

Министерство образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
“Московский физико-технический институт (государственный
университет)”

Физтех-школа фундаментальной и прикладной физики
Кафедра теоретической астрофизики и квантовой теории поля

Применение тензорных сетей для исследования СПИНОВЫХ СИСТЕМ

Выпускная квалификационная работа (бакалаврская работа)

Работу выполнил:

Студент 625 группы

Измайлов Камиль Ильдарович

Научный руководитель:

к.ф.-м.н., Дымарский Анатолий Яковлевич

Долгопрудный, 2020

Оглавление

Введение	3
Тензорные сети	4
Расчет энергии основного состояния спиновой цепочки при помощи тензорных сетей.....	8
Исследование квантовой запутанности	12
Процессы термализации и транспорта.....	15
Метод выполнения	17
Результаты	19
Заключение.....	21
Список использованной литературы.....	22
Приложение	23

Введение

За последние десятилетия тензорные сети пережили взрывной рост в решении различных проблем. Сюда можно отнести проблемы, связанные с голографическим принципом в теории струн или AdS/CFT соответствием в квантовой гравитации. Но в первую очередь их применение оказалось очень эффективным в исследовании квантовых систем многих тел, в частности спиновых систем, чему и посвящена данная работа. Основной проблемой при изучении систем многих тел является невероятно большая размерность соответствующего Гильбертова пространства или т.н. катастрофа размерностей. К примеру, для системы из N спинов размерность соответствующего Гильбертова пространства равна 2^N , т.е. размерность системы растет экспоненциально с ростом числа частиц. Для многих задач оказывается, что не все состояния в Гильбертовом пространстве одинаково значимы: некоторые из них более релевантны, чем другие. Например, многие гамильтонианы, описывающие взаимодействия частиц друг с другом, являются локальными (взаимодействуют только соседние частицы). Можно показать, что подобная локальность приводит к тому, что низкоэнергетические состояния соответствуют лишь небольшой части исходного гильбертова пространства. Тензорные сети позволяют учитывать такую локальность и эффективно решать проблему больших размерностей.

В данной работе были изучены различные применения тензорных сетей для исследования одномерных спиновых цепей. В качестве основной модели была рассмотрена модель Изинга. Для данной модели были найдены основные состояния и их энергии для цепей различной длины, рассмотрена квантовая запутанность, на основании поведения которой был обнаружен фазовый переход, а также рассмотрены процессы термализации.

Основным инструментом изучения было численное моделирование на компьютере. Весь программный код писался на языке Python. Выбор этого языка обусловлен его удобством, простотой и тем, что для него существует огромное число различных библиотек (в том числе и по тензорным сетям), которые значительно облегчают процесс написания кода. Весь написанный код можно найти в приложении к данной работе.

Тензорные сети

Исследование квантовых систем многих тел является, вероятно, самой сложной проблемой в физике конденсированных сред. Стандартным подходом к исследованию подобных моделей является создание упрощенных моделей, отражающих основные физические свойства системы, и дальнейшие попытки их решения. Точные аналитические решения удается получить в исключительно удачных случаях, в остальных же ситуациях приходится использовать численные методы для расчета. Тензорные сети как раз и являются таким эффективным методом расчета.

Для определения термина “тензорные сети” рассмотрим систему из N частиц. Пусть каждая частица может принимать p состояний. Тогда любое состояние системы можно представить в следующем виде:

$$|\Psi\rangle = \sum_{n_1 n_2 \dots n_N} \Psi^{n_1, n_2, \dots, n_N} |n_1\rangle \otimes |n_2\rangle \otimes \dots \otimes |n_N\rangle \quad (1)$$

В данном выражении $\Psi^{n_1, n_2, \dots, n_N}$ есть p^N комплексных чисел (независимых с точностью до нормировки). Теперь заметим, что числа $\Psi^{n_1, n_2, \dots, n_N}$ можно рассматривать как компоненты тензора Ψ , у которого N индексов n_1, n_2, \dots, n_N , каждый из которых может принимать p значений. В терминах тензорных сетей тензором называется многомерный массив комплексных чисел $\Psi^{n_1, n_2, \dots, n_N}$. Заметим, что вся информация о состоянии системы содержится в тензоре $\Psi^{n_1, n_2, \dots, n_N}$, так что в дальнейшем в большинстве случаев при описании состояний вместо кет-вектора $|\Psi\rangle$ будем использовать обозначение $\Psi^{n_1, n_2, \dots, n_N}$. Учет каждой компоненты тензора $\Psi^{n_1, n_2, \dots, n_N}$ представляет из себя тяжелую вычислительную задачу, и зачастую описание системы при помощи учета всех таких чисел оказывается избыточным и вычислительно неэффективным для решения задач многих тел. Одна из целей тензорных сетей как раз и является упрощение описания состояния $\Psi^{n_1, n_2, \dots, n_N}$. Подобное упрощение достигается за счет разбиения “большого” тензора $\Psi^{n_1, n_2, \dots, n_N}$ на много “маленьких” тензоров меньшего ранга, с которым гораздо легче работать. Набор таких маленьких тензоров называется *тензорной сетью*.

Рассмотрим случай $N = 2$. Пусть каждая частица может принимать 2 состояния ($p=2$): со спином вверх и со спином вниз. Если система находится в чистом состоянии (нет запутанности), то выполнено

$$\Psi^{n_1, n_2} = A^{n_1} A^{n_2}, \quad (2)$$

где индексы n_1 и n_2 пробегает 2 значения, а вектора A^{n_1} и A^{n_2} есть амплитуды состояний (в данном случае спиноры Паули), описывающие каждый спин по отдельности.

Если система находится в смешанном состоянии, то выполнено

$$\Psi^{n_1, n_2} = \sum_j A_j^{n_1} A_j^{n_2}, \quad (3)$$

где индекс j как раз отвечает за запутанность нашего состояния. В данном случае тензор Ψ^{n_1, n_2} разбивается на два тензора $A_j^{n_1}$ и $A_j^{n_2}$. В данном случае тензорной сетью называется набор $\{A_j^{n_1}, A_j^{n_2}\}$. Ясно, что для чистого состояния индекс j принимает только одно значение и выражение (3) переходит в (2).

Для системы из N частиц разложение для $\Psi^{n_1, n_2, \dots, n_N}$ имеет вид

$$\Psi^{n_1, n_2, \dots, n_N} = \sum_{j_1, j_2, \dots, j_N} A_{j_1}^{n_1} A_{j_1 j_2}^{n_2} A_{j_2 j_3}^{n_3} \dots A_{j_N}^{n_N}, \quad (4)$$

Индексы j_1, j_2, \dots, j_N , отвечающие за запутанность, называются *bond indices*, а количество значений, которые они могут принимать, называется *bond dimension*.

