

    for(long int c=0; c<rows; c++)
    {
        Udat.push_back(Hs[b*rows+c]);
    }

    double d = 0.0;
    for(long int m=0; m<cols; m++)//Indexes qubit
    {
        for(long int n=1; n<=rows; n++)
        {
            spinDat.push_back(store[m + (n-1)*cols]);
        }

        double sum = 0.0;
        double init = 0.0;

        sum = inner_product(Udat.begin(), Udat.end(), spinDat.begin(),
            init);
    }
}

```

```

        d = d + (sum - ratio)*(sum - ratio);

        spinDat.clear();

    }

    total2 = total2 + d;

    Udat.clear();
}

delta = sqrt(total2 / (cols*rows));//Data points/deviations
data.push_back(delta);

store.clear();

}// end for if( j == ..)

//Ending Tasks
delete [] Evs;
Hs.clear();

}

flips.clear();

}//End of maxQubit for-loop

//Save data

```

```

ofstream myfile_bin("odd2.bin", ios::binary | ios::binary); // Write data
    to binary file
if(myfile_bin.is_open())
{
    for(int s=0; s<data.size(); s++)
    {
        myfile_bin << data[s]<<" ";
    }
    myfile_bin.close();
}
else
{
    cout<<"Unable to open file"<<endl;
}

ofstream myfile2("odd2.txt"); // Write data to text file
if(myfile2.is_open())
{
    for(int s=0; s<data.size(); s++)
    {
        myfile2 << data[s] <<endl;
    }
    myfile2.close();
}
else
{
    cout<<"Unable to open file"<<endl;
}

```

```

ofstream myfile_bin1("spinfo2.bin", ios::binary | ios::binary); // Write
    data to binary file
if(myfile_bin1.is_open())
{
    for(int s=0; s<num_spins.size(); s++)
    {
        myfile_bin1 << num_spins[s] <<"_";
    }
    myfile_bin1.close();
}
else
{
    cout<<"Unable to open file "<<endl;
}

ofstream myfile3("spinfo2.txt"); // Write data to text file
if(myfile3.is_open())
{
    for(int s=0; s<num_spins.size(); s++)
    {
        myfile3 << num_spins[s]<<endl;
    }
    myfile3.close();
}
else
{
    cout<<"Unable to open file "<<endl;
}
//

```

```
clock_t ends = clock();
cout<<"Runtime:␣" << ( ( clock() - start ) / (double)CLOCKS_PER_SEC ) <<
    "␣seconds" << '\n';

return 0;

}
```

```
//-----Additional Routines-----
```

```
long int bin(long int n, long int k)
{

    long int num;
    long int den;
    long int i=1;

    if (n<k)

        {
            return 0;
        }

}
```

```

        else if ( k == 0)
    {
        return 1;
    }

    else
    {
        den = 1;

        for (i = 1; i <= k; i = i+1)
        {
            den = i*den;

            num = 1;
        }
        for (i = n; i >=(n-k+1); i--)
        {
            num = i*num;
        }
        return(num/den);
    }
}

```

```

vector<int> XXActOn(long int a, vector<int> &array, vector<int> &
    arrayTemp2, int length)
{
    for(long int k=0; k<length; k++)
    {
        arrayTemp2.push_back(array[k]);
    }
}

```

```

if(arrayTemp2[a] == 1 && arrayTemp2[a+1] == 1)
{
    arrayTemp2[a] = 0;
    arrayTemp2[a+1] = 0;
}
else if (arrayTemp2[a] == 0 && arrayTemp2[a+1] == 0)
{
    arrayTemp2[a] = 1;
    arrayTemp2[a+1] = 1;
}
else if(arrayTemp2[a] == 0 && arrayTemp2[a+1] == 1)
{
    arrayTemp2[a] = 1;
    arrayTemp2[a+1] = 0;
}
else // (arrayTemp2[a] == 1 && arrayTemp2[a+1] == 0)
{
    arrayTemp2[a] = 0;
    arrayTemp2[a+1] = 1;
}

return arrayTemp2;
}

vector<int> XZActOn(long int a, vector<int> &array, vector<int> &
    arrayTemp2, int length)
{
for(long int k=0; k<length; k++)
{
    arrayTemp2.push_back(array[k]);
}
}

```



```

}

if(arrayTemp2[a] == 1 && arrayTemp2[a+1] == 1)
{
    arrayTemp2[a] = 0;
    arrayTemp2[a+1] = -1;
}
else if (arrayTemp2[a] == 0 && arrayTemp2[a+1] == 0)
{
    arrayTemp2[a] = 1;
    arrayTemp2[a+1] = 0;
}
else if(arrayTemp2[a] == 0 && arrayTemp2[a+1] == 1)
{
    arrayTemp2[a] = 1;
    arrayTemp2[a+1] = -1;
}
else // (arrayTemp2[a] == 1 && arrayTemp2[a+1] == 0)
{
    arrayTemp2[a] = 0;
    arrayTemp2[a+1] = 0;
}

return arrayTemp2;
}

vector<int> ZXActOn(long int a, vector<int> &array, vector<int> &
    arrayTemp2, int length)
{
for(long int k=0; k<length; k++)
{

```

