Toy Systems and Quantum Master Equations

(Provisional Title)

Draft

A Thesis

Presented to

The Division of Mathematics and Natural Sciences

Reed College

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Arts

Alex Striff

March 20, 2021

Approved for the Division
(Physics)

Darrell Schroeter

Acknowledgements

I want to thank a few people.

Abstract

The preface pretty much says it all.

Table of Contents

List of Figures

List of Tables

Introduction

Introductory content.

Chapter 1

Open quantum systems

What happens after performing an operation on a physical system? Our actions are uncertain, so the best we can do is to assign probabilities to the possible outcomes. Quantum mechanics is a theory for determining the probabilities of such outcomes. Physical theories model the relevant aspects of phenomena by abstracting away unnecessary information. Newtonian mechanics considers point masses while neglecting the material composition of bodies. This enables a simpler description of motion. Just as Newtonian mechanics models physical systems as "Newtonian systems" of point masses, quantum mechanics models physical systems with **quantum states** which represent only the probabilistic description of operation outcomes. A penny and a quarter are different physical systems that correspond to the same "quantum system," as far as the outcomes of coin tosses are concerned.[1]

1.1 A sketch of quantum mechanics

What follows is a sketch of quantum theory that helps motivate the mathematical formalism we use later.

We are interested in performing operations that have consistent effects, for otherwise we could make little sense of the world. Consider performing an operation with m outcomes on a quantum system. Given a particular outcome, this operation is a **measurement** if repeating the operation gives the same outcome with probability one.[2]

[1] That is, in an ideal sense. Actual coins made from many atoms are not two-outcome systems. For example, the coin atoms could melt into a blob, rendering them unflippable.

[2] On this view, the result of a measurement is defined operationally. This avoids confusion with

We may then characterize a quantum system by its **dimension** N, which is the maximum number of outcomes distinguishable by a measurement.[3] Conversely, an operation with more than N outcomes cannot result in a single outcome with certainty.

1.1.1 Entropy

Since quantum states encode the probabilities of outcomes, it is relevant to have a measure for how uncertain we are about which outcome will happen.

To do so, we must quantify the uncertainty expressed by a probability distribution of outcomes. The most successful definition of this uncertainty due to Shannon considers the **self-information** or **surprisal** I of an event with probability p [1]. This is a function I that has the following intuitive properties:[4]

1. $I(1) = 0$: Certain events are uninformative or unsurprising.
2. $I(p) < I(p')$ if $p' < p$: Less probable events are more surprising.
3. $I(p) \geqslant 0$: The information learned from an event is nonnegative.
4. $I(p,\, p') = I(p) + I(p')$: The information learned from independent events is the sum of the information learned from each event.

The only functions with these properties are $I(p) = b \log p$ for $b < 0$. Choosing b amounts to choosing the base of the logarithm and the unit of information. It is standard to use base two for which the unit of the self-information

$$I(p) = -\log p \tag{1.1}$$

is the **bit**. The uncertainty expressed by a probability distribution of outcomes is called the **Shannon entropy**, and is defined as the expected surprisal of an outcome:

$$H = \langle I \rangle \tag{1.2}$$

properties of *physical* systems: an outcome with probability one is a different kind of thing than a physical property.

[3] This supposes that such systems exist, which requires experimental verification. For example, Stern-Gerlach type experiments demonstrate the existence of a two-dimensional quantum system. Given what we know about systems like atoms in quantum physics, it is justified to assert the existence of quantum systems with any finite dimension. Infinite dimensional systems present some issues, but they may be regarded as being purely mathematical, since they can be well-approximated by finite dimensional systems as far as measurement outcomes are concerned. We usually consider only finite-dimensional systems in what follows.

[4] Technically, I is defined for a random variable X with support $\mathcal{X}$, so that $I : \mathcal{X} \to \mathbb{R}$ is a function of the probability of an event: $I(x) = f(P(x))$.

$$= -\sum_i p_i \log p_i, \tag{1.3}$$

where $0 \log 0 := \lim_{p \to 0} p \log p = 0$. Thus an operation with a certain outcome has $H = 0$. The most uncertain one can be in a m-outcome operation is to have $p_i = 1/m$. In that case, $H = \log m$.

We now define the **von Neumann entropy** S of a quantum state ρ to be the minimum entropy of any measurement of the state. If $S(\rho) = 0$, there is a measurement with a definite outcome for ρ. Since the other outcomes are excluded, ρ is called a **pure state**, rather than a **mixed state** which has probability spread out over more than one outcome.

1.1.2 Hardy's postulates

Now that we have the notion of a pure state, we may indicate postulates of quantum theory.

We expect that K real numbers are needed to describe a quantum state ρ and to predict the probabilities of outcomes. In quantum theory, we posulate the following statements about N and K:

1. K is a function of N, which we may select to be the smallest value consistent with the remaining postulates.
2. A N-dimensional system constrained to only M states distinguishable by measurement behaves like a system of dimension M.
3. A composite system consisting of subsystems A and B satisfies $N = N_A N_B$ and $K = K_A K_B$.

These statements apply equally well to classical probability theory. Quantum theory is distinguished by the final postulate:

4. There is a continuous and reversible transformation between any two pure states of a system.

These postulates are enough to reproduce quantum mechanics as we know it. While this has been just a sketch, Hardy gives a more precise description and demonstrates agreement with the formalism to follow [2]. In particular, Hardy shows that $K = N^2$, which corresponds to considering a complex Hilbert space.

1.2 The mathematical formalism

Different mathematical formalisms for quantum mechanics differ in how they represent states and operations, but they agree on the assignment of probabilities to outcomes. Motivated by section 1.1.2, we now make the following postulates:

Postulate 1. A quantum system is described by a complex Hilbert space $\mathcal{H}$.

Postulate 2. An outcome corresponds to an **effect** E, which is a self-adjoint operator on $\mathcal{H}$ such that $0 \leqslant E \leqslant I$.[5]

Postulate 3. A state corresponds to a **probability measure** P on effects. That is:

1. $0 \leqslant P(E) \leqslant 1$ for all effects E,
2. $P(I) = 1$,
3. $P(E_1 + E_2 + \cdots) = P(E_1) + P(E_2) + \cdots$ for any sequence of events with $E_1 + E_2 + \cdots \leqslant I$.

Postulate 4. States form a convex set. If $\sum_i p_i = 1$, the convex sum of states $\{P_i\}$ is defined by

$$\left(\sum_i p_i P_i\right)(E) = \sum_i p_i P_i(E). \tag{1.4}$$

Such a combination is known as an **ensemble**.

It is simple to prove that any such probability measure P on an effect E may be represented by the **Born rule**

$$P(E) = \operatorname{tr}(\rho E), \tag{1.5}$$

where $\rho \geqslant 0$ is a self-adjoint operator known as a **density operator** [3, 4].[6] Equation (1.5) implies that $\operatorname{tr}\rho = 1$ and that a convex sum of states is represented by the same sum of density operators. The Born rule uniquely identifies density operators with states, so we will use the term state to refer to density operators from now on.

An **observable** result of an operation is described by an assignment of each outcome m to an effect E_m, where $\sum_m E_m = I$. Since effects are positive operators that determine the probabilities of each outcome, such an observable is called a **positive operator valued measure** (POVM). The special case where the effects are projectors is called a **projection valued measure** (PVM). We now describe how operations change states:

[5] The notation $A \leqslant B$ means that $\langle v|A|v\rangle \leqslant \langle v|B|v\rangle$ for all $v \in \mathcal{H}$.

[6] The more specific case where effects are restricted to be projections $|v\rangle\langle v|$ for $v \in \mathcal{H}$ is significantly harder, and is known as Gleason's theorem [5].

Postulate 5. An **operation** with outcome m is described by a map $\mathcal{O}_m$. The state ρ after the operation becomes

$$\rho'_m = P(m)^{-1}\mathcal{O}_m\rho. \tag{1.6}$$

Inherent in postulate 5 is the normalization condition $\operatorname{tr}\rho' = 1$, which implies that $\operatorname{tr}\mathcal{O}_m\rho = \operatorname{tr}(\rho E_m)$, as well as the requirement that $\mathcal{O}_m\rho \geqslant 0$. If an operation is performed but the outcome is unknown, we may assign the state

$$\rho' = \sum_m P(m)\rho'_m = \sum_m \mathcal{O}_m\rho, \tag{1.7}$$

so that an effect E has the expected probability $\operatorname{tr}(\rho' E) = \langle \operatorname{tr}(\rho'_m E)\rangle_m$.

The state of the ensemble $\rho = \sum_i p_i\rho_i$ after an operation with outcome m is

$$\frac{\mathcal{O}_m\rho}{\operatorname{tr}\mathcal{O}_m\rho} = \sum_i P(i\,|\,m)\frac{\mathcal{O}_m\rho_i}{\operatorname{tr}\mathcal{O}_m\rho_i} \tag{1.8}$$

By Bayes' theorem,

$$P(i\,|\,m) = \frac{P(i)P(m\,|\,i)}{P(m)} = \frac{p_i \operatorname{tr}\mathcal{O}_m\rho_i}{\operatorname{tr}\mathcal{O}_m\rho}. \tag{1.9}$$

Now eq. (1.8) becomes

$$\mathcal{O}_m\left(\sum_i p_i\rho_i\right) = \sum_i p_i\mathcal{O}_m\rho_i. \tag{1.10}$$

Thus operations are convex linear.

1.2.1 *Composite systems*

TODO

1.2.2 *Closed dynamics*

TODO

Only unitarys preserve the (relative) entropy [6].

1.2.3 *Open dynamics*

Assume completely positive [7, 8, 9].

Chapter 2

Density operator theory

If quantum mechanics is so weird, then why aren't we? The answer lies in the phenomenon of **decoherence**, which was first considered in depth in the 1970's by Zeh [10]. Despite being a late comer to the history of quantum mechanics, the theory of decoherence is crucial to understanding how classical results are obtained from many interacting quantum systems. While there are other routes to decoherence, the most common is through interaction with a memoryless (Markovian) environment [11]. This leads to the theory of quantum Markovian master equations, which describe transformations of a system in the presence of such an environment. The most general form of these transformations was first extended to quantum mechanics by Gorini, Kossakowski, Lindblad, and Sudarshan to give the **gkls** or **Lindblad equation** [12, 13].

This chapter will explain the relevant theoretical background in sections 2.2 and 2.3, before presenting the general theory which leads to the Lindblad equation in section 2.5, following the text of Breuer and Petruccione [14] and to a lesser extent [15] (which has some flaws). This leads us to consider the weak-coupling limit in section 2.6 and an application to atomic physics in section 2.8. Issues with the method are discussed briefly in section 2.9.

2.1 The quantum formalism as information processing

A goal of science is to predict the results of our actions through experiments. The formalism of quantum mechanics provides guidance on what results to expect, given what one already believes about an experiment.

Upon taking an action, we expect that different sensations could result. In experiments, we limit sensations to parts of the world that we understand.

Experiments have different outcomes.

Consider an experiment with M designated outcomes. Before, one expects each outcome with some probability. An experiment constitutes a **measurement** if repeating the experiment does not change the probability of an outcome. That is, measuring an outcome again tells you nothing. More precisely, the outcome of a measurement proper cannot be dependent on additional information, so that one is able to undo a measurement by forgetting the outcome.

Clarify and distinguish selective vs nonselective measurements, and relate to decoherence.

A state ρ encodes one's beliefs about the outcomes of measurements, if any. States are classified by the numbers of outcomes they can describe.

As the result of an experiment, one assigns a probability p_m to each outcome m. The defining characteristic of a state ρ is the maximum amount of information it can contain about the outcomes of a measurement. The entropy $H(m)$.

The size of a system is the largest number N of outcomes distinguishable by a measurement.

A quantum state ρ of **dimension** N represents one's beliefs about measurements with at most N outcomes.

One takes an action with an uncertain outcome. This uncertainty can be quantified by how much one would bet (b) that it would not happen. If something is certain, nobody would bet anything and $b = 0$. If something is certain not to happen, one would bet as much as possible, written as $b = \infty$. The more uncertain, the more one would bet. Also, if two actions are independent, then one bets for both the sum of the bets for each.

This uncertainty is quantified by a probability p_1. If one takes two actions, there are four possibilities.

For independent actions, performing one action does not change the probability of the others. Then $\rho = (p_1, p_2)$ describes bettability for both.

$$P_m(p_1, \ldots, p_n) = p_m \tag{2.1}$$

Given what one knows, one gives a probability assignment. But one may have reason to give several probability assignments, each with some further probability that it is sound. This is expressed as a formal linear combination:

$$\rho = \sum_i p_i \rho_i, \tag{2.2}$$

which means that the probability of outcome m is

$$P_m(\rho) = \sum_i p_i P_m(\rho_i), \tag{2.3}$$

We need three ingredients:

- The quantum state ρ.
- A map $\langle \rho \rangle$ for the probability that one of the outcomes is realized.
- Maps $\mathcal{O}_m$ for $m \in M$.

The state after measurement is $\mathcal{O}_m(\rho)$. One may combine states ρ_i with probabilities p_i to form a new state ρ such that

$$P_m(\rho) = \sum_i p_i P_m(\rho_i) \tag{2.4}$$

$$\mathcal{O}_m(\rho) = \sum_i p_i P_m(\rho_i) \mathcal{O}_m(\rho_i) \tag{2.5}$$

Complete positivity?

$$\mathcal{O}_m\left(\sum_i p_i \rho_i\right) = \sum_i p_i \mathcal{O}_m(\rho_i) \tag{2.6}$$

$$(\mathcal{O}_m \otimes I)\left(\sum_i p_i \rho_i \otimes \rho_i'\right) = \sum_i p_i \mathcal{O}_m(\rho_i) \otimes \rho_i' \tag{2.7}$$

- Actions $\mathcal{A}$.
- Bet ρ.
- Certain bet I_A.

- $A(I_{A'}) = p_{A'A} I_A$.
- $A(aI_{A'} + I_B) = aA(I_{A'}) + A(I_B)$.
- Bet for two independent actions $\rho \otimes \rho'$
- Split bet $\rho = p\rho + (1-p)\rho'$, where $0 \leq p \leq 1$.
- Bet probability $\langle \rho \rangle$.
- $\langle I \rangle = 1$.
- $\langle 0 \rangle = 0$.
- $\langle \rho \otimes \rho' \rangle = \langle \rho \rangle \langle \rho' \rangle$.
- $\langle p\rho + p'\rho' \rangle = p \langle \rho \rangle + p' \langle \rho' \rangle$.
- $\mathcal{A}_m$

Postulate 6. A **state** is described by a **density operator** ρ, which is positive and self-adjoint.

The result of an operation on a state may be uncertain. So long as the result is as expected, we say that the state remains present.

Postulate 7. The probability $\langle \rho \rangle$ that a state is present is $\operatorname{tr} \rho$.

Often a state is always present, and this is supposed to be the starting point for a series of operations. Such states have $\operatorname{tr} \rho = 1$ and are said to be **normalized**.

One may be uncertain about which state to use. We may consider states ρ_i, each with credence p_i. The expected probability that the state is present is then

$$\sum_i p_i \operatorname{tr} \rho_i = \operatorname{tr}\left(\sum_i p_i \rho_i \right). \tag{2.8}$$

Thus postulate 7 allows one to encode this collection of states as the convex sum $\rho = \sum_i p_i \rho_i$. We have $\sum_i p_i = 1$ so that a combination of normalized states remains normalized.

Postulate 8. An **operation**[1] on a state with M possible outcomes is described by a set of M maps $\mathcal{O}_m : \mathcal{L} \to \mathcal{L}$. After possibly effecting outcome m, the state is $\mathcal{O}_m \rho$.

[1] Quantum operations are also called generalized measurements. However, many operations are not thought of as measurements of physical quantities.

Thus the probability of outcome m is $\operatorname{tr} \mathcal{O}_m \rho$. There are two ways that an operation may change expectations about a state.[2]

1. To perform an operation on a state is to form expectations about the state by considering each of the possible outcomes:

$$\rho' = \sum_m \mathcal{O}_m \rho. \tag{2.9}$$

2. If outcome m is possibly present, one may instead form expectations conditioned on the event that outcome m was the result of the operation:

$$\rho' = \frac{\mathcal{O}_m \rho}{\operatorname{tr} \mathcal{O}_m \rho}. \tag{2.10}$$

One may again consider a collection of states, where one expects the probability of outcome m to be

$$\sum_i p_i \operatorname{tr} \mathcal{O}_m \rho_i = \operatorname{tr} \left(\sum_i p_i \mathcal{O}_m \rho_i \right). \tag{2.11}$$

This should be consistent with

$$\operatorname{tr} \mathcal{O}_m \rho = \operatorname{tr} \mathcal{O}_m \left(\sum_i p_i \rho_i \right). \tag{2.12}$$

This suggests our next postulate.

Postulate 9. Operations are convex linear:

$$\mathcal{O}_m \left(\sum_i p_i \rho_i \right) = \sum_i p_i \mathcal{O}_m(\rho_i). \tag{2.13}$$

Some operations are independent. This situation may be described with two states $\rho_1 \in \mathcal{L}_1$ and $\rho_2 \in \mathcal{L}_2$. We would like to consider both states simultaneously as a composite state (ρ_1, ρ_2), such as where both ρ_1 is present and ρ_2 is not. As before, one may be uncertain which composite state to choose, so we may form combinations like

$$\rho = p(\rho_1, \rho_2) + p'(\rho_1', \rho_2). \tag{2.14}$$

[2] For operations which may be thought of as measurements, these two ways are also known as selective and non-selective measurements, respectively. Selective measurements measure an outcome as usual, while non-selective measurements are said to measure an outcome and then forget the results.

This means that we consider the composite state (ρ_1, ρ_2) with credence p and (ρ_1', ρ_2) with credence p'. In either case, we consider that the second state is ρ_2, while the first state is ρ_1 or ρ_1'. As such, this is the same as considering the composite state

$$\rho' = (p\rho_1 + p'\rho_1', \rho_2). \tag{2.15}$$

While ρ and ρ' formally different linear combinations of states, they mean the same thing. We thus consider composite states as equivalence classes of formal linear combinations. The classes are known as **tensors**, and the corresponding map $\otimes$ from $\mathscr{L}_1 \times \mathscr{L}_2$ to the space of tensors $\mathscr{L}_1 \otimes \mathscr{L}_2$ is known as the **tensor product**. The composite of ρ_1 and ρ_2 is written as $\rho_1 \otimes \rho_2$. The probability that both systems are present is $\mathrm{tr}(\rho_1 \otimes \rho_2) := \mathrm{tr}\,\rho_1 \,\mathrm{tr}\,\rho_2$.

Maps $\mathcal{O}_1$ on $\mathscr{L}_1$ and $\mathcal{O}_2$ on $\mathscr{L}_2$ admit a composite map which acts on $\mathscr{L}_1 \otimes \mathscr{L}_2$ defined by

$$(\mathcal{O}_1 \otimes \mathcal{O}_2)(\rho_1 \otimes \rho_2) = \mathcal{O}_1\rho_1 \otimes \mathcal{O}_2\rho_2 \tag{2.16}$$

and extended linearly.[3] As such, an operation on ρ_1 can only change ρ_1, no matter how it is considered as a composite with other states. This is our final postulate.

Postulate 10. Operation maps are **completely positive**. That is, for any Liouville space $\mathscr{L}'$, the map $\mathcal{O}_m \otimes I$ defined by

$$(\mathcal{O}_m \otimes I)\left(\sum_i A_i \otimes B_i\right) := \sum_i \mathcal{O}_m(A_i) \otimes B_i \tag{2.17}$$

is positive if $A_i \in \mathscr{L}$ and $B_i \in \mathscr{L}'$ are positive.

2.2 *Different perspectives on density operators*

For a statistical perspective on quantum mechanics, we will make two postulates. The mathematical background is the **Liouville space** $\mathscr{L}(\mathscr{H})$ for the Hilbert space $\mathscr{H}$.

Definition 1 (Liouville space)**.** The space $\mathscr{L}(\mathscr{H})$ is the complex Hilbert space of operators A on $\mathscr{H}$ for which the norm induced by the inner product $\langle A|B\rangle \equiv \mathrm{tr}\left(A^\dagger B\right)$ is finite.

Postulate 11. A quantum system may be understood as a statistical ensemble ρ with observables O which are both described by elements of $\mathscr{L}(\mathscr{H})$, where the ensemble average of O is $\langle O\rangle_\rho \equiv \langle O|\rho\rangle$ and O is Hermitian.

[3] Thus $\mathscr{L}_1 \otimes \mathscr{L}_2$ is also a Liouville space under the Hilbert-Schmidt inner product.

The usual properties of the **density operator** ρ follow from considering various averages. The only way for $\langle \alpha I \rangle_\rho = \alpha$ for all physical constants $\alpha \in \mathbb{C}$ is if $\operatorname{tr} \rho = 1$. For $\langle O \rangle_\rho$ to be real, ρ must be *self-adjoint*, and if O is also positive, then ρ must be *positive* for $\langle O \rangle_\rho$ to be positive [16].

Postulate 12. The density operator for an isolated system with Hamiltonian H evolves unitarily in time according to the **Liouville–von Neumann equation**

$$\dot{\rho} = [H, \rho]/\mathrm{i}\hbar. \tag{2.18}$$

While we usually consider the density operator to change in time, the time dependence may be shifted onto the observables. Consider a quantum system with unitary time-evolution operator $U(t)$, so that we may express the density operator for the system as $\rho(t) = U(t)\rho U^\dagger(t)$, where $\rho = \rho(0)$. If we compute the ensemble average of an observable $O(t)$ and cycle the trace, we find

$$\langle O(t) \rangle_{\rho(t)} = \operatorname{tr}\Big(O(t)U(t)\rho U^\dagger(t)\Big) \tag{2.19}$$

$$= \operatorname{tr}\Big(U^\dagger(t)O(t)U(t)\rho\Big) \equiv \langle O_H(t) \rangle_\rho \tag{2.20}$$

where O_H is the observable in the **Heisenberg picture**, as opposed to the **Schrödinger picture**, where the operators are time-independent. If we can split the Hamiltonian into the form $H = H_0 + H_I(t)$, then $U(t)$ splits into the product of $U_0(t) = e^{H_0 t/\mathrm{i}\hbar}$ and $U_I(t) = U_0^\dagger(t)U(t)$. Cycling over only $U_I(t)$ in eq. (2.19) gives the **interaction picture** operators

$$O_I(t) = U_0^\dagger(t)O(t)U_0(t) \tag{2.21a}$$

$$\rho_I(t) = U_I(t)\rho U_I^\dagger(t). \tag{2.21b}$$

Without $H_I(t)$, eq. (2.21) reduces to the Schrödinger picture, and without H_0, eq. (2.21) reduces to the Heisenberg picture. The time-dependence of the interaction picture density operator from differentiating eq. (2.21b) is (suppressing time dependences)

$$\begin{aligned}
\mathrm{i}\hbar\,\dot{\rho}_I(t) &= \mathrm{i}\hbar \frac{\mathrm{d}}{\mathrm{d}t}\Big(U_0^\dagger \rho(t) U_0\Big) \\
&= -U_0^\dagger H_0^\dagger \rho(t) U_0 + U_0^\dagger \rho(t) H_0 U_0 \\
&\quad + U_0^\dagger [H_0, \rho(t)] U_0 + U_0^\dagger [H_I, \rho(t)] U_0 \\
&= U_0^\dagger \Big[U_0 H_I' U_0^\dagger, \rho(t)\Big] U_0 \\
&= \big[H_I', \rho_I(t)\big],
\end{aligned} \tag{2.22}$$

where H_I' denotes the interaction Hamiltonian $H_I(t)$ in the interaction picture. This is just eq. (2.18) with the interaction Hamiltonian.

With this understanding of the behavior of isolated systems, it may be surprising that postulates II and I2 are actually insufficient to describe common systems. For example, the allowed energies for the harmonic oscillator are unbounded, so the Hamiltonian is not an element of the Liouville space. We will see later how this issue is related to the dynamics of a composite quantum system in section 2.9, but will now move on to considering the more general dynamics of interacting quantum systems.

2.3 Composite quantum systems

Clean up inner product notation.

Consider a quantum system described by a Hilbert space $\mathcal{H}$ that appears to be made of two subsystems.

Consider two quantum systems described by Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$. We should be able to view both systems as parts of a composite quantum system described by some Hilbert space $\mathcal{H}$. For the moment, we will only consider the Hilbert spaces as vector spaces. Mathematically, we have a map $\phi : \mathcal{H}_A \times \mathcal{H}_B \to \mathcal{H}$. The simplest choice would be $\mathcal{H} = \mathcal{H}_A \times \mathcal{H}_B$, but this does not make physical sense. For the assignment of composite states to be consistent, if a linear combination is taken in $\mathcal{H}_A$ then joined with $\mathcal{H}_B$, the result must be the same as first joining and then taking the linear combination. If $|a_1\rangle$, $|a_2\rangle \in \mathcal{H}_A$, $|b\rangle \in \mathcal{H}_B$, and $\alpha, \beta \in \mathbb{C}$, then

$$\phi(\alpha\,|a_1\rangle + \beta\,|a_2\rangle\,,\,|b\rangle) = \alpha\phi(|a_1\rangle\,,\,|b\rangle) + \beta\phi(|a_2\rangle\,,\,|b\rangle), \tag{2.23}$$

so that ϕ is bilinear.

Taking $\mathcal{H} = \mathcal{H}_A \times \mathcal{H}_B$ by doing operations elementwise is known as the **direct sum** $\mathcal{H}_A \oplus \mathcal{H}_B$. Applying $\phi' : \mathcal{H}_A \times \mathcal{H}_B \to \mathcal{H}_A \oplus \mathcal{H}_B$ gives

$$\begin{aligned}
\alpha\phi'(|a_1\rangle\,,\,|b\rangle) + \beta\phi'(|a_2\rangle\,,\,|b\rangle) &= \alpha(|a_1\rangle\,,\,|b\rangle) + \beta(|a_2\rangle\,,\,|b\rangle) && (2.24)\\
&= (\alpha\,|a_1\rangle + \beta\,|a_2\rangle\,,\,2\,|b\rangle) && (2.25)\\
&\neq (\alpha\,|a_1\rangle + \beta\,|a_2\rangle\,,\,|b\rangle) && (2.26)\\
&= \phi'(\alpha\,|a_1\rangle + \beta\,|a_2\rangle\,,\,|b\rangle), && (2.27)
\end{aligned}$$

since $|b\rangle$ is normalized in $\mathcal{H}_B$. Thus the direct sum is not a physically consistent construction of a composite system.

Instead, we may define the composite system and corresponding map so that they satisfy eq. (2.23). We consider the combination of two states first as a symbol, giving the set of symbols

$$B = \left\{\tilde{\phi}(|a\rangle, |b\rangle) \,:\, |a\rangle \in \mathcal{H}_A,\, |b\rangle \in \mathcal{H}_B\right\}. \tag{2.28}$$

The formal complex linear combinations of elements of B form a vector space, known as the **free vector space** $F(\mathcal{H}_A \times \mathcal{H}_B)$. The composite system is then the quotient vector space $\mathcal{H} = F(\mathcal{H}_A \times \mathcal{H}_B)/\sim$, where $\sim$ is an equivalence relation that identifies linear combinations like in eq. (2.23). These equivalence classes are known as **tensors**. We write the class $\phi(|a\rangle, |b\rangle) = [\tilde{\phi}(|a\rangle, |b\rangle)]$ as $|a\rangle \otimes |b\rangle$, which is often abbreviated to $|a\rangle|b\rangle$ or $|a, b\rangle$.

To make $\mathcal{H}$ a Hilbert space, we may complete it with a suitable inner product. Consider the special case when the states are normalized and $\langle a_1|a_2\rangle = 0$. We have another consistency criterion: the composite state should also be normalized. We find

$$\begin{aligned}
1 &= \langle(\alpha|a_1\rangle + \beta|a_1\rangle) \otimes |b\rangle|(\alpha|a_1\rangle + \beta|a_1\rangle) \otimes |b\rangle\rangle && (2.29)\\
&= |\alpha|^2 \langle|a_1\rangle \otimes |b\rangle||a_1\rangle \otimes |b\rangle\rangle + |\beta|^2 \langle|a_2\rangle \otimes |b\rangle||a_2\rangle \otimes |b\rangle\rangle && (2.30)\\
&+ \alpha\beta^* \langle|a_1\rangle \otimes |b\rangle||a_2\rangle \otimes |b\rangle\rangle + \alpha^*\beta \langle|a_2\rangle \otimes |b\rangle||a_1\rangle \otimes |b\rangle\rangle && (2.31)\\
&= |\alpha|^2 + |\beta|^2 + \alpha\beta^* \langle|a_1\rangle \otimes |b\rangle||a_2\rangle \otimes |b\rangle\rangle + \alpha^*\beta \langle|a_2\rangle \otimes |b\rangle||a_1\rangle \otimes |b\rangle\rangle. && (2.32)
\end{aligned}$$

This requires that

$$\alpha\beta^* \langle|a_1\rangle \otimes |b\rangle||a_2\rangle \otimes |b\rangle\rangle = -\alpha^*\beta \langle|a_1\rangle \otimes |b\rangle||a_2\rangle \otimes |b\rangle\rangle^*, \tag{2.33}$$

so if $\alpha,\ \beta \neq 0$, the inner product $\langle|a_1\rangle \otimes |b\rangle||a_2\rangle \otimes |b\rangle\rangle$ must be purely imaginary. But since the superposition state is to 1, this establishes that $|a_1\rangle \otimes |b\rangle$ and $|a_2\rangle \otimes |b\rangle$ must be orthonormal. In general, consistency requires that

$$\left\langle|a_i\rangle \otimes |b_m\rangle\middle||a_j\rangle \otimes |b_n\rangle\right\rangle = \delta_{ij}\delta_{mn}. \tag{2.34}$$

Hence consistency has determined the inner product. For arbitrary vectors $|a\rangle = \sum_i c_i |a_i\rangle$, $|\tilde{a}\rangle = \sum_j c'_j |a_j\rangle$, $|b\rangle = \sum_m d_m |b_m\rangle$, and $\left|\tilde{b}\right\rangle = \sum_n d'_n |b_n\rangle$,

$$\begin{aligned}
\left\langle|a\rangle \otimes |b\rangle\middle||\tilde{a}\rangle \otimes \left|\tilde{b}\right\rangle\right\rangle &= \sum_{ijmn} c_i \tilde{c}_j^* d_m \tilde{d}_n^* \langle|a_i\rangle \otimes |b_m\rangle||a_j\rangle \otimes |b_n\rangle\rangle && (2.35)\\
&= \sum_{ijmn} c_i \tilde{c}_j^* d_m \tilde{d}_n^* \delta_{ij}\delta_{mn} && (2.36)\\
&= \left(\sum_{ij} c_i \tilde{c}_j^* \langle a_i|a_j\rangle\right)\left(\sum_{mn} d_m \tilde{d}_n^* \langle b_m|b_n\rangle\right) && (2.37)
\end{aligned}$$

$$= \langle a|\tilde{a}\rangle \left\langle b\middle|\tilde{b}\right\rangle . \tag{2.38}$$

Equation (2.38) is then a valid definition for the inner product, with consistency upheld by extending to linear combinations of tensors.

Given an operator A on $\mathcal{H}_A$, what is the corresponding operator $\tilde{A}$ on $\mathcal{H}$? For $\langle a, b|\tilde{A}|a, b\rangle = \langle a|A|a\rangle$, we must have

$$\tilde{A} = A \otimes I, \tag{2.39}$$

where action of the operator $A \otimes B$ on $\mathcal{H}$ is defined by

$$(A \otimes B)(|a\rangle \otimes |b\rangle) = A\,|a\rangle \otimes B\,|b\rangle . \tag{2.40}$$

The **adjoint** $T^\dagger$ of $T : \mathcal{H}_A \to \mathcal{H}_B$ satisfies that

$$\langle T\,|a\rangle || b\rangle\rangle = \left\langle |a\rangle \middle| T^\dagger\,|b\rangle \right\rangle \quad \text{for} \quad |a\rangle \in \mathcal{H}_A,\ |b\rangle \in \mathcal{H}_B. \tag{2.41}$$

Thus for $A \otimes B : \mathcal{H} \to \mathcal{H}$,

$$\begin{aligned}
\langle (A \otimes B)(|a\rangle \otimes |b\rangle) || c\rangle \otimes |d\rangle\rangle &= \langle A\,|a\rangle \otimes B\,|b\rangle || c\rangle \otimes |d\rangle\rangle && (2.42)\\
&= \langle A\,|a\rangle || c\rangle\rangle \langle B\,|b\rangle || d\rangle\rangle && (2.43)\\
&= \left\langle |a\rangle \middle| A^\dagger\,|c\rangle \right\rangle \left\langle |b\rangle \middle| B^\dagger\,|d\rangle \right\rangle && (2.44)\\
&= \left\langle |a\rangle \otimes |b\rangle \middle| A^\dagger\,|c\rangle \otimes B^\dagger\,|d\rangle \right\rangle && (2.45)\\
&= \left\langle |a\rangle \otimes |b\rangle \middle| \left(A^\dagger \otimes B^\dagger\right)(|c\rangle \otimes |d\rangle) \right\rangle . && (2.46)
\end{aligned}$$

Hence

$$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger. \tag{2.47}$$

Given $|a, b\rangle \in \mathcal{H}$, we will often use an operator on part of $\mathcal{H}$ as if it were an operator on $\mathcal{H}$. For example, $A\,|a, b\rangle$ is taken to mean $(A \otimes I)\,|a, b\rangle$. This can be dangerous: we also write expressions like $(A + B)\,|a, b\rangle$, which we intend to mean $A\,|a\rangle \otimes B\,|b\rangle$, but this conflicts with using AB to do the same thing. Additionally, using the product notation makes $BA = (I \otimes B)(A \otimes I) = AB$ so that the "commutator" $[A, B] = 0$.

How might a density operator ρ for the composite system admit a **reduced density operator** ρ_A for system A? Consider an observable O of system A, for which the corresponding composite observable is $O \otimes I$. Regardless of the representation (composite or subsystem), the ensemble average of O should be the same:

$$\langle O_A \otimes I\rangle_\rho = \langle O_A\rangle_{\rho_A} . \tag{2.48}$$

To see what ρ_A is, take an orthonormal complete basis $\{A_j\}$ of Hermitian operators for $\mathscr{L}(\mathscr{H}_A)$ and $\{B_k\}$ for $\mathscr{L}(\mathscr{H}_B)$, so the density operator may be expressed as

$$\rho = \sum_{jk} A_j \otimes B_k \langle A_j \otimes B_k | \rho \rangle . \tag{2.49}$$

We may then compute that

$$\begin{aligned}
\rho_A &= \sum_i A_i \langle A_i | \rho_A \rangle && (2.50)\\
&= \sum_i A_i \langle A_i \otimes I | \rho \rangle \quad \text{eq. (2.48)} && (2.51)\\
&= \sum_i A_i \left\langle A_i \otimes I \middle| \textstyle\sum_{jk} A_j \otimes B_k \langle A_j \otimes B_k | \rho \rangle \right\rangle \quad \text{eq. (2.49)} && (2.52)\\
&= \sum_{ijk} A_i \langle A_i \otimes I | A_j \otimes B_k \rangle \langle A_j \otimes B_k | \rho \rangle && (2.53)\\
&= \sum_{ijk} A_i \langle A_i | A_j \rangle \langle B_k | I \rangle \langle A_j \otimes B_k | \rho \rangle && (2.54)\\
&= \sum_{ijk} A_i \delta_{ij} \operatorname{tr} B_k \langle A_j \otimes B_k | \rho \rangle && (2.55)\\
&= \sum_{jk} A_j \operatorname{tr} B_k \langle A_j \otimes B_k | \rho \rangle . && (2.56)
\end{aligned}$$

Definition 2. At this point, it makes sense to define the **partial trace** by

$$\operatorname{tr}_B (A \otimes B) = A \operatorname{tr} B \tag{2.57}$$

and extending linearly.

With this definition, we continue from eq. (2.56) to find that

$$\begin{aligned}
\rho_A &= \sum_{jk} \operatorname{tr}_B (A_j \otimes B_k) \langle A_j \otimes B_k | \rho \rangle && (2.58)\\
&= \operatorname{tr}_B \left(\textstyle\sum_{jk} A_j \otimes B_k \langle A_j \otimes B_k | \rho \rangle \right) && (2.59)\\
&= \operatorname{tr}_B \rho . && (2.60)
\end{aligned}$$

Thus the reduced density matrix for system A is obtained from the full density matrix by taking the partial trace over system B.

2.4 Entropy

Unitarys cool by [6].

However, the reduction of the density operator by "tracing out" B comes at the cost of losing information about the correlation between A and B. Quantitatively, the **relative entropy** between the correlated and uncorrelated density operators is

$$S(\rho \| \rho_A \otimes \rho_B) \tag{2.61}$$
$$\equiv \operatorname{tr} \rho(\ln \rho - \ln(\rho_A \otimes \rho_B)) \tag{2.62}$$
$$= \operatorname{tr}(\rho \ln \rho) - \operatorname{tr}_A \operatorname{tr}_B (\rho \ln(\rho_A \otimes I)) - \operatorname{tr}_B \operatorname{tr}_A (\rho \ln(I \otimes \rho_B)) \tag{2.63}$$
$$= S(\rho_A) + S(\rho_B) - S(\rho). \tag{2.64}$$

Together with the Klein inequality which states that relative entropies are non-negative (theorem 2.11.1), we have that

$$S(\rho) \leq S(\rho_A) + S(\rho_B), \tag{2.65}$$

with equality when $\rho = \rho_A \otimes \rho_B$. Other expected properties hold, such as that the relative entropy is invariant under unitary transformations (theorem 2.11.2), or that the relative entropy between subsystems is less than that between combined systems (theorem 2.11.3).

We would like to know the **reduced dynamics** of the quantum system S when in contact with an **environment** system B. We suppose that the composite system has a Hamiltonian of the form

$$H_{SB}(t) = H_S \otimes I + I \otimes H_B + H_I(t) \tag{2.66}$$

and that the environment is in equilibrium, so the composite density operator is $\rho(t) = \rho_S(t) \otimes \rho_B$. In terms of the unitary time-evolution operator $U(t)$ for the system, eq. (2.18) becomes $\rho(t) = U(t)\rho(0)U^\dagger(t)$. Taking the partial trace over the environment gives the time-evolved system density operator

$$\rho_S(t) = \operatorname{tr}_B \left(U(t)(\rho_S(0) \otimes \rho_B)U^\dagger(t) \right). \tag{2.67}$$

Whatever eq. (2.67) evaluates to, it will be an example of a **dynamical map** $\mathcal{V}(t)$ that time-evolves the system according to $\rho_S(t) = \mathcal{V}(t)\rho_S(0)$. While $\mathcal{V}(t)$ seems abstract, we know it should output a density operator. Then for $\mathcal{V}(t)$ to be a valid map on system density operators, it should *preserve the trace* of the input density operator. In fact, as a valid map $\mathcal{V}(t) \otimes I$ on the composite system, the composite density operator should remain positive. This property of $\mathcal{V}(t)$ is called **complete positivity**. Thus the valid maps on system density operators are **completely positive and trace-preserving** (CPTP).

2.5 The Lindblad equation

With the idea of random interactions with an environment in mind, we will assume that the maps $\{\mathcal{V}(t) : t \geq 0\}$ are also *memoryless* or **Markovian**, so that they form a **quantum dynamical semigroup** satisfying

$$\mathcal{V}(t_1)\mathcal{V}(t_2) = \mathcal{V}(t_1 + t_2) \quad \text{for} \quad t_1, t_2 \geq 0. \tag{2.68}$$

The action of the dynamical semigroup on the system describes an irreversible process. As such, the relative entropy between an arbitrary system ensemble $\rho(t)$ and an equilibrium ensemble ρ_0 cannot decrease (by eq. (2.67) and theorems 2.11.2 and 2.11.3):

$$\begin{aligned} S(\mathcal{V}(t)\rho \| \mathcal{V}(t)\rho_0) &= S\Big(\mathrm{tr}_B\left[U(t)(\rho \otimes \rho_B)U^\dagger(t)\right] \| \rho_0\Big) && (2.69)\\ &\leq S\Big(U(t)(\rho \otimes \rho_B)U^\dagger(t) \| \rho_0 \otimes \rho_B\Big) && (2.70)\\ &= S(\rho \otimes \rho_B \| \rho_0 \otimes \rho_B) && (2.71)\\ &= S(\rho \| \rho_0). && (2.72) \end{aligned}$$

We would like to determine the **infinitesimal generator** $\mathscr{L}$ for the quantum dynamical semigroup which allows the dynamical maps to be expressed as $\mathcal{V}(t) = e^{\mathscr{L}t}$, analogously to how a time-independent Hamiltonian is a generator for the unitary time-evolution operator $e^{Ht/\mathrm{i}\hbar}$. Following this analogy, the Schrödinger equation is replaced by the **Markovian quantum master equation** $\dot{\rho}_S = \mathscr{L}\rho_S$, which generalizes eq. (2.18) to typically non-unitary CTCP maps of density operators, provided that they are Markovian.