Тензорная сеть вида (4) называется Matrix Product State (MPS). MPS является наиболее распространенным примером тензорной сети, так она является очень мощным инструментом для симуляций одномерных квантовых систем. Существуют тензорные сети и другого вида (к примеру, Projected Entangled Pair States (PEPS), см. рис.1), но в данной работе изучаются одномерные спиновые цепочки, поэтому в качестве инструмента используется только MPS.

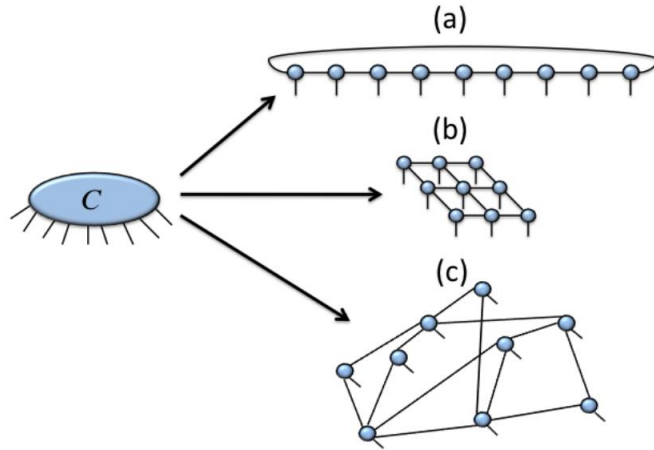


Рис.1 Разложение тензора C на тензорные сети (a) MPS с периодическими граничными условиями (b) PEPS с открытыми границами (c) произвольную тензорную сеть

Подробное введение в аппарат тензорных сетей с различными примерами их применения можно найти в работе [1].

На языке тензорных сетей можно описывать не только состояния, но и операторы, действующие на них. Вернемся к системе из N частиц

$$|\Psi\rangle = \sum_{i_1 i_2 \dots i_N} \Psi^{n_1, n_2 \dots n_N} |n_1\rangle \otimes |n_2\rangle \otimes \dots \otimes |n_N\rangle \quad (5)$$

Рассмотрим оператор \hat{O} , действующий на это состояние. Вопрос: можно ли представить оператор \hat{O} в виде, похожим на (5)? Да, можно. Такое разложение имеет вид

$$\hat{O} = \sum_{i' i} O_{n'_1, n'_2 \dots n'_N}^{n_1, n_2 \dots n_N} |n_1\rangle \otimes |n_2\rangle \otimes \dots \otimes |n_N\rangle \langle n'_1| \otimes \langle n'_2| \otimes \dots \otimes \langle n'_N| \quad (6)$$

Нетрудно видеть, что любой оператор, действующий на данную систему, можно представить в виде (6). Действительно, обозначим $|n_1\rangle \otimes |n_2\rangle \otimes \dots \otimes |n_N\rangle = |n_1, n_2 \dots n_N\rangle$.

Теперь используя

$$\hat{O}|n_1, n_2 \dots n_N\rangle = \sum_{i'_1, i'_2 \dots i'_N} O_{n'_1, n'_2 \dots n'_N}^{n_1, n_2 \dots n_N} |n_1, n_2 \dots n_N\rangle, \quad (7)$$

находим

$$O_{n'_1, n'_2 \dots n'_N}^{n_1, n_2 \dots n_N} = \langle n'_1, n'_2 \dots n'_N | \hat{O} |n_1, n_2 \dots n_N\rangle \quad (8)$$

Как видим, оператор \hat{O} полностью описывается тензором $O_{n'_1, n'_2, \dots, n'_N}^{n_1, n_2, \dots, n_N}$. Также, как и для состояний тензор $O_{n'_1, n'_2, \dots, n'_N}^{n_1, n_2, \dots, n_N}$ можно разложить в тензорную сеть

$$O_{n'_1, n'_2, \dots, n'_N}^{n_1, n_2, \dots, n_N} = \sum_{j_1, j_2, \dots, j_N} W_{j_1}^{n_1, n'_1} W_{j_1 j_2}^{n_2, n'_2} W_{j_2 j_3}^{n_3, n'_3} \dots W_{j_N}^{n_N, n'_N} \quad (9)$$

Представление $O_{n'_1, n'_2, \dots, n'_N}^{n_1, n_2, \dots, n_N}$ в виде тензорной сети (9) называется Matrix Product Operator (MPO). Графически такое разложение представлено на рис.2. В дальнейшем при изучении спиновых цепочек в качестве гамильтониана будет использоваться модель Изинга, которую мы будем представлять в виде MPO. Детальный обзор MPO и представлений операторов в его виде (в том числе и гамильтониана Изинга) можно найти здесь [2]

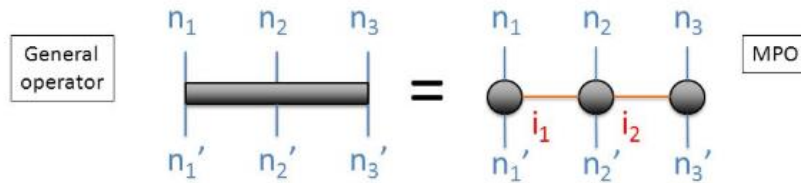


Рис.2 представление оператора в виде Matrix product operator (MPO)

Расчет энергии основного состояния спиновой цепочки при помощи тензорных сетей

В данном разделе приведем первый пример использования тензорных сетей для исследования спиновых цепочек. Рассмотрим одномерную спиновую цепочку с периодическими граничными условиями. В качестве гамильтониана возьмем неинтегрируемую модель Изинга с поперечным и продольным магнитными полями:

$$H = - \sum_i \sigma_i^z \sigma_{i+1}^z + h \sum_i \sigma_i^z + g \sum_i \sigma_i^x \quad (10)$$

Поставим вопрос о нахождении основного состояния данной системы и энергии этого состояния. Заметим, что для заданного гамильтониана H и произвольного нетривиального состояния $|\Psi\rangle$ выполнено:

$$\frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} \geq E_0, \quad (11)$$

где E_0 – энергия основного состояния. Для нахождения E_0 найдем следующий минимум:

$$\min_{|\Psi\rangle} (\langle \Psi | H | \Psi \rangle - \lambda \langle \Psi | \Psi \rangle), \quad (12)$$

где λ – множитель Лагранжа.

В качестве описания состояния цепочки спинов $|\Psi\rangle$ возьмем тензорную сеть Matrix Product State

$$\Psi^{n_1, n_2, \dots, n_N} = \sum_{j_1, j_2, \dots, j_N} A_{j_1}^{n_1} A_{j_1 j_2}^{n_2} A_{j_2 j_3}^{n_3} \dots A_{j_N}^{n_N}, \quad (13)$$

а гамильтониан опишем как Matrix Product Operator

$$H_{n'_1, n'_2, \dots, n'_N}^{n_1, n_2, \dots, n_N} = \sum_{j_1, j_2, \dots, j_N} W_{j_1}^{n_1, n'_1} W_{j_1 j_2}^{n_2, n'_2} W_{j_2 j_3}^{n_3, n'_3} \dots W_{j_N}^{n_N, n'_N} \quad (14)$$

Теперь перепишем $\langle \Psi | H | \Psi \rangle$ в терминах тензорных сетей

$$\langle \Psi | H | \Psi \rangle = \sum_{n, n'} \Psi^{n_1, n_2, \dots, n_N} H_{n'_1, n'_2, \dots, n'_N}^{n_1, n_2, \dots, n_N} \Psi^{n'_1, n'_2, \dots, n'_N} \quad (15)$$

В общем случае минимизация такого выражения должна происходить по всем параметрам тензорной сети $\Psi^{n_1, n_2, \dots, n_N}$. Такой подход является весьма тяжелым для реализации.

