

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Епифанова Галина Сергеевна

Выпускная квалификационная работа – магистерская диссертация  
по направлению 11.04.04 «Электроника и наноэлектроника»

студента образовательной программы магистратуры  
«Прикладная физика»  
наименование образовательной программы

Научный руководитель  
к.ф.-м.н., доцент

Р.Ш. Ихсанов

Консультант  
член-корр. РАН, д.ф.-м.н., профессор

---

---

М. Ю. Каган

Москва 2018

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Московский институт электроники и математики им. А.Н. Тихонова**

**ЗАДАНИЕ  
на выполнение магистерской диссертации**

студенту группы МФ3161 Епифановой Галине Сергеевне

1. Тема работы

Численный расчет фазовой диаграммы двух связанных цепочек холодных атомов в рамках модели Хаббарда с притяжением

2. Цель работы

Разработка программного обеспечения и численный расчет фазовой диаграммы двух связанных цепочек холодных атомов в рамках модели Хаббарда с притяжением.

3. Формулировка задания

1) Разработка программы расчета энергетического спектра одномерной цепочки спинов методом прямой диагонализации.

2) Расчет фазовой диаграммы одной цепочки холодных атомов в рамках модели Хаббарда с притяжением для различных значений параметров модели.

3) Расчет фазовой диаграммы двух связанных цепочек холодных атомов.

Проект ВКР должен быть предоставлен студентом в срок до «25» декабря 2017 г.

Научный руководитель ВКР      «25» декабря 2017 г.      \_\_\_\_\_ Р.Ш. Ихсанов

Первый вариант ВКР предоставлен студентом в срок до «31» марта 2018 г.

Научный руководитель ВКР      «30» марта 2018 г.      \_\_\_\_\_ Р.Ш. Ихсанов

Итоговый вариант ВКР предоставлен студентом в срок до «30» апреля 2018 г.

Научный руководитель ВКР      «27» апреля 2018 г.      \_\_\_\_\_ Р.Ш. Ихсанов

Задание выдано студенту      «15» декабря 2017 г.      \_\_\_\_\_ Р.Ш. Ихсанов

Задание принято к исполнению студентом      «15» декабря 2017 г.      \_\_\_\_\_ Г.С. Епифанова

## Оглавление

Введение.....	5
1. Одиночный спин. Гильбертово пространство. ....	6
2. N спинов. Тензорное произведение Гильбертовых пространств.....	14
3. Модель Хаббарда. ....	16
4. Методы моделирования и программное обеспечение .....	17
5. Расчет спиновых цепочек.....	30
6. Методы расчета фазовой диаграммы связанных цепочек атомов .....	33
7. Результаты расчетов фазовых диаграмм .....	37
7.1. Одна цепочка атомов .....	37
7.3. Три взаимодействующие цепочки атомов.....	41
Заключение .....	42
Библиографический список.....	43
Приложение 1. ....	44

## **Аннотация**

В настоящее время в мире проводится большой объем теоретических и экспериментальных исследований в области физики сильно коррелированных систем. Интерес к таким системам особенно возрос в последнее время, когда с развитием техники низких температур открылись новые возможности и направления в этих исследованиях.

В данной работе рассматривались одномерные связанные цепочки холодных атомов, взаимодействующих посредством атомных спиновых магнитных моментов. На основе модели Хаббарда с притяжением в работе был проведен численный расчет фазовой диаграммы одной, двух и трех цепочек холодных атомов. Расчеты были проведены с использованием программного обеспечения собственной разработки в системе Matlab и с использованием программного пакета свободного распространения ALPS, управляемого командами на языке Python.

Фазовые диаграммы были рассчитаны для различных параметров модели и различного числа цепочек, был проверен ряд качественных и количественных теоретических результатов.

## **Abstract**

Currently, a large amount of theoretical and experimental research in the field of physics of strongly correlated systems is carried out in the world. Interest in such systems has increased particularly in recent years, when the development of low temperature technology has opened up new opportunities and trends in these studies.

In this paper, one-dimensional bound chains of cold atoms interacting by means of atomic spin magnetic moments were considered. On the basis of the Hubbard model with attraction, the numerical calculation of the phase diagram of one, two and three chains of cold atoms was carried out. The calculations were carried out using software of own development in Matlab system and using software package of free distribution of ALPS, managed by commands in Python.

Phase diagrams were calculated for different model parameters and different number of chains, a number of qualitative and quantitative theoretical results were tested.

## Введение

Несмотря на свою простоту, модель Хаббарда очень распространена и эффективно описывает взаимодействие между частицами, как фермионами, так и бозонами. Модель часто применяется для описания многих физических систем, вплоть до высокотемпературных сверхпроводников. В настоящей работе модель Хаббарда с притяжением реализована численно методом прямой диагонализации для расчета фазовых диаграмм одной, двух и трех цепочек холодных атомов. При этом, менялись значения параметров моделей.

Одной из задач ВКР стояла разработка программного обеспечения, позволяющего рассчитать энергетический спектр одномерной цепочки спинов методом прямой диагонализации.

В первой главе настоящей работы описывается алгебра гильбертового пространства для случая с многочастичной системой: вводится обозначение векторов, основных операторов, тензорного произведения гильбертовых пространств и особенностями этой системы. Во второй главе описана модель Хаббарда, которая используется в расчетах численный расчет фазовой диаграммы одной, двух и трех цепочек холодных атомов. В следующей главе описываются методы, использованные при моделировании в программе Matlab, а также говорится об особенностях использования пакета ALPS. В главах с четвертой по шестую содержится описание проведенных расчетов и результаты работы.

## 1. Одиночный спин. Гильбертово пространство.

Для описания произвольной физической системы требуется ввести понятие состояния. Если в классической теории состояние системы можно описать набором заданных координат и скоростей всех частей системы в определенный момент времени, то описание микросистем требует более детальных и глубоких параметров. Квантовомеханические системы описываются конечномерными комплексными гильбертовыми пространствами.

В квантовой механике векторное пространство состоит из элементов  $|A\rangle$ , которые называются кет-векторами (кетами), которые представляют собой векторы состояния системы. Такие вектора следует рассматривать в качестве векторов-столбцов. Для любого кет-вектора  $|A\rangle$  в дуальном пространстве существует бра-вектор, обозначаемый  $\langle A|$  – вектор-столбец.

Скалярное произведение векторов в квантовой механике называется *внутренним произведением* и обозначается  $\langle B|A\rangle$ . Результатом этой операции является комплексное число, которое линейно зависит от  $|A\rangle$  и антилинейно от  $|B\rangle$ . Выразим скалярное произведение через операции с компонентами:

$$\langle B|A\rangle = (\beta_1^* \ \beta_2^* \ \beta_3^* \ \beta_4^*) \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \beta_1^* \alpha_1 + \beta_2^* \alpha_2 + \beta_3^* \alpha_3 + \beta_4^* \alpha_4$$

Длина вектора состояния определяется равенством:  $\|a\| = \sqrt{\langle a|a\rangle}$ .

Само Гильбертово пространство можно определить следующими пунктами:

- а) векторное пространство комплексных чисел  $\mathbb{C}$ . Обозначение вектора (или луча) в  $\mathbb{H}$  принято в виде  $|a\rangle$ ;

б) пространство с определенным скалярным произведением  $\langle a|b\rangle$ , удовлетворяющим следующим свойствам:

– положительная определенность:  $\langle a|a\rangle > 0$

– линейность:  $\langle a|c_1 b + c_2 c\rangle = c_1 \langle a|b\rangle + c_2 \langle a|c\rangle$ , где  $c_1$  и  $c_2$  – комплексные числа

– эрмитовское сопряжение:  $\langle a|b\rangle = \langle b|a\rangle^*$

в) пространство, нормированное по норме  $\|a\| = \sqrt{\langle a|a\rangle}$

Действием линейного оператора  $\hat{F}$ :  $|y\rangle = \hat{F}|x\rangle$  можно назвать перевод состояния  $|x\rangle$  в состояние  $|y\rangle$ .

Для линейного оператора  $\hat{F}$  верно равенство:

$$\hat{F}(c_1|x_1\rangle + c_2|x_2\rangle) = c_1\hat{F}|x_1\rangle + c_2\hat{F}|x_2\rangle$$

Где  $c_1$  и  $c_2$  – произвольные комплексные числа.

Коммутатор операторов  $\hat{A}$  и  $\hat{B}$  обозначается символом  $[\hat{A}, \hat{B}]$  и определяется равенством:

$$[\hat{A}, \hat{B}] = \hat{A} \cdot \hat{B} - \hat{B} \cdot \hat{A}$$

Уравнение на собственные значения и собственные функции в квантовой теории выглядит:

$$\hat{F}|a\rangle = \alpha|a\rangle$$

Число  $\alpha$  – собственное значение оператора  $\hat{F}$ , а вектор  $|a\rangle$  – собственный вектор линейного оператора  $\hat{F}$ .

Условие полноты векторов состояний представляется:

$$\sum_n |n\rangle\langle n| = 1 \quad \text{или} \quad \int |n\rangle\langle n| dx = 1$$



В результате измерения состояние  $|\psi\rangle$  переходит (редуцируется) в состояние  $|\psi\rangle \Rightarrow |\Phi_n\rangle = |n\rangle$ , которое является собственным для оператора  $\hat{F}$ . Нормированное состояние после измерения  $|\Phi_n\rangle$  имеет вид:

$$|\Phi_n\rangle = \frac{1}{\sqrt{\langle\psi|P|\psi\rangle}} \cdot P_n|\psi\rangle$$

Введем понятие пси-функции  $\psi$ , которая описывает состояние системы. Физических свойств у нее нет, это комплексная величина, которую нельзя измерить. Квадрат модуля пси-функции  $|\psi|^2$  – это физическая величина, описывающая плотность вероятности нахождения частицы.

$|\psi|^2 dV = dP$  описывает вероятность нахождения частицы в элементе объема  $dV$ .

$$\psi = \psi_0 e^{ikx} \quad k = \frac{2\pi}{\lambda} \quad x - \text{координата}$$

Условие нормировки волновой функции:

$$\int_V |\psi|^2 dV = 1$$

Это условие говорит о том, что вероятность нахождения частицы в объеме  $dV$  равна единице.

Если  $f$  – физическая величина, то значения, которые она может принимать  $f_n$  – являются собственными значениями данной физической величины.

Собственные функции  $\psi_n$  – волновые функции, которые описывают такое состояние системы, находясь в котором физическая величина с некоторой вероятностью принимает значение  $f_n$ .

Принцип суперпозиции состояний:

$$\psi = \sum_n C_n \psi_n$$

Физическая величина  $f$  принимает значение  $f_n$  с вероятностью  $|C_n|^2$ .