We will find that the most general form of $\mathscr{L}$ is given by the **Lindblad equation** eq. (2.84). To obtain this result, first consider diagonalizing ρ_B as $\rho_B = \sum_j \lambda_j |\phi_j\rangle\langle\phi_j|$ with orthonormal vectors $\phi_j \in \mathcal{H}_B$, where $\sum_j \lambda_j = 1$. Then eq. (2.67) becomes (writing ρ_S as ρ)

$$\begin{aligned} \rho(t) &= \sum_{ij} \langle\phi_i|U(t)\big(\rho(0) \otimes \lambda_j |\phi_j\rangle\langle\phi_j|\big)U^\dagger(t)|\phi_i\rangle && (2.73)\\ &= \sum_{ij} \lambda_j \langle\phi_i|U(t)|\phi_j\rangle \rho(0) \langle\phi_j|U^\dagger(t)|\phi_i\rangle && (2.74)\\ &= \sum_{ij} M_{ij}(t)\rho(0)M_{ij}^\dagger(t), && (2.75) \end{aligned}$$

where $M_{ij}(t) \equiv \sqrt{\lambda_j}\langle\phi_i|U(t)|\phi_j\rangle$. This decomposition in terms of the M_{ij} is an instance of the Choi-Kraus representation theorem (theorem 2.11.4). We can express the M_{ij} in

terms of an orthonormal complete basis $\{F_n\}$ for $\mathscr{L}(\mathscr{H}_S)$ as $M_{ij} = \sum_k F_k \langle F_k | M_{ij} \rangle$. Then eq. (2.75) becomes

$$\rho(t) = \sum_{mn} c_{mn}(t) F_m \rho(0) F_n^\dagger, \tag{2.76}$$

where

$$c_{mn}(t) \equiv \sum_{ij} \langle F_m | M_{ij}(t) \rangle \langle M_{ij}(t) | F_n \rangle . \tag{2.77}$$

For convenience, we may choose $F_{d^2} = I/\sqrt{d}$, where $d = \dim(\mathscr{H}_S)$. With an eye towards simplifying eq. (2.81), we eliminate the explicit time dependence of eq. (2.77) by defining

$$a_{mn} \equiv \lim_{t \to 0^+} \frac{c_{mn}(t) - d\delta_{d^2 d^2}}{t} \tag{2.78}$$

and introduce the sum of Kraus operators

$$F = \frac{1}{\sqrt{d}} \sum_{n=1}^{d^2-1} a_{nd^2} F_n \tag{2.79}$$

$$= \frac{F + F^\dagger}{2} + \mathrm{i} \frac{F - F^\dagger}{2\mathrm{i}} \equiv G + H/\mathrm{i}\hbar, \tag{2.80}$$

where we have decomposed the sum F into Hermitian and anti-Hermitian parts and included $\hbar$ so that H will have dimensions of energy. Now we may write the master equation $\mathscr{L}\rho = \dot{\rho}$ as

$$\begin{aligned}
\dot{\rho} &= \lim_{\Delta t \to 0^+} \frac{\mathscr{V}(\Delta t)\rho - \rho}{\Delta t} \\
&= \lim_{\Delta t \to 0^+} \left(\frac{c_{d^2 d^2} - d}{d \Delta t} \rho + \sum_{m,n=1}^{d^2-1} \frac{c_{mn}(\Delta t)}{\Delta t} F_m \rho F_n^\dagger \right. \\
&\left. + \frac{1}{\sqrt{d}} \sum_{n=1}^{d^2-1} \left(\frac{c_{nd^2}(\Delta t)}{\Delta t} F_n \rho + \frac{c_{d^2 n}(\Delta t)}{\Delta t} \rho F_n^\dagger \right) \right) \qquad (2.81) \\
&= \frac{a_{d^2 d^2}}{d} \rho + F\rho + \rho F^\dagger + \sum_{m,n=1}^{d^2-1} a_{mn} F_m \rho F_n^\dagger \\
&= \frac{a_{d^2 d^2}}{d} \rho + \{G, \rho\} + \frac{[H, \rho]}{\mathrm{i}\hbar} + \sum_{m,n=1}^{d^2-1} a_{mn} F_m \rho F_n^\dagger \\
&= \{G', \rho\} + \frac{[H, \rho]}{\mathrm{i}\hbar} + \sum_{m,n=1}^{d^2-1} a_{mn} F_m \rho F_n^\dagger, \qquad (2.82)
\end{aligned}$$

where $G' = G + a_{d^2d^2}I/d$. Since $\mathscr{V}(t)$ is trace-preserving, $\operatorname{tr}\dot{\rho} = 0$. Applying this condition to eq. (2.82) and cycling the trace gives

$$0 = \operatorname{tr}\left(2G'\rho + \sum_{m,n=1}^{d^2-1} a_{mn}F_n^\dagger F_m\rho\right),$$

so $G' = -\sum_{m,n=1}^{d^2-1} a_{mn}F_n^\dagger F_m/2$. This allows us to write eq. (2.82) as

$$\dot{\rho} = \frac{[H,\rho]}{\mathrm{i}\hbar} + \sum_{m,n=1}^{d^2-1} a_{mn}\left(F_m\rho F_n^\dagger - \frac{1}{2}\left\{F_n^\dagger F_m, \rho\right\}\right), \tag{2.83}$$

which is the first form of the *Lindblad equation*. This may be simplified further if we diagonalize the coefficient matrix a by applying a unitary transformation u to give $a = u\gamma u^\dagger$, where the $\{\gamma_k\}_{k=1}^{d^2-1}$ are the non-negative eigenvalues of a. This is possible since the coefficient matrix c is seen from eq. (2.77) to be Hermitian, and eq. (2.78) then gives that a is Hermitian. We may then express $F_{n\neq d^2} = \sum_{k=1}^{d^2-1} L_n u_{nk}$ in terms of the **Lindblad operators** L_n to find

$$\dot{\rho} = \frac{[H,\rho]}{\mathrm{i}\hbar} + \sum_{k=1}^{d^2-1} \gamma_k\left(L_k\rho L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho\right\}\right) \equiv \mathscr{L}\rho, \tag{2.84}$$

which is the *diagonal form* of the Lindblad equation. The eigenvalues γ_k have dimensions of inverse time and the Lindblad operators may be taken to be traceless. The second term is often called the **dissipator** $\mathscr{D}$ (see section 2.6), so the Lindbladian may be separated into unitary and non-unitary parts.

2.6 The weak-coupling limit

Now that we have found the general form for a stochastic CTCP generator, we must now determine the conditions for interaction Hamiltonian in eq. (2.66) to give rise to Markovian dynamics. While there are several different regimes where this is true, we will consider the **weak-coupling** limit which we justify by supposing that the environment is similar to a **harmonic bath** of many harmonic oscillators.

We start by expressing the interaction Hamiltonian in terms of Hermitian operators as

$$H_I = \sum_\alpha A_\alpha \otimes B_\alpha.$$

We suppose that the system in isolation would have *discrete* energy levels, so the eigenoperators of the superoperator $\mathcal{S} = [H_S, -]$ form a complete basis for $\mathcal{L}(\mathcal{H}_S)$. We then may write $A_\alpha = \sum_\omega A_{\alpha\omega}$, where

$$[H_S, A_{\alpha\omega}] = -\omega A_{\alpha\omega}. \tag{2.85}$$

Using eq. (2.85) to commute past the exponential in eq. (2.21a) gives $A'_{\alpha\omega} = e^{-\mathrm{i}\omega t} A_{\alpha\omega}$ in the interaction picture. Thus the interaction Hamiltonian in the interaction picture is

$$H'_I = \sum_{\alpha\omega} e^{-\mathrm{i}\omega t} A_{\alpha\omega} \otimes B'_\alpha, \tag{2.86}$$

where $B'_\alpha(t) = e^{-H_B t/\mathrm{i}\hbar} B_\alpha e^{H_B t/\mathrm{i}\hbar}$ per eq. (2.21a).

Since we are interested in how fluctuations in different environment modes are related, we will consider the **reservoir correlation functions**

$$\langle B^\dagger_\alpha(t) B_\beta(t-s)\rangle_{\rho_B} \tag{2.87}$$

and their one-sided Fourier transform

$$\Gamma_{\alpha\beta}(\omega) \equiv \int_0^\infty \mathrm{d}s\, e^{\mathrm{i}\omega s} \left\langle B^\dagger_\alpha(t) B_\beta(t-s)\right\rangle_{\rho_B} \tag{2.88}$$

$$\equiv \mathrm{i} S_{\alpha\beta}(\omega) + \gamma_{\alpha\beta}(\omega)/2, \tag{2.89}$$

where the corresponding matrix $S = (\Gamma - \Gamma^\dagger)/2\mathrm{i}$ is Hermitian and the matrix corresponding to the full Fourier transform

$$\gamma_{\alpha\beta}(\omega) \equiv \int_{-\infty}^\infty \mathrm{d}s\, e^{\mathrm{i}\omega s} \left\langle B^\dagger_\alpha(t) B_\beta(t-s)\right\rangle_{\rho_B} \tag{2.90}$$

is positive.

With this setup, we may now move to the main derivation. It is helpful to consider the interaction picture time evolution eq. (2.22) in the integral form

$$\rho(t) = \rho(0) - \mathrm{i}\int_0^t \mathrm{d}s\, [H_I(s), \rho(s)].$$

Applying eq. (2.22) again and tracing out the environment gives the closed equation

$$\dot\rho_S(t) = -\int_0^t \mathrm{d}s\, \mathrm{tr}_B\, [H_I(t), [H_I(s), \rho_S(s)\otimes\rho_B]]$$

for the system density operator. In doing so we have made two assumptions: that

$$\mathrm{tr}_B\, [H_I(t), \rho(0)] = 0,$$

which is the **weak-coupling approximation**, and that

$$\rho(t) = \rho_S(t) \otimes \rho_B,$$

which is the **Born approximation**. It should be noted that weak-coupling follows if the reservoir averages of the interactions vanish: $\langle B_\alpha(t)\rangle_{\rho_B} = 0$.

We now make the **Markov approximation** that $\rho_S(s) = \rho_S(t)$, so that the time-evolution only depends on the present time, to obtain the **Redfield equation**. To simplify further, we make the substitution $s \mapsto t - s$ and set the upper limit of the integral to infinity:

$$\dot{\rho}_S = -\int_0^\infty \mathrm{d}s\, \mathrm{tr}_B\,[H_I(t), [H_I(t-s), \rho_S(t) \otimes \rho_B]]. \tag{2.91}$$

This is justified when the reservoir correlation functions in eq. (2.88) vanish quickly over a time τ_B that is smaller than the relaxation time τ_R (see section 2.9). Substituting eq. (2.86) into eq. (2.91) and using eq. (2.88) gives

$$\dot{\rho}_S = 2\,\mathrm{He} \sum_{\alpha\beta\omega\omega'} e^{\mathrm{i}(\omega'-\omega)t} \Gamma_{\alpha\beta}(\omega)\Big(A_{\beta\omega}\rho_S A^\dagger_{\alpha\omega'} - A^\dagger_{\alpha\omega'} A_{\beta\omega}\rho_S\Big), \tag{2.92}$$

where $\mathrm{He}\,\Gamma \equiv (\Gamma + \Gamma^\dagger)/2$. If the typical times

$$\tau_S = |\omega' - \omega|^{-1} \quad \text{for} \quad \omega' \neq \omega$$

for system evolution are large compared to the relaxation time τ_R, then the contribution from the fast-oscillating terms of eq. (2.92) where $\omega' \neq \omega$ may be neglected. This **rotating wave** or **secular approximation** is analogous to how we consider the high-energy position distribution in the infinite square well to be uniform, even though it is actually a fast-oscillating function. By coarse-graining in this sense, we obtain

$$\dot{\rho}_S = 2\,\mathrm{He} \sum_{\alpha\beta\omega} \Gamma_{\alpha\beta}(\omega)\Big(A_{\beta\omega}\rho_S A^\dagger_{\alpha\omega} - A^\dagger_{\alpha\omega} A_{\beta\omega}\rho_S\Big). \tag{2.93}$$

Now applying the decomposition eq. (2.89) gives the interaction picture Lindblad equation

$$\dot{\rho}_S = \mathrm{i}[H_{LS}, \rho_S] + \mathcal{D}\rho_S, \tag{2.94}$$

where the **Lamb shift Hamiltonian** is

$$H_{LS} = \sum_{\alpha\beta\omega} S_{\alpha\beta}(\omega) A^\dagger_{\alpha\omega} A_{\beta\omega}, \tag{2.95}$$

and the *dissipator* is

$$\mathcal{D}\rho_S = \sum_{\alpha\beta\omega} \gamma_{\alpha\beta}\Big(A_{\beta\omega}\rho_S A_{\alpha\omega}^\dagger - \frac{1}{2}\big\{A_{\alpha\omega}^\dagger A_{\beta\omega}, \rho_S\big\}\Big). \tag{2.96}$$

The Lamb shift (or environment renormalization) Hamiltonian commutes with the system Hamiltonian since eq. (2.85) implies that $[H_S, A_{\alpha\omega}^\dagger A_{\beta\omega}] = 0$. Adding the system's Hamiltonian H_S to H_{LS} and diagonalizing gives the Schrödinger picture Lindblad equation eq. (2.84).

2.7 Relaxation to thermal equilibrium

The system will generally relax from its initial configuration to a stationary solution of eq. (2.84) (see section 2.9). We expect that the thermal state

$$\rho_S = \frac{e^{-\beta H_S}}{Z} \quad \text{where} \quad Z = \mathrm{tr}\big(e^{-\beta H_S}\big)$$

would be the equilibrium state. This is true when the reservoir correlation functions obey the KMS condition [17, 18]

$$\big\langle B_\alpha^\dagger(t) B_\beta(0)\big\rangle_{\rho_B} = \big\langle B_\beta(0) B_\alpha^\dagger(t + \mathrm{i}\beta)\big\rangle_{\rho_B}, \tag{2.97}$$

which is true when the environment is in the thermal state $\rho_B = e^{-\beta H_B}/\mathrm{tr}\big(e^{-\beta H_B}\big)$.

2.8 A two-level atom

To demonstrate the use of the Lindblad equation, we will study a model for the decay of a two-level atom. Our aim is to glimpse why electrons in atoms undergo optical decay, even though excited states are stable atomic states. Suppose that the atom has Hamiltonian $H_S = \hbar\omega\sigma_3/2$, where $\sigma_3 = |1\rangle\langle 1| - |0\rangle\langle 0|$. The operators $\sigma_- = |0\rangle\langle 1|$ and $\sigma_+ = |1\rangle\langle 0|$ are Lindblad operators, since they are eigenoperators of the superoperator $[H_S, -]$, like in eq. (2.85). These correspond to lowering and raising the energy by $\hbar\omega$, and will be our analogues of the emission and absorption processes. The derivation of section 2.5 is similar for a bath of photons in equilibrium, and our assumptions are justified because typical atomic relaxation times of about 20 ns are much slower than the periods of electromagnetic waves [19]. Ignoring the Lamb shift (which only offsets) and considering only the effects at ω, eq. (2.94) becomes

$$\dot\rho = \gamma_0(N+1)\Big(\sigma_-\rho\sigma_+ - \frac{1}{2}\{\sigma_+\sigma_-, \rho\}\Big)$$

$$+\gamma_0 N\left(\sigma_+\rho\sigma_- - \frac{1}{2}\{\sigma_-\sigma_+,\rho\}\right) \equiv \mathcal{D}\rho, \tag{2.98}$$

where $N = 1/(e^{\beta\hbar\omega}-1)$. This is straightforward to solve given the properties of the Pauli matrices. From the initial density operator $\rho(0) = |1\rangle\langle 1|$, we find that the population of upper level is

$$\rho_{11}(t) = \frac{N}{2N+1}\left(1 - e^{-\gamma t}\right), \quad \text{where} \quad \gamma = \gamma_0(2N+1).$$

This is consistent with what we observe in atomic spectra: an exponential decay to an equilibrium level which gives Lorentzian peaks. At low temperatures ($N \to 0$) the system approaches the ground state in accordance with the third law of thermodynamics, and at high temperatures ($N \gg 1$), the level is half-occupied and the absorption is saturated [20] (*Cf.* [4]).

2.9 Limitations

Though the Lindblad equation is widely applicable, there are some situations in which key assumptions in its derivation break down. For one, we have glossed over the issue of *ergodicity* in considering a harmonic bath. Since we needed to assume a discrete spectrum, the correlation functions eq. (2.87) will be quasi-periodic and will not decay as we required. It is only in the limit of a reservoir with infinitely many degrees of freedom that we expect non-periodic behavior like decay to emerge, but then there may be issues with unbounded operators. This can lead to the interesting behavior of spontaneous symmetry breaking and phase transitions.

There are also many systems with dynamics that occur on time scales comparable to the relaxation time. For example, a paper published less than a month ago (!) demonstrates how fast pulsed laser experiments can probe the relaxation of temporarily polarized gas molecules due to collisions [21, 22].

2.10 Conclusion

We have seen how the general consideration of Markovian CTCP maps on density operators leads to the Lindblad equation, and considered the weak-coupling limit as an example of a physical regime where the assumption of stochastic dynamics is valid. However,

[4] It's cool how this provides a fundamental explanation of the *Doppler-free saturated absorption spectroscopy of Rubidium vapor* JLAB experiment that I did.

we have only scratched the surface of what can be done with the Lindblad equation, especially with respect to solving it. Since the two-level atom is a small system, it is simple to diagonalize, but larger systems provide more difficulty as the dimension of the Hilbert space grows. Numerical solutions are complicated by the additional requirement of trace preservation, but they can still be done in many situations [15, 21]. The theory of open quantum systems gives some fundamental justifications for the assumptions of equilibrium statistical mechanics, as was briefly noted in section 2.7, and has made the picture of decoherence a bit more clear.

2.11 Mathematical details

Definition 3 (Tensor product)**.** Consider vector spaces $V(k)$, $W(k)$, and Z. For any bilinear map $h : V \times W \to Z$, the **tensor product** $V \otimes W$ and associated bilinear map $\phi : V \times W \to V \otimes W$ map have the property that there is a unique linear map $g : V \otimes W \to Z$ such that $h = g \circ \phi$. For tensor products of Hilbert spaces, the inner product is defined on each element of a product and then the space is completed. There is then a natural correspondence between the element $v \otimes f$ of the tensor product $V \otimes V^*$ and the linear map $T : V \to V$ defined by $Tx = f(x)v$.

This induces an extension of Dirac notation where all pairs $f \otimes x$ of dual and usual vectors from the same space are evaluated as $\langle f|x\rangle = f(x)$ and extended linearly. For example, given a linear operator $U : V \otimes W \to V \otimes W$ and a basis $|\phi_i\rangle$ for W, the partial trace over W may be expressed as $\mathrm{tr}_W U = \langle\phi_i|U|\phi_i\rangle$. This forms the justification of the step from eq. (2.73) to eq. (2.74) and of the manipulations in theorem 2.11.4.

Theorem 2.11.1 (Klein inequality)**.** *For density operators ρ and ρ', $S(\rho\|\rho') \geq 0$, with equality if and only if $\rho = \rho'$.*

Proof. The case for equality is trivial, so we will consider $\rho \neq \rho'$. Let $\mathcal{F}\rho = \rho \ln \rho$, so that we may express the relative entropy as

$$S(\rho\|\rho') = \mathrm{tr}(\mathcal{F}\rho - \mathcal{F}\rho' - \delta\mathcal{F}'\rho'),$$

where $\delta = \rho - \rho'$. We then have for $0 < t < 1$ that

$$\rho' + t\delta = t\rho + (1-t)\rho'.$$

Now let $f(t) = \mathrm{tr}\,(\mathcal{F}(\rho' + t\delta))$. Since the trace is monotonic and convex, f is convex and $f(t) \leq f(0) + t(f(1) - f(0))$. Rearranging and taking the limit as $t \to 0^+$ gives

$$f'(0) \leq f(1) - f(0),$$

which evaluates to

$$\operatorname{tr}(\delta\mathcal{F}'\rho') \leq \operatorname{tr}\mathcal{F}\rho - \operatorname{tr}\mathcal{F}\rho'. \qquad \square$$

Theorem 2.11.2. *For a unitary operator U and density operators ρ and ρ',*

$$S(U\rho U^\dagger \| U\rho' U^\dagger) = S(\rho\|\rho').$$

Proof. Since we may cycle the traces, it suffices to show that

$$\ln\left(U\rho U^\dagger\right) = \ln\rho.$$

This follows from Jacobi's formula for invertible matrices when applied to the logarithm that takes us from a Lie group to its corresponding Lie algebra, giving $\operatorname{tr}\circ\det = \operatorname{tr}\circ\log$. $\square$

Theorem 2.11.3. *For density operators ρ and ρ',*

$$S(\operatorname{tr}_B\rho\|\operatorname{tr}_B\rho') \leq S(\rho\|\rho'),$$

with equality if and only if ρ or ρ' is uncorrelated.

Theorem 2.11.4 (Choi-Kraus representation [15])**.** *A superoperator $\mathcal{S}$ on a density operator ρ is completely positive and trace-preserving if and only if it may be represented as*

$$\rho = \sum_{k=1}^{K} M_k\rho M_k^\dagger, \quad \textit{where} \quad \sum_{k=1}^{K} M_k M_k^\dagger = I.$$

2.12 The Born–Markov approximation for the Ising chain in a bath

The bath Hamiltonian is

$$H_B = \sum_{\boldsymbol{k},\lambda} \hbar\omega_{\boldsymbol{k}} a^\dagger_{\boldsymbol{k},\lambda} a_{\boldsymbol{k},\lambda}. \tag{2.99}$$

The vacuum energy $\sum_{\boldsymbol{k},\lambda}\hbar\omega_{\boldsymbol{k}}/2$ is dropped, since it diverges in the continuum limit.

The interaction Hamiltonian for spin-s objects in a magnetic field is[5]

$$H_I = -\int \mathrm{d}\boldsymbol{r}\ \boldsymbol{\mu}\cdot\boldsymbol{B} \tag{2.100}$$

[5]The time dependence of the field is absorbed into the operators $a_{\boldsymbol{k},\lambda}$, and the prefactor is chosen so that these operators are dimensionless, but $\boldsymbol{B}$ is not.

$$
\begin{aligned}
&= -\int \mathrm{d}\boldsymbol{r} \sum_i m_s g_s \mu_B \delta(\boldsymbol{r}_i)\boldsymbol{\sigma}_i \\
&\cdot \mathrm{i} \sum_{\boldsymbol{k},\lambda} \sqrt{\frac{\hbar c^2 \mu_0}{2V\omega_{\boldsymbol{k}}}}\Big((\boldsymbol{k}\times\boldsymbol{e}_{\boldsymbol{k},\lambda}) e^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}} a_{\boldsymbol{k},\lambda} - \big(\boldsymbol{k}\times\boldsymbol{e}^*_{\boldsymbol{k},\lambda}\big) e^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}} a^\dagger_{\boldsymbol{k},\lambda}\Big)
\end{aligned} \tag{2.101}
$$

$$
= -\sum_{i,\mu} m_s g_s \mu_B \sigma_i^\mu B_i^\mu, \tag{2.102}
$$

where we have defined the Hermitian operator

$$
B_i^\mu = \mathrm{i} \sum_{\boldsymbol{k},\lambda} \sqrt{\frac{\hbar c^2 \mu_0}{2V\omega_{\boldsymbol{k}}}}\Big(\big(\boldsymbol{k}\times\boldsymbol{e}_{\boldsymbol{k},\lambda}\big)_\mu e^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}_i} a_{\boldsymbol{k},\lambda} - \big(\boldsymbol{k}\times\boldsymbol{e}^*_{\boldsymbol{k},\lambda}\big)_\mu e^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}_i} a^\dagger_{\boldsymbol{k},\lambda}\Big). \tag{2.103}
$$

In the interaction picture:

$$
B_i^\mu(t) = e^{\mathrm{i}H_B t/\hbar} B_i^\mu e^{-\mathrm{i}H_B t/\hbar} \tag{2.104}
$$

$$
= \mathrm{i} \sum_{\boldsymbol{k},\lambda} \sqrt{\frac{\hbar c^2 \mu_0}{2V\omega_{\boldsymbol{k}}}}\Big(\big(\boldsymbol{k}\times\boldsymbol{e}_{\boldsymbol{k},\lambda}\big)_\mu e^{\mathrm{i}(\boldsymbol{k}\cdot\boldsymbol{r}_i-\omega_{\boldsymbol{k}}t)} a_{\boldsymbol{k},\lambda} - \big(\boldsymbol{k}\times\boldsymbol{e}^*_{\boldsymbol{k},\lambda}\big)_\mu e^{-\mathrm{i}(\boldsymbol{k}\cdot\boldsymbol{r}_i-\omega_{\boldsymbol{k}}t)} a^\dagger_{\boldsymbol{k},\lambda}\Big). \tag{2.105}
$$

The spectral correlation tensor is then

$$
\Gamma_{i\mu,j\nu}(\omega) = \frac{1}{\hbar^2}\int_0^\infty \mathrm{d}s\, e^{\mathrm{i}\omega s} \left\langle B_i^\mu(t)^\dagger B_j^\nu(t-s)\right\rangle \tag{2.106}
$$

$$
\begin{aligned}
&= -\frac{1}{\hbar^2}\frac{\hbar c^2\mu_0}{2V}\int_0^\infty \mathrm{d}s \sum_{\boldsymbol{k},\boldsymbol{k}',\lambda,\lambda'} \sqrt{\frac{1}{\omega_{\boldsymbol{k}}\omega_{\boldsymbol{k}'}}}\ : \\
&\big(\boldsymbol{k}\times\boldsymbol{e}_{\boldsymbol{k},\lambda}\big)_\mu \big(\boldsymbol{k}'\times\boldsymbol{e}_{\boldsymbol{k}',\lambda'}\big)_\nu e^{\mathrm{i}(\boldsymbol{k}\cdot\boldsymbol{r}_i-\omega_{\boldsymbol{k}}t+\boldsymbol{k}'\cdot\boldsymbol{r}_j-\omega_{\boldsymbol{k}'}(t-s)+\omega s)} \left\langle a_{\boldsymbol{k},\lambda} a_{\boldsymbol{k}',\lambda'}\right\rangle \\
&- \big(\boldsymbol{k}\times\boldsymbol{e}_{\boldsymbol{k},\lambda}\big)_\mu \big(\boldsymbol{k}'\times\boldsymbol{e}^*_{\boldsymbol{k}',\lambda'}\big)_\nu e^{\mathrm{i}(\boldsymbol{k}\cdot\boldsymbol{r}_i-\omega_{\boldsymbol{k}}t-\boldsymbol{k}'\cdot\boldsymbol{r}_j+\omega_{\boldsymbol{k}'}(t-s)+\omega s)} \left\langle a_{\boldsymbol{k},\lambda} a^\dagger_{\boldsymbol{k}',\lambda'}\right\rangle \\
&- \big(\boldsymbol{k}\times\boldsymbol{e}^*_{\boldsymbol{k},\lambda}\big)_\mu \big(\boldsymbol{k}'\times\boldsymbol{e}_{\boldsymbol{k}',\lambda'}\big)_\nu e^{-\mathrm{i}(\boldsymbol{k}\cdot\boldsymbol{r}_i-\omega_{\boldsymbol{k}}t-\boldsymbol{k}'\cdot\boldsymbol{r}_j+\omega_{\boldsymbol{k}'}(t-s)-\omega s)} \left\langle a^\dagger_{\boldsymbol{k},\lambda} a_{\boldsymbol{k}',\lambda'}\right\rangle \\
&+ \big(\boldsymbol{k}\times\boldsymbol{e}^*_{\boldsymbol{k},\lambda}\big)_\mu \big(\boldsymbol{k}\times\boldsymbol{e}^*_{\boldsymbol{k}',\lambda'}\big)_\nu e^{-\mathrm{i}(\boldsymbol{k}\cdot\boldsymbol{r}_i-\omega_{\boldsymbol{k}}t+\boldsymbol{k}'\cdot\boldsymbol{r}_j-\omega_{\boldsymbol{k}'}(t-s)-\omega s)} \left\langle a^\dagger_{\boldsymbol{k},\lambda} a^\dagger_{\boldsymbol{k}',\lambda'}\right\rangle.
\end{aligned} \tag{2.107}
$$

In the thermal state

$$
\rho_B = \frac{e^{-\beta H_B}}{\mathrm{tr}\, e^{-\beta H_B}} = \prod_{\boldsymbol{k},\lambda}\big(1-e^{-\beta\hbar\omega_{\boldsymbol{k}}}\big) e^{-\beta\hbar\omega_{\boldsymbol{k}} a^\dagger_{\boldsymbol{k},\lambda} a_{\boldsymbol{k},\lambda}} \tag{2.108}
$$

Since $\left[a_{\boldsymbol{k},\lambda}, a^\dagger_{\boldsymbol{k}',\lambda'}\right] = \delta_{\boldsymbol{k},\boldsymbol{k}'}\delta_{\lambda\lambda'}\boldsymbol{I}$,

$$
\left\langle a^\dagger_{\boldsymbol{k},\lambda} a_{\boldsymbol{k}',\lambda'}\right\rangle = \mathrm{tr}\big(e^{-\beta H_B}\big)^{-1}\, \mathrm{tr}\big(e^{-\beta H_B} a^\dagger_{\boldsymbol{k},\lambda} a_{\boldsymbol{k}',\lambda'}\big) \tag{2.109}
$$

$$= \mathrm{tr}\left(e^{-\beta H_B}\right)^{-1} \mathrm{tr}\left(e^{-\beta H_B} a_{\boldsymbol{k}',\lambda'} a^\dagger_{\boldsymbol{k},\lambda}\right) - \delta_{\boldsymbol{k},\boldsymbol{k}'}\delta_{\lambda\lambda'} \tag{2.110}$$

$$= \mathrm{tr}\left(e^{-\beta H_B}\right)^{-1} \mathrm{tr}\left(e^{\beta\hbar\omega_k} a_{\boldsymbol{k}',\lambda'} e^{-\beta H_B} a^\dagger_{\boldsymbol{k},\lambda}\right) - \delta_{\boldsymbol{k},\boldsymbol{k}'}\delta_{\lambda\lambda'} \tag{2.111}$$

$$= e^{\beta\hbar\omega_k} \left\langle a^\dagger_{\boldsymbol{k},\lambda} a_{\boldsymbol{k}',\lambda'} \right\rangle - \delta_{\boldsymbol{k},\boldsymbol{k}'}\delta_{\lambda\lambda'} \tag{2.112}$$

$$= \delta_{\boldsymbol{k},\boldsymbol{k}'}\delta_{\lambda\lambda'} n_B(\omega_{\boldsymbol{k}}). \tag{2.113}$$

Similarly,

$$\left\langle a_{\boldsymbol{k},\lambda} a_{\boldsymbol{k}',\lambda'} \right\rangle = \left\langle a^\dagger_{\boldsymbol{k},\lambda} a^\dagger_{\boldsymbol{k}',\lambda'} \right\rangle = 0 \tag{2.114}$$

$$\left\langle a_{\boldsymbol{k},\lambda} a^\dagger_{\boldsymbol{k}',\lambda'} \right\rangle = \delta_{\boldsymbol{k}\boldsymbol{k}'}\delta_{\lambda\lambda'}(1 + n_B(\omega_{\boldsymbol{k}})). \tag{2.115}$$

Then for a thermal bath, the spectral correlation tensor becomes

$$\begin{aligned} \Gamma_{i\mu,j\nu}(\omega) = \frac{c^2\mu_0}{2\hbar V} \int_0^\infty \mathrm{d}s \sum_{\boldsymbol{k},\lambda} \frac{1}{\omega_{\boldsymbol{k}}} \; : \\ \left(\boldsymbol{k}\times\boldsymbol{e}_{\boldsymbol{k},\lambda}\right)_\mu \left(\boldsymbol{k}\times\boldsymbol{e}^*_{\boldsymbol{k},\lambda}\right)_\nu e^{\mathrm{i}(\boldsymbol{k}\cdot(\boldsymbol{r}_i-\boldsymbol{r}_j)+s(\omega-\omega_{\boldsymbol{k}}))}(1+n_B(\omega_{\boldsymbol{k}})) \\ + \left(\boldsymbol{k}\times\boldsymbol{e}^*_{\boldsymbol{k},\lambda}\right)_\mu \left(\boldsymbol{k}\times\boldsymbol{e}_{\boldsymbol{k},\lambda}\right)_\nu e^{-\mathrm{i}(\boldsymbol{k}\cdot(\boldsymbol{r}_i-\boldsymbol{r}_j)-s(\omega+\omega_{\boldsymbol{k}}))} n_B(\omega_{\boldsymbol{k}}). \end{aligned} \tag{2.116}$$

$\boldsymbol{k}$ in exponent of original integral. Is there a reason why the external field $\boldsymbol{k}$ would be the same as a reciprocal lattice vector? Like optical phonon branch. Magnons? Just no absorption for off frequencies? Or do typical scales enforce limits?

To evaluate eq. (2.116), we now consider a chain of N spins along the z-axis, so that $\boldsymbol{r}_i = r_i\hat{\boldsymbol{z}}$.[6] Then $\boldsymbol{k}\cdot(\boldsymbol{r}_i - \boldsymbol{r}_j) = k_z \Delta r_{ij}$.

In the continuum limit,

$$\frac{1}{V}\sum_{\boldsymbol{k}} \mapsto \int \frac{\mathrm{d}\boldsymbol{k}}{(2\pi)^3} = \frac{1}{(2\pi c)^3} \int_0^\infty \mathrm{d}\omega_k \, \omega_k^2 \int \mathrm{d}\Omega\,, \tag{2.117}$$

where the integral over solid angle is

$$\int \mathrm{d}\Omega = \int \mathrm{d}\phi \int \mathrm{d}\theta \sin\theta. \tag{2.118}$$

To apply this limit to eq. (2.116), we first note that

$$\sum_\lambda \left(\boldsymbol{k}\times\boldsymbol{e}_{\boldsymbol{k},\lambda}\right)_\mu \left(\boldsymbol{k}\times\boldsymbol{e}^*_{\boldsymbol{k},\lambda}\right)_\nu = \sum_{abcd} \varepsilon_{\mu ab}\varepsilon_{\nu cd} k^a k^c \sum_\lambda e^b_{\boldsymbol{k},\lambda} \left(e^d_{\boldsymbol{k},\lambda}\right)^* \tag{2.119}$$

[6] We could consider any axis given the spherical symmetry, but the z-axis is the simplest to evaluate.

$$= \sum_{abcd} \varepsilon_{\mu ab}\varepsilon_{\nu cd} k^a k^c \left(\delta_{bd} - \frac{k^b k^d}{k^2}\right) \tag{2.120}$$

$$= \sum_{abc} \varepsilon_{\mu ab}\varepsilon_{\nu cb} k^a k^c \tag{2.121}$$

$$= \sum_{ac} \left(\delta_{\mu\nu}\delta_{ac} - \delta_{\mu c}\delta_{a\nu}\right) k^a k^c \tag{2.122}$$

$$= k^2 \delta_{\mu\nu} - k^\mu k^\nu. \tag{2.123}$$

Thus

$$\int \mathrm{d}\Omega\, e^{\pm \mathrm{i} k_z \Delta r_{ij}} \sum_\lambda \left(\boldsymbol{k} \times \boldsymbol{e}_{\boldsymbol{k},\lambda}\right)_\mu \left(\boldsymbol{k} \times \boldsymbol{e}^*_{\boldsymbol{k},\lambda}\right)_\nu = \frac{8\pi\omega_k^2}{3c^2} \delta_{\mu\nu} \Omega_{\mu\nu}\left(\frac{\omega_k \Delta r_{ij}}{c}\right), \tag{2.124}$$

where

$$\Omega_{\mu\nu}(u) = \left(\delta_{\nu z} - \frac{\delta_{\nu x} + \delta_{\nu y}}{2}\right) \frac{\operatorname{sinc} u - \cos u}{u^2} + \frac{\delta_{\nu x} + \delta_{\nu y}}{2} \operatorname{sinc} u. \tag{2.125}$$

Now eq. (2.124) gives that the continuum limit of the spectral correlation tensor for the spin chain is

$$\begin{aligned} \Gamma_{i\mu,j\nu}(\omega) = \delta_{\mu\nu} \frac{\mu_0}{6\pi^2 \hbar c^3} \int_0^\infty \mathrm{d}\omega_k\, \omega_k^3 \Omega_{\mu\nu}\left(\frac{\omega_k \Delta r_{ij}}{c}\right) : \\ (1 + n_B(\omega_k)) \int_0^\infty \mathrm{d}s\, e^{\mathrm{i}s(\omega - \omega_k)} + n_B(\omega_k) \int_0^\infty \mathrm{d}s\, e^{\mathrm{i}s(\omega + \omega_k)}. \end{aligned} \tag{2.126}$$

We now use that

$$n_B(-\omega) = -(1 + n_B(\omega)) \tag{2.127}$$

and

$$\int_0^\infty \mathrm{d}s\, e^{-\mathrm{i}\omega s} = \pi\delta(\omega) - \mathrm{i}\mathcal{P}\frac{1}{\omega}, \tag{2.128}$$

where $\mathcal{P}$ denotes the Cauchy principal value, to find

$$\Gamma_{i\mu,j\nu}(\omega) = \frac{1}{2}\gamma_{i\mu,j\nu}(\omega) + \mathrm{i}S_{i\mu,j\nu}(\omega), \tag{2.129}$$

where

$$\gamma_{i\mu,j\nu}(\omega) = \delta_{\mu\nu} \frac{\mu_0 \omega^3}{3\pi\hbar c^3} \Omega_{\mu\nu}\left(\frac{|\omega| \Delta r_{ij}}{c}\right)(1 + n_B(\omega)) \tag{2.130}$$

$$S_{i\mu,j\nu}(\omega) = \delta_{\mu\nu} \frac{\mu_0}{6\pi^2 \hbar c^3} \mathcal{P} \int_0^\infty \mathrm{d}\omega_k\, \omega_k^3 \Omega_{\mu\nu}\left(\frac{\omega_k \Delta r_{ij}}{c}\right)\left(\frac{1 + n_B(\omega_k)}{\omega - \omega_k} + \frac{n_B(\omega_k)}{\omega + \omega_k}\right). \tag{2.131}$$

Should the n_B and $1 + n_B$ above be flipped? Otherwise I don't think the principal value integral is finite as $\omega_k \to \infty$. However, the divergent form is also in [14, p. 145].

2.12.1 Old

We consider the Hamiltonian (in natural units)

$$H = \left(-J\sum_i \sigma_{zi}\sigma_{z(i+1)} - h\sum_i \sigma_{xi}\right) + \sum_{ik}\omega_{ik}a_{ik}^\dagger a_{ik} + \sum_{ik} C_k\left(a_{ik}^\dagger + a_{ik}\right)\sigma_{zi} \tag{2.132}$$

$$\equiv H_S \otimes I + I \otimes H_B + H_I. \tag{2.133}$$

The Schrodinger picture operators of the interaction Hamiltonian

$$H_I = \sum_i A_i \otimes B_i \tag{2.134}$$

are

$$A_i = \sigma_{zi} \tag{2.135}$$

$$B_i = \sum_k C_k\left(a_{ik}^\dagger + a_{ik}\right) \equiv \sum_k B_{ik}. \tag{2.136}$$

In the interaction picture,

$$A_i(t) = \sigma_{zi} \tag{2.137}$$

and

$$B_{ik}(t) = C_k\left(e^{\mathrm{i}\omega_{ik}t}a_{ik}^\dagger + e^{-\mathrm{i}\omega_{ik}t}a_{ik}\right). \tag{2.138}$$

Proof. Consider an observable A which satisfies

$$[H, A] = \omega A \tag{2.139}$$

for a Hamiltonian H. Such an **eigenoperator** of H has

$$H^n A = H^{n-1}AH + H^{n-1}[H, A] \tag{2.140}$$

$$= H^{n-1}A(H + \omega I) \tag{2.141}$$

$$= A(H + \omega I)^n. \tag{2.142}$$

Then

$$e^H A = \sum_{n\geqslant 0}\frac{H^n A}{n!} \tag{2.143}$$

$$= A\sum_{n\geqslant 0}\frac{(H + \omega I)^n}{n!} \tag{2.144}$$

$$= Ae^{H+\omega I} \tag{2.145}$$

$$= Ae^{\omega}e^{H}, \quad \text{BCH} \tag{2.146}$$

so the interaction picture operator is

$$A(t) = e^{\mathrm{i}Ht}Ae^{-\mathrm{i}Ht} \tag{2.147}$$

$$= e^{\mathrm{i}\omega t}Ae^{\mathrm{i}Ht}e^{-\mathrm{i}Ht} \tag{2.148}$$

$$= e^{\mathrm{i}\omega t}A. \tag{2.149}$$

In our case, $H = H_B$ and we have the eigenoperators

$$\left[H_B, a_{jl}\right] = \sum_{ik}\left[a_{ik}^{\dagger}a_{ik}, a_{jl}\right] \quad \text{definition of } H_B \tag{2.150}$$

$$= -\sum_{ik}\delta_{ij}\delta_{kl}a_{jl} \quad \text{commutation relations} \tag{2.151}$$

$$= -a_{jl}, \tag{2.152}$$

which follows from the commutation relations

$$\left[a^{\dagger}a, a\right] = a^{\dagger}aa - aa^{\dagger}a \tag{2.153}$$

$$= a^{\dagger}aa - a^{\dagger}aa - a\left[a, a^{\dagger}\right] \tag{2.154}$$

$$= -a \tag{2.155}$$

and

$$\left[a^{\dagger}a, a^{\dagger}\right] = a^{\dagger}aa^{\dagger} - a^{\dagger}a^{\dagger}a \tag{2.156}$$

$$= a^{\dagger}aa^{\dagger} - a^{\dagger}aa^{\dagger} - a^{\dagger}\left[a^{\dagger}, a\right] \tag{2.157}$$

$$= a^{\dagger}. \qquad \square$$

For the Born–Markov approximation to hold, we must verify that

1. $\mathrm{tr}_B\left[H_I(t), \rho_0\right] = 0$, or also that $0 = \langle B_{ik}(t)\rangle_{\rho_B^{(I)}(t)} = \langle B_{ik}\rangle_{\rho_B^{(S)}(t)}$.
2. $\rho(t) \approx \rho_S(t) \otimes \rho_B$ (see [14, p. 131]).
3. The reservoir correlation functions $\left\langle B_{ik}^{\dagger}(t)B_{jl}(t-s)\right\rangle_{\rho_B^{(I)}(t)}$ decay quickly over a time τ_B much less than the relaxation time τ_R.

Condition **1** is satisfied if the bath is in a thermal state, which we will also assume at $t = 0$ for condition **2**. The validity of **2** rests on **3**: we do not require that the bath is truly stationary, but only that it is approximately so on the coarser timescale of system evolution.