Вместо этого проведем минимизацию *тензор за тензором*. Это означает, что будем проводить минимизацию относительно первого тензора $A_{j_1}^{n_1}$ исходной тензорной сети, зафиксировав остальные. Далее перейдем ко второму тензору $A_{j_1 j_2}^{n_2}$ и повторим минимизацию, затем к третьему и т.д. Будем проходить по всей цепочке до тех пор, пока не будет достигнута нужная точность.

Рассмотрим процесс минимизации тензор за тензором подробнее. Представим, что мы зафиксировали все тензоры в тензорной сети, кроме одного, который мы обозначим как A . Теперь коэффициенты тензора A являются вариационными параметрами задачи:

$$\min_A (\langle \Psi | H | \Psi \rangle - \lambda \langle \Psi | \Psi \rangle) = \min_A (\vec{A}^\dagger H_{eff} \vec{A} - \lambda \vec{A}^\dagger N \vec{A}), \quad (16)$$

где \vec{A} это вектор, составленный из компонент тензора A , H_{eff} это эффективный гамильтониан, представленный в форме матрицы, N – матрица ответственная за нормировку. Переход от тензоров к векторам графически изображен на рис.3.

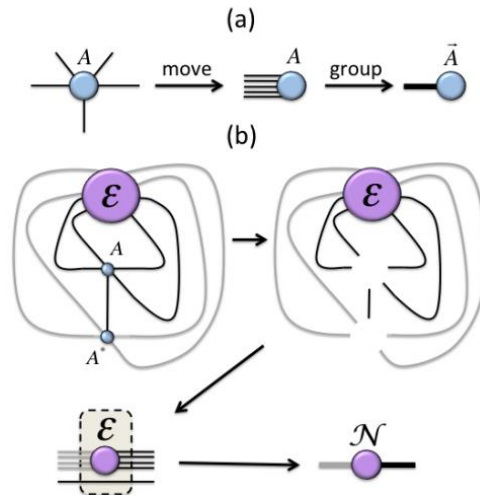


Рис.3. (a) Переход от тензора A к вектору \vec{A} ; (b) представление матрицы N

Возвращаясь к задаче минимизации, получаем условие

$$\frac{\partial}{\partial \vec{A}} (\vec{A}^\dagger H_{eff} \vec{A} - \lambda \vec{A}^\dagger N \vec{A}) = 0 \quad (17)$$

Отсюда получаем т.н. *the generalized eigenvalue problem*

$$H_{eff}\vec{A} = \lambda N\vec{A} \quad (18)$$

Решений у этого уравнения существует несколько*. Выбирая из них то, которое соответствует минимальному значению λ , получаем решение минимизационной задачи, которым затем в исходной тензорной сети заменим тензор A (для этого преобразуем вектор \vec{A} назад в форму тензора). Соответствующее минимальное значение параметра λ обозначим как E_0^A .

Находя E_0^A для одного тензора, переходим к следующему, решаем минимизационную задачу для него и т.д. Пройдя всю цепочку слева направо, начнем тот же процесс в другую сторону. После нескольких таких проходов будет получена последовательность $\{E_0^A\}$. Данная последовательность сходится к энергии основного состояния системы E_0 .

Ниже приведены таблицы с полученными таким способом энергиями E_0 с точностью до 2 знака. Значения приведены для цепочек различной длины и различных параметров h .

Длина цепочки $N = 3; g = 1.05$

h	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Энергия, E_0	-3.67	-3.81	-3.99	-4.21	-4.44	-4.68	-4.93	-5.19	-5.45

Длина цепочки $N = 4; g = 1.05$

h	0.1	0.2	0.3	0.4	0.5	0,6	0.7	0.8	0.9
Энергия, E_0	-5.02	-5.24	-5.52	-5.83	-6.16	-6.49	-6.83	-7.18	-7.53

Длина цепочки $N = 5; g = 1.05$

h	0.1	0.2	0.3	0.4	0.5	0,6	0.7	0.8	0.9
Энергия, E_0	-6.38	-6.70	-7.07	-7.46	-7.88	-8.31	-8.74	-9.18	-9.63

Длина цепочки $N = 6; g = 1.05$

h	0.1	0.2	0.3	0.4	0.5	0,6	0.7	0.8	0.9
Энергия, E_0	-7.75	-8.15	-8.61	-9.10	-9.60	-10.12	-10.64	-11.18	-11.72

* Для решения этого уравнения существуют различные библиотеки на разных языках программирования. Один из таких пакетов я и использовал в своей работе.

Длина цепочки $N = 7$; $g = 1.05$

h	0.1	0.2	0.3	0.4	0.5	0,6	0.7	0.8	0.9
Энергия, E_0	-9.12	-9.61	-10.16	-10.73	-11.32	-11.93	-12.55	-13.18	-13.81

Длина цепочки $N = 8$; $g = 1.05$

h	0.1	0.2	0.3	0.4	0.5	0,6	0.7	0.8	0.9
Энергия, E_0	-10.50	-11.07	-11.70	-12.37	-13.05	-13.75	-14.45	-15.17	-15.90

Длина цепочки $N = 9$; $g = 1.05$

h	0.1	0.2	0.3	0.4	0.5	0,6	0.7	0.8	0.9
Энергия, E_0	-11.87	-12.53	-13.25	-14.00	-14.77	-15.56	-16.36	-17.17	-17.99

Исследование квантовой запутанности

Тензорные сети позволяют эффективно изучать квантовую запутанность. Как было сказано ранее, при представлении исходного состояния в виде тензорной сети возникают дополнительные степени свободы – *bond dimensions* (bd). Покажем, что bd как раз и отвечают за квантовую запутанность.