```

arrayTemp2 . push_back ( array [ k ] );
}

if ( arrayTemp2 [ a ] == 1 && arrayTemp2 [ a+1 ] == 1 )
{
    arrayTemp2 [ a ] = -1;
    arrayTemp2 [ a+1 ] = 0;
}
else if ( arrayTemp2 [ a ] == 0 && arrayTemp2 [ a+1 ] == 0 )
{
    arrayTemp2 [ a ] = 0;
    arrayTemp2 [ a+1 ] = 1;
}
else if ( arrayTemp2 [ a ] == 0 && arrayTemp2 [ a+1 ] == 1 )
{
    arrayTemp2 [ a ] = 0;
    arrayTemp2 [ a+1 ] = 0;
}
else // ( arrayTemp2 [ a ] == 1 && arrayTemp2 [ a+1 ] == 0 )
{
    arrayTemp2 [ a ] = -1;
    arrayTemp2 [ a+1 ] = 1;
}

return arrayTemp2;
}

vector<int> ZZActOn( long int a , vector<int> &array , vector<int> &
    arrayTemp2 , int length )
{

```

```

for(long int k=0; k<length; k++)
{
    arrayTemp2.push_back(array[k]);
}

if(arrayTemp2[a] == 1 && arrayTemp2[a+1] == 1)
{
    arrayTemp2[a] = -1;
    arrayTemp2[a+1] = -1;
}
else if (arrayTemp2[a] == 0 && arrayTemp2[a+1] == 0)
{
    arrayTemp2[a] = 0;
    arrayTemp2[a+1] = 0;
}
else if(arrayTemp2[a] == 0 && arrayTemp2[a+1] == 1)
{
    arrayTemp2[a] = 0;
    arrayTemp2[a+1] = -1;
}
else // (arrayTemp2[a] == 1 && arrayTemp2[a+1] == 0)
{
    arrayTemp2[a] = -1;
    arrayTemp2[a+1] = 0;
}
return arrayTemp2;
}

```

```

double computeXX(long int col_num, vector<int> &arrayZZ, vector<int> &
    arrayTemp)
{
    double dummy = 1.0;
    double piece = 1.0;
    int itr;

    for(itr=0; itr<col_num; itr++)
    {
        if((arrayZZ[itr] == 0 && arrayTemp[itr] == 1) || (arrayZZ[itr] == 1
            && arrayTemp[itr] == 0))
        {
            piece = 0;
        }
        else if(arrayZZ[itr] == 0 && arrayTemp[itr] == 0)
        {
            piece = 1;
        }
        else if(arrayZZ[itr] == 1 && arrayTemp[itr] == 1)
        {
            piece = 1;
        }
        dummy = piece*dummy;
    }

    return dummy;
}

```

```

double computeZZ(long int col_num, vector<int> &arrayZZ, vector<int> &
    arrayTemp)
{
    double dummy = 1.0;
    double piece = 1.0;
    double init = 0.0;
    int itr;

    for(itr=0; itr<col_num; itr++)
    {
        if(arrayZZ[itr] == -1 && arrayTemp[itr] == 1)
        {
            piece = -1;
        }
        else if(arrayZZ[itr] == 0 && arrayTemp[itr] == 0)
        {
            piece = 1;
        }
        else if(arrayZZ[itr] == 1 && arrayTemp[itr] == -1)
        {
            piece = -1;
        }
        else if((arrayZZ[itr] == 0 && arrayTemp[itr] == 1) || (arrayZZ[itr]
            == 1 && arrayTemp[itr] == 0))
        {
            piece = 0;
        }
        else if((arrayZZ[itr] == 0 && arrayTemp[itr] == -1) || (arrayZZ[itr]
            == -1 && arrayTemp[itr] == 0))
        {
            piece = 0;
        }
    }
}

```

```

    }
    else if(arrayZZ[itr] == -1 && arrayTemp[itr] == -1)
    {
        piece = 1;
    }
    else if(arrayZZ[itr] == 1 && arrayTemp[itr] == 1)
    {
        piece = 1;
    }

    dummy = piece*dummy;
}

return dummy;
}

double computeXZ(long int col_num, vector<int> &arrayXZ, vector<int> &
    arrayTemp)
{
    double dummy = 1.0;
    double piece = 1.0;
    int itr;

    for(itr=0; itr<col_num; itr++)
    {
        if((arrayXZ[itr] == 0 && arrayTemp[itr] == 1) || (arrayXZ[itr] == 1
            && arrayTemp[itr] == 0))
        {
            piece = 0;
        }
        else if(arrayXZ[itr] == 0 && arrayTemp[itr] == 0)

```