In the thermal state

$$\rho_{\text{th}} = \frac{e^{-\beta H_B}}{\operatorname{tr} e^{-\beta H_B}}, \tag{2.158}$$

the reservoir correlation functions are

$$\left\langle B_{ik}^\dagger(t) B_{jl}(t-s)\right\rangle = \left\langle B_{ik}^\dagger(s) B_{jl}(0)\right\rangle \tag{2.159}$$

$$= \delta_{ij}\delta_{kl} C_k C_l \left(e^{-\mathrm{i}\omega_{ik} s}(n_B(\omega_{ik})+1) + e^{\mathrm{i}\omega_{ik} s} n_B(\omega_{ik})\right). \tag{2.160}$$

Proof. We have that [14, p. 144]

$$\left\langle a_{ik} a_{jl}\right\rangle = 0 \tag{2.161}$$

$$\left\langle a_{ik}^\dagger a_{jl}^\dagger\right\rangle = 0 \tag{2.162}$$

$$\left\langle a_{ik}^\dagger a_{jl}\right\rangle = \delta_{ij}\delta_{kl}\, n_B(\omega_{ik}) \tag{2.163}$$

$$\left\langle a_{ik} a_{jl}^\dagger\right\rangle = \delta_{ij}\delta_{kl}(n_B(\omega_{ik})+1), \tag{2.164}$$

where

$$n_B(\omega) = \frac{1}{e^{\beta\omega}-1} \tag{2.165}$$

is the **Planck distribution**. We may then compute that

$$\left\langle B_{ik}^\dagger(t) B_{jl}(t-s)\right\rangle \tag{2.166}$$

$$= C_k C_l \left\langle \left(e^{-\mathrm{i}\omega_{ik} t} a_{ik} + e^{\mathrm{i}\omega_{ik} t} a_{ik}^\dagger\right)\left(e^{\mathrm{i}\omega_{jl}(t-s)} a_{jl}^\dagger + e^{-\mathrm{i}\omega_{jl}(t-s)} a_{jl}\right)\right\rangle \tag{2.167}$$

$$= C_k C_l \left(e^{-\mathrm{i}\omega_{ik} t + \mathrm{i}\omega_{jl}(t-s)} \left\langle a_{ik} a_{jl}^\dagger\right\rangle + e^{-\mathrm{i}\omega_{ik} t - \mathrm{i}\omega_{jl}(t-s)} \left\langle a_{ik} a_{jl}\right\rangle\right. \tag{2.168}$$

$$\left. + e^{\mathrm{i}\omega_{ik} t + \mathrm{i}\omega_{jl}(t-s)} \left\langle a_{ik}^\dagger a_{jl}^\dagger\right\rangle + e^{\mathrm{i}\omega_{ik} t - \mathrm{i}\omega_{jl}(t-s)} \left\langle a_{ik}^\dagger a_{jl}\right\rangle\right). \tag{2.169}$$

$$= \delta_{ij}\delta_{kl} C_k C_l \left(e^{-\mathrm{i}\omega_{ik} s}(n_B(\omega_{ik})+1) + e^{\mathrm{i}\omega_{ik} s} n_B(\omega_{ik})\right). \qquad \square$$

Thus the **spectral correlation tensor** is

$$\Gamma_{ij}(\omega) \equiv \sum_{kl} \int_0^\infty \mathrm{d}s\, e^{\mathrm{i}\omega s} \left\langle B_{ik}^\dagger(t) B_{jl}(t-s)\right\rangle_{\rho_B} \tag{2.170}$$

$$= \delta_{ij} \sum_k \int_0^\infty \mathrm{d}s\, C_k^2 \left(e^{\mathrm{i}(\omega-\omega_{ik})s}(n_B(\omega_{ik})+1) + e^{\mathrm{i}(\omega+\omega_{ik})s} n_B(\omega_{ik})\right) \tag{2.171}$$

$$= \delta_{ij} \sum_k C_k^2 \left(\pi\delta(\omega_{ik}-\omega) - \mathrm{i}\mathcal{P}\frac{1}{\omega_{ik}-\omega}\right)(n_B(\omega_{ik})+1) \tag{2.172}$$

$$+ \left(\pi\delta(-\omega_{ik}-\omega) + \mathrm{i}\mathcal{P}\frac{1}{\omega_{ik}+\omega}\right) n_B(\omega_{ik}) \tag{2.173}$$

$$\equiv \delta_{ij}\left(\frac{\gamma(\omega)}{2} + \mathrm{i}S(\omega)\right), \tag{2.174}$$

where we have used that

$$\int_0^\infty \mathrm{d}s\, e^{-\mathrm{i}\omega s} = \pi\delta(\omega) - \mathrm{i}\mathcal{P}\frac{1}{\omega}, \tag{2.175}$$

and $\mathcal{P}$ denotes the Cauchy principal value. We now take the continuum limit of a large ID cavity (with $\omega = ck$)[7]

$$\sum_k \mapsto \frac{L}{\pi}\int_0^\infty \mathrm{d}k = \frac{L}{\pi c}\int_0^\infty \mathrm{d}\omega \tag{2.176}$$

so that we have[8]

$$\gamma(\omega) = \frac{2L}{c}C(\omega)^2(n_B(\omega)+1) \tag{2.177}$$

and

$$S(\omega) = \mathcal{P}\int_0^\infty \mathrm{d}\omega_k\, \frac{L}{c}C(\omega_k)^2\left(\frac{n_B(\omega_{ik})+1}{\omega-\omega_{ik}} + \frac{n_B(\omega_{ik})}{\omega+\omega_{ik}}\right). \tag{2.178}$$

Since the only eigenvalue of $H_S{}^\angle$ is $\omega = 0$, we must require that

$$\gamma(0) = \lim_{\omega\to 0}\frac{2L}{c}C(\omega)^2(n_B(\omega)+1) \tag{2.179}$$

is finite. There are two relevant cases.

If $C(\omega)$ initially grows faster than $\sqrt{\omega}$, then $\gamma(0) = 0$ and the jump operators vanish, leaving just the Lamb shift. We then evaluate

$$S(0) = \mathcal{P}\int_0^\infty \mathrm{d}\omega_k\, \frac{L}{c}C(\omega_k)^2\left(\frac{n_B(\omega_{ik})+1}{-\omega_{ik}} + \frac{n_B(\omega_{ik})}{\omega_{ik}}\right) \tag{2.180}$$

$$= -\frac{L}{c}\mathcal{P}\int_0^\infty \mathrm{d}\omega_k\, \frac{C(\omega_k)^2}{\omega_k} \quad \text{let } \omega_{ik} := \omega_k \tag{2.181}$$

In this case, the exact form of $C(\omega_k)$ does not matter much. So long as it also goes to zero, $S(0)$ will take some constant negative value.

If instead, $C(\omega) \propto \sqrt{\omega}$, then $\gamma(0)$ is finite. A common choice makes the **spectral density**

$$J(\omega) \equiv \frac{2\alpha}{\pi}\int_0^\infty \mathrm{d}\omega_k\, C(\omega_k)^2\delta(\omega-\omega_k) \tag{2.182}$$

[7] We are assuming that there is only one mode per frequency per site in ID, rather than that there may be many modes for a given frequency, as usual in 3D.

[8] Note that the Planck distribution satisfies $n_B(\omega) + 1 = -n_B(-\omega)$.

$$= \frac{2\alpha}{\pi} C(\omega)^2 \tag{2.183}$$

be

$$J(\omega) = \frac{2\alpha}{\pi} \frac{\omega}{1 + (\omega/\Omega)^2}. \tag{2.184}$$

This is known as the **Ohmic spectral density** with cutoff frequency Ω, which gives rise to frequency-independent damping with a rate proportional to α [14, p. 175]. Then

$$S(0) = -\frac{2\alpha}{\pi} \mathcal{P} \int_0^\infty \frac{\mathrm{d}\omega_k}{1 + (\omega_k/\Omega)^2} \tag{2.185}$$

$$= -\frac{\Omega}{\alpha}. \tag{2.186}$$

Thus the Lamb shift Hamiltonian

$$H_{LS} = \sum_{ij} \delta_{ij} S(0) \sigma_{zi} \sigma_{zj} = -\frac{N\Omega}{\alpha} I$$

only shifts the energy of the chain. With an Ohmic bath, we find

$$\gamma(0) = \lim_{\omega \to 0} \frac{4\alpha L}{\pi c} \frac{\omega}{1 + (\omega/\Omega)^2} (n_B(\omega) + 1) \tag{2.187}$$

$$= \frac{2L}{c} \frac{1}{\beta}, \tag{2.188}$$

so the dissipator is

$$\mathcal{D}\rho_S = \frac{4L}{\pi c} \frac{\alpha}{\beta} \sum_i (\sigma_{zi} \rho_S \sigma_{zi} - \rho_S). \tag{2.189}$$

Thus neglecting the Lamb shift and $4L/\pi c$, we have that the system density matrix in the interaction picture obeys

$$\dot{\rho}_S(t) = \frac{\alpha}{\beta} \sum_i (\sigma_{zi} \rho_S(t) \sigma_{zi} - \rho_S(t)) \tag{2.190}$$

$$= \frac{\alpha}{\beta} \sum_i [\sigma_{zi}, \rho_S(t)] \sigma_{zi}. \tag{2.191}$$

The reduced density matrix entries for each site are then determined by the equations

$$\dot{\rho}_{00} = 0 \tag{2.192}$$

$$\dot{\rho}_{11} = 0 \tag{2.193}$$

$$\dot{\rho}_{01} = -\frac{2\alpha}{\beta} \rho_{01} \tag{2.194}$$

$$\dot{\rho}_{01} = -\frac{2\alpha}{\beta} \rho_{10}. \tag{2.195}$$

Proof. If $A|a\rangle = a|a\rangle$, then

$$A\angle\, |a\rangle\langle b| = (a-b)\,|a\rangle\langle b|\,,$$

where the superoperator $A\angle$ is defined by $A \angle B \equiv [A, B]$. □

2.13 Solving eigenoperator problems in coordinates

Given a complete basis A_i for $\mathscr{L}(\mathscr{H}_A)$ and the Hamiltonian

$$H = \sum_i h_i A_i, \tag{2.196}$$

we want to find eigenoperators

$$A = \sum_j a_j A_j, \tag{2.197}$$

of $H\angle$. If

$$\big[A_i, A_j\big] = \sum_k s_{ijk} A_k \tag{2.198}$$

$$s_{ijk} = \big\langle \big[A_i, A_j\big] | A_k \big\rangle\,, \tag{2.199}$$

then the eigenvalue equation is

$$[H, A] = \omega A \tag{2.200}$$

$$\sum_{ijk} h_i a_j s_{ijk} A_k = \sum_j \omega a_j A_j \tag{2.201}$$

$$\sum_{jk} S_{jk} a_j A_k = \sum_j \omega a_j A_j, \tag{2.202}$$

where the matrix $\boldsymbol{S}$ has coefficients

$$S_{jk} = \sum_i h_i s_{ijk} = \big\langle A_j | H\angle | A_k \big\rangle\,. \tag{2.203}$$

Thus eigenoperators may be found by solving the ordinary eigenvalue problem

$$\boldsymbol{S}^{\mathsf{T}} \boldsymbol{a} = \omega \boldsymbol{a}. \tag{2.204}$$

2.14 Superoperators in coordinates

Consider an orthonormal basis $|i\rangle$ for $\mathscr{H}$ and thus $\mathscr{L}$. This induces a basis for superoperators by pre and post-multiplication.

The superoperator matrix element for left-multiplication is

$$\langle ab|A_L|cd\rangle = \sum_{ef} \langle ab|A_{ef}|ef\rangle\,|cd\rangle \tag{2.205}$$

$$= \sum_{ef} \delta_{fc}\,\langle ab|A_{ef}|ed\rangle \tag{2.206}$$

$$= \sum_{ef} \delta_{fc}\delta_{ae}\delta_{bd}A_{ef} \tag{2.207}$$

$$= \delta_{db}A_{ac}, \tag{2.208}$$

while that for right-multiplication is

$$\langle ab|A_R|cd\rangle = \sum_{ef} \langle ab|A_{ef}|cd\rangle\,|ef\rangle \tag{2.209}$$

$$= \sum_{ef} \delta_{de}\,\langle ab|A_{ef}|cf\rangle \tag{2.210}$$

$$= \sum_{ef} \delta_{de}\delta_{ac}\delta_{bf}A_{ef} \tag{2.211}$$

$$= \delta_{ac}A_{db}. \tag{2.212}$$

Thus for both multiplications, we have that

$$\langle ab|A_LB_R|cd\rangle = \sum_{ef} \langle ab|A_L|ef\rangle\,\langle ef|B_R|cd\rangle \tag{2.213}$$

$$= \sum_{ef} \delta_{fb}\delta_{ec}A_{ae}B_{df} \tag{2.214}$$

$$= A_{ac}B_{db}. \tag{2.215}$$

In particular, if $B = A^\dagger$, this is $A_{ac}A^*_{bd}$, so that the trace of the superoperator is $\|A\|^2$.

Given A, we have the commutator superoperator $[A] = A_L - A_R$ and the anticommutator superoperator $\{A\} = A_L + A_R$. We then see that

$$\frac{1}{2}\,\langle ab|\{A^\dagger A\}|cd\rangle = \frac{1}{2}\sum_k \left(\delta_{db}A_{ak}A^*_{ck} + \delta_{ac}A_{dk}A^*_{bk}\right), \tag{2.216}$$

so the elements for the dissipator $\mathscr{D} = \sum_i \gamma_i\Big(A_LA_R^\dagger - \{A^\dagger A\}/2\Big)$ are

$$\langle ab|\mathscr{D}|cd\rangle = \sum_i \gamma_i\left(A_{ac}A^*_{bd} - \frac{1}{2}\sum_k \left(\delta_{db}A_{ak}A^*_{ck} + \delta_{ac}A_{dk}A^*_{bk}\right)\right) \tag{2.217}$$

2.15 Weak coupling separates unitary and dissipative evolution

$$\left[\left[A_i^\dagger A_i\right], \left\{A_j^\dagger A_j\right\}\right]\rho = 0 \tag{2.218}$$

$$\langle a|\left[A_\omega^\dagger A_\omega, B_\omega^\dagger B_\omega\right]|e\rangle \tag{2.219}$$

$$= A_{ab}A_{bc}B_{cd}B_{de}\,\omega_{ba}\omega_{bc}\omega_{dc}\omega_{de} - B_{ab}B_{bc}A_{cd}A_{de}\,\omega_{ba}\omega_{bc}\omega_{dc}\omega_{de} \tag{2.220}$$

$$= (A_{ab}A_{bc}B_{cd}B_{de} - B_{ab}B_{bc}A_{cd}A_{de})\omega_{ba}\omega_{bc}\omega_{dc}\omega_{de} \tag{2.221}$$

$$= 0 \tag{2.222}$$

$$\left[\left[A_i^\dagger A_i\right], (A_j)_L\left(A_j^\dagger\right)_R\right]\rho = 0 \tag{2.223}$$

$$\langle a|A^\dagger AB\rho B^\dagger - B\rho B^\dagger A^\dagger A - BA^\dagger A\rho B^\dagger + B\rho A^\dagger AB^\dagger|f\rangle \tag{2.224}$$

$$= A_{ab}A_{bc}B_{cd}\rho_{de}B_{ef}\,\omega_{ba}\omega_{bc}\omega_{cd}\omega_{fe} - B_{ab}\rho_{bc}B_{cd}A_{de}A_{ef}\,\omega_{ab}\omega_{dc}\omega_{ed}\omega_{ef} \tag{2.225}$$

$$- B_{ab}A_{bc}A_{cd}\rho_{de}B_{ef}\,\omega_{ab}\omega_{cb}\omega_{cd}\omega_{fe} + B_{ab}\rho_{bc}A_{cd}A_{de}B_{ef}\,\omega_{ab}\omega_{dc}\omega_{de}\omega_{fe} \tag{2.226}$$

$$= 0 \tag{2.227}$$

Chapter 3

The Ising model as an open quantum system

3.1 Solution of the transverse Ising model

We consider the Hamiltonian

$$H = -J \sum_{i \in \mathbb{Z}_N} S_i^x S_{i+1}^x - \Gamma \sum_{i \in \mathbb{Z}_N} S_i^z, \tag{3.1}$$

which we nondimensionalize as

$$\frac{4}{J} H = - \sum_{i \in \mathbb{Z}_N} \sigma_i^x \sigma_{i+1}^x + \lambda \sum_{i \in \mathbb{Z}_N} \sigma_i^z \tag{3.2}$$

Note that H does not have $4/J$ below. Fix this and change λ (Striff) to $-\alpha$.

for the periodic transverse Ising chain with N spins. We notice that the operators

$$\sigma_i^{\pm} = \frac{\sigma_i^x \pm \mathrm{i}\sigma_i^y}{2} \tag{3.3}$$

satisfy

$$\sigma_i^z = 2\sigma_i^+ \sigma_i^- - I \tag{3.4}$$

and have commutators

$$\left[\sigma_i^+, \sigma_j^-\right] = \frac{1}{4}\left[\sigma_i^x + \mathrm{i}\sigma_i^y, \sigma_j^x - \mathrm{i}\sigma_j^y\right] \tag{3.5}$$

$$= \frac{1}{4}\left(\left[\sigma_i^x, \sigma_j^x\right] + \left[\sigma_i^y, \sigma_j^y\right] + \mathrm{i}\left[\sigma_i^y, \sigma_j^x\right] - \mathrm{i}\left[\sigma_i^x, \sigma_j^y\right]\right) \tag{3.6}$$

$$= \delta_{ij}\sigma_i^z. \tag{3.7}$$

Thus their anticommutators are

$$\left\{\sigma_i^+, \sigma_j^-\right\} = 2\sigma_i^+\sigma_j^- - \left[\sigma_i^+, \sigma_j^-\right] \tag{3.8}$$

$$= 2\sigma_i^+\sigma_j^- - \delta_{ij}\sigma_i^z \tag{3.9}$$

$$= \delta_{ij}\boldsymbol{I} + 2\sigma_i^+\sigma_j^-(1-\delta_{ij}). \tag{3.10}$$

It could be helpful to think of the $\sigma_i^\pm$ as fermion creation and annihilation operators, but they do not anticommute at different sites.

How might we construct operators that satisfy the fermionic canonical anticommutation relations (CARs) from the Pauli operators? Suppose we have such operators c_i. Given a tuple $\boldsymbol{n} = (n_i)_{i\in\mathbb{Z}_N}$, we have the corresponding states

$$|\boldsymbol{n}\rangle = \prod_{i\in\mathbb{Z}_N}\left(c_i^\dagger\right)^{n_i}|\boldsymbol{0}\rangle, \tag{3.11}$$

where $|\boldsymbol{0}\rangle$ denotes the vacuum state. It then follows that

$$c_i|\boldsymbol{n}\rangle = -n_i(-1)^{n_{<i}}|\boldsymbol{n}_{i\leftarrow 0}\rangle \tag{3.12}$$

$$c_i^\dagger|\boldsymbol{n}\rangle = -(1-n_i)(-1)^{n_{<i}}|\boldsymbol{n}_{i\leftarrow 1}\rangle, \tag{3.13}$$

where $\boldsymbol{n}_{i\leftarrow m} = \boldsymbol{n}$ with $n_i = m$ and $n_{<i} = \sum_{j<i} n_j$.

Thus the number operator is

$$c_i^\dagger c_i|\boldsymbol{n}\rangle = (1-0)(-1)^{n_{<i}}n_i(-1)^{n_{<i}}|\boldsymbol{n}_{i\leftarrow 1}\rangle \tag{3.14}$$

$$= n_i|\boldsymbol{n}\rangle. \tag{3.15}$$

This leads us to consider

$$c_i = -\left(\prod_{j<i} -\sigma_j^z\right)\sigma_i^- \tag{3.16}$$

acting on the states

$$|\boldsymbol{n}\rangle = \prod_{i\in\mathbb{Z}_N}\left(\sigma_i^+\right)^{n_i}|\boldsymbol{0}\rangle, \tag{3.17}$$

Fix my sign errors.

where $|\mathbf{0}\rangle = |\uparrow\rangle^{\otimes N}$ is the state with all z-spins up, or all zero qubits. This gives the same result as eq. (3.12), so the c_i satisfy the CARs. This process of mapping spin-1/2 sites to non-local fermions is known as the **Jordan-Wigner transformation**. We may then compute that the inverse transformations are

$$\sigma_i^+\sigma_i^- = c_i^\dagger c_i \tag{3.18}$$

$$\sigma_i^z = 2c_i^\dagger c_i - I \tag{3.19}$$

$$\sigma_i^x = -\left(\prod_{j<i}\left(I - 2c_j^\dagger c_j\right)\right)\left(c_i^\dagger + c_i\right) \tag{3.20}$$

$$\sigma_i^y = \mathrm{i}\left(\prod_{j<i}\left(I - 2c_j^\dagger c_j\right)\right)\left(c_i^\dagger - c_i\right). \tag{3.21}$$

While σ_i^x remains complicated, the product $\sigma_i^x\sigma_{i+1}^x$ does not. For $i < N-1$,

$$\sigma_i^x\sigma_{i+1}^x = \left(\prod_{j<i}\left(2c_j^\dagger c_j - I\right)\right)\left(c_i^\dagger + c_i\right)\left(\prod_{j<i+1}\left(2c_j^\dagger c_j - I\right)\right)\left(c_{i+1}^\dagger + c_{i+1}\right) \tag{3.22}$$

$$= \left(c_i^\dagger + c_i\right)\left(I - 2c_i^\dagger c_i\right)\left(c_{i+1}^\dagger + c_{i+1}\right) \tag{3.23}$$

$$= \left(c_i^\dagger - c_i\right)\left(c_{i+1}^\dagger + c_{i+1}\right), \tag{3.24}$$

and for $i = N-1$,

$$\sigma_{N-1}^x\sigma_0^x = \left(\prod_{j<N-1}\left(2c_j^\dagger c_j - I\right)\right)\left(c_{N-1}^\dagger + c_{N-1}\right)\left(c_0^\dagger + c_0\right). \tag{3.25}$$

We may now perform the Jordan-Wigner transformation of eq. (3.2) to obtain

Check boundary term to Pfeuty (2.4). Why can it be neglected for large N?

$$H = \sum_i \left(c_i - c_i^\dagger\right)\left(c_{i+1}^\dagger + c_{i+1}\right) + \lambda\sum_i 2c_i^\dagger c_i - \lambda N I \tag{3.26}$$

$$-\left(I - \prod_{j<N-1}\left(2c_j^\dagger c_j - I\right)\right)\left(c_{N-1} - c_{N-1}^\dagger\right)\left(c_0^\dagger + c_0\right) \tag{3.27}$$

We now Fourier transform with

$$c_i = \frac{1}{\sqrt{N}}\sum_k e^{\mathrm{i}ki}C_k \tag{3.28a}$$

$$C_k = \frac{1}{\sqrt{N}} \sum_i e^{-\mathrm{i}ki} c_i \tag{3.28b}$$

and

$$c_i^\dagger = \frac{1}{\sqrt{N}} \sum_k e^{-\mathrm{i}ki} C_k^\dagger \tag{3.28c}$$

$$C_k^\dagger = \frac{1}{\sqrt{N}} \sum_i e^{\mathrm{i}ki} c_i^\dagger. \tag{3.28d}$$

Proof.

$$c_0 = \frac{1}{\sqrt{N}} \sum_k C_k \tag{3.29}$$

$$c_N = \frac{1}{\sqrt{N}} \sum_k e^{\mathrm{i}kN} C_k. \tag{3.30}$$

We then must require that

$$kN \equiv 0 \pmod{2\pi} \tag{3.31}$$

$$k = \frac{2\pi n}{N} - \frac{N - [N \text{ odd}]}{N}\pi, \quad n \in \mathbb{Z}_N. \tag{3.32}$$

For N odd, what is C_π?

$$C_\pi = \frac{1}{\sqrt{N}} \sum_i e^{-\mathrm{i}\pi i} c_i. \tag{3.33}$$

Since $e^{-\mathrm{i}\pi i} = e^{\mathrm{i}\pi i}$, $C_\pi = C_{-\pi}$. □

Proof. Consider N fermionic operators c_i and a $N \times N$ unitary matrix $\boldsymbol{U}$. We may change bases with

$$C_k^\dagger = \sum_i U_{ik} c_i^\dagger. \tag{3.34}$$

Then

$$\left\{C_k, C_{k'}^\dagger\right\} = \sum_{ij} U_{ik}^* U_{jk'} \left\{c_i, c_j^\dagger\right\} \tag{3.35}$$

$$= \sum_i U_{ik}^* U_{ik'} \tag{3.36}$$

$$= \left(\boldsymbol{U}^\dagger \boldsymbol{U}\right)_{kk'} \tag{3.37}$$

$$= \delta_{kk'}, \tag{3.38}$$

and similar for the other fermionic (anti)-commutation relations. □

Proof. For the Fourier transform,

$$F_{ik} = \frac{1}{\sqrt{N}} e^{\mathrm{i}ki}. \tag{3.39}$$

We may then confirm that

$$\left(F^\dagger F\right)_{kk'} = \sum_i \frac{1}{N} e^{\mathrm{i}(k'-k)i} \tag{3.40}$$

$$= \delta_{kk'}. \tag{3.41}$$

Thus the Fourier transform is unitary. □

Now since

$$\frac{1}{N} \sum_{i \in \mathbb{Z}_N} e^{\mathrm{i}(k'-k)i} = \delta_{kk'}, \tag{3.42}$$

and also

$$C_{-k} = C_k^* \tag{3.43}$$

$$= \frac{1}{\sqrt{N}} \sum_i e^{-\mathrm{i}(-k)i} c_i \tag{3.44}$$

$$= \frac{1}{N} \sum_{ik'} e^{\mathrm{i}(k'+k)i} C_{k'}, \tag{3.45}$$

we have that

$$\sum_i c_i^\dagger c_i = \frac{1}{N} \sum_{ikk'} e^{\mathrm{i}(k'-k)i} C_k^\dagger C_{k'} \tag{3.46}$$

$$= \sum_k C_k^\dagger C_k, \tag{3.47}$$

$$\sum_i \left(c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i\right) = \frac{1}{N} \sum_{ikk'} e^{\mathrm{i}(k'-k)i} \left(e^{\mathrm{i}k'} + e^{-\mathrm{i}k}\right) C_k^\dagger C_{k'} \tag{3.48}$$

$$= \sum_k 2\cos k \; C_k^\dagger C_k, \tag{3.49}$$

$$\sum_i \left(c_{i+1} c_i + c_i^\dagger c_{i+1}^\dagger\right) = \frac{1}{N} \sum_{ikk'} \left(e^{\mathrm{i}(k'+k)i} e^{\mathrm{i}k} C_k C_{k'} + e^{-\mathrm{i}(k'+k)i} e^{-\mathrm{i}k'} C_k^\dagger C_{k'}^\dagger\right) \tag{3.50}$$

$$= \sum_k \left(e^{-\mathrm{i}k} C_{-k} C_k + e^{\mathrm{i}k} C_k^\dagger C_{-k}^\dagger\right). \tag{3.51}$$

Thus eq. (3.27) is now

$$H = -\sum_k 2\cos k \; C_k^\dagger C_k + \sum_k \left(e^{-\mathrm{i}k} C_{-k} C_k + e^{\mathrm{i}k} C_k^\dagger C_{-k}^\dagger\right) + \sum_k 2\lambda C_k^\dagger C_k - \lambda N I \tag{3.52}$$

$$= \sum_k (\lambda - \cos k)\Big(C_k^\dagger C_k + C_{-k}^\dagger C_{-k}\Big) + \sum_k \mathrm{i}\sin k\Big(C_{-k}C_k - C_k^\dagger C_{-k}^\dagger\Big) - \lambda N I \tag{3.53}$$

$$= \sum_k (\lambda - \cos k)\Big(C_k^\dagger C_k - C_{-k}C_{-k}^\dagger\Big) + \sum_k \mathrm{i}\sin k\Big(C_{-k}C_k - C_k^\dagger C_{-k}^\dagger\Big) \tag{3.54}$$

$$= \sum_k \boldsymbol{v}_k^\dagger \boldsymbol{H}_k \boldsymbol{v}_k, \tag{3.55}$$

where

$$\boldsymbol{H}_k = \begin{bmatrix} \lambda - \cos k & -\mathrm{i}\sin k \\ \mathrm{i}\sin k & \cos k - \lambda \end{bmatrix}, \tag{3.56}$$

$$\boldsymbol{v}_k = \begin{bmatrix} C_k \\ C_{-k}^\dagger \end{bmatrix}, \tag{3.57}$$

and we have used that

$$\sum_k \cos k = 0. \tag{3.58}$$

Since the $\boldsymbol{H}_k$ are Hermitian, they may be diagonalized by a unitary transformation of the $\boldsymbol{v}_k$. The $\boldsymbol{H}_k$ are traceless, so they have the eigenvalues

$$E_k^\pm = \pm\sqrt{-\det \boldsymbol{H}_k} \tag{3.59}$$

$$= \pm\sqrt{\lambda^2 - 2\lambda\cos k + \cos^2 k + \sin^2 k} \tag{3.60}$$

$$= \pm\sqrt{\lambda^2 - 2\lambda\cos k + 1}. \tag{3.61}$$

The eigenvectors are then

$$\boldsymbol{q}_k^\pm = \begin{bmatrix} -\mathrm{i}\sin k \\ E_k^\pm - (\lambda - \cos k) \end{bmatrix}, \tag{3.62}$$

except if $k = 0$ or $-\pi$, in which case

$$\boldsymbol{q}_k^- = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{q}_k^+ = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.63}$$

The $k = -\pi$ case does not appear if N is odd. If also $\lambda = -1$, then $\boldsymbol{H}_{-\pi} = \boldsymbol{0}$. To construct the unitary transformation, we must normalize the $\boldsymbol{q}_k^\pm$. We find that

$$\left\|\boldsymbol{q}_k^\pm\right\|^2 = \left(E_k^\pm - (\lambda - \cos k)\right)^2 + \sin^2 k \tag{3.64}$$

$$= \left(E_k^\pm\right)^2 + \lambda^2 + \cos^2 k - 2\lambda\cos k - 2E_k^\pm(\lambda - \cos k) + 1 - \cos^2 k \tag{3.65}$$

$$= 2E_k^\pm\big(E_k^\pm - (\lambda - \cos k)\big). \tag{3.66}$$

Now

$$\frac{(\boldsymbol{q}_k^{\pm})_1}{\|\boldsymbol{q}_k^{\pm}\|} = \frac{-\mathrm{i}\sin k}{\sqrt{2E_k^{\pm}\left(E_k^{\pm} - (\lambda - \cos k)\right)}} \tag{3.67}$$

$$= \frac{-\mathrm{i}\sin k}{\sqrt{2|E_k^{\pm}|\left(|E_k^{\pm}| \mp (\lambda - \cos k)\right)}} \tag{3.68}$$

and

$$\frac{(\boldsymbol{q}_k^{\pm})_2}{\|\boldsymbol{q}_k^{\pm}\|} = \pm\sqrt{\frac{E_k^{\pm} - (\lambda - \cos k)}{2E_k^{\pm}}} \tag{3.69}$$

$$= \pm\sqrt{\frac{|E_k^{\pm}| \mp (\lambda - \cos k)}{2|E_k^{\pm}|}} \tag{3.70}$$

$$\boldsymbol{U}_k^{\dagger} = \begin{bmatrix} (\hat{\boldsymbol{q}}_k^{-})^{\dagger} \\ (\hat{\boldsymbol{q}}_k^{+})^{\dagger} \end{bmatrix}. \tag{3.71}$$

Then with $E_k = |E_k^{\pm}|$,

$$\eta_k^{\pm} = \frac{\mathrm{i}\sin k}{\sqrt{2E_k(E_k \pm (\cos k - \lambda))}} C_k \pm \sqrt{\frac{E_k \pm (\cos k - \lambda)}{2E_k}} C_{-k}^{\dagger} \tag{3.72}$$

so that

$$\left\{(\eta_k^{\pm})^{\dagger}, \eta_k^{\pm}\right\} = \frac{\sin^2 k}{2E_k(E_k \pm (\cos k - \lambda))} I + \frac{E_k \pm (\cos k - \lambda)}{2E_k} I \tag{3.73}$$

$$= I \tag{3.74}$$

$$\left\{(\eta_k^{\pm})^{\dagger}, \eta_k^{\mp}\right\} = \frac{\sin^2 k}{2E_k\sqrt{E_k \pm (\cos k - \lambda)}\sqrt{E_k \mp (\cos k - \lambda)}} I - \frac{\sqrt{E_k \pm (\cos k - \lambda)}\sqrt{E_k \mp (\cos k - \lambda)}}{2E_k} I \tag{3.75}$$

$$= 0. \tag{3.76}$$

Note that eq. (3.72) is consistent with the edge cases in the limits $k \to -\pi$ and $k \to 0$. If also $\lambda = -1$, then we impose that $\eta_{-\pi} = C_{\pi}^{\dagger}$, which is the same as if $\lambda \neq -1$.

Equation (3.55) becomes

$$H = \sum_k E_k^{+}(\eta_k^{+})^{\dagger}\eta_k^{+} + \sum_k E_k^{-}(\eta_k^{-})^{\dagger}\eta_k^{-}. \tag{3.77}$$

Since

$$\left(\eta_{-k}^{-}\right)^{\dagger} = \eta_k^{+} =: \eta_k \tag{3.78}$$

and $E_{-k}^{\pm} = E_k^{\pm}$, we may reduce eq. (3.77) to

$$H = \sum_k E_k \eta_k^{\dagger}\eta_k - \sum_k E_k\left(I - \eta_k^{\dagger}\eta_k\right) \tag{3.79}$$

$$= \sum_k 2E_k \eta_k^{\dagger}\eta_k - I\sum_k E_k. \tag{3.80}$$

The unitary transformation of the $C_{\pm k}$ to obtain $\eta_k^{\pm}$ is an instance of a fermionic **Bogoliubov transformation**:

$$C_k = u f_k + v g_k^{\dagger} \tag{3.81a}$$

and

$$C_{-k} = -v f_k^{\dagger} + u g_k. \tag{3.81b}$$

For these transformations to preserve the CARs,

$$\left\{C_k^{\dagger}, C_k\right\} = |u|^2\left\{f_k^{\dagger}, f_k\right\} + |v|^2\left\{g_k, g_k^{\dagger}\right\} + u^* v\left\{f_k^{\dagger}, g_k^{\dagger}\right\} + v^* u\{g_k, f_k\} \tag{3.82}$$

$$= \left(|u|^2 + |v|^2\right)I, \tag{3.83}$$

so we must have

$$|u|^2 + |v|^2 = 1. \tag{3.84}$$

We may choose

$$u = e^{\mathrm{i}\phi_1}\cos\theta \tag{3.85}$$

$$v = e^{\mathrm{i}\phi_2}\sin\theta \tag{3.86}$$

for real angles ϕ_1, ϕ_2, and θ.

It remains to perform the inverse transformations to obtain the σ_i^x in terms of the $\eta_k^{\pm}$. Shift of perspective: Instead of decomposing the Hilbert space as the tensor product of spin operators, we decompose it as the tensor product of fermionic operators, where the fermionic operators are restricted to their subspaces in order to form an orthonormal basis for the Liouville space. We may then recover σ_i^x by the corresponding isomorphism of Liouville spaces. We choose

$$c, \quad c^{\dagger}, \quad c^{\dagger}c, \quad \text{and} \quad cc^{\dagger} \tag{3.87}$$

as our basis, since they are both orthonormal and eigenoperators of $c^\dagger c^\angle$ with respective eigenvalues

$$-1, \quad 1, \quad 2, \quad \text{and} \quad 0.$$

Theorem 3.1.1. *The fermionic operators* $\bigcup_i \left\{c_i,\ c_i^\dagger,\ c_i^\dagger c_i,\ c_i c_i^\dagger\right\}$ *form an orthonormal complete basis for the Liouville space* $\mathscr{L}\left(\mathscr{H}_2^{\otimes N}\right)$*?*

Proof. We apply the CARs to show orthogonality.

$$\begin{aligned}
\langle c_i | c_j \rangle &= \operatorname{tr}\left(c_i^\dagger c_j\right) && (3.88)\\
&= \operatorname{tr}\left(\delta_{ij} I - c_j c_i^\dagger\right) && (3.89)\\
&= \delta_{ij} \operatorname{tr} I - \operatorname{tr}\left(c_j c_i^\dagger\right) && (3.90)\\
&= 2^N \delta_{ij} - \operatorname{tr}\left(c_i^\dagger c_j\right) && (3.91)\\
&= 2^{N-1} \delta_{ij}, && (3.92)\\
\left\langle c_i^\dagger \middle| c_j^\dagger \right\rangle &= \operatorname{tr}\left(c_i c_j^\dagger\right) && (3.93)\\
&= 2^{N-1} \delta_{ij}, && (3.94)\\
\left\langle c_i^\dagger \middle| c_j \right\rangle &= \operatorname{tr}(c_i c_j) && (3.95)\\
&= -\operatorname{tr}(c_j c_i) && (3.96)\\
&= 0, && (3.97)\\
\left\langle c_i \middle| c_j^\dagger \right\rangle &= \operatorname{tr}\left(c_i^\dagger c_j^\dagger\right) && (3.98)\\
&= -\operatorname{tr}\left(c_j^\dagger c_i^\dagger\right) && (3.99)\\
&= 0, && (3.100)\\
\left\langle c_i^\dagger c_i \middle| c_j \right\rangle &= \operatorname{tr}\left(c_i^\dagger c_i c_j\right) && (3.101)\\
&= -\operatorname{tr}\left(c_i^\dagger c_j c_i\right) && (3.102)\\
&= -\operatorname{tr}\left(c_i c_i^\dagger c_j\right) && (3.103)\\
\left\langle c_i^\dagger c_i \middle| c_j^\dagger c_j \right\rangle &= \operatorname{tr}\left(c_i^\dagger c_i c_j^\dagger c_j\right) && (3.104)\\
&= \delta_{ij} \operatorname{tr}\left(c_i^\dagger c_j\right) - \operatorname{tr}\left(c_i^\dagger c_j^\dagger c_i c_j\right) && (3.105)\\
&= \delta_{ij} \operatorname{tr}\left(c_i^\dagger c_j\right) + \operatorname{tr}\left(c_i^\dagger c_j^\dagger c_j c_i\right) && (3.106)\\
&= \delta_{ij} \operatorname{tr}\left(c_i^\dagger c_j\right) + \operatorname{tr}\left(c_i c_i^\dagger c_j^\dagger c_j\right) && (3.107)
\end{aligned}$$

$$= \delta_{ij}\,\mathrm{tr}\left(c_i^\dagger c_j\right) + \mathrm{tr}\left(c_j^\dagger c_j\right) - \mathrm{tr}\left(c_i^\dagger c_i c_j^\dagger c_j\right) \tag{3.108}$$

$$= \frac{\delta_{ij}}{2}2^{N-1} + \frac{1}{2}2^{N-1} \tag{3.109}$$

$$= (\delta_{ij}+1)2^{N-2}, \tag{3.110}$$

$$\left\langle c_i^\dagger c_i \middle| c_j c_j^\dagger \right\rangle = \mathrm{tr}\left(c_i^\dagger c_i c_j c_j^\dagger\right) \tag{3.111}$$

$$= \mathrm{tr}\left(c_i^\dagger c_i\right) - \mathrm{tr}\left(c_i^\dagger c_i c_j^\dagger c_j\right) \tag{3.112}$$

$$= 2^{N-1} - (\delta_{ij}+1)2^{N-2} \tag{3.113}$$

$$= (1-\delta_{ij})2^{N-2}. \tag{3.114}$$

Thus only c_i and $c_i^\dagger$ are orthogonal, but still require normalization. The full set is not orthogonal. □

We may then write

$$\sigma_x = \sum_{ia} x_{ia} c_{ia}, \tag{3.115}$$

where

$$x_{ia} = \langle \sigma_x | c_{ia} \rangle \tag{3.116}$$

and c_{ia} is restricted to subspace i. That is,

$$c_{1a} = c_a \otimes 0^{\otimes(N-1)}, \tag{3.117}$$

not

$$c_{1a} = c_a \otimes I^{\otimes(N-1)} \tag{3.118}$$

as usual.

Combining eqs. (3.28) and (3.72) gives

$$\eta_k = \frac{-\mathrm{i}\sin k}{\sqrt{2E_k(E_k+(\cos k-\lambda))}}\frac{1}{\sqrt{N}}\sum_i e^{-\mathrm{i}ki}c_i \tag{3.119}$$

$$+\sqrt{\frac{E_k+(\cos k-\lambda)}{2E_k}}\frac{1}{\sqrt{N}}\sum_i e^{-\mathrm{i}ki}c_i^\dagger. \tag{3.120}$$

$$\eta_k^\dagger\eta_k = \frac{\sin^2 k}{2E_k(E_k+(\cos k-\lambda))}\frac{1}{N}\sum_{ij} e^{\mathrm{i}k(i-j)}c_i^\dagger c_j \tag{3.121}$$

$$+\frac{E_k+(\cos k-\lambda)}{2E_k}\frac{1}{N}\sum_{ij} e^{\mathrm{i}k(i-j)}c_i c_j^\dagger \tag{3.122}$$

$$+\frac{i\sin k}{2E_k}\frac{1}{N}\sum_{ij}e^{ik(i-j)}\left(c_i^\dagger c_j^\dagger - c_i c_j\right) \tag{3.123}$$

Should the η_k match Pfeuty? If so, then hmm: these coefficients are complex while Pfeuty's are real. But I don't think they need to match. Having the same Hamiltonian and spectrum of energies says that the two representations of the Hamiltonian in the occupation basis for the different η_k are the same (up to permutation). However, since the spectrum is degenerate (Mathematica), these states may be rotated around within each degenerate eigenspace without changing the representation. It is thus possible that Pfeuty's and my η_k are both correct. While my way of obtaining the η_k through a Fourier transform is more straightforward, it gives more complicated coefficients for the c_i than the method of Lieb et. al., which is less direct (a harder eigenvalue problem than 2×2 matrices), but gives nice coefficients for the c_i [23, pp. 452–454].

We recall eq. (3.20) as

$$\sigma_i^x = -\left(\prod_{j<i}\left(c_j^\dagger c_j - c_j c_j^\dagger\right)\right)\left(c_i^\dagger + c_i\right), \tag{3.124}$$

and write

$$\eta_k = \sum_i a_{ik}c_i + b_{ik}c_i^\dagger \tag{3.125}$$

$$\eta_k^\dagger\eta_k = \sum_{ij}\left(a_{ik}^* c_i^\dagger + b_{ik}^* c_i\right)\left(a_{jk}c_j + b_{jk}c_j^\dagger\right) \tag{3.126}$$

$$= \sum_{ij}\left(a_{ik}^* a_{jk} c_i^\dagger c_j + b_{ik}^* b_{jk} c_i c_j^\dagger + a_{ik}^* b_{jk} c_i^\dagger c_j^\dagger + b_{ik}^* a_{jk} c_i c_j\right) \tag{3.127}$$

$$= \sum_i\left(|a_{ik}|^2 c_i^\dagger c_i + |b_{ik}|^2 c_i c_i^\dagger\right) \tag{3.128}$$

$$+ \sum_{i\neq j}\left(a_{ik}^* a_{jk} c_i^\dagger c_j + b_{ik}^* b_{jk} c_i c_j^\dagger + a_{ik}^* b_{jk} c_i^\dagger c_j^\dagger + b_{ik}^* a_{jk} c_i c_j\right). \tag{3.129}$$

Suppose that eq. (3.125) has been inverted to give coefficients α_{ik} and β_{ik} such that

$$c_i = \sum_k \alpha_{ik}\eta_k + \beta_{ik}\eta_k^\dagger. \tag{3.130}$$

We may then expand eq. (3.20) as

$$\sigma_i^x = -\left(\prod_{j<i}\left(2c_j^\dagger c_j - I\right)\right)\left(c_i^\dagger + c_i\right) \tag{3.131}$$

$$= -\left(\prod_{j<i}\left(2\left(\sum_{kk'}\alpha_{ik}^{*}\alpha_{jk}\eta_k^{\dagger}\eta_{k'} + \alpha_{ik}^{*}\beta_{jk}\eta_k^{\dagger}\eta_{k'}^{\dagger} + \beta_{ik}^{*}\alpha_{jk}\eta_k\eta_{k'} + \beta_{ik}^{*}\beta_{jk}\eta_k\eta_{k'}^{\dagger}\right) - I\right)\right) \tag{3.132}$$

$$\left(\sum_k \left(\alpha_{ik} + \beta_{ik}^{\dagger}\right)\eta_k + \left(\alpha_{ik}^{\dagger} + \beta_{ik}\right)\eta_k^{\dagger}\right) \tag{3.133}$$

$$\sim -\left(\prod_j (x_j - I)\right) \tag{3.134}$$

$$\left(\sum_k \left(\alpha_{ik} + \beta_{ik}^{\dagger}\right)\eta_k + \left(\alpha_{ik}^{\dagger} + \beta_{ik}\right)\eta_k^{\dagger}\right) \tag{3.135}$$

$$\sim -\left(\sum_{I\,:\,J\to\mathbb{Z}_2}\prod_{j\in J}(-1)^{I(j)+1}x_j^{I(j)}\right) \tag{3.136}$$

$$\left(\sum_k \left(\alpha_{ik} + \beta_{ik}^{\dagger}\right)\eta_k + \left(\alpha_{ik}^{\dagger} + \beta_{ik}\right)\eta_k^{\dagger}\right) \tag{3.137}$$

We see that σ_i^x is a generic normal order expression in the η_k, so the eigenoperators we found at each site do not suffice, even if we got around them not being orthogonal.