Для наглядности, начнем с примера тензорной сети, которая называется *Projected Entangled Pair State* (PEPS). Оценим энтропию запутанности для граничных тензоров длиной L (см. рис.4). Для этого введем мультииндекс $\bar{\alpha} = \{\alpha_1, \alpha_2 \dots \alpha_{4L}\}$, который представляет из себя набор всех индексов bd на границе. Пусть каждый из индексов α_i принимает D значений. Тогда $\bar{\alpha}$ принимает D^{4L} значений. Перепишем теперь в терминах кет-векторов исходное состояние через “внутреннюю” и “внешнюю” части PEPS:

Основным инструментом изучения запутанности является матрицы плотности. Введем редуцированную матрицу плотности

$$\rho_{in} = \sum_{\bar{\alpha}, \bar{\alpha}'} X_{\bar{\alpha}, \bar{\alpha}'} |in(\bar{\alpha})\rangle \langle in(\bar{\alpha}')| \quad (19)$$

где $X_{\bar{\alpha}, \bar{\alpha}'} \equiv \langle out(\bar{\alpha}') | out(\bar{\alpha}) \rangle$. Из приведенного выражения ясно, что ранг редуцированной матрицы плотности не превышает D^{4L} . Теперь, обратившись к определению энтропии запутанности $S(L) = -\text{tr}(\rho_{in} \log \rho_{in})$, получим следующую оценку

$$S(L) \leq 4L \log D \quad (20)$$

Из этой оценки можно заключить что каждый “разорванный” bd вносит вклад в энтропию не больше, чем $\log D$.

Перейдем теперь к изучению энтропии запутанности на примере модели Изинга и увидим, что в данной модели имеется фазовый переход. Рассмотрим гамильтониан вида

$$H = -1/4 \sum_i \sigma_i^z \sigma_{i+1}^z + 1/2 * g \sum_i \sigma_i^x, \quad (21)$$

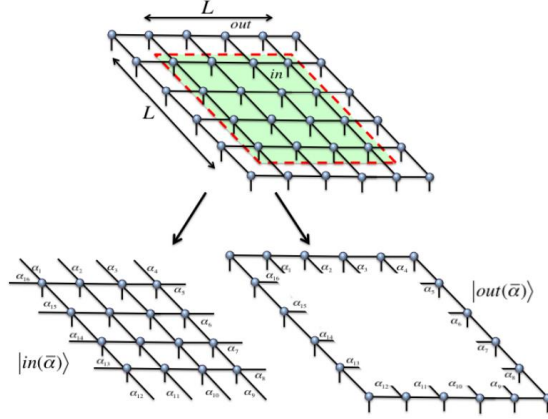


Рис.5 разложение состояния на “внутреннюю” и “внешнюю” части

В качестве состояния берем Matrix Product State. Рассмотрим основное состояние для цепи из N спинов. Алгоритм получения основного состояния описан в прошлом разделе. Обозначим это состояние $\Psi^{n_1, n_2 \dots n_N}$. По аналогии с примером для PEPS, посчитаем энтропию запутанности для отдельной подцепочки длины M исходной цепочки. Для этого посчитаем редуцированную матрицу плотности

$$\begin{aligned} \rho_{in}^{n_1' \dots n_M', n_1 \dots n_M} &= \sum \Psi^{n_1, n_2 \dots n_M, n_{M+1} \dots n_N} \Psi^{n_1', n_2' \dots n_M', n_{M+1} \dots n_N} = \\ &= \sum_{n_{M+1}, n_{M+2} \dots n_N} \sum_{j_1, j_2 \dots j_N} \sum_{j_1, j_2 \dots j_N} A_{j_1}^{n_1} A_{j_1 j_2}^{n_2} \dots A_{j_M j_{M+1}}^{n_M} \dots A_{j_N}^{n_N} A_{j_1'}^{n_1'} A_{j_1' j_2'}^{n_2'} \dots A_{j_M' j_{M+1}'}^{n_M'} \dots A_{j_N'}^{n_N'} \end{aligned} \quad (22)$$

Глядя на выражение (19), выделим матрицу $X_{\bar{\alpha}, \bar{\alpha}'}$ (Для цепочки мультииндекс $\bar{\alpha}$ переходит в обычный индекс j_M и $X_{\bar{\alpha}, \bar{\alpha}'}$ является обычной матрицей)

$$X_{j_M, j_M'} = \sum_{n_{M+1}, n_{M+2} \dots n_N} \sum_{j_{M+1}, j_{M+2} \dots j_N} \sum_{j_{M+1}', j_{M+2}' \dots j_N'} A_{j_M j_{M+1}}^{n_M} \dots A_{j_N}^{n_N} A_{j_M' j_{M+1}'}^{n_M'} \dots A_{j_N'}^{n_N'} \quad (23)$$

Далее посчитаем энтропию запутанности

$$S(M) = -\text{tr}(\rho_{in} \log \rho_{in}) = - \sum_{j_M} X_{j_M, j_M} \log X_{j_M, j_M} \quad (24)$$

Построим теперь графики для $S(M)$ для различных констант g и увидим, что в системе имеется фазовый переход при $g \approx 0,5$ (рис.6).