Среднее значение физической величины в состоянии  $\psi$ :

$$\langle f \rangle = \sum_n |C_n|^2 f_n$$

$$\sum_n |C_n|^2 = 1$$

Как указывалось ранее, каждой физической величине в квантовой механике можно сопоставить некоторый оператор  $f \rightarrow \hat{f}$ . Среднее значение физической величины в состоянии  $\psi$  можно выразить с помощью оператора:

$$\langle f \rangle = \int_V \psi^* \hat{f} \psi dV$$

Условие линейности:

$$\hat{f}\psi = \sum_n C_n \hat{f}\psi_n$$

Рассмотрим основные операторы физических величин.

#### I. Оператор координат.

$$\hat{x} = x$$

Уравнение на собственные функции и собственные значения:

$$\hat{x}\psi = x\psi$$

#### II. Оператор проекции импульсов.

$$\hat{p}_x = -i\hbar \frac{\partial}{\partial x} \quad \hbar = \frac{h}{2\pi}$$

$$\hat{p}_x\psi = p_x\psi$$

$$\psi = \psi_0 e^{ik_x x}$$

### III. Оператор вектора импульса.

$$\hat{\vec{p}} = \hat{p}_x \vec{i} + \hat{p}_y \vec{j} + \hat{p}_z \vec{k} = -i\hbar \left( \frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} \right) = -i\hbar \vec{\nabla}$$

$\vec{\nabla}$  - оператор Набла.

### IV. Оператор кинетической энергии.

Кинетическая энергия определяется:

$$K = \frac{mv^2}{2} = \frac{p^2}{2m}$$

Оператор кинетической энергии выражается:

$$\hat{K} = \frac{\hat{p}^2}{2m}$$

$$\begin{aligned} \hat{p}^2 &= \hat{p} \cdot \hat{p} = (-i\hbar)^2 \left( \frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} \right) \left( \frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} \right) = \\ &= -\hbar^2 \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) = -\hbar^2 \Delta \end{aligned}$$

$\Delta$  - оператор Лапласа

Таким образом, оператор кинетической энергии:

$$\hat{K} = \frac{\hat{p}^2}{2m} = -\frac{\hbar^2}{2m} \Delta$$

### V. Оператор потенциальной энергии.

$$\hat{U} = U(x, y, z)$$

$U(x, y, z)$  – потенциальная функция.

### VI. Оператор полной энергии.

Поскольку полная энергия есть сумма кинетической и потенциальной:

$$\hat{E} = \hat{K} + \hat{U} = -\frac{\hbar^2}{2m} \Delta + U(x, y, z)$$

Уравнение на собственные функции и собственные значения для полной энергии:

$$\hat{E}\psi = E\psi$$

Таким образом, получилось уравнение Шредингера для стационарных состояний:

$$-\frac{\hbar^2}{2m}\Delta\psi + U(x, y, z)\psi = E\psi$$

Для свободных частиц  $U(x, y, z) = 0$ .

## VII. Оператор гамильтониана.

Зная функцию  $\Psi(x, y, z, t)$  в момент времени  $t_n$ , можно найти  $\Psi_m$  во момент времени  $t_m$ .

Оператор гамильтониана, поскольку он линейный и эрмитов, соответствует некоторой физической величине – полной энергии  $\hat{H} \rightarrow E$ .

$E_n$  – собственные значения  $\hat{H}$ .  $\Psi_n$  – собственные функции  $\hat{H}$ .

$$\hat{H}\Psi_n = E_n\Psi_n$$

Общее уравнение Шредингера:

$$i\hbar\frac{\partial\Psi}{\partial t} = \hat{H}\Psi, \quad \Psi(x, y, z, t)$$

Для стационарного уравнения Шредингера:

$$\hat{H}\phi = E\phi, \quad \phi = \psi e^{iEt/\hbar}$$

Оператор гамильтониана соответствующей энергии:

$$\hat{H} = -\frac{\hbar^2}{2m}\Delta + U(x, y, z)$$

## VIII. Оператор момента импульса.

В классической механике момент импульса материальной точки относительно начала отсчета определяется:

$$\vec{L} = [\vec{r}, \vec{p}] = m[\vec{r}, \vec{v}]$$

где  $\vec{r}$  – радиус-вектор частицы, а  $\vec{p}$  – ее импульс. Аналогично этому определению вводится оператор момента импульса в квантовой механике:

$$\hat{\vec{L}} = [\hat{\vec{r}}, \hat{\vec{p}}] = -i\hbar[\hat{\vec{r}}, \hat{\vec{\nabla}}]$$

Это векторный оператор, поэтому его можно представить как сумму операторов проекций:

$$\hat{\vec{L}} = \hat{L}_x \vec{e}_x + \hat{L}_y \vec{e}_y + \hat{L}_z \vec{e}_z$$

Где операторы проекций момента импульса определяются:

$$\hat{L}_x = -i\hbar \left( y \frac{\partial}{\partial z} - z \frac{\partial}{\partial y} \right)$$

$$\hat{L}_y = -i\hbar \left( z \frac{\partial}{\partial x} - x \frac{\partial}{\partial z} \right)$$

$$\hat{L}_z = -i\hbar \left( x \frac{\partial}{\partial y} - y \frac{\partial}{\partial x} \right)$$

Можно определить коммутаторы:

$$[\hat{L}_x, \hat{L}_y] = i\hbar \hat{L}_z$$

$$[\hat{L}_y, \hat{L}_z] = i\hbar \hat{L}_x$$

$$[\hat{L}_z, \hat{L}_x] = i\hbar \hat{L}_y$$

#### IX. Оператор квадрата момента импульса.

Из операторов  $\hat{L}_x$ ,  $\hat{L}_y$ ,  $\hat{L}_z$  составим оператор квадрата абсолютной величины вектора момента:

$$\hat{L}^2 = \hat{L}_x^2 + \hat{L}_y^2 + \hat{L}_z^2$$

Этот оператор коммутативен с каждым из операторов  $\hat{L}_x$ ,  $\hat{L}_y$ ,  $\hat{L}_z$ , как было отмечено ранее.

Вместо операторов  $\hat{L}_x$  и  $\hat{L}_y$  часто бывает удобнее пользоваться их комплексными комбинациями:

$$\hat{L}_+ = \hat{L}_x + i\hat{L}_y$$

$$\hat{L}_- = \hat{L}_x - i\hat{L}_y$$

Для этих комбинаций справедливы следующие соотношения:

$$\begin{aligned}
\hat{L}_+ + \hat{L}_- &= 2\hat{L}_z \\
\{\hat{L}_+, \hat{L}_-\} &= 2\hbar\hat{L}_z \\
\{\hat{L}_z, \hat{L}_+\} &= \hbar\hat{L}_+ \\
\{\hat{L}_z, \hat{L}_-\} &= -\hbar\hat{L}_- \\
\hat{L}^2 &= \hat{L}_+\hat{L}_- + \hat{L}_z^2 - \hbar\hat{L}_z \\
\hat{L}^2 &= \hat{L}_-\hat{L}_+ + \hat{L}_z^2 + \hbar\hat{L}_z
\end{aligned}$$

Теперь перечислим, как выглядят основные операторы с использованием вторичное квантование.

Оператор импульса:

$$\begin{aligned}
p &= \sum_{k', k''} \left\langle k'' \left| \frac{\hbar}{i} \frac{\partial}{\partial x} \right| k' \right\rangle a_{k''}^+ a_{k'} = \sum_{k', k''} \left[ \frac{\hbar}{i} \int e^{-ik'x} (ik'') e^{ik''x} dx \right] a_{k''}^+ a_{k'} = \\
&= \sum_{k', k''} \hbar k' \delta_{k' k''} a_{k''}^+ a_{k'} = \hbar \sum_k k a_k^+ a_k
\end{aligned}$$

Оператор кинетической энергии:

$$K = \frac{p^2}{2m} = \frac{\hbar^2}{2m} \sum_k k^2 a_k^+ a_k$$

Оператор числа частиц:

$$N = \sum_k a_k^+ a_k$$

Где с  $a_k^+$  - оператор рождения, а  $a_k$  - уничтожения.

Теперь перейдем к описанию систем с несколькими частицами.

## 2. N спинов. Тензорное произведение Гильбертовых пространств.

Пусть  $H_j, j = 1, 2$  – гильбертовы пространства двух квантовых систем со скалярными произведениями  $\langle \cdot | \cdot \rangle_j$ . Тогда система, состоящая из двух квантовых систем должна описываться тензорным произведением гильбертовых пространств, построение которого будет описано дальше.

Пусть элемент  $\psi \in H$  определяет антилинейную функцию  $\psi(\phi) = \langle \phi | \psi \rangle$  аргумента  $\phi \in H$ . Пусть имеются два элемента  $\psi_j \in H_j, j = 1, 2$ . Тогда би-антилинейную функцию аргументов  $\phi_j \in H_j, j = 1, 2$  обозначим  $\psi_1 \otimes \psi_2$  и определим соотношением:

$$(\psi_1 \otimes \psi_2)(\phi_1, \phi_2) = \langle \phi_1 | \psi_1 \rangle_1 \langle \phi_2 | \psi_2 \rangle_2$$

Рассмотрим векторное пространство  $L$  конечных линейных комбинаций таких функций  $\sum_j c_j \psi_1^j \otimes \psi_2^j$ . Введем скалярное произведение на пространстве  $L$ , полагая:

$$\langle \varphi_1 \otimes \varphi_2 | \psi_1 \otimes \psi_2 \rangle = \langle \varphi_1 | \psi_1 \rangle_1 \langle \varphi_2 | \psi_2 \rangle_2$$

Определенное выше пространство  $L$  с указанным скалярным произведением называется *тензорным произведением гильбертовых пространств  $H_1$  и  $H_2$ :  $H_1 \otimes H_2$* .

Если  $\{e_1^j\}, \{e_2^k\}$  – ортонормированные базисы в  $H_1$  и  $H_2$  соответственно, то ортонормированным базисом в  $H_1 \otimes H_2$  будет  $\{e_1^j \otimes e_2^k\}$ .  
 $\dim H_1 \otimes H_2 = \dim H_1 \times \dim H_2$ .