Evaluation with projectors is intractable because a projector has one occupation state, but an η_k has many. Normal order expressions will have an intermediate number of non-identity factors.

Therefore, evaluate with 2^N dimensional matrices on a computer? Then ρ will only be known in coordinates and we may only consider numerical solutions for small N.

3.2 Computation of transverse-Ising superoperators

TODO: Redefine spin operators and the rest to depend on N and maybe λ, so that we can look at what happens as those parameters are varied.

3.2.1 Utility functions

```
trnorm(op) = tracenorm(dense(op))
acomm(x, y) = x*y + y*x
matriximage(A) = Gray.(1 .- abs.(A ./ max(1, maximum(A))))
```

```
matriximage(A::Operator) = matriximage(dense(A).data)

function acomm_table(a, b, N, image)
    A = Matrix{Float64}(undef, N, N)
    Threads.@threads for i in 1:N
        for j in i:N
            A[j, i] = trnorm(acomm(a(i), b(j))) / 2^N
        end
    end
    L = LowerTriangular(A)
    image ? matriximage(L) : L
end

function acomm_tables(c, N=Int(log2(size(c(1))[1])); image=true)
    ct = dagger ∘ c
    acomms = [
        acomm_table(ct, c, N, image),
        acomm_table(c, c, N, image),
        acomm_table(ct, ct, N, image),
    ]
    if image
        hcat(acomms...)
    else
        for ac in acomms
            display(ac)
        end
    end
end

tpow(x, n) = ⊗(repeat([x], n)...);
```

3.2.2 *Spins*

First, we set up the spin-1/2 sites and Pauli operators.

```
sb = SpinBasis(1//2)
sx, sy, sz, sp, sm = sigmax(sb), sigmay(sb), sigmaz(sb), sigmap(sb), sigmam(sb)
up, down = spinup(sb), spindown(sb)
```

```
ψx = eigenstates(dense(sx))[2][1];
```

3.2.3 *Ising model*

To create a N-site spin loop, we need to construct the basis for the loop and operators for each site.

```
N = 8
sB = tpow(sb, N)
I = identityoperator(sB)
site(i, op) = embed(sB, (i-1)%N + 1, op) # Periodic boundary
acomm_tables(i -> site(i, sm))
```

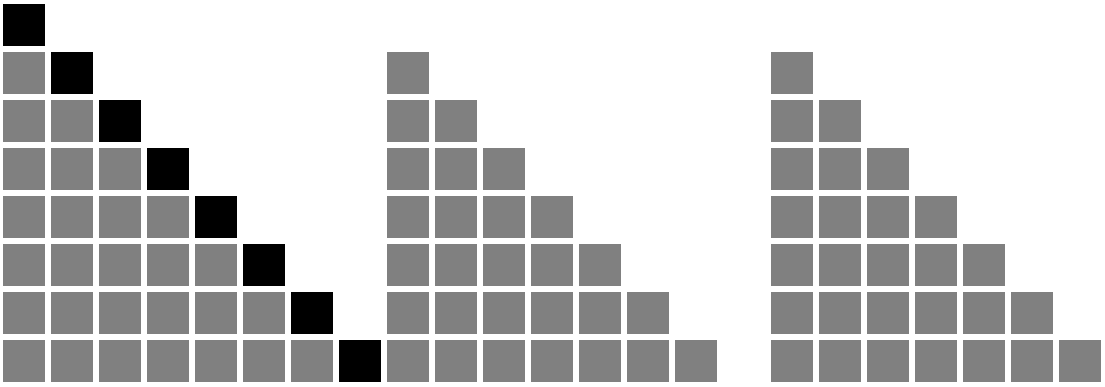

Figure 3.1

We now construct the transverse-field Ising Hamiltonian (Pfeuty)

$$\frac{4H}{J} = -\frac{1}{\lambda}\sum_i \sigma_z^i - \sum_{i=1}^{N} \sigma_i^x \sigma_{i+1}^x, \tag{3.138}$$

where $2\lambda = J/\Gamma$.

```
λ = 1 # Pfeuty with Γ=1, so
Hs = -sum((1/λ)*site(i, sz) + site(i, sx)*site(i+1, sx) for i in 1:N);
```

3.2.4 *Fermion definitions*

```
k(m, N=N) = 2π*(m-1)/N - π*(N - (N%2))/N; # for m in 1:N
function c(i) # for i in 1:N
```

```
    i = (i-1)%N + 1
    A = -(i == 1
        ? I
        : prod(-site(j, sz) for j in 1:(i-1)))
    A * site(i, sm)
end
ct(i) = dagger(c(i))
acomm_tables(c)
```

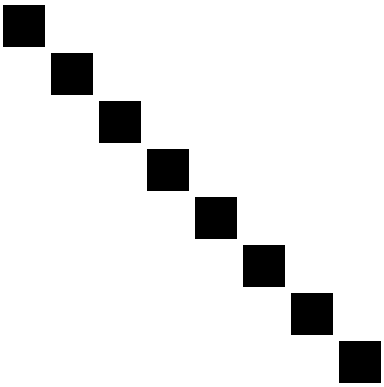

Figure 3.2

Verify that the inverse Jordan-Wigner transformation reproduces all the σ_i^x:

```
csx(i) = -(i == 1 ? I : prod(I - 2*ct(j)*c(j) for j in 1:(i-1))) * (ct(i) +
↪  c(i))
[csx(i) - site(i, sx) for i in 1:N]' .▷ trnorm
```

```
1×8 Array{Float64,2}:
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

The Jordan-Wigner transformed Hamiltonian:

```
L = sum(ct(i)*c(i) for i in 1:N)
Hend = (ct(N) - c(N))*(ct(1) + c(1))*((sparse ∘ exp ∘ dense)(im*π*L) + I)
Hc = I*(N/λ) + Hend - sum(
    (2/λ)*ct(i)*c(i) + (ct(i) - c(i))*(ct(i+1) + c(i+1))
    for i in 1:N)
trnorm(Hs - Hc) / 2^N
```

```
9.727141784868502e-16
```

It looks like the trace norm of the boundary term relative to that of the whole Hamiltonian diminishes with increasing N. But does this imply that it may be dropped?

```
trnorm(Hend) / trnorm(Hs)
```

```
0.3112746194488054
```

Striff

My λ differs from Pfeuty's (**TODO:** $\lambda \mapsto -\alpha$)

```
λS = -1 / λ
```

```
-1.0
```

First, we define the Fourier-transformed fermion operators and verify that the corresponding Hamiltonian is correct.

```
C(k) = sum(exp(-im*k*i) * c(i) for i in 1:N) / √N
Ct(k) = dagger(C(k))
Hk(k) = λS ≈ -1 && k ≈ -π ? zeros(2, 2) : [
    λS - cos(k)  -im*sin(k)
    im*sin(k)    cos(k) - λS
]
vk(k) = [C(k); Ct(-k)]
vkt(k) = [Ct(k) C(-k)];
```

```
acomm_tables(C∘k)
```

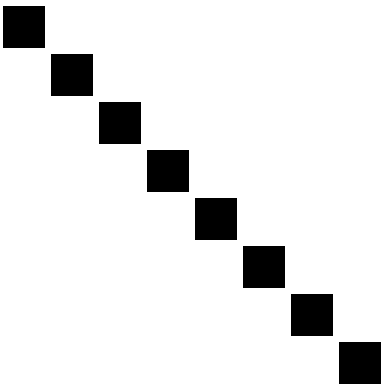

Figure 3.3

```
HC = sum((vkt(k(m))*Hk(k(m))*vk(k(m)))[1] for m in 1:N)
(trnorm((Hs - Hend) - HC), trnorm(Hs - HC)) ./ 2^N
```

```
(1.1408333660893532e-15, 1.0000000000000004)
```

Now, we define E_k and the operators $\eta_k = \eta_k^+$.

```
E(m, N=N) = √(λS^2 - 2λS*cos(k(m, N)) + 1)
aS(m) = ((N%2 == 0 && m == 1) || m == 1 + floor(N//2) ? 0 :
    im*sin(k(m)) / √(2E(m)*(E(m) + cos(k(m)) - λS)))
bS(m) = ((N%2 == 0 && m == 1) || m == 1 + floor(N//2) ? 1 :
    √((E(m) + cos(k(m)) - λS) / (2E(m))))
ηS(m) = aS(m)*C(k(m)) + bS(m)*Ct(-k(m))
ηSt(m) = dagger(ηS(m))
acomm_tables(ηS)
```

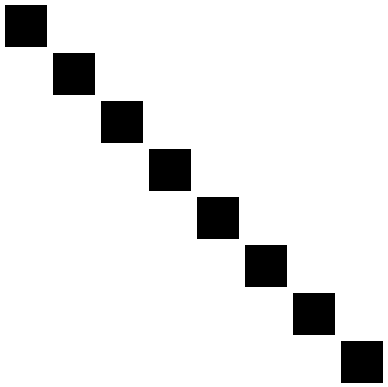

Figure 3.4

Check if the Striff η_k Hamiltonian is correct:

```
E0S = -sum(E(m) for m in 1:N);
HηS = sum(2E(m)*ηSt(m)*ηS(m) for m in 1:N) + E0S*I
(trnorm((Hs - Hend) - HηS), trnorm(Hs - HηS)) ./ 2^N
```

```
(3.598942310911585e-15, 1.0000000000000013)
```

Try to directly diagonalize $\boldsymbol{H}_k$ to find the $\eta_k^\pm$:

```
Heigs = [eigen(Hk(k(m))) for m in 1:N]
EC(m) = Heigs[m].values[2] # Positive energy
ηC(m) = Heigs[m].vectors'[2,1] * C(k(m)) + Heigs[m].vectors'[2,2] * Ct(-k(m))
ηCt(m) = dagger(ηC(m))
ηmC(m) = Heigs[m].vectors'[1,1] * C(k(m)) + Heigs[m].vectors'[1,2] * Ct(-k(m))
```

```
ηmCt(m) = dagger(ηmC(m))
acomm_tables(ηC)
```

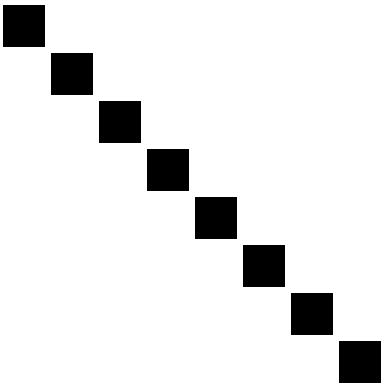

Figure 3.5

```
acomm_tables(ηmC)
```

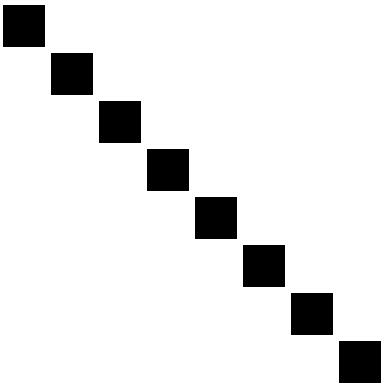

Figure 3.6

```
[ηC(m) - ηS(m) for m in 1:N]' .▷ trnorm
```

```
1×8 Array{Float64,2}:
 0.0  1.2307e-14  2.00972e-14  0.0  0.0  0.0  2.00972e-14  1.2307e-14
```

```
HηC = sum(2EC(m) * (ηCt(m)*ηC(m) - I/2) for m in 1:N)
(trnorm((Hs - Hend) - HηC), trnorm(Hs - HηC)) ./ 2^N
```

```
(2.4774661993019866e-15, 1.0000000000000007)
```

```
HηpmC = sum(EC(m) * (ηCt(m)*ηC(m) - ηmCt(m)*ηmC(m)) for m in 1:N)
(trnorm((Hs - Hend) - HηpmC), trnorm(Hs - HηpmC)) ./ 2^N
```

```
(1.1633808429634807e-15, 1.0)
```

Thus the diagonalization is correct. We may now construct the vacuum state from the sum of number operators.

```
# NηS = sum(ηSt(m)*ηS(m) for m in 1:N)
# vacS = eigenstates(NηS, 1, info=false)[2][1] # Indices select first
↪  eigenstate
# ((ηSt(1)*ηS(1) + ηSt(2)*ηS(2)) - 2*I) * (ηSt(1) * ηSt(2) * vacS) ▷ norm
```

Pfeuty

Pfeuty: η's do not satisfy the CARs? Typo, or **problem**?

```
Λ(m) = √(λ^2 + 2λ*cos(k(m)) + 1)
ϕ(k, i) = √(2/N) * (k > 0 ? sin(k*i) : cos(k*i))
ψ(k, i) = -((1 + λ*cos(k))*ϕ(k, i) + λ*sin(k)*ϕ(-k, i)) / Λ(k)
a(m, i) = (ϕ(k(m), i) + ψ(k(m), i)) / 2
b(m, i) = (ϕ(k(m), i) - ψ(k(m), i)) / 2
η(m) = sum(a(m, i)*c(i) + b(m, i)*ct(i) for i in 1:N)
ηt(m) = dagger(η(m));

acomm_tables(η)
```

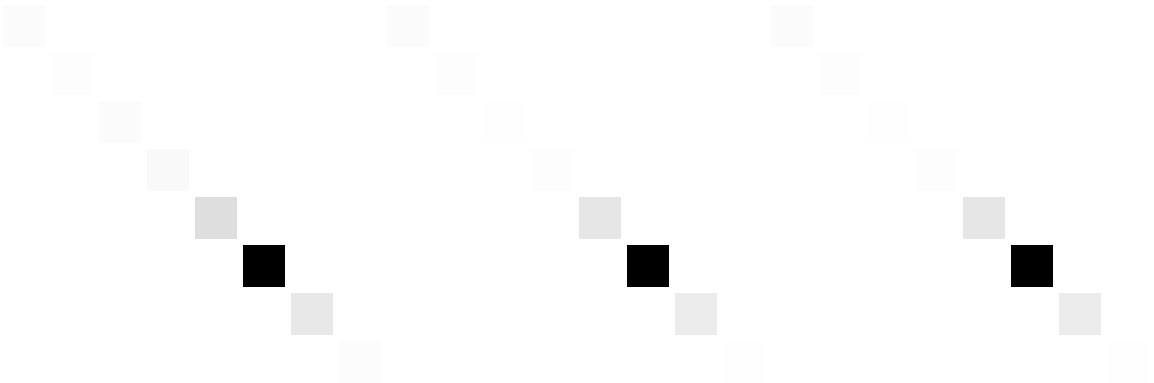

Figure 3.7

Check if the Pfeuty Hamiltonian is correct:

```
E0 = -sum(Λ(m) for m in 1:N) / 2;
Hη = sum(Λ(m)*ηt(m)*η(m) for m in 1:N) + E0*I;

(trnorm((Hs - Hend) - Hη), trnorm(Hs - Hη)) ./ 2^N
```

```
(65.37447076675215, 65.37447076675215)
```

3.2.5 *Starting from the many-body basis instead*

```
import QuantumOpticsBase: ManyBodyBasis, SparseOperator, isnonzero

function destroyfermion(b::ManyBodyBasis, index::Int)
    c = SparseOperator(b)
    # <{m}_j| c |{m}_i>
    for i in 1:length(b)
        occ_i = b.occupations[i]
        if occ_i[index] == 0
            continue
        end
        sign = sum(occ_i[1:(index-1)]) % 2 == 0 ? 1 : -1
        for j in 1:length(b)
            if isnonzero(occ_i, b.occupations[j], index)
                c.data[j, i] = sign * sqrt(occ_i[index])
            end
        end
    end
    c
end
```

```
destroyfermion (generic function with 1 method)
```

```
minusk(m, N=N) = N%2 == 0 ? (m == 1 ? 1 : N - (m - 2)) : (N - (m - 1)); # π ≡
 ↪ -π

f = NLevelBasis(N) # "Levels" 1 to N are indices of k's
mbb = ManyBodyBasis(f, fermionstates(f, [0:N...]))
Imb = identityoperator(mbb)
ηmb(m) = destroyfermion(mbb, m)
ηtmb(m) = dagger(ηmb(m))
Hf = diagonaloperator(f, @. Λ(k(1:N)) + E0)
Hmb = manybodyoperator(mbb, Hf)
acomm_tables(ηmb)
```

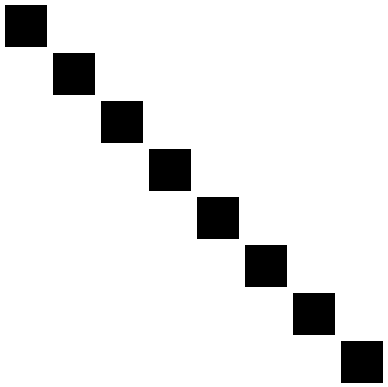

Figure 3.8

```
Cmbs = [([amS(m) bmS(m); aS(m) bS(m)] * [ηtmb(minusk(m)), ηmb(m)])[1] for m in
↪  1:N]
acomm_tables(m -> Cmbs[m])
```

```
UndefVarError: amS not defined

Stacktrace:
 [1] (::var"#35#36")(::Int64) at ./none:0
 [2] iterate at ./generator.jl:47 [inlined]
 [3] collect(::Base.Generator{UnitRange{Int64},var"#35#36"}) at ./arra .jl:686
 [4] top-level scope at In[30]:1
```

```
Heigs[1].vectors
```

```
2×2 Array{Float64,2}:
 1.0  0.0
 0.0  1.0
```

```
Cmbs = [(Heigs[m].vectors * [ηtmb(minusk(m)), ηmb(m)])[1] for m in 1:N]
acomm_tables(m -> Cmbs[m])
```

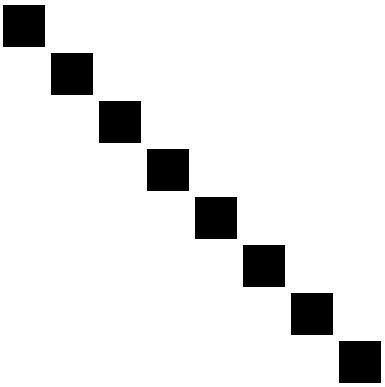

Figure 3.9

```
cmb(i) = sum(exp(im*k(m)*(i-1)) * Cmbs[m] for m in 1:N) / √N
cmbt(i) = dagger(cmb(i))
sxmb(i) = -(i == 1 ? Imb : prod(Imb - 2*cmbt(j)*cmb(j) for j in 1:(i-1))) *
↪  (cmbt(i) + cmb(i))
acomm_tables(cmb)
```

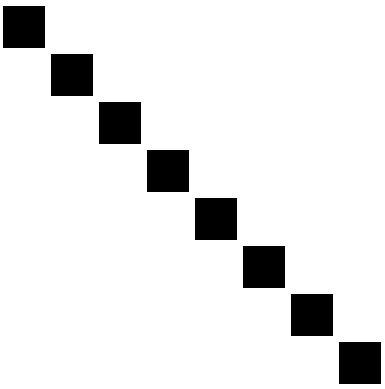

Figure 3.10

3.2.6 Energy spacings

The eigenvalues of the eigenoperators we seek are energy spacings. We should then figure out how they are distributed, and verify that the smallest energy spacing is large enough to be represented as a floating point number. Even with floating point errors, we may find rigorous bounds on the minimum energy difference using interval arithmetic.

```
using IntervalArithmetic
setformat(:midpoint);
```

Note that ΔEs generally double-counts due to parity symmetry.

```
function ΔEs(NE)
    Eks = (m -> E(@interval(m), NE)).(1:NE)
    diffweights = Iterators.product(repeat([[-1,0,1]], NE)...)
    Ediffs = (abs(w ⋅ Eks) for w in diffweights)
    Ediffs = Ediffs ./ abs(-(extrema(Ediffs)...))
    Ediffs ▷ collect ▷ vec ▷ sort
end
;
```

```
NE = 4
ΔEplot = plot(mid.(ΔEs(NE)), 0:1//3^NE:1-eps(),
    label=latexstring("N = $NE"),
    xlim=(0, 1),
    ylim=(0, 1),
    key=:bottomright,
    xlabel=L"x \in \mathrm{supp}(\Delta E/\Delta E_\text{max})",
    ylabel=L"P(\Delta E/\Delta E_\text{max} \leqslant x)")
for NE in 5:10
    plot!(ΔEplot, mid.(ΔEs(NE)), 0:1//3^NE:1-eps(), label=latexstring("N =
    ↪ $NE"))
end
ΔEplot
```

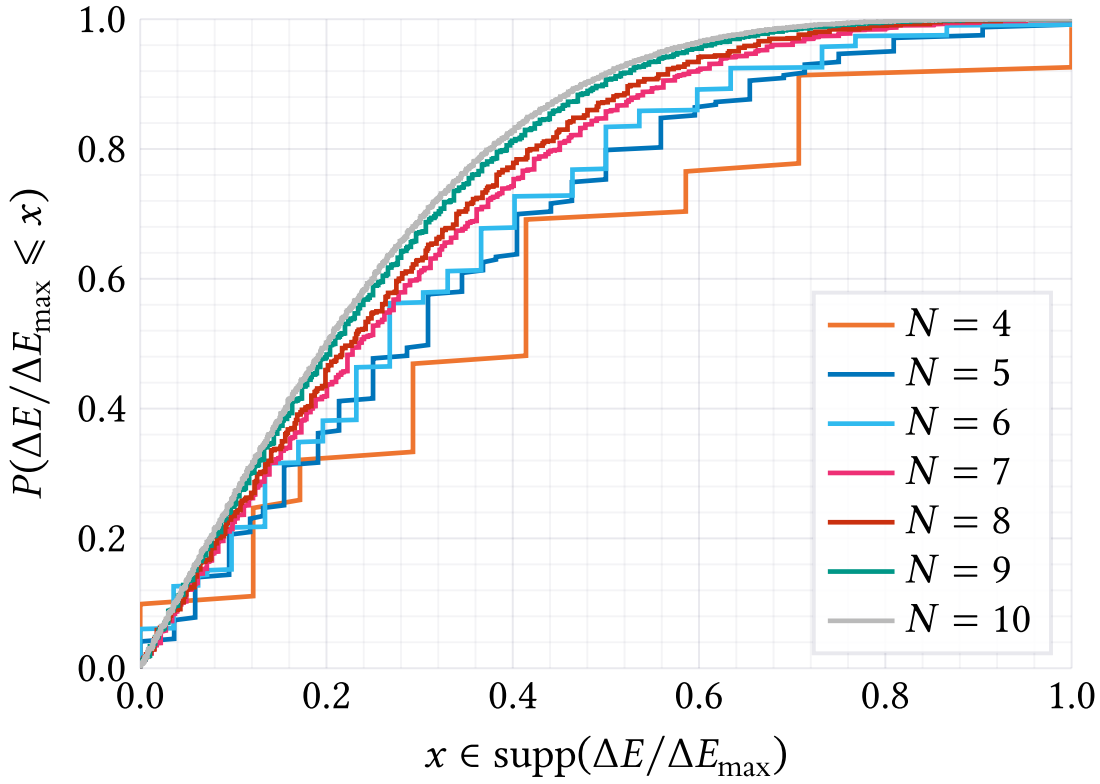

Figure 3.11

As expected, the density approaches a truncated normal distribution.

```
using KernelDensity
ΔE_kde = kde(mid.(ΔEs(12)))
plot(ΔE_kde.x, ΔE_kde.density)
```

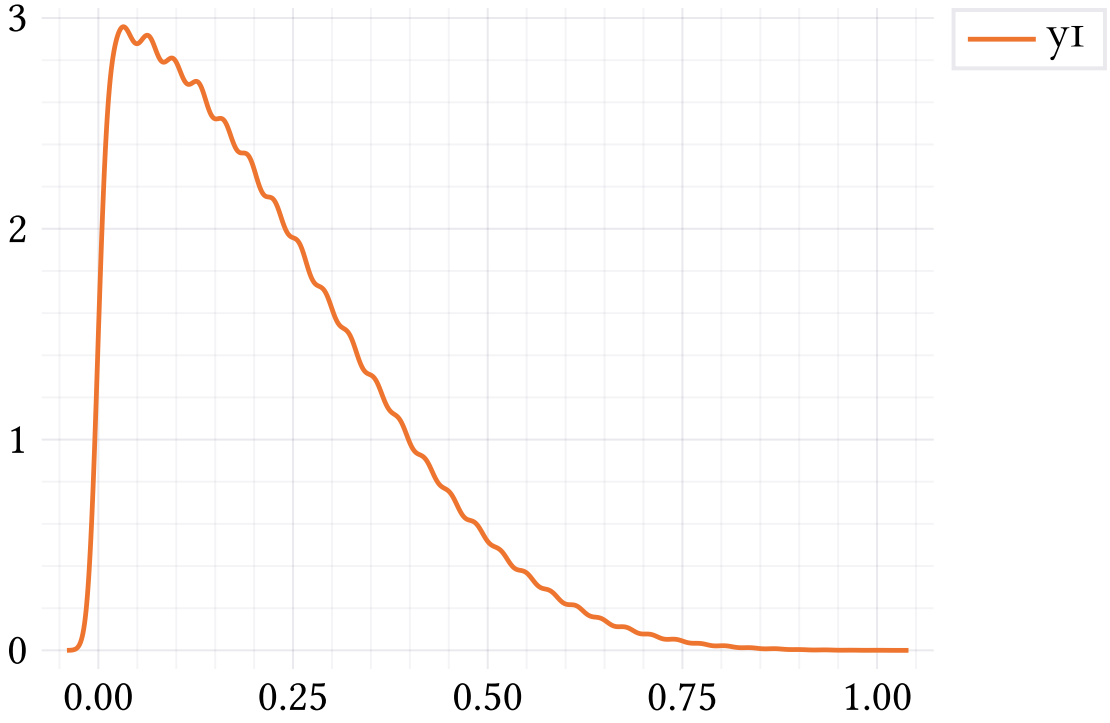

Figure 3.12

This rigorously shows that the smallest ΔE is either within the first interval (nonzero) or the second interval (very close to zero).

```
using DataStructures
```

```
ArgumentError: Package DataStructures not found in current path:
- Run `import Pkg; Pkg.add("DataStructures")` to install the DataStructures package



Stacktrace:
 [1] require(::Module, ::Symbol) at ./loading.jl:893
```

```
minΔEs = []
Threads.@threads for NE in 2:15 # A parallel fold would be faster
    push!(minΔEs, NE ⇒
        foldr((x, (m, z)) -> 0 ∈ x ? (m, x ∪ z) : (min(x, m), z),
            ΔEs(NE),
            init=(@interval(Inf), @interval(0))))
```

```
end
Dict(minΔEs) ▷ sort
```

```
OrderedCollections.OrderedDict{Int64,Tuple{Interval{Float64},Interval{Float64}}} with 13 er
  2  => (1 ± 7.45059e-09, 3.72529e-09 ± 3.7253e-09)
  3  => (0.25 ± 1.9429e-16, 1.249e-16 ± 1.24901e-16)
  4  => (0.12132 ± 1.73028e-09, 1.54307e-09 ± 1.54307e-09)
  5  => (0.0364745 ± 7.63279e-17, 6.00386e-17 ± 6.00387e-17)
  7  => (0.00544257 ± 9.80119e-17, 7.10263e-17 ± 7.10263e-17)
  8  => (0.00355265 ± 7.43639e-10, 7.41006e-10 ± 7.41007e-10)
  9  => (0.00363698 ± 1.6957e-16, 1.90378e-16 ± 1.90379e-16)
  10 => (0.000841145 ± 5.90525e-10, 5.90028e-10 ± 5.90029e-10)
  11 => (0.000302047 ± 2.03668e-16, 2.15277e-16 ± 2.15277e-16)
  12 => (0.000463957 ± 4.90672e-10, 4.90444e-10 ± 4.90444e-10)
  13 => (2.37052e-05 ± 2.3923e-16, 2.60954e-16 ± 2.60955e-16)
  14 => (5.20027e-05 ± 4.19762e-10, 4.1974e-10 ± 4.1974e-10)
  15 => (0.000244098 ± 2.19307e-16, 2.56035e-16 ± 2.56036e-16)
```

From the plot, we can derive an approximate lower bound for the smallest energy spacing. From the small N plots, it is clear that the energy differences are not uniformly separated. However, the CDF (from sorted indices) is uniformly spaced. Since we have computed 3^N energy differences, we may then expect the typical spacing between ΔE values at x to be $Q'(x)3^{-N}$, where $Q(x)$ is the limiting differentiable quantile function (inverse CDF). Suppose that for large N, $Q'(0) \to k$. From the plots, we estimate that $k > 1/3$. We may then make the following estimates:

```
Dict(NE ⇒ (1/3) * 3.0^-NE for NE in 2:15) ▷ sort
```

```
OrderedCollections.OrderedDict{Int64,Float64} with 14 entries:
  2  => 0.037037
  3  => 0.0123457
  4  => 0.00411523
  5  => 0.00137174
  6  => 0.000457247
  7  => 0.000152416
  8  => 5.08053e-5
  9  => 1.69351e-5
  10 => 5.64503e-6
```

```
  11 => 1.88168e-6
  12 => 6.27225e-7
  13 => 2.09075e-7
  14 => 6.96917e-8
  15 => 2.32306e-8
```

Note that these are all smaller than the actual minimum energy differences, as desired. Now we may estimate the N where the spacing is small enough to cause problems to be around

```
-log(3, 3*eps()) ▷ floor ▷ Int
```

```
31
```

3.2.7 *Construction of superoperators*

```
function addentry!(dict, key, value, isequal=isequal)
    for k in keys(dict)
        if isequal(k, key)
            push!(dict[k], value)
            return dict
        end
    end
    dict[key] = [value]
    dict
end

stateE(occ, N=length(occ)) = E0S + 2*sum(n == 1 ? E(m) : 0 for (m, n) in
↪  enumerate(occ))

states = fermionstates(f, [0:N...])
energies = Dict{Float64, Vector{Vector{Int64}}}()
for state in states
    addentry!(energies, stateE(state), state, ≈)
end
projectors = Dict(energy ⇒ sum(sparse(projector(basisstate(mbb, state))) for
↪  state in states)
    for (energy, states) in energies)
```

```
ωs = Dict{Float64,Vector{Tuple{Float64,Float64}}}()
for E1 in keys(energies), E2 in keys(energies)
    addentry!(ωs, E2 - E1, (E1, E2), ≈)
end
```

Now we may compute the superoperators for σ_x^i.

```
using ProgressMeter

function Aωs(i)
    A = sxmb(i)
    d = Dict{Float64, typeof(A)}()
    for (ω, Ediffs) in ωs
        Aω = sum(projectors[E1] * A * projectors[E2] for (E1, E2) in Ediffs)
        if Aω != zero(A)
            d[ω] = Aω
        end
    end
    d
end
```

```
Aωs (generic function with 1 method)
```

Since the spectral correlation tensor is already diagonal, the `Aωs(i)` are the Lindblad operators.

3.2.8 Time evolution

```
using ProgressMeter
```

We start the system off in a random state.

```
ψs = randstate(mbb);
```

Now we time-evolve the system and look at the expectation values of some reduced operators on the spin chain. We start with just the isolated chain.

```
function fout(t, ψ)
    next!(progress)
    ψn = normalize(ψ)
    real([
            expect(sxmb(1), ψn)
```

```
            ])
end
ts = [0:0.05:20;]
progress = Progress(length(ts))
ts, fouts = timeevolution.schroedinger(ts, ψs, Hmb, fout=fout);
```

```
Progress: 100%|████████████████████████████████████████| Time: 0:00:00
```

```
plot(xlabel="Time", ylabel=L"\left\langle\pauli_1^x\right\rangle")
plot!(ts, hcat(fouts...)',
    title = latexstring("$N spins, \$\\lambda = $λ\$"),
    color = :gray,
    label = [
        "Closed"
        ])
```

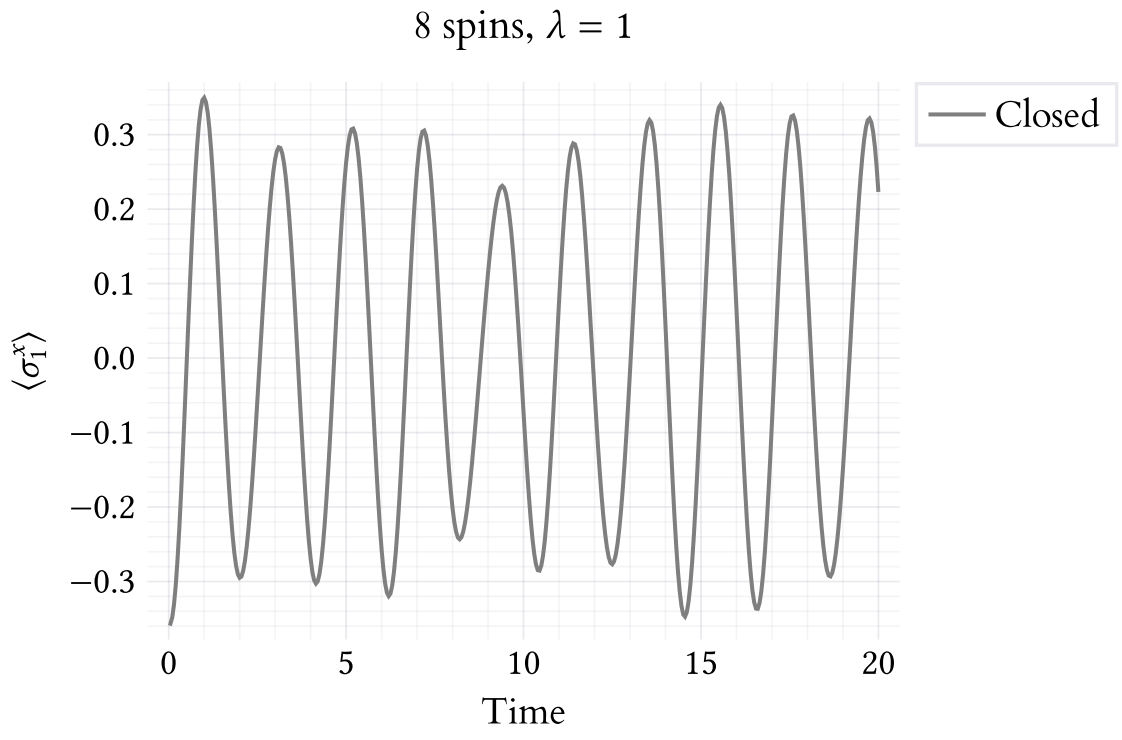

Figure 3.13: Isolated Ising time evolution.

```
savefig("closed.pdf")
```

Now we consider the open system.

```
β, Ω = 100, 1
nB(ω, β) = 1 / (exp(β*ω) - 1)
coupling(ω, Ω) = abs(ω) / (1 + (ω / Ω)^2)
γ(ω, β=β, Ω=Ω) = coupling(ω, Ω)^2 * (nB(ω, β) + 1)
jumpoperators = typeof(first(Aωs(1)).second)[]
γs = Float64[]
progress = Progress(N)
Threads.@threads for i in 1:N
    for (ω, L) in Aωs(i)
        γω = γ(ω)
        if γω > 0
            push!(jumpoperators, L)
            push!(γs, γω)
        end
    end
    next!(progress)
end
```

```
Progress: 100%|████████████████████████████████████████| Time: 0:00:01
```

```
progress = Progress(length(ts));
ts, fouts = timeevolution.mcwf(ts, ψs, Hmb, jumpoperators, rates=γs,
 ↪  fout=fout);
```

```
Progress: 100%|████████████████████████████████████████| Time: 0:00:05
```

```
plot!(ts, hcat(fouts...)',
    title = latexstring("$N spins, \$\\lambda = $λ\$"),
    label = [
        "MCWF"
        ])
```

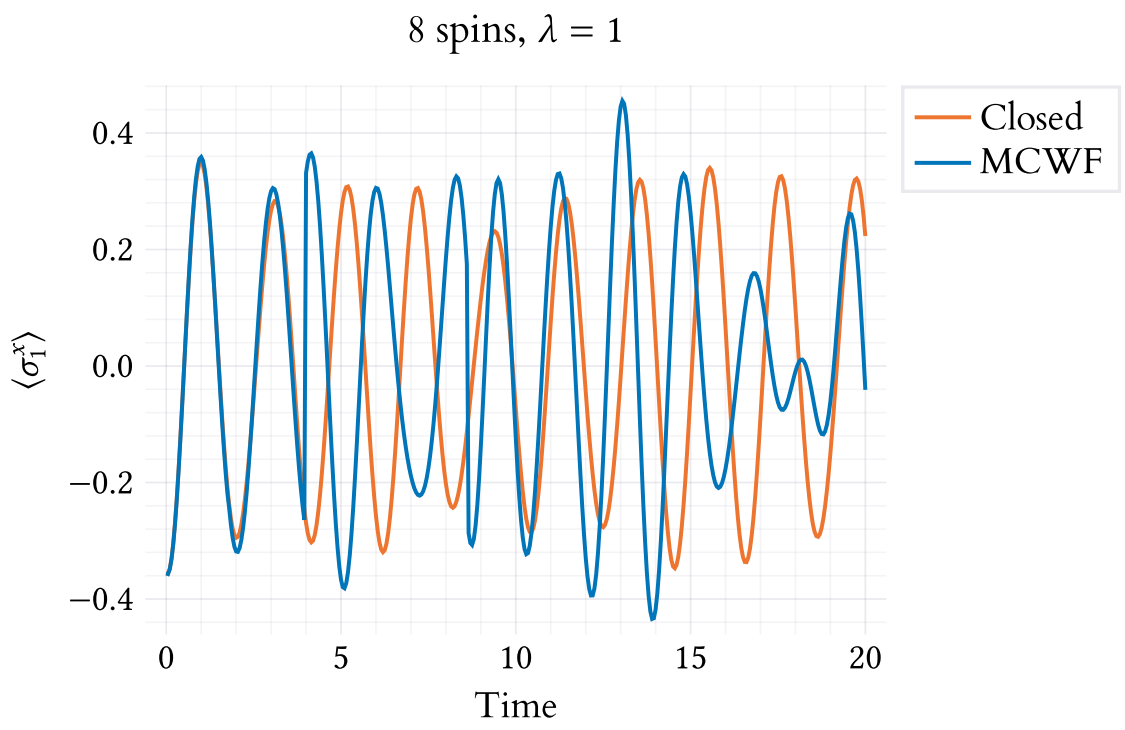

Figure 3.14: Open Ising time evolution.

```
progress = Progress(length(ts));
ts, fouts = timeevolution.master(ts, ψs, Hmb, jumpoperators, rates=γs,
 ↪  fout=fout);
```

```
Progress: 100%|████████████████████████████████████████| Time: 0:23:39
```

```
plot!(ts, hcat(fouts...)',
    title = latexstring("$N spins, \$\\lambda = $λ\$"),
    label = [
        "Master"
        ])
```

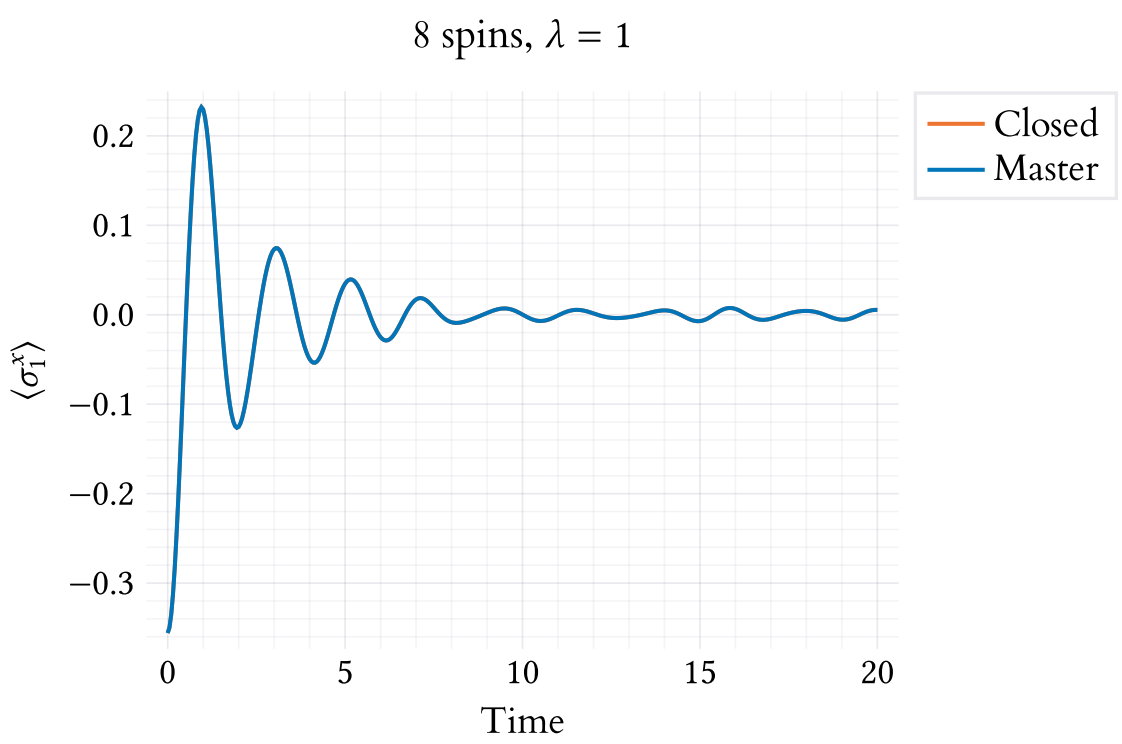

Figure 3.15: Open Ising time evolution.