```

    {
        piece = 1;
    }
else if(arrayXZ[itr] == 1 && arrayTemp[itr] == 1)
    {
        piece = 1;
    }
else if(arrayXZ[itr] == 0 && arrayTemp[itr] == -1)
    {
        piece = 0;
    }
else if(arrayXZ[itr] == 1 && arrayTemp[itr] == -1)
    {
        piece = -1;
    }
////
else if(arrayXZ[itr] == -1 && arrayTemp[itr] == 0)
    {
        piece = 0;
    }
else if(arrayXZ[itr] == -1 && arrayTemp[itr] == -1)
    {
        piece = 1;
    }
else if(arrayXZ[itr] == -1 && arrayTemp[itr] == 1)
    {
        piece = -1;
    }
dummy = piece*dummy;
}

```

```

    return dummy;
}

double computeZX(long int col_num, vector<int> &arrayZX, vector<int> &
    arrayTemp)
{
    double dummy = 1.0;
    double piece = 1.0;
    int itr;

    for(itr=0; itr<col_num; itr++)
    {
        if((arrayZX[itr] == 0 && arrayTemp[itr] == 1) || (arrayZX[itr] == 1
            && arrayTemp[itr] == 0))
        {
            piece = 0;
        }
        else if(arrayZX[itr] == 0 && arrayTemp[itr] == 0)
        {
            piece = 1;
        }
        else if(arrayZX[itr] == 1 && arrayTemp[itr] == 1)
        {
            piece = 1;
        }
        else if(arrayZX[itr] == -1 && arrayTemp[itr] == 0)
        {
            piece = 0;
        }
        else if(arrayZX[itr] == -1 && arrayTemp[itr] == 1)
        {

```



```

    piece = -1;
}
///  

else if(arrayZX[itr] == 0 && arrayTemp[itr] == -1)
{
    piece = 0;
}
else if(arrayZX[itr] == -1 && arrayTemp[itr] == -1)
{
    piece = 1;
}
else if(arrayZX[itr] == 1 && arrayTemp[itr] == -1)
{
    piece = -1;
}
dummy = piece*dummy;
}

return dummy;
}

long int diagonalize_real(long int Dim, double* Mat, double* Evals) {

    // The unitary U comes back as Mat
    // If the input matrix is Mat = H, then
    //       $H = U \text{diag}(Evals) U^T$ 
    // and  $H U = U \text{diag}(eig)$  so that the columns of U are the
    //      eigenvectors

    long int info = 0;
    const char* do_vecs = "V", *up = "U";

```

```

double work_dummy[1];

// query to see what the optimal work size is
long int lwork = -1;
dsyev_(do_vecs, up, &Dim, Mat, &Dim, Evals, work_dummy, &lwork, &info);

lwork = static_cast<long int>(work_dummy[0]);
double* work = new double[lwork];

// calculate the eigenvalues and eigenvectors of A
dsyev_(do_vecs, up, &Dim, Mat, &Dim, Evals, work, &lwork, &info);

delete [] work;

return info;
}

```

CITATIONS

- [1] K. Jacobs, *An Introduction to Quantum Measurement Theory*. Unpublished Book.
- [2] J. Sethna, *Statistical Mechanics: Entropy, Order Parameters and Complexity*. USA: Oxford University Press, 2006.
- [3] S. Popescu, A. Short, and A. Winter, “Entanglement and the foundations of statistical mechanics,” *Nature Physics*, vol. 2, p. 754, 2006.
- [4] H. Ekstein, “Ergodic theorem for interacting systems,” *Phys. Rev.*, vol. 107, p. 333, 1957.
- [5] J. Deutsch, “Quantum statistical mechanics in a closed system,” *Phys. Rev. A*, vol. 43, p. 2046, 1991.
- [6] M. Srednicki, “Chaos and quantum thermalization,” *Phys. Rev. E*, vol. 50, pp. 888–901, Aug 1994.
- [7] M. Rigol, V. Dunjko, and M. Olshanii, “Thermalization and its mechanism for generic isolated quantum systems,” *Nature*, vol. 452, p. 854, 2008.
- [8] E. Wigner, “Characteristic vectors of bordered matrices with infinite dimensions,” *Ann. Of Math.*, vol. 62, p. 548, 1955.

LIST OF REFERENCES

- J. Deutsch, "Quantum statistical mechanics in a closed system," *Phys. Rev. A*, vol. 43, p. 2046, 1991.
- H. Ekstein, "Ergodic theorem for interacting systems," *Phys. Rev.*, vol. 107, p. 333, 1957.
- K. Jacobs, *An Introduction to Quantum Measurement Theory*. Unpublished Book.
- S. Popescu, A. Short, and A. Winter, "Entanglement and the foundations of statistical mechanics," *Nature Physics*, vol. 2, p. 754, 2006.
- M. Rigol, V. Dunjko, and M. Olshanii, "Thermalization and its mechanism for generic isolated quantum systems," *Nature*, vol. 452, p. 854, 2008.
- J. Sethna, *Statistical Mechanics: Entropy, Order Parameters and Complexity*. USA: Oxford University Press, 2006.
- M. Srednicki, "Chaos and quantum thermalization," *Phys. Rev. E*, vol. 50, pp. 888–901, Aug 1994.
- E. Wigner, "Characteristic vectors of bordered matrices with infinite dimensions," *Ann. Of Math.*, vol. 62, p. 548, 1955.