Следовательно, любой вектор  $\psi \in H_1 \otimes H_2$  однозначно выражается:

$$|\psi\rangle = \sum_{j,k} c_{jk} |e_1^j\rangle \otimes |e_2^k\rangle$$

Произведем суммирование по индексу  $j$ , также обозначим  $d_1 = \dim H_1$ ,  $d_2 = \dim H_2$ ,  $|\psi_k\rangle = \sum_{j=1}^{d_1} c_{jk} |e_1^j\rangle \in H_1$ . Тогда получим, что в общем случае гильбертово произведение  $H_1 \otimes H_2$  изоморфно прямой ортогональной сумме слагаемых  $H_1 \otimes \dots \otimes H_1$ :

$$|\psi\rangle = \sum_{k=1}^{d_2} |\psi_k\rangle \otimes |e_2^k\rangle$$

Теперь зададим тензорное произведение операторов  $X_j$  в пространствах  $H_j$ :

$$(X_1 \otimes X_2)(\psi_1 \otimes \psi_2) = X_1 \psi_1 \otimes X_2 \psi_2$$

Если зафиксируем базис в  $H_2$ , чтобы  $H_1 \otimes H_2$  представлялось как  $H_1 \otimes \dots \otimes H_1$ , то  $X_1 \otimes X_2$  можно представить в виде блочной матрицы:  $[X_1 x_2^{jk}]$ , где  $[x_2^{jk}]$  – матрица оператора  $X_2$  в базисе  $\{e_2^k\}$ . Тогда произвольный вектор  $X$  в  $H_1 \otimes H_2$  будет задаваться блочной матрицей  $[X_{jk}]$ .

Пусть некий линейный оператор  $\hat{M}$  действует в пространстве состояний составной системы. Тогда элементы матрицы оператора выражаются:

$$\langle a' b' | \hat{M} | a b \rangle = \hat{M}_{a' b' a b}$$



### 3. Модель Хаббарда.

Модель Хаббарда — приближение, используемое в физике твёрдого тела для описания перехода между проводящим и диэлектрическим состояниями, с учетом перескоков электронов на соседние атомы и кулоновского отталкивания на узле. В модели используется гамильтониан Хаббарда (1), состоящий из двух слагаемых [1]. Первый член гамильтониана (кинетическая энергия) описывает перескоки электронов на соседние узлы с амплитудой  $t_{ij}$ , второй член описывает кулоновское отталкивание электронов на узле с потенциалом  $U$  и учитывает, что одновременно на узле могут находиться частицы только с противоположным спином.

$$H = \sum_{i \neq j} \sum_{\sigma} t_{ij} a_{i\sigma}^+ a_{i\sigma} + \frac{1}{2} \sum_{i\sigma} U a_{i\sigma}^+ a_{i\sigma} a_{i,-\sigma}^+ a_{i,-\sigma} \quad (1)$$

В приближении ближайших соседей гамильтониан Хаббарда запишется в следующем виде:

$$H = U \sum_i n_{i\downarrow} n_{i\uparrow} + t \sum_{\langle i,j \rangle} \sum_{\sigma} a_{i\sigma}^+ a_{j\sigma} \quad (2)$$

Впервые модель была предложена (в 1963 году) для описания электронов в твёрдых телах, помогла объяснить фазовые переходы «металл — изолятор» в переходных металлах с узкими зонами. С тех пор модель особенно интересна при изучении высокотемпературной сверхпроводимости, наноструктур, квантовых точек и ям. Позднее стала использоваться при описании поведения ультра холодных атомов в оптических решётках.

## 4. Методы моделирования и программное обеспечение

Как было сказано ранее, в работе использовалась система Matlab и программный пакет ALPS. Matlab удобен в использовании, поскольку он содержит различные встроенные функции, позволяющие решать множество технических задач, и обладает достаточно простым интерфейсом. Система Matlab умеет проводить операции со скалярами, векторами, матрицами и единицами измерения, поэтому она часто используется для численных и символьных вычислений.

Пакет ALPS (Algorithms and libraries for Physics Simulation) – это программное обеспечение, позволяющее моделировать сильно коррелированные квантовомеханические системы с моделями решеток. Этот программный пакет свободно распространяется, управление происходит с помощью команд на языке Python. Пользователь составляет задание в формате данных XML, указывая в нем необходимые для расчета параметры. Программа позволяет проводить параллельное или последовательное численное моделирование с использованием методов Монте-Карло, точной диагонализации, метода ренормализационной группы матрицы плотности (DMRG).

### *Модели, определенные по умолчанию*

List of models defined in the default model library file:

model name	list of available parameters
spin	J Jz Jxy J0 Jz0 Jxy0 J1 Jz1 Jxy1 h Gamma D K K0 K1
boson Hubbard	mu t V U t0 t1 V0 V1
hardcore boson	same as above
fermion Hubbard	same as above
spinless fermions	mu t V t0 t1 V0 V1
Kondo lattice	mu t J
t-J	mu t J V J t0 t1 t2 V0 V1 V2 J0 J1 J2

## Структура файла модели

```
<MODEL>

#Определение базисных состояний на узле:
<SITEBASIS name="..."> #условное обозначение базиса
...#описание
</SITEBASIS>

#Определение базиса на решетке:
<BASIS name="..."> #условное обозначение базиса
... #описание
</BASIS>

#Определение Гамильтониана:
<HAMILTONIAN name="..."> #условное обозначение Гамильтониана
... #описание
</HAMILTONIAN>

</MODEL>
```

## Определение базисных состояний на узле

Атрибуты квантовых чисел:

- ✓ *min* и *max* для квантового числа  $N$ :  $N \in [\min, \min+1, \dots, \max]$
- ✓ *type*: bosonic (the default) or fermionic

### Пример 1:

```
<SITEBASIS name="hardcore boson">
  <QUANTUMNUMBER name="N" min="0" max="1"/> #описание квантового
числа N
</SITEBASIS>
```

### Пример 2:

Спин  $\frac{1}{2}$  на узле.

```
<SITEBASIS name="spin-1/2">
```

```

    <QUANTUMNUMBER name="S" min="1/2" max="1/2"/> #описание
квантового числа S
    <QUANTUMNUMBER name="Sz" min="-1/2" max="1/2"/> #описание
квантового числа  $S_z$ 
</SITEBASIS>

```

### Пример 3:

Фермион на узле.

```

<SITEBASIS name="fermion">
    <QUANTUMNUMBER name="Nup" min="0" max="1" type="fermionic"/>
#кв. число  $N_{up}$ 
    <QUANTUMNUMBER name="Ndown" min="0" max="1" type="fermionic"/>
#кв. число  $N_{down}$ 
</SITEBASIS>

```

Введение параметров (определяем имя и значение по умолчанию):

```

<SITEBASIS name="boson">
    <PARAMETER name="Nmax" default="infinity"/> #параметр  $N_{max}$ 
    <QUANTUMNUMBER name="N" min="0" max="Nmax"/>
</SITEBASIS>

```

```

<SITEBASIS name="spin">
    <PARAMETER name="local_spin" default="1/2"/>
    <QUANTUMNUMBER name="S" min="local_spin" max="local_spin"/>
    <QUANTUMNUMBER name="Sz" min="-S" max="S"/>
</SITEBASIS>

```

Определение через число частиц и спин:

```

<SITEBASIS name="t-J">
    <QUANTUMNUMBER name="N" min="0" max="1" type="fermionic"/>
    <QUANTUMNUMBER name="S" min="N/2" max="N/2"/>
    <QUANTUMNUMBER name="Sz" min="-S" max="S"/>
</SITEBASIS>

```

Определение через число частиц спинов вверх и вниз:

```

<SITEBASIS name="alternative t-J">
  <QUANTUMNUMBER name="Nup" min="0" max="1" type="fermionic"/>
  <QUANTUMNUMBER name="Ndown" min="0" max="1-Nup"
type="fermionic"/>
</SITEBASIS>

```

### ***Определение решетки***

Определение через ссылку на предопределенный базис узла:

```

<BASIS name="spin">
  <SITEBASIS ref="spin"/>
</BASIS>

```

Определение базиса узла вместе с решеткой:

```

<BASIS name="spin">
  <SITEBASIS name="spin-1">
    <QUANTUMNUMBER name="S" min="1" max="1"/>
    <QUANTUMNUMBER name="Sz" min="-1" max="1"/>
  </SITEBASIS>
</BASIS>

```

Решетка с базисом:

```

<BASIS name="Kondo lattice">
  <SITEBASIS type="0" ref="fermion"/>    #узел типа "fermion"
  <SITEBASIS type="1" ref="spin-1/2"/>   #узел типа "spin-1/2"
</BASIS>

```

```

<BASIS name="spin">

  <SITEBASIS type="0" ref="spin">
    <PARAMETER name="local_spin" value="local_S0"/>
    <PARAMETER name="local_S0" value="local_S"/>
    <PARAMETER name="local_S" value="1/2"/>
  </SITEBASIS>

  <SITEBASIS type="1" ref="spin">
    <PARAMETER name="local_spin" value="local_S1"/>
    <PARAMETER name="local_S1" value="local_S"/>
    <PARAMETER name="local_S" value="1/2"/>
  </SITEBASIS>

```

```
</SITEBASIS>

</BASIS>
```

### ***Операторы на узлах решетки***

Оператор определяется через: имя, матричный элемент и изменение квантового числа.

#### ***Simple site operators***

```
<SITEBASIS name="spin">
  <PARAMETER name="local_spin" default="1/2"/>
  <QUANTUMNUMBER name="S" min="local_spin" max="local_spin"/>
  <QUANTUMNUMBER name="Sz" min="-S" max="S"/>

  <OPERATOR name="Splus" matrixelement="sqrt(S*(S+1)-Sz*(Sz+1))">
    <CHANGE quantumnumber="Sz" change="1"/>
  </OPERATOR>

  <OPERATOR name="Sminus" matrixelement="sqrt(S*(S+1)-Sz*(Sz-1))">
    <CHANGE quantumnumber="Sz" change="-1"/>
  </OPERATOR>

  <OPERATOR name="Sz" matrixelement="Sz"/>
</SITEBASIS>
```

### ***Наш случай***

Параметры Гамильтониана:  $\{\mu, t, t_0, t_1, U, V, V_0, V_1\}$

#### ***1. Модель "fermion Hubbard":***

```

<HAMILTONIAN name="fermion Hubbard">
  <PARAMETER name="mu" default="0"/>
  <PARAMETER name="t" default="1"/>
  <PARAMETER name="V" default="0"/>
  <PARAMETER name="t'" default="0"/>
  <PARAMETER name="V'" default="0"/>
  <PARAMETER name="U" default="0"/>
  <PARAMETER name="t0" default="t"/>
  <PARAMETER name="t1" default="t'"/>
  <PARAMETER name="V0" default="V"/>
  <PARAMETER name="V1" default="V'"/>
  <BASIS ref="fermion"/>
  <SITETERM site="i">
    <PARAMETER name="mu#" default="mu"/>
    <PARAMETER name="U#" default="U"/>
    -mu#*n(i)+U#*n_up(i)*n_down(i)
  </SITETERM>
  <BONDTERM source="i" target="j">
    <PARAMETER name="t#" default="0"/>
    <PARAMETER name="V#" default="0"/>
    -t#*fermion_hop(i,j) + V#*n(i)*n(j)
  </BONDTERM>
</HAMILTONIAN>