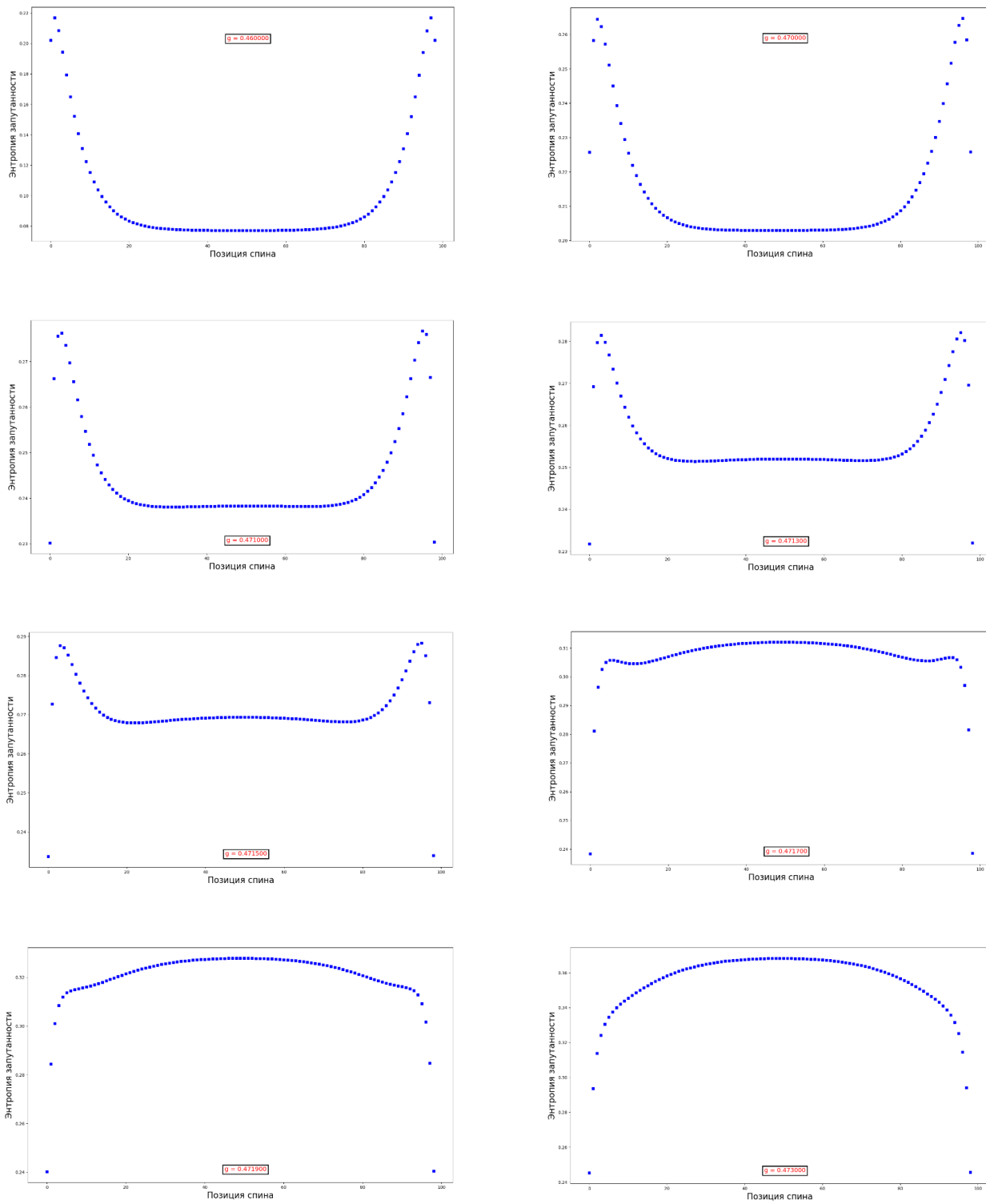


Рис.6 График зависимости $S(M)$ для различных констант g для цепочки из 100 спинов

Как видно из рис.6 в системе (20) имеется фазовый переход при $g_3 \approx 0.4715$. Это значение отличается от теоретического $g_T = 0.5$. Различие g_T и g_3 объясняется конечным размером исследуемой цепочки и равенство $g_3 = g_T$ достигается при $N \rightarrow \infty$.

Процессы термализации и транспорта

В данной главе поставим вопрос об изучении термализации (thermalization) квантовых систем многих тел. В общем случае термализацией называется процесс, при котором система, находясь в некотором начальном состоянии, переходит к состоянию термодинамического равновесия [4]. В квантовом случае процесс термализации состоит в том, что локальные наблюдаемые величины стремятся к своим равновесным значениям. Процесс термализации для интегрируемых квантовых систем отличается от такового для неинтегрируемых. В первом случае удастся показать, как системы релаксируют к своим стационарным состояниям [5], в то время как для неинтегрируемых систем природа термализации неясна. Существует т.н. *eigenstate thermalization hypothesis* (ETH), которая утверждает, что для неинтегрируемых систем можно построить локальные операторы, позволяющие малым подсистемам исходной системы достигать термодинамического равновесия. В данной главе рассмотрим неинтегрируемую модель Изинга, для которой единственной известной сохраняющейся величиной является энергия. Мы попробуем при помощи тензорных сетей построить операторы, обеспечивающие подобную релаксацию и, сравнив их с процессом диффузии, понять отвечают ли они диффузии или некому другому неизвестному процессу транспорта.

Рассмотрим бесконечномерную спиновую цепочку с гамильтонианом

$$H = - \sum_i \sigma_i^z \sigma_{i+1}^z + h \sum_i \sigma_i^z + g \sum_i \sigma_i^x \quad (25)$$

Выделим из исходной цепочки подцепочку длиной M и рассмотрим произвольный эрмитовый бесследовый оператор \hat{O}_M , который действует на эту подцепочку. Динамика этого оператора описывается его коммутатором с Гамильтонианом $[\hat{O}_M, H]$. Поставим вопрос, о нахождении *самого медленного* оператор \hat{O}_M , определение которого дано ниже. Для количественной оценки коммутатора используем квадрат нормы Фробениуса $\text{tr}(\hat{O}_M \hat{O}_M^\dagger)$ (несмотря на то, что оператор \hat{O}_M эрмитовый, сохраним знак эрмитового сопряжения † по причинам, описанным ниже(см. Метод выполнения)). Введем функцию $f(\hat{O}_M)$

$$f(\hat{O}_M) = \frac{\text{tr}([\hat{O}_M, H][\hat{O}_M, H]^\dagger)}{\text{tr}(\hat{O}_M \hat{O}_M^\dagger)} \quad (26.1)$$

Для удобства будем считать \hat{O}_M нормированным на единицу и опустим знаменатель в (25.1)

$$f(\hat{O}_M) = \frac{\text{tr}([\hat{O}_M, H][\hat{O}_M, H]^\dagger)}{\text{tr}(\hat{O}_M \hat{O}_M^\dagger)} = \text{tr}([\hat{O}_M, H][\hat{O}_M, H]^\dagger) \quad (26.2)$$

Обозначим $\lambda(M) = \min_{\{\hat{O}_M\}} f(\hat{O}_M)$, где $\{\hat{O}_M\}$ -семейство всех бесследовых эрмитовых

операторов. Оператор \hat{O}_M , реализующий данный минимум, как раз и будет искомым *самым медленным* оператором. Покажем, что функция $f(\hat{O}_M)$ будет ограничивать время термализации (англ. *thermalization*) для оператора \hat{O}_M . В момент времени $t=0$ поместим систему в некоторое начальное смешанное состояние, описываемое матрицей плотности $\rho = I/Z + \epsilon \hat{O}_M$, где число Z ответственно за нормировку, а ϵ выбрано для того, чтобы сделать ρ неотрицательной. В данном случае оператор \hat{O}_M рассматривается как малая неоднородность. Определим $o_M(t)$ как среднее значение оператора \hat{O}_M/ϵ в момент времени t

$$o_M(t) = \left(\frac{1}{\epsilon}\right) \text{tr}(\rho \hat{O}_M(t)) = \text{tr}(\hat{O}_M \hat{O}_M(t)) \quad (27)$$

Воспользуемся неравенством Коши-Буняковского

$$\left| \frac{d^2 o_M(t)}{dt^2} \right| = |\text{tr}([\hat{O}_M(t), H][\hat{O}_M, H])| \leq f(\hat{O}_M) \quad (28)$$

Отсюда оценим разницу между $o_M(t)$ и $o_M(0)$ (заметим, что $o_M(0) = 1$)

$$|o_M(t) - o_M(0)| = \left| \int_0^t d\tau \int_0^\tau d\tau' \frac{d^2 o_M(\tau')}{d\tau'^2} \right| \leq \frac{f(\hat{O}_M) t^2}{2} \quad (29)$$

Обозначим o_M в термодинамическом пределе как o_M^{th} . Нетрудно видеть, что для o_M^{th} выполнено $o_M^{th} = \text{tr}(\hat{O}_M) = 0$. Теперь получаем оценку для разницы o_M^{th} и $o_M(t)$

$$|o_M(t) - o_M^{th}| \geq |o_M(0) - o_M^{th}| - |o_M(0) - o_M(t)| \geq 1 - \frac{f(\hat{O}_M)t^2}{2} \quad (30)$$

Из уравнения (30) видим, что $f(\hat{O}_M)$ ограничивает снизу время релаксации $o_M(t)$ к своему термодинамическому пределу o_M^{th} (время термализации): $\tau \geq f(\hat{O}_M)^{-1/2}$. Вспомним, про $\lambda(M)$. Так как $\lambda(M)$ является минимумом $f(\hat{O}_M)$ при $t = 0$, то получим, что оператор, реализующий данный минимум, будет самым медленно релаксирующим.