```
rubric = RGB(0.7, 0.05, 0.0);
```

```
plot!(mts, hcat(mfouts...)',
    title = latexstring("$N spins, \$\\lambda = $λ\$"),
    color = rubric,
    label = [
        "Open"
        ])
```

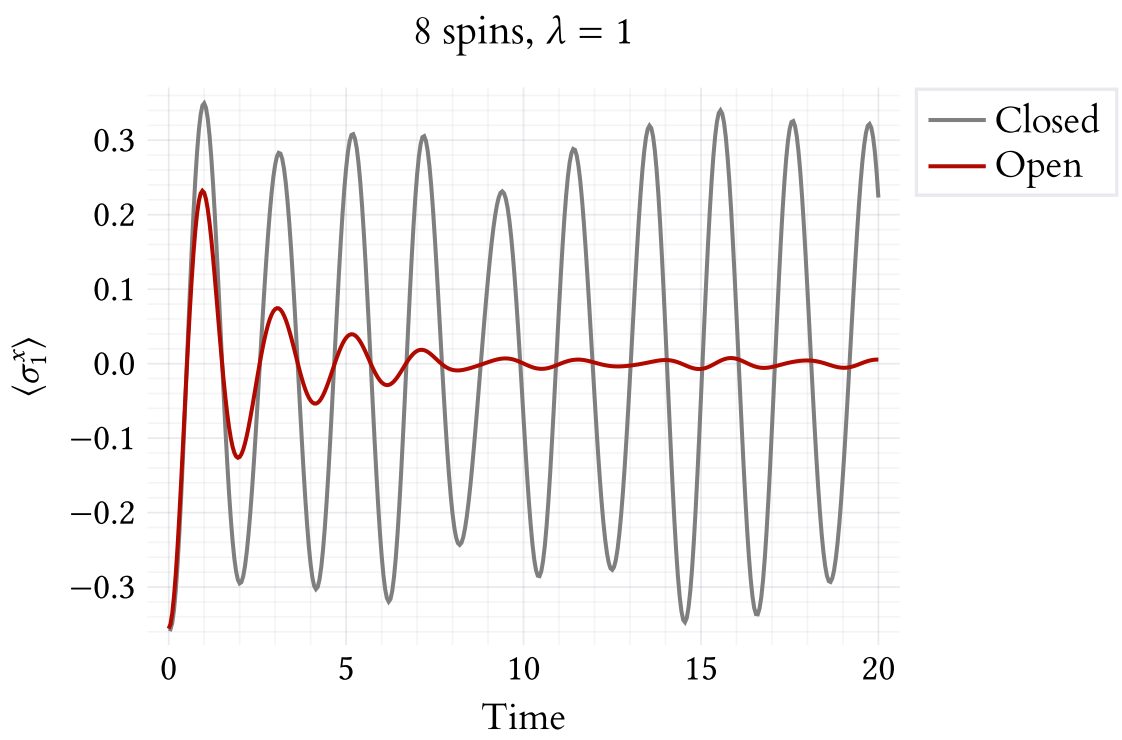

Figure 3.16: Open Ising time evolution.

```
savefig("open.pdf")
```

3.3 Computation of transverse-Ising jump operators

```
using BenchmarkTools, Revise
includet("TransverseIsingModels.jl")
using .TransverseIsingModels
TIM = TransverseIsingModels;
```

Setup

```
title(s::TransverseIsingModel{N}) where N = latexstring("$(N)-spin transverse
 ↪  Ising model with \$\\lambda =\$ $(s.λ)");

# These can likely be made more efficient.
opip(A, B) = tr(dagger(A) * B)
opip(A) = tr(dagger(A) * A)
opnorm(A) = √opip(A)
```

```
opnormalize(A) = A / opnorm(A)
opcos(A, B) = real(abs(opip(A, B)) / (opnorm(A) * opnorm(B)))

project(P, J) = opip(P / opip(P), J)
# project(P, J) = J == zero(J) ? missing : opip(P / opip(P), J)
jumpcos(P, J) = opcos(P, J);
# jumpcos(P, J) = J == zero(J) ? missing : opcos(P, J);
```

Jump operator computations

```
using Statistics

function J(s, A, P, ΔEs, Πs)
    project(P, jumpoperator(ΔEs, A, Πs))
end
function jumpcomputation(s, bes)
    Πs, ωs = jumpinfo(s; basiseigensys=bes)
    _jumpcomputation(s, bes, Πs, ωs)
end;

_op(s, sa, i, bes) = changebasis(sa(s, i)::typeof(one(Tbasis(s)));
 ↪  basiseigensys=bes)
function _jumpcomputation(s, bes, Πs, ωs)
    sum(values(ωs)) do ΔEs
        all([sx, sy, sz]) do sa
            all([sx, sy, sz]) do sb
                all(0:(s.N-1)) do j
                    Js = Iterators.dropwhile(ismissing, (J(s, _op(s, sa, i,
                     ↪  bes), _op(s, sb, ((i+j-1) % s.N) + 1, bes), ΔEs, Πs)
                     ↪  for i in 1:s.N))
                    if isempty(Js)
                        true
                    else
                        J1, restJs = Iterators.peel(Js)
                        all(restJs) do Ji
                            ismissing(Ji) || TIM.isapprox.(J1, Ji,
                             ↪  atol=1e-12)::Bool
                        end
                    end
                end
            end
```

```
        end
    end
end;

function find_projections(s, bes, Πs, ωs)
    As = interactions(s; basiseigensys=bes) ▷ collect
    Ps = sitejumps(s; basiseigensys=bes) ▷ collect
    Pnorm2s = map(opip, Ps)
    projections = map(values(ωs)) do ΔEs
        Js = map(As) do A
            jumpoperator(ΔEs, A, Πs)
        end
        Js = filter(x -> !iszero(x), Js)
        map(Iterators.product(Js, zip(Ps, Pnorm2s))) do (J, (P, Pn))
            opip(P, J) / Pn ▷ abs
        end
    end
    Iterators.flatten(projections) ▷ collect
end;

sproj = find_projections(tim, bes, Πs, ωs);

length(filter(x -> x > 0.01, sproj) ▷ collect) / length(sproj)
```

```
0.04687208216619981
```

```
gr()
```

```
Plots.GRBackend()
```

```
histogram(sproj, normalize=true,
    xlim=(-0.01, 0.05),
    xlabel=L"Projection magnitude $\abs{\ip{\hat{\opr{P}}}{\opr{J}}}$",
    ylabel="Density",
    title=title(tim),
    legend=false)
projkde = kde(sproj)
plot!(projkde.x, projkde.density)
vline!([1 / 2^tim.N])
```

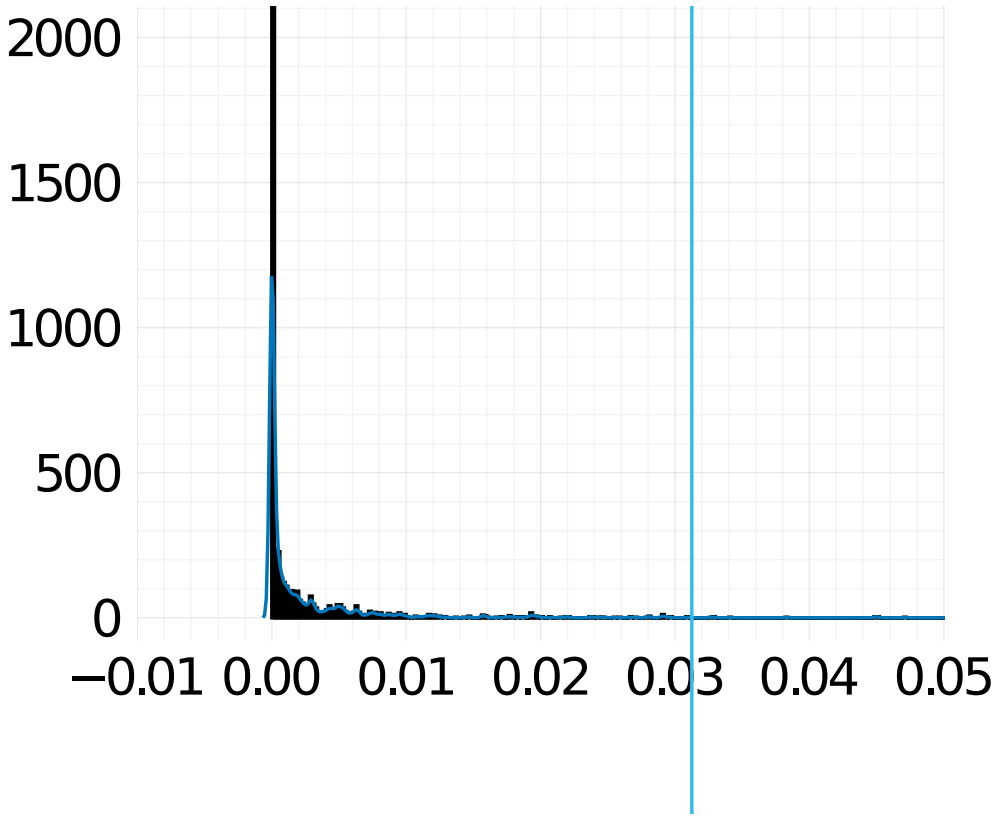

Figure 3.17

```
GKS: glyph missing from current font: 73
GKS: glyph missing from current font: 108
\abs{\ip{\hat{\opr{P}}}{\opr{J}}}$: undefined symbol
GKS: glyph missing from current font: 73
GKS: glyph missing from current font: 108
\abs{\ip{\hat{\opr{P}}}{\opr{J}}}$: undefined symbol
GKS: Rectangle definition is invalid in routine SET_VIEWPORT
\abs{\ip{\hat{\opr{P}}}{\opr{J}}}$: undefined symbol
\abs{\ip{\hat{\opr{P}}}{\opr{J}}}$: undefined symbol
GKS: glyph missing from current font: 73
GKS: glyph missing from current font: 108
GKS: glyph missing from current font: 73
GKS: glyph missing from current font: 108
GKS: glyph missing from current font: 73
GKS: glyph missing from current font: 108
```

```
njumps(tim) = jumpinfo(tim, basiseigensys=basiseigen(tim))[2] ▷ length;

N = 8
λs = -12:0.01:12
plot(λ -> njumps(TransverseIsingModel(N, exp(λ))), λs, xlabel=L"\ln\lambda",
 ↪  ylabel="Jumps", key=false, title="N = $N")
```

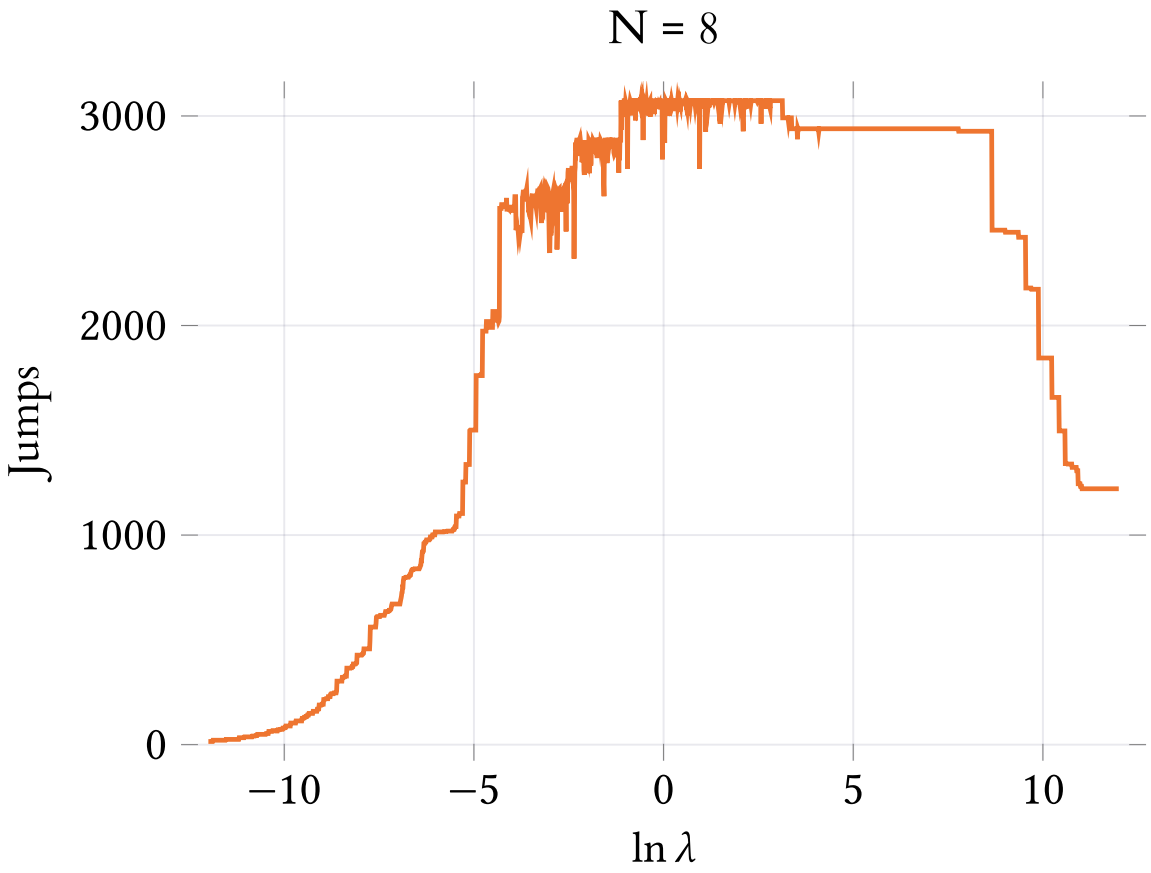

Figure 3.18

```
function timviolin!(p, N, m)
    λ = exp(m)
    tim = TransverseIsingModel(N, λ)
    bes = basiseigen(tim)
    As = [changebasis(op(tim, i); basiseigensys=bes)
        for i in 1:tim.N, op in [sx, sy, sz]]
    Ps = [changebasis(op(tim, i); basiseigensys=bes)
        for i in 1:tim.N, op in [sx, sy, sz]]
    Πs, ωs = jumpinfo(tim; basiseigensys=bes)
```

```
    ωprojections = find_projections(tim, bes, Πs, ωs)
    violin!(p, repeat([m], length(ωprojections)), ωprojections,
        ylim=(-0.1 / 2^N, 2.0 / 2^N)
    )
end;
```

```
using StatsPlots
```

```
┌ Info: Precompiling StatsPlots [f3b207a7-027a-5e70-b257-86293d7955fd]
└ @ Base loading.jl:1278
```

```
N = 6
p = plot(key=false, title="N = $N");
for m in (-10):2:10
    timviolin!(p, N, m)
end
hline!(p, [1 / 2^N])
```

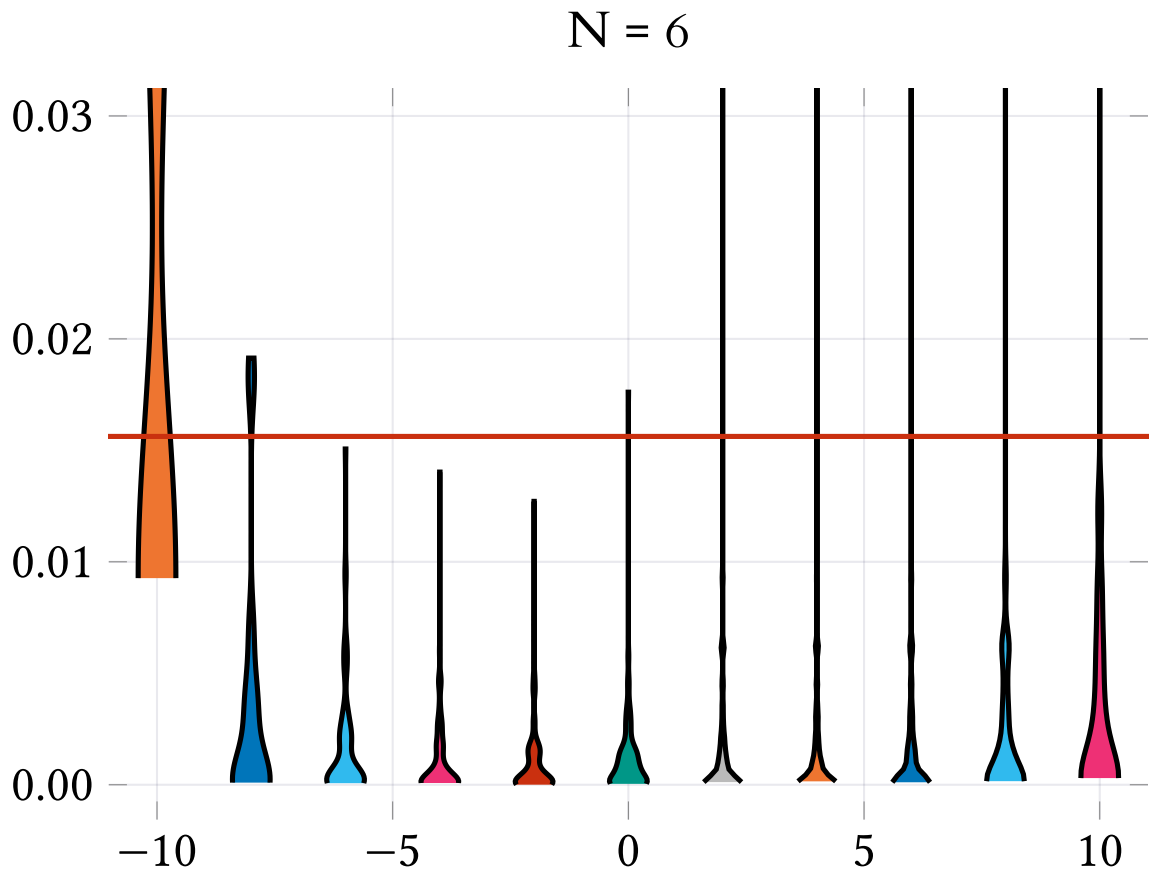

Figure 3.19

Note that some values above 2^{-N} are cut off to show more detail.

```
plot(ωprojections,
    xlabel=L"Energy difference $\omega$",
    ylabel="Mean projected magnitude",
    ylim=(0, 1 / 2^tim.N),
    key=false)
```

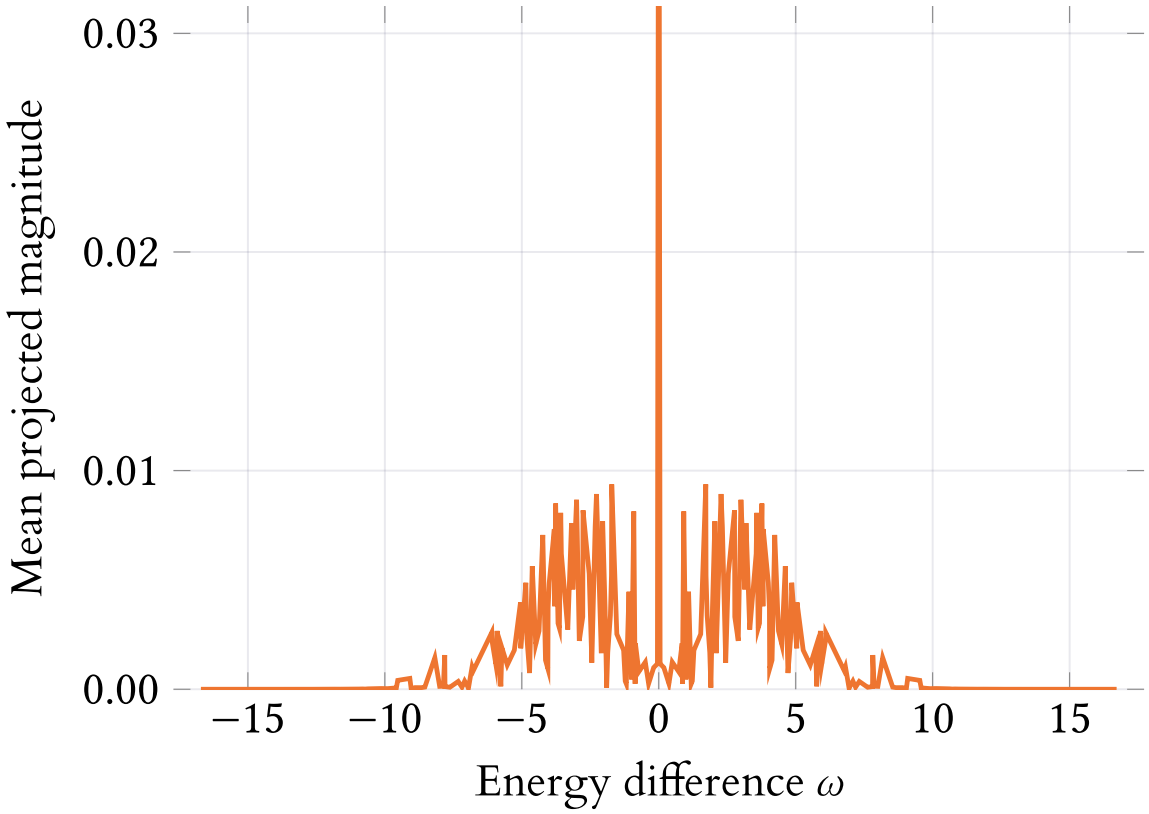

Figure 3.20

```
using KernelDensity

proj = collect(values(ωprojections))
histogram(proj, normalize=true,
    xlabel=L"Projection magnitude $\abs{\ip{\hat{\opr{P}}}{\opr{J}}}$",
    ylabel="Density",
    title=title(tim),
    legend=false)
projkde = kde(proj)
plot!(projkde.x, projkde.density)
vline!([1 / 2^tim.N])
```

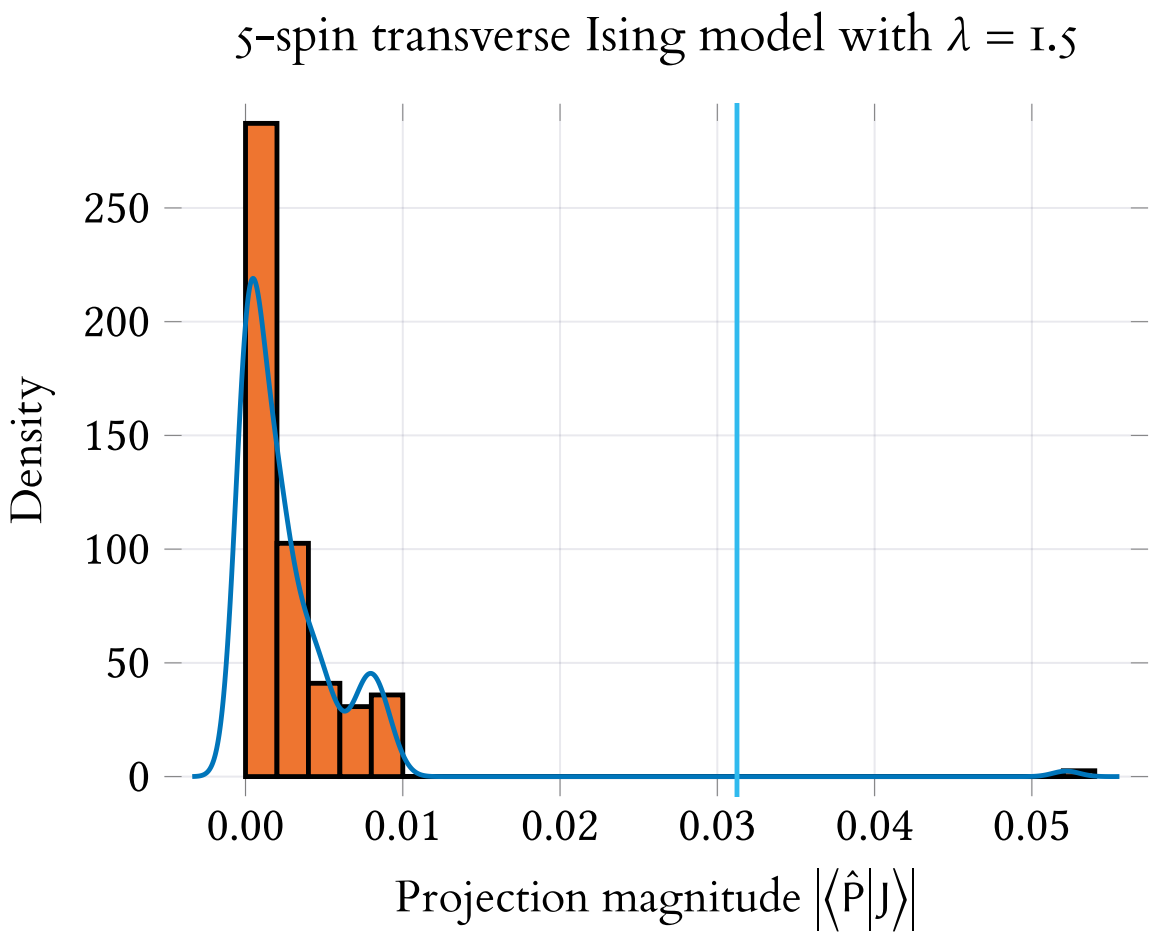

Figure 3.21

For fermions

```
function fermions(s::TIM.SpinSystem{N}; basiseigensys) where N
    (changebasis(op(s, m); basiseigensys=basiseigensys)
        for m in 1:N, op in [TIM.η, TIM.ηt])
end;

function fermion_projections(s, bes, Πs, ωs)
    As = fermions(s; basiseigensys=bes) ▷ collect
    Ps = sitejumps(s; basiseigensys=bes) ▷ collect
    Pnorm2s = map(opip, Ps)
#     projections = dictmap(ωs) do ΔEs
    projections = map(values(ωs)) do ΔEs
```

```
        Js = map(As) do A
            jumpoperator(ΔEs, A, Πs)
        end
        Js = filter(x -> !iszero(x), Js)
        map(Iterators.product(Js, zip(Ps, Pnorm2s))) do (J, (P, Pn))
            opip(P, J) / Pn ▷ abs
        end ▷ mean
    end
    projections = filter(v -> !isnan(v), projections)
end;
```

```
tim = TransverseIsingModel(5, 0.5)
bes = basiseigen(tim)
Πs, ωs = jumpinfo(tim; basiseigensys=bes);
```

```
ηprojections = fermion_projections(tim, bes, Πs, ωs);
```

```
ηproj = collect(values(ηprojections))
histogram(ηproj, normalize=true,
    xlabel=L"Projection magnitude $\abs{\ip{\hat{\opr{P}}}{\opr{J}}}$",
    ylabel="Density",
    title=title(tim),
    legend=false)
ηprojkde = kde(ηproj)
plot!(ηprojkde.x, ηprojkde.density)
vline!([1 / 2^tim.N])
```

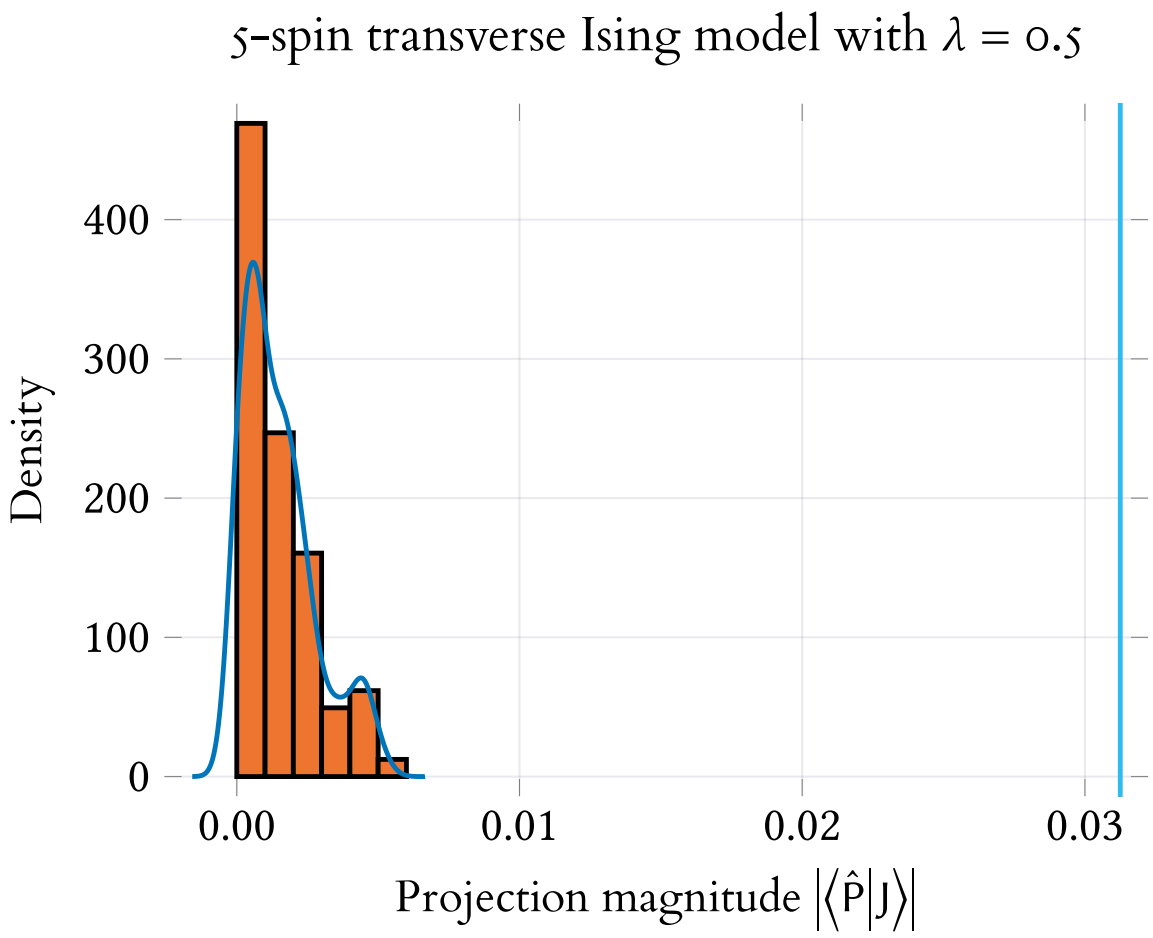

Figure 3.22

TODO - Fix slowness: projections (maybe also from precompilation and type-instability)
- Check with the η_k?

```
using SymPy;
include("TransverseIsingModels.jl")
using .TransverseIsingModels
TIM = TransverseIsingModels;

⊗ₖ = kron;
const σ0 = [1 0; 0 1];
const σx = [0 1; 1 0];
const σy = [0 -im; im 0];
const σz = [1 0; 0 -1];
const σp = [0 1; 0 0];
const σm = [0 0; 1 0];
```

3.4 Computation of jump operators

```
function symeigen(H)
    symeig = H.eigenvects()
    vals, vecs = eltype(H)[], []
    for (λ, _, vs) in symeig
        for v in vs
            push!(vals, λ)
            push!(vecs, vec(v))
        end
    end
    vals, vecs
end;

function addentry!(dict, key, value; isequal=isequal)
    for k in keys(dict)
        if isequal(k, key)
            push!(dict[k], value)
            return dict
        end
    end
    dict[key] = [value]
    dict
end;

firstvalue(i, (x, y)) = x
lastvalue(i, (x, y)) = y
function dictby(A; isequal=isequal, keyof=firstvalue, valof=lastvalue)
    i0, x0 = 1, first(A)
    k0, v0 = keyof(i0, x0), valof(i0, x0)
    dict = Dict(k0 ⇒ typeof(v0)[])
    dict = Dict()
    for (i, x) in enumerate(A)
        k, v = keyof(i, x), valof(i, x)
        addentry!(dict, k, v, isequal=isequal)
    end
    dict
```

```
end;

import QuantumOpticsBase.projector
projector(ψ::AbstractVector) = ψ * ψ';

function incentry!(dict, key; isequal=isequal)
    for k in keys(dict)
        if isequal(k, key)
            dict[k] += 1
            return dict
        end
    end
    dict[key] = 1
    dict
end;

function combinejumps(Js)
    d = Dict()
    for J in Js
        incentry!(d, J)
    end
    [√(one(eltype(J)) * N)*J for (J, N) in d]
end;

function jumps(vals, vecs, As; combine=true, isequal=isequal)
    eigendict = dictby(zip(vals, vecs))
    ωs = dictby(((E2 - E1, (E1, E2)) for E1 in keys(eigendict) for E2 in
     ↪ keys(eigendict)), isequal=isequal)
    Πs = TIM.projectors(eigendict)
    Jωs = dictmap(ΔEs -> filter(x -> !isequal(x, zero(x)),
     ↪ [simplify.(sum(Πs[E1]*A*Πs[E2] for (E1, E2) in ΔEs)) for A in As]), ωs)
    combine ? dictmap(combinejumps, Jωs) : Jωs
end
dipolejumps(vals, vecs; kwargs...) = jumps(vals, vecs,
    [σx ⊗ₖ σ0,  σy ⊗ₖ σ0,  σz ⊗ₖ σ0, σ0 ⊗ₖ σx,  σ0 ⊗ₖ σy,  σ0 ⊗ₖ σz];
    kwargs...);

@vars s1p⇒"σ₁⁺" s1m⇒"σ₁⁻" commutative=false
@vars s2p⇒"σ₂⁺" s2m⇒"σ₂⁻" commutative=false
@vars s1x⇒"σ₁ˣ" s2x⇒"σ₂ˣ" commutative=false
@vars s1y⇒"σ₁ʸ" s2y⇒"σ₂ʸ" commutative=false
```

```
@vars s1z⇒"σ₁ᶻ" s2z⇒"σ₂ᶻ" commutative=false
@vars n1 ⇒"n₁"   n2⇒"n₂"  commutative=false
@vars g d real=true;

spinops = [s1p, s1m, s2p, s2m, s1x, s2x, s1y, s2y, s1z, s2z, n1, n2];
_symspinop = cat([n1 s1p; s1m (1 - n1)], [n2 s2p; s2m (1 - n2)], dims=3)
_dummy_spinop = Dict(s ⇒ sympy.Dummy(s.name) for s in spinops);
site_collect_ops(siteops) = push!([_dummy_spinop[op] for op in siteops], 1)
_collect_ops = [a * b for a in site_collect_ops([s1p, s1m, n1]) for b in
 ↪  site_collect_ops([s2p, s2m, n2])];

u, v, w = Wild(:u), Wild(:v), Wild(:w);

symspinop(l, r, i) = _symspinop[l+1, r+1, i];
function jumpsimplify(J)
    s = mapreduce(+, CartesianIndices(J)[J .!= 0]) do I
        x = J[I]
        i, j = Tuple(I - CartesianIndex(1, 1))
        x * symspinop(i÷2, j÷2, 1) * symspinop(i%2, j%2, 2)
    end
    s = subs(expand(s), √(g^2 + 1) ⇒ d)
    s = s.simplify()
    s = s.xreplace(_dummy_spinop)
    for op in _collect_ops
        s = s.collect(op)
    end
    s.simplify()
end;

⇒ₛ(a, b) = _dummy_spinop[a] ⇒ b
thesis_latex(J) = sympy.latex(J,
        imaginary_unit = "\\im",
        symbol_names = Dict(
            g   ⇒  "g",
            d   ⇒  "d",
            n1  ⇒ₛ "\\opr{n}_1",
            n2  ⇒ₛ "\\opr{n}_2",
            s1p ⇒ₛ "\\pauli_1^+",
            s2p ⇒ₛ "\\pauli_2^+",
            s1m ⇒ₛ "\\pauli_1^-",
            s2m ⇒ₛ "\\pauli_2^-",
```

```
        ));

function save_jump_latex(path, sJωs)
    open(path, "w") do file
        for (ω, Js) in sJωs
            sω = factor(subs(expand(ω), √(g^2 + 1) ⇒ d))
            println(file, thesis_latex(sω))
            println(file, " \\\\")
            for J in Js
                println(file, thesis_latex(J))
                println(file, " \\\\")
            end
            println(file, " \\\\")
        end
    end
end;
```

3.4.1 Nondegenerate jump operators

```
H = -2*(σx ⊗ₖ σx) - g*(σz ⊗ₖ σ0 + σ0 ⊗ₖ σz)
```

$$\begin{bmatrix} -2g & 0 & 0 & -2 \\ 0 & 0 & -2 & 0 \\ 0 & -2 & 0 & 0 \\ -2 & 0 & 0 & 2g \end{bmatrix}$$

```
sJωs = dictmap(Js -> jumpsimplify.(Js), dipolejumps(symeigen(H)...;
↪  combine=true))
save_jump_latex("nondegen-jumps.tex", sJωs)
```

3.4.2 Degenerate jump operators

```
H0 = subs.(-2*(σx ⊗ₖ σx) - g*(σz ⊗ₖ σ0 + σ0 ⊗ₖ σz), g ⇒ 0)
```

$$\begin{bmatrix} 0 & 0 & 0 & -2 \\ 0 & 0 & -2 & 0 \\ 0 & -2 & 0 & 0 \\ -2 & 0 & 0 & 0 \end{bmatrix}$$

```
sJωs0 = dictmap(Js -> jumpsimplify.(Js), dipolejumps(symeigen(H0)...;
 ↪  combine=true))
save_jump_latex("degen-jumps.tex", sJωs0)
```

3.5 Liouvillian

3.5.1 Superoperators

```
leftmul(A)  = one(A) ⊗ₖ A
rightmul(A) = permutedims(A) ⊗ₖ one(A)
comm(A, B) = A*B - B*A
acomm(A, B) = A*B + B*A
commwith(A) = leftmul(A) - rightmul(A)
acommwith(A) = leftmul(A) + rightmul(A);

nB(ω, β) = 1 / (exp(β*ω) - 1)
γdiv(ω; β) = isapprox(ω, 0, atol=1e-9) ? 0 : ω^3 * (nB(ω, β) + 1)
ℋLSterm(J, S) = S * commwith(J' * J)
ℋLS(Jωs, S; params...) = -im * sum(ℋLSterm(J, S(ω; params...)) for (ω, Js) in
 ↪  Jωs for J in Js)
𝒟term(J, γ) = γ * (leftmul(J) * rightmul(J') - acommwith(J' * J) / 2)
𝒟(Jωs, γ; params...) = sum(𝒟term(J, γ(ω; params...)) for (ω, Js) in Jωs for J
 ↪  in Js);
𝒟LRterm(J, γ) = γ * (leftmul(J) * rightmul(J'))
𝒟LR(Jωs, γ; params...) = sum(𝒟LRterm(J, γ(ω; params...)) for (ω, Js) in Jωs for
 ↪  J in Js);
𝒟Aterm(J, γ) = γ * (- acommwith(J' * J) / 2)
𝒟A(Jωs, γ; params...) = sum(𝒟Aterm(J, γ(ω; params...)) for (ω, Js) in Jωs for J
 ↪  in Js);
```

```
function numeigen(H)
    vals, vecs = eigen(H)
    vals, eachcol(vecs)
end

isequalto(atol=1e-9) = (x, y) -> isapprox(x, y, atol=atol)

function unitary_rates(H, dict, γ; params...)
    Jωs = dipolejumps(numeigen(N.(subs.(H, dict)))...;
        combine=true, isequal=isequalto())
    lambham = (x -> isequalto()(x, zero(x)) ? zero(x) : x).(ℋLS(Jωs, γ;
     ↪  params...))
    eigvals(lambham)
end

function dissipation_rates(H, dict, γ; params...)
    Jωs = dipolejumps(numeigen(N.(subs.(H, dict)))...;
        combine=true, isequal=isequalto())
    dissipator = (x -> isequalto()(x, zero(x)) ? zero(x) : x).(𝒟(Jωs, γ;
     ↪  params...))
    eigvals(dissipator)
end;

function dissipation_rates(H, γ; params...)
    Jωs = dipolejumps(numeigen(H)...; combine=true, isequal=isequalto())
    dissipator = (x -> isequalto()(x, zero(x)) ? zero(x) : x).(𝒟(Jωs, γ;
     ↪  params...))
    eigvals(dissipator)
end;
```

3.5.2 *Jump operators and commutation*

```
Jωs = dipolejumps(numeigen(-(σx ⊗ₖ σx) - 0*(σz ⊗ₖ σ0 + σ0 ⊗ₖ σz))...;
 ↪  combine=true, isequal=isequalto())
```

```
Dict{Float64,Array{Array{Complex{Float64},2},1}} with 3 entries:
  0.0  => Array{Complex{Float64},2}[[0.0+0.0im 1.11022e-16+0.0im 1.0+0.0im 0.0+…
  -2.0 => Array{Complex{Float64},2}[[0.0+0.0im 0.0-0.5im 0.0-0.5im 0.0+0.0im; 0…
```

```
  2.0  => Array{Complex{Float64},2}[[0.5+0.0im 0.0+0.0im 0.0+0.0im -0.5+0.0im; …
```

```
sum(sum(abs.(comm(J1, J2' * J2))) for Js1 in values(Jωs) for J1 in Js1 for Js2
↪  in values(Jωs) for J2 in Js2)
```

```
256.00000000000006
```

```
sum(sum(abs.(comm(J1' * J1, J2' * J2))) for Js1 in values(Jωs) for J1 in Js1
↪  for Js2 in values(Jωs) for J2 in Js2)
```

```
2.7533531010703882e-14
```

Why do the unitary (Lamb-shift or other) and dissipative superoperators commute?

```
β = 1e-2
lambham = ℋLS(Jωs, (ω; kwargs...) -> rand(), β = β) # Arbitrary S(ω)
diss = 𝒟(Jωs, γdiv, β = β)
disslr = 𝒟LR(Jωs, γdiv, β = β)
dissa = 𝒟A(Jωs, γdiv, β = β);
```

```
vals, vecs = eigen(diss)
ρst = reshape(vecs[:, end], (4, 4))
ρst /= tr(ρst)
real(vals)
```

```
16-element Array{Float64,1}:
 -3200.10666595557
 -1616.0533329777838
 -1616.0533329777836
 -1616.0533329777827
 -1600.0533329777843
 -1600.0533329777838
 -1600.0533329777836
 -1600.0533329777834
 -1600.0533329777834
 -1600.053332977783
 -1600.0533329777827
 -1600.0533329777827
 -1584.0533329777845
 -1584.0533329777836
 -1584.0533329777836
     3.402833570476105e-14
```

```
sum(abs, diss'  - diss)
```

```
255.9999999999844
```

```
eigvals((x -> (x' + x)/2)(rand(4, 4)))
```

```
4-element Array{Float64,1}:
 -0.38422479522604447
 -0.05135214229868798
  0.9085668376203021
  2.241724276596741
```

```
length(diss)^(1//4)
```

```
4.0
```

```
sum(abs(tr(reshape(diss * x, Int(√(size(diss)[1])), :))) for x in
↪  eachcol(Matrix(I, size(diss)...)))
```

```
5.115907697472721e-13
```

```
sum(abs(tr(reshape(vals[i] * x, Int(√(length(x))), :))) for (i, x) in
↪  enumerate(eachcol(vecs)))
```

```
2.7752196124678597e-12
```

None of the eigenvectors correspond to positive, Hermitian operators, except for the always stable state.