```

### Бесконечные двумерные решетки

*Пример 1. Два базисных вектора с координатами (1;0) и (0.5;1).*

```

<LATTICE name="2D" dimension="2">
  <BASIS>
    <VECTOR> 1 0 </VECTOR>
    <VECTOR> 0.5 1 </VECTOR>
  </BASIS>
</LATTICE>

```

**Пример 2.** Два базисных вектора с координатами  $(a;0)$  и  $(b \sin \varphi; b \cos \varphi)$ .

```
<LATTICE name="2D" dimension="2">
```

*#Определение параметров:*

```
<PARAMETER name="a" default="1"/>
```

```
<PARAMETER name="b" default="1"/>
```

```
<PARAMETER name="phi" default="Pi/2"/>
```

*#Определение элементарной ячейки:*

```
<BASIS>
```

```
<VECTOR> a 0 </VECTOR>
```

```
<VECTOR> b*sin(phi) b*cos(phi) </VECTOR>
```

```
</BASIS>
```

```
</LATTICE>
```

## Конечные двумерные решетки

**Пример 1.** Решетка 5x3.

```
<FINITELATTICE name="5x3">
```

```
<LATTICE name="2D" dimension="2"/>
```

```
<EXTENT dimension="1" size="5"/> #5 ячеек по x
```

```
<EXTENT dimension="2" size="3"/> #3 ячеек по y
```

```
</FINITELATTICE>
```





**Пример 2. Решетка 4x4.**

```
<FINITELATTICE>  
  <LATTICE dimension="3"/>  
  <EXTENT size="4"/>  
</FINITELATTICE>
```

**Пример 3. Полубесконечная решетка  $\infty \times 2$ .**

```
<FINITELATTICE name="strip">  
  <LATTICE name="2D" dimension="2"/>  
  <EXTENT dimension="2" size="2"/>  
</FINITELATTICE>
```



**Пример 4. Конечная решетка  $L \times W$ .**

```
<FINITELATTICE>  
  <LATTICE name="2D" dimension="2"/>  
  <PARAMETER name="L" />  
  <PARAMETER name="W" default="2" /> #значение по умолчанию  
  <EXTENT dimension="1" size="L" />  
  <EXTENT dimension="2" size="W" />  
</FINITELATTICE>
```

**Пример 5. Конечная решетка  $L \times W \times H$ .**

```
<FINITELATTICE>
```

```

<LATTICE name="3D" dimension="3"/>
<PARAMETER name="L" />
<PARAMETER name="W" default="L" /> #значение по умолчанию
<PARAMETER name="H" default="W" /> #значение по умолчанию
<EXTENT dimension="1" size="L" />
<EXTENT dimension="2" size="W" />
<EXTENT dimension="3" size="H" />
</FINITELATTICE>

```

## Граничные условия

**Пример 1.** Периодическая в двух направлениях решетка  $L \times L$ .

```

<FINITELATTICE>
  <LATTICE name="2D" dimension="2"/>
  <EXTENT size="L" />
  <BOUNDARY type="periodic" />
</FINITELATTICE>

```

**Пример 2.** Периодическая в одном направлении решетка  $L \times W$ .

```

<FINITELATTICE name="strip">
  <LATTICE name="strip" dimension="2"/>
  <PARAMETER name="W" default="2" />
  <EXTENT dimension=1 size="L" />
  <EXTENT dimension=2 size="W" />
  <BOUNDARY dimension="1" type="periodic" /> #в x-направлении
  периодическая

```

```
<BOUNDARY dimension="2" type="open" /> #в y-направлении открытая  
</FINITELATTICE>
```

### Определение решеток наследыванием

**Пример 1.** “Полное” определение решетки  $L \times W$ , периодической в обоих направлениях.

```
<FINITELATTICE name = "finite tilted 2D">
```

*#Определение элементарной ячейки:*

```
<LATTICE name="tilted 2D" dimension="2">
```

```
<BASIS>
```

```
<VECTOR> 1 0 </VECTOR>
```

```
<VECTOR> 0.5 1 </VECTOR>
```

```
</BASIS>
```

```
</LATTICE>
```

*#конец определения элементарной ячейки*

```
<PARAMETER name="BC" default="periodic" />
```

```
<EXTENT dimension="1" size="L" />
```

```
<EXTENT dimension="2" size="W" />
```

```
<BOUNDARY type="BC" />
```

```
</FINITELATTICE>
```

**Пример 2.** Определение наследыванием решетки  $L \times W$ , периодической в обоих направлениях.

*#Определение элементарной ячейки:*

```
<LATTICE name="tilted 2D" dimension="2"
```

```

<BASIS>

  <VECTOR> 1 0 </VECTOR>

  <VECTOR> 0.5, 1 </VECTOR>

</BASIS>

</LATTICE>

```

*#Определение решетки:*

```

<FINITELATTICE name = "finite tilted 2D">

  <LATTICE ref="tilted 2D"> #ссылка на ранее определенную
элементарную ячейку

  <PARAMETER name="BC" default="periodic" />

  <EXTENT dimension="1" size="L" />

  <EXTENT dimension="2" size="W" />

  <BOUNDARY type="BC" />

</FINITELATTICE>

```

### Определенные в ALPS решетки

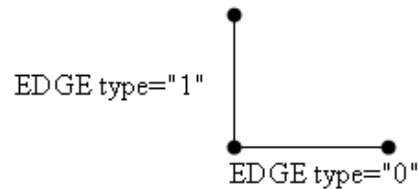
	Parameter	Value	Remark
1	LATTICE	chain lattice	periodic, homogeneous
	L		length
2	LATTICE	open chain lattice	open, homogeneous
	L		length
3	LATTICE	square lattice	periodic, homogeneous
	L		length
	W	default: L	width
4	LATTICE	open square lattice	open, homogeneous
	L		length
	W	default: L	width
5	LATTICE	Kagome lattice	periodic, homogeneous
	L		length
	W	default: L	width

6	LATTICE	honeycomb lattice	periodic, homogeneous
	L		length
	W	default: L	width
7	LATTICE	inhomogeneous square lattice	open, inhomogeneous
	L		length
	W	default: L	width
8	LATTICE	simple cubic lattice	periodic, homogeneous
	L		length
	W	default: L	width
	H	default: W	height
9	LATTICE	inhomogeneous simple cubic lattice	open, inhomogeneous
	L		length
	W	default: L	width
	H	default: W	height

## Наш случай:

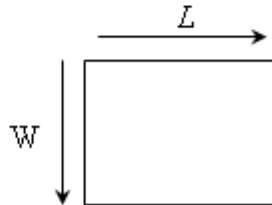
*Элементарная ячейка:*

```
<UNITCELL name="anisotropic2d" dimension="2">  
  <VERTEX/>  
  <EDGE type="0"><SOURCE vertex="1" offset="0 0"/><TARGET vertex="1" offset="1 0"/></EDGE>  
  <EDGE type="1"><SOURCE vertex="1" offset="0 0"/><TARGET vertex="1" offset="0 1"/></EDGE>  
</UNITCELL>
```



*Решетка:*

```
<LATTICEGRAPH name = "open ladder" vt_name="OpenLadderLattice">  
  <FINITELATTICE>  
    <LATTICE ref="square lattice"/>  
    <PARAMETER name="W" default="2"/>  
    <EXTENT dimension="1" size="L"/>  
    <EXTENT dimension="2" size="W"/>  
    <BOUNDARY dimension="1" type="open"/>  
    <BOUNDARY dimension="2" type="open"/>  
  </FINITELATTICE>  
  <UNITCELL ref="anisotropic2d"/>  
</LATTICEGRAPH>
```



## 5. Расчет спиновых цепочек

С помощью описанных в разделе 3 методов проведены следующие расчеты.

В приближении ближайших соседей гамильтониан Хаббарда записывается:

$$H = U \sum_i n_{i\downarrow} n_{i\uparrow} + t \sum_{\langle i,j \rangle} \sum_{\sigma} a_{i\sigma}^+ a_{j\sigma} \quad (2)$$

В программе ALPS принято другое написание (другой знак для  $t$ ):

$$H = U \sum_i n_{i\downarrow} n_{i\uparrow} - t \sum_{\langle i,j \rangle} \sum_{\sigma} a_{i\sigma}^+ a_{j\sigma} \quad (3)$$

**Тест 1.** Одна частица на цепочке из четырех узлов. Произведено сравнение расчетов матрицы гамильтониана и ее спектра в двух программах: на ALPS и Matlab. В программе на Matlab был использован метод точной диагонализации гамильтониана, а на ALPS – метод DMRG,  $U = 0$ ,  $t = 1$ .

Фрагмент кода на ALPS:

```
[[x=[0 1 2 3]
  y=[-1.61803399 -0.61803399 0.61803399 1.61803399]
  props={'observable': 'Energy', 'NUMBER_EIGENVALUES': 1.0,
```

Ниже показаны результаты расчетов программы на Matlab (матрица гамильтониана и ее спектр):

```

m_Ham =

    0     1     0     0
    1     0     1     0
    0     1     0     1
    0     0     1     0

ans =

-1.618033988749895e+000
-6.180339887498949e-001
 6.180339887498949e-001
 1.618033988749895e+000

```

Как видно, совпадение в расчетах происходит до девятой значащей цифры.

**Тест 2.** Одна частица в периодической цепочке из  $N$  узлов,  $U = 0$ . Спектр энергии имеет вид

$$E_k = -2t \cos\left(\frac{2\pi}{N}k\right), \quad k = \overline{1, N}, \quad (4)$$

где  $t > 0$  для гамильтониана вида (2).

Программа, написанная в системе Matlab, дает весь спектр вида (3), ALPS дает энергию основного состояния. Все сходится (в случае ALPS при больших  $N$ , поскольку в нем считаются только открытые, а не периодические цепочки).

**Тест 3.** Запускаем расчет энергии основного состояния двух противоположно направленных спинов на цепочке длиной  $L$  ( $L$  - число узлов). Считаем энергию связи димера:  $E_b = 2E_1 - E_2$ , где  $E_1$  - энергия (основного состояния) одного спина,  $E_2$  - энергия пары. Действительно, энергия пары оказывается меньше, чем удвоенная энергия одного, т.е. димер устойчив. Считаем энергию связи димера по формуле (используется гамильтониана вида (2), т.е.  $t > 0$ ):

$$E_b = -2(1+t) + \sqrt{U^2 + 4(1+t)^2}, \quad (5)$$



Также считаем погрешность приближения этого значения численным расчетом (расчеты проведены на ALPS). Получаем нижеприведенный график (рис.1), из которого видно, что погрешность убывает по закону  $\propto 1/L^2$ .