Метод выполнения

Опишем метод нахождения $\lambda(M)$ и соответствующего оператора \hat{O}_M при помощи тензорных сетей. Как и в прошлых разделах будем описывать \hat{O}_M и H через Matrix Product Operator. Для оператора H , действующего на бесконечную цепочку, будем рассматривать только ту его часть, которая действует на подцепочку, на которую действует оператор \hat{O}_M

$$H_{n'_1, n'_2, \dots, n'_N}^{n_1, n_2, \dots, n_N} = \sum_{j_1, j_2, \dots, j_N} W_{j_1}^{n_1, n'_1} W_{j_1 j_2}^{n_2, n'_2} W_{j_2 j_3}^{n_3, n'_3} \dots W_{j_N}^{n_N, n'_N} \quad (31)$$

$$O_M_{n'_1, n'_2, \dots, n'_M}^{n_1, n_2, \dots, n_M} = \sum_{j_1, j_2, \dots, j_M} V_{j_1}^{n_1, n'_1} V_{j_1 j_2}^{n_2, n'_2} V_{j_2 j_3}^{n_3, n'_3} \dots V_{j_M}^{n_M, n'_M} \quad (32)$$

Сделаем несколько замечаний.

Во-первых, заметим, что получить $\lambda(M)$ можно, минимизируя $f(\hat{O}_M)$ не только по эрмитовым операторам (именно поэтому в выражении (26.1) мы сохранили знак \dagger .) Действительно, любой оператор \hat{O}_M можно представить в виде суммы $\hat{O}_M = \hat{A} + i\hat{B}$, где \hat{A} и \hat{B} эрмитовые операторы. Тогда получим

$$\begin{aligned} f(\hat{O}_M) &= \text{tr}([\hat{O}_M, H][\hat{O}_M, H]^\dagger) = \text{tr}([\hat{A} + i\hat{B}, H][\hat{A} + i\hat{B}, H]^\dagger) = \\ &= \text{tr}([\hat{A}, H][\hat{A}, H]^\dagger) - i\text{tr}([\hat{A}, H][\hat{B}, H]^\dagger) + i\text{tr}([\hat{A}, H][\hat{B}, H]^\dagger) \\ &+ \text{tr}([\hat{B}, H][\hat{B}, H]^\dagger) = \text{tr}([\hat{A}, H]^2) + \text{tr}([\hat{B}, H]^2) = f(\hat{A}) + f(\hat{B}) \end{aligned} \quad (33)$$

Из выражения (33) видно, что минимизация $f(\hat{O}_M)$ по произвольным операторам эквивалентна минимизации по эрмитовым, поэтому в дальнейшем для простоты будем искать минимум через произвольные операторы (но при этом бесследовые) .

Во-вторых заметим, что оператор $O_{n'_1, n'_2, \dots, n'_M}^{n_1, n_2, \dots, n_M}$, представленный в виде МРО, можно представить как состояние Matrix Product State в цепочке кубитов с размерностью $d = 4$. Действительно, приведем $O_{n'_1, n'_2, \dots, n'_M}^{n_1, n_2, \dots, n_M}$ к виду Matrix Product State путем объединения пар индексов (n_i, n'_i) в один индекс $\tilde{n}_i = n'_i * n_i$. Отсюда получаем представление $O^{\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_M}$

$$O_M^{\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_M} = \sum_{j_1, j_2, \dots, j_M} V_{j_1}^{\tilde{n}_1} V_{j_1 j_2}^{\tilde{n}_2} V_{j_2 j_3}^{\tilde{n}_3} \dots V_{j_M}^{\tilde{n}_M}, \quad (34)$$

где тензоры $V_{j_i}^{\tilde{n}_i}$ определены следующим образом

$$V_{j_k j_l}^{\tilde{n}_i} \equiv V_{j_k j_l}^{n'_i * n_i} = V_{j_k j_l}^{n_i, n'_i} \quad (35)$$

Теперь оператор \hat{O}_M можно представить как состояние $|O_M\rangle$. Как было сказано ранее, мы рассматриваем нормированные \hat{O}_M . Тогда нетрудно видеть, что и состояния $|O_M\rangle$ будут нормированными

$$\langle O_M | O_M \rangle = \text{tr}(\hat{O}_M \hat{O}_M^\dagger) = \sum_{\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_M} O_M^{\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_M} O_M^{\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_M} = 1, \quad (36)$$

Выражение (26.2) для функции $f(\hat{O}_M)$ можно переписать в следующем виде

$$f(\hat{O}_M) = \text{tr}([\hat{O}_M, H][\hat{O}_M, H]^\dagger) = \langle O_M | \tilde{H} | O_M \rangle, \quad (37)$$

где \tilde{H} - это новый эффективный гамильтониан, действующий на состояния $|O_M\rangle$. Связь между H и \tilde{H} в терминах тензорной сети можно увидеть при помощи равенства (30)

$$\begin{aligned} f(\hat{O}_M) &= \text{tr}([\hat{O}_M, H][\hat{O}_M, H]^\dagger) = \text{tr}((\hat{O}_M H - H \hat{O}_M)(\hat{O}_M H - H \hat{O}_M)^\dagger) = \\ &= \text{tr}(\hat{O}_M H H \hat{O}_M^\dagger - H \hat{O}_M H \hat{O}_M^\dagger - \hat{O}_M H \hat{O}_M^\dagger H + H \hat{O}_M \hat{O}_M^\dagger H) = \\ &= \text{tr}(H \hat{O}_M^\dagger \hat{O}_M H - 2H \hat{O}_M H \hat{O}_M^\dagger + H \hat{O}_M \hat{O}_M^\dagger H) = \end{aligned} \quad (38)$$

$$\begin{aligned}
&= \sum_{i,n,j,k} H_{i_1,i_2\dots i_M}^{n_1,n_2\dots n_M} O_{i_1,i_2\dots i_M}^{j_1,j_2\dots j_M} O_{k_1,k_2\dots k_M}^{j_1,j_2\dots j_M} H_{n_1,n_2\dots n_M}^{k_1,k_2\dots k_M} - \\
&\quad - 2H_{j_1,j_2\dots j_M}^{n_1,n_2\dots n_M} O_{i_1,i_2\dots i_M}^{j_1,j_2\dots j_M} H_{k_1,k_2\dots k_M}^{i_1,i_2\dots i_M} O_{k_1,k_2\dots k_M}^{n_1,n_2\dots n_M} + \\
&\quad + H_{i_1,i_2\dots i_M}^{n_1,n_2\dots n_M} O_{j_1,j_2\dots j_M}^{i_1,i_2\dots i_M} O_{j_1,j_2\dots j_M}^{k_1,k_2\dots k_M} H_{n_1,n_2\dots n_M}^{k_1,k_2\dots k_M} = \\
&= \sum_{i,n,j,k} O_M^{\bar{k}_1,\bar{k}_2\dots\bar{k}_M} \tilde{H}_{j_1,\tilde{j}_2\dots\tilde{j}_M}^{\bar{k}_1,\bar{k}_2\dots\bar{k}_M} O_M^{\tilde{j}_1,\tilde{j}_2\dots\tilde{j}_M}
\end{aligned}$$

В последней строчке был осуществлен переход от оператора Matrix Product Operator $O_M^{j_1,j_2\dots j_M}_{i_1,i_2\dots i_M}$ к состоянию Matrix Product State $O_M^{\bar{k}_1,\bar{k}_2\dots\bar{k}_M}$.