```
for ρ in eachcol(vecs[:,end-3:end])
    ρ0 = reshape(ρ, (4, 4))
    display(eigen(real(ρ0 / tr(ρ0))))
end
```

```
Eigen{Float64,Float64,Array{Float64,2},Array{Float64,1}}
values:
4-element Array{Float64,1}:
  -5.737322230551043e15
 -52.342948063813516
  52.34421597333877
   5.737322230551046e15
vectors:
4×4 Array{Float64,2}:
 -0.380044  -0.352762   0.504322  0.338341
```

```
  0.596294   0.612829   0.49564   0.620907
  0.596294  -0.612829  -0.49564   0.620907
 -0.380044   0.352762  -0.504322  0.338341

Eigen{Float64,Float64,Array{Float64,2},Array{Float64,1}}
values:
4-element Array{Float64,1}:
 -2.123017224175257e15
 -7.028669182732243
  8.104815207140181
  2.123017224175256e15
vectors:
4×4 Array{Float64,2}:
  0.707107      0.00740933  -0.659096   2.1026e-16
 -5.93617e-16  -0.707068    -0.256109   0.707107
  2.59664e-15   0.707068     0.256109   0.707107
  0.707107     -0.00740933   0.659096  -3.04953e-16

Eigen{Complex{Float64},Complex{Float64},Array{Complex{Float64},2},Array{Complex{Float
values:
4-element Array{Complex{Float64},1}:
                 0.0 - 4.534028352184157e15im
                 0.0 + 4.534028352184157e15im
 0.42388699980216415 - 79.15815120249324im
 0.42388699980216415 + 79.15815120249324im
vectors:
4×4 Array{Complex{Float64},2}:
   0.636807-6.39026e-15im  …   0.119039+0.466273im
 -0.0911605+0.293541im        -0.518092-4.81718e-15im
 -0.0911605+0.293541im         0.518092+0.0im
   0.636807-0.0im             -0.119039-0.466273im

Eigen{Float64,Float64,Array{Float64,2},Array{Float64,1}}
values:
4-element Array{Float64,1}:
 0.24750008333000023
 0.24750008333000043
 0.2524999166699996
```

```
 0.25249991666999966
vectors:
4×4 Array{Float64,2}:
 -0.707107      0.0971961   0.309007  -0.707107
 -3.14018e-16  -0.700395   -0.636015   3.14018e-16
  3.70346e-16   0.700395   -0.636015  -3.70346e-16
  0.707107     -0.0971961   0.309007  -0.707107
```

```
v0 = reshape(vecs[:,end], 4, 4)
v0 /= tr(v0)
v0vals, v0vecs = eigen(v0)
```

```
Eigen{Complex{Float64},Complex{Float64},Array{Complex{Float64},2},Array{Complex{Float64},1}
values:
4-element Array{Complex{Float64},1}:
 0.24750008333000015 - 0.0im
  0.2475000833300002 + 0.0im
  0.2524999166699996 + 0.0im
 0.25249991666999966 + 0.0im
vectors:
4×4 Array{Complex{Float64},2}:
     0.707107+0.0im    0.30821+0.0im  -0.309723+0.0im      0.707107+0.0im
  3.14018e-16-0.0im  -0.636402+0.0im   0.635667+0.0im  -3.14018e-16+0.0im
 -3.70346e-16-0.0im   0.636402+0.0im   0.635667+0.0im   3.70346e-16+0.0im
    -0.707107-0.0im   -0.30821+0.0im  -0.309723+0.0im      0.707107+0.0im
```

```
tr((σz ⊗ₖ σ0 + σ0 ⊗ₖ σz)' * (σz ⊗ₖ σ0 + σ0 ⊗ₖ σz))
```

```
8
```

```
ψ1 = [0 1 1 0] / √2
ψ2 = [1 0 0 1] / √2
ρ1, ρ2 = ψ1'*ψ1, ψ2'*ψ2;
```

```
diss * vec(ρ1)
```

```
16-element Array{Complex{Float64},1}:
   396.0133332444459 + 0.0im
                 0.0 + 0.0im
                 0.0 + 0.0im
   -396.013333244446 + 0.0im
```

```
                0.0 + 0.0im
   -396.013333244446 + 0.0im
 -1188.0399997333382 + 0.0im
                0.0 + 0.0im
                0.0 + 0.0im
 -1188.0399997333382 + 0.0im
 -396.01333324444624 + 0.0im
                0.0 + 0.0im
   -396.013333244446 + 0.0im
                0.0 + 0.0im
                0.0 + 0.0im
   396.0133332444462 + 0.0im
```

```
diss * vec(ρ2)
```

```
16-element Array{Complex{Float64},1}:
 -396.01333324444624 + 0.0im
                0.0 + 0.0im
                0.0 + 0.0im
  -1188.039999733338 + 0.0im
                0.0 + 0.0im
   396.0133332444461 + 0.0im
   -396.013333244446 + 0.0im
                0.0 + 0.0im
                0.0 + 0.0im
   -396.013333244446 + 0.0im
   396.0133332444459 + 0.0im
                0.0 + 0.0im
  -1188.039999733338 + 0.0im
                0.0 + 0.0im
                0.0 + 0.0im
 -396.01333324444573 + 0.0im
```

```
v0' - v0
```

```
4×4 Array{Complex{Float64},2}:
          0.0+0.0im  -1.28727e-18-0.0im  …   8.67362e-19+0.0im
  1.28727e-18-0.0im           0.0+0.0im       6.3636e-21-0.0im
```

```
 -1.30933e-18+0.0im  -1.73472e-18+0.0im     -8.76772e-21+0.0im
 -8.67362e-19+0.0im   -6.3636e-21-0.0im              0.0-0.0im
```

```
v0vecs' * v0vecs
```

```
4×4 Array{Complex{Float64},2}:
          1.0+0.0im     0.435874+0.0im  …  -5.59158e-15+0.0im
     0.435874-0.0im          1.0+0.0im       2.8046e-15+0.0im
 -2.48451e-15-0.0im  1.16293e-15-0.0im        -0.438014+0.0im
 -5.59158e-15-0.0im   2.8046e-15-0.0im              1.0+0.0im
```

Compare thermal states:

```
trnorm2(A) = tr(A' * A);
```

```
trnorm2(Hx), trnorm2(Hz / √2)
```

```
UndefVarError: Hx not defined

Stacktrace:
 [1] top-level scope at In[43]:1
```

```
Hz = -(σz ⊗ₖ σ0 + σ0 ⊗ₖ σz)
ρthz = exp(-β*Hz)
ρthz /= tr(ρthz)
```

```
4×4 Array{Float64,2}:
 0.255025  0.0       0.0       0.0
 0.0       0.249975  0.0       0.0
 0.0       0.0       0.249975  0.0
 0.0       0.0       0.0       0.245025
```

```
Hx = -(σx ⊗ₖ σx)
ρth = exp(-β*Hx)
ρth /= tr(ρth)
```

```
4×4 Array{Float64,2}:
 0.25        0.0         0.0         0.00249992
 0.0         0.25        0.00249992  0.0
 0.0         0.00249992  0.25        0.0
 0.00249992  0.0         0.0         0.25
```

```
tr(Hz*exp(-Hz)), tr(Hx*exp(-Hx))
```

```
(-14.507441631388076, -4.7008047745752055)
```

```
Hint(a) = (1-a)*Hx + a*Hz / √2;
```

```
plot(a -> tr(Hint(a)*exp(-β*Hint(a))), xlim=(0, 1))
```

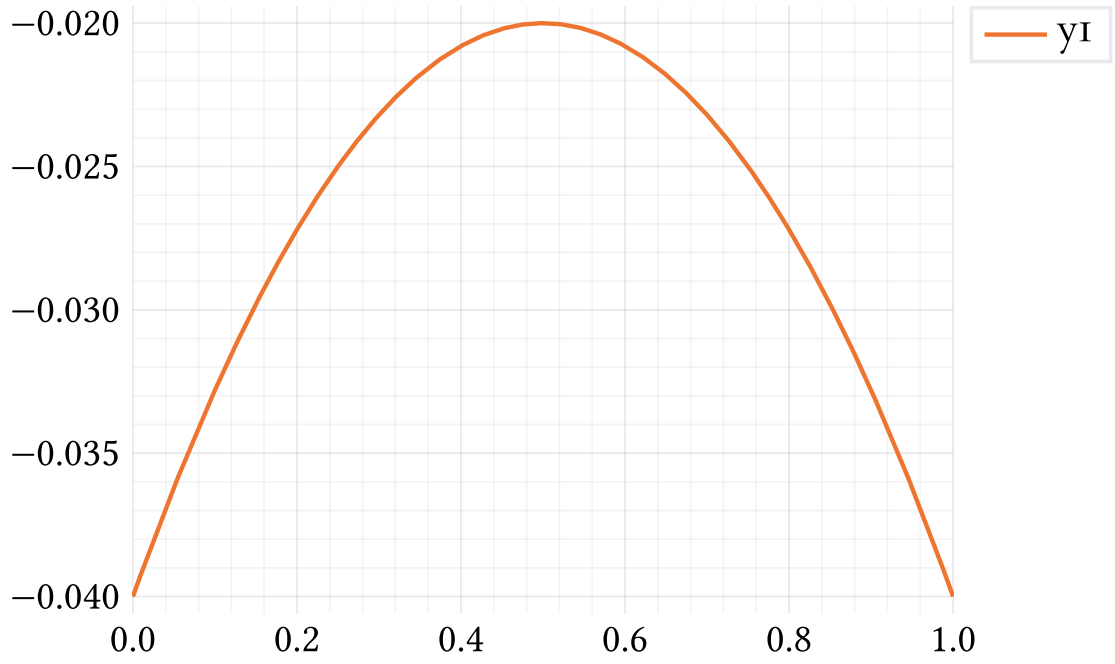

Figure 3.23

```
reshape(diss * vec(ρth), (4, 4))
```

```
4×4 Array{Complex{Float64},2}:
          0.0+0.0im           0.0+0.0im  …  -4.03455e-13+0.0im
          0.0+0.0im           0.0+0.0im              0.0+0.0im
          0.0+0.0im  -4.73732e-13+0.0im              0.0+0.0im
 -4.03233e-13+0.0im           0.0+0.0im      5.68434e-14+0.0im
```

```
eigen(ρth)
```

```
Eigen{Float64,Float64,Array{Float64,2},Array{Float64,1}}
values:
4-element Array{Float64,1}:
 0.24750008333000018
```

```
 0.24750008333000018
 0.2524999166699998
 0.2524999166699998
vectors:
4×4 Array{Float64,2}:
  0.0       -0.707107  0.0       0.707107
  0.707107   0.0       0.707107  0.0
 -0.707107   0.0       0.707107  0.0
  0.0        0.707107  0.0       0.707107
```

The thermal state is a linear combination of eigenmodes which are not necessarily positive (do not admit interpretation as density operators).

```
vec(ρth)' * vecs[:,end-3:end]
```

```
1×4 Adjoint{Complex{Float64},Array{Complex{Float64},1}}:
 2.43455e-17-0.0im  -7.76289e-17-0.0im  -2.67437e-17-0.0im  0.500025-0.0im
```

```
diss' - diss .▷ abs ▷ sum
```

```
255.99999999998434
```

```
disslr' - disslr .▷ abs ▷ sum
```

```
255.99999999998272
```

The LR-part of the dissipator is not Hermitian.

```
comm(lambham, diss) .▷ abs ▷ sum
```

```
8.057665376938247e-12
```

```
comm(lambham, dissa) .▷ abs ▷ sum
```

```
3.639421096592869e-12
```

The Lamb-shift and dissipative superoperators commute.

```
(im*lambham)' - (im*lambham) .▷ abs ▷ sum
```

```
0.0
```

The full Lamb-shift superoperator is Hermitian, as expected.

```
comm(lambham, disslr) .▷ abs ▷ sum
```

```
5.304619880528804e-12
```

```
comm(lambham - im*commwith(-0.5*(σx ⊗ₖ σx) - (σz ⊗ₖ σ0 + σ0 ⊗ₖ σz)), diss) .▷
↪  abs ▷ sum
```

```
19840.63999573337
```

```
Js = vcat(values(Jωs)...);
```

```
Ts = [rand(4, 4) for _ in 1:(length(Js) / 2)]
# Ts = (T -> (T + T') / 2).(Ts)
Ts = [Ts; (T -> T').(Ts)];
```

```
(Ts -> sum(abs, comm(sum(T -> commwith(T' * T), Ts), sum(T -> leftmul(T) *
↪  rightmul(T') - acommwith(T)/2, Ts)))).([Ts, Js])
```

```
2-element Array{Float64,1}:
 1691.2295023217935
    8.312762260824573e-14
```

Since the $A_i^\dagger A_i$ commute with each other:

```
(Ts -> sum(sum(abs, comm(T1'*T1, T2'*T2)) for T1 in Ts for T2 in Ts))(Js)
```

```
2.7533531010703882e-14
```

But

```
(Ts -> sum(sum(abs, comm(T1'*T1, T2)) for T1 in Ts for T2 in Ts))(Js)
```

```
256.0000000000001
```

so there must be a different reason why the LR term commutes with the unitary superoperator, as shown here for the Lamb-shift term:

```
(Ts -> sum(sum(abs, leftmul(T1'*T1*T2)*rightmul(T2') -
↪  leftmul(T2)*rightmul(T2'*T1'*T1)
           - leftmul(T2*T1'*T1)*rightmul(T2') +
            ↪  leftmul(T2)*rightmul(T1'*T1*T2'))
       for (T1, T2) in Iterators.product(Ts, Ts)))(Js)
```

```
3.6548541970660153e-13
```

```
(Ts -> sum(sum(abs, leftmul(T1'*T1*T2)*rightmul(T2') -
↪  leftmul(T2)*rightmul(T2'*T1'*T1)
           - leftmul(T2*T1'*T1)*rightmul(T2') +
            ↪  leftmul(T2)*rightmul(T1'*T1*T2'))
       for (T1, T2) in Iterators.product(Ts, Ts)))(Js)
```

```
3.6548541970660153e-13
```

```
(Ts -> sum(sum(abs, comm(commwith(T1'*T1), leftmul(T2)*rightmul(T2')))
        for (T1, T2) in Iterators.product(Ts, Ts)))(Js)
```

```
3.1619151741324463e-13
```

```
sum(abs.(lambham*diss - diss*lambham))
```

```
8.057665376938247e-12
```

The system Hamiltonian must be compatible.

```
ℋx = commwith(σx ⊗ₖ σy)
sum(abs.(ℋx*diss - diss*ℋx))
```

```
25984.853327644527
```

```
unitary_rates(-g*(σx ⊗ₖ σx) - Sym(1)*(σz ⊗ₖ σ0 + σ0 ⊗ₖ σz), g ⇒ 0.5, (ω;
 ↪  kwargs...) -> ω, β = β)
```

```
16-element Array{Complex{Float64},1}:
  -7.105427357601002e-15 + 17.462565002615 99im
  -5.329070518200751e-15 + 12.731282501307998im
  -9.227884241529766e-16 - 8.000000000000004im
   -4.98851266322101e-16 - 17.462565002615975im
  -3.551676818936971e-16 + 5.018048615358172e-17im
 -1.1092188935359755e-16 + 8.000000000000005im
 -2.0328790734103208e-20 + 12.73128250130797im
   6.809297825563993e-18 - 4.7312825013079856im
  3.2504381131115423e-16 + 4.731282501307987im
  4.1881535493684293e-16 - 2.6582660060945874e-16im
    4.44090998525497e-16 + 4.731282501307 99im
   8.925966649376186e-16 - 4.731282501307985im
  1.0009190718918834e-15 - 1.8267373944428236e-16im
  1.0122187765602485e-15 + 1.4434433083475654e-16im
   3.552713678800501e-15 - 12.731282501307987im
   6.862566070964249e-15 - 12.731282501308im
```

```
dissipation_rates(-g*(σx ⊗ₖ σx) - Sym(1)*(σz ⊗ₖ σ0 + σ0 ⊗ₖ σz), g ⇒ 0.5, γdiv,
 ↪  β = β)
```

```
16-element Array{Complex{Float64},1}:
```

```
      -3600.2820688037864 + 0.0im
      -3024.7533057095684 + 0.0im
      -2019.4659823157658 + 0.0im
       -2019.465982315764 + 0.0im
      -2005.4659823157656 + 0.0im
      -2005.4659823157656 + 0.0im
      -2000.1099983722622 + 0.0im
      -2000.1099983722602 + 0.0im
      -2000.0833325055642 + 0.0im
      -2000.0833325055637 + 0.0im
      -1994.7273485620638 + 0.0im
      -1994.7273485620617 + 0.0im
      -1980.7273485620644 + 0.0im
      -1980.7273485620588 + 0.0im
       -1375.351287242299 + 0.0im
 -1.4432293251112616e-12 + 0.0im
```

3.5.3 *Dissipator eigenvalue plots*

```
rubric = RGB(0.7, 0.05, 0.0);
```

```
"`entropy(H, β)` is the entropy of the thermal state in bits."
function entropy(H, β)
    Es = eigvals(H)
    Z = sum(exp(-β*E) for E in Es)
    Ps = [exp(-β*E) / Z for E in Es]
    -sum(iszero(P) ? zero(P) : P*log2(P) for P in Ps)
end

function energyvariance(H, β)
    Es = eigvals(H)
    Z = sum(exp(-β*E) for E in Es)
    Ps = [exp(-β*E) / Z for E in Es]
    H1 = sum(E*P for (E, P) in zip(Es, Ps))
    H2 = sum(E^2*P for (E, P) in zip(Es, Ps))
    H2 - H1^2
end;
```

```
function ηenergyvariance(H, η)
    Es = eigvals(H) / η
    Z = sum(exp(-E) for E in Es)
    Ps = [exp(-E) / Z for E in Es]
    H1 = sum(E * P for (E, P) in zip(Es, Ps))
    H2 = sum(E^2 * P for (E, P) in zip(Es, Ps))
    H2 - H1^2
end;
```

We map $E \mapsto E/\eta$ so that η is a dimensionless inverse temperature β.

```
plot(T -> energyvariance(Hx, 1/T), xlim=(1e-1, 1e2), xscale=:log10)
```

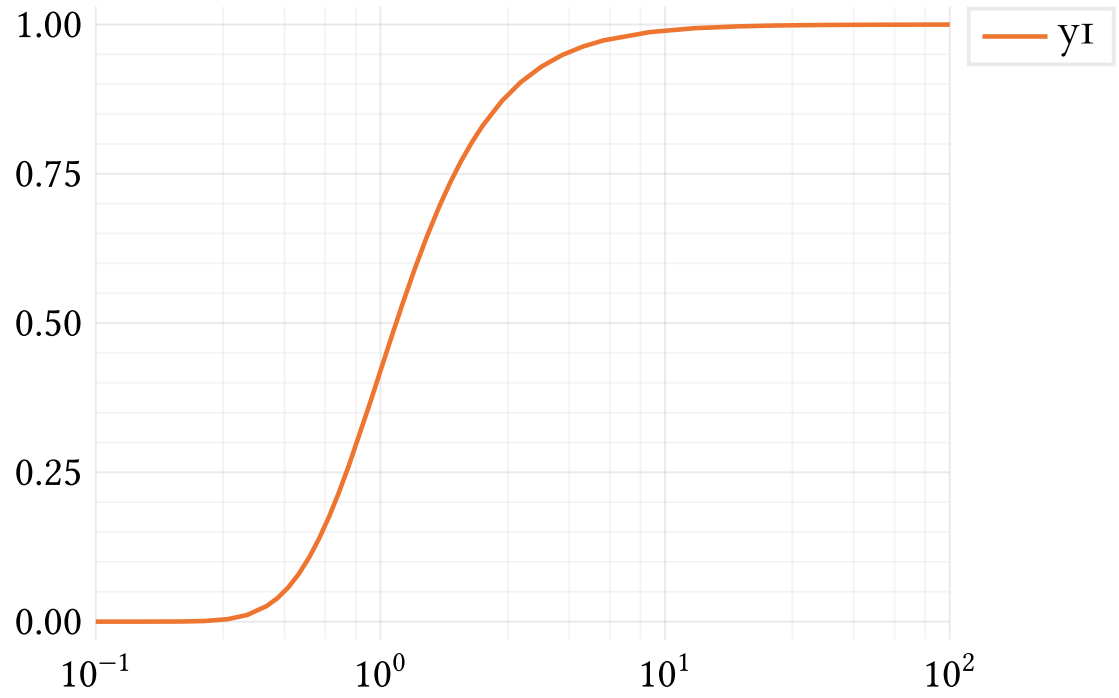

Figure 3.24

```
plot(η -> ηenergyvariance(Hx, η), xlim=(1e-1, 1e2), xscale=:log10)
```

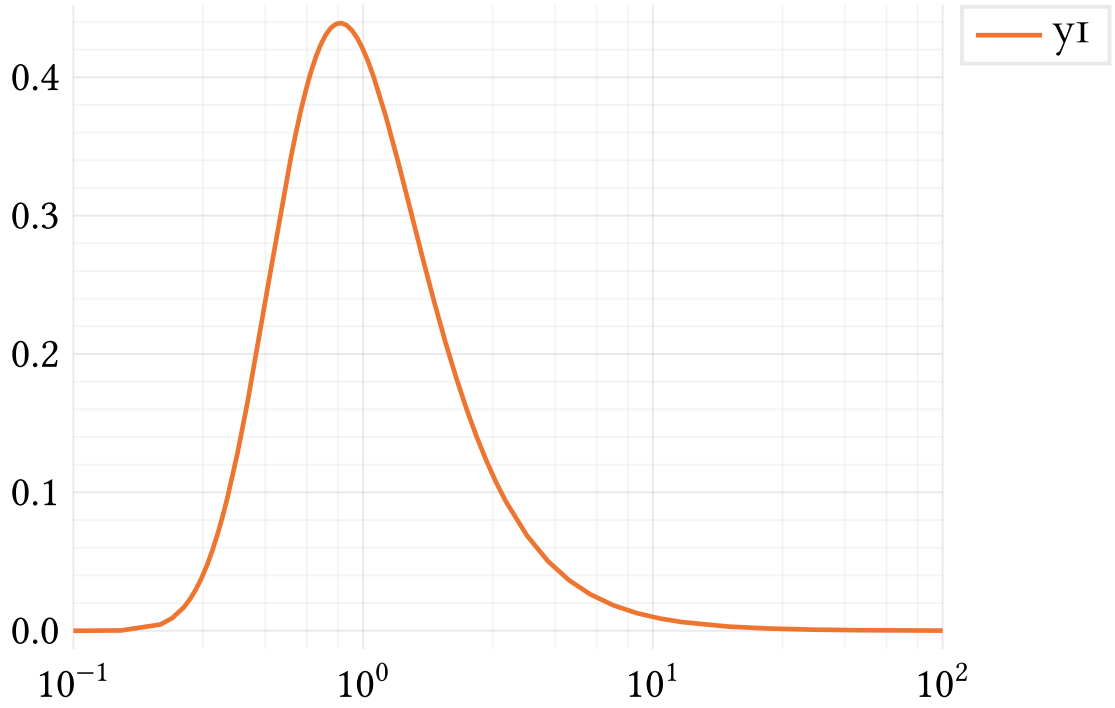

Figure 3.25

```
using Roots
```

```
ηenergyvariance(Hx, 1), ηenergyvariance(2*Hx, 2)
```

```
(0.41997434161402614, 0.41997434161402614)
```

```
interpscale(g) = √(8*(g - 1/2)^2 + 2)
```

```
trnorm(A) = √(tr(A'*A))
trnormalize(A) = A / trnorm(A)
∠(A, B) = acos(trnorm(trnormalize(A)' * trnormalize(B)))
function slerp(A, B, g)
    θ = ∠(A, B)
    (sin((1-g)*θ)*A + sin(g*θ)*B) / sin(θ)
end

# Requirement: `normslerp` must not change the reference Hamiltonian.
function normslerp(A, B, g)
    C = slerp(A, B, g)
    C /= trnorm(C)
end
```

```
# normslerp(H1, H2, g) = ((1-g)*H1 + √(2)*g*H2) #/ interpscale(g)
# normslerp(H1, H2, g) = ((1-g)*H1 + g*H2)

Hint0(g) = normslerp(trnormalize(-Hx), trnormalize(-Hz), g)

# Is this confused between A and η?
# Ainvar(g, η) = find_zero(A -> ηenergyvariance(Hint0(0), η) -
 ↪  ηenergyvariance(A*Hint0(g), η), η)
Ainvar(g) = find_zero(A -> ηenergyvariance(Hint0(0), 1) -
 ↪  ηenergyvariance(Hint0(g), 1/A), 1)
Ainvar(g, η) = Ainvar(g)
# Ainvar(g, β) = 1
Hint(g, η) = Ainvar(g, η) * Hint0(g)

function plot_varrel_rates(g0s, β; kwargs...)
    η = β
    β = 1
    H(g) = Hint(g, η)
    rates_ising = [real(dissipation_rates(H(g), γdiv, β = β)) for g in g0s]
    rates_free_ising = real(dissipation_rates(H(0), γdiv, β = β))[1:end-0];
    rates_free_trans = real(dissipation_rates(H(1), γdiv, β = β))[1:end-0];

    plot(g0s, -hcat(rates_ising...)'[:,1:end-0],
#         title=L"\eta = %η",
        xlabel=L"Relative angle $g$",
        ylabel="Relative dissipator eigenvalues (negated)",
        color=:black,
        alpha=0.25,
        key=false;
        kwargs...)

    scatter!(repeat([g0s[1] - 2e-2], length(rates_free_ising)),
     ↪  -rates_free_ising,
        marker=(:rtriangle, 2, rubric),
        markerstrokecolor=rubric
    )
    scatter!(repeat([g0s[end] + 2e-2], length(rates_free_trans)),
     ↪  -rates_free_trans,
```

```
        marker=(:ltriangle, 2, rubric),
        markerstrokecolor=rubric
    )
end

function plot_η_rates(η0s, g; kwargs...)
    rates = [real(dissipation_rates(Hint(g, η), γdiv, β = 1)) for η in η0s]
    plot(η0s, -hcat(rates...)'[:,1:end-1],
        title=L"g = %$g",
        xlabel=L"\eta",
        color=:black,
        alpha=0.25;
        kwargs...)
end;

import Plots.PlotMeasures: pt;

function rate_plot(g0s, βs, f; kwargs...)
    cols = 3
    width = 800
    s = width // cols
    d = length(βs) // cols
    plot([f(g0s, β) for β in βs]...,
        layout=(:, cols),
        size=(width, s*d),
        ylabel="Nonzero dissipator eigenvalues (negated)",
        top_margin=6pt,
        bottom_margin=6pt,
        yscale=:log10,
        key=false;
        kwargs...
    )
end;

η0s = 10 .^ range(-1, 1.5, length=128)
βs = [1e-1, 2e-1, 5e-1, 1e0, 2e0, 5e0, 1e1, 2e1, 5e1];
```

TODO: Why are the rates not constant over η? Heat capacity issue at low/high temperatures?

Cause: `Ainvar` at low η is not 1, but should always be 1 for $g = 0$ (reference).

TODO: Make one plot against g, since nondim. diss. don't change with η (for $A = 1$)? For `Ainvar`?

(This is just the last grid `rate_plot`). What happens to zero-tending eigenvalues?

```
# rate_plot(η0s, [0, 0.5, 1], plot_η_rates, link=:all, xscale=:log10)
# rate_plot(range(1e-3, 1-1e-3, length=64), βs, plot_varrel_rates)
```

Slerp

```
plot_varrel_rates(range(1e-3, 1-1e-3, length=64), 1)
```

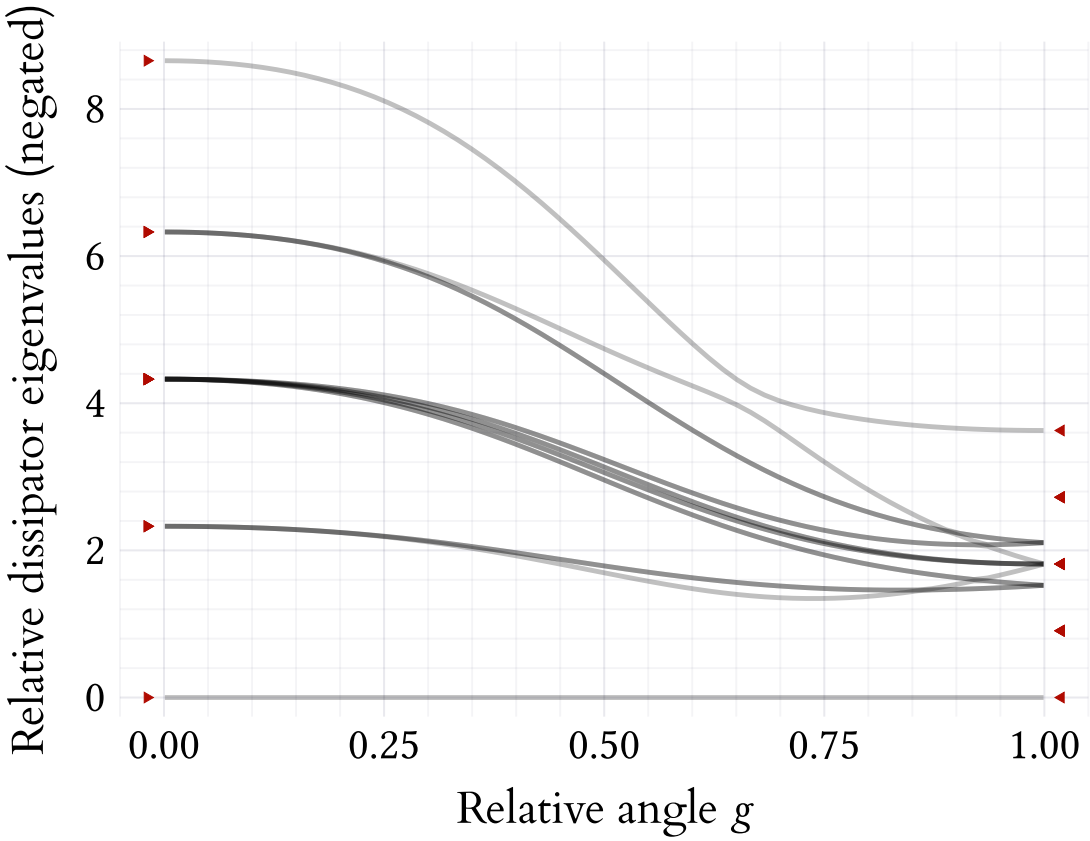

Figure 3.26

Slerp with 0.25 on H_z

```
plot_varrel_rates(range(1e-3, 1-1e-3, length=64), 1)
```

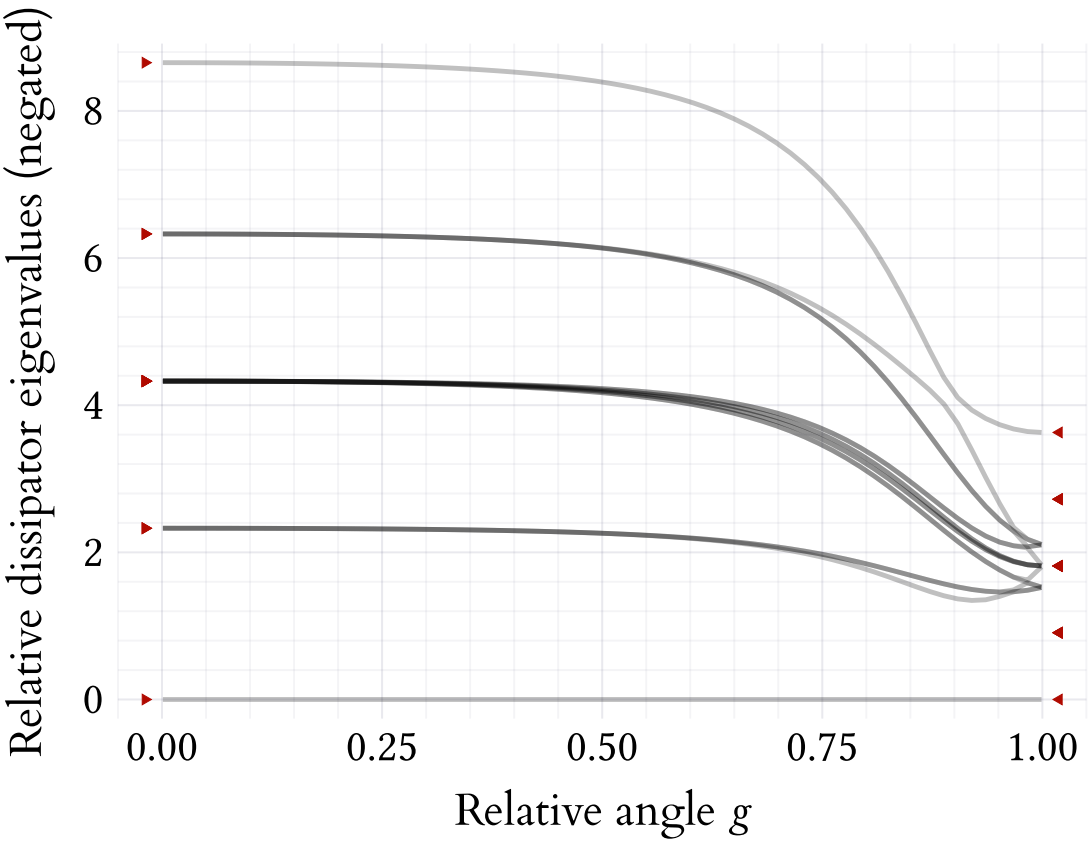

Figure 3.27

Lerp

```
plot_varrel_rates(range(1e-3, 1-1e-3, length=64), 1)
```

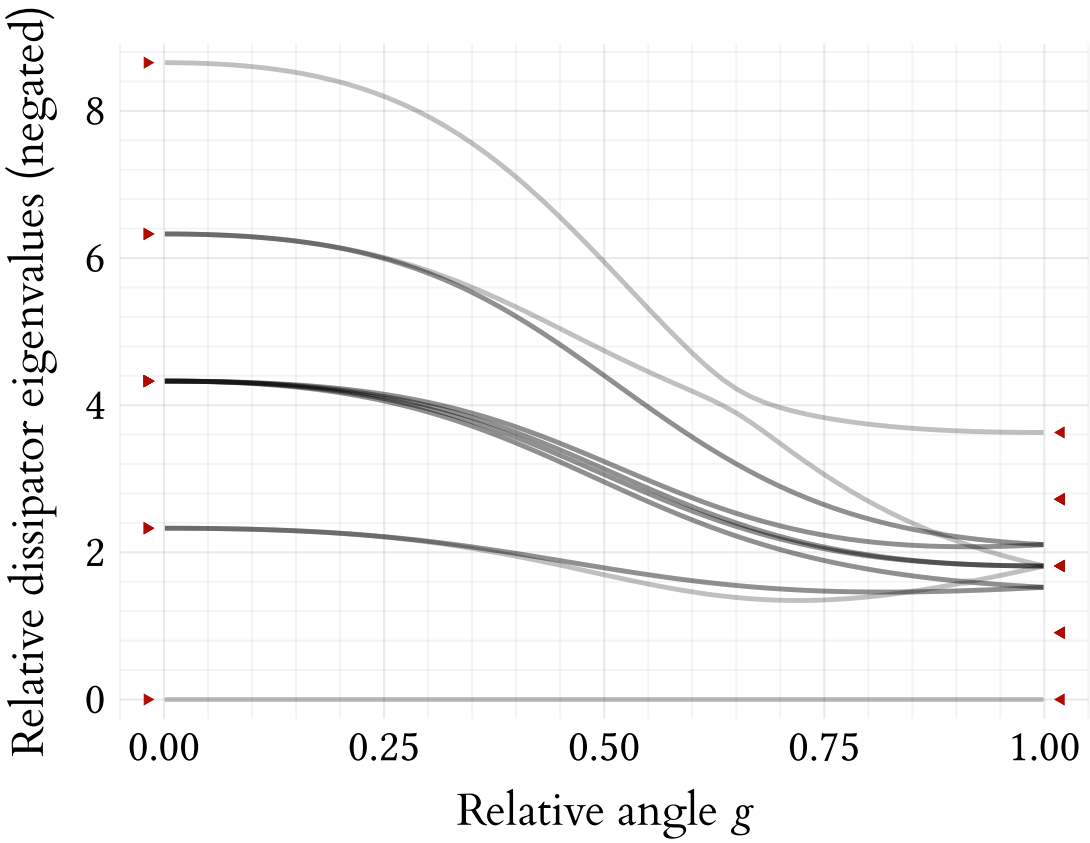

Figure 3.28

Lerp with 0.25 on H_z

```
plot_varrel_rates(range(1e-3, 1-1e-3, length=64), 1)
```

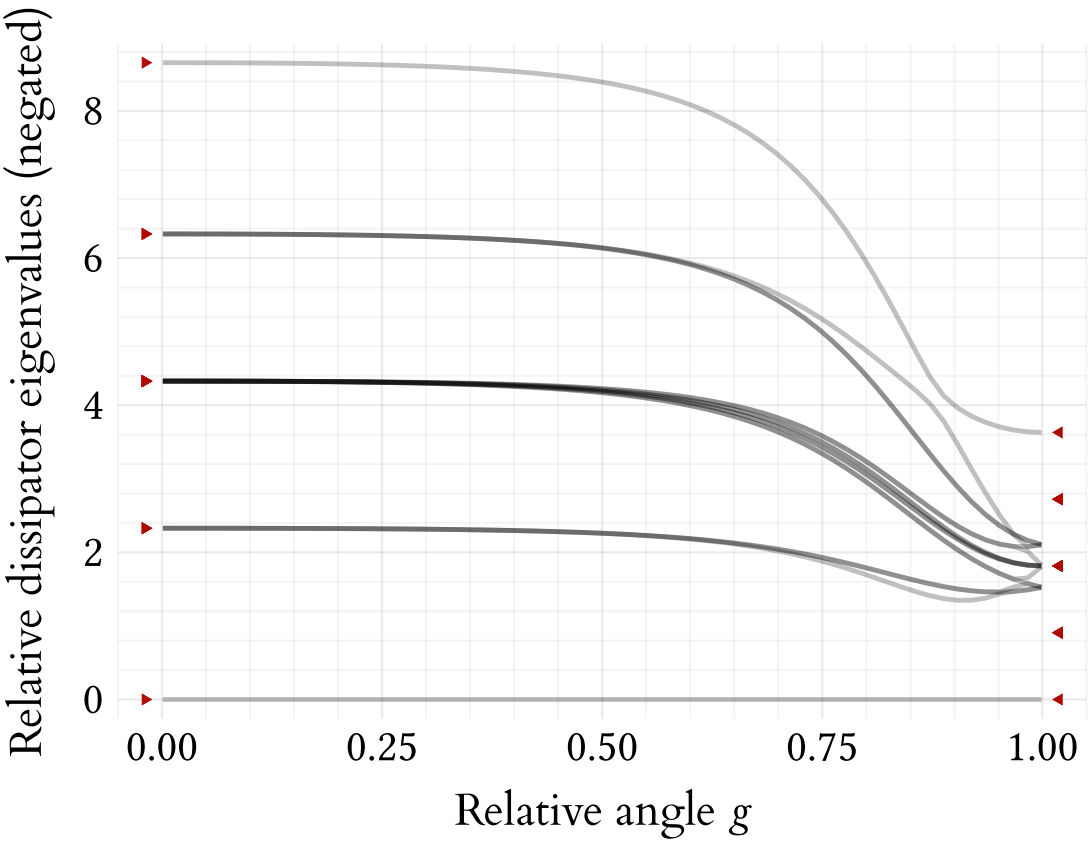

Figure 3.29

Lerp with 2 on H_z

```
plot_varrel_rates(range(1e-3, 1-1e-3, length=64), 1)
```

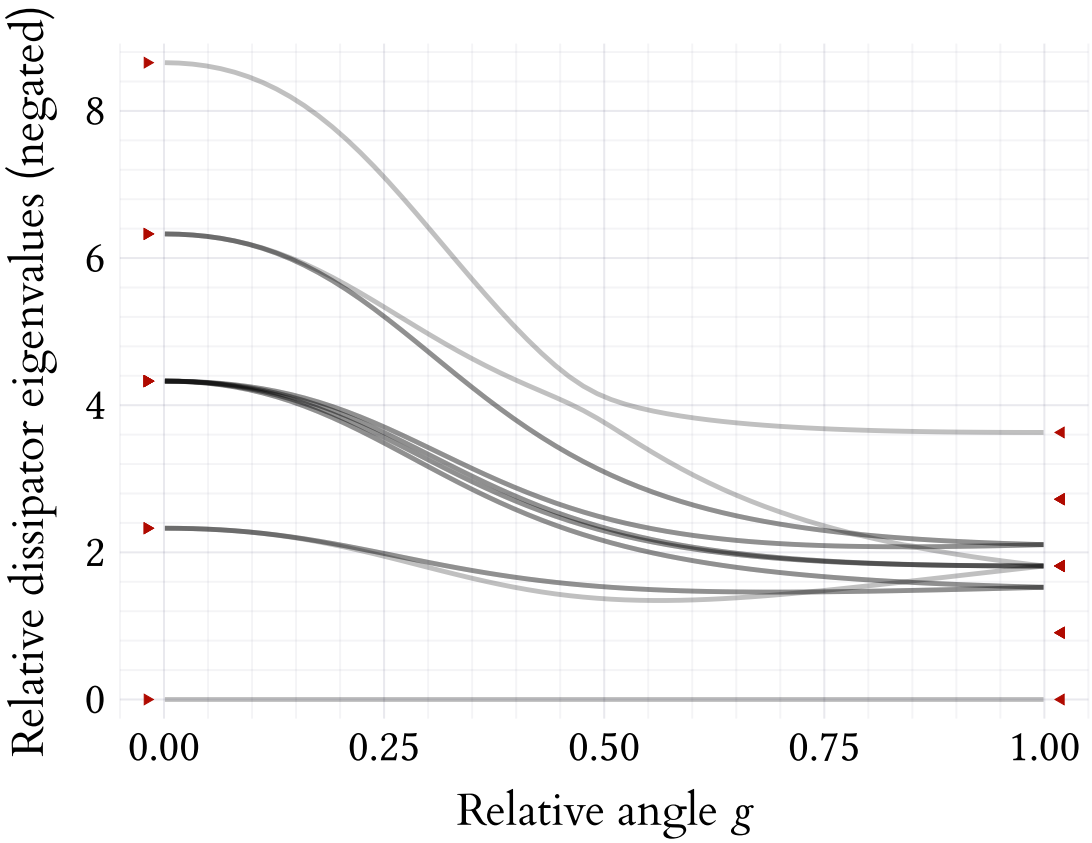

Figure 3.30

Now look at the unnormalized interpolation.