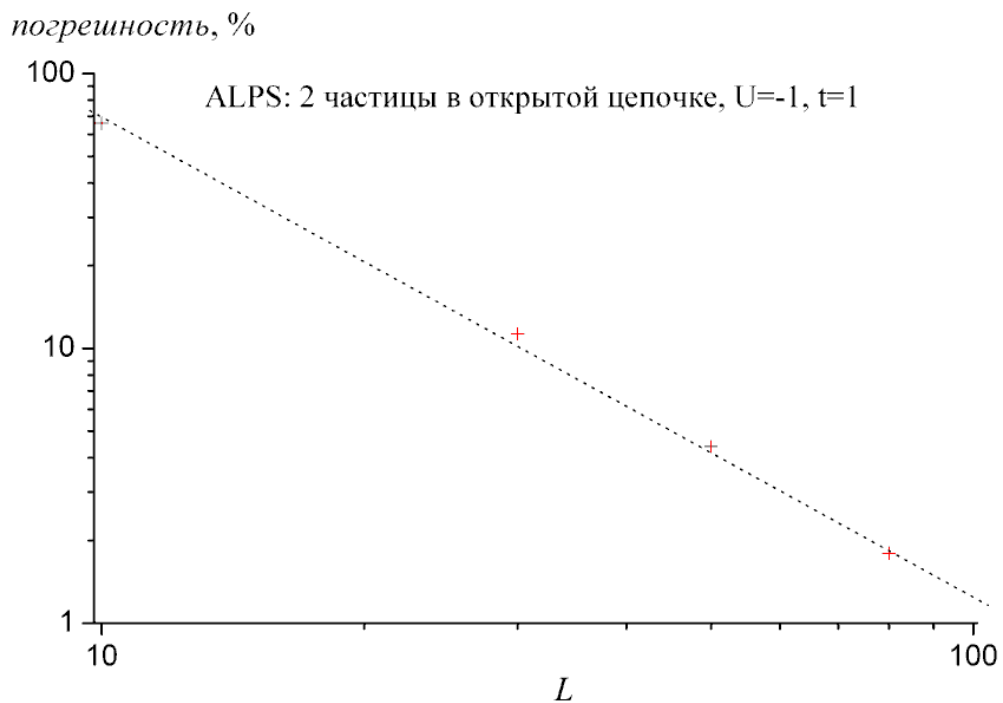


Рис. 1. Погрешность приближения численного расчета энергии связи димера

## 6. Методы расчета фазовой диаграммы связанных цепочек атомов

**Обозначения:**

$$N_{\Sigma} = N_{up} + N_{down}$$

$$N_{\Delta} = N_{up} - N_{down}$$

Химпотенциал:  $\mu = \left( \frac{\partial E}{\partial N_{\Sigma}} \right)_{N_{\Delta}}$

Магнитное поле:  $h = \left( \frac{\partial E}{\partial N_{\Delta}} \right)_{N_{\Sigma}}$

Большая каноническая энергия:

$$\Omega(N_{up}, N_{down}) = E(N_{up}, N_{down}) - \mu(N_{up}, N_{down})N_{\Sigma} - h(N_{up}, N_{down})N_{\Delta}$$

$L$  - число узлов цепочки

**Фазовые переходы:**

1. Линия “поляризованный газ - неполяризованный газ” (из фазы  $P=0$  в фазу  $P>0$ ).

Разностные формулы для производных ( $N_{up} = N_{down}$ ):

$$\mu = \left( \frac{\partial E}{\partial N_{\Sigma}} \right)_{N_{\Delta}} = \frac{E(N_{up} + 1, N_{down} + 1) - E(N_{up}, N_{down})}{(N_{\Sigma} = N_{up} + N_{down} + 2) - (N_{\Sigma} = N_{up} + N_{down}) = 2}$$

$$h = \left( \frac{\partial E}{\partial N_{\Delta}} \right)_{N_{\Sigma}} = \frac{E(N_{up} + 1, N_{down} - 1) - E(N_{up}, N_{down})}{(N_{\Delta} = 2) - (N_{\Delta} = 0) = 2}$$

2. Линия “вакуум - неполяризованный газ”.

Стартуем из точки  $(h, \mu) = (0, -\infty)$ , увеличивая  $\mu$ , до точки появления первой пары. При появлении пары большая каноническая энергия системы (одной пары) равна энергии вакуума:

$$E(1,1) - \mu(1,1)N_{\Sigma} - h(1,1)N_{\Delta} = 0, \text{ где } N_{\Sigma} = 2, N_{\Delta} = 0.$$

Отсюда, уравнение границы - горизонталь  $\mu = \frac{E(1,1)}{2}$ .

3. Линия “вакуум – полностью поляризованный газ”.

Пусть полностью поляризованный газ состоит из одной частицы:

$$E(1,0) - \mu(1,0)N_{\Sigma} - h(1,0)N_{\Delta} = 0,$$

тогда  $\mu = -h + E(1,0)$  - уравнение границы.

$$E(1,0) = -mt, \text{ где } m = \begin{cases} 2, & \text{for 1 leg} \\ 3, & \text{for 2 legs} \\ \text{not so simple} \end{cases}$$

4. Линия “полностью поляризованный газ – частично поляризованный газ”.

Полностью поляризованный газ:  $N_{up} > 0, N_{down} = 0$ , частично поляризованный газ:  $N_{up} > 0, N_{down} = 1$ . Тогда:

$$\mu = \left( \frac{\partial E}{\partial N_{\Sigma}} \right)_{N_{\Delta}} = \frac{E(N_{up} + 1, 1) - E(N_{up}, 1)}{\Delta N_{\Sigma}} = \frac{E(N_{up} + 1, 1) - E(N_{up}, 0)}{2}.$$

$$h = \left( \frac{\partial E}{\partial N_{\Delta}} \right)_{N_{\Sigma}} = \frac{E(N_{up} - 1, 1) - E(N_{up}, 1)}{\Delta N_{\Delta}} = \frac{E(N_{up} - 1, 1) - E(N_{up}, 0)}{-2}$$

5. Линия “полностью поляризованный газ 1 – полностью поляризованный газ 2”.

Фаза  $FP_1$  (полностью поляризованный газ 1):  $0 < N_{up} < L, N_{down} = 0$

Фаза  $FP_2$  (полностью поляризованный газ 2):  $N_{up} = L, N_{down} = 0$

Потенциалы  $\Omega$  фаз:

$$\Omega_{FP-1}(N_{up}) = E(N_{up}, 0) - \mu(N_{up}, 0)N_{up} - h(N_{up}, 0)N_{up},$$

$$\Omega_{FP-2} = E(L, 0) - \mu(L, 0)L - h(L, 0)L.$$

Приравниваем потенциалы фаз на границе раздела с учетом того, что для фазы  $FP_1$  на границе  $N_{up} = L-1$ , а так же  $\mu$  и  $h$  для обеих фаз одинаковы:

$$E(L-1,0) - (L-1)\mu - (L-1)h = E(L,0) - L\mu - Lh \Rightarrow \mu + h = E(L,0) - E(L-1,0) .$$

С учетом того, что  $E(L,0) - E(L-1,0) = -E(1,0)$ , получим уравнение фазовой границы:

$$E(L-1,0) - (L-1)\mu - (L-1)h = E(L,0) - L\mu - Lh \Rightarrow \mu = -h - E(1,0) .$$

Если сравнить с уравнением линии 3 ( $\mu = -h + E(1,0)$ ), то видно, что они параллельны.

### **Расчет второй производной энергии основного состояния по полному числу частиц**

#### **Обозначения:**

$L$  - число узлов цепочки;

$N_+$  - число частиц со спинами вверх,  $N_-$  - число частиц со спинами вниз;

$$N_\Sigma = N_+ + N_-, \quad N_\Delta = N_+ - N_-;$$

Преобразования между переменными  $\{N_+, N_-\}$ :

$$\begin{cases} N_\Sigma = N_+ + N_- \\ N_\Delta = N_+ - N_- \end{cases},$$

Преобразования между переменными  $\{N_\Sigma, N_\Delta\}$ :

$$\begin{cases} N_+ = \frac{N_\Sigma + N_\Delta}{2} \\ N_- = \frac{N_\Sigma - N_\Delta}{2} \end{cases}$$

Химпотенциал:

$$\mu = \left( \frac{\partial E}{\partial N_\Sigma} \right)_{N_\Delta}$$

Магнитное поле:

$$h = \left( \frac{\partial E}{\partial N_{\Delta}} \right)_{N_{\Sigma}}$$

Разностные формулы для производных:

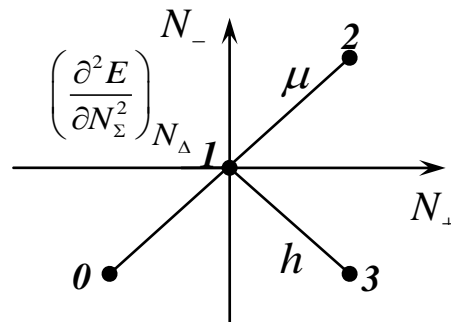
$$\mu = \left( \frac{\partial E}{\partial N_{\Sigma}} \right)_{N_{\Delta}} = \frac{E(N_{\Sigma} + 2, N_{\Delta}) - E(N_{\Sigma}, N_{\Delta})}{2} = \frac{E(N_{+} + 1, N_{-} + 1) - E(N_{+}, N_{-})}{2}$$

$$h = \left( \frac{\partial E}{\partial N_{\Delta}} \right)_{N_{\Sigma}} = \frac{E(N_{\Sigma}, N_{\Delta} + 2) - E(N_{\Sigma}, N_{\Delta})}{2} = \frac{E(N_{+} + 1, N_{-} - 1) - E(N_{+}, N_{-})}{2}$$

$$\begin{aligned} \left( \frac{\partial^2 E}{\partial N_{\Sigma}^2} \right)_{N_{\Delta}} &= \frac{E(N_{\Sigma} - 2, N_{\Delta}) - 2E(N_{\Sigma}, N_{\Delta}) + E(N_{\Sigma} + 2, N_{\Delta})}{2} = \\ &= \frac{E(N_{+} - 1, N_{-} - 1) - 2E(N_{+}, N_{-}) + E(N_{+} + 1, N_{-} + 1)}{2} \end{aligned}$$

Шаблон для расчета величин  $\mu$  (точки 1 и 2),  $h$  (точки 1 и 3) и  $\left( \frac{\partial^2 E}{\partial N_{\Sigma}^2} \right)_{N_{\Delta}}$

(точки 0, 1 и 2):



## 7. Результаты расчетов фазовых диаграмм

Ниже приведены полученные результаты расчетов фазовых диаграмм.