Из выражения (37) видно, что задача нахождения $\lambda(M)$ сводится к нахождению минимума $\langle O_M | \tilde{H} | O_M \rangle$. Заметим, что данная задача эквивалентна задаче нахождения основного состояния и его энергии для цепочки кубитов с гамильтонианом \tilde{H} . Иными словами, для нахождения $\lambda(M)$ можно воспользоваться ранее описанным алгоритмом нахождения основного состояния цепочки спинов, сделав замену $\Psi^{n_1,n_2\dots n_N} \rightarrow O_M^{\bar{k}_1,\bar{k}_2\dots\bar{k}_M}$, $H_{n'_1,n'_2\dots n'_N} \rightarrow \tilde{H}_{j_1,\tilde{j}_2\dots\tilde{j}_M}^{\bar{k}_1,\bar{k}_2\dots\bar{k}_M}$.

Результаты

Используя, описанный выше алгоритм, были найдены выражения для $\lambda(M)$ для значений M от 2 до 10. Для оператора \hat{E}_M , обеспечивающего диффузию энергии, зависимость $f(\hat{E}_M)$ от длины цепочки M равна $f(\hat{E}_M) \sim 1/M^2$ [3]. Предполагая зависимость для $\lambda(M)$ степенной построим график зависимости $\ln(\lambda(M))$ от $\ln(M)$ и сравним полученные результаты с соответствующей зависимостью для \hat{E}_M . Для $\ln \lambda(M)$ была получена зависимость $\ln \lambda(M) = 1.5 - 2.8 \ln M$. Откуда $\lambda(M) \sim 1/M^{2.8}$, т.е. $\lambda(M)$ убывает быстрее, чем $f(\hat{E}_M)$. Подобное расхождение может быть намеком на наличие некоего другого транспорта в системе, но для подтверждения или опровержения такой гипотезы нужно изучить поведение $\lambda(M)$ при больших M , для чего требуется большие вычислительные мощности

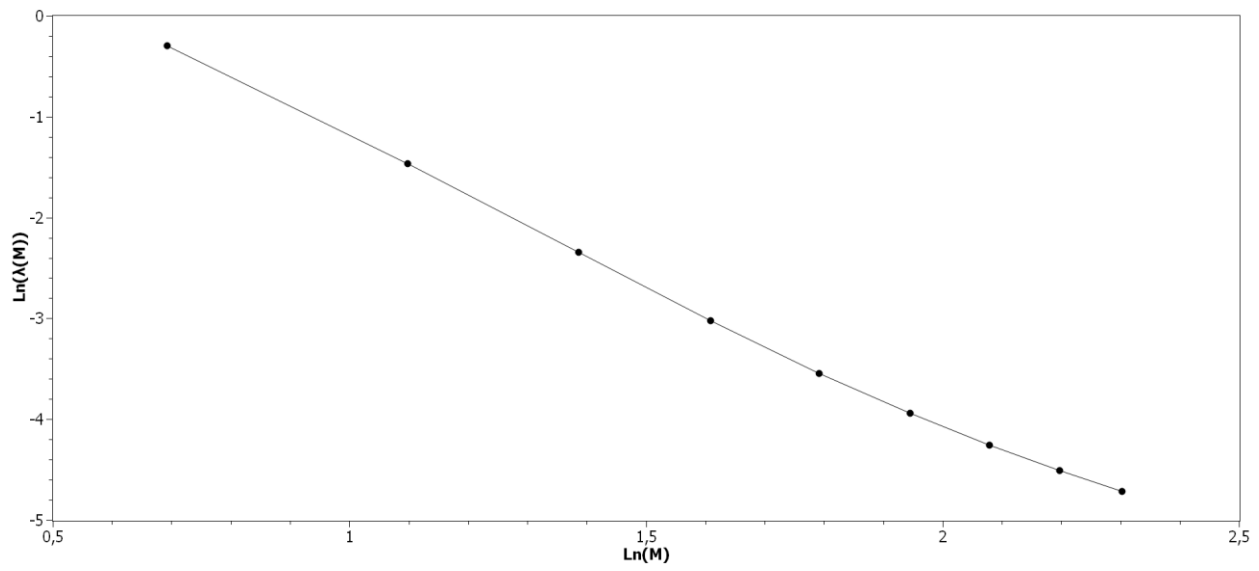


Рис.7 Зависимость $\ln(\lambda(M))$ от $\ln(M)$

Заключение

В данной работе были изучены применения тензорных сетей для исследования одномерных спиновых систем. Было показано как при помощи тензорных сетей можно эффективно находить основные состояния и смотреть устройство квантовой запутанности, по которым можно судить о поведении системы. Также были рассмотрены процессы термализации и найдены намеки на возможное наличие ранее неизвестных процессов транспорта на одномерных спиновых системах.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- [1] Orús, Román. “A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States.” *Annals of Physics* 349 (2014): 117–158. arXiv:1306.2164
- [2] Pirvu, Betal. “Matrix Product Operator Representations.” *New Journal of Physics* 12.2 (2010): 025012. arXiv:0804.3976
- [3] Kim, Hyungwon et al. “Slowest Local Operators in Quantum Spin Chains.” *Physical Review E* 92.1 (2015): n. pag. arXiv:1410.4186
- [4] Srednicki, Mark. “Chaos and Quantum Thermalization.” *Physical Review E* 50.2 (1994): 888–901. arXiv:cond-mat/9403051
- [5] Noah Linden, Sandu Popescu, Anthony J. Short, and Andreas Winter, “Quantum mechanical evolution towards thermal equilibrium,” *Phys. Rev. E* 79, 061103 (2009).
- [6] Deutsch, Joshua M. “Eigenstate Thermalization Hypothesis.” *Reports on Progress in Physics* 81.8 (2018): 082001. arXiv:1805.01616
- [7] Mori, Takashi et al. “Thermalization and Prethermalization in Isolated Quantum Systems: a Theoretical Overview.” *Journal of Physics B: Atomic, Molecular and Optical Physics* 51.11 (2018): 112001. arXiv:1712.08790
- [8] Marcos Rigol, Vanja Dunjko, Vladimir Yurovsky, and Maxim Olshanii, “Relaxation in a completely integrable many-body quantum system: An ab initio study of the dynamics of the highly excited states of 1d lattice hardcore bosons,” *Phys. Rev. Lett.* 98, 050405 (2007)
- [9] Marcos Rigol, Vanja Dunjko, and Maxim Olshanii, “Thermalization and its mechanism for generic isolated quantum systems,” *Nature* 452, 854–858 (2008).
- [10] Christian Gogolin, Markus P. Müller, and Jens Eisert, “Absence of thermalization in nonintegrable systems,” *Phys. Rev. Lett.* 106, 040401 (2011).
- [11] M. C. Bañuls, J. I. Cirac, and M. B. Hastings, “Strong and weak thermalization of infinite nonintegrable quantum systems,” *Phys. Rev. Lett.* 106, 050405 (2011).