```
lgs = range(1e-3, 1-1e-3, length=64)
energies = [eigvals(((1 - g)*Hx/trnorm(Hx) + √(2)*g*Hz/trnorm(Hz)) /
 ↪  interpscale(g)) for g in lgs]
Hnorms = [trnorm2(((1 - g)*Hx/trnorm(Hx) + √(2)*g*Hz/trnorm(Hz)) /
 ↪  interpscale(g)) for g in lgs]
energydifferences = [[x - y for (x, y) in Iterators.product(Es, Es)] for Es in
 ↪  energies];

energies = [eigvals(Hint(g, 1e0)) for g in lgs];

plot(lgs, hcat(energies...)', color=:black, key=false, xlabel=L"g",
 ↪  ylabel=L"E")
```

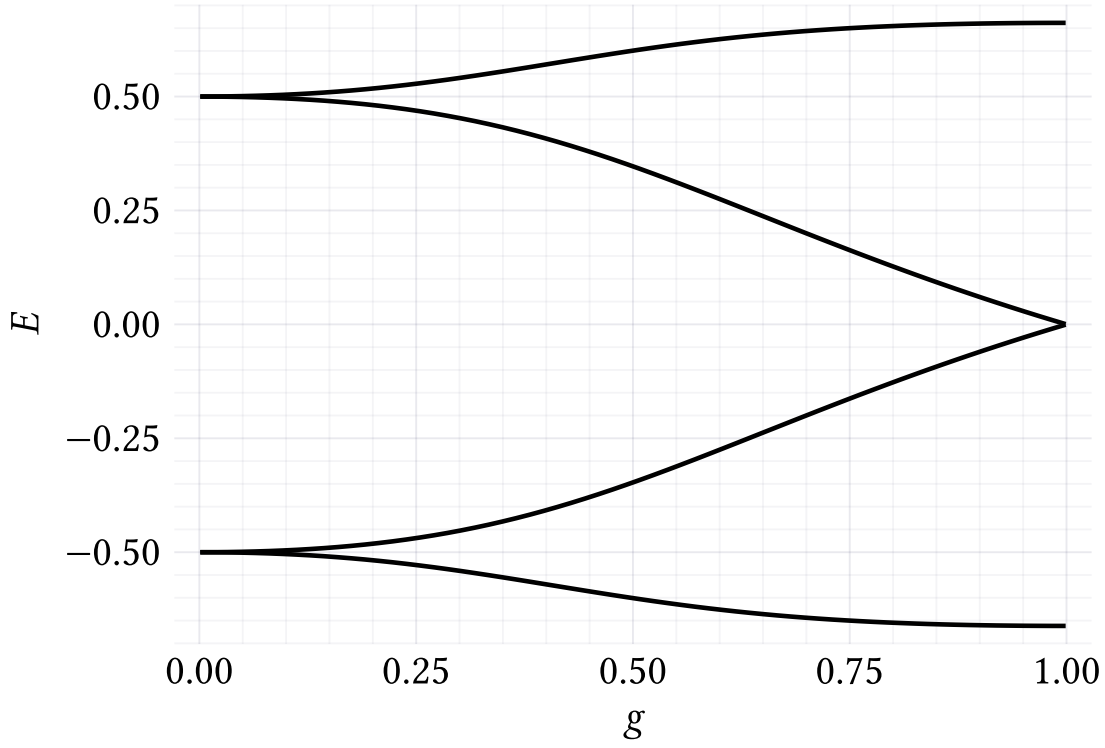

Figure 3.31

```
# https://github.com/JuliaDiff/ForwardDiff.jl/issues/111#issuecomment-638251496
import ForwardDiff: Dual
import LinearAlgebra: eigvals

function LinearAlgebra.eigvals(A::Symmetric{<:Dual{Tg,T,N}}) where
 ↪ {Tg,T<:Real,N}
    λ,Q = eigen(Symmetric(getproperty.(parent(A), :value)))
    partials = ntuple(j → diag(Q' * getindex.(getproperty.(A, :partials), j) *
     ↪ Q), N)
    Dual{Tg}.(λ, tuple.(partials...))
end;
```

TODO: Fix eigvals redefinition issue

```
Hlerp(g) = (1-g)*trnormalize(Hx) + g*trnormalize(Hz)
lerpEs(g) = eigvals(Symmetric(Hlerp(g)));

# This hangs:
# derivative(g → eigvals(Symmetric(Hlerp(g))), 0.5)
```

```
import ForwardDiff: derivative
derivative(f) = x -> derivative(f, x);


plot(g -> energyvariance(normslerp(trnormalize(Hx), trnormalize(Hz), g, βA),
 ↪  βA),
    xlim=(0, 1),
    xlabel=L"g",
    ylabel=L"\ev{\ham^2}_c",
    key=false,
)
```

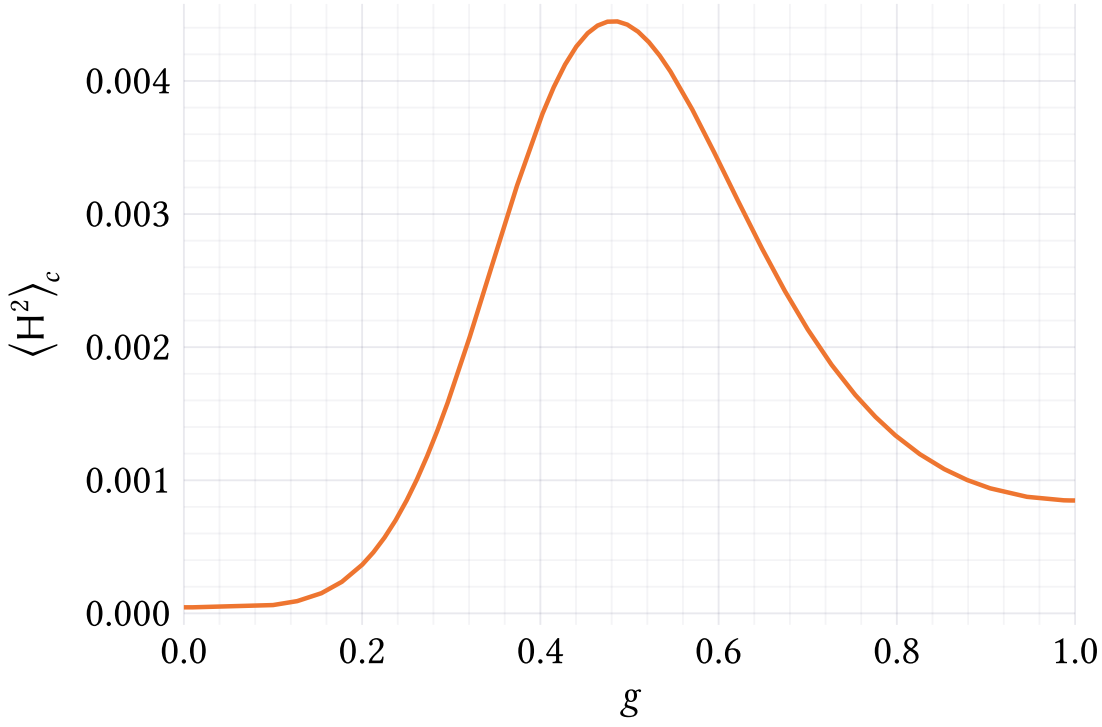

Figure 3.32

```
β0 = 1e0
η0 = 1e0
plot([A -> ηenergyvariance(A*Hint0(g), η0) for g in 0:0.25:1], xlim=(0, 15*η0),
 ↪  xlabel=L"A", ylabel=L"\ev{H^2}_\eta")
```

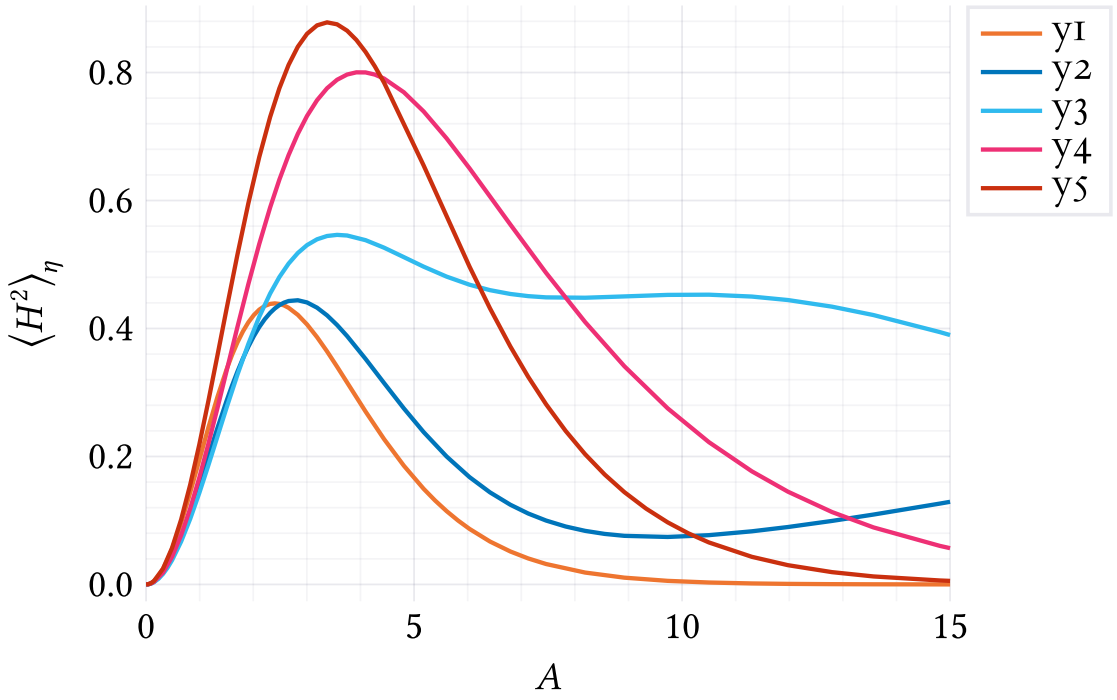

Figure 3.33

```
β0 = 1e0
η0 = 2e-1
# plot([A -> ηenergyvariance(A*Hint0(g), η0) for g in 0:0.25:1], xlim=(0,
↪  15*η0), xlabel=L"A", ylabel=L"\ev{H^2}_\eta")
plot([A -> ηenergyvariance(Hint0(0), η0) - ηenergyvariance(A*Hint0(g), η0) for
↪  g in 0:0.25:1], xlim=(0, 50*η0), xlabel=L"A", ylabel=L"\ev{H^2}_\eta")
```

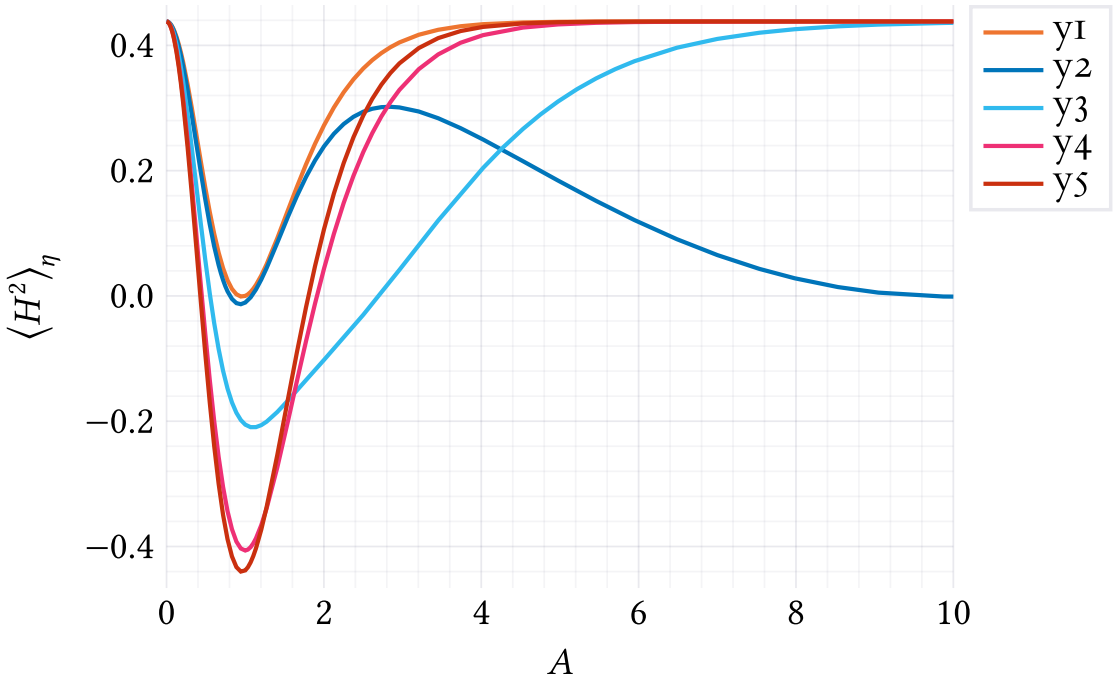

Figure 3.34

```
plot(A -> ηenergyvariance(Hint0(0), η0) - ηenergyvariance(A*Hint0(2e-2), η0),
 ↪  xlim=(0, 50*η0), xlabel=L"A", ylabel=L"\ev{H^2}_\eta")
```

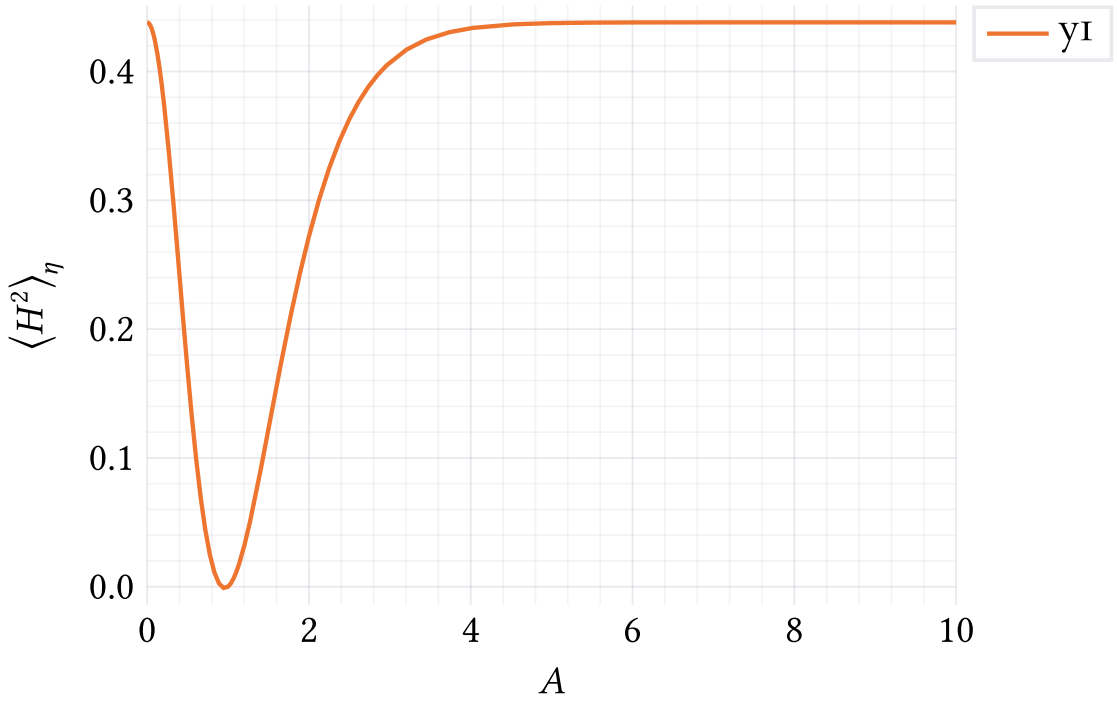

Figure 3.35

```
η0 = 1e-1
plot(g -> find_zeros(A -> ηenergyvariance(Hint0(0), η0) -
↪  ηenergyvariance(A*Hint0(g), η0), 0, 1)[1], xlim=(0, 1))
```

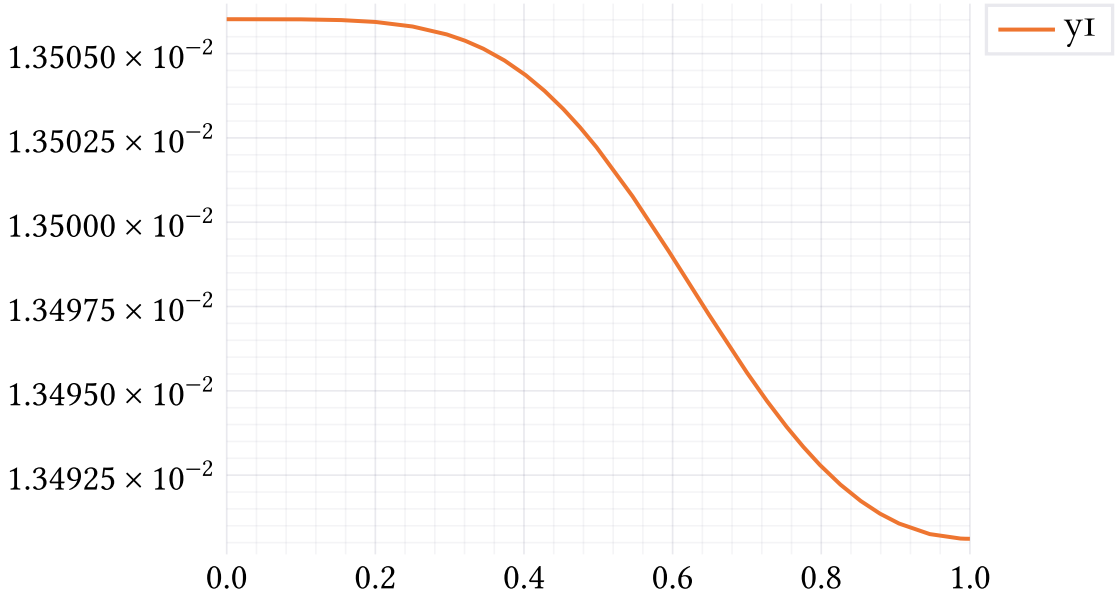

Figure 3.36

```
plot(g -> energyvariance(Hint(g), β0), xlim=(0, 1), xlabel=L"g",
↪  ylabel=L"\ev{H^2}_c")
```

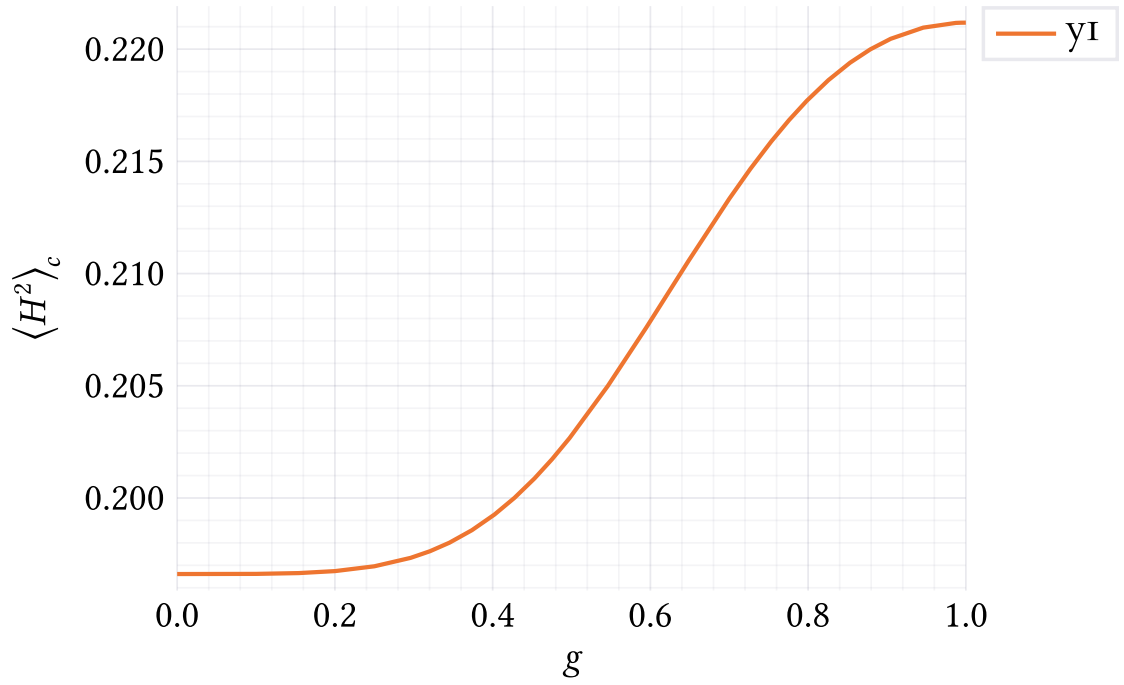

Figure 3.37

```
plot(g -> energyvariance(Ainvar(g, β0)*Hint(g), β0), xlim=(0, 1), xlabel=L"g",
↪  ylabel=L"\ev{H^2}_c")
```

```
┌ Warning: No strict ticks found
└ @ PlotUtils /home/jfjhh/.julia/packages/PlotUtils/es5pb/src/ticks.jl:283
```

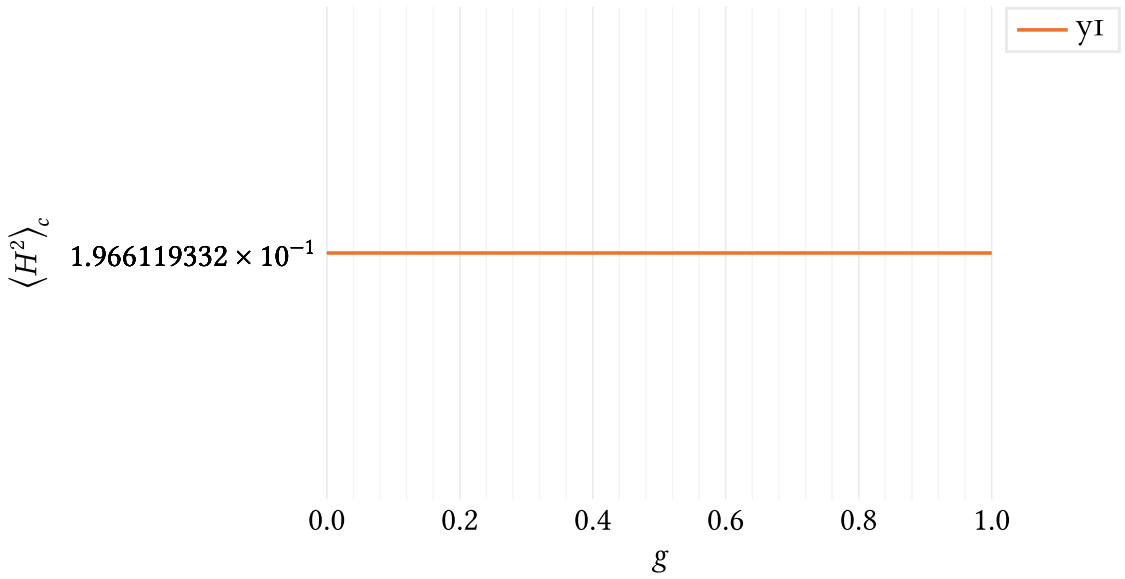

Figure 3.38

```
plot(g -> ηenergyvariance(Hint(g, η0), η0), xlim=(0, 1), xlabel=L"g",
↪  ylabel=L"\ev{H^2}_\eta")
```

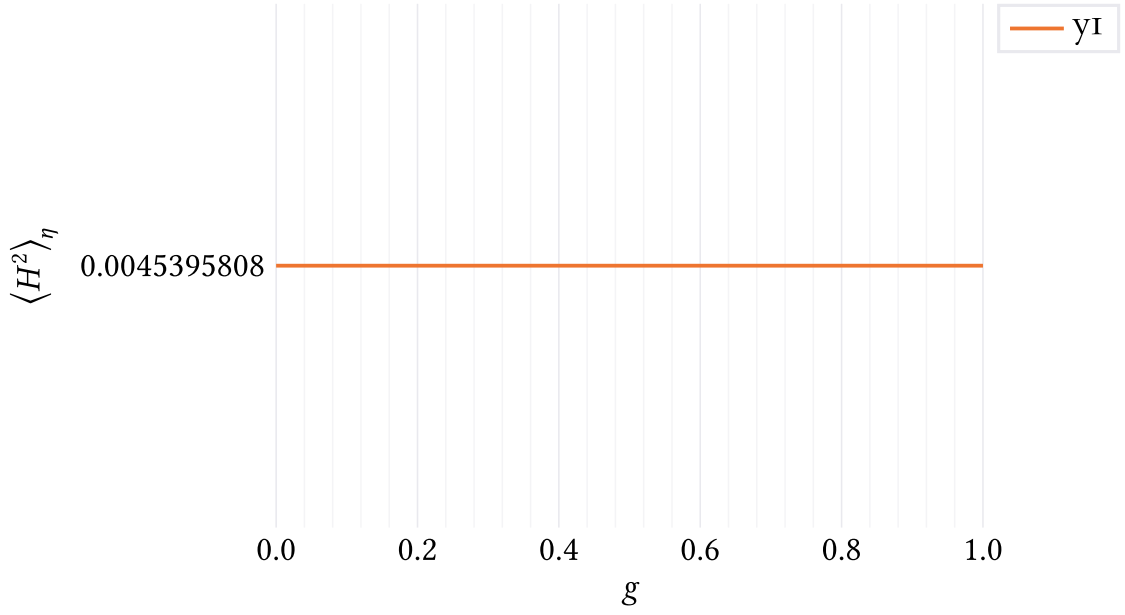

Figure 3.39

```
plot(η -> ηenergyvariance(Hint(0, η), η), xlim=(1e-2, 1e1), xscale=:log10,
↪  xlabel=L"\eta", ylabel=L"\ev{H^2}_\eta")
```

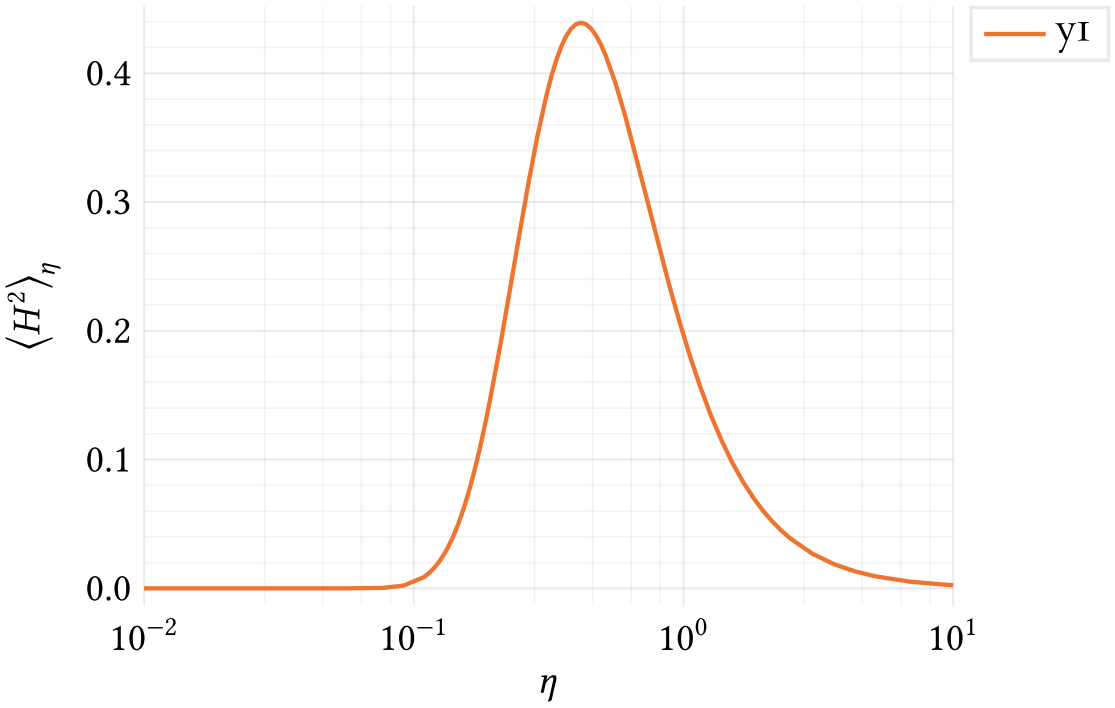

Figure 3.40

```
plot([g -> Ainvar(g, η) for η in [1e-1, 2e-1, 5e-1, 1e0, 1e1]], xlim=(0, 1),
    xlabel=L"g",
    ylabel=L"A",
)
```

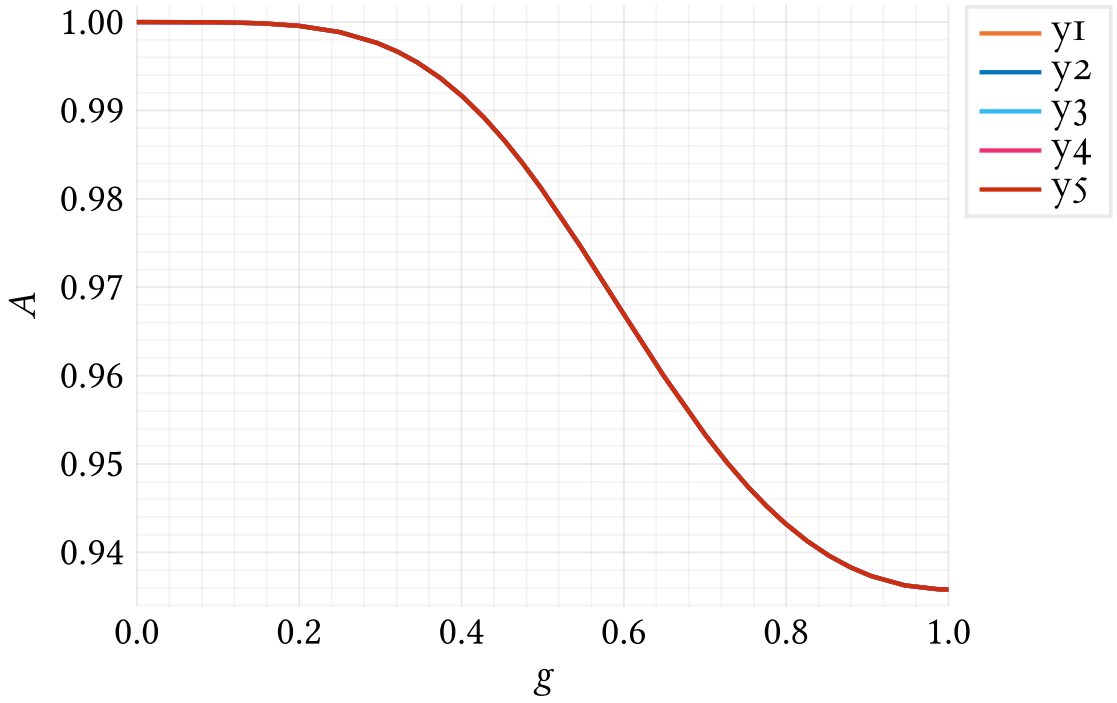

Figure 3.41

TODO: `Ainvar` for $g = 0$ should always be 1.

```
plot(η -> Ainvar(0, η), xlim=(1e-1, 1))
```

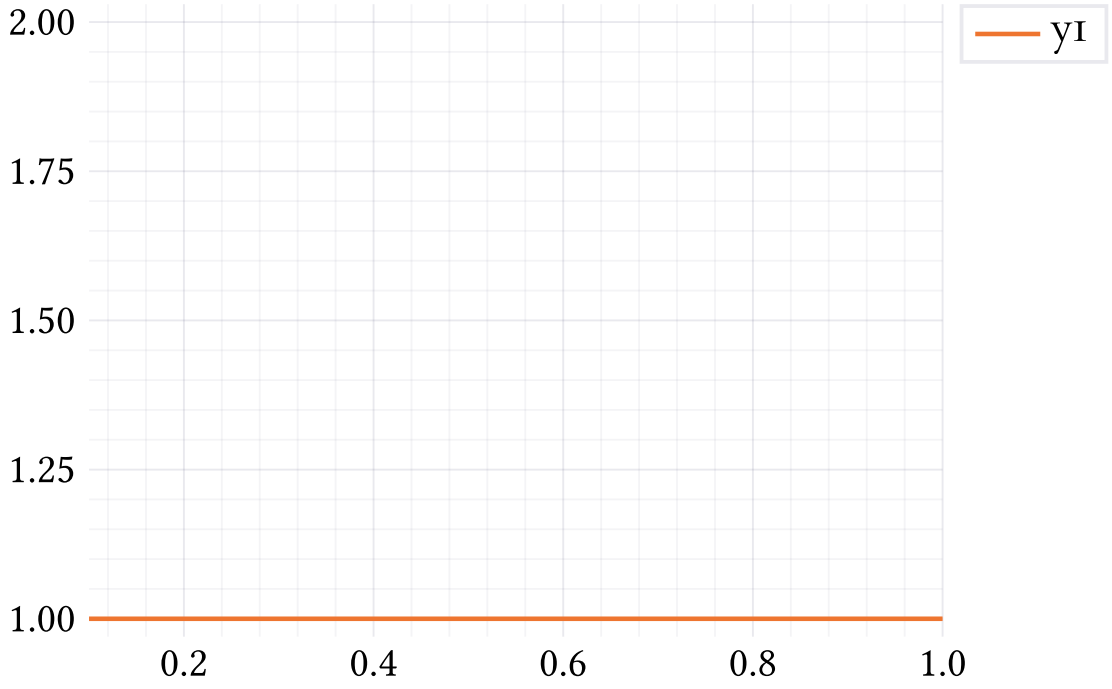

Figure 3.42

3.5.4 Plots for seminar

```
σ2zs(n) = -1 / (2n + 1)
σ2z(τ, n) = (1 - σ2zs(n))*exp(-τ) + σ2zs(n)
nH = 0
plot(τ -> (1 - σ2z(τ, nH))/2, label="Ground state",
    color=:black,
    xlim=(0, 5),
    ylim=(-0.01, 1),
    xlabel=L"Time ($\gamma t$)",
    ylabel="Populations",
    title="Decay from the excited state in vacuum",
)
plot!(τ -> (1 + σ2z(τ, nH))/2, xlim=(0, 5), label="Excited state",
 ↪  linestyle=:dash, color=:black)
scatter!(5 * [1, 1], [1 + σ2zs(nH), 1 - σ2zs(nH)] / 2, label="Equilibrium
 ↪  value",
```

```
    marker=(:ltriangle, 4, rubric),
    markerstrokecolor=rubric)
```

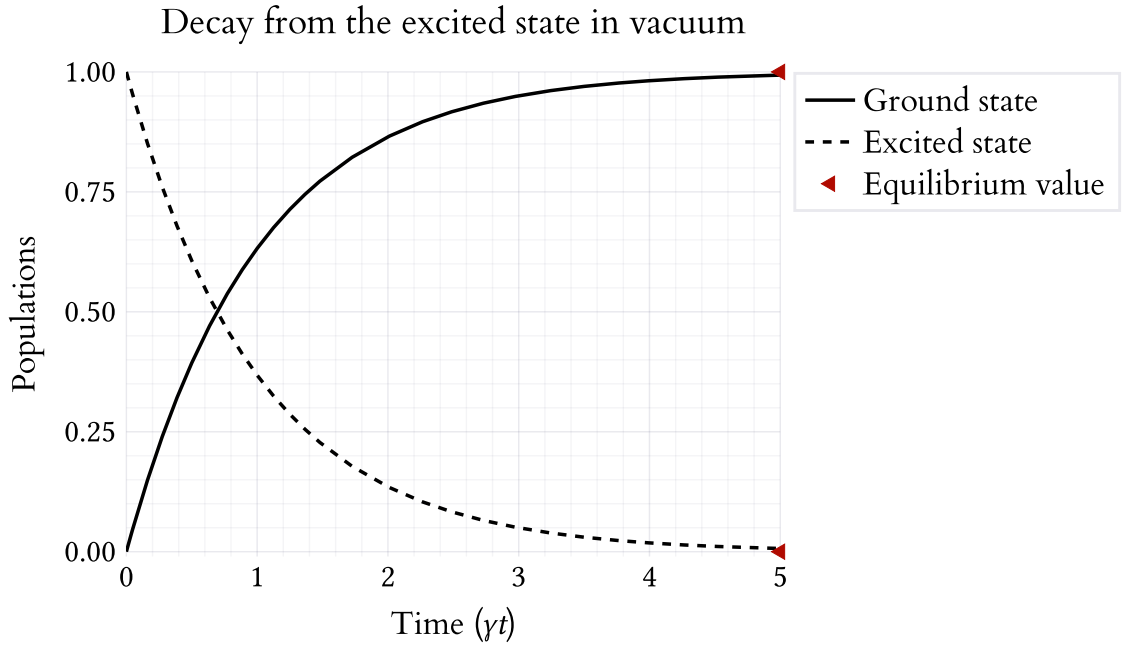

Figure 3.43