### 7.1. Одна цепочка атомов

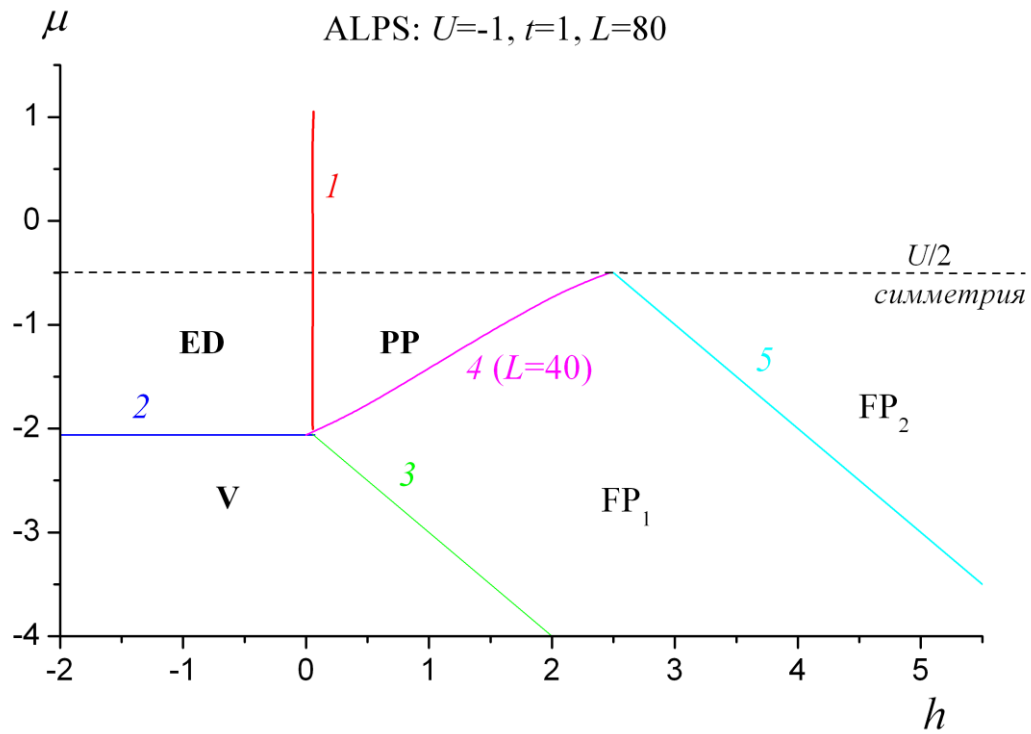


Рис. 2. Фазовая диаграмма одномерной цепочки при  $U = -1$

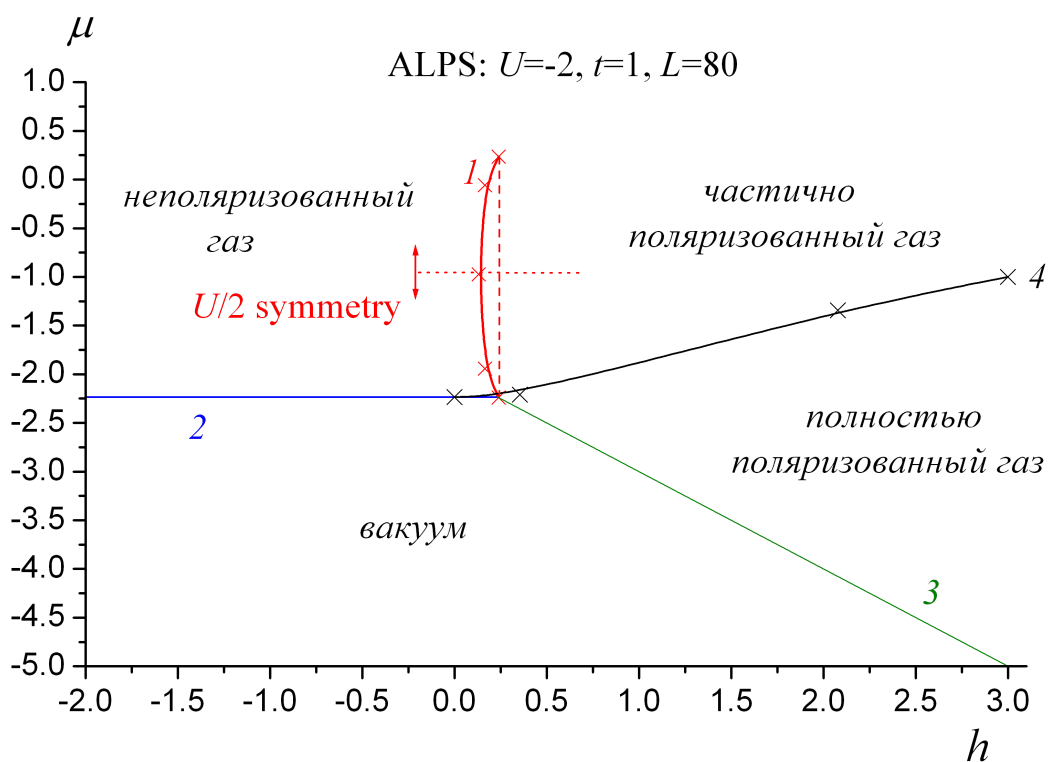


Рис. 3. Фазовая диаграмма одномерной цепочки при  $U = -2$

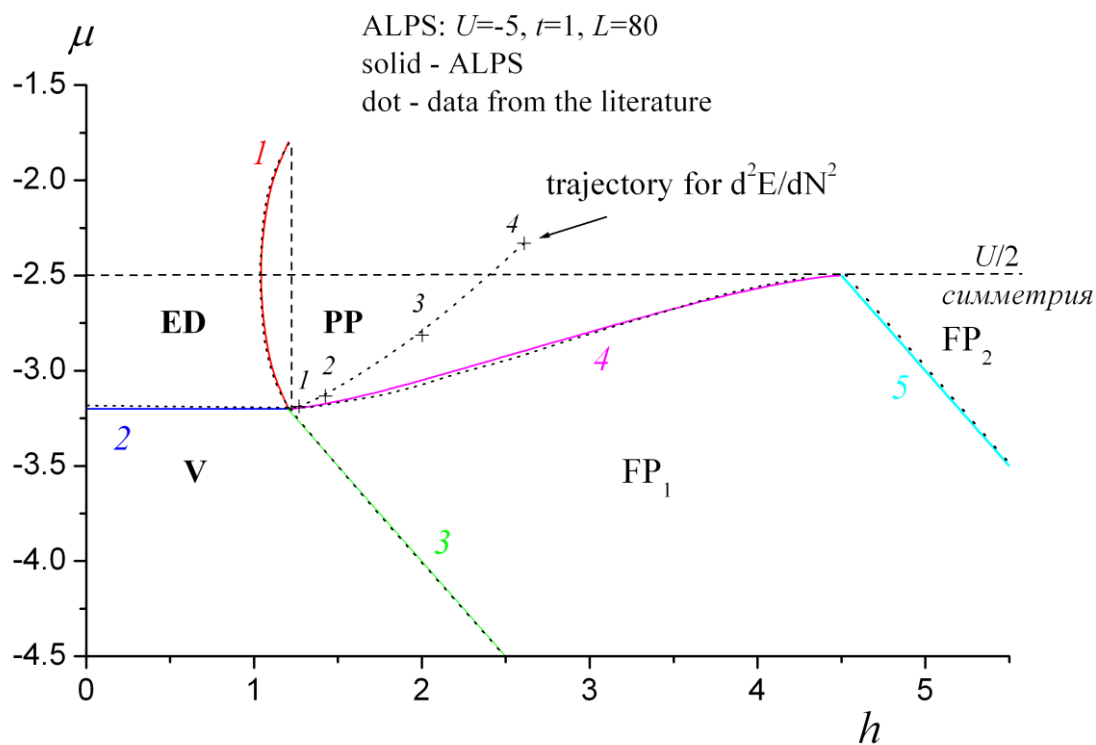


Рис. 4. Фазовая диаграмма одномерной цепочки при  $U = -5$

Результаты расчетов  $\left(\frac{\partial^2 E}{\partial N_{\Sigma}^2}\right)_{N_{\Delta}}$  (см. рис 4):

№ точки	$N_+$	$N_-$	$h$	$\mu$	$\left(\frac{\partial^2 E}{\partial N_{\Sigma}^2}\right)_{N_{\Delta}}$
1	10	5	1.27	-3.19	0.00215
2	20	10	1.43	-3.13	0.0052
3	40	20	2.00	-2.81	0.0140
4	60	30	2.61	-2.33	0.0145