Приложение

Ниже представлены программные коды при помощи которых были выполнены все вычисления, представленные в работе

Код для нахождения основного состояния и его энергии

```
import numpy as np
from numpy import linalg as LA
import matplotlib.pyplot as plt

from ncon import ncon
from doDMRG_MPO import doDMRG_MPO

chi = 16;
OPTS_numsweeps = 4 # number of DMRG sweeps
OPTS_dispon = 2 # level of output display
OPTS_updateon = True # level of output display
OPTS_maxit = 2 # iterations of Lanczos method
OPTS_krydim = 4 # dimension of Krylov subspace

chid = 2
for Nsites in range(2, 13):
    print("Nsites = ", Nsites)
    for j in range(10):
        h = j/10
        print("h = ", h)
        g = 1.05
        sX = np.array([[0, 1], [1, 0]])
        sY = np.array([[0, -1j], [1j, 0]])
        sZ = np.array([[1, 0], [0, -1]])
        sI = np.array([[1, 0], [0, 1]])
        M = np.zeros([4,4,chid,chid]);
        M[0,0,::] = sI; M[3,3,::] = sI
        M[0,1,::] = sZ; M[1,3,::] = -sZ
        M[0,2,::] = h*sZ; M[0,3,::] = g*sX
        M[2,2,::] = sI
        ML = np.array([1,0,0,0]).reshape(4,1,1) #left MPO boundary
        MR = np.array([0,0,1,1]).reshape(4,1,1) #right MPO boundary
        E = []
        ##### Initialize MPS tensors
        A = [0 for x in range(Nsites)]
        A[0] = np.random.rand(1,chid,min(chi,chid))
        for k in range(1,Nsites):
            A[k] = np.random.rand(A[k-1].shape[2],chid,min(min(chi,A[k-1].shape[2]*chid),chid**(Nsites-k-1)))

        ##### Do DMRG sweeps (2-site approach)
        En1, A, sWeight, B = doDMRG_MPO(A,ML,M,MR,chi, numsweeps = OPTS_numsweeps,
dispon = OPTS_dispon,
updateon = OPTS_updateon, maxit = OPTS_maxit,
krydim = OPTS_krydim)

        ##### Increase bond dim and reconverge
        chi = 32;
        En2, A, sWeight, B = doDMRG_MPO(A,ML,M,MR,chi, numsweeps = OPTS_numsweeps,
dispon = OPTS_dispon,
updateon = OPTS_updateon, maxit = OPTS_maxit,
krydim = OPTS_krydim)
```

Код для расчета энтропии запутанности

```
import numpy as np
from scipy import linalg as la
import matplotlib.pyplot as plt
from doDMRG_MPO import doDMRG_MPO
import tensornetwork as tn

def create_2_MPS(s):
    MPS1=[]
    MPS2=[]
    for i in range(N):
        MPS1.append(tn.Node(s[i]))
        MPS2.append(tn.conj(MPS1[i]))
    for i in range(N-1):
        MPS1[i][2] ^ MPS1[i+1][0]
        MPS2[i][2] ^ MPS2[i+1][0]
    return (MPS1, MPS2)

def getr(s, k):
    (MPS1, MPS2) = create_2_MPS(s)
    for i in range(N):
        MPS1[i][1] ^ MPS2[i][1]
    right = MPS1[N-1] @ MPS2[N-1]
    for i in range(N-2, k, -1):
        right = MPS1[i] @ right
        right = right @ MPS2[i]
    if (k != N-2):
        m = right.tensor[:,0,0,:]
    else:
        m = right.tensor[:,0,:,0]
    return m

def diag_Entropy(r):
    diag = np.diag(np.diag(r))
    lndiag = la.logm(diag)
    entropy = -np.trace(np.dot(diag, lndiag))
    return entropy

def ground_state():
    global chi
    global chid
    m = []
    for i in range(N):
        m.append(0)
    m[0] = np.random.rand(1, chid, min(chi, chid))
    chid = 2
    opX = np.array([[0, 1], [1, 0]])
    opZ = np.array([[1, 0], [0, -1]])
    Iden = np.array([[1, 0], [0, 1]])
    MPO = np.zeros([3, 3, chid, chid]);
    MPO[0,0,::] = Iden
    MPO[1,0,::] = opZ
    MPO[2,0,::] = h*opZ + g*opX
    MPO[2,1,::] = -opZ
```



```

MPO[2,2,::] = Iden
MPOleft = np.array([0,0,1]).reshape(3,1,1)
MPOright = np.array([1,0,0]).reshape(3,1,1)
for k in range(1,N):
    m[k] = np.random.rand(m[k-1].shape[2], chid, min(min(chi, m[k-1].shape[2]*chid), chid**(N-k-1)))
    (En1, m, sWeight, B) = doDMRG_MPO(m, MPOleft, MPO, MPOright, chi,
numsweps = 4, dispo = 0, updateon = True, maxit = 2, krydim = 4)
    chi = 32
    (En2, m, sWeight, B) = doDMRG_MPO(m, MPOleft, MPO, MPOright, chi,
numsweps = 4, dispo = 0, updateon = True, maxit = 2, krydim = 4)
    return m

def printloc():
    global h
    h = float(input('Введите h: '))
    m=ground_state()
    entropy = []
    print('h =', h)
    print('g =', g)
    print('Энтропия запутанности для разных позиций:')
    for i in range(N-1):
        r=getr(m,i)
        entropy.append(diag_Entropy(r))
        print('Позиция спина = ', i, ': ',entropy[i])
    l = np.arange(N-1)
    ff = plt.figure(figsize=(18, 10))
    plt.plot(l, entropy, 'bs', label="Entaglement entropy")
    plt.xlabel("Позиция спина", fontsize=20)
    plt.ylabel("Энтропия запутанности", fontsize=20)
    plt.axis('tight')
    plt.show()

chi = 16
chid = 2
N = 100
g = 0
printloc()

```