```
savefig("populations.pdf")
```

3.6 Two-spin degenerate jump operators

(Divided by 2.)

For $\omega = 0$:

$$2\sigma_1^x \tag{3.139}$$
$$2\sigma_2^x \tag{3.140}$$

For $\omega = -4$:

$$\sigma_1^y - \mathrm{i}\sigma_1^z\sigma_2^x \tag{3.141}$$
$$\mathrm{i}\sigma_1^y\sigma_2^x + \sigma_1^z \tag{3.142}$$
$$-\mathrm{i}\sigma_1^x\sigma_2^z + \sigma_2^y \tag{3.143}$$
$$\mathrm{i}\sigma_1^x\sigma_2^y + \sigma_2^z \tag{3.144}$$

For $\omega = 4$:

$$\sigma_1^y + \mathrm{i}\sigma_1^z\sigma_2^x \tag{3.145}$$
$$-\mathrm{i}\sigma_1^y\sigma_2^x + \sigma_1^z \tag{3.146}$$
$$\mathrm{i}\sigma_1^x\sigma_2^z + \sigma_2^y \tag{3.147}$$
$$-\mathrm{i}\sigma_1^x\sigma_2^y + \sigma_2^z \tag{3.148}$$

3.7 Two-spin nondegenerate jump operators

Simplify further: better operators and collecting noncommutative terms as usual.

See figs. 3.44 and 3.45.

Let $d = \sqrt{g^2 + 1}$.

For $\omega = -4d$:

$$-\sqrt{2}g\sigma_1^x\sigma_2^x + \sqrt{2}g\sigma_1^y\sigma_2^y + \sqrt{2}\mathrm{i}d\sigma_1^x\sigma_2^y + \sqrt{2}\mathrm{i}d\sigma_1^y\sigma_2^x + \sqrt{2}\sigma_1^z + \sqrt{2}\sigma_2^z \tag{3.208}$$

For $\omega = -2d - 2$:

$$-\mathrm{i}g^2\sigma_1^x - \mathrm{i}g^2\sigma_1^x\sigma_2^z + g^2\sigma_1^z\sigma_2^y + g^2\sigma_2^y - gd\sigma_1^y - gd\sigma_1^y\sigma_2^z + \mathrm{i}gd\sigma_1^z\sigma_2^x + \mathrm{i}gd\sigma_2^x - g\sigma_1^y + \mathrm{i}g\sigma_1^z\sigma_2^x - \mathrm{i}d\sigma_1^x\sigma_2^z + d\sigma_2^y - \mathrm{i}\sigma_1^x\sigma_2^z + \sigma_2^y \tag{3.209}$$
$$g^2\sigma_1^x + g^2\sigma_1^x\sigma_2^z + \mathrm{i}g^2\sigma_1^z\sigma_2^y + \mathrm{i}g^2\sigma_2^y - \mathrm{i}gd\sigma_1^y - \mathrm{i}gd\sigma_1^y\sigma_2^z - gd\sigma_1^z\sigma_2^x - gd\sigma_2^x + \mathrm{i}g\sigma_1^y\sigma_2^z + g\sigma_2^x - d\sigma_1^x - \mathrm{i}d\sigma_1^z\sigma_2^y + \sigma_1^x + \mathrm{i}\sigma_1^z\sigma_2^y \tag{3.210}$$
$$\mathrm{i}g^2\sigma_1^y + \mathrm{i}g^2\sigma_1^y\sigma_2^z + g^2\sigma_1^z\sigma_2^x + g^2\sigma_2^x - gd\sigma_1^x - gd\sigma_1^x\sigma_2^z - \mathrm{i}gd\sigma_1^z\sigma_2^y - \mathrm{i}gd\sigma_2^y + g\sigma_1^x + \mathrm{i}g\sigma_1^z\sigma_2^y - \mathrm{i}d\sigma_1^y\sigma_2^z - d\sigma_2^x + \mathrm{i}\sigma_1^y\sigma_2^z + \sigma_2^x \tag{3.211}$$

$$g^2\sigma_1^y + g^2\sigma_1^y\sigma_2^z - \mathrm{i}g^2\sigma_1^z\sigma_2^x - \mathrm{i}g^2\sigma_2^x + \mathrm{i}gd\sigma_1^x + \mathrm{i}gd\sigma_1^x\sigma_2^z - gd\sigma_1^z\sigma_2^y - gd\sigma_2^y + \mathrm{i}g\sigma_1^x\sigma_2^z - g\sigma_2^y + d\sigma_1^y - \mathrm{i}d\sigma_1^z\sigma_2^x + \sigma_1^y - \mathrm{i}\sigma_1^z\sigma_2^x \tag{3.212}$$

For $\omega = 2d + 2$:

$$\mathrm{i}g^2\sigma_1^x + \mathrm{i}g^2\sigma_1^x\sigma_2^z + g^2\sigma_1^z\sigma_2^y + g^2\sigma_2^y - gd\sigma_1^y - gd\sigma_1^y\sigma_2^z - \mathrm{i}gd\sigma_1^z\sigma_2^x - \mathrm{i}gd\sigma_2^x - g\sigma_1^y - \mathrm{i}g\sigma_1^z\sigma_2^x + \mathrm{i}d\sigma_1^x\sigma_2^z + d\sigma_2^y + \mathrm{i}\sigma_1^x\sigma_2^z + \sigma_2^y \tag{3.213}$$

$$-\mathrm{i}g^2\sigma_1^y - \mathrm{i}g^2\sigma_1^y\sigma_2^z + g^2\sigma_1^z\sigma_2^x + g^2\sigma_2^x - gd\sigma_1^x - gd\sigma_1^x\sigma_2^z + \mathrm{i}gd\sigma_1^z\sigma_2^y + \mathrm{i}gd\sigma_2^y + g\sigma_1^x - \mathrm{i}g\sigma_1^z\sigma_2^y + \mathrm{i}d\sigma_1^y\sigma_2^z - d\sigma_2^x - \mathrm{i}\sigma_1^y\sigma_2^z + \sigma_2^x \tag{3.214}$$

$$g^2\sigma_1^y + g^2\sigma_1^y\sigma_2^z + \mathrm{i}g^2\sigma_1^z\sigma_2^x + \mathrm{i}g^2\sigma_2^x - \mathrm{i}gd\sigma_1^x - \mathrm{i}gd\sigma_1^x\sigma_2^z - gd\sigma_1^z\sigma_2^y - gd\sigma_2^y - \mathrm{i}g\sigma_1^x\sigma_2^z - g\sigma_2^y + d\sigma_1^y + \mathrm{i}d\sigma_1^z\sigma_2^x + \sigma_1^y + \mathrm{i}\sigma_1^z\sigma_2^x \tag{3.215}$$

$$g^2\sigma_1^x + g^2\sigma_1^x\sigma_2^z - \mathrm{i}g^2\sigma_1^z\sigma_2^y - \mathrm{i}g^2\sigma_2^y + \mathrm{i}gd\sigma_1^y + \mathrm{i}gd\sigma_1^y\sigma_2^z - gd\sigma_1^z\sigma_2^x - gd\sigma_2^x - \mathrm{i}g\sigma_1^y\sigma_2^z + g\sigma_2^x - d\sigma_1^x + \mathrm{i}d\sigma_1^z\sigma_2^y + \sigma_1^x - \mathrm{i}\sigma_1^z\sigma_2^y \tag{3.216}$$

For $\omega = 0$:

$$4\sqrt{2}g^4 + 4\sqrt{2}g^4\sigma_1^z + 4\sqrt{2}g^4\sigma_1^z\sigma_2^z + 4\sqrt{2}g^4\sigma_2^z + 4\sqrt{2}g^3\sigma_1^x\sigma_2^x - 4\sqrt{2}g^3\sigma_1^y\sigma_2^y + 4\sqrt{2}g^2 + 2\sqrt{2}g^2\sigma_1^z + 4\sqrt{2}g^2\sigma_1^z\sigma_2^z + 2\sqrt{2}g^2\sigma_2^z + 2\sqrt{2}g\sigma_1^x\sigma_2^x - 2\sqrt{2}g\sigma_1^y\sigma_2^y \tag{3.217}$$

For $\omega = 2 - 2d$:

$$\mathrm{i}g^2\sigma_1^x + \mathrm{i}g^2\sigma_1^x\sigma_2^z + g^2\sigma_1^z\sigma_2^y + g^2\sigma_2^y + gd\sigma_1^y + gd\sigma_1^y\sigma_2^z + \mathrm{i}gd\sigma_1^z\sigma_2^x + \mathrm{i}gd\sigma_2^x - g\sigma_1^y - \mathrm{i}g\sigma_1^z\sigma_2^x - \mathrm{i}d\sigma_1^x\sigma_2^z - d\sigma_2^y + \mathrm{i}\sigma_1^x\sigma_2^z + \sigma_2^y \tag{3.218}$$

$$g^2\sigma_1^y + g^2\sigma_1^y\sigma_2^z + \mathrm{i}g^2\sigma_1^z\sigma_2^x + \mathrm{i}g^2\sigma_2^x + \mathrm{i}gd\sigma_1^x + \mathrm{i}gd\sigma_1^x\sigma_2^z + gd\sigma_1^z\sigma_2^y + gd\sigma_2^y - \mathrm{i}g\sigma_1^x\sigma_2^z - g\sigma_2^y - d\sigma_1^y - \mathrm{i}d\sigma_1^z\sigma_2^x + \sigma_1^y + \mathrm{i}\sigma_1^z\sigma_2^x \tag{3.219}$$

$$g^2\sigma_1^x + g^2\sigma_1^x\sigma_2^z - \mathrm{i}g^2\sigma_1^z\sigma_2^y - \mathrm{i}g^2\sigma_2^y - \mathrm{i}gd\sigma_1^y - \mathrm{i}gd\sigma_1^y\sigma_2^z + gd\sigma_1^z\sigma_2^x + gd\sigma_2^x - \mathrm{i}g\sigma_1^y\sigma_2^z + g\sigma_2^x + d\sigma_1^x - \mathrm{i}d\sigma_1^z\sigma_2^y + \sigma_1^x - \mathrm{i}\sigma_1^z\sigma_2^y \tag{3.220}$$

$$-\mathrm{i}g^2\sigma_1^y - \mathrm{i}g^2\sigma_1^y\sigma_2^z + g^2\sigma_1^z\sigma_2^x + g^2\sigma_2^x + gd\sigma_1^x + gd\sigma_1^x\sigma_2^z - \mathrm{i}gd\sigma_1^z\sigma_2^y - \mathrm{i}gd\sigma_2^y + g\sigma_1^x - \mathrm{i}g\sigma_1^z\sigma_2^y - \mathrm{i}d\sigma_1^y\sigma_2^z + d\sigma_2^x - \mathrm{i}\sigma_1^y\sigma_2^z + \sigma_2^x \tag{3.221}$$

For $\omega = 2d - 2$:

$$g^2\sigma_1^y + g^2\sigma_1^y\sigma_2^z - \mathrm{i}g^2\sigma_1^z\sigma_2^x - \mathrm{i}g^2\sigma_2^x - \mathrm{i}gd\sigma_1^x - \mathrm{i}gd\sigma_1^x\sigma_2^z + gd\sigma_1^z\sigma_2^y + gd\sigma_2^y + \mathrm{i}g\sigma_1^x\sigma_2^z - g\sigma_2^y - d\sigma_1^y + \mathrm{i}d\sigma_1^z\sigma_2^x + \sigma_1^y - \mathrm{i}\sigma_1^z\sigma_2^x \tag{3.222}$$

$$g^2\sigma_1^x + g^2\sigma_1^x\sigma_2^z + \mathrm{i}g^2\sigma_1^z\sigma_2^y + \mathrm{i}g^2\sigma_2^y + \mathrm{i}gd\sigma_1^y + \mathrm{i}gd\sigma_1^y\sigma_2^z + gd\sigma_1^z\sigma_2^x + gd\sigma_2^x + \mathrm{i}g\sigma_1^y\sigma_2^z + g\sigma_2^x + d\sigma_1^x + \mathrm{i}d\sigma_1^z\sigma_2^y + \sigma_1^x + \mathrm{i}\sigma_1^z\sigma_2^y \tag{3.223}$$

$$-\mathrm{i}g^2\sigma_1^x - \mathrm{i}g^2\sigma_1^x\sigma_2^z + g^2\sigma_1^z\sigma_2^y + g^2\sigma_2^y + gd\sigma_1^y + gd\sigma_1^y\sigma_2^z - \mathrm{i}gd\sigma_1^z\sigma_2^x - \mathrm{i}gd\sigma_2^x - g\sigma_1^y + \mathrm{i}g\sigma_1^z\sigma_2^x + \mathrm{i}d\sigma_1^x\sigma_2^z - d\sigma_2^y - \mathrm{i}\sigma_1^x\sigma_2^z + \sigma_2^y \tag{3.224}$$

$$\mathrm{i}g^2\sigma_1^y + \mathrm{i}g^2\sigma_1^y\sigma_2^z + g^2\sigma_1^z\sigma_2^x + g^2\sigma_2^x + gd\sigma_1^x + gd\sigma_1^x\sigma_2^z + \mathrm{i}gd\sigma_1^z\sigma_2^y + \mathrm{i}gd\sigma_2^y + g\sigma_1^x + \mathrm{i}g\sigma_1^z\sigma_2^y + \mathrm{i}d\sigma_1^y\sigma_2^z + d\sigma_2^x + \mathrm{i}\sigma_1^y\sigma_2^z + \sigma_2^x \tag{3.225}$$

For $\omega = 4d$:

$$-\sqrt{2}g\sigma_1^x\sigma_2^x + \sqrt{2}g\sigma_1^y\sigma_2^y - \sqrt{2}\mathrm{i}d\sigma_1^x\sigma_2^y - \sqrt{2}\mathrm{i}d\sigma_1^y\sigma_2^x + \sqrt{2}\sigma_1^z + \sqrt{2}\sigma_2^z \tag{3.226}$$

For $\omega = 4$:

$$-\mathrm{i}\sigma_1^x\sigma_2^y + \mathrm{i}\sigma_1^y\sigma_2^x - \sigma_1^z + \sigma_2^z \tag{3.227}$$

$$\mathrm{i}\sigma_1^x\sigma_2^y - \mathrm{i}\sigma_1^y\sigma_2^x + \sigma_1^z - \sigma_2^z \tag{3.228}$$

For $\omega = -4$:

$$\mathrm{i}\sigma_1^x\sigma_2^y - \mathrm{i}\sigma_1^y\sigma_2^x - \sigma_1^z + \sigma_2^z \tag{3.229}$$

$$-\mathrm{i}\sigma_1^x\sigma_2^y + \mathrm{i}\sigma_1^y\sigma_2^x + \sigma_1^z - \sigma_2^z \tag{3.230}$$

3.8 Previous calculations

Note that for c_i and d_i from an orthonormal basis for $\mathscr{L}_i$,

$$\left\langle \bigotimes_i a_i c_i \middle| \bigotimes_i b_i d_i \right\rangle = \operatorname{tr}\left(\bigotimes_i a_i^* b_i \, c_i^\dagger d_i \right) \tag{3.231}$$

$$= \prod_i \operatorname{tr}\left(a_i^* b_i \, c_i^\dagger d_i \right) \tag{3.232}$$

$$= \prod_i a_i^* b_i \, \langle c_i | d_i \rangle \tag{3.233}$$

$$= \begin{cases} \prod_i a_i^* b_i & c_i = d_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.234}$$

Thus

$$\langle \sigma_i^x | \eta_k \rangle = -\left\langle c_i^\dagger + c_i \middle| \sum_j a_{jk} c_j + b_{jk} c_j^\dagger \right\rangle \tag{3.235}$$

$$= a_{ik} + b_{ik} \tag{3.236}$$

$$\left\langle \sigma_i^x \middle| \eta_k^\dagger \right\rangle = \langle \eta_k | \sigma_i^x \rangle \tag{3.237}$$

$$= a_{ik}^* + b_{ik}^* \tag{3.238}$$

$$\left\langle \sigma_i^x \middle| \eta_k^\dagger \eta_k \right\rangle = 0 \quad \text{since trace } i \text{ vanishes} \tag{3.239}$$

$$\left\langle \sigma_i^x \middle| \eta_k \eta_k^\dagger \right\rangle = \left\langle \eta_k^\dagger \eta_k \middle| \sigma_i^x \right\rangle \tag{3.240}$$

$$= 0. \tag{3.241}$$

Thus

$$\sigma_i^x = \sum_k A_{ik} + A_{ik}^\dagger = 2 \operatorname{He} \sum_k A_{ik}, \tag{3.242}$$

where

$$A_{ik} = (a_{ik} + b_{ik}) \eta_k. \tag{3.243}$$

According to Pfeuty, if $\lambda \neq 0$ then

$$a_{ik} + b_{ik} = \sqrt{\frac{2}{N}} \begin{cases} \sin(ki) & k > 0 \\ \cos(ki) & k \leqslant 0, \end{cases} \tag{3.244}$$

and if $\lambda = 0$ then $a_{ik} + b_{ik} = 1/\sqrt{N}$.

Now since η_k is an eigenoperator of the Hamiltonian with eigenvalue $\omega_k = -E_k$, and $\eta_k^\dagger$ similarly has $\omega_k = E_k$, we may use eqs. (2.96), (2.177) and (2.184) to find that the

dissipator is

$$\begin{aligned}\mathcal{D}\rho &= \sum_{ik}\frac{2L}{c}C(-E_k)^2 n_B(E_k)\Big(A_{ik}\rho A_{ik}^\dagger - \frac{1}{2}\Big\{A_{ik}^\dagger A_{ik},\rho\Big\}\Big) \\ &+ \sum_{ik}\frac{2L}{c}C(E_k)^2(n_B(E_k)+1)\Big(A_{ik}^\dagger\rho A_{ik} - \frac{1}{2}\Big\{A_{ik}A_{ik}^\dagger,\rho\Big\}\Big) \end{aligned} \tag{3.245}$$

$$\begin{aligned}&= \sum_{ik}\frac{2L}{c}C(-E_k)^2 n_B(E_k)|a_{ik}+b_{ik}|^2\Big(\eta_k\rho\eta_k^\dagger - \frac{1}{2}\Big\{\eta_k^\dagger\eta_k,\rho\Big\}\Big) \\ &+ \sum_{ik}\frac{2L}{c}C(E_k)^2(n_B(E_k)+1)|a_{ik}+b_{ik}|^2\Big(\eta_k^\dagger\rho\eta_k - \frac{1}{2}\Big\{\eta_k\eta_k^\dagger,\rho\Big\}\Big) \end{aligned} \tag{3.246}$$

$$\begin{aligned}&= \sum_{k}\frac{2L}{Nc}C(-E_k)^2 n_B(E_k)\Big(\eta_k\rho\eta_k^\dagger - \frac{1}{2}\Big\{\eta_k^\dagger\eta_k,\rho\Big\}\Big) \\ &+ \sum_{k}\frac{2L}{Nc}C(E_k)^2(n_B(E_k)+1)\Big(\eta_k^\dagger\rho\eta_k - \frac{1}{2}\Big\{\eta_k\eta_k^\dagger,\rho\Big\}\Big), \end{aligned} \tag{3.247}$$

since

$$\sum_i |a_{ik}+b_{ik}|^2 = \frac{2}{N}\sum_i \sin^2(ki) = \frac{1}{N}. \tag{3.248}$$

If $\rho_0 = |\mathbf{0}\rangle\langle\mathbf{0}| = \otimes_k \eta_k\eta_k^\dagger$, then at $t=0$

$$\mathcal{D}\rho = \sum_{ik}\frac{2L}{c}C(E_k)^2(n_B(E_k)+1)A_{ik}^\dagger A_{ik} \neq 0, \tag{3.249}$$

so the ground state of the closed system is not stationary?

3.9 Characterization of two-level dissipators

If we have a two-level system with density operator ρ, a general form of the dissipator is

$$\begin{aligned}\mathcal{D}\rho &= \gamma_-|h_-|^2\Big(\sigma^-\rho\sigma^+ - \frac{1}{2}\{\sigma^+\sigma^-,\rho\}\Big) \\ &+ \gamma_+|h_+|^2\Big(\sigma^+\rho\sigma^- - \frac{1}{2}\{\sigma_-\sigma^+,\rho\}\Big) \\ &+ \gamma_z|h_z|^2(\sigma^z\rho\sigma^z - \rho), \end{aligned} \tag{3.250}$$

where each of the Pauli operators σ^a is assumed to be an eigenoperator of some Hamiltonian with eigenvalue ω_a, and $\gamma_a = \gamma(\omega_a)$. We also have coefficients from expressing part of an interaction Hamiltonian as $\sum_a h_a\sigma^a$.

We evaluate eq. (3.250) by substituting

$$\rho = \rho_{00}\sigma^-\sigma^+ + \rho_{01}\sigma^- + \rho_{10}\sigma^+ + \rho_{11}\sigma^+\sigma^-. \tag{3.251}$$

This gives

$$\begin{aligned}
\mathcal{D}\rho &= \gamma_+|h_+|^2(\sigma^+\sigma^- - \sigma_-\sigma^+)\rho_{00} + \gamma_-|h_-|^2(\sigma^-\sigma^+ - \sigma^+\sigma^-)\rho_{11} \\
&\quad + \gamma_-|h_-|^2\Big(-\frac{1}{2}\{\sigma^+\sigma^-,\sigma^-\}\Big)\rho_{01} + \gamma_+|h_+|^2\Big(-\frac{1}{2}\{\sigma_-\sigma^+,\sigma^-\}\Big)\rho_{01} \\
&\quad + \gamma_-|h_-|^2\Big(-\frac{1}{2}\{\sigma^+\sigma^-,\sigma^+\}\Big)\rho_{10} + \gamma_+|h_+|^2\Big(-\frac{1}{2}\{\sigma_-\sigma^+,\sigma^+\}\Big)\rho_{10} \\
&\quad + \gamma_z|h_z|^2(\sigma^z\sigma^-\sigma^z - \sigma^-)\rho_{01} + \gamma_z|h_z|^2(\sigma^z\sigma^+\sigma^z - \sigma^+)\rho_{10}
\end{aligned} \tag{3.252}$$

$$\begin{aligned}
&= \gamma_+|h_+|^2(\sigma^+\sigma^- - \sigma_-\sigma^+)\rho_{00} + \gamma_-|h_-|^2(\sigma^-\sigma^+ - \sigma^+\sigma^-)\rho_{11} \\
&\quad - \gamma_-|h_-|^2\frac{1}{2}\sigma^-\rho_{01} - \gamma_+|h_+|^2\frac{1}{2}\sigma^-\rho_{01} - \gamma_z|h_z|^2 2\sigma^-\rho_{01} \\
&\quad - \gamma_-|h_-|^2\frac{1}{2}\sigma^+\rho_{10} - \gamma_+|h_+|^2\frac{1}{2}\sigma^+\rho_{10} - \gamma_z|h_z|^2 2\sigma^+\rho_{10}.
\end{aligned} \tag{3.253}$$

Ignoring the Lamb shift, the entries of the density matrix in the interaction picture are then given by

$$\dot\rho_{00} = -\gamma_+|h_+|^2\rho_{00} + \gamma_-|h_-|^2\rho_{11} \tag{3.254a}$$

$$\dot\rho_{01} = -\frac{1}{2}\big(\gamma_-|h_-|^2 + \gamma_+|h_+|^2 + 4\gamma_z|h_z|^2\big)\rho_{01} \tag{3.254b}$$

$$\dot\rho_{10} = -\frac{1}{2}\big(\gamma_-|h_-|^2 + \gamma_+|h_+|^2 + 4\gamma_z|h_z|^2\big)\rho_{10} \tag{3.254c}$$

$$\dot\rho_{11} = \gamma_+|h_+|^2\rho_{00} - \gamma_-|h_-|^2\rho_{11}. \tag{3.254d}$$

In the case where $h_z = 0$ and $h_- = h_+ =: h$, eq. (3.254) reduces to

$$\dot\rho_{00}|h|^{-2} = -\gamma_+\rho_{00} + \gamma_-\rho_{11} \tag{3.255a}$$

$$\dot\rho_{01}|h|^{-2} = -\frac{1}{2}(\gamma_- + \gamma_+)\rho_{01} \tag{3.255b}$$

$$\dot\rho_{10}|h|^{-2} = -\frac{1}{2}(\gamma_- + \gamma_+)\rho_{10} \tag{3.255c}$$

$$\dot\rho_{11}|h|^{-2} = \gamma_+\rho_{00} - \gamma_-\rho_{11}. \tag{3.255d}$$

Thus as $t \to \infty$, ρ approaches a pure state with both levels fractionally occupied? (Wait, this neglects the unitary evolution. Will do in full.) If so, compare to canonical occupations?

3.10 *Pfeuty scratch work*

I have

$$\lambda = -\frac{1}{\lambda_{\mathrm{Pf}}} = -\frac{2\Gamma}{J}, \tag{3.256}$$

and my Hamiltonian eq. (3.2) should have

$$H \mapsto \frac{4}{J} H \tag{3.257}$$

to be correctly nondimensionalized and match Pfeuty's results. Then

$$E_k = \frac{\Lambda_k}{\lambda_{\mathrm{Pf}}} = -\Lambda_k \lambda. \tag{3.258}$$

Pfeuty defines:

$$\lambda = \frac{J}{2\Gamma} \tag{3.259}$$

$$a_i = S_{xi} - \mathrm{i} S_{yi} \tag{3.260}$$

$$a_i^\dagger = S_{xi} + \mathrm{i} S_{yi} \tag{3.261}$$

$$c_i = \exp\left(\pi \mathrm{i} \sum_{j=1}^{i-1} a_j^\dagger a_j\right) a_i \tag{3.262}$$

$$c_i^\dagger = a_i^\dagger \exp\left(-\pi \mathrm{i} \sum_{j=1}^{i-1} a_j^\dagger a_j\right) \tag{3.263}$$

$$\eta_k = \sum_i \left(\frac{\varphi_{ki} + \psi_{ki}}{2} c_i + \frac{\varphi_{ki} - \psi_{ki}}{2} c_i^\dagger\right) \tag{3.264}$$

$$\varphi_{ki} = \sqrt{\frac{2}{N}} \begin{cases} \sin(ki) & k > 0 \\ \cos(ki) & k \leqslant 0 \end{cases} \tag{3.265}$$

$$\psi_{ki} = -\Lambda_k^{-1}((1 + \lambda \cos k)\varphi_{ki} + (\lambda \sin k)\varphi_{-ki}) \tag{3.266}$$

$$\Lambda_k^2 = 1 + \lambda^2 + 2\lambda \cos k \tag{3.267}$$

$$k = \frac{2\pi m}{N} \quad \text{for} \quad m = -\frac{N}{2}, \ldots, \frac{N}{2} - 1, \quad N \text{ even}. \tag{3.268}$$

We would like to express the S_{xi} in terms of eigenoperators of the system Hamiltonian

$$H = \Gamma \sum_k \Lambda_k \eta_k^\dagger \eta_k - \frac{\Gamma}{2} \sum_k \Lambda_k. \tag{3.269}$$

$$\begin{aligned}\left\{\eta_k, \eta_k^\dagger\right\} &= \sum_{ij} \frac{\varphi_{ki} + \psi_{ki}}{2} \frac{\varphi_{kj} - \psi_{kj}}{2} \left\{c_i, c_j\right\} + \sum_{ij} \frac{\varphi_{ki} + \psi_{ki}}{2} \frac{\varphi_{kj} + \psi_{kj}}{2} \left\{c_i, c_j^\dagger\right\} \\ &+ \sum_{ij} \frac{\varphi_{ki} - \psi_{ki}}{2} \frac{\varphi_{kj} - \psi_{kj}}{2} \left\{c_i^\dagger, c_j\right\} + \sum_{ij} \frac{\varphi_{ki} - \psi_{ki}}{2} \frac{\varphi_{kj} + \psi_{kj}}{2} \left\{c_i^\dagger, c_j^\dagger\right\} \qquad (3.270) \\ &= \frac{I}{4} \sum_i \left(\varphi_{ki} + \psi_{ki}\right)^2 + \left(\varphi_{ki} - \psi_{ki}\right)^2 \qquad (3.271) \\ &= \frac{I}{2} \sum_i \varphi_{ki}^2 + \psi_{ki}^2 \qquad (3.272) \\ &= I? \qquad (3.273)\end{aligned}$$

According to Mathematica, no? But [23, pp. 452–454] is probably right.

$$2\,(d+1)$$

$$-2n_1\left(\sigma_2^+\left(dg-d+g^2+1\right)+\sigma_2^-\left(dg+d-g^2-1\right)\right)+\sigma_1^+\left(2n_2g\,(d+g-1)-d+g+1\right)+\sigma_1^-\left(2n_2g\,(-d+g+1)-d-g+1\right)+\sigma_2^+\left(-d+g+1\right)+\sigma_2^-\left(d+g-1\right)$$

$$\mathrm{i}\left(2n_1\left(\sigma_2^+\left(dg+d+g^2+1\right)+\sigma_2^-\left(-dg+d+g^2+1\right)\right)-\sigma_1^+\left(2n_2g\,(d+g+1)+d-g+1\right)+\sigma_1^-\left(-2n_2g\,(d-g+1)+d+g+1\right)-\sigma_2^+\left(d-g+1\right)-\sigma_2^-\left(d+g+1\right)\right)$$

$$2n_1g\left(\sigma_2^+\left(d+g-1\right)+\sigma_2^-\left(-d+g+1\right)\right)-\sigma_1^+\left(2n_2\left(dg-d+g^2+1\right)+d-g-1\right)-\sigma_1^-\left(2n_2\left(dg+d-g^2-1\right)-d-g+1\right)+\sigma_2^+\left(-d+g+1\right)-\sigma_2^-\left(d+g-1\right)$$

$$\mathrm{i}\left(-2n_1g\left(\sigma_2^+\left(d+g+1\right)+\sigma_2^-\left(d-g+1\right)\right)+\sigma_1^+\left(2n_2\left(dg+d+g^2+1\right)-d+g-1\right)-\sigma_1^-\left(-2n_2\left(-dg+d+g^2+1\right)+d+g+1\right)-\sigma_2^+\left(d-g+1\right)+\sigma_2^-\left(d+g+1\right)\right)$$

$$-2\,(d+1)$$

$$-2n_1\left(\sigma_2^+\left(dg+d-g^2-1\right)+\sigma_2^-\left(dg-d+g^2+1\right)\right)+\sigma_1^+\left(2n_2g\left(-d+g+1\right)-d-g+1\right)+\sigma_1^-\left(2n_2g\,(d+g-1)-d+g+1\right)+\sigma_2^+\left(d+g-1\right)+\sigma_2^-\left(-d+g+1\right)$$

$$\mathrm{i}\left(2n_1g\left(\sigma_2^+\left(d-g+1\right)+\sigma_2^-\left(d+g+1\right)\right)+\sigma_1^+\left(-2n_2\left(-dg+d+g^2+1\right)+d+g+1\right)-\sigma_1^-\left(2n_2\left(dg+d+g^2+1\right)-d+g-1\right)-\sigma_2^+\left(d+g+1\right)+\sigma_2^-\left(d-g+1\right)\right)$$

$$\mathrm{i}\left(-2n_1\left(\sigma_2^+\left(-dg+d+g^2+1\right)+\sigma_2^-\left(dg+d+g^2+1\right)\right)-\sigma_1^+\left(-2n_2g\,(d-g+1)+d+g+1\right)+\sigma_1^-\left(2n_2g\,(d+g+1)+d-g+1\right)+\sigma_2^+\left(d+g+1\right)+\sigma_2^-\left(d-g+1\right)\right)$$

$$2n_1g\left(\sigma_2^+\left(-d+g+1\right)+\sigma_2^-\left(d+g-1\right)\right)-\sigma_1^+\left(2n_2\left(dg+d-g^2-1\right)-d-g+1\right)-\sigma_1^-\left(2n_2\left(dg-d+g^2+1\right)+d-g-1\right)-\sigma_2^+\left(d+g-1\right)+\sigma_2^-\left(-d+g+1\right)$$

$$4d$$

$$2\sqrt{2}\left(n_1+n_2-\sigma_1^+\sigma_2^+\left(d+g\right)+\sigma_1^-\sigma_2^-\left(d-g\right)-1\right)$$

$$2\,(d-1)$$

$$2n_1g\left(\sigma_2^+\left(d+g+1\right)-\sigma_2^-\left(d-g+1\right)\right)+\sigma_1^+\left(2n_2\left(dg+d+g^2+1\right)-d+g-1\right)+\sigma_1^-\left(-2n_2\left(-dg+d+g^2+1\right)+d+g+1\right)+\sigma_2^+\left(d-g+1\right)+\sigma_2^-\left(d+g+1\right)$$

$$\mathrm{i}\left(-2n_1\left(\sigma_2^+\left(dg-d+g^2+1\right)-\sigma_2^-\left(dg+d-g^2-1\right)\right)-\sigma_1^+\left(2n_2g\,(d+g-1)-d+g+1\right)+\sigma_1^-\left(2n_2g\left(-d+g+1\right)-d-g+1\right)+\sigma_2^+\left(-d+g+1\right)-\sigma_2^-\left(d+g-1\right)\right)$$

$$\mathrm{i}\left(-2n_1g\left(\sigma_2^+\left(d+g-1\right)-\sigma_2^-\left(-d+g+1\right)\right)-\sigma_1^+\left(2n_2\left(dg-d+g^2+1\right)+d-g-1\right)+\sigma_1^-\left(2n_2\left(dg+d-g^2-1\right)-d-g+1\right)-\sigma_2^+\left(-d+g+1\right)-\sigma_2^-\left(d+g-1\right)\right)$$

$$2n_1\left(\sigma_2^+\left(dg+d+g^2+1\right)-\sigma_2^-\left(-dg+d+g^2+1\right)\right)+\sigma_1^+\left(2n_2g\,(d+g+1)+d-g+1\right)+\sigma_1^-\left(-2n_2g\,(d-g+1)+d+g+1\right)-\sigma_2^+\left(d-g+1\right)+\sigma_2^-\left(d+g+1\right)$$

$$0$$

$$4\sqrt{2}g\left(n_1g\left(4n_2\left(g^2+1\right)-1\right)-n_2g+\sigma_1^+\sigma_2^+\left(2g^2+1\right)+\sigma_1^-\sigma_2^-\left(2g^2+1\right)+g\right)$$

$$-2\,(d-1)$$

$$-2n_1g\left(\sigma_2^+\left(d-g+1\right)-\sigma_2^-\left(d+g+1\right)\right)+\sigma_1^+\left(-2n_2\left(-dg+d+g^2+1\right)+d+g+1\right)+\sigma_1^-\left(2n_2\left(dg+d+g^2+1\right)-d+g-1\right)+\sigma_2^+\left(d+g+1\right)+\sigma_2^-\left(d-g+1\right)$$

$$-2n_1\left(\sigma_2^+\left(-dg+d+g^2+1\right)-\sigma_2^-\left(dg+d+g^2+1\right)\right)+\sigma_1^+\left(-2n_2g\,(d-g+1)+d+g+1\right)+\sigma_1^-\left(2n_2g\,(d+g+1)+d-g+1\right)+\sigma_2^+\left(d+g+1\right)-\sigma_2^-\left(d-g+1\right)$$

$$\mathrm{i}\left(-2n_1\left(\sigma_2^+\left(dg+d-g^2-1\right)-\sigma_2^-\left(dg-d+g^2+1\right)\right)-\sigma_1^+\left(2n_2g\left(-d+g+1\right)-d-g+1\right)+\sigma_1^-\left(2n_2g\,(d+g-1)-d+g+1\right)+\sigma_2^+\left(d+g-1\right)-\sigma_2^-\left(-d+g+1\right)\right)$$

$$\mathrm{i}\left(-2n_1g\left(\sigma_2^+\left(-d+g+1\right)-\sigma_2^-\left(d+g-1\right)\right)-\sigma_1^+\left(2n_2\left(dg-d+g^2-1\right)-d-g+1\right)+\sigma_1^-\left(2n_2\left(dg-d+g^2+1\right)+d-g-1\right)+\sigma_2^+\left(d+g-1\right)+\sigma_2^-\left(-d+g+1\right)\right)$$

$$-4d$$

$$2\sqrt{2}\left(n_1+n_2+\sigma_1^+\sigma_2^+\left(d-g\right)-\sigma_1^-\sigma_2^-\left(d+g\right)-1\right)$$

$$4$$

$$-2n_1+2n_2+2\sigma_1^+\sigma_2^--2\sigma_1^-\sigma_2^+$$

$$2n_1-2n_2-2\sigma_1^+\sigma_2^-+2\sigma_1^-\sigma_2^+$$

$$-4$$

$$2n_1-2n_2+2\sigma_1^+\sigma_2^--2\sigma_1^-\sigma_2^+$$

$$-2n_1+2n_2-2\sigma_1^+\sigma_2^-+2\sigma_1^-\sigma_2^+$$

Figure 3.44: The jump operators for the nondegenerate case ($g \neq 0$).

$$0 \tag{3.191}$$

$$4\sigma_1^+ + 4\sigma_1^- \tag{3.192}$$

$$4\sigma_2^+ + 4\sigma_2^- \tag{3.193}$$

$$-4 \tag{3.194}$$

$$4n_1 + 2\sigma_1^+(\sigma_2^+ + \sigma_2^-) - 2\sigma_1^-(\sigma_2^+ + \sigma_2^-) - 2 \tag{3.195}$$

$$2i\left(-\sigma_1^+(2n_2-1) - \sigma_1^-(2n_2-1) - \sigma_2^+ + \sigma_2^-\right) \tag{3.196}$$

$$4n_2 + 2\sigma_1^+(\sigma_2^+ - \sigma_2^-) + 2\sigma_1^-(\sigma_2^+ - \sigma_2^-) - 2 \tag{3.197}$$

$$2i\left(-2n_1(\sigma_2^+ + \sigma_2^-) - \sigma_1^+ + \sigma_1^- + \sigma_2^+ + \sigma_2^-\right) \tag{3.198}$$

$$4 \tag{3.199}$$

$$4n_1 - 2\sigma_1^+(\sigma_2^+ + \sigma_2^-) + 2\sigma_1^-(\sigma_2^+ + \sigma_2^-) - 2 \tag{3.200}$$

$$4n_2 - 2\sigma_1^+(\sigma_2^+ - \sigma_2^-) - 2\sigma_1^-(\sigma_2^+ - \sigma_2^-) - 2 \tag{3.201}$$

$$2i\left(2n_1(\sigma_2^+ + \sigma_2^-) - \sigma_1^+ + \sigma_1^- - \sigma_2^+ - \sigma_2^-\right) \tag{3.202}$$

$$2i\left(\sigma_1^+(2n_2-1) + \sigma_1^-(2n_2-1) - \sigma_2^+ + \sigma_2^-\right) \tag{3.203}$$

Figure 3.45: The jump operators for the degenerate case ($g = 0$).

Conclusion

Here's a conclusion, demonstrating the use of all that manual incrementing and table of contents adding that has to happen if you use the starred form of the chapter command. The deal is, the chapter command in LaTeX does a lot of things: it increments the chapter counter, it resets the section counter to zero, it puts the name of the chapter into the table of contents and the running headers, and probably some other stuff.

Appendix A

Computer details

A.1 Julia version information

```
versioninfo()
```

```
Julia Version 1.4.0
Commit b8e9a9ecc6 (2020-03-21 16:36 UTC)
Platform Info:
  OS: Linux (x86_64-pc-linux-gnu)
  CPU: Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz
  WORD_SIZE: 64
  LIBM: libopenlibm
  LLVM: libLLVM-8.0.1 (ORCJIT, haswell)
```

```
using Pkg
Pkg.activate(".")
```

```
 Activating environment at `~/drive/thesis/notebooks/Project.toml`
```

```
Pkg.status()
```

```
Status `~/drive/thesis/notebooks/Project.toml`
  [7d9fca2a] Arpack v0.4.0
```

```
  [b964fa9f] LaTeXStrings v1.2.0
  [eff96d63] Measurements v2.3.0
  [3b7a836e] PGFPlots v3.3.3
  [91a5bcdd] Plots v1.6.12
  [6e0679c1] QuantumOptics v0.8.2
  [1986cc42] Unitful v1.5.0
  [37e2e46d] LinearAlgebra
```

```
using LinearAlgebra
BLAS.vendor()
```

```
:mkl
```

A.2 Notebook Preamble

```
using Plots, LaTeXStrings
using Unitful, Measurements
using LinearAlgebra, Arpack, QuantumOptics

# import PGFPlots: pushPGFPlotsPreamble, popPGFPlotsPreamble
# popPGFPlotsPreamble() # If reevaluating, so no duplicates
# pushPGFPlotsPreamble("
#     \\usepackage{amsmath}
#     \\usepackage{physics}
#     \\usepackage{siunitx}
#     \\usepackage[full]{textcomp} % to get the right copyright, etc.
#     \\usepackage{libertinus-otf}
#     \\usepackage[scaled=.95,type1]{cabin} % sans serif in style of Gill Sans
#     \\usepackage[T1]{fontenc} % LY1 also works
#     \\setmainfont[Numbers={OldStyle,Proportional}]{fbb}
#     \\usepackage[supstfm=fbb-Regular-sup-t1]{superiors}
#     \\usepackage[cal=boondoxo,bb=boondox,frak=boondox]{mathalfa}
#     \\input{latexdefs}
#     ")
# pgfplots()

import PGFPlotsX
```

```
# If reevaluating, so no duplicates
!isempty(PGFPlotsX.CUSTOM_PREAMBLE) && pop!(PGFPlotsX.CUSTOM_PREAMBLE)
push!(PGFPlotsX.CUSTOM_PREAMBLE, "
    \\usepackage{amsmath}
    \\usepackage{physics}
    \\usepackage{siunitx}
    \\usepackage[full]{textcomp} % to get the right copyright, etc.
    \\usepackage{libertinus-otf}
    \\usepackage[scaled=.95,type1]{cabin} % sans serif in style of Gill Sans
    \\usepackage[T1]{fontenc} % LY1 also works
    \\setmainfont[Numbers={OldStyle,Proportional}]{fbb}
    \\usepackage[supstfm=fbb-Regular-sup-t1]{superiors}
    \\usepackage[cal=boondoxo,bb=boondox,frak=boondox]{mathalfa}
    \\input{$(pwd())/latexdefs.tex}
    ");
pgfplotsx()

using PlotThemes
theme(:vibrant,
    size=(400, 300),
    dpi=300,
    titlefontsize=12,
    tickfontsize=11,
    legendfontsize=11,
)
```

References

[1] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3): 379–423, July 1948. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1948.tb01338.x. 1.1.1

[2] Lucien Hardy. Quantum Theory From Five Reasonable Axioms. *ArXivquant-Ph0101012*, September 2001. URL http://arxiv.org/abs/quant-ph/0101012. Comment: 34 pages. Version 4: Improved proofs of K=N$\hat{r}$ and D=DT. Discussion of state update rule after measurement added. Various clarifications in proofs Version 2: Axiom 2 modified and corresponding corrections made to proof in Sec. 8.1. Typos and minor errors fixed. 1.1.2

[3] P. Busch. Quantum States and Generalized Observables: A Simple Proof of Gleason's Theorem. *Phys. Rev. Lett.*, 91(12):120403, September 2003. doi: 10.1103/PhysRevLett.91.120403. URL https://link.aps.org/doi/10.1103/PhysRevLett.91.120403. 1.2

[4] Paul Busch, Marian Grabowski, and Pekka J. Lahti. *Operational Quantum Physics*. Springer Science & Business Media, November 1997. ISBN 978-3-540-59358-4. 1.2

[5] Andrew M. Gleason. Measures on the Closed Subspaces of a Hilbert Space. In C. A. Hooker, editor, *The Logico-Algebraic Approach to Quantum Mechanics: Volume I: Historical Evolution*, The University of Western Ontario Series in Philosophy of Science, pages 123–133. Springer Netherlands, Dordrecht, 1975. ISBN 978-94-010-1795-4. doi: 10.1007/978-94-010-1795-4_7. URL https://doi.org/10.1007/978-94-010-1795-4_7. 6

[6] Lajos Molnár and Patrícia Szokol. Maps on states preserving the relative entropy II. *Linear Algebra and its Applications*, 432(12):3343–3350, July 2010. ISSN 0024-3795. doi: 10.1016/j.laa.2010.01.025. URL https://www.sciencedirect.com/science/article/pii/S0024379510000388. 1.2.2, 2.4

[7] Michael E. Cuffaro and Wayne C. Myrvold. On the Debate Concerning the Proper Characterization of Quantum Dynamical Evolution. *Philosophy of Science*, 80(5):1125–1136, December 2013. ISSN 0031-8248. doi: 10.1086/673733. URL https://www.journals.uchicago.edu/doi/10.1086/673733. 1.2.3

[8] Philip Pechukas. Reduced Dynamics Need Not Be Completely Positive. *Phys. Rev. Lett.*, 73(8):1060–1062, August 1994. doi: 10.1103/PhysRevLett.73.1060. URL https://link.aps.org/doi/10.1103/PhysRevLett.73.1060. 1.2.3

[9] Anil Shaji and E. C. G. Sudarshan. Who's afraid of not completely positive maps? *Physics Letters A*, 341(1):48–54, June 2005. ISSN 0375-9601. doi: 10.1016/j.physleta.2005.04.029. URL https://www.sciencedirect.com/science/article/pii/S0375960105005748. 1.2.3

[10] H. D. Zeh. On the interpretation of measurement in quantum theory. *Found. Phys.*, 1(1):69–76, March 1970. ISSN 1572-9516. doi: 10.1007/BF00708656. URL https://doi.org/10.1007/BF00708656. 2

[11] Maximilian Schlosshauer. Quantum decoherence. *Phys. Rep.*, 831:1–57, 2019. ISSN 0370-1573. doi: 10.1016/j.physrep.2019.10.001. URL http://www.sciencedirect.com/science/article/pii/S0370157319303084. 2

[12] Vittorio Gorini, Andrzej Kossakowski, and E. C. G. Sudarshan. Completely positive dynamical semigroups of N-level systems. *J. Math. Phys.*, 17(5):821–825, 1976. doi: 10.1063/1.522979. URL https://aip.scitation.org/doi/abs/10.1063/1.522979. 2

[13] G. Lindblad. On the generators of quantum dynamical semigroups. *Commun. Math. Phys.*, 119:48, 1976. doi: 10.1007/BF01608499. 2

[14] Heinz Peter Breuer and Francesco Petruccione. *The Theory of Open Quantum Systems*. Oxford University Press, 2002. ISBN 978-0-19-852063-4. 2, 2.12, **2.**, 2.12.1, 2.12.1

[15] Daniel Manzano. A short introduction to the Lindblad master equation. *AIP Adv.*, 10(2):025106, 2020. doi: 10.1063/1.5115323. 2, 2.10, 2.11.4

[16] Steven Weinberg. Quantum mechanics without state vectors. *Phys. Rev. A*, 90(4): 042102, October 2014. doi: 10.1103/PhysRevA.90.042102. URL https://link.aps.org/doi/10.1103/PhysRevA.90.042102. 2.2

[17] Ryogo Kubo. Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems. *J. Phys. Soc. Jpn.*, 12(6):570–586, 1957. doi: 10.1143/JPSJ.12.570. 2.7

[18] Paul C. Martin and Julian Schwinger. Theory of many-particle systems. I. *Phys. Rev.*, 115(6):1342–1373, September 1959. doi: 10.1103/PhysRev.115.1342. URL https://link.aps.org/doi/10.1103/PhysRev.115.1342. 2.7

[19] Daniel A. Steck. Rubidium 87 D line data, 2019, accessed April 20, 2020. URL https://steck.us/alkalidata. 2.8

[20] Daryl W. Preston. Doppler-free saturated absorption: Laser spectroscopy. *Am. J. Phys.*, 64(11):1432–1436, 1996. doi: 10.1119/1.18457. URL https://doi.org/10.1119/1.18457. 2.8

[21] Junyang Ma, Haisu Zhang, Bruno Lavorel, Franck Billard, Edouard Hertz, Jian Wu, Christian Boulet, Jean-Michel Hartmann, and Olivier Faucher. Observing collisions beyond the secular approximation limit. *Nat. Commun.*, 10(1):5780, December 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-13706-0. 2.9, 2.10

[22] Junyang Ma, H. Zhang, B. Lavorel, F. Billard, J. Wu, C. Boulet, J.-M. Hartmann, and O. Faucher. Ultrafast collisional dissipation of symmetric-top molecules probed by rotational alignment echoes. *Phys. Rev. A*, 101(4):043417, April 2020. doi: 10.1103/PhysRevA.101.043417. URL https://link.aps.org/doi/10.1103/PhysRevA.101.043417. 2.9

[23] Elliott Lieb, Theodore Schultz, and Daniel Mattis. Two soluble models of an antiferromagnetic chain. *Annals of Physics*, 16(3):407–466, December 1961. ISSN 0003-4916. doi: 10.1016/0003-4916(61)90115-4. URL http://www.sciencedirect.com/science/article/pii/0003491661901154. 3.1, 3.10

Index