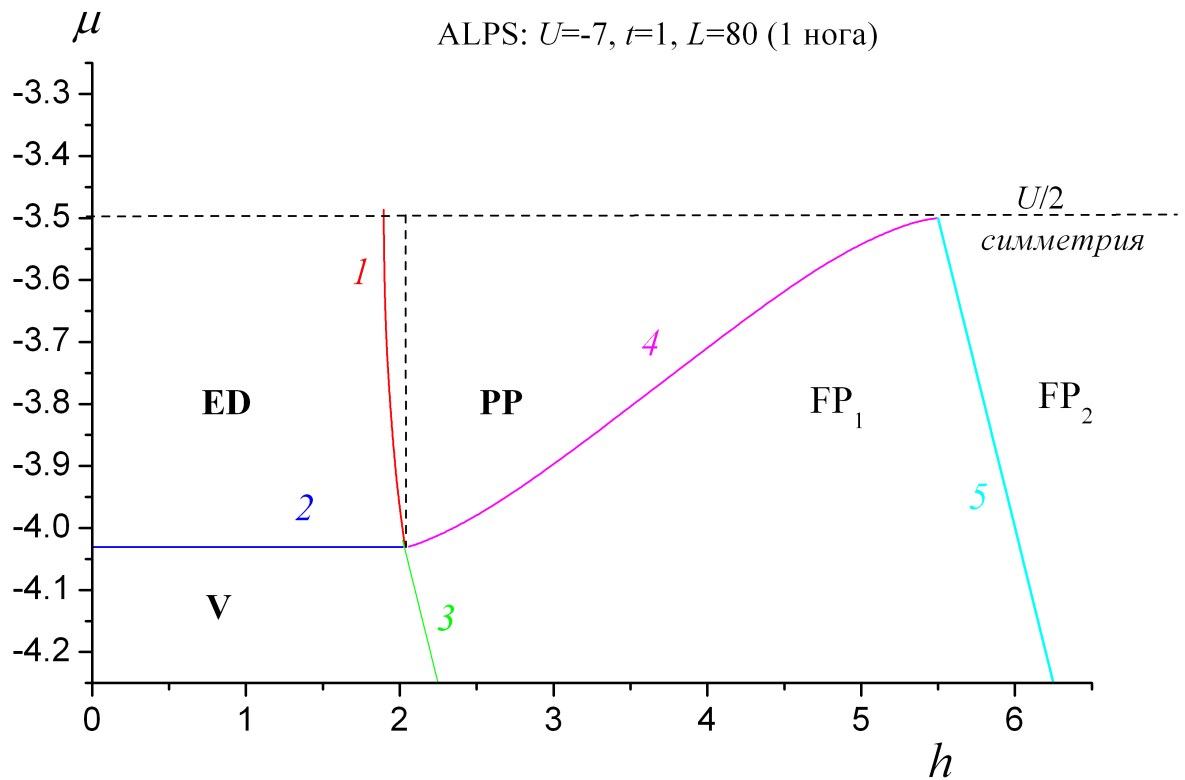


Рис. 5. Фазовая диаграмма одномерной цепочки при  $U = -7$



## 7.2. Две взаимодействующие цепочки атомов

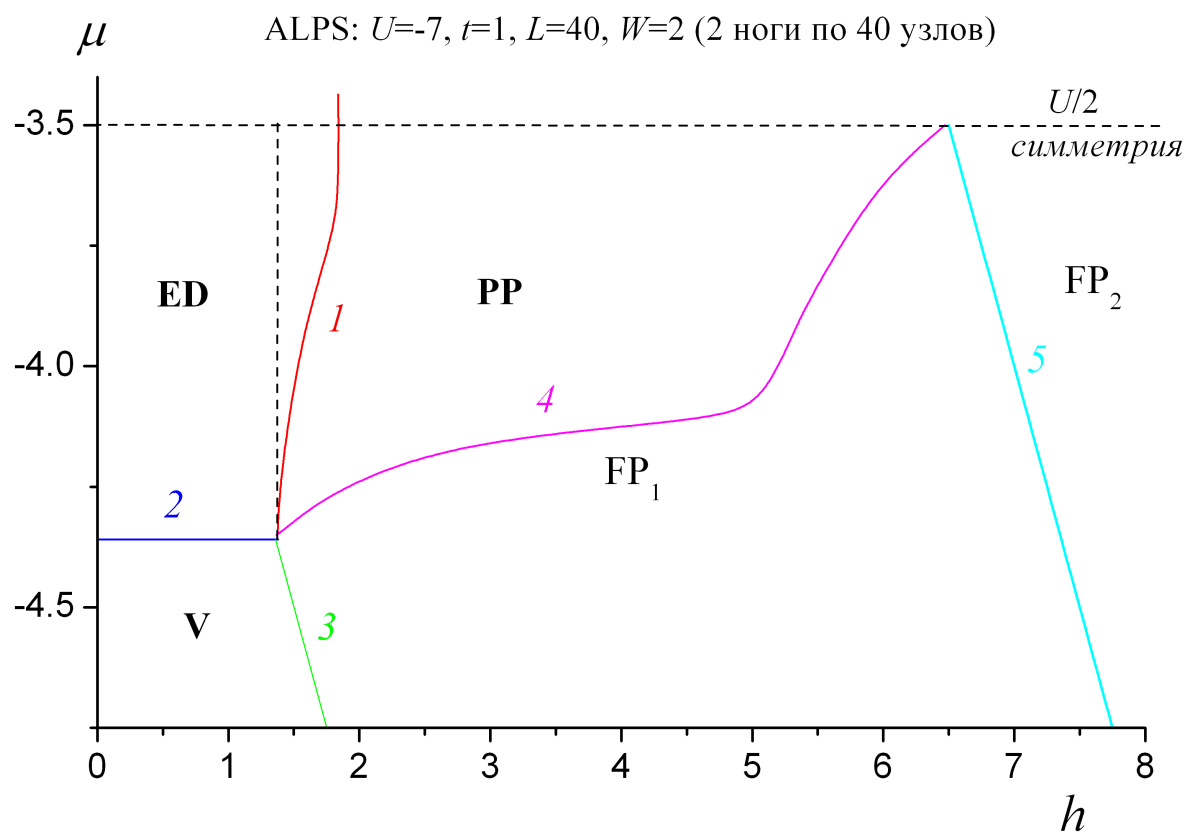


Рис. 6. Фазовая диаграмма двух связанных цепочек при  $U = -7$

### 7.3. Три взаимодействующие цепочки атомов

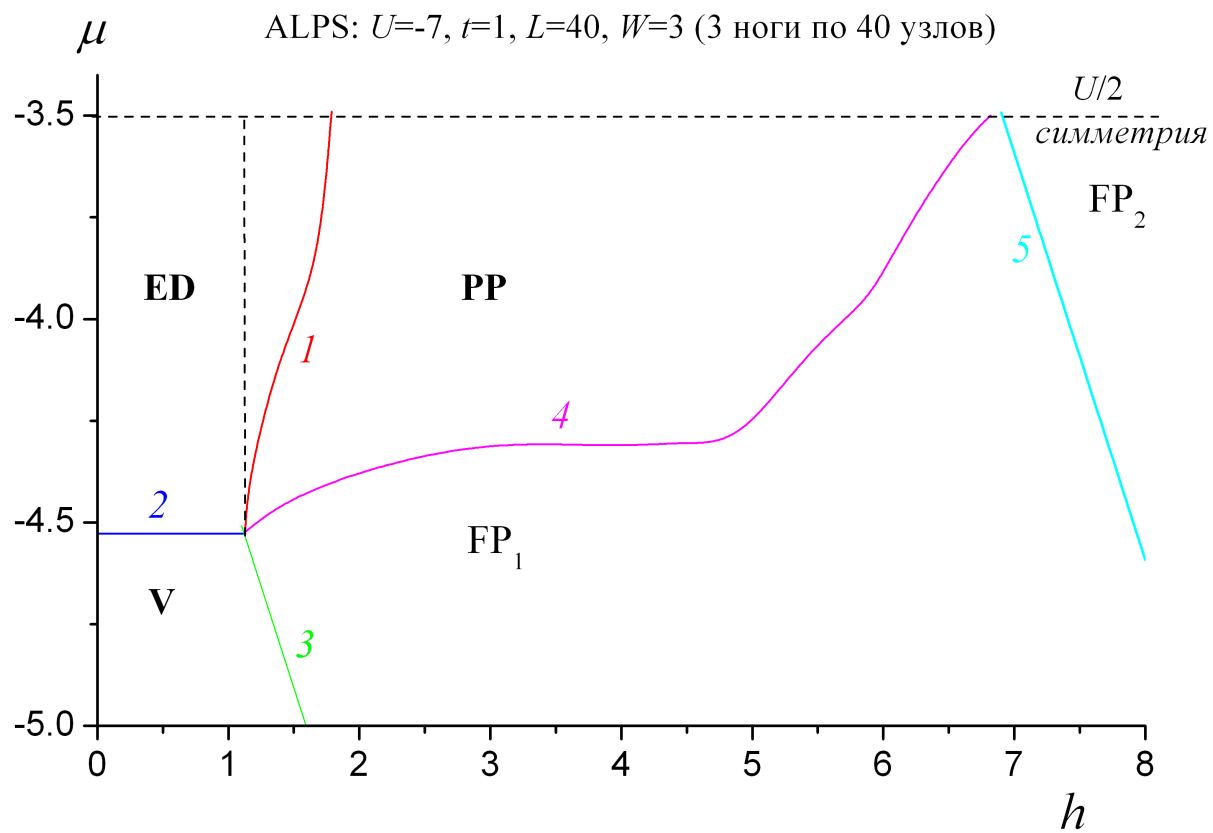


Рис. 7. Фазовая диаграмма трех связанных цепочек при  $U = -7$

## Заключение

В ходе выполнения магистерской диссертации было разработано программное обеспечение для расчета энергетического спектра одномерной цепочки спинов методом прямой диагонализации в рамках модели Хаббарда. Проведен расчет фазовой диаграммы одной, двух и трех связанных цепочек холодных атомов в рамках модели Хаббарда с притяжением для различных значений параметров модели. Выявлены основные закономерности эволюции фазовых диаграмм таких систем при изменении параметров модели и изменении числа взаимодействующих цепочек атомов. Стоит отметить, что расчет диаграммы трех связанных цепочек атомов уже выходил за первоначально выданное задание.

В работе методами численного моделирования проверен ряд теоретических результатов (выражение для энергии связи димера и уравнения для некоторых фазовых границ).

В работе получено, что топология фазовых диаграмм двух и более связанных цепочек атомов качественно не меняется при увеличении числа цепочек, но качественно отличается от топологии диаграммы одной цепочки. Этот результат получен в настоящей работе впервые и представляет реальную научную ценность.

Стоит отметить, что в ходе выполнения работы было разработано программное обеспечение, которое будет использовано в дальнейшем при обучении студентов магистерской программы “Материалы, приборы, нанотехнологии”.

## Библиографический список

1. Кашурников В.А., Красавин А.В., Вычислительные методы в квантовой физике: Учебное пособие. М.: МИФИ, 200. – 412с.
2. Hubbard J., Electron correlations in narrow energy bands-3, 4, "Proc. Roy. Soc. A", 1963, v. 276, p. 238; 1964, v. 281, p. 401; 1965, v. 285, p. 541
3. *F. Heidrich-Meisner, G. Orso, and A.E. Feiguin* "Phase separation of trapped spin-imbalanced Fermi gases in one-dimensional optical lattices", Phys. Rev. A 81, 053602
4. A. S. Holevo, Statistical definition of observable and the structure of statistical models. Rep. Math. Phys., 1985, v.22, N3.
5. Ландау Л.Д., Лифшиц Е.М., Краткий курс теоретической физики: Квантовая физика. М., 1972, .-368с.

# Приложение 1.

## Текст программы расчета энергии спиновом кластера

```
% Main module

t_init = cputime;
EnterData %preparing data for the solver
NewVariables %declaring new variables
Basis %generating basis
m_Ham = Hamiltonian;
eig(m_Ham)
t_fin = cputime;
fprintf(1,'Calculation time, min: %.0f\n',(t_fin-t_init)/60)
%      Result of a-operator act on a state vector
function f = a(type,number,state_vect)
%Arguments:_____
%type - type of operator: 1 - creation,-1 - annihilation;
%number - number of node state (to create or annihilate);
%state_vect - state vector (like [10100101110]).
%Returning value: new state vector.
%_____
N = size(state_vect,2); %size of the state vector
if iscorr(state_vect)
if appzer(type,number,state_vect) %apparent zeros
f = zeros(1,N);
else %the main procedure
f = (-1)^numofones(number,state_vect)*changestate(type,number,state_vect);
end
else
f = zeros(1,N);
disp('Incorrect format of the state vector!')
end
%      Checking format of the state vector
function f = iscorr(state_vect)
%Arguments:_____
%state_vect - state vector (like [10100101110]).
%Returning value: 1 - if the state vector is correct (contains just ones and zeros), 0 - otherwise.
%_____
N = size(state_vect,2); %size of the state vector
f = 1;
for i=1:N
if state_vect(i)~=0 && state_vect(i)~=1 && state_vect(i)~-1
f = 0;
break
end
end
%      Apparent zero
function f = appzer(type,number,state_vect)
%Arguments:_____
%type - type of operator: 1 - creation,-1 - annihilation;
%number - number of node state (to create or annihilate);
%state_vect - state vector (like [10100101110]).
%Returning value: 1 - if resulting vector is zero, 0 - otherwise.
%_____
f = 0;
switch type
case 1
if state_vect(number) == 1
f = 1;
end
```

```

case -1
if state_vect(number) == 0
f = 1;
end
otherwise
disp('Incorrect operator type!')
f = 1;
end
% Changing one state
function f = changestate(type,number,state_vect)
% Arguments
% type - type of operator: 1 - creation,-1 - annihilation;
% number - number of node state (to create or annihilate);
% state_vect - state vector.
% Returning value: state_vect with 1 state changed.
%
f = state_vect;
switch type
case 1 %create an electron
if sov(state_vect) >= 0
f(number) = 1;
else f(number) = -1;
end
case -1 %annihilate an electron
f(number) = 0;
otherwise
disp('Incorrect operator type!')
f = 0*state_vect;
end
% Number of ones before given number
function f = numofones(number,state_vect)
% Arguments:
% number - number of node state;
% state_vect - state vector.
% Returning value: number of ones before given number.
%
if number == 1
f = 0;
else
f = sum(state_vect(1:number-1));
end

% Result of a_plus_i_a_j-operator act on a state vector
function v_f = a_plus_i_a_j(i,j,v_F_q);
% Arguments:
% i - number of a_plus-operator;
% j - number of a_minus-operator;
% v_F_q - state vector (without spin, only positive components).
% Returning value: new state vector.
m = min(i,j); n = max(i,j);
k_perm = (-1)^sum(v_F_q(m:n-1));
if i < j
k_sign = 1;
else
k_sign = -1;
end
% a_j|v_F_q>:
k_zero = 1;
if v_F_q(j)
v_F_q(j) = 0;
else
k_zero = 0;
end
end

```

```

%a_plus_i|v_F_q>:
if v_F_q(i)
    k_zero = 0;
else
    v_F_q(i) = 1;
end
v_f = k_perm*k_zero*k_sign*v_F_q;

```

---

```

%           Generating basis
function Basis
global m_basis N_nodes N_part_up N_part_dwn

m_basis_up = GenBasis(N_nodes,N_part_up); %basis functions for spin-up electrons
N_up = size(m_basis_up,1); %number of them
m_basis_dwn = GenBasis(N_nodes,N_part_dwn); %... spin-down electrons
N_dwn = size(m_basis_dwn,1); %number of them
%General basis:
N_set = N_up*N_dwn;
k = 1;
m_basis = zeros(N_set,2*N_nodes);
for i=1:N_up
    for j=1:N_dwn
        m_basis(k,:) = [m_basis_up(i,:),m_basis_dwn(j,:)];
        k = k+1;
    end
end
end

```

---

```

%Entering all parameters from outer files
%Model parameters:_____
global eps0 t fl_bc N_nodes N_part_up N_part_dwn
load 'ModParams.dat' -ascii
eps0 = ModParams(1); t = ModParams(2); fl_bc = ModParams(3);
N_nodes = ModParams(4); N_part_up = ModParams(5); N_part_dwn = ModParams(6);
clear ModParams

```

---

```

%           Generating the basis set
function m_f = GenBasis(N_nodes,N_part)
%Arguments:_____
%N_nodes - number of nodes;
%N_part - number of particles.
%Returning value: m_f - matrix of basis set.
%_____
N_basis = nchoosek(N_nodes,N_part);
m_basis1 = nchoosek(1:N_nodes,N_part);
m_f = zeros(N_basis,N_nodes);
for i=1:N_basis
    for j=1:N_part
        m_f(i,m_basis1(i,j)) = 1;
    end
end
end

```

```

%           Hamiltonian matrix
function m_f = Hamiltonian
%Arguments:_____
% none
%Returning value: Hamiltonian matrix.
%_____
global m_basis N_nodes

```

```

N = size(m_basis,1); %dimension of state space

```

```

m_f = Ham_pot(N) + Ham_kin(N,m_basis(:,1:N_nodes),m_basis(:,N_nodes+1:2*N_nodes)) + ...
      Ham_kin(N,m_basis(:,N_nodes+1:2*N_nodes),m_basis(:,1:N_nodes));

```

```

%      Hamiltonian matrix (potential part)

```

```

function m_f = Ham_pot(N)

```

```

%Arguments:_____

```

```

% N - size of the matrix

```

```

%Returning value: potential part of Hamiltonian matrix.

```

```

%_____

```

```

global eps0 t m_basis N_nodes

```

```

m_f = zeros(N);

```

```

for i=1:N

```

```

    state_vect_up = m_basis(i,1:N_nodes); state_vect_dwn = m_basis(i,N_nodes+1:2*N_nodes);

```

```

    m_f(i,i) = eps0*state_vect_up*state_vect_dwn';

```

```

end

```

```

%      Hamiltonian matrix (kinetic part)

```

```

function m_f = Ham_kin(N,m_basis,m_basis_aux)

```

```

%Arguments:_____

```

```

% N - size of the matrix;

```

```

% m_basis - basis matrix (spin up), m_basis_aux - basis matrix (spin down) or vice-versa.

```

```

%Returning value: kinetic part of Hamiltonian matrix.

```

```

%_____

```

```

global eps0 t fl_bc

```

```

m_f = zeros(N);

```

```

N_nodes = size(m_basis,2); %number of nodes

```

```

arr_trr1(1:N_nodes) = 0; arr_trr2(1:N_nodes) = 0; %arrays for scalar products

```

```

for p=1:N

```

```

    for q=1:N

```

```

        if p~=q %just non-diagonal elements

```

```

            v_F_p = m_basis(p,:); v_F_q = m_basis(q,:);

```

```

            for i=1:N_nodes %sum over the nodes_____

```

```

                %A+(i)*A-(i-1):_____

```

```

                %Determining number of A-(i-1) operator:

```

```

                j = i - 1;

```

```

                fl_calc = 1; %flag indicator: to calc or not to calc the matrix element

```

```

                if j == 0

```

```

                    switch fl_bc

```

```

                        case 0 %the chain is not closed

```

```

                            fl_calc = 0;

```

```

                            arr_trr1(i) = 0;

```

```

                        case 1 %the chaine is closed

```

```

                            j = N_nodes;

```

```

                        otherwise

```

```

                            disp('Wrong type of boundary conditions!')

```

```

                        end

```

```

                    end

```

```

                %Calculating the first item of the matrix element:

```

```

                if fl_calc

```

```

                    v_tr = a_plus_i_a_j(i,j,v_F_q);

```

```

                    arr_trr1(i) = scpr([v_F_p,m_basis_aux(p,:)],[v_tr,m_basis_aux(q,:)]);

```

```

                end

```

```

                %A+(i)*A-(i+1):_____

```

```

                %Determining number of A-(i-1) operator:

```

```

                j = i + 1;

```

```

                fl_calc = 1;

```

```

                if j == N_nodes + 1

```

```

                    switch fl_bc

```

```

                        case 0 %the chaine is not closed

```

```

                            fl_calc = 0;

```

```

                            arr_trr2(i) = 0;

```

```

                        case 1 %the chaine is closed

```



```

        j = 1;
        otherwise
            disp('Wrong type of boundary conditions!')
        end
    end
    %Calculating the second item of the matrix element:
    if fl_calc
        v_tr = a_plus_i_a_j(i,j,v_F_q);
        arr_trr2(i) = scpr([v_F_p,m_basis_aux(p,:)],[v_tr,m_basis_aux(q,:)]);
    end
    %...A+(i)*A-(i+1)_____
end %sum over the nodes_____

m_f(p,q) = t*( sum(arr_trr1) + sum(arr_trr2) );
end
end
end

```

---

```

load 'H_example.dat' -ascii
m_Hex = H_example;
clear H_example

```

---

```

%           Main module
t_init = cputime;
EnterData %preparing data for the solver
NewVariables %declaring new variables
Basis %generating basis
m_Ham = Hamiltonian;
eig(m_Ham)
t_fin = cputime;
fprintf(1,'Calculation time, min: %.0f\n',(t_fin-t_init)/60)

```

---

```

%      Result of n-operator act on a state vector
function f = n(state_vect)
%Arguments:_____
%state_vect - state vector.
%Returning value: number of particles in this state.
%_____
f = sum(abs(state_vect));

```

---

```

%      Eigenvalue of n_spin_down-operator on a state vector
function f = n_spin_dwn_i(state_vect_spin,i)
%Arguments:_____
%state_vect_spin - state vector with spin;
%i - number of node.
%Returning value: number of particles in spin up mode in this state.
%_____
global N_nodes

state_vect_dwn = state_vect_spin(N_nodes+1:2*N_nodes);
f = abs(state_vect_up(i));

```

---

```

%      Eigenvalue of n_spin_up-operator act on a state vector
function f = n_spin_up_i(state_vect_spin,i)
%Arguments:_____
%state_vect_spin - state vector with spin;

```

```

%i - number of node.
%Returning value: number of particles in spin up mode in this state.
% _____
global N_nodes

```

```

state_vect_up = state_vect_spin(1:N_nodes);
f = abs(state_vect_up(i));

```

---

```

%New parameters:
% m_basis - basis of wave functions
global m_basis

```

---

```

%      Scalar product of two state vectors
function f = scpr(state_vect1,state_vect2)
%Arguments:_____
%state_vect1,2 - state vectors.
%Returning value: scalar product.
% _____
N1 = size(state_vect1,2); %size of the first state vector
N2 = size(state_vect2,2); %size of the second state vector
if N1 ~= N2
    disp('Different sizes of state vectors!')
    f = 0;
else %main procedure
    v1 = abs(state_vect1); v2 = abs(state_vect2);
    if sum(v1) == 0 && sum(v2) == 0 %if one of them =0
        f = 0
    else %procedure for nonzero vectors
        f = 1;
        for i = 1:N1
            if v1(i) ~= v2(i)
                f = 0;
                break
            end
        end
        if f
            f = f*sov(state_vect1)*sov(state_vect2);
        end
    end
end

```

---

```

%      Sign of vector
function f = sov(state_vect)
%Arguments:_____
%state_vect - state vector.
%Returning value: sign of first nonzero component of the vector.
% _____
N = size(state_vect,2); %size of the first state vector
f = 1;
for i = 1:N
    if state_vect(i) ~= 0
        f = sign(state_vect(i));
        break
    end
